

Универзитет у Београду  
Математички факултет



Весна Д. Катанић

Креирање балансираних и повезаних кластера  
са циљем унапређења процеса доставе у  
проблему последње миље

мастер рад

Београд, 2022.

**Ментор:**

др Стефан Мишковић, доцент  
Универзитет у Београду, Математички факултет

**Чланови комисије:**

др Александар КАРТЕЉ, доцент  
Универзитет у Београду, Математички факултет

др Нина РАДОЛИЧИЋ МАТИЋ, доцент  
Универзитет у Београду, Математички факултет

Датум одбране: \_\_\_\_\_

*Захаљујем се ментору др Стефану Мишковићу на свим савешима, као и на великом сирћењу шокот израде рада. Велику захвалност дућујем и комананији Mily Technologies d.o.o., која је омоћућила израду овог рада и његово шесћирање на реалним подацима.*

**Наслов мастер рада:** Креирање балансираних и повезаних кластера са циљем унапређења процеса доставе у проблему последње миље

**Резиме:** Проблем последње миље (енг. *last mile delivery problem*) се бави достављањем пошиљака од транспортног центра до крајњег корисника. Овај проблем је, са порастом броја наручених пакета преко интернета, постао водећи проблем у логистици. Решавањем проблема последње миље подиже се ефикасност целокупног ланца испоруке, што директно утиче на задовољство крајњих корисника. Идеја да се сваком куриру додели регион у коме ће свакодневно радити се одавно користи и постала је незаменљиви део многих логистичких решења. Међутим, са порастом броја пакета које је потребно доставити и честим осцилацијама у броју расположивих курира, региони генерисани ручно на основу знања и интуиције појединаца, значајно су ограничили ефикасност доставе у последњој миљи.

У овом раду биће приказане методе помоћу којих је могуће вршити аутоматско креирање региона са равномерном расподелом пошиљака у њима. За конструкцију региона у овом раду искоршћене су четири методе: стандардни *k-means* алгоритам са локалном претрагом, стандардни *k-means* алгоритам са *VNS* методом, вишеструки *k-means* алгоритам са локалном претрагом и вишеструки *k-means* алгоритам са *VNS* методом.

Поред метода и имплементације, биће представљено и поређење ефикасности ових приступа. Приказано је поређење избалансираности решења и поређење оптималних рута креираним како на регионима генерисаним методама описаним у овом раду тако и са тренутним решењем које користи једна логистичка компанија. Вишеструки *k-means* алгоритма са *VNS* методом, дао је најбоље резултате по питању избалансираности кластера, које може донети побољшање у ефикасности доставе од 15% у односу на постојеће решење. На основу резултата овог рада, посматрана логистичка компанија је кренула у процес имплементације приказаног решења.

**Кључне речи:** кластероване, оптимизација, достава, логистика, метода променљивих околина, проблем последње миље

# Садржај

<b>1</b>	<b>Увод</b>	<b>1</b>
<b>2</b>	<b>Опис проблема</b>	<b>5</b>
2.1	Преглед литературе . . . . .	5
2.2	Мотивација . . . . .	7
<b>3</b>	<b>Примена алгоритама</b>	<b>8</b>
3.1	Трансформација и припрема улазних података . . . . .	9
3.2	Генерисање почетног решења . . . . .	11
3.3	Поправка решења . . . . .	16
<b>4</b>	<b>Експериментални резултати</b>	<b>24</b>
4.1	Стандардни <i>k-means</i> алгоритам са локалном претрагом . . . . .	25
4.2	Стандардни <i>k-means</i> алгоритам са <i>VNS</i> методом . . . . .	29
4.3	Вишеструки <i>k-means</i> алгоритам са локалном претрагом . . . . .	32
4.4	Вишеструки <i>k-means</i> алгоритам са <i>VNS</i> методом . . . . .	36
4.5	Поређење добијених резултата са резултатима посматране логистичке компаније . . . . .	39
<b>5</b>	<b>Закључак</b>	<b>45</b>
	<b>Библиографија</b>	<b>47</b>

# Глава 1

## Увод

Проблем последње миље (енг. *last mile delivery problem*) се бави достављањем пошиљака од транспортног центра до крајњег корисника [3]. Односи се на бројне неефикасности последњег корака у ланцу снабдевања, током којег се купцу доставља поручена роба. Решавањем проблема последње миље подиже се ефикасност целокупног ланца испоруке, што директно утиче на задовољство крајњих корисника. Ово је такође и најскупљи и најдужи корак у процесу испоруке робе и често чини и до 50% укупних трошкова. Адекватна расподела посла, благовремено извештавање корисника о тренутку доставе пакета и проналажење оптималних путања за доставу су саставни делови овог проблема.

Основни концепт за доставу пошиљака у проблему последње миље, који се примењује широм света, базира се на употреби доставних возила, где свако возило вози по један курир. Курир, сваког дана добија одређену количину пошиљака које пакује у комби. Затим се вози до сваке адресе, и доноси пакет до врата корисника. Поред овог, стандардног начина доставе, последњих година ради се и на проналажењу нових алтернативних начина доставе, поготово за ужи центар града, где саобраћајне гужве значајно отежавају процес доставе. Један такав пример је достава помоћу транспортних бицикала (енг. *cargo bikes*). Основна предност доставе помоћу бицикала је могућност приступа зонама у којима је забрањен саобраћај моторних возила у центру града, као и смањена емисија штетних гасова. Међутим, како је број пакета које је могуће упаковати на бицикл, значајно мањи него у случају комбија, потребно је да се бицикли више пута допуне у складишту или на некој за то посебно одређеној локацији. Достава транспортним бициклима се успешно примењује у Бечу [2]

и Кембриџу [18]. Један нови вид доставе је комбиновање доставе комбијима и бициклима. Достава пакета се врши помоћу комбија од транспортног центра до обода града, где део пакета узимају бицикли и врше њихову доставу. За то време се комби помера на други део града, одакле се опет наставља достава уз помоћ бицикала. Овим приступом се повећава ефикасност доставе, јер бицикли не морају да се враћају до фиксних центара у граду по нове пакете.

Други мало другачији начин решења проблема доставе је употреба пакетомата. Пакетомати представљају за то посебно дизајниране контејнере у које курири остављају пакете, који корисници могу да покупе било кад у току дана. Овим приступом је значајно смањена километража коју курир треба да пређе. Такође је решен и проблем у случају да корисник није код куће у тренутку кад курир покуша доставу пакета.

Поред ових решења која су већ ушла у употребу, разматра се и аутономна достава пакета помоћу дрона и робота [1]. Употреба дрона за доставу пакета су успешно тестирале компаније *DHL*, *Amazon* и *Alibaba*. Како су дрони углавном ограничени на достављање једног не претерано тешког пакета, достављање велике количине пакета би захтевало и значајан број дрона, па остаје отворено питање да ли би њихова примена допринела смањењу трошкова доставе. Употребу робота, који користе пешачке стазе за кретање, тестирају компаније попут *Starship Technologies*, *Robby* и *Amazon Scout*. Предност робота у односу на дроне је што могу да доставе значајно теже пакете, као и могућност приступа областима у којима је забрањено летење дрона, као на пример у близини аеродрома.

Иако се већина нових истраживања базирана на самом начину доставе, за успешност целокупног ланца неопходно је решити све његове делове, од расподеле пошиљака, преко проналажења рута, до благовременог обавештавања корисника о тренутку доставе.

Како у Србији тако и у свету, брз раст интернет куповине ставља велики притисак на доставну инфраструктуру, притом смањујући приходе компанија због недостатка поверења купаца и растућих логистичких трошкова. Тај притисак је додатно појачан КОВИД-19 пандемијом. Поред пораста броја пакета, компаније су суочене и са недостатком курира и њиховим честим изостанцима.

У овом раду бавићемо се проблемом оптималне расподеле посла на примеру једне логистичке компаније. Наизглед, примена рутирања за расподелу по-

сла и проналажење рута даје оптимално решење. Међутим, велика количина података, временска сложеност и комплексност израчунавања ове алгоритме чине непримењивим у пракси. Једно широко прихваћено решење овог проблема у логистици је подела области на регионе. Идеја да се сваком куриру додели један регион у коме ће свакодневно радити се одавно користи и постала је незамењиви део многих логистичких решења. Овој приступ куририма је омогућио да стекну нека специфична знања везана за свој регион, попут тога, где је могуће паркирати возила, које су улице неприступачне комбијем и слично. Један проблем ручно генерисаних региона је и неуравнотежена количина посла по куриру. Како број пошиљака по региону варира из дана у дан, често се дешава да курир добије или превише или премало посла за тај дан. Већина курирских служби данас користе регионе како би одредили који курир доставља које пошиљке. Међутим, мењање региона је компликован процес који се ради једном у два, три месеца, углавном након што њихови недостаци направе озбиљан проблем. Са порастом броја пакета и честим осцилацијама у броју расположивих курира, региони генерисани ручно на основу знања и интуиције појединаца, значајно су ограничила ефикасност доставе у последњој миљи.

Циљ овог рада је да се омогући свакодневно аутоматско креирање региона који ће омогућити равномерну расподелу посла за курире и унапредити њихову ефикасност у последњој миљи. У литератури овај проблем је познат као *districting* проблем. *Districting* проблем [11] је проблем груписања мањих географских области у веће географске кластере, дистрикте, тако да су они избалансирани, повезани и компактни. Избалансираност представља жељу да су дистрикти истих величина, на пример у нашем случају по броју пошиљака. Дистрикт је географски компактан ако је неког правилног облика, без истезања и рупа. Најчешћи примери мањих области које се групишу су, корисници, улице или области које припадају одређеним поштанским бројевима. Ови проблеми се најчешће примењују у случају креирања политичких дистрикта (енг. *political districting*), дизајна области продаје (енг. *sales territory design*) и дизајна области за теренске раднике (енг. *service districting*).

Креирање политичких дистрикта представља поделу града или државе на изборне јединице. Да би се избегла политичка неравноправност, многе државе су креирале посебне независне комитете, који се баве одређивањем услова под којим се креирају политички дистрикти. У зависности од државе користе се



различити критеријуми, јер на политичкој сцени не постоји концензус, као ни закон или географска одредица на основу којих треба креирати критеријум, за процес стварања дистрикта. Креирање политичких дистрикта неопходно је поновити свремена на време углавном након 4, 5 година како би се у обзир узела промене у структури становништва.

У дизајну области продаје зарада се остварује на основу продатих производа. Циљ је сваком продавцу доделити потенцијалне муштерије тако да сви продавци имају једноко време путовања и једнаку могућност зараде, а да притом свака муштерија има једног, одређеног продавца. Три стандардна критеријума за креирање дистрикта у овом случају су избалансираност, повезаност и компактност. У овом случају неопходно је балансирање по више критеријума.

Иако широко распрострањени, ови проблеми немају јасну математичку дефиницију, већ се тачнија дефиниција услова мења од проблема до проблема пре свега на основу доступних информација у конкретном случају.

# Глава 2

## Опис проблема

### 2.1 Преглед литературе

Проблем последње миље односи се на све логистичке акције повезане са доставом поштиљака до крајњих корисника [3]. Као резултат наглог пораста трговине преко интернета у последњих пар година, проблем последње миље је постао водећи проблем у истраживањима везаним за логистику, како градских, тако и приградских насеља [21]. Boysen et al. [3] је шематски представио најважније концепте који се користе у проблему последње миље. Најинтересантнији део овог рада су тренутни изазови са којима се суочавају различити приступи решавању проблема последње миље. Неки од њих су:

- Пораст обима посла. Нови токови урбанизације и пораст е-комерца стварају велике изазове за логистичке компаније. Они сматрају да ће комбинација, географске концентрације популације у великим градовима, и пораста броја наручених пакета преко интернета, по особи, проузроковати стални раста броја поштиљака које курир треба да достави.
- Цена, овде се ослањају на истраживање спроведено у Финској, које показује да традиционални начин доставе, помоћу доставних возила, кошта између 2 и 6 евра по достави, у зависности од густине насељености.
- Притисак на време доставе. Како очекивања крајњих корисника расту, ставља се све већи притисак на компаније да раде доставу поштиљака истог или наредног дана у односу на дан крирања поштиљке. Ово резултује у тесним роковима које курири требају да испоштују. Променљива

дистрибуција пошиљака, која варира у зависности од доба године или дана у недељи, додатно компликује процес, захтевајући велику флексибилност у кратком временском периоду.

Даље, представљено је и истраживање о коришћењу рутирања у стандардном начину доставе, помоћу доставних возила. Овде су указали на неке од проблема рутирања. Указали су на истраживање спроведено у Штутгарту где је показано да иста рута може да захтева и до 43% више времена, у зависности да ли се на пут креће у среду у 9:30, или у суботу у 12:00. Овим је указано да избор тренутка поласка курира може да има занчајан утицај на време потребно да се обави достава пошиљака.

Највећи број компанија које се баве доставом пошиљака у проблему последње миље, избегавају свакодневно креирање оптималних рута за посматрани скуп клијената. Чешће се области у којима врше доставу деле на мање регионе, који остају непромењени дужи временски период. Сваком региону додељен је један курир, који у њему врши доставу. Путања којом ће се вршити достава се креира по региону, за скуп клијената који припадају том региону. Креирање путања се врши или мануелно, на основу специфичних знања курира, или уз помоћ техничких решења. Проблем креирања региона спада у групу *districting* проблема. *Districting* проблеми су широко распрострањени и честа су тема истраживања. Међутим, већина истраживања се фокусира на друге примене ових проблема, као што су креирање политичких дистрикта [11] или дизајн области продаје [17]. Највећи изазов са којим се сусрећу *districting* проблеми, примењени на проблем последње миље, је потреба да се региони креирају без неког детерминистичког знања о скупу клијената за које се креирају. Такви региони треба да буду компактни, како руте криране унутар њих не би биле предугачке. Даље, посао треба да буде равномерно распоређен по регионима, при чему вероватноћа да се у неком региону препуни доставно возило треба да остане мала. У наставку ће бити кратко описани неки радови који су успешно креирали регионе, под горе наведеним условима. Naugland et al. [8] користи алгоритам који се састоји из две фазе, прво клијенте дели по регионима, а затим се траже оптималне руте за сваки регион. Примењена је табу претрага за минимизовање целокупне цене рута. Сличан проблем је представио Lei et al. [13], назван *the vehicle routing and districting problem with stochastic customers*. Опет примењене су две фазе, прво креирање дистрикта, а затим рутирање. Са циљем да минимизују функцију циља, кре-

ирану комбиновањем цене возила, цене рутирања и компактности региона, представљена је метехеуристика која ради претрагу на великом скупу суседа.

## 2.2 Мотивација

Компанија Mily Technologies d.o.o. је српска *startup* компанија која нуди софтверске производе и услуге за оптимизацију процеса доставе. Један од проблема са којим се компанија сусрела је да курири нису у могућности да доставе све пошиљке које су пристигле претходног дана.

Све пошиљке пристигле одрђеног дана, се скупљају у складишту и деле на регионе. Региони представљају делове града, сличне поштанским бројевима. За разлику од поштанских бројева, региони се мењају на свака два, три месеца са циљем да омогуће ефикаснију доставу. Овај процес се одвија ручно на основу знања и интуиције појединаца. Сваки курир ради у једном региону и има задатак да достави све пошиљке пристигле у његов регион.

На основу анализе расположивих историских података, утврђено је да региони креирају врло неравномерну расподелу пошиљака по куриру. Ово за резултат има курире који не успеју да доставе све задужене пошиљке, а са друге стране неки посао заврше прерано.

Једно од питања које се поставља је да ли би равномерна расподела пошиљака по регионима могла да побољша ефикасност доставе. Да би равномерна расподела пошиљака по регионима била могућа, неопходно је да се региони мењају свакодневно, на основу пошиљака пристиглих на тај дан.

Идеја је да се омогући свакодневно аутоматско креирање балансираних региона, који би обезбедили да сви курири достављају једнак број пошиљака. Проблем креирања балансираних региона може се дефинисати на следећи начин:

За дати скуп пошиљака, потребно је дату област поделити на  $k$  подобласти (региона), тако да сваки регион има једнак број пошиљака. При томе сваки регион треба да буде повезан и компактан. Такође, облици тих региона треба да буду што је могуће правилнији, тако да буду примењиви у пракси.

## Глава 3

# Примена алгоритама

У овом поглављу, биће представљене четири методе употребљене за креирање избалансираних региона. Све методе се састоје из три основна корака:

- трансформација и припрема улазних података,
- генерисање почетног решења,
- итеративно унапређење почетног решења.

Методе се разликују по алгоритмима употребљеним за генерисање почетног решења и алгоритмима употребљеним за итеративно унапређење почетног решења, и оне су:

- стандардни *k-means* алгоритам са локалном претрагом,
- вишеструки *k-means* алгоритам са локалном претрагом,
- стандардни *k-means* алгоритам са *VNS* методом,
- вишеструки *k-means* алгоритам са *VNS* методом.

У наставку ће бити детаљно описани сви кораци при решавању проблема, као и алгоритми који су употребљени.

### 3.1 Трансформација и припрема улазних података

Први корак у решавању проблема је употреба  $H3$  [20] хексагона.  $H3$  је гео-просторни хијерархијски систем индексирања Земље, који је дели на шестоугоане ћелије. Хексагони су се испоставили као одличан избор за потребе анализе података, јер сваки има тачно 6 суседа и раздаљина између центра једног хексагона и свих његових суседа је иста. Овај систем индексирања има 16 нивоа, индексираних од 0 до 15. Сваки следећи ниво основни хексагон дели на 7 мањих хексагона. Ниво 0 има 110 хексагона просечне површине  $4357449 \text{ km}^2$ , док ниво 15 има укупно  $569707381193150$  хексагона просечне површине  $0.000000895 \text{ km}^2$ .



Слика 3.1:  $H3$  основни хексагон подељен на 7 мањих

У овом раду  $H3$  је искоришћен за представљање области и региона, пре свега због своје једноставности, могућности да се лако одреде границе области и суседи, као и припадност тачке одређеној области.

Како се претпоставља да је област задата уз помоћ  $H3$  хексагона, први корак у припреми података је груписање пошиљака у хексагоне. Тиме се наш проблем своди на проблем груписања хексагона. Овај корак омогућио



Слика 3.2: Пример области представљене уз помоћ  $H3$

је стабилније време извршавања које не зависи од броја тачака, већ само од величине области. Такође, овим су уклоњене све мање неправилности у подацима настале непрецизношћу *GPS* уређаја. Хексагони су искоришћени и за дефинисање повезаности. За потребе овог рада тачке се сматрају повезаним ако припадају суседним хексагонима. На слици 3.2 је приказан пример области задате уз помоћ  $H3$  хексагона.

Следећи корак је инспирисан радом [4] и састоји се у конструкцији графа повезаности хексагона. За потребе овог рада граф је конструисан тако што су центри хексагона претворени у темена, а два темена су повезана граном, само ако припадају суседним хексагонима. За разлику од суседних  $H3$  хексагона, примена графа повезаности омогућује једноставну промену повезаности као на пример у случају проласка неке реке или планине кроз посматрану област.

## 3.2 Генерисање почетног решења

Следећи корак при решавању овог проблема је генерисање почетног решења, које ће се затим итеративно поправљати док се не добије крајње решење. Почетно решење генерисано је помоћу алгоритама за кластеровање тачака примењеним на  $H3$  хексагонима.

Кластеровање је скуп техника које се користе за поделу података на групе или кластере. Кластери представљају групе података у којима су сви елементи једне групе сличниј елементима те групе, него елементима других група. Алгоритми за кластеровање се деле у три основне групе:

- партиционо кластеровање,
- хијерархијско кластеровање,
- кластеровање засновано на густини.

Партиционо кластеровање (енг. *partitional clustering*) [5] дели тачке у групе међу којима нема преклапања. Дакле, ниједна тачка не може да буде садржана у више кластера, и сваки кластер мора да има бар једну тачку. Ови алгоритми као улазни параметар захтевају број кластера на које је потребно поделити податке. Углавном користе итеративни процес за додељивања тачака кластерима.

Хијерархијско кластеровање (енг. *hierarchical clustering*) [10] одређује кластере креирањем хијерархије, применом метода одоздо нагоре или одозго надоле. Ако се примењује метод одоздо нагоре ови алгоритми у сваком кораку групишу две најближе тачке. Процес се понавља док се не добије једна тачка. Ако се примењује техника одозго надоле креће се од једног кластера који се у сваком кораку даље дели, док све тачке не постану засебни кластери. Овим процесима се креира структура података слична стаблу названа дендограм. Алгоритми хијерархијског кластеровања, такође захтевају број кластера као улазни параметар. Резултат се добија пресеком дендограма на одговарајућој дубини.

Кластеровање засновано на густини (енг. *density-based clustering*) [12] групише тачке на основу густине података у неком региону. За разлику од већине алгоритама за кластеровање ови алгоритми не захтевају број кластера као улазни параметар. Користе се када је потребно добити кластер који нису



сферног облика. Кластероване засновано на густини није прикладно ако се користе високодимензионални подаци.

За генерисање почетног решења тестиране су следеће две варијанте *k-means* алгоритама:

- стандардни *k-means*,
- вишеструки *k-means*.

*K-means* [14] алгоритам је најједноставнији алгоритам ненадгледаног учења. Припада групи партиционог кластероване. Овај алгоритам групише тачке у односу на њихово одстојање од центра кластера. За резултат даје кластере у којима су све тачке кластера ближе центру свог кластера, него центрима других кластера.

### Стандардни *k-means*

*K-means* алгоритам представља најједноставнији и најчешће коришћен алгоритам за кластероване. Проналази унапред задат број кластера у вишедимензионом скупу података. Састоји се од следећих корака:

1. Креће се са  $k$  насумично изабраних тачака, које представљају центре кластера.
2. Свака тачка се додељује кластеру чијем је центру најближа.
3. Поново се израчунавају центри за све кластере који су добили нове тачке.
4. Понављају се кораци 2 и 3 докле год се померају центри кластера.

*K-means* алгоритам има много примена, неке од њих су у: класификацији докумената, препознавању опасних места, подели корисника, препознаванњу превара.

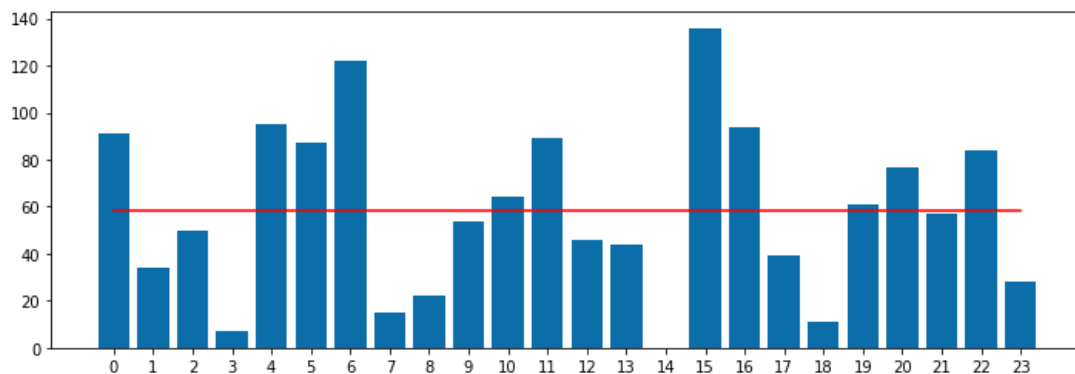
Како је почетно решење генерисано насумичним избором тачака, овај алгоритам може дати различите резултате за исте улазне податке.

Предност *k-means* алгоритама је у његовој једноставности и лакој имплементацији. Такође, лако је примењив и на велике скупове података, гарантује конвергенцију и лако се прилагођава новим примерима.



Слика 3.3: Пример примене стандардног *k-means* алгорита за генерисање почетног решења

Такође, решење има веома правилне облике. Међутим, проблем је што у обзир не узима број тачака у кластерима, па почетно решење добијено применом овог алгорита може имати врло неповољну тј. неуниформну расподелу. На слици 3.3 је приказан изглед почетног решења генерисаног применом стан-



Слика 3.4: Број пошљака по кластерима генерисаним применом стандардног *k-means* алгорита за добијање почетног решења

дардног *k-means* алгорита за генерисање почетног решења, а на слици 3.4 расподела пошљака по кластерима, након примене стандардног *k-means* алгорита за генерисање почетног решења. У овом раду искоришћена је имплементација *k-means* алгорита која се налази у склопу *scikit-learn* библиотеке [16] за *Python* програмски језик.

## Вишестуки *k-means*

Уместо употребе *k-means* алгоритма за добијање кластера, алгоритам је употребљен за добијање  $N \cdot 16/100$  кластера, где  $N$  представља укупан број хексагона у области. Новодобијени број кластера изабран је експерименталним путем као број који се најбоље показао на највећем скупу тестираних примера. Прво је пробано са  $k^2$  кластера. Међутим, ова варијанта је давала добре резултате у једном случају и лоше у другом, у зависности у ком односу су број кластера и број хексагона  $N$ , из којег се састоји посматрана област. Из тог разлога новодобијени број креиран је само на основу броја хексагона у области. Велики број новодобијених кластера у поређењу са бројем хексагона од којих се састоји посматрана област, давао је облике који личе на линије чијим спајањем су добијани врло неправилни облици. У случају да је број новодобијених кластера превише мали, кластери креирани овом методом би били превелики и нарушили би избалансираност овако генерисаног решења. Након добијања  $N \cdot 16/100$  кластера, ти кластери се спајају док се поново не добије  $k$  кластера. Спајање два кластера се врши ако је њихова укупна тежина мања или једнака од иделане тежине кластера увећане за константу.

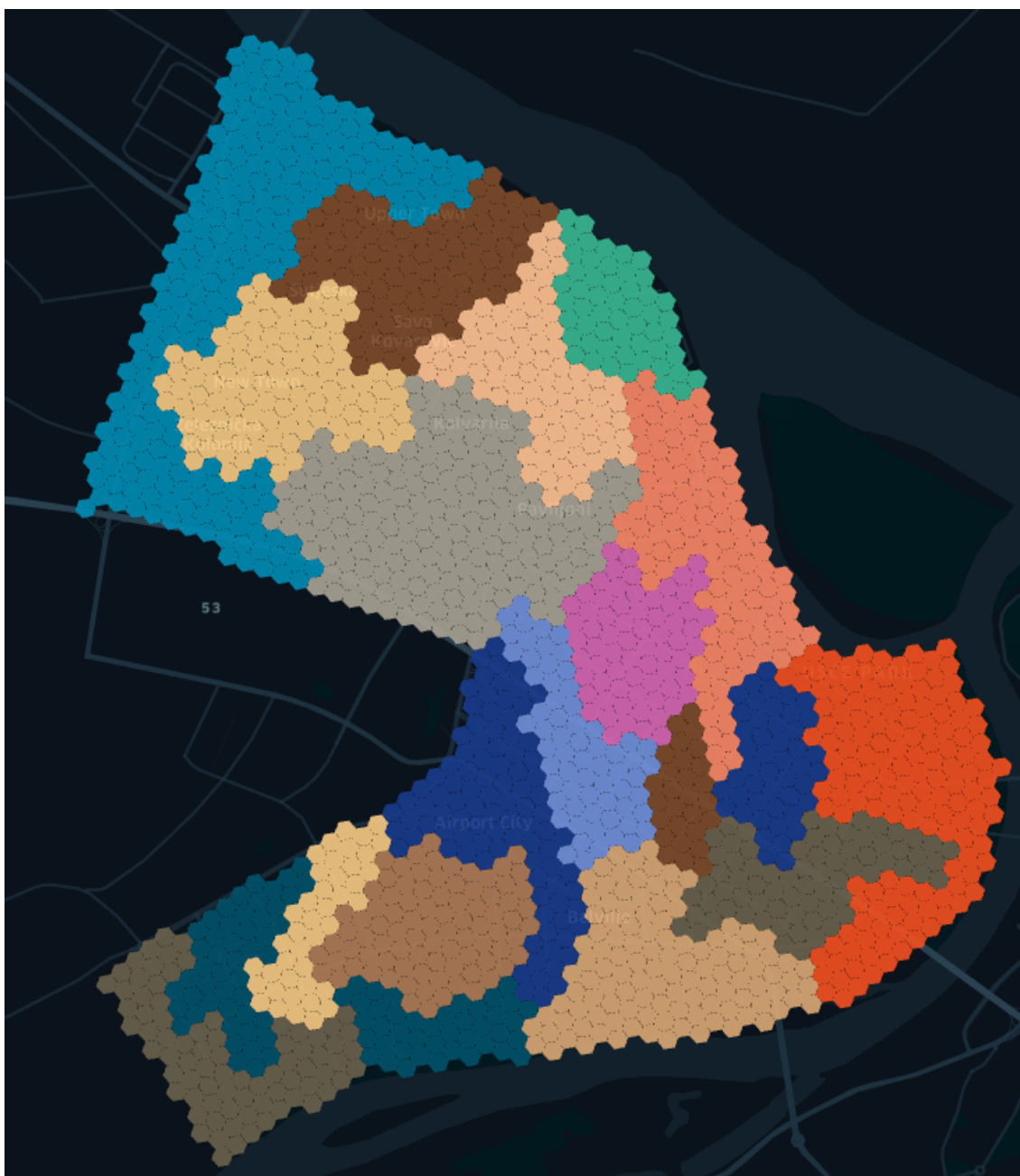
Прво се покушава спајање кластера који имају најмањи број суседа. Уколико више није могуће вршити спајања под постојећим условима, а није достигнут циљани број кластера, константа се повећава два пута и процес се понавља. Овим методом остварен је задовољавајући компромис између избалансираности почетног решења, брзине доласка до решења и изгледа кластера. На слици 3.5 дат је изглед области подељене на  $N \cdot 16/100$  кластер, а на слици 3.6 изглед исте области након што су кластери спојени. На слици 3.7 приказан је график избалансираности почетног решења добијеног применом вишеструког *k-means* алгоритма.



Слика 3.5: Пример примене вишестуког *k-means* алгорита пре спајања

### 3.3 Поправка решења

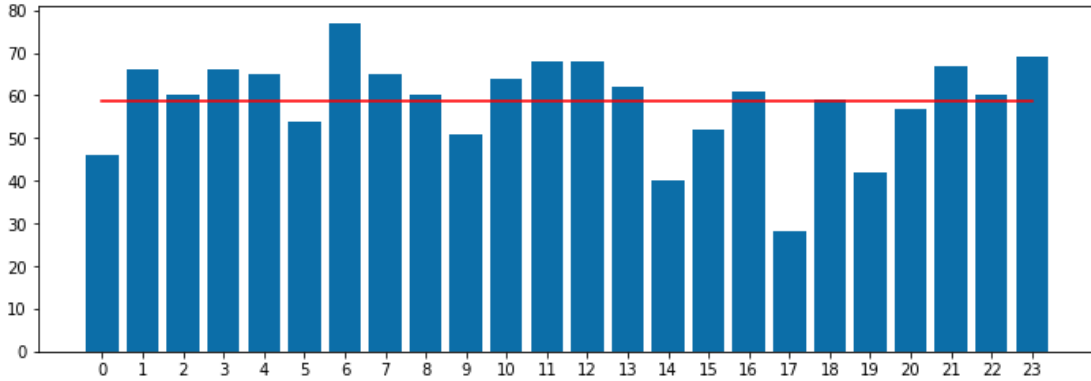
Након генерисања почетног решења следећи корак је итеративно поправљање тог решења. За поправљање почетног решења искоришћена је локална претрага и метода променљивих околина, које ће бити детаљније описане у наставку овог рада.



Слика 3.6: Пример примене вишестуког *k-means* алгорита након спајања

## Локална претрага

Локална претрага (енг. *local search*) [9] је хеуристичка метода за решавање комбинаторних оптимизационих проблема. У случају локалне претраге, свакој инстанци додељен је коначан скуп могућих решења, при чему свако решење има придружену одговарајућу цену. Циљ је наћи решење која има



Слика 3.7: Број пошиљака по кластерима генерисаним применом вишеструког  $k$ -means алгорита за добијање почетног решења

најмању (или највећу) могућу цену. Како би се дошло до траженог решења алгоритам посматра суседе тренутног решења. Суседи представљају скуп могућих решења који су у неком смислу близу посматраног решења. Алгоритам локалне претраге креће од независно генерисаног почетног решења, и у сваком кораку тренутно решење замењује суседним решењем које има бољу вредност, докле год такво решење постоји. Када такво решење више не постоји пронађено решење се сматра локалним оптимумом.

У овом раду локалну претрагу крећемо од почетног решења генерисаног алгоритмима за кластеровање. Прво посматрамо кластере најмање тежине. Затим се итерира кроз све његове суседне хексагоне и тражи се најбољи потез пребацивања хексагона из једног кластера у други. Током процеса конструкције кластера, један од циљева је да сваки кластер добије тежину што ближе циљаној тежини. Циљана тежина задата је са

$$target(C) = |C| * WAvg$$

где је  $WAvg$  сума свих тежина хексагона подељена са бројем кластера  $k$ .  $weightC(s)$  представља суму тежина свих хексагона у кластеру  $s$ , док  $wm(s)$  задата са

$$wm(s) = |weightC(s) - WAvg|$$

представља удаљеност кластера  $s$  од циљане тежине. Вредност потеза којим се хексагон  $i$  пребациује из кластера  $r$  у кластер  $s$  задат је са:

$$weightValueOfMove(r, s, i) = wm(r) + wm(s) - wm(newS) - wm(newR),$$

при чему је  $newS$  кластер добијен од кластера  $s$  додавањем хексагона  $i$ , а  $newR$  кластер добијен од кластера  $r$  избацивањем хексагона  $i$ . Даље, сматра се да је потез пребацивања хексагона  $i$  из кластера  $r$  у кластер  $s$  поправљајући ако задовољен један од следећих услова:

- $weightValueOfMove(r, s, i) > 0$
- $weightValueOfMove(r, s, i) = 0$  и  $weightC(r) > WAvg$  и  $weightC(s) < WAvg$
- $weightValueOfMove(r, s, i) = 0$  и  $weightC(r) \leq WAvg$  и  $weightC(r) - weight(i) > weightC(s)$

Други циљ конструкције кластера је да се очува компактност и правилност. Сматра се да је кластер компактан ако у њему не постоји хексагон чијим избацивањем кластер постаје неповезан. Да би се остварио овај циљ, потези који кваре компактност сматрају се невалидним потезима. Дакле, да би се потез пребацивања хексагона  $i$  из кластера  $r$  у кластер  $s$  сматрао изводљивим, потез мора да задовољи следеће критеријуме:

- да су кластери  $s$  и  $r$  повезани након пребацивања,
- да је потез поправљајући,
- да потез не креира некомпактне кластере.

Даље, од свих валидних потеза бира се онај који ће најмање повећати број хексагона са којим се посматрани кластер граничи. Уколико има више таквих опција бира се онај хексагон који има најбољи однос броја суседних хексагона у кластеру  $r$  и кластеру  $s$ . Овим поступком се форсира померање кластера постепено корак по корак како би се у што већој мери очувала компактност кластера.

Локална претрага је једноставан алгоритам који је лако мењати и прилагодити конкретном проблему. Како овај алгоритам проналази локалан оптимум, решење добијено овим алгоритмом може идаље давати поприлично неповољну, неуниформну расподелу, као што ће бити приказано у поглављу 4.



## Метода променљивих околина

Метода променљивих околина (енг. *Variable neighbourhood search - VNS*) [7] је уопштење локалне претраге, у којем се врши систематско претраживање више околина почетног решења. Метода променљивих околина се базира на следећим принципима:

- Локални минимум за један тип околина не мора нужно да буде локални минимум за други тип околина.
- Глобални минимум представља локални минимум за све типове околина.
- За многе проблеме у пракси, локални минимум више типова околина су релативно близу један другог.

Основни кораци методе су:

- Генерисање насумично изабраног решења  $x'$  у околини  $l_k$  од тренутно посматраног решења  $x$ .
- Примена методе локалне претраге на решење  $x'$  за добијање локалног минимума  $x''$ .
- Ако је новодобијено решење  $x''$  боље од тренутно посматраног  $x$ , прелази се на посматрање новодобијеног решења  $x = x''$  и процес се понавља са првим разматраним типом околина  $k = 1$ . У супротном се прелази на следећу околину  $k = k + 1$ .

У литератури постоји неколико варијанти основног алгорита, међу којима се најчешће срећу редукована, основна и уопштена метода променљивих околина.

Основна карактеристика редуковане методе променљивих околина је изостанак корака локалне претраге. У овој методи се бира насумично решење из посматране околине тренутног решења. Ако је новоизабрано решење боље од тренутног, претрага се наставља од њега. У супротном, повећава се разматрана околина.

Код основне методе променљивих околина се прво бира насумично решење  $x$  у посматраној околини, на које се затим примењује метода локалне претраге, како би се добио локални минимум. Ако је новодобијено решење

боље од тренутног, претрага се наставља од њега. У супротном, повећава се разматрана околина.

Код уопштене методе променљивих околина користи се метода променљивог спуста уместо локалне претраге. Омогућава коришћење два различита типа околине.

У овом раду примењена је уопштена *VNS*, која се састоји из два корака, размдравања решења и локалне претраге. Примена *VNS* алгорита, креће од почетног решења генерисаног неком од претходно описаних метода за генерисање почетног решења. Псеудокод примењеног алгорита дат је у наставку:

```

f ← solution_level(x)
while i < 60 do
    l ← 1
    while l < lmax do
        x1 ← shake_solution(l)
        l1 ← 1
        while l1 < l1max do
            x2 ← improve_solution(l1)
            if solution_level(x2) < solution_level(x1) then
                x1 ← x2
                l1 ← 1
            else
                l1 ← l1 + 1
            end if
        end while
        if solution_level(x2) < solution_level(x) then
            x ← x2
            l ← 1
        else
            l ← l + 1
        end if
    end while
    i ← i + 1
end while
f = solution_level(x)

```

Метода *solution\_level* за тренутно решење *x* дефинисана је са:

$$\sum_{n=1}^k wm(s_k),$$

где је  $s_k$  кластер  $k$ , посматраног решења  $x$ .

Метода *shake\_solution*( $l$ ) представља корак размрдавања посматраног решења  $x$  у околини  $l$ . У овом кораку се бира насумичан кластер и од његових суседних хексагона се бира насумични хексагон који се убацује у кластер. Овај процес се понавља одговарајући број пута у зависности од посматране околине. Дакле околину  $l_1$  тренутног решења добијамо од постојећег решења

избором насумичног кластера и избором насумичног хексагона међу његовим суседима и убацивањем изабраног хексагон у изабрани кластер.

Метода *improve\_solution(l)* представља корак локалне претраге. У овом кораку се насумично бира кластер и врши се његово поправљање методом локалне претраге, докле год је то могуће. Овај процес се понавља одговарајући број пута у зависности од посматране околине. Дакле, у овом случају околину  $l_1$  добијамо насумичним избором кластера, који се затим поправља локалном претрагом.

Применом методе променљивих околина решен је проблем локалног оптимума који се јавља у случају локалне претраге. Међутим, овај алгоритам је значајно спорији.

## Глава 4

# Експериментални резултати

Методе представљене у овом раду су имплементирани коришћењем *Python* програмског језика. Све инстанце проблема тестиране су на *MacBook Pro* рачунару, са *macOS Big Sur* оперативним системом. Рачунар поседује *2 GHz Quad-Core Intel Core i5* процесор са *16 GB RAM* меморије.

За визуализацију података и креирање мапа коришћен је *kepler.gl*. *Kepler.gl* [19] је веб апликација високих перформанси која се користи за визуализацију великих скупова гео-просторних података. Ова апликација је у стању да рендерује милионе тачака, и примени различите гео-просторне агрегације. Апликацију је развила компанија *Uber* за потребе визуализације својих података.

Како су методе описане у овом раду развијане са циљем да се примене у случају посматране логистичке компаније, оне су и тестиране на њеним подацима. Подаци су креирани на основу историјских података за један радни дан. У разматрање су узете све пошиљке које су курири доставили тог дана у својим областима. У наставку ће бити представљени резултати експеримената који укључују:

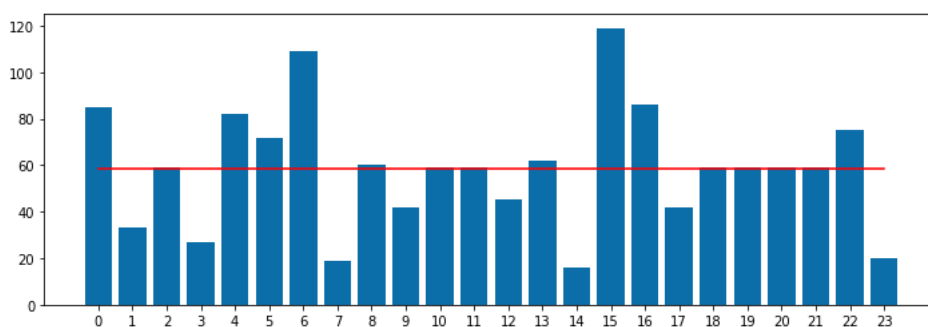
- Стандардни *k-means* алгоритам са локалном претрагом;
- Стандардни *k-means* алгоритам са *VNS* методом;
- Вишеструки *k-means* алгоритам са локалном претрагом;
- Вишеструки *k-means* алгоритам са *VNS* методом.

Један од циљева овог рада било је и унапређење ефикасности доставе. Метода тестирања биће детаљније описана у секцији 4.5.

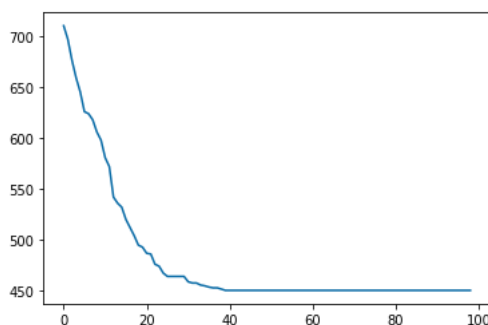
## 4.1 Стандардни *k-means* алгоритам са локалном претрагом

У овој секцији биће представљени резултати тестирања стандардног *k-means* алгоритма са локалном претрагом на две различите области у којима се врши достава.

Прва разматрана област покрива Нови Београд. У овој области за посматрани дан достављено је 1407 пошиљака, које су подељене на 24 региона.

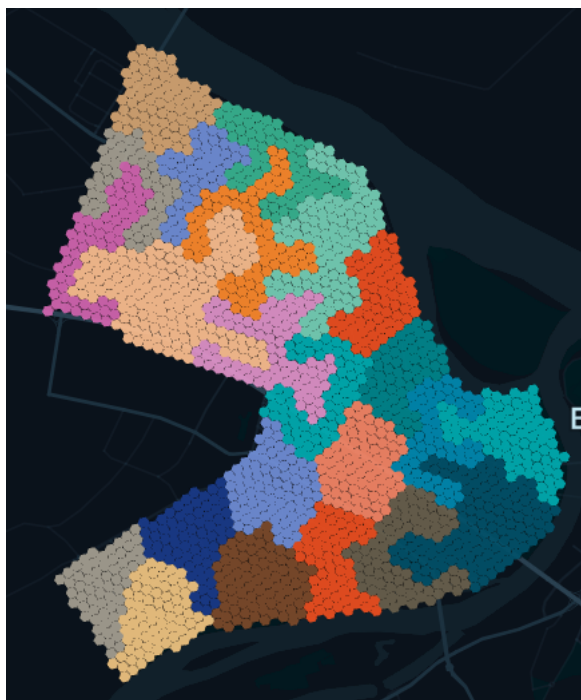


Слика 4.1: Број пошиљака по кластеру након примене стандардног *k-means* алгоритма са локалном претрагом, на примеру Новог Београда



Слика 4.2: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни *k-means* алгоритам са локалном претрагом, на примеру Новог Београда

На слици 4.3 је приказан крајњи изглед кластера након примене стандардног *k-means* алгоритма са локалном претрагом, а на слици 4.1 је приказана крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на



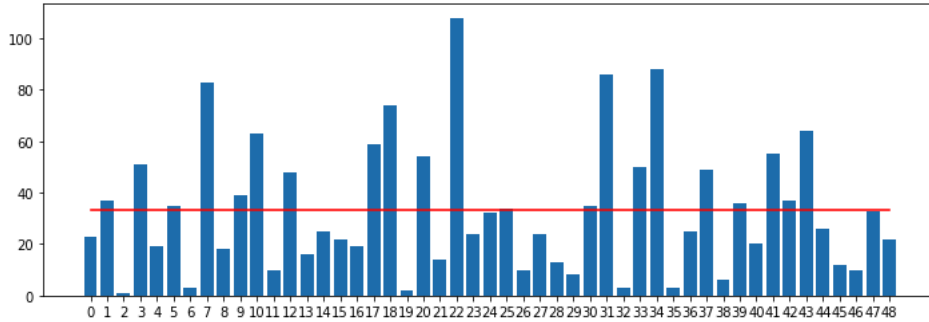
Слика 4.3: Изглед региона након примене стандардног *k-means* алгоритма са локалном претрагом, на примеру Новог Београда

$y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом случају 58. Кластер са највише пошиљака је кластер број 15 који је на слици представљен браон бојом и простире се кроз 63. блок. Кластери са најмање пошиљака су кластери 7 и 14 који су на слици представљени розе и крем бојом и налазе се у Земуну. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 450.

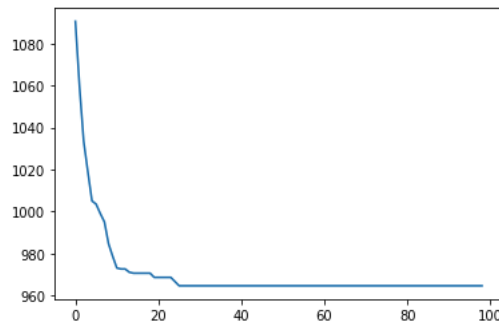
Као што се може видети на слици 4.2 стандардни *k-means* алгоритам са локалном претрагом је достигао локални минимум након 40 итерација и алгоритам више није могао да унапреди решење. Релативно лоше крајње решење је последица врло неизбалансираног почетног решења. Расподела пошиљака по кластерима за почетно решење приказана је у претходном поглављу, на слици 3.3.

Друга разматрана област покрива центар Београда. У овој области за посматрани дан достављено је 1628 пошиљака, које су подељене на 49 региона.

На слици 4.6 је приказан крајњи изглед кластера након примене стандардног *k-means* алгоритма са локалном претрагом, а на слици 4.4 је приказана



Слика 4.4: Број пошиљака по кластеру након примене стандардног  $k$ -means алгоритма са локалном претрагом, на примеру центра Београда

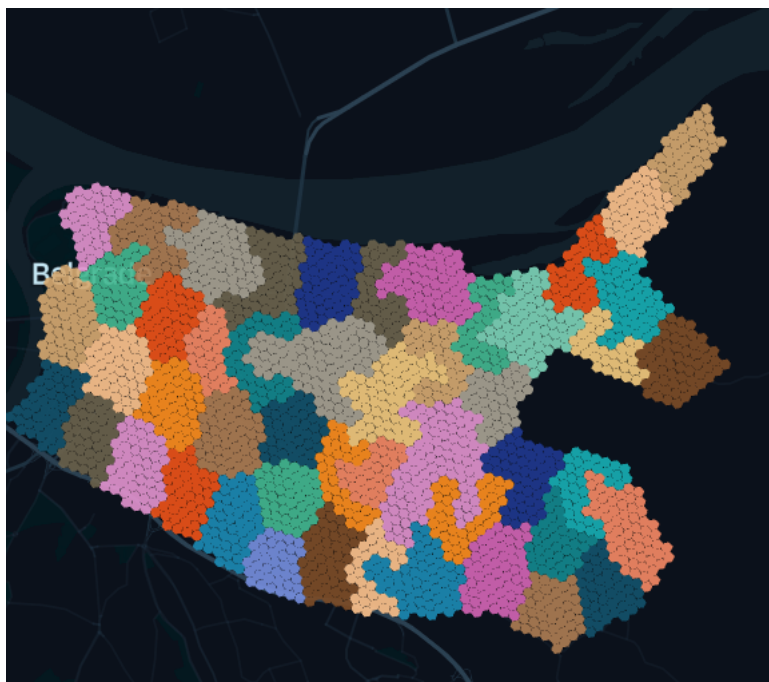


Слика 4.5: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни  $k$ -means алгоритам са локалном претрагом, на примеру центра Београда

крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере, која је у овом случају 33.22. Кластер са највише пошиљака је кластер број 18 који је на слици представљен браон бојом и покрива део око Београда на води. Овај кластер има два изразито велика хексагона, од по 18 пошиљака у сваком. Кластери са најмање пошиљака су кластери 8, 21, 36 који су на слици представљени наранџастом, маслинасто зеленом и светло плавом бојом. Ови региони се налазе на самој граници области уз Дунав. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 975.57.

Као што се може видети на слици 4.5 стандардни  $k$ -means алгоритам са локалном претрагом је достигао локални минимум након 25 итерација и алго-





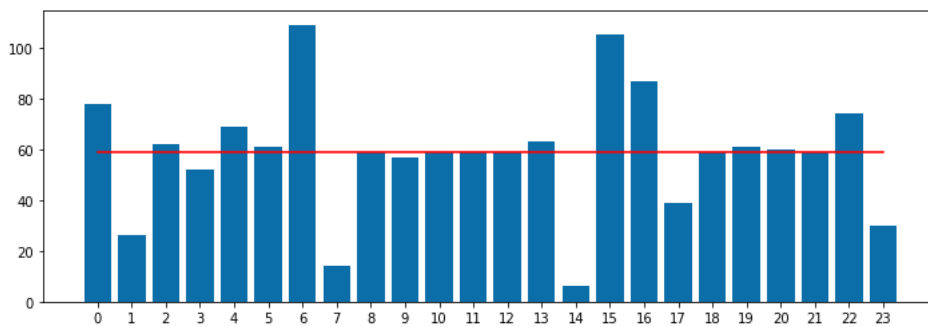
Слика 4.6: Изглед региона након примене стандардног *k-means* алгоритма са локалном претрагом, на примеру центра Београда

ритам више није могао да унапреди решење. Релативно лоше крајње решење је и у овом случају последица врло неизбалансираног почетног решења.

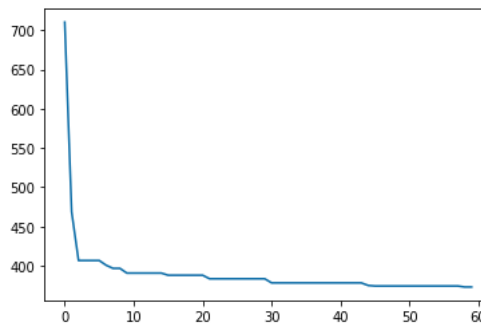
## 4.2 Стандардни $k$ -means алгоритам са $VNS$ методом

У овој секцији биће представљени резултати тестирања стандардног  $k$ -means алгоритма са  $VNS$  методом на две различите области у којима се врши достава.

Прва разматрана област покрива Нови Београд. У овој области за посматрани дан достављено је 1407 пошиљака, које су подељене на 24 региона.

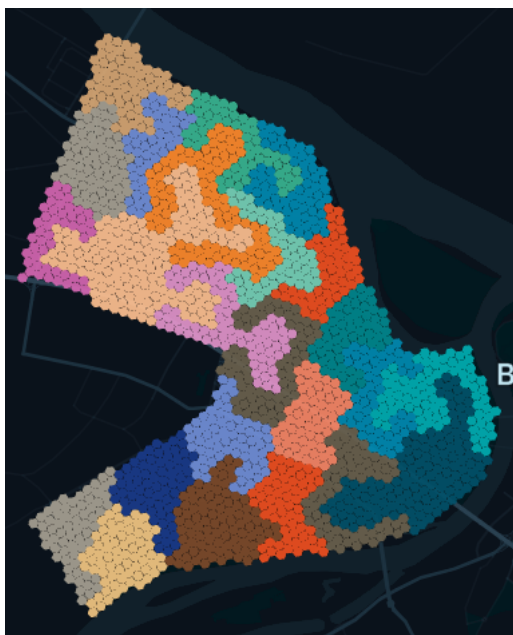


Слика 4.7: Број пошиљака по кластеру након примене стандардног  $k$ -means алгоритма са  $VNS$  методом, на примеру Новог Београда



Слика 4.8: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни  $k$ -means алгоритма са  $VNS$  методом, на примеру Новог Београда

На слици 4.7 је приказана крајња расподела пошиљака по кластерима након покренутих 60 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом



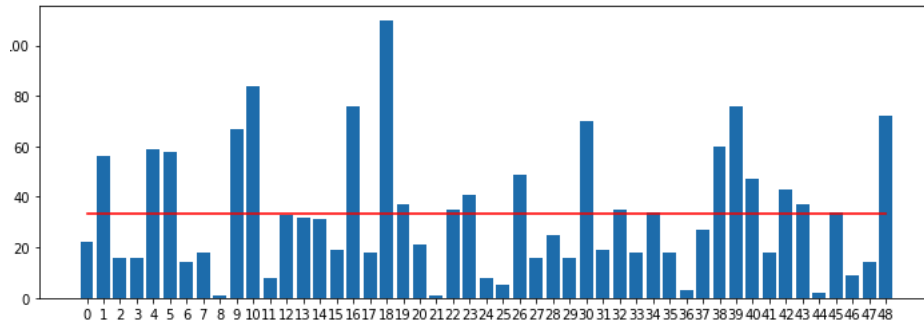
Слика 4.9: Изглед региона након примене стандардног *k-means* алгоритма са *VNS* методом, на примеру Новог Београда

случају 58. На слици 4.9 је приказан крајњи изглед кластера. Кластери са највише пошиљака су кластери 6 и 15 који су на слици представљени тамно плавом и бресква-наранџастом бојом. Кластери са најмање тачака су кластер 7 и 14 представљени розе и крем бојом. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 372.75.

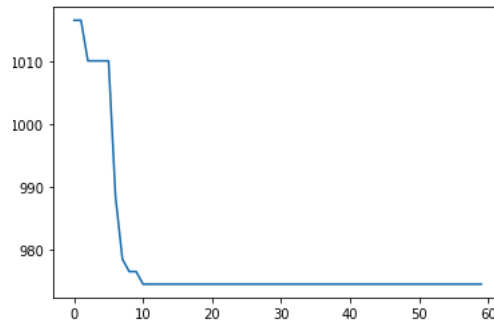
Стандардни *k-means* алгоритам са *VNS* методом је успео да изађе из локалног минимума, који је достигла локална претрага и да на тај начин пронађе боље решење. Међутим, како је почетно решење врло неизбалансирано, ни овај алгоритам није успео много да унапреди коначно решење. Алгоритам је заустављен након 60 итерација, како би се ограничило време извршавања.

Друга разматрана област покрива центар Београда. У овој области за посматрани дан достављено је 1628 пошиљака, које су подељене на 49 региона.

На слици 4.12 је приказан крајњи изглед кластера након примене стандардног *k-means* алгоритма са *VNS* методом, а на слици 4.10 је приказана крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на *y*-оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом случају 33.22. Кластер са нај-



Слика 4.10: Број пошиљака по кластеру након примене стандардног *k-means* алгоритма са *VNS* методом, на примеру центра Београда



Слика 4.11: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни *k-means* алгоритам са *VNS* методом, на примеру центра Београда

више пошиљака је кластер број 18 који је на слици представљен браон бојом и покрива део око Београда на води. Овај кластер има два изразито велика хексагона, од по 18 пошиљака у сваком. Кластери са најмање пошиљака су кластери 8, 21, 36 који су на слици представљени наранџастом, маслинасто зеленом и светло плавом бојом. Ови региони се налазе на самој граници области уз Дунав. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 964.57.

Као што се може видети на слици 4.11, стандардни *k-means* алгоритам са локалном претрагом је достигао локални минимум након 10 итерација и алгоритам више није могао да унапреди решење. Релативно лоше крајње решење је и у овом случају последица врло неизбалансираног почетног решења.



Слика 4.12: Изглед региона након примене стандардног  $k$ -means алгоритма са VNS методом, на примеру центра Београда

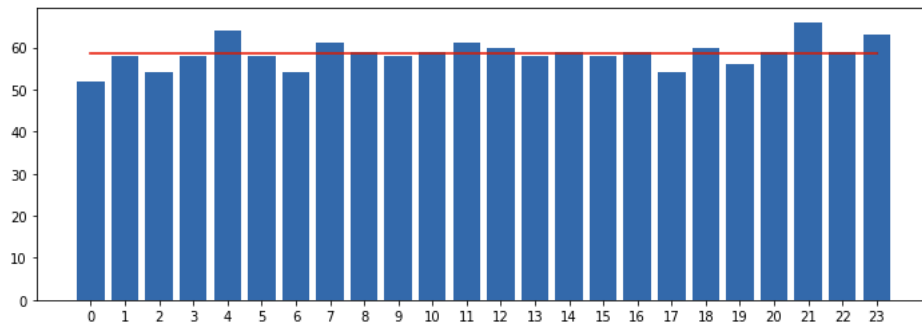
### 4.3 Вишеструки $k$ -means алгоритам са локалном претрагом

У овој секцији биће представљени резултати тестирања вишеструког  $k$ -means алгоритма са локалном претрагом на две различите области у којима се врши достава.

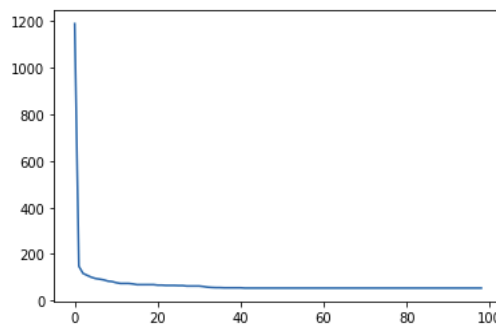
Прва разматрана област покрива Нови Београд. У овој области за посматрани дан достављено је 1407 пошиљака, које су подељене на 24 региона.

На слици 4.13 је приказана крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом случају 58. На слици 4.15 је приказан крајњи изглед кластера. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 53.75.

Како је почетно решење генерисано вишеструким  $k$ -means алгоритмом,



Слика 4.13: Број пошиљака по кластеру након примене вишеструког  $k$ -means алгоритма са локалном претрагом, на примеру Новог Београда

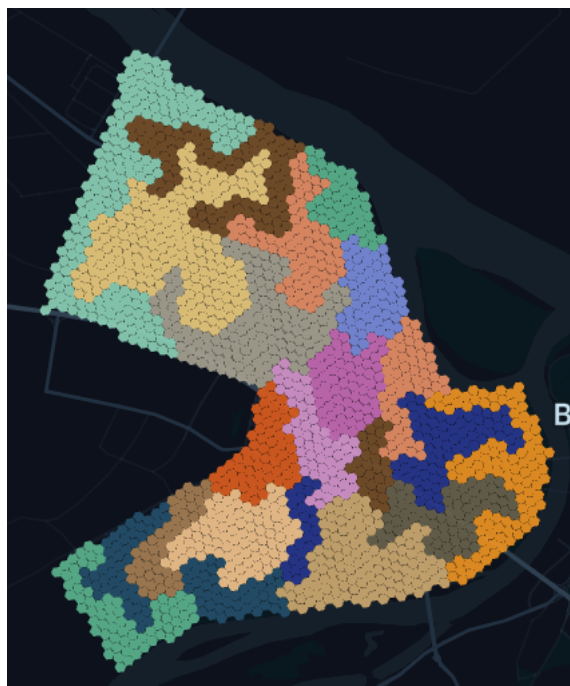


Слика 4.14: Укупно апсолутно растојањ кластера од идеалне тежине кроз итерације за вишеструки  $k$ -means алгоритам са локалном претрагом, на примеру Новог Београда

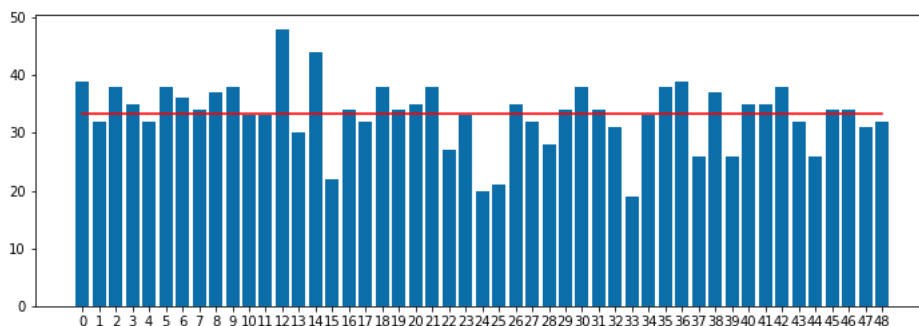
боље избалансирано од почетног решења генерисаног стандардним  $k$ -means алгоритмом, вишеструки  $k$ -means алгоритам са локалном претрагом дао је значајно боље коначно решење од стандардног  $k$ -means алгоритма са локалном претрагом. Расподела пошиљака по кластерима за почетно решење приказана је у претходном поглављу, на слици 3.7

Друга разматрана област покрива центар Београда. У овој области за посматрани дан достављено је 1628 пошиљака, које су подељене на 49 региона.

На слици 4.18 је приказан крајњи изглед кластера након примене стандардног  $k$ -means алгоритма са локалном претрагом, а на слици 4.16 је приказана крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере, која је у овом случају 33.22. Кластер са

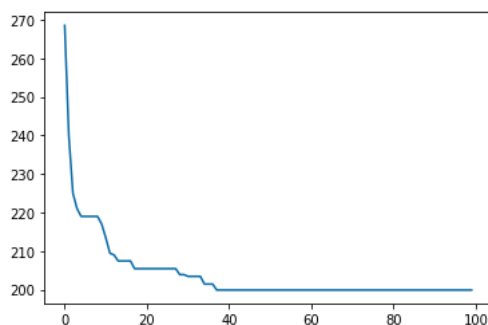


Слика 4.15: Изглед региона након примене вишеструког  $k$ -means алгоритма са локалном претрагом, на примеру Новог Београда

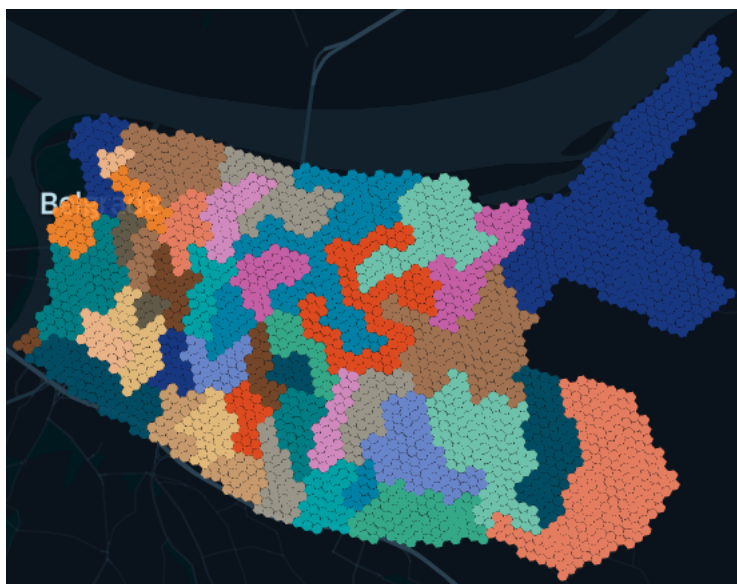


Слика 4.16: Број пошилака по кластеру након примене стандардног  $k$ -means алгоритма са локалном претрагом, на примеру центра Београда

највише пошилака је кластер број 12 који је на слици представљен браон бојом и покрива део око Београда на води. Овај кластер има два изразито велика хексагона, од по 18 пошилака у сваком. Кластери са најмање пошилака су кластери 15, 23, 33 који су на слици представљени розе, светло браон и маслинасто зеленом бојом. Регион број 15 се налази у центру области. Регион број 23 обухвата Дорћол, док регион 33 обухвата део око Народног позоришта. Укупно апсолутно растојање свих кластера од своје идеалне тежине у



Слика 4.17: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни *k-means* алгоритам са локалном претрагом, на примеру центра Београда



Слика 4.18: Изглед региона након примене стандардног *k-means* алгоритма са локалном претрагом, на примеру центра Београда

овом случају је 198.

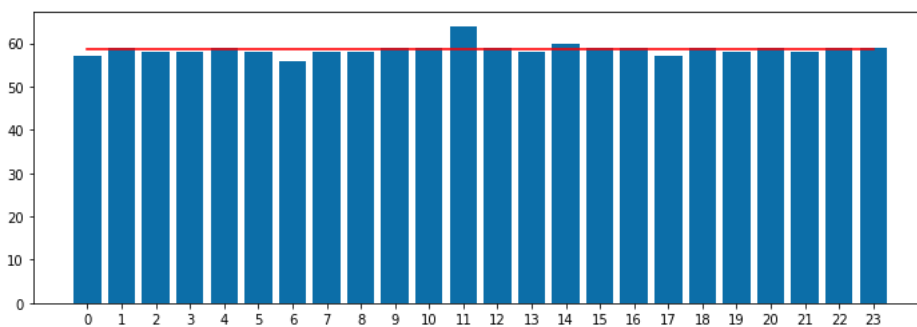
Као што се може видети на слици 4.17, стандардни *k-means* алгоритам са локалном претрагом је достигао локални минимум након 38 итерација и алгоритам више није могао да унапреди решење. И у овом случају се показало да почетно решење значајно утиче како на крању дистрибуцију тако и на облике и позиције проблематичних кластера.



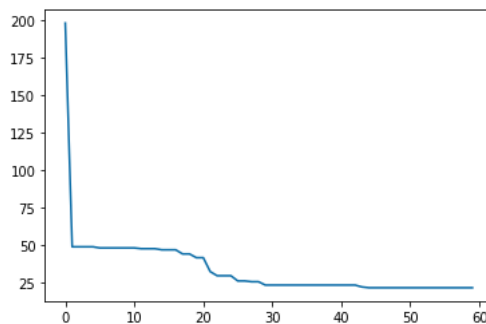
## 4.4 Вишеструки $k$ -means алгоритам са $VNS$ методом

У овој секцији биће представљени резултати тестирања вишеструког  $k$ -means алгоритма са  $VNS$  методом на две различите области у којима се врши достава.

Прва разматрана област покрива Нови Београд. У овој области за посматрани дан достављено је 1407 пошиљака, које су подељене на 24 региона.

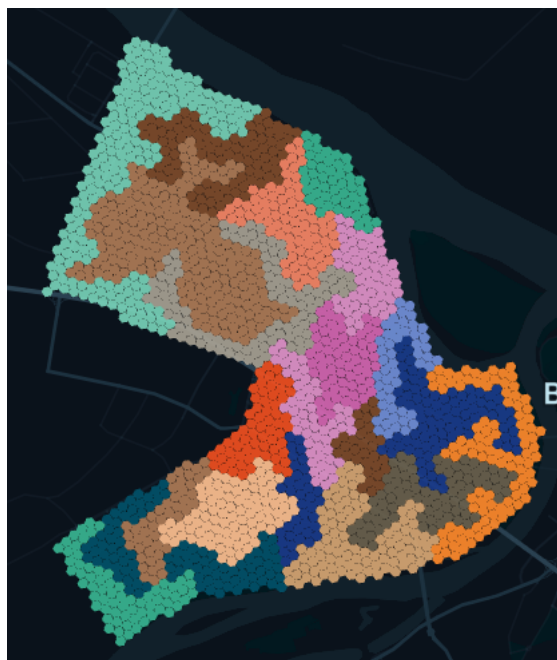


Слика 4.19: Број пошиљака по кластеру након примене вишеструког  $k$ -means алгоритма са  $VNS$  методом, на примеру Новог Београда



Слика 4.20: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за вишеструки  $k$ -means алгоритам са  $VNS$  методом, на примеру Новог Београда

На слици 4.19 је приказана крајња расподела пошиљака по кластерима након покренутих 60 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом



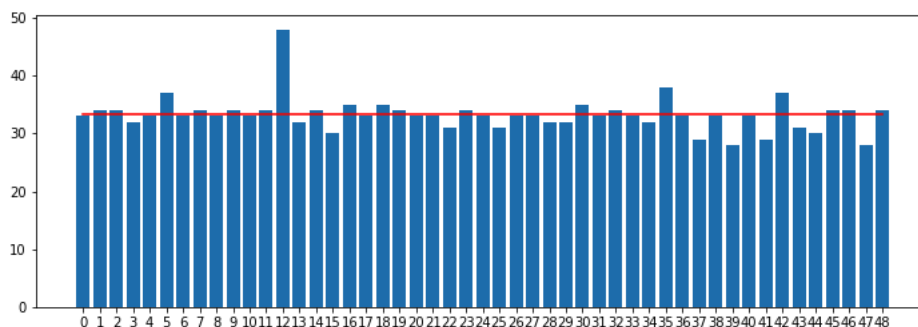
Слика 4.21: Изглед региона након примене вишеструког  $k$ -means алгоритма са  $VNS$  методом, на примеру Новог Београда

случају 58. На слици 4.21 је приказан крајњи изглед кластера. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 21.75.

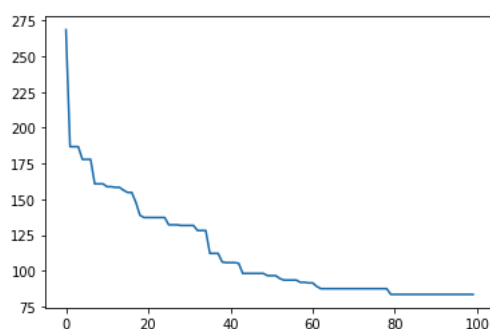
Очекивано, вишеструки  $k$ -means алгоритам са  $VNS$  методом дао је најбоље решење. У овом случају почетно решење је било довољно добро па је алгоритам успео да нађе адекватно крајње решење након 60 итерација.

Друга разматрана област покрива центар Београда. У овој области за посматрани дан достављено је 1628 пошиљака, које су подељене на 49 региона.

На слици 4.24 је приказан крајњи изглед кластера након примене стандардног  $k$ -means алгоритма са  $VNS$  методом, а на слици 4.22 је приказана крајња расподела пошиљака по кластерима након покренутих 100 итерација алгоритма. Сваки стубац на слици представља један кластер. Вредности на  $y$ -оси представљају укупан број пошиљака. Црвеном линијом представљена је циљана тежина за кластере која је у овом случају 33.22. Кластер са највише пошиљака је кластер број 12 који је на слици представљен браон бојом и покрива део око Београда на води. Овај кластер има два изразито велика хексагона, од по 18 пошиљака у сваком, међутим у овом случају кластер је успешно сведен на само четири хексагона. Кластери са најмање пошиљака су



Слика 4.22: Број пошиљака по кластеру након примене стандардног *k-means* алгоритма са *VNS* методом, на примеру центра Београда



Слика 4.23: Укупно апсолутно растојање кластера од идеалне тежине кроз итерације за стандардни *k-means* алгоритам са *VNS* методом, на примеру центра Београда

кластери 39, 47 који су на слици представљени љубичастом и сивом бојом. Укупно апсолутно растојање свих кластера од своје идеалне тежине у овом случају је 83.

Очекивано, вишеструки *k-means* алгоритам са *VNS* методом дао је најбоље решење. У овом случају почетно решење је било довољно добро па је алгоритам успео да нађе адекватно крајње решење након 81 итерација, као што се може видети на слици 4.23.



Слика 4.24: Изглед региона након примене стандардног *k-means* алгоритма са *VNS* методом, на примеру центра Београда

## 4.5 Поређење добијених резултата са резултатима посматране логистичке компаније

Како су методе описане у овом раду развијане са циљем да се примене у случају посматране логистичке компаније, оне су и тестиране на њеним реалним подацима. Подаци на којима је рад тестиран креирани су на основу историјских података за један радни дан. У разматрање су узете све пошиљке који су курири доставили тог дана у својим областима.

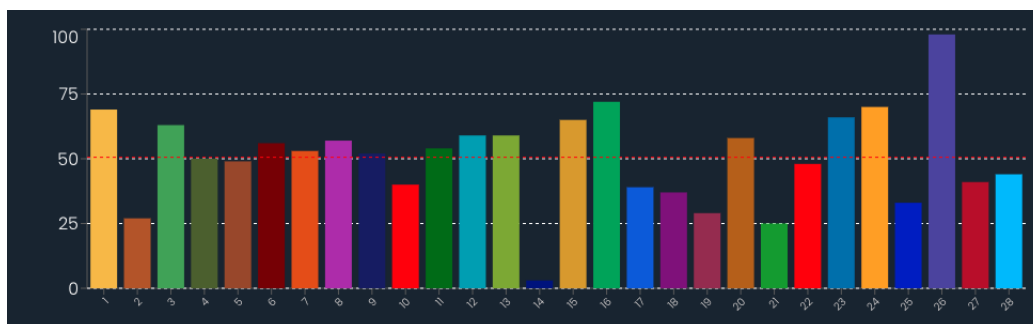
Да би се утврдило да ли новодобијени кластери могу да унапреде ефикасност доставе, извршено је поређењем оптималних путања. Прво су узети историјски подаци посматране логистичке компаније за један дан. Подаци се састоје од свих пошиљака који су курири доставили тог дана. Затим је за сваког курира и његов регион креирана оптимална рута. Након тога на основу истих података генерисани су и региони применом вишеструког *k-means* алгоритма са локалном претрагом и вишеструког *k-means* алгоритма са *VNS* методом. Генерисане су и оптималне руте и за те регионе.

Поређење оптималних путања у оба случаја дало је исту меру, узимајући у обзир само шта је курир урадио, а избацујући из разматрања све недостатке његове руте, генерисане искључиво на основу његовог знања и интуиције.

Два најважнија корака у процесу проналажења оптималних путања су генерисање матрице цена и оптимизациони алгоритам.

Матрица цена, се може дефинисати у терминима дистанце или времена путовања између парова тачака. Другим речима, за  $N$  датих тачака матрица цена ће бити димензија  $N \times N$ , где тачка  $(i, j)$  представља дистанцу (или време вожње) између тачака  $i$  и  $j$ . За добијање матрица цена коришћена је *OSRM* библиотеке. *OSRM* [15] је *open source* библиотека која представља модул за рутирање, дизајнирана да ради са *OpenStreet* мапама.

Оптимизациони алгоритми за оптимизацију руте припадају групи широко распрострањених алгоритама, од којих је најпознатији проблем путујућег трговца. Сви ови алгоритми захтевају матрицу цена као улазни податак. На основу матрице цене ови алгоритми користе различите хеуристичке методе за проналажење оптималног редоследа и руте између тачака. За ову сврху искоришћена је *Google OR tools* [6] библиотека.

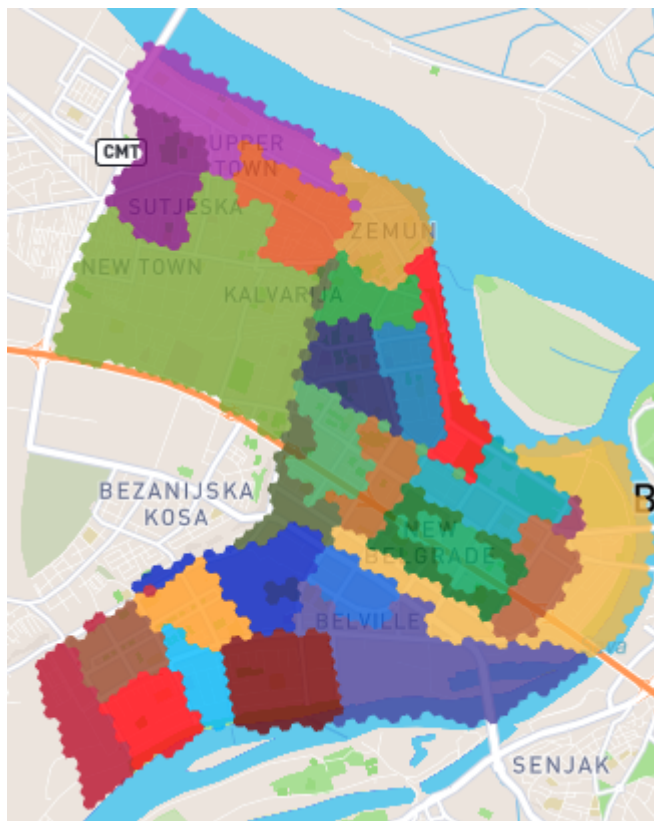


Слика 4.25: Број пошиљака по регионима какве користи посматрана логистичка компанија, на примеру Новог Београда

У посматраном експерименту логистичка компанија је користила 28 курира распоређених на 28 региона и доставила 1407 пошиљака на простору Новог Београда. Алгоритми приказани у овом раду су исти број пошиљака поделили на 24 региона. На основу приказаних резултата може се закључити да се са бољом расподелом посла, исти посао може обавити са 15% мање људи, при чему је максимална пређена дистанца увећана за 2%, а укупна дистанца смањена за 8%.

Табела 4.1: Приказ километраже и времена возње по регионима какве користи посматрана логистичка компанија

Километража	Време возње
38.06	64
32.66	57
33.18	65
31.31	55
46.30	77
27.01	41
38.60	65
36.64	58
32.08	49
42.22	71
17.91	30
26.21	46
36.12	60
31.53	53
41.72	76
37.11	66
46.35	84
37.03	63
38.32	62
42.47	71
37.27	59
38.67	67
39.17	67
43.03	76
40.31	79
44.47	71
65.28	100
33.09	56



Слика 4.26: Изглед региона које користи посматрана логистичка компанија, на примеру Новог Београда

Максимално време вожње на примеру региона посматране логистичке компаније је 100 минута. Док је максимална пређена дистанца 65.28 km. Минимално време вожње је 30 минута, а минимална дистанца је 17.91 km. На примеру региона конструисани вишеструким *k-means* алгоритмом са локалном претрагом, максимално време вожње је 113 минута, а максимална пређена дистанца је 71.82 km. Минимално време вожње је 53 минута, док је минимална пређена дистанца 28.51 km. На примеру региона генарисаним вишеструким *k-means* алгоритмом са *VNS* методом, максимално време вожње је 110 минута, а максимална пређена дистанца је 67.81 km. Минимално време вожње у овом случају је 52 минута, а минимална дистанца је 29.71 km.

Укупна пређена километража у случају посматране логистичке компаније је 1054.12 km, а укупно време вожње је 1788 минута. У случају вишеструког *k-means* алгоритма са локалном претрагом укупна километража је 975.75 km, а укупно време вожње је 1683 минута. У случају вишеструког *k-means*

Табела 4.2: Приказ километраже и времена вожње по регионима за вишеструки *k-means* алгоритам са локалном претрагом

Километража	Време вожње
38.83	67
49.05	79
50.11	92
40.25	70
47.12	87
37.35	66
45.77	76
44.33	80
43.03	78
34.44	58
46.40	75
33.98	58
31.88	53
35.19	64
40.77	69
71.82	113
42.95	74
28.51	50
36.91	69
34.12	61
34.95	57
36.75	66
30.98	53
40.26	68

алгоритма са *VNS* методом укупна километража је 1691 km, а укупно време вожње је 979.16 минута. На основу наведеног може се закључити да је балансирана расподела посла може да повећа ефикасност доставе. Детаљан преглед километраже и времена вожње за три тестирана сценарија дат је у табелама 4.1, 4.2, 4.3, 4.4.



Табела 4.3: Приказ километраже и времена вожње по регионима за вишеструки  $k$ -means алгоритам са  $VNS$  методом

Километража	Време вожње
38.85	64
50.19	77
48.83	89
39.53	71
46.91	86
38.81	69
47.90	80
46.72	88
45.11	77
34.80	59
43.79	74
33.49	61
33.67	59
37.59	67
38.40	66
67.81	110
40.23	67
28.38	49
37.95	66
35.77	59
36.56	63
36.80	65
29.71	52
41.36	73

Табела 4.4: Поређење укупне километраже и времена вожње

Алгоритам	Укупна километража	Укупно време вожње
лог. компанија	1054.12	1788.00
локална претрага	975.75	1683.00
$VNS$ метода	979.16	1691.00

# Глава 5

## Закључак

Проблем последње миље се бави достављањем пошиљака од транспортног центра до крајњег корисника. Адекватна расподела посла, благовремено извештавање корисника о тренутку доставе и проналажење оптималних путања су саставно делови овог проблема. У овом раду бавили смо се проблемом адекватне расподеле посла, поделом пошиљака на регионе, са циљем унапређења ефикасности доставе.

Анализиране су четири методе за креирање балансираних региона:

- Стандардни *k-means* алгоритам са локалном претрагом;
- Стандардни *k-means* алгоритам са *VNS* методом;
- Вишеструки *k-means* алгоритам са локалном претрагом;
- Вишеструки *k-means* алгоритам са *VNS* методом.

Резултатати експеримената на две различите градске области показали су да стандардни *k-means* алгоритам са локалном претрагом даје најмање избалансиране кластере. Стандардни *k-means* алгоритам са *VNS* методом је у оба случаја дао мало боље резултате. Обе приказане методе су на крају дале поприлично неизбалансиране регионе. Вишеструки *k-means* алгоритам са локалном претрагом у оба случаја је дао значајно боље резултате. Најбоље резултате по питању избалансираности кластера дао је вишеструки *k-means* алгоритам са *VNS* методом. Овим је показано да почетно решење значајно утиче на крајњи резултат извршавања алгоритма.

Поређењем оптималних рута на примеру Новог Београда, показано је да адекватна расподела посла може да повећа ефикасност доставе. Исту коли-

чину посла које тренутно ради 28 курира, могуће је урадити са 24 курира, тј. са 15% мање људи, при чему максимална пређена дистанца по куриру није повећана за више од 2% (4 km). Укупна пређена дистанца смањена је за 8%.

На основу приказаних резултата посматрана логистичка компанија кренула је у процес имплементације предложених решења на примеру Новог Београда.

Алгоритми приказани у овом раду би се даље могли унапредити балансирањем кластера не само по броју пошиљака, већ и урачунавањем броја и запремине пакета, или времена потребног да се одређена пошиљка достави. Даље, употреба мреже улица за дефинисање повезаности би могла повољно да утиче на путање креиране у тако генерисаним кластерима. Мењањем величине хексагона који се користи или генерисањем кластера за податке на недељном или месечном нивоу су такође параметри који би се могли размотрити у даљем раду на овом проблему.

# Библиографија

- [1] Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981, 2018.
- [2] Alexandra Anderluh, Vera C Hemmelmayr, and Pamela C Nolz. Synchronizing vans and cargo bikes in a city distribution network. *Central European Journal of Operations Research*, 25(2):345–376, 2017.
- [3] Nils Boysen, Stefan Fedtke, and Stefan Schwerdfeger. Last-mile delivery concepts: a survey from an operational research perspective. *Or Spectrum*, 43(1):1–58, 2021.
- [4] Buyang Cao and Fred Glover. Creating balanced and connected clusters to improve service delivery routes in logistics planning. *Journal of Systems Science and Systems Engineering*, 19(4):453–480, 2010.
- [5] M Emre Celebi. *Partitional clustering algorithms*. Springer, 2014.
- [6] Google. Google or tools.
- [7] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.
- [8] Dag Haugland, Sin C Ho, and Gilbert Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.
- [9] David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

- [10] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.
- [11] Jörg Kalcsics and Roger Z Ríos-Mercado. Districting problems. In *Location science*, pages 705–743. Springer, 2019.
- [12] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(3):231–240, 2011.
- [13] Hongtao Lei, Gilbert Laporte, and Bo Guo. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1-2):67–85, 2012.
- [14] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.
- [15] Dennis Luxen and Christian Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11*, pages 513–516, New York, NY, USA, 2011. ACM.
- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [17] Roger Z Ríos-Mercado. *Optimal districting and territory design*, volume 284. Springer, 2020.
- [18] Gabriele Schliwa, Richard Armitage, Sara Aziz, James Evans, and Jasmine Rhoades. Sustainable city logistics—making cargo cycles viable for urban freight transport. *Research in Transportation Business & Management*, 15:50–57, 2015.
- [19] Uber Technologies. Kepler.gl, 2018.
- [20] Uber Technologies. H3 geospatial indexing system, 2022.

- [21] Yuxiang Zeng, Yongxin Tong, and Lei Chen. Last-mile delivery made practical: An efficient route planning framework with theoretical guarantees. *Proceedings of the VLDB Endowment*, 13(3):320–333, 2019.