

Универзитет у Београду  
Математички факултет

Александра Цекић

**Примене неких модификација методе  
променљивих околина у кластеровању**

мастер рад

Београд  
Септембар, 2016

**Ментор:**

др Александар Савић, доцент, Математички факултет, Универзитет у Београду

**Чланови комисије:**

др Зоран Станић, ванредни професор, Математички факултет, Универзитет у Београду

др Зорица Дражић, доцент, Математички факултет, Универзитет у Београду

**Аутор:**

Александра Цекић

## **Сажетак**

Овај рад се бави истраживањем ефикасности методе променљивих околина (енг. VNS) и неких њених модификација у партиционом кластеровању. Проблем кластеровања се дефинише као проналажење минималне вредности суме квадрата растојања тачака скупа који се кластерује до њиховог центроида. Главне фазе у решавању постављеног проблема су креирање околина одређеног допустивог решења и проналажење побољшања коришћењем неке од модификација. Детаљно је изложена нова идеја околине и начин њене имплементације је илустрован псеудокодом. Експериментални резултати на крају показују ефикасност имплементације у примени на неким јавно доступним и познатим скуповима података.

Кључне речи: метода променљивих околина, модификације, кластеровање, индекси валидности кластеровања, скупови података

## **Abstract**

This paper investigates the efficiency of Variable Neighborhood Search (VNS) heuristic and its modifications in partitional clustering. The clustering problem is defined in terms of minimization of the sum of squared distances between a clustering dataset points and their cluster's centroid. The main stages in resolving the described clustering problem are constructing neighborhoods of a feasible solution and improving it with a specific modification of VNS. A new idea of constructing neighborhood is presented in details, and its implementation is illustrated by pseudocode. At the end, experimental results show the efficiency of implementation in application on some publicly available and well known datasets.

Key words: Variable neighborhood search, modifications, clustering, cluster validity indices, datasets

## Садржај

|                                                                |    |
|----------------------------------------------------------------|----|
| 1. УВОД .....                                                  | 1  |
| 2. Метода променљивих околина .....                            | 2  |
| 2.1. Метода променљивог спуста .....                           | 2  |
| 2.2. Редукована метода променљивих околина.....                | 4  |
| 2.3. Основна и општа метода променљивих околина.....           | 6  |
| 2.4. Закошена метода променљивих околина.....                  | 9  |
| 3. КЛАСТЕРОВАЊЕ .....                                          | 11 |
| 3.1. Различити типови кластровања .....                        | 13 |
| 3.1.1. Хијерархијско и партиционо кластерованье .....          | 13 |
| 3.1.2. Ексклузивно, неексклузивно и фази-кластерованье .....   | 14 |
| 3.1.3. Потпуно и парцијално кластерованье .....                | 15 |
| 3.2. Различити типови кластера .....                           | 15 |
| 3.3. Индеки валидности .....                                   | 18 |
| 3.3.1. Унутрашњи критеријуми индекса валидности .....          | 18 |
| 3.3.2. Калински-Харабац индекс.....                            | 21 |
| 3.3.3. Дејвис-Болдин индекс .....                              | 22 |
| 3.3.4. Дан индекс .....                                        | 22 |
| 3.3.5. Силуета индекс.....                                     | 23 |
| 3.3.6. CS мера.....                                            | 24 |
| 4. МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА ЗА ПАРТИЦИОНО КЛАСТЕРОВАЊЕ ..... | 26 |
| 4.1. Функција циља .....                                       | 27 |
| 4.2. Околинe.....                                              | 28 |
| 4.2.1. Структура околинe 1.....                                | 29 |
| 4.2.2. Структура околина 2.....                                | 31 |
| 4.3. Локално претраживање .....                                | 33 |
| 4.4. Избор почетног решења .....                               | 34 |
| 4.5. Оптимални број кластера .....                             | 35 |
| 4.6. Критеријуми заустављања .....                             | 35 |
| 5. ИМПЛЕМЕНТАЦИЈА АЛГОРИТМА .....                              | 37 |
| 5.1. Основни VNS .....                                         | 37 |
| 5.2. Општи VNS .....                                           | 39 |
| 6. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ.....                              | 42 |
| 7. ЗАКЉУЧАК.....                                               | 57 |
| ЛИТЕРАТУРА.....                                                | 59 |

# 1. УВОД

Посматрајмо следећи проблем комбинаторне или глобалне оптимизације

$$\min_{x \in X} f(x) \quad (1.1)$$

дефинисан за  $x \in X$  где је  $f(x)$  функција циља која се минимизује, а  $X$  је скуп допустивих решења. Решење  $x^* \in X$  је оптимално ако  $f(x^*) \leq f(x), \forall x \in X$ .

Бројни проблеми овог облика, који се проучавају у комбинаторној оптимизацији, операционим истраживањима и другим областима, су превелики за егзактно решавање у разумном времену. Постоји на хиљаде проблема који су НП-тешки, тј. за које алгоритам полиномијалне сложености у функцији од димензије проблема није познат. Додатно, у неким случајевима где за проблем постоји алгоритам полиномске сложености, степен полинома може бити толико велики да се проблем у најгорем случају не може решити у разумном времену. Такви проблеми се често могу решити једино применом неке хеуристичке методе.

## 2. МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА

Метода променљивих околина (енг. *Variable Neighborhood Search*, VNS) је метахеуристика први пут предложена у раду [1] (1997). У основи методе је идеја о систематској промени околина у којима се тражи оптимално решење тако да се итеративно напредује ка оптималном решењу али се истовремено избегавају долине са локалним минимумима. Метода систематски користи следеће три чињенице [2]:

1. Локални минимум у некој околини није обавезно локални минимум и за неку другу околину.
2. Глобални минимум је локални минимум у свим околинама.
3. За многе проблеме локални минимуми у једној или више околина су релативно близу једни другима.

У поглављима која следе, биће представљене неке најпознатије модификације методе променљивих околина, [1], [3], [9] које ће касније бити коришћене за имплементацију алгоритма и тестирање.

### 2.1. Метода променљивог спуста

Нека је са  $N(x)$  означена нека околина допустивог решења  $x$  у скупу  $X$ . Хеуристика најстрмијег спуста се састоји од избора почетног решења  $x$ , налажења правца најстрмијег спуста од  $x$  унутар  $N(x)$  и померања у тачку минимума функције  $f(x)$  унутар  $N(x)$  у том правцу. Ако не постоји правац спуста, хеуристика стаје, у супротном се понавља поступак. Овај скуп правила приказан је следећим псеудокодом:

---

Иницијализација:

Изабрати структуру околине  $N(x)$  и почетну вредност  $x \in X$ ;

**repeat**

1. Наћи  $x' = \operatorname{argmin}_{x \in N(x)} f(x)$ ;
2. Ако је  $f(x') < f(x)$  поставити  $x \leftarrow x'$  и поновити поступак; у супротном, зауставити се.

**end**

---

### Псеудокод 2.1 - Стратегија најстрмијег спуста

Приметимо да је околина  $N(x)$  дефинисана за свако  $x \in X$ . У дискретној оптимизацији околина тачке се често добија неком простом модификацијом, нпр. комплентирањем једне или две њене 0-1 координате. Онда се у сваком кораку испитује свака тачка из  $N(x)$  околине. Пошто овај приступ може захтевати доста времена, друга опција је коришћење хеуристике првог спуста. Померање се дешава чим се пронађе правац спуста, као што је илустровано псеудокодом 2.2.

---

Иницијализација:

Изабрати структуру околине  $N(x)$  и почетно решење  $x \in X$ ;

**repeat**

1. Наћи прво решење  $x' \in N(x)$ ;
2. Ако је  $f(x') > f(x)$ , наћи следеће решење  $x'' \in N(x)$ ; поставити  $x' \leftarrow x''$  и прећи поново на корак (2); у супротном, поставити  $x \leftarrow x'$  и прећи на корак (1);
3. Ако су размотрена сва решења  $N(x)$ , зауставити се.

**end**

---

### Псеудокод 2.2 - Стратегија првог спуста

Показује се да комбинација ових двеју варијанти има највише предности. Метода која се тако добија назива се методом променљивог спуста (енг. *Variable Neighborhood Descent*,

VND) и заснована је на чињеници да једно решење не мора бити локални минимум за две различите околине. Шема VND алгоритма представљена је псеудокодом 2.3.

---

Иницијализација:

Изабрати скуп структура околина  $N_l$ , за  $l = 1, \dots, l_{\max}$ , које ће бити коришћене у претрази и наћи почетно решење  $x$ ;

Поставити  $l \leftarrow 1$ ;

**repeat** следеће кораке **until**  $l \leq l_{\max}$ :

- a. *Истраживање околине.* Наћи најбољег суседа  $x'$  од  $x$  из околине  $N_l(x)$  ( $x' \in N_l(x)$ );
- b. *Да ли се померити или не?* Ако је решење  $x'$ , добијено на тај начин, боље од  $x$ , поставити  $x \leftarrow x'$  и  $l \leftarrow 1$ ; у супротном, поставити  $l \leftarrow l + 1$ ;

**end**

---

### Псеудокод 2.3 - VND алгоритам

Приликом примене овог алгоритма, треба водити рачуна о следећем:

- Коју сложеност одговарајући кораци имају?
- Који је најбољи редослед њихове примене?
- Да ли су изабрани кораци довољни да обезбеде комплетно испитивање области која садржи  $x$ ?
- Колико прецизно решење желимо?

Узимајући у обзир одговоре на ова питања могу се значајно побољшати перформансе алгоритма.

## 2.2. Редукована метода променљивих околина

Претпоставимо да је локални минимум  $x'$  функције  $f$  достигнут. Тада бисмо желели да напустимо тренутну долину и портажимо још боље решење. Са тим циљем постављамо питања:



- У ком правцу ићи?
- Колико далеко?
- Како би требало изменити кораке ако не доводе до успеха?

Прво питање се односи на могућност достизања било које допустиве тачке  $x \in X$ . Најпростији одговор је изабрати случајан правац, међутим ако се на исплатив начин може закључити који је погоднији правац кретања, то може скратити време потребно за проналажење апроксимације оптималног решења.

Друго питање је суштинско. У многим комбинаторним и оптимизационим проблемима, локални оптимуми су релативно близу једни другима. Тако да када се локални оптимум једном достигне, може се десити да он носи имплицитну информацију о близини другог бољег, можда глобалног, оптимума. Природно је истражити прво његову околину. Међутим, ако је долина која окружује локални оптимум  $x'$  велика, то неће бити довољно. Шта радити следеће је треће питање. Природан одговор је ићи још даље.

Разматра се скуп околина  $N_1(x), N_2(x), \dots, N_{k_{max}}(x)$  око тренутне тачке  $x$  (која може и не мора бити локални оптимум). Обично су ове околине угњежене, тј. свака садржи претходну. Онда се бира случајно тачка из прве околине. Ако је вредност функције  $f$  у тој тачки боља од вредности у тренутној тачки, даљу претрагу настављамо у околини новонађеног локалног оптимума. У супротном, настављамо са следећом околином. Пошто се размотре све околине, креће се опет од прве све док се не испуни критеријум заустављања (обично ће то бити максимално време протекло од последњег побољшања или максималан број итерација). Овако дефинисана метода представља редуковану варијанту методе променљивих околина (енг. *Reduced Variable Neighborhood Search*, *RVNS*) и представљена је псеудокодом 2.4.

---

Иницијализација:

Изабрати скуп структура околина  $N_k$ , за  $k = 1, \dots, k_{max}$ , које ће бити коришћене у претрази;  
наћи почетно решење  $x$ ; изабрати критеријум заустављања;

Поставити  $k \leftarrow 1$ ;

**repeat** следеће кораке **until**  $k \leq k_{max}$ :

- а. *Размрдавање*. Генерисати случајну тачку  $x'$  из  $k$ -те околине од  $x$  ( $x' \in N_k(x)$ );
- б. *Да ли се померити или не?* Ако је ово решење боље од тренутног, померити се у ново решење ( $x \leftarrow x'$ ) и наставити са претрагом у околини  $N_1$  ( $k \leftarrow 1$ ), у супротном, поставити  $k \leftarrow k + 1$ ;

**end**

---

Псеудокод 2.4 - RVNS алгоритам

### 2.3. Основна и општа метода променљивих околина

Спајање претходне две модификације доводи до опште методе променљивих околина (енг. *General Variable Neighborhood Search*, GVNS). Прво ће бити размотрено како спојити локалну претрагу са систематским променама околина око нађеног локалног оптимума, чиме се добја основна метода променљивих околина (енг. *Basic Variable Neighborhood Search*, BVNS), представљена псеудокодом 2.5:

---

Иницијализација:

Изабрати структуре околина  $N_k$ , за  $k = 1, \dots, k_{max}$ , које ће бити коришћене за претрагу; наћи почетно решење  $x$ ; изабрати критеријум заустављања;

Понављати следећи низ корака док се не задовољи критеријум заустављања:

Поставити  $k \leftarrow 1$ ;

**repeat** следеће кораке **until**  $k \leq k_{max}$ :

- a. *Размрдавање*. Генерисати случајну тачку  $x'$  из  $k$ -те околине од  $x$  ( $x' \in N_k(x)$ );
- b. *Локална претрага*. Применити неку методу локалне претраге са  $x'$  као почетним решењем; означити са  $x''$  локални оптимум добијен на тај начин;
- c. *Да ли се померити или не?* Ако је локални оптимум  $x''$  бољи од тренутно најбољег решења  $x$ , померити ( $x \leftarrow x''$ ) и наставити са претрагом у околини  $N_1(k \leftarrow 1)$ , у супротном, поставити  $k \leftarrow k + 1$ ;

**end**

---

#### Псеудокод 2.5 - BVNS алгоритам

Према овој шеми, прво се изабере скуп околина  $N_k$ ,  $k = 1, \dots, k_{max}$  таквих да је  $N_k(x)$   $k$ -та околина решења  $x$ . Тачка  $x'$  се бира случајно из прве околине  $N_1(x)$  тачке  $x$ . Онда се користећи локалну претрагу и полазећи из  $x'$  налази локални оптимум  $x''$ . Ако се деси да је  $x'' = x$ , то значи да је процедура опет на дну исте долине. У том случају процедура наставља са следећом околином  $N_k(x)$ ,  $k \geq 2$ . У супротном, ако је  $x'' \neq x$ , имамо две могућности. Прво да је  $f(x'') \geq f(x)$ , тј. локални оптимум који је нађен није бољи од тренутно најбољег решења. У том случају, такође, процедура наставља са следећом околином. Друга могућност је да је  $f(x'') < f(x)$ , тј. нови локални оптимум је бољи од тренутно најбољег решења. У том случају се претрага наставља у околини  $x''$ . Креће се опет од прве околине. Ако се стигне до последње околине без налажења бољег решења од тренутног, претрага креће опет од прве околине, док се не испуни критеријум заустављања, нпр. максимално време или максималан број итерација од последњег побољшања.

Ако се уместо просте локалне претраге користи *VND* и ако се нађено почетно решење побољшава користећи редуковани *VNS*, добија се општа метода променљивих околина представљену псеудокодом 2.6.

---

Иницијализација:

Изабрати скуп структура околина  $N_k$ , за  $k = 1, \dots, k_{max}$ , које ће бити коришћене за фазу размрдавања, и скуп структура околина  $N_l$ , за  $l = 1, \dots, k_{max}$ , које ће бити коришћене за локалну претрагу; наћи почетно решење  $x$ ; изабрати критеријум заустављања;

Понављати следећи низ корака док се не задовољи критеријум заустављања:

Поставити  $k \leftarrow 1$ ;

**repeat** следеће кораке **until**  $k \leq k_{max}$ ;

- a. *Размрдавање*. Генерисати случајну тачку  $x'$  из  $k$ -те околине  $N_k(x)$ ;
- b. *Локална претрага коришћењем методе променљивог спуста*.
- c. Поставити  $l \leftarrow 1$ ;
- d. Понављати следеће кораке док не добијемо да је  $l \leftarrow l_{max}$ :
  - *Истраживање околине*. Наћи најбољег суседа  $x''$  решења  $x'$  у околини  $N_l(x')$ ;
  - *Да ли се померити или не?* Ако је  $f(x'') < f(x')$  поставити  $(x' \leftarrow x'')$  и  $l \leftarrow 1$ ; у супротном, поставити  $l \leftarrow l + 1$
- e. *Да се померити или не?* Ако је локални оптимум  $x''$  бољи од тренутно најбољег решења  $x$ , померити  $(x \leftarrow x'')$  и наставити са претрагом у околини  $N_1$  ( $k \leftarrow 1$ ), у супротном, поставити  $k \leftarrow k + 1$ ;

**end**

---

#### Псеудокод 2.6 - GVNS алгоритам

Приликом избора околине корисно је водити рачуна о следећим питањима:

- Која су неопходна својства околина да би се дошло до глобалног оптималног или скоро оптималног решења?
- Која својства околина ће поспешити налажење решења близу оптималног?

- Да ли би околине требало да буду угњежене? У супротном, како би их требало уредити?
- Шта је пожељна величина околине?

Одговори на ова питања доприносе бржем проналажењу решења и избегавању заглављивања у неки локални минимум, а детаљније ће бити размотрени у поглављу које се бави дефинисањем структуре околина.

## 2.4. Закошена метода променљивих околина

Многи проблеми имају груписане, или на неки начин нагомилане локалне оптимуме. Често су њихове функције циља глобално конвексне. Међутим, дешава се да неки примери имају више раздвојених и могуће удаљених долина које садрже локалне оптимуме. Ако бисмо повећавали околине, метода би постала вишеструко локално претраживање (енг. *multistart*). Овај проблем превазилазимо закошеном методом променљивих околина (енг. *Skewed Variable Neighborhood Search, SVNS*).

Ова варијанта методе дозвољава прелазак у лошије решење ако се разликују у довољној мери. Прелазак у лошије решење  $x''$  из тренутног  $x$  биће дозвољен ако је испуњен услов  $f(x'') - \alpha \rho(x'', x) < f(x)$ , где је  $f(\cdot)$  вредност функције циља,  $\alpha$  унапред задати параметар, а  $\rho(x'', x)$  растојање  $x''$  од  $x$ , нпр. Хамингово растојање ако су решења вектори нула и јединица или еуклидско растојање у случају континуалних проблема. Алгоритам закошене методе променљивих околина дат је псеудокодом 2.7.

---

Иницијализација.

Изабрати скуп структура околина  $N_k$ , за  $k = 1, \dots, k_{\max}$ , које ће бити коришћене за претрагу; наћи почетно решење  $x$  и вредност  $f(x)$ ; поставити  $x_{opt} \leftarrow x, f_{opt} \leftarrow f(x)$ ; изабрати критеријум заустављања и вредност параметра  $\alpha$ ;

Понављати следећи низ корака док се не задовољи критеријум заустављања

Поставити  $k \leftarrow 1$ ;

**repeat** следеће кораке **until**  $k \leq k_{\max}$ :

- a. *Размрдавање*. Генерисати случајну тачку  $x'$  из  $k$ -те околине од  $x$ ;
- b. *Локална претрага*. Применити неку методу локалне претраге са  $x' \in N_k(x)$  као почетним решењем; означити са  $x''$  локални оптимум добијен на тај начин;
- c. *Да ли је дошло до побољшања?* Ако је  $f(x'') < f_{opt}$  поставити  $f_{opt} \leftarrow f(x'')$  и  $x_{opt} \leftarrow x''$
- d. *Да ли се померити или не?* Ако је  $f(x'') - \alpha \rho(x'', x) < f(x)$  поставити  $x \leftarrow x''$  и  $k \leftarrow 1$ , у супротном, поставити  $k \leftarrow k + 1$ ;

**end**

---

Псеудокод 2.7 - SVNS алгоритам

### 3. КЛАСТЕРОВАЊЕ

Не постоји универзална и прецизна дефиниција појма кластер, али најчешће се користи у следећим значењима:

- Кластер је скуп сличних ентитета, а ентитети из различитих кластера су различити [4].
- Кластер је агрегат тачака у простору такав да је растојање између било које две тачке у кластеру мање од растојања између тачке у кластеру и тачке која није у њему [5].
- Кластери се могу описати као непрекидне области простора са релативно великом густином тачака, раздвојене од других таквих области областима са релативно малом густином тачака [6].

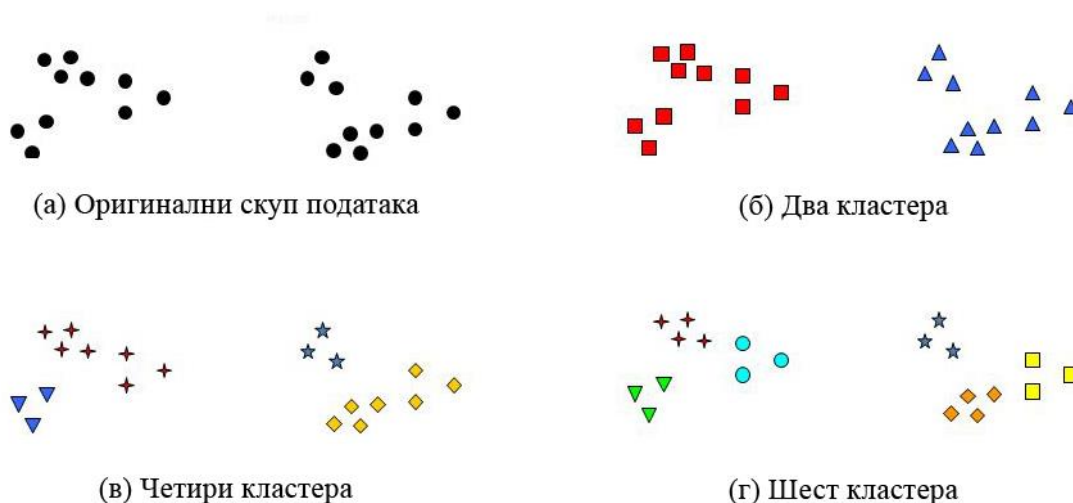
У сваком случају, објекти из истог кластера би требало да буду на неки начин међусобно слични или блиски, док би они из различитих кластера требало да буду међусобно различити, односно далеки. Широко распрострањена употреба неформалних дефиниција попут претходних открива један од доминантних и основних проблема кластер анализе: тешкоћу проналажења једне формалне, свеобухватне дефиниције кластерованја и појма кластера. Еверит [4] указује да не само да је тешко дати такву формалну дефиницију, већ би и она као таква могла бити погрешно постављена. Разлог је то што је појам кластера уопштење произвољне димензије онога што човек опажа у две или три димензије, као густе ‘слојеви’ или ‘облаци’ у простору података, људска интуиција коју је немогуће заменити једним одређеним критеријумом.

Покушајмо да дамо неке дефиниције кластерованја, које, иако можда нису универзалне, ће нам на систематски начин дати идеју о томе шта је кластерованје.

Кластер анализа проучава податке само на основу информација које описују те податке и њихових веза. Циљ је да објекти унутар групе буду слични једни другима или на неки начин повезани, а различити од објеката из друге групе, односно неповезани. Што је већа

сличност, односно хомогеност, унутар групе и већа разлика међу групама, то је кластеровање боље [7].

У многим применама, појам кластера није добро дефинисан. Следећа слика [7] илуструје у чему се огледа тешкоћа одлучивања шта сачињава кластере:



Слика 3.1 - Различити начини кластеровања истог скупа података

Слика 3.1 приказује двадесет тачака и три различита начина њихове поделе по групама. Облик маркера означава припадност кластеру. Сlike (б) и (г) деле тачке на два и шест делова, редом. Међутим, очигледна подела сваког од два већа кластера на три поткластера је производ човекове визуелне перцепције. Такође, не би било погрешно рећи да дате тачке формирају четири кластера (в). Ова слика илуструје да дефиниција кластера може бити непрецизна и да најбоља дефиниција зависи од природе података и жељених резултата.

Кластер анализа је повезана са другим техникама које се користе да поделе податке у групе. На пример, кластеровање се може посматрати као облик класификације који етикетира податке ознакама класе (кластера). Међутим, кластеровање изводи те етикете-ознаке из података. С друге стране, класификација неозначеним објектима додељује ознаку класе користећи модел развијен из објеката са познатом ознаком класе. Из тог



разлога се кластер анализа некада назива ненадгледаном (енг. *unsupervised*) класификацијом.

Такође, иако се појмови сегментација и партиционисање понекад користе као синоними за кластеровање, ови појмови се често користе за приступе ван традиционалних граница кластер анализе. Нпр. термин партиционисање се често користи у вези са техникама поделе графа на подграфове које нису строго повезане са кластеровањем. Сегментација се често односи на поделу података у групе користећи једноставне технике, нпр., фотографија се може поделити на сегменте само на основу интензитета и боје пиксела. Ипак, задаци у партиционисању графова и сегментације слике и тржишта се често тичу кластер анализе.

### **3.1. Различити типови кластровања**

Постије различите врсте кластровања, а овде ћемо разликовати неколико типова: хијерархијско насупротив партиционом, ексклузивно наспрам неексклузивног, односно наспрам фази-кластеровања, и потпуно спрам делимичног.

#### **3.1.1. Хијерархијско и партиционо кластеровање**

Највише дискутована разлика међу различитим типовима кластеровања је да ли је скуп кластера угњежден или није, или у традиционалној терминологији речено, да ли је хијерархијски или партициони. Партиционо кластеровање је проста подела скупа података у подскупове (кластере) који се не преклапају (енг. *non-overlapping*), односно имају празан пресек, тако да сваки објекат припада тачно једном скупу.

Ако се допусти да кластери имају поткластере, добија се хијерархијско кластеровање које је скуп угњеждених кластера организованих као дрво. Сваки чвор (кластер) дрвета, осим лист-чворова, је унија своје деце (поткластера), а корен дрвета је кластер који садржи све објекте. Приметимо и да се на хијерархијско кластеровање може гледати као на низ

партиционих кластеровања, а партиционо кластеровање се може добити узимањем било ког елемента тог низа, тј. узимањем одређеног нивоа хијерархијског стабла.

### 3.1.2. Ексклузивно, неексклузивно и фази-кластеровање

Сва кластеровања са слике 3.1 су ексклузивна, пошто сваки објекат додељују тачно једном кластеру. Постоје многе ситуације у којима једна тачка са разлогом може бити смештена у више од једног кластера и у тим ситуацијама је боље користити неексклузивно (енг. *overlapping*) кластеровање. У најширем смислу, коришћење неексклузивног кластеровања је последица чињенице да један објекат може истовремено припадати већем броју група или класа. Нпр., особа на факултету може бити и уписани студент и запослени факултета. Неексклузивно кластеровање се такође често користи када је, нпр., један објекат између два или више кластера и могао би се доделити било ком од ових кластера. Уместо да се објекат произвољно додељује једном кластеру, он се смешта у све једнако добре кластере.

У фази-кластеровању сваки објекат припада сваком кластеру са одређеном тежином између 0 и 1 (вероватноћом), укључујући и те две граничне вредности: 0 означава да објекат уопште не припада, а тежина 1 да потпуно припада датом кластеру. Другим речима, кластери су фази-скупови. Фази-скуп је онај коме објекат припада са одређеном тежином између 0 и 1. У фази-кластеровању додатно претпостављамо и да је збир тежина једног објекта по скуповима једнака 1. Слично, пробабилистичке технике кластеровања израчунавају вероватноћу са којом једна тачка припада сваком кластру и збир тих вероватноћа, такође, мора бити једнак 1. Пошто је збир тежина припадности или вероватноћа за неки објекат једнак 1, пробабилистичко или фази-кластеровање не одражава праву ситуацију припадности више кластера, као у примеру запосленог студента, где један објекат припада већем броју класа. Међутим, овај приступ је најпогоднији за избегавање насумичног додељивања објекта једном кластеру у ситуацији када он може бити близу неколико кластера. У пракси, пробабилистичко или фази кластеровање се често претвори у ексклузивно кластеровање додељивањем сваког објекта оном кластеру за који му је тежина припадности или вероватноћа највећа.

### **3.1.3. Потпуно и парцијално кластеровање**

Потпуно кластеровање додељује сваки објект некоем кластеру, док код парцијалног кластеровања и то не мора бити случај. Може се десити да неки објект не припада ни једној добро дефинисаној групи. Често се дешава да објекти из скупа података представљају шум или неку незанимљиву позадину.

## **3.2. Различити типови кластера**

Кластеровање има за циљ да пронађе корисне групе објеката, тј. кластере где је корисност одређена циљем анализе података. Поред различитих начина кластеровања, постоји неколико различитих врста кластера које су се показале најкориснијим у пракси. Са намером да визуелно прикажемо разлике међу овим типовима кластера, користимо дводимензионалне тачке на слици 3.2. Међутим, наглашавамо да су типови кластера описани овде једнако валидни и за друге типове података.

### **Добро раздвојени кластери (енг. *well-separated*)**

Кластер је скуп објеката у коме је сваки објект ближи или сличнији сваком објекту у том кластеру него било ком објекту ван посматраног кластера. Понекад се користи некакав праг да обезбеди да сви објекти у кластеру буду довољно блиски или слични једни другима (слика 3.2 (а)). Ови кластери не морају бити глобуларни (округли, лоптасти) већ могу имати било какав облик.

### **Кластери засновани на центру (енг. *center-based, prototype-based*)**

Кластер је скуп објеката где је сваки објект ближи центру или прототипу који дефинише тај кластер него прототипу било ког другог кластера. За податке са непрекидним атрибутима, прототип кластера је често центроид, тј. средња вредност свих тачака кластера. Када прављење центроида нема смисла, нпр. када су атрибути података

категорије, прототип је често медоид, тј. најрепрезентативнији податак кластера. Овакви кластери су обично лоптастог облика. Видети слику 3.2. (б).

### **Кластери засновани на графовима (енг. *contiguous*)**

Ако су подаци представљени као граф, где су чворови подаци, а гране представљају везе међу подацима, онда се кластер дефинише као повезана компонента, односно група објеката који су повезани један са другим али немају веза са објектима из других група. Важан пример овог типа кластеровања су кластери засновани на суседству, где су два податка повезана једино ако су на растојању мањем од неког назначеног. Одавде следи да је сваки објекат у оваквом типу кластера ближи неком другом објекту из тог кластера него било ком објекту из неког другог кластера. Овај тип кластера илустрован је на слици 3.2. (в).

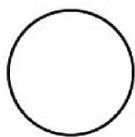
### **Кластери засновани на густини (енг. *density-based*)**

Кластер је густа област објеката која је окружена облашћу мале густине. Дефиниција кластера заснована на густини се често користи када су кластери неправиног облика или испреплетани и када је присутан шум. Супротно овом типу, кластери засновани на графовима не би добро радили на подацима са слике 3.2 (д) пошто би шум тежио да формира мостове између кластера.

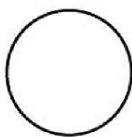
### **Концептуални кластери (енг. *conceptual*)**

Општије говорећи, можемо дефинисати кластер као скуп објеката који деле неко својство. Ова дефиниција обухвата све претходне дефиниције кластера, нпр. објекти из кластера заснованих на центрима деле својство да су сви најближи истом центроиду или медоиду. Међутим, концептуални приступ додатно укључује нове типове кластера. Размотримо кластере на слици 3.2 (е). Област троугла је припојена правоугаонику, а два круга се

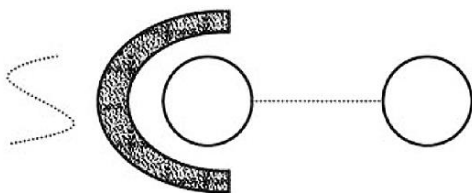
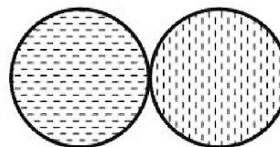
преклапају. У оба случаја алгоритам за кластеровање би морао имати врло прецизно одређен појам кластера да би успешно детектовао баш ове кластере. Процес налажења ових кластера се назива концептуално кластеровање.



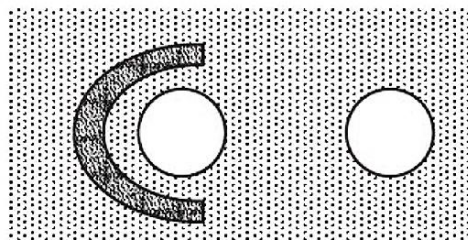
(а) Добро раздвојени кластери. Свака тачка је ближа свим осталим тачкама унутар свог кластера него било којој тачки изван њега.



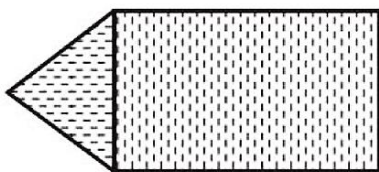
(б) Кластери засновани на центру. Свака тачка је ближа центру свог кластера него центру било ког другог кластера.



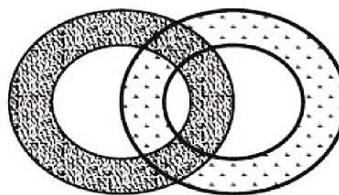
(в) Кластери засновани на графовима. Свака тачка је ближа бар једној тачки у свом кластеру него било којој тачки изван њега.



(д) Кластери засновани на густини. Кластери су области велике густине раздвојени областима мале густине.



(е) Концептуални кластери. Тачке у кластеру деле неко опште својство проистекло из целокупног скупа података. (Тачке у пресеку прстенова припадају обома.)



Слика 3.2 – Различити типови кластера

### 3.3. Индекси валидности

У највећем броју случајева за испитивање ваљаности решења алгоритма кластеровања користи се класа статистичко-математичких функција које су базиране на појмовима сличности или различитости појединачних података. Ове функције се називају индекси валидности кластера (енг. *cluster validity indices*) и могу се користити за утврђивање најбоље поделе података, као и за проналажење оптималног броја кластера.

Постоје скупови стварних података за које је утврђена позната истина (енг. *ground truth*) или златни стандард (енг. *golden standard*), тј. подела података на кластере на основу њихове стварне припадности одређеном кластеру. Индекси валидности, попут индекса чистоће (енг. *Purity*), Ранд (енг. *William M. Rand*) индекса или индекса нормализованих међусобних информацаија (енг. *Normalized mutual information*), који се ослањају на упоређивање решења кластеровања са златним стандардом, називају се индекси спољашњих критеријума (енг. *external criterion indices*).

За разлику од њих, индекси унутрашњег критеријума (енг. *internal criterion indices*), о којима ће бити речи у овом поглављу, не ослањају се на познату истину, већ се само за испитивано решење, коришћем различитих метода, утврђује квалитет кластеровања. Ове методе, односно унутрашњи критеријуми индекса валидности подељени су у три групе по начину на који испитују кластеровање и користе растојања између парова тачака, тачака и центроида својих кластера, као и растојања између центроида различитих кластера.

#### 3.3.1. Унутрашњи критеријуми индекса валидности

##### 3.3.1.1. Расутост унутар кластера

Расутост унутар кластера  $C_k$ , означена са  $WGSS^{\{k\}}$  (енг. *Within Group Sum of Squares*), представља суму квадрата растојања између сваке тачке  $M_i^{\{k\}} \in C_k$  и центроида њеног кластера  $G^{\{k\}}$  и представљена је формулом (3.1).

$$WGSS^{\{k\}} = \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|^2 \quad (3.1)$$

где је  $I_k$  скуп индекса тачака кластера  $C_k$ . Тако је укупна расутост унутар свих кластера једнака сума расутости за сваки кластер:

$$WGSS = \sum_{k=1}^K WGSS^{\{k\}} \quad (3.2)$$

где је са  $K$  означен укупан број кластера. За сваки кластер  $C_k$  сума растојања квадрата његове сваке тачке до његовог центроида такође представља и суму квадрата растојања између свих парова тачака унутар кластера подељену са бројем тачака у кластеру  $n_k$ :

$$\begin{aligned} WGSS^{\{k\}} &= \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|^2 \\ &= \sum_{i < j \in I_k} \frac{1}{n_k} \|M_i^{\{k\}} - M_j^{\{k\}}\|^2. \end{aligned} \quad (3.3)$$

Ако окренемо формулу, добијамо:

$$\begin{aligned} \sum_{i \neq j} \|M_i^{\{k\}} - M_j^{\{k\}}\|^2 &= 2 \sum_{i < j} \|M_i^{\{k\}} - M_j^{\{k\}}\|^2 \\ &= 2n_k \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|^2 \\ &= 2n_k WGSS^{\{k\}}. \end{aligned} \quad (3.4)$$

### 3.3.1.2. Расутост између кластера

Расутост између кластера, означена са  $BGSS$  (енг. *Between Group Sum of Squares*) представља расутост центроида сваког кластера у односу на барицентар целокупног скупа података. То је сума

тежина квадрата растојања између  $G^{\{k\}}$  и  $G$ , са тежином  $n_k$ , где је  $G^{\{k\}}$  центроид кластера  $C_k$  који се састоји од  $n_k$  тачака, а  $G$  барицентар целокупног скупа података:

$$BGSS = \sum_{k=1}^K n_k \|G^{\{k\}} - G\|^2. \quad (3.5)$$

### 3.3.1.3. Парови тачака

Ова метода, за разлику од расутости базираних на суми растојања тачака до центроида, о којој је било речи раније, заснива се на разликовању парова тачака који се налазе у оквиру истог кластера и парова тачака из различитих кластера.

У кластеру  $C_k$ , постоји тачно  $n_k(n_k - 1)/2$  различитих парова тачака, па се укупан број оваквих парова  $N_W$  може записати на следећи начин:

$$\begin{aligned} N_W &= \sum_{k=1}^K \frac{n_k(n_k - 1)}{2} \\ &= \frac{1}{2} \left( \sum_{k=1}^K n_k^2 - \sum_{k=1}^K n_k \right) \\ &= \frac{1}{2} \left( \sum_{k=1}^K n_k^2 - N \right). \end{aligned} \quad (3.6)$$

Укупан број различитих парова тачака у целокупном скупу података је:

$$N_T = \frac{N(N - 1)}{2}. \quad (3.7)$$

Како је  $N = \sum_{k=1}^K n_k$ , могуће је овај израз написати:



$$\begin{aligned}
N_T &= \frac{N_k(N_k - 1)}{2} = \frac{1}{2} \left( \sum_{k=1}^K n_k \right)^2 - \frac{1}{2} \sum_{k=1}^K n_k \\
&= \frac{1}{2} \left( \sum_{k=1}^K n_k^2 + 2 \sum_{k < k'} n_k n_{k'} \right) - \frac{1}{2} \sum_{k=1}^K n_k \\
&= N_W + \sum_{k < k'} n_k n_{k'}.
\end{aligned} \tag{3.8}$$

Ако се са  $N_B$  означи укупан број парова тачака које не припадају истим кластерима, важи  $N_T = N_W + N_B$ , па се може извести:

$$N_B = \sum_{k < k'} n_k n_{k'}. \tag{3.9}$$

Сваки од индекса валидности би требало да, користећи неке од унутрашњих критеријума, обезбеди мерење следећих аспеката поделе:

- Подаци унутар кластера би требало да буду што је могуће сличнији један другом.
- Подаци између два кластера би требало да буду што различитији, тј. да кластери буду раздвојени.

Ови индекси имају сличне карактеристике као функције циља и њиховим минимизовањем или максимизовањем, уз коришћење неког алгоритма за оптимизацију, може се доћи до оптималне поделе скупа података у кластере. Међутим, иако оне могу боље него раније наведене функције циља да доведу до квалитетније поделе података, често су много веће сложености, па је стога и време извршавања дуже и непрактично у конкретним проблемима.

### 3.3.2. Калински-Харабац индекс

Користећи једнакости (3.2) и (3.5), Калински-Харабац (енг. *Calinski-Harabasz*) индекс се дефинише као:

$$C = \frac{BGSS/(K-1)}{WGSS/(N-1)} = \frac{N-K}{K-1} \frac{BGSS}{WGSS}. \quad (3.10)$$

### 3.3.3. Дејвис-Болдин индекс

Означимо са  $\delta_k$  средње растојање тачака кластера  $C_k$  до њиховог центроида  $G^{\{k\}}$ :

$$\delta_k = \frac{1}{n_k} \sum_{i \in I_k} \|M_i^{\{k\}} - G^{\{k\}}\|. \quad (3.11)$$

Такође, означимо са

$$\Delta_{kk'} = d(G^{\{k\}}, G^{\{k'\}}) = \|G^{\{k\}} - G^{\{k'\}}\| \quad (3.12)$$

растојање између центроида  $G^{\{k\}}$  и  $G^{\{k'\}}$  кластера  $C_k$  и  $C_{k'}$ .

За сваки кластер  $C_k$ ,  $k = 1, \dots, K$  се рачуна максимална вредност  $\mathcal{M}_k$  количника  $\frac{\delta_k + \delta_{k'}}{\Delta_{kk'}}$  за све индексе  $k \neq k'$ . Дејвис-Болдин (енг. *Davies-Bouldin*) индекс је средња вредност количника  $\mathcal{M}_k$  узимајући у обзир све кластере:

$$C = \frac{1}{K} \sum_{k=1}^K \mathcal{M}_k = \frac{1}{K} \sum_{k=1}^K \max_{k' \neq k} \left( \frac{\delta_k + \delta_{k'}}{\Delta_{kk'}} \right). \quad (3.13)$$

### 3.3.4. Дан индекс

Означимо са  $d_{min}$  минимално растојање између тачака различитих кластера, а са  $d_{max}$  максимално растојање унутар кластера.

Растојање између кластера  $C_k$  и  $C_{k'}$  је растојање између њихових најближих тачака:

$$d_{kk'} = \min_{\substack{i \in I_k \\ j \in I_{k'}}} \|M_i^{\{k\}} - M_j^{\{k'\}}\|, \quad (3.14)$$

а  $d_{min}$  је најмање од свих растојања  $d_{kk'}$ :

$$d_{min} = \min_{k \neq k'} d_{kk'}. \quad (3.15)$$

За сваки кластер  $C_k$ , означимо са  $D_k$  највеће растојање измађу две тачке у кластеру (које се често назива полупречником кластера):

$$D_k = \max_{\substack{i, j \in I_k \\ i \neq j}} \|M_i^{\{k\}} - M_j^{\{k\}}\|. \quad (3.16)$$

Тада је  $d_{max}$  највеће од ових растојања:

$$d_{max} = \max_{1 \leq k \leq K} D_k. \quad (3.17)$$

Дан (енг. *Dunn*) индекс се дефинише као количник између  $d_{min}$  и  $d_{max}$ .

### 3.3.5. Силуета индекс

Нека је средње растојање  $a(i)$  тачке  $M_i$  унутар кластера дефинисано као средње растојање тачке  $M_i$  до осталих тачака кластера коме она припада, тј. ако је  $M_i \in C_k$ , биће

$$a(i) = \frac{1}{n_k - 1} \sum_{\substack{i' \in I_k \\ i' \neq i}} d(M_i, M_{i'}). \quad (3.18)$$

С друге стране, нека је  $\mathfrak{d}(M_i, C_{k'})$  средње растојање тачке  $M_i$  до тачака свих осталих кластера  $C_{k'}$

$$\mathfrak{d}(M_i, C_{k'}) = \frac{1}{n_{k'}} \sum_{i' \in I_{k'}} d(M_i, M_{i'}). \quad (3.19)$$

Такође, означимо најмање од ових средњих растојања

$$b(i) = \min_{k \neq k'} d(M_i, C_{k'}). \quad (3.20)$$

Вредност  $k'$  за коју се постиже овај минимум указује на најбољи избор кластера у који би требало пребацити тачку  $M_i$ .

За сваку тачку  $M_i$  формира се количник

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3.21)$$

који представља ширину силуете за тачку  $M_i$ . Његова вредност се креће између -1 и 1. Вредност близу 1 указује да је тачка придружена правом кластеру, док вредност близу -1 указује да би тачку требало придружити другом кластеру.

Средња вредност ширина силуете за дати кластер  $C_k$  се назива средња силуета кластера и означава се са  $s_k$ :

$$s_k = \frac{1}{n_k} \sum_{i \in I_k} s(i). \quad (3.22)$$

На крају, глобални силуета индекс је средња вредност средњих силуета свих кластера:

$$C = \frac{1}{K} \sum_{k=1}^K s_k. \quad (3.23)$$

### 3.3.6. CS мера

CS мера представља средњу вредност просечних растојања између сваке тачке  $i$  и њој најдаље тачке  $j$  унутар истог кластера подељену са средњом вредношћу растојања између центроида сваког кластера  $k$  и њему најближег центроида неког другог кластера  $k'$ :

$$CS(K) = \frac{\frac{1}{K} \sum_{k=1}^K \left[ \frac{1}{n_k} \sum_{i \in I_k} \max_{j \in I_k} \|M_i^{\{k\}} - M_j^{\{k\}}\| \right]}{\frac{1}{K} \sum_{k=1}^K \min_{k' \neq k} \|G^{\{k\}} - G^{\{k'\}}\|} \quad (3.24)$$

где је  $K$  укупан број кластера,  $\max_{j \in I_k} \|M_i^{\{k\}} - M_j^{\{k\}}\|$  максимално растојање између  $i$ -те тачке  $k$ -тог кластера и неке друге тачке истог кластера,  $n_k$  број тачака у  $k$ -том кластеру, а  $\min_{k' \neq k} \|G^{\{k\}} - G^{\{k'\}}\|$  најмање растојање између центроида  $k$ -тог и неког другог кластера.

## 4. МЕТОДА ПРОМЕНЉИВИХ ОКОЛИНА ЗА ПАРТИЦИОНО КЛАСТЕРОВАЊЕ

Нека је  $X$  скуп од  $n$  тачака у  $d$  димензионом простору.

$$X = \{x_1, \dots, x_n\},$$

$$x_j = (x_{1j}, \dots, x_{dj}) \in \mathbb{R}^d. \quad (4.1)$$

Циљ је наћи поделу скупа  $X$  на  $k$  дисјунктних подскупова или кластера  $C_i$  тако да је вредност изабране функције циља минимална. Ако је реч о еуклидском простору, пример функције циља може бити сума квадрата растојања сваке тачке од центроида  $G_i$  њеног кластера:

$$\min \sum_{i=1}^k \sum_{x_l \in C_i} \|x_l - G_i\|^2$$
$$G_i = \frac{1}{n_i} \sum_{x_l \in C_i} x_l \quad (4.2)$$

где са  $\|\cdot\|$  означена еуклидска норма и

$$n_i = |C_i| \quad (4.3)$$

и важи

$$n_1 + \dots + n_K = N \quad (4.4)$$

Поред наведеног, у зависности од сврхе кластеровања функција циља може бити дефинисана на много других начина, што ће детаљније бити размотрено у наредном поглављу.

Нека је дата почетна подела скупа  $X$  на кластере и нека она уједно представља и почетно решење за VNS алгоритам. Такође, нека је дефинисана структура околина произвољног решења. Околина једног решења обично садржи решења на неки начин њему блиска. У најпростијем случају, околина решења се добија свим могућим пребацивањима тачке  $x_j$  из кластера  $C_l$  тренутног решења у кластер  $C_i$ ,  $i \neq l$ . Затим се тражи најбољи сусед у смислу минималне вредности функције циља у околини почетног решења. Ако је најбоље решење нађено у околини боље од тренутног, долази до померања у новонађено решење и алгоритам наставља са претрагом његове околине. У супротном, ако боље решење од тренутног није пронађено, алгоритам стаје.

У наставку овог поглавља детаљније ће бити размотрени делови VNS алгоритма попут: функције циља, околина, размрдавања, локалног претраживања, критеријума заустављања и др.

#### 4.1. Функција циља

Задатак рачунарског кластеровања података је обрађиван у различитим научним областима попут теорије графова, статистике, вештачких неуронских мрежа, теорије фази скупова итд. Један од најпопуларнијих приступа у овом смеру је формулација кластеровања као проблема оптимизације, где се најбоља подела датог скупа података постиже минимизовањем/максимизовањем једне (енг. *single-objective clustering*) или више (енг. *multi-objective clustering*) функција циља. Функција циља се обично дефинише тако да представља одређену статистичко-математичку везу између појединачних података и скупа репрезентативних ентитета сваког кластера (центроида). [10]

Две методе кластер анализе могу се разликовати по одабиру функције циља, тј. да ли је она усмерена на мерење хомогености унутар кластера или хетерогености између кластера, као и према процедури позивања мере. Функционална зависност изабрана као мера унутар-кластерске хомогености (или међукластерске хетерогености) углавном одражава пожељан начин груписања и зависи од природе проблема.

Избор функције циља и њена ефикасност зависе од циља кластеровања и датог скупа података. Да би се говорило о томе да ли је кластеровање добро, потребно је претходно јасно утврдити шта је циљ кластеровања. Најчешће су то јасна раздвојеност кластера, хомогеност унутар кластера (мала растојања), равномерна распоређеност тачака унутар кластера, добра репрезентација података кластера помоћу центроида, формирање области високе густине без празнина унутар себе, формирање кластера приближно уједначених величина и облика итд. Сходно изабраном циљу кластеровања треба изабрати адекватну функцију.

Један од најчешће коришћених примера функције циља је сума квадрата растојања тачака кластера од центроида тог кластера, дата у уводу овог поглавља. Ова функција циља је погодна за кластеровање скупова података са јасно раздвојеним кластерима без преклапања, сфероидног облика и уједначених величина. Предност ове функције је њена релативно мала сложеност  $O(n)$  што значајно може уштедети време и простор при извршавању VNS алгоритма, нарочито при учесталом позивању функције циља. Међутим, главна мана ове функције циља је што се добри резултати кластеровања могу постићи само за скупове података са кластерима сличних облика и величина.

Функција циља која превазилази овај проблем је сума просечних растојања парова тачака унутар кластера:

$$\min \sum_{i=1}^m \frac{1}{n_i(n_i - 1)} \sum_{\substack{x_l \in C_i \\ x_l \in C_i \\ l < l'}} \|x_l - x_{l'}\|^2, \quad (4.5)$$

Мана ове функције у односу на претходно наведену је за степен већа сложеност.

## 4.2. ОкоLINE

Основна идеја VNS-а је базирана на систематској промени околина што омогућава проналажење локалних минимума, као и излажење из њих. Приликом избора структура, околина треба водити рачуна о питањима наведеним у поглављу 2.3 [11]:



Прва два питања се тичу способности VNS хеуристике да пронађе најбоље долине и да то уради релативно брзо. Да би се избегло упадање у неку долину, док можда постоје и дубље од ње, унија околина око неког допустивог решења  $x$  би требала да садржи цео допустив скуп.

$$X \subseteq N_1(x) \cup N_2(x) \cup \dots \cup N_{k_{max}}(x), \quad \forall x \in X \quad (4.6)$$

Док ови скупови покривају  $X$  није неопходно да они међусобно буду дисјунктни, што је лакше за имплементацију, нпр. када се користе угњеждене околине, тј.

$$N_1(x) \subset N_2(x) \subset \dots \subset N_{k_{max}}(x), \quad X \subset N_{k_{max}}(x), \quad \forall x \in X \quad (4.7)$$

С друге стране, ако својства околина не испуњавају овај услов, још увек је могуће претражити цео простор  $X$  претраживањем малих околина око тачака на одређеној путањи (добијеној померањем у све боље решење), али се то више не може гарантовати. Примера ради, у већ поменутом кластеровању које користи минимум суме квадрата, околине дефинисане пребацивањем једне или више тачака из једног кластера у други не омогућавају алгоритму да у многим случајевима напусти локални оптимум. За разлику од тога, померање центара кластера превазилази наведени проблем. [11]

Како се највећи број класичних алгоритама за кластеровање заснованих на оптимизацији ослања на технике локалног претраживања за оптимизацију функције циља, велика мана њихових метода локалне претраге је склоност метода упадању у неки од локалних оптимума функције циља.

Једна од основних структура околина, која ће управо бити објашњена, превазилази овај проблем.

#### 4.2.1. Структура околина 1

Угњеждене околине се лако добијају за многе комбинаторне проблеме дефинишући прву околинину  $N_1(x)$  типом премештања, а онда се остале околине  $N_k(x)$ ,  $k = 2, \dots, k_{max}$  добијају понављајући то премештање  $k$  пута. Својство овако дефинисаних структура је да

се њихова величина повећава са повећањем  $k$ . Стога, као што је често случај, ако се много пута пролази кроз читав низ околина, оне прве ће бити темељније истражене од последњих. У светлу чињенице под редним бројем 3 из увода у друго поглавље овог рада ово је пожељно својство, јер локални оптимуми теже да буду близу један другом.

Нека је  $S$  скуп од  $n$  тачака које желимо да поделимо у  $m$  кластера,  $m \geq 2$ . Нека је  $X$  скуп  $n$ -димензионалних вектора са целобројним вредностима координата, где свака координата означава редни број кластера коме припада одговарајућа тачка, тј.  $X$  садржи све векторе  $x = (x_1, \dots, x_n)$ , где је  $x_i$  редни број кластера за  $i$ -ту тачку,  $1 \leq x_i \leq m$ .  $k$ -ту околину  $N_k(x)$  вектора  $x$  чине сви вектори којима је  $k$  тачака премештено у друге кластере, односно:

$$N_k(x) = \{x' = (x'_1, \dots, x'_n) \mid x_i \neq x'_i, i \in I, I \subset \{1, 2, \dots, n\} \wedge |I| = k\} \quad (4.8)$$

$k$ -ту околину допустивог решења  $x$  добијамо пребацивањем његових  $k$  од  $n$  тачака у неки од  $m - 1$  преосталих кластера, тј.

$$|N_k| = \binom{n}{k} (m - 1). \quad (4.9)$$

Одавде се примећује да за велике  $n$  и  $m$  добијамо околине велике кардиналности, непрактичне чак и за рачунарска извршавања.

$$\begin{aligned} \sum_{k=1}^m |N_k| &= \sum_{k=1}^m \binom{n}{k} (m - 1) \\ &= (m - 1) \left( n + \frac{n(n-1)}{2} + \frac{n(n-1)(n-2)}{3!} \right. \\ &\quad \left. + \frac{n \cdot \dots \cdot (n-m+1)}{m!} \right) \\ &\sim O\left(\left(\frac{n}{m}\right)^m\right). \end{aligned} \quad (4.10)$$

Овако дефинисане околине имају поменуто својство инклузивности, тј.

$$N_1(x) \subset N_2(x) \subset \dots \subset N_{n-1}(x) \quad (4.11)$$

и такође важи

$$N_1(x) \cup N_2(x) \cup \dots \cup N_{n-1}(x) = X \quad (4.12)$$

где у оба случаја ово важи  $\forall x \in X$ .

Пролажењем кроз све векторе овако дефинисаних околина, обилазе се сва допустива решења, што увек гарантује проналажење глобалног оптимума.

Узрок лошег понашања неке локалне претраге може бити непотребно посећивање свих решења из околине. У раду [11] Хансен и Младеновић предлажу да се пронађу подскупови околина за које се потенцијално очекује да ће дати боље решење и да се само они посете приликом локалне претраге. У најбољем случају требало би пронаћи правило по коме се аутоматски заобилазе решења из околине чија вредност функције циља није боља од тренутне.

Један од начина да се ово реши је да се структура околина дефинише на следећи начин.

#### 4.2.2. Структура околина 2

Нека је  $x$  почетно решење проблема поделе скупа  $S$  од  $n$  тачака у  $m$  кластера представљено било којом таквом поделом. Нека је сваком од кластера израчунат центроид по формули (4.2).

Нека је изабран један од  $m$  кластера решења  $x$ . За сваку његову тачку се проналази најближи од  $m$  центроида. Ако се најближи центроид разликује од центроида кластера у коме се та тачка налази, онда се та тачка пребације у кластер најближег центроида. У супротном ништа се не мења. Све могуће поделе добијене избором једног од  $m$  кластера и оваквим пребацивањем одговарајућих тачака изабраног кластера чине прву околину решења  $x$ .

Нека је сада изабрано  $k$  од  $m$  кластера решења  $x$ . Израчунава се барицентар свих тачака из  $k$  изабраних кластера по поменутој формули (2.4). Центроид који зависи од тачака из различитих кластера зваћемо барицентар. За сваку тачку из  $k$  изабраних кластера налази се најближи од пресоталих  $m - k$  центроида. Ако је тачка ближа овако добијеном центроиду него барицентру, онда се она пребацује у кластер тог центроида.

Приликом пребацивања тачака из једног кластера у други може доћи до пражњења једног или више кластера. Претпоставимо да се на овакав начин испразни  $i$  кластера. Налазимо кластер  $C$  са највећим полупречником, где је полупречник дефинисан као највеће растојање неке тачке кластера од његовог центроида. За центроид празног кластера постављамо пронађену тачку која има највеће растојање од центроида кластера  $C$  и пребацујемо је у тај кластер. За сваку тачку кластера  $C$  проверавамо да ли је ближа његовом центроиду или новопостављеном центроиду празног кластера. Ако је ближа центроиду празног кластера, пребацујемо је у тај кластер. На тај начин је извршена подела кластера  $C$ . Овај поступак поделе понављамо  $i$  пута, с напоменом да се у свакој наредној итерацији у разматрање узимају и кластери добијени поделом у претходној итерацији.

Све могуће поделе добијене оваквим избором  $k$  од  $m$  кластера и пребацивањем одговарајућих тачака изабраних кластера чине  $k$ -ту околину решења  $x$ .

Одмах се примећује да овако формиране околине имају далеко мањи број суседа полазног решења него претходни пример околина. Како се од полазног решења, бирањем једног кластера и пребацивањем одговарајућих тачака, може добити тачно један сусед полазног решења, то се у првој околини бирањем  $m$  различитих кластера може добити највише  $m$  различитих суседа. Слично се у  $k$ -тој околини избором једне комбинације од  $k$  различитих кластера и пребацивањем њених тачака може добити тачно један сусед полазног решења  $x$  из те околине. Како је број различитих комбинација  $k$  изабраних кластера од  $m$  различитих  $\binom{m}{k}$ , то је број суседа у  $k$ -тој околини  $\binom{m}{k}$ .

Максималан број околина у овако дефинисаној структури је  $m - 1$ . Укупан број суседа у свим околинама решења  $x$  је

$$\sum_{k=1}^{m-1} \binom{m}{k} = \sum_{k=0}^m \binom{m}{k} - \binom{m}{0} - \binom{m}{m} \quad (4.13)$$

$$= 2^m - 2$$

што је много мање од (4.10), броја свих могућих суседа.

Додатно убрзање алгоритма може се постићи дељењем постојеће околине на ситније делове и давањем предности избору суседа који имају веће шансе да дају боље резултате. Оваквим поделама се често прибегава у фази локалног претраживања. Један пример који се ослања на овде дефинисану околину биће изложен у следећем поглављу.

### 4.3. Локално претраживање

Два основна начина рада локалног претраживања представљају проналажење најбољег решења унутар дате околине и проналажење првог бољег решења унутар исте. За проналажење најбољег суседа потребно је посетити све суседе одређеној околини, што за удаљене околине може бити врло непрактично због потенцијално великог броја сусуда. За разлику од овога проналажење првог побољшања је знатно брже, мада не гарантује проналажење најбољег решења у тој околини. На путу до оптималног решења локална претрага се позива више пута и ближе околине се темељније истражују.

Када се спроводи локална претрага могуће је поделити околину која се претражује на више мањих околина и применити неку од варијанти VNS-а за претраживање те околине. На примеру околина дефинисаних у претходном поглављу, биће илустрована оваква даља подела.

Произвољна  $k$ -та околина описана у 4.2.2. се може поделити тако да се не узимају све могуће комбинације кластера реда  $k$ , већ комбинације релативно блиских кластера, јер оне имају веће шансе да брже доведу до бољих резултата. Блиским кластерима се сматрају кластери са међусобно блиским центроидима. Прецизније говорећи, ако се дели  $k$ -та околина VNS-а, изабере се један од  $m$  кластера и пронађе се његових  $k - 1$  кластера

суседа. Прву околину локалне претраге у  $k$ -тој околини VNS-а чине сви могући избори једног кластера са његових  $k - 1$  најближих кластера. Тиме се постиже да као и до сада у  $k$ -тој околини VNS-а спајамо  $k$  кластера и рачунамо њихов барицентар, само што се сада најпре узимају међусобно најближи кластери. За  $j$ -ту околину локалне претраге је, такође, потребно на почетку изабрати један од  $m$  кластера. Из скупа од  $k - 1 + j$  најближих кластера формирају се све могуће комбинације реда

$$k - 1 \quad (4.14)$$

и придружују посматраном кластеру. Сва таква придруживања могућих комбинација неком од  $m$  кластера чине  $j$ -ту околину локалног претраживања у  $k$ -тој околини VNS-а. Имајући у виду овакву поделу VNS околине, може се приметити да се у првој околини VNS-а једном од изабраних кластера не придружује ни један други кластер, што следи из (4.154) по коме се у  $k$ -тој околини изабраном кластеру придружује  $k - 1$  кластер.

#### 4.4. Избор почетног решења

Поред склоности упадања у локалне минимуме функције циља, перформансе метода локалне претраге су често врло осетљиве на почетне вредности параметара претраге [10]. Избор почетног решења може значајно утицати на време потребно VNS-у да дође до глобално оптималног решења. Један од начина да се формира почетно решење јесте да се свака од  $n$  тачака скупа  $S$ , који се кластерује, припоји једном од  $m$  кластера. Наравно, приликом таквог формирања кластера мора се водити рачуна о томе да се сваком кластеру припоји бар једна тачка, како неки од кластера не би остао празан. Тестирање VNS алгоритма са овако одабраним решењем показало је слабије перформансе и склоност упадању у локалне минимуме који нису ни близу глобалном оптимуму.

Најједноставније побољшање избора почетног решења може се постићи тако што се из скупа од  $n$  тачака случајно изабере  $m$  тачака и свака од осталих тачака се придружи најближој од  $m$  изабраних. Тако формирана подела улазних података представља почетно решење проблема. Овако изабрано почетно решење олакшава алгоритму проналажење

оптималног решења, јер су међусобно блиске тачке већ добрим делом придружене истом кластеру.

#### **4.5. Оптимални број кластера**

Иако неки радови из области препознавања облика (енг. *pattern*) описују кластеровање као методу ненадгледаног учења (енг. *Unsupervised Learning Method*), највећи број традиционалних алгоритама за кластеровање захтева унапред одређен број кластера у процесу поделе података у кластере, што га чини не баш потпуно ненадгледаним. С друге стране, у пракси често није могуће знати чак ни процену броја природно раздвојених кластера у скупу података који до тада није обрађиван. У таквим случајевима, ако алгоритам за кластеровање захтева број кластера као улазни параметар, обично се алгоритам покреће за више различитих бројева кластера на улазу. За оптималан број кластера се узима онај број за који је алгоритам дао најбоље решење.

#### **4.6. Критеријуми заустављања**

Најопштије говорећи, VNS алгоритам се састоји од секвенцијалног примењивања методе размрдавања и методе локалне претраге у дефинисаним околинама посматраног решења. Један пар извршавања размрдавања и локалне претраге представља једну итерацију овог алгоритма. У идеалном случају да би се дошло до оптималног решења, требало би обићи сва допустива решења и израчунати вредност функције циља за свако од њих. Алгоритам би стао пошто би завршио обилажење свих допустивих решења, а за оптимално решење би се узело оно за које је функција циља имала најмању вредност. Међутим, код проблема великих димензија број допустивих решења је толико велики да би обилажење свих захтевало непрактично много времена. Због тога је најчешће неопходно дефинисати неки други критеријум заустављања, а решење са најмањом функцијом циља које се пронађе док се не задовољи критеријум заустављања се узима за апроксимацију оптималног решења.

Једноставан критеријум заустављања који се често примењује је број итерација алгоритма. Његова основна мана је што нам произвољан избор максималног броја итерација не гарантује да се већ у некој од следећих итерација неће наћи значајно боље решење. У једној итерацији алгоритам не мора обавезно доћи до бољег решења у односу на решење постигнуто у некој од претходних итерација. Ако су околине тако дефинисане да неке од њих имају велики број суседа или постоји велики број околина које се претражују, то може довести до дугог рада алгоритма без побољшања. Због тога се често као критеријум заустављања користи број узастопних итерација алгоритма у којима није постигнуто никакво побољшање. На тај начин се смањује вероватноћа да ће се у некој од наредних итерација пронаћи значајно боље решење. С друге стране, овај критеријум заустављања допушта врло мала побољшања која не доводе до посебно бољег решења, а често могу значајно продужити време извршавања алгоритма. Да би се то избегло дефинише се жељена минимална величина побољшања и број узастопних итерација  $p$  за које се проверава да ли су довеле до побољшања већег од дефинисаног минималног побољшања. Ако је у  $p$  узастопних итерација побољшање мање од задатог минимума, алгоритам стаје.



## 5. ИМПЛЕМЕНТАЦИЈА АЛГОРИТМА

Улазни параметри VNS алгоритма су број кластера и максималан број итерација.

Почетно решење се генерише на случајан начин. Из датог скупа тачака се насумично изабере онолико тачака колики је број кластера и назову се центроидима, а све остале тачке датог скупа се припоје најближем центроиду. Након завршеног припајања свих тачака најближем центроиду, формирају се почетни кластери. Да би почетно решење било конзистентно, неопходно је прерачунати вредности центроида за сваки кластер после завршеног припајања тачака. На тај начин се добије жељени број насумично генерисаних кластера, што представља почетно решење.

### 5.1. Основни VNS

За имплементацију алгоритма коршћена је друга варијанта структура околина описана у 4.2.2. За дато решење  $x$ , околина  $N_k(x)$  ће садржати  $\binom{m}{k}$  суседа. За овако дефинисане околине неопходан услов је да је број околина до ког алгоритам иде строго мањи од укупног броја кластера. Бројач итерација се на почетку постави на 1.

VNS алгоритам почиње фазом размрдавања у којој се из задате околине решења  $x$  произвољно бира ново решење. То решење из  $k$ -те околине решења  $x$  се добија тако што се случајно изабере  $k$  кластера, прерачуна се њихов барицентар и за сваку тачку изабраних кластера се провери да ли је ближа барицентру или неком од центроида преосталих кластера. Ако се испостави да постоји један или више центроида на мањем растојању до посматране тачке од њеног барицентра, посматрана тачка се пребацује у кластер чији јој је центроид најближе. После свих оваквих пребацивања се проверава да ли је и колико од изабраних кластера остало без иједне тачке. Нека је, ради разматрања тог специфичног случаја, остало  $i$ ,  $i < k$  кластера празно, да се не би нарушио услов да наше решење има  $m$  кластера,  $i$  пута ћемо поделити највећи кластер и тиме на крају фазе размрдавања у сваком случају имати решење са  $m$  кластера. Највећим кластером се сматра кластер који има највеће растојање између свог центроида и неке од својих тачака. За нови

центроид једног од испражњених кластера се узима најудаљенија тачка од свог центроида и припаја се том испражњеном кластеру. За све остале тачке највећег кластера се проверава јесу ли ближе новопостављеном центруиду него центруиду највећег кластера. Ако је тако, тачка се пребацује у тај испражњени кластер. На тај начин је највећи кластер подељен на два дела. Ово дељење се понови још  $i - 1$  пута, с тим што се приликом проналажења највећег кластера сада разматра и новонастали кластер добијен дељењем највећег кластера у претходној итерацији. На крају процеса размрдавања добијено решење означимо са  $x'$ .

Локално претраживање креће у првој околини,  $k = 1$ , решења  $x'$ . Тражи се најбољи сусед у тој околини. Када се испитају сви суседи у тој околини, проверава се да ли је вредност функције циља за најбољег суседа  $x''$  боља од вредности функције циља за решење  $x$ . Ако јесте,  $x$  постаје  $x''$ ,  $x \leftarrow x''$ ,  $k = 1$ , и алгоритам наставља из новог решења  $x$ . Ако није, индекс околине која се претражује се повећава за 1,  $k = k + 1$ . Ако се достигне максимални број околина, а не пронађе се боље решење од тренутног  $x$ , бројач итерација се увећава за 1 и алгоритам креће испочетка.

Алгоритам стаје када се достигне максималан број итерација дефинисан као улазни параметар алгоритма.

---

*/\* Иницијализација \*/*

$S$  је скуп тачака које се кластерују;

Изабрати жељени број кластера  $m$ ;

Изабрати максималан број итерација макс\_број\_итерација;

Изабрати структуре околина  $N_k$ ,  $k = 1, \dots, m$  које ће се користити за размрдавање и претрагу;

На случајан начин изабрати почетно решење  $x \in X$ :

Насумично изабрати  $m$  тачака из  $S$ ;

За сваку од  $m$  тачака формирати кластер који је садржи и нека је она његов центроид  $c_i$ ,  $i = 1, \dots, m$ ;

Све остале тачке из скупа  $S$  припојити кластеру са најближим центроидом;

Прерачунати центроиде за сваки кластер;

Поставити  $x^* \leftarrow x$  и  $f^* \leftarrow f(x)$ ;

Поставити број\_итерација = 1;

**repeat** наредне кораке **until** број\_итерација  $\leq$  макс\_број\_итерација

Поставити  $k = 1$ ;

```

start:  repeat наредне кораке until  $k < m$ 
        /* Размрдавање */
        Насумично изабрати  $k$  кластера;
        Израчунати њихов барицентар  $b$ ;
        За сваку тачку  $t$  из  $k$  кластера урадити следеће
            пронаћи најближи центроид  $c$  од преосталих  $m - k$  центроида;
            if растојање од  $t$  до  $c <$  растојање од  $t$  до  $b$  then
                Пребацити тачку  $t$  у кластер са центроидом  $c$ 
        Нека је  $i$  број кластера који су остали празни,  $i \leq k$ 
         $j = 1$ ;
        repeat наредне кораке until  $j \leq i$ 
            Пронаћи кластер  $C'$  са највећим растојањем између његовог центроида  $c'$  и
            неке његове тачке  $t'$ ;
            Припојити тачку  $t'$   $j$ -том празном кластеру и поставити је за његов центроид;
            Све тачке кластера  $C'$  које су ближе тачки  $t'$  него центроиду  $c'$  пребацити у  $j$ -ти
            кластер
             $j$ -том кластеру и кластеру  $c'$  прерачунати центроиде.
             $j = j + 1$ ;
        end
        Овако добијено решење означити са  $x'$ 
        /* Локална претрага */
        У  $k$ -тој околини решења  $x'$  пронаћи најбољег суседа  $x''$ 
        if  $f(x'') < f^*$  then
            Поставити  $x^* \leftarrow x''$  и  $f^* \leftarrow f(x'')$  goto start;
        else
            Поставити  $k = k + 1$ ;
        end
        број_итерација = број_итерација + 1;
end

```

---

Псеудокод 5.1 - Детаљна шема имплементације BVNS-а за проблем кластеровања  
скупа тачака  $S$  у  $m$  кластера

Решење  $x^*$  је апроксимативно решење проблема.

## 5.2. Општи VNS

Следи псеудокод имплементације опште модификације VNS-а која се од основне разликује у примени VND алгоритма у фази локалне претраге. ОкоLINE коришћене у VND-у су описане у поглављу 4.3, а овде је изложен псеудокод без понављања објашњења.

---

```

/* Иницијализација */
S је скуп тачака које се кластерују;
Изабрати жељени број кластера  $m$ ;
Изабрати максималан број итерација макс_број_итерација;
Изабрати структуре околина  $N_k$ ,  $k = 1, \dots, m - 1$  које ће се користити за размрдавање;
Изабрати структуре околина  $N_l$ ,  $l = 1, \dots, m - (k - 1)$ ,  $m > k + 1$  које ће се користити за локалну претрагу методом VND-а;
На случајан начин изабрати почетно решење  $x \in X$ :
    Насумично изабрати  $m$  тачака из  $S$ ;
    За сваку од  $m$  тачака формирати кластер који је садржи и нека је она његов центроид
     $c_i$ ,  $i = 1, \dots, m$ ;
    Све остале тачке из скупа  $S$  припојити кластеру са најближим центроидом;
    Прерачунати центроиде за сваки кластер;
Поставити  $x^* \leftarrow x$  и  $f^* \leftarrow f(x)$ ;
Поставити број_итерација = 1;
repeat наредне кораке until број_итерација  $\leq$  макс_број_итерација
    Поставити  $k = 1$ ;
start: repeat наредне кораке until  $k < m$ 
    /* Размрдавање */
    Извршити кораке размрдавања на исти начин као у BVNS
end
Овако добијено решење означити са  $x'$ ;
/* Локална претрага */
Поставити  $l = 1$ ;
Поставити  $x^{**} \leftarrow x'$  и  $f^{**} \leftarrow f(x')$ 
repeat наредне кораке until  $l < m - (k - 1)$ 
    Направити скуп  $L$  свих комбинација за одабир  $k$  од  $m$  кластера у околини  $l$ :
    За сваки кластер  $C$  урадити следеће:
        Пронаћи  $k + l - 2$  најближа центроида центроиду кластера  $C$ ;
        Свакој комбинацији одабира  $k - 1$  од  $k + l - 2$  кластера припојити
        кластер  $C$  и додати је у  $L$ ;
    Пронаћи најбоље решење у околини  $l$  решења  $x'$ :
    За сваку комбинацију  $e \in L$  урадити следеће:
    Израчунати барицентар  $b$  за све тачке  $k$  кластера комбинације  $e$ ;
    За сваку тачку  $t$  из  $k$  кластера комбинације  $e$  урадити следеће:
        пронаћи најближи центроид  $c$  тачки  $t$  од преосталих  $m - k$  центроида;
        if растојање од  $t$  до  $c <$  растојање од  $t$  до  $b$  then
            Пребацити тачку  $t$  у кластер са центроидом  $c$ 
        Нека је  $i$  број кластера који су остали празни,  $0 \leq i \leq k$ 
         $j = 1$ ;
        repeat наредне кораке until  $j \leq i$ 
            Пронаћи кластер  $C'$  са највећим растојањем између његовог
            центроида  $c'$  и неке његове тачке  $t'$ ;
            Припојити тачку  $t'$   $j$ -том празном кластеру и поставити је за
            његов центроид;

```

```

        Све тачке кластера  $C'$  које су ближе тачки  $t'$  него центроиду  $c'$ 
        пребацити у  $j$ -ти кластер
         $j$ -том кластеру и кластеру  $c'$  прерачунати центроиде.
         $j = j + 1$ ;
    end
    Овако добијено решење означити са  $x''$ ;
    if  $f(x'') < f^{**}$  then
        Поставити  $x^{**} \leftarrow x''$  и  $f^{**} \leftarrow f(x'')$ ;
        Решење  $x^{**}$  је најбољи сусед решења  $x'$  у околини  $l$ ;
        if  $f^{**} < f(x')$  then
            break;
        else
            Поставити  $l = l + 1$ ;
    end
    if  $f^{**} < f^*$  then
        Поставити  $x^* \leftarrow x^{**}$  и  $f^* \leftarrow f^{**}$  goto start;
    else
        Поставити  $k = k + 1$ ;
        број_итерација = број_итерација + 1;
end

```

---

Псеудокод 5.2 - Детаљна шема имплементације GVNS-а за проблем кластеровања  
скупа тачака  $S$  у  $m$  кластера

## 6. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Као што је већ речено, циљ кластеровања је, међу свим могућим поделама скупа података, пронаћи најбољу поделу према некој мери квалитета. Међутим, ако претпоставимо да је скуп података које кластерујемо заправо узорак са стварним класама у позадини, циљ није наћи најбољу поделу датог узорака, већ апроксимацију стварне поделе.

У истраживању су коришћени скупови стварних и генеричких података. Скупови јавно доступних стварних података су преузети са сајта <https://archive.ics.uci.edu/ml/datasets/>, док су три примера генеричких скупова преузета са сајта <https://cs.joensuu.fi/sipu/datasets/>, а четири су случајно генерисана за потребе овог рада.

Прво ће бити размотрени резултати добијени коришћењем скупова стварних података. Овде је  $n$  број података у скупу,  $d$  је број њихових својстава и  $k$  ја број кластера.

1. Ирис (енг. *Iris plants database*)  $n = 150$ ,  $d = 4$ ,  $k = 3$  [18]: Ово је врло добро познат скуп података који има 150 појединачних података, сваки податак има 4 посматрана својства и припада једном од 3 кластера. Скуп података се састоји од три различите врсте цвета ирис: *Iris setosa*, *Iris virginica* и *Iris versicolour*. За сваку врсту је сакупљено по 50 узорака са 4 посматрана својства (дужина листића чашице, ширина листића чашице, дужина латице, ширина латице).
2. Стакло (енг. *Glass*)  $n = 214$ ,  $d = 9$ ,  $k = 6$  [19]: Ови подаци су узети из 6 различитих врста стакала: плутајуће<sup>1</sup> стакло за прозоре зграда (70), неплутајуће стакло за прозоре зграда (79), плутајуће стакло за прозоре возила (17), стакло за посуде (13), стакло за столове (9) и стакло за фарове (29), сваки тип са 9 посматраних својстава. То су индекс рефракције, натријум, магнезијум, алуминијум, силицијум, калијум, калцијум, баријум и гвожђе.
3. Вино (енг. *Wine*)  $n = 178$ ,  $d = 13$ ,  $k = 3$  [20]: Овај скуп података је резултат хемијске анализе винових лоза које расту у истом делу Италије, али потичу од различитих

---

<sup>1</sup>Енг. *float glass*. Плутајуће стакло је стаклена плоча направљена изливањем стакла у метални калуп. Ова метода обезбеђује уједначену дебљину и веома равну површину стакла.

сорт. Анализом је установљена количина 13 различитих састојака пронађених у свакој од три врсте вина.

4. Скуп самогласника (енг. *Vowel*)  $n = 871$ ,  $d = 3$ ,  $k = 6$  [21]: Овај скуп података се састоји од 871 звучног записа индијанских самогласника. Сваки има три посматрана својства која се налазе у неком фреквенцијском опсегу и припада једном од шест кластера који се преклапају.

У табели 6.1 приказани су резултати који су добијени различитим модификацијама VNS-а примењеним на управо описане скупове података. Свака модификација је за одређени скуп података тестирана 20 пута и у табели су дате просечне вредности за сваки од посматраних параметара. У колони VNS је списак модификација VNS-а које су тестиране, а редови означени као "позната истина" садрже вредности засноване на реалној припадности појединачног податка одређеном кластеру. Број поред закошене модификације VNS-а је вредност параметра  $\alpha$  који се користи у овој модификацији у 2.4. Колона Време представља време у милисекундама које је било потребно за извршавање одређене модификације VNS-а. Колона означена са  $f(x)$  представља добијене вредности за функцију циља. У овом случају, за функцију циља је коришћена сума квадрата растојања тачака кластера од њиховог центроида која се минимизује. Наредних пет колона приказују вредности за различите индексе валидности кластерованја. Редом, то су *CS* – *CS* мера, *DB* – Дејвис-Болдин индекс, *CH* - Калински-Харабац индекс, *D* – Дан индекс и *S* – силуэта индекс. За прва два је пожељно да имају што мање вредности, док је за преостала три обрнуто. Најбоље постигнуте вредности за функцију циља и вредности индекса су истакнуте подебљаним фонтом. У последњој колони су просечне величине кластера од најмањег до највећег, раздвојене тачком и зарезом. За све модификације максимални број околина VNS-а је био 3 и коришћена је стратегија налажења најбољег суседа у датој околини. критеријум зауставља је 20 итерација без промене.

| Скуп података | VNS             | Време (ms) | $f(x)$               | CS            | DB            | CH               | D             | S             | Величине кластера                               |
|---------------|-----------------|------------|----------------------|---------------|---------------|------------------|---------------|---------------|-------------------------------------------------|
| Iris          | Редукована      | 1.5500     | 92.4180              | 0.8800        | 0.6356        | 480.3685         | 0.0773        | 0.5342        | [30.9; 49.2; 69.9]                              |
|               | Општа           | 1.6000     | 89.3775              | 0.8588        | 0.6132        | 495.3506         | 0.0746        | 0.5399        | [32.35; 49.6; 68.05]                            |
|               | Основна         | 4.7000     | <b>82.6981</b>       | <b>0.8417</b> | 0.6134        | <b>529.4474</b>  | <b>0.0896</b> | <b>0.5491</b> | [35.2; 50.55; 64.25]                            |
|               | Закошена - 0.01 | 4.4000     | 83.8998              | 0.8543        | <b>0.6079</b> | 524.6608         | 0.0879        | 0.5452        | [35.6; 50.4; 64]                                |
|               | Закошена - 0.1  | 4.3667     | 83.0418              | 0.8497        | 0.6102        | 527.5259         | 0.0883        | 0.5458        | [35.95; 50.73; 63.32]                           |
|               | Закошена - 0.9  | 4.5000     | 82.8652              | 0.8504        | 0.6157        | 528.9160         | 0.0843        | 0.5472        | [35.8; 50.67; 63.53]                            |
|               | Позната истина  | -          | 89.3868              | 0.9705        | 0.7517        | 483.0125         | 0.0585        | 0.5033        | [50; 50; 50]                                    |
| Wine          | Редукована      | 4.1500     | 2731603.5507         | 1.0236        | 0.5110        | 480.6000         | 0.0190        | 0.5471        | [41.3; 52.65; 84.05]                            |
|               | Општа           | 2.4500     | 2736429.3005         | 0.9948        | 0.5306        | 480.5402         | 0.0218        | 0.5483        | [35.65; 50.35; 92]                              |
|               | Основна         | 5.7500     | <b>2480862.6939</b>  | 1.0061        | <b>0.4973</b> | <b>532.5609</b>  | 0.0199        | <b>0.5634</b> | [42.95; 57.3; 77.75]                            |
|               | Закошена - 0.01 | 6.2333     | 2641585.3587         | <b>0.9833</b> | 0.5064        | 498.5690         | <b>0.0231</b> | 0.5612        | [37.45; 51.23; 89.32]                           |
|               | Закошена - 0.1  | 6.5000     | 2535214.1407         | 1.0018        | 0.5103        | 520.6191         | 0.0191        | 0.5588        | [41.98; 55.13; 80.88]                           |
|               | Закошена - 0.9  | 6.6667     | 2548324.1771         | 0.9976        | 0.5002        | 517.5916         | 0.0197        | 0.5567        | [40.85; 53.75; 83.4]                            |
|               | Позната истина  | -          | 5232632.3662         | 2.1723        | 1.4993        | 205.4971         | 0.0048        | 0.2143        | [48; 59; 71]                                    |
| Glass         | Редукована      | 0.0000     | 402.7893             | 1.0993        | 0.9080        | 96.0402          | 0.0604        | 0.4085        | [2.25; 7.75; 14.9; 23.75; 40.05; 125.3]         |
|               | Општа           | 5.9000     | 390.2231             | <b>1.0363</b> | <b>0.8306</b> | 100.6016         | <b>0.0638</b> | 0.3778        | [2.3; 4.55; 12.45; 23.45; 37.55; 133.7]         |
|               | Основна         | 59.6500    | 368.6762             | 1.1426        | 0.8611        | 108.6963         | 0.0626        | 0.4151        | [2.75; 6.45; 14.3; 24.85; 36.2; 129.45]         |
|               | Закошена - 0.01 | 65.5500    | 364.8570             | 1.0982        | 0.8745        | 110.1824         | 0.0597        | <b>0.4226</b> | [2.45; 6.33; 15.02; 25.63; 38.35; 126.22]       |
|               | Закошена - 0.1  | 61.7667    | 366.9918             | 1.1250        | 0.8698        | 109.4209         | 0.0613        | 0.3974        | [2.68; 6.45; 14.48; 24.78; 37.48; 128.12]       |
|               | Закошена - 0.9  | 60.4000    | <b>364.1985</b>      | 1.1307        | 0.8903        | <b>110.5166</b>  | 0.0547        | 0.4154        | [2.53; 7.65; 16.32; 26.47; 39.52; 121.52]       |
|               | Позната истина  | -          | 911.2041             | 4.2955        | 3.7363        | 19.4179          | 0.0154        | -0.0267       | [9; 13; 17; 29; 70; 76]                         |
| Vowel         | Редукована      | 19.0500    | 36914271.9172        | 1.5565        | 1.0014        | 1195.8195        | 0.0397        | 0.3318        | [67.9; 99.85; 123.5; 150.75; 178.05; 250.95]    |
|               | Општа           | 18.6500    | 38051898.4729        | <b>1.4611</b> | 0.9676        | 1158.9163        | 0.0430        | 0.3453        | [52.4; 101.7; 129.35; 148.55; 183.5; 255.5]     |
|               | Основна         | 220.2500   | <b>35092823.1627</b> | 1.4903        | 0.9665        | <b>1263.1337</b> | <b>0.0518</b> | 0.3436        | [73.75; 109.7; 133.7; 153.4; 183; 217.45]       |
|               | Закошена - 0.01 | 227.6167   | 35304362.3989        | 1.4814        | 0.9571        | 1254.7037        | 0.0492        | <b>0.3459</b> | [71.27; 110.5; 133.68; 154.3; 180.67; 220.58]   |
|               | Закошена - 0.1  | 224.3333   | 35178682.0405        | 1.4838        | <b>0.9569</b> | 1260.0743        | 0.0469        | 0.3454        | [72.73; 111.73; 133.78; 153.67; 178.53; 220.55] |
|               | Закошена - 0.9  | 214.7833   | 35534705.5644        | 1.4960        | 0.9675        | 1248.2418        | 0.0483        | 0.3413        | [75.73; 109.7; 130.87; 149.53; 181.3; 223.87]   |
|               | Позната истина  | -          | 65983577.6043        | 3.0111        | 1.6444        | 589.1757         | 0.0000        | 0.1237        | [72; 89; 151; 172; 180; 207]                    |

Табела 6.1 - Резултати модификација VNS-а на стварним скуповима података



За 3 од 4 тестирана скупа података, основна варијанта VNS-a је дошла до најмање вредности функције циља, док се за један скуп података најбоње показала закошена варијанта VNS-a са параметром  $\alpha = 0.9$ . Међутим, у свим случајевима се примећује да су вредности функција циља добијене овим два метода прилично блиске.

Ако упоредимо добијене резултате са резултатима из рада [15], примећујемо да су за Дан и Дејвис-Болдин индексе на скупу података *Iris* VNS-ом постигнути бољи резултати. Такође, на скупу података *Wine* постигнуте су боље вредности за индексе Дејвис-Болдин и Силуета, док су за индекс Дан добијени нешто слабији резултати. Вредности добијене за Дан индекс за скуп података *Glassco* значајно боље од оних добијених у раду [15] коришћењем алгоритама *k-means* и *SOM*<sup>2</sup>. У поређењу са резултатима различитих алгоритама из [16] углавном су све вредности добијене модификацијама VNS-a за скуп података *Vowel* лошије. Упоређивањем са још неким доступним резултатима, примећено је да различити алгоритми често дају лоше резултате за овај скуп података. Главни разлог за то је што се кластери скупа података *Vowel* у огромној мери преклапају [17] што максимално отежава адекватно кластеровање.

Генерално гледано основна варијанта VNS-a је у највећем броју случајева дала најбоље вредности за индексе валидности кластеровања, али ни у ком случају се величине стварних кластера и кластера добијеним коришћењем VNS-a не поклапају. То нипошто не говори о квалитету постигнутог кластеровања, јер је циљ кластеровања била минимизација суме растојања тачака од њиховог центроида. Све модификације VNS-a са неком методом локалног претраживања постигле су без изузетка боље, па чак и далеко боље, вредности за функцију циља од онога што је позната истина.

У табели 6.2 су приказани резултати за три различите функције циља.

Функција циља 1 је сума квадрата растојања тачака  $x_i$  од центроида њиховог кластера  $G_k$  (видети 4.2):

$$f(x) = \min \sum_{k=1}^m \sum_{x_i \in C_k} \|x_i - G_k\|^2.$$

---

<sup>2</sup>Енг. Self-Organizing Map

| Функција циља         | Време (ms) | Позиви мера | $f(x)$        | K1   | K2    | K3    |       |       |       |
|-----------------------|------------|-------------|---------------|------|-------|-------|-------|-------|-------|
| <b>Iris</b>           |            |             |               |      |       |       |       |       |       |
| <b>Основни VNS</b>    |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 1.8953     | 16302684.0  | 103.3001      | 31.7 | 48.6  | 69.8  |       |       |       |
| Функција циља 2       | 1.8480     | 16318231.5  | 11.0449       | 28.3 | 48.6  | 73.1  |       |       |       |
| Функција циља 3       | 7.5773     | 8754284.4   | 2.6519        | 15.5 | 47.2  | 87.2  |       |       |       |
| <b>Општи VNS</b>      |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 1.1407     | 15914110.2  | 104.6740      | 30.8 | 48.4  | 70.8  |       |       |       |
| Функција циља 2       | 1.1720     | 15924588.1  | 10.9307       | 28.4 | 48.5  | 73.1  |       |       |       |
| Функција циља 3       | 4.6093     | 8570451.4   | 2.6782        | 17.2 | 46.5  | 86.2  |       |       |       |
| <b>Редуковани VNS</b> |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 1.0167     | 15680610.2  | 106.2576      | 30.8 | 48.3  | 70.8  |       |       |       |
| Функција циља 2       | 0.9300     | 15686554.2  | 11.0236       | 28.7 | 48.5  | 72.8  |       |       |       |
| Функција циља 3       | 2.8580     | 8460842.9   | 2.6789        | 16.8 | 47.4  | 85.8  |       |       |       |
| <b>Wine</b>           |            |             |               |      |       |       |       |       |       |
| <b>Основни VNS</b>    |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 2.8760     | 7701311.0   | 2929365.5887  | 37.3 | 51.8  | 88.8  |       |       |       |
| Функција циља 2       | 2.6700     | 7716528.7   | 2020.2145     | 29.6 | 54.0  | 94.4  |       |       |       |
| Функција циља 3       | 16.9740    | 4205089.3   | 401.1177      | 26.6 | 49.4  | 102.0 |       |       |       |
| <b>Општи VNS</b>      |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 1.7487     | 7324191.4   | 2995473.3221  | 36.3 | 51.3  | 90.4  |       |       |       |
| Функција циља 2       | 1.6413     | 7334262.8   | 1995.6659     | 29.2 | 53.8  | 95.0  |       |       |       |
| Функција циља 3       | 10.5067    | 4029068.3   | 402.9311      | 26.6 | 49.7  | 101.7 |       |       |       |
| <b>Редуковани VNS</b> |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 1.3420     | 7099605.6   | 2995057.1496  | 36.6 | 51.3  | 90.1  |       |       |       |
| Функција циља 2       | 1.2720     | 7105351.1   | 2046.3696     | 29.5 | 54.0  | 94.5  |       |       |       |
| Функција циља 3       | 6.1447     | 3924841.0   | 401.9740      | 25.5 | 49.4  | 103.1 |       |       |       |
| <b>Glass</b>          |            |             |               |      |       |       |       |       |       |
| <b>Основни VNS</b>    |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 34.8153    | 4293062.8   | 398.7444      | 2.7  | 7.2   | 16.1  | 27.0  | 40.9  | 120.1 |
| Функција циља 2       | 34.2687    | 4415456.8   | 4.7894        | 1.7  | 3.7   | 10.2  | 23.7  | 39.4  | 135.3 |
| Функција циља 3       | 183.8480   | 2395090.1   | 8.7627        | 2.2  | 6.0   | 15.7  | 30.5  | 50.9  | 108.8 |
| <b>Општи VNS</b>      |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 28.7733    | 931576.1    | 406.1283      | 2.8  | 7.3   | 16.6  | 27.6  | 41.7  | 118.0 |
| Функција циља 2       | 28.2433    | 1035736.5   | 5.0408        | 1.8  | 4.2   | 11.1  | 24.7  | 40.7  | 131.5 |
| Функција циља 3       | 155.5447   | 526081.6    | 9.0737        | 2.3  | 6.7   | 16.9  | 31.2  | 50.8  | 106.0 |
| <b>Редуковани VNS</b> |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 2.4513     | 90429.1     | 422.9239      | 2.5  | 7.3   | 17.0  | 28.2  | 42.6  | 116.4 |
| Функција циља 2       | 2.3900     | 98538.8     | 5.3312        | 1.9  | 4.6   | 11.2  | 23.9  | 42.2  | 130.3 |
| Функција циља 3       | 8.7300     | 44596.3     | 9.4021        | 2.5  | 7.9   | 19.3  | 32.9  | 50.7  | 100.8 |
| <b>Vowel</b>          |            |             |               |      |       |       |       |       |       |
| <b>Основни VNS</b>    |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 116.3760   | 12598567.4  | 37506722.0512 | 61.8 | 102.0 | 128.1 | 153.5 | 185.1 | 240.6 |
| Функција циља 2       | 111.6273   | 12733293.3  | 3148.3469     | 29.2 | 86.5  | 129.5 | 163.9 | 201.6 | 260.3 |
| Функција циља 3       | 1498.5320  | 6840678.4   | 1288.9021     | 8.6  | 28.0  | 64.2  | 131.8 | 247.6 | 390.7 |
| <b>Општи VNS</b>      |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 97.8253    | 8957705.4   | 37917723.7642 | 61.0 | 101.4 | 128.2 | 154.1 | 184.2 | 242.2 |
| Функција циља 2       | 91.7593    | 9070840.8   | 3207.5884     | 30.3 | 86.0  | 127.8 | 164.3 | 200.6 | 262.1 |
| Функција циља 3       | 1278.8580  | 4880734.5   | 1320.1138     | 10.8 | 33.0  | 70.4  | 133.2 | 241.2 | 382.4 |
| <b>Редуковани VNS</b> |            |             |               |      |       |       |       |       |       |
| Функција циља 1       | 8.8793     | 8050098.1   | 39015692.5530 | 55.6 | 97.9  | 125.7 | 153.2 | 187.4 | 251.4 |
| Функција циља 2       | 7.9660     | 8058264.8   | 3338.4036     | 28.5 | 81.6  | 124.4 | 162.1 | 200.7 | 273.7 |
| Функција циља 3       | 74.9700    | 4367471.4   | 1369.9122     | 13.2 | 40.2  | 79.8  | 141.3 | 234.3 | 362.1 |

Табела 6.2 - Резултати модификација VNS-а за различите функција циља

| Услов заустављања        | Итерације   | Време           | Позиви мере       | $f(x)$               |
|--------------------------|-------------|-----------------|-------------------|----------------------|
| <b>Glass</b>             |             |                 |                   |                      |
| <b>Основни VNS</b>       |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>38.9000</b>  | <b>5014296.0</b>  | 370.2213             |
| Мало побољшање           | <b>32.9</b> | 59.3333         | 5124015.6         | <b>366.2959</b>      |
| Без побољшања            | 34.3        | 59.6500         | 5047232.5         | 368.6762             |
| <b>Општи VNS</b>         |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>61.0333</b>  | <b>1885510.1</b>  | 370.2098             |
| Мало побољшање           | <b>33.0</b> | 94.5889         | 2068611.4         | 366.6301             |
| Без побољшања            | 37.6        | 101.4833        | 1939703.2         | <b>365.2649</b>      |
| <b>Редуковани VNS</b>    |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>3.9500</b>   | <b>147068.5</b>   | 393.0299             |
| Мало побољшање           | <b>35.5</b> | 4.9167          | 154359.1          | <b>389.1192</b>      |
| Без побољшања            | 35.9        | 0.0000          | 149313.2          | 402.7893             |
| <b>Iris</b>              |             |                 |                   |                      |
| <b>Основни VNS</b>       |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>2.3500</b>   | <b>16423543.6</b> | 83.4747              |
| Мало побољшање           | 23.9        | 4.1333          | 16443067.9        | 83.5522              |
| Без побољшања            | <b>22.6</b> | 4.7000          | 16429689.8        | <b>82.6981</b>       |
| <b>Општи VNS</b>         |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>1.0000</b>   | <b>16048276.9</b> | 100.6143             |
| Мало побољшање           | <b>25.9</b> | 1.6333          | 16056896.8        | 94.3685              |
| Без побољшања            | 27.3        | 1.5500          | 16050973.7        | <b>91.5466</b>       |
| <b>Редуковани VNS</b>    |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>0.8000</b>   | <b>15723124.9</b> | <b>91.0465</b>       |
| Мало побољшање           | <b>26.5</b> | 1.5667          | 15728314.4        | 100.5218             |
| Без побољшања            | 26.7        | 1.5500          | 15724776.9        | 92.4180              |
| <b>Vowel</b>             |             |                 |                   |                      |
| <b>Основни VNS</b>       |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>115.6500</b> | <b>13384985.7</b> | 35347846.6335        |
| Мало побољшање           | <b>32.5</b> | 177.0833        | 13496065.8        | 35413844.4029        |
| Без побољшања            | 40.7        | 220.2500        | 13420130.4        | <b>35092823.1627</b> |
| <b>Општи VNS</b>         |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>200.2833</b> | <b>9985688.7</b>  | 35756369.4680        |
| Мало побољшање           | <b>34.2</b> | 316.8667        | 10188523.5        | 35500004.2615        |
| Без побољшања            | 36.5        | 322.1167        | 10044604.1        | <b>35025163.0554</b> |
| <b>Редуковани VNS</b>    |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>7.8000</b>   | <b>8106208.0</b>  | 37296802.1929        |
| Мало побољшање           | <b>39.1</b> | 15.2833         | 8114088.9         | 37876357.7285        |
| Без побољшања            | 49.6        | 19.0500         | 8108607.2         | <b>36914271.9172</b> |
| <b>Wine</b>              |             |                 |                   |                      |
| <b>Основни VNS</b>       |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>5.1000</b>   | <b>7819668.3</b>  | <b>2469245.1269</b>  |
| Мало побољшање           | 24.5        | 6.7167          | 7839318.1         | 2522227.6627         |
| Без побољшања            | <b>22.3</b> | 5.7500          | 7825684.1         | 2480862.6939         |
| <b>Општи VNS</b>         |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>1.9667</b>   | <b>7453343.0</b>  | 2833466.4623         |
| Мало побољшање           | <b>25.5</b> | 2.4778          | 7461639.3         | <b>2817859.0137</b>  |
| Без побољшања            | 25.7        | 2.4333          | 7455987.9         | 2844339.0838         |
| <b>Редуковани VNS</b>    |             |                 |                   |                      |
| Ограничен број итерација | 20.0        | <b>1.6500</b>   | <b>7139956.0</b>  | <b>2680603.3110</b>  |
| Мало побољшање           | <b>24.7</b> | 1.6833          | 7144946.0         | 2795641.6814         |
| Без побољшања            | 25.6        | 4.1500          | 7141596.2         | 2731603.5507         |

Табела 6.3 - Резултати модификација VNS-а за различите критеријуме заустављања

| Гранична вредност     | Итерације   | Време (ms)      | Позиви мере       | f(x)                 |
|-----------------------|-------------|-----------------|-------------------|----------------------|
| <b>Glass</b>          |             |                 |                   |                      |
| <b>Основни VNS</b>    |             |                 |                   |                      |
| 0.0001                | 38.4        | 68.2500         | 5164026.1         | <b>361.4863</b>      |
| 0.001                 | <b>29.0</b> | <b>53.0000</b>  | 5121735.6         | 369.0328             |
| 0.01                  | 31.2        | 56.7500         | <b>5086285.2</b>  | 368.3686             |
| <b>Општи VNS</b>      |             |                 |                   |                      |
| 0.0001                | 33.9        | 97.6333         | 2135630.1         | <b>362.2487</b>      |
| 0.001                 | 35.0        | 103.5333        | 2065862.0         | 366.0825             |
| 0.01                  | <b>30.1</b> | <b>82.6000</b>  | <b>2004342.2</b>  | 371.5591             |
| <b>Редуковани VNS</b> |             |                 |                   |                      |
| 0.0001                | 36.5        | <b>3.9000</b>   | 157049.0          | <b>385.4371</b>      |
| 0.001                 | 39.0        | 5.4500          | 154331.9          | 393.9237             |
| 0.01                  | <b>30.9</b> | 5.4000          | <b>151696.5</b>   | 387.9967             |
| <b>Iris</b>           |             |                 |                   |                      |
| <b>Основни VNS</b>    |             |                 |                   |                      |
| 0.0001                | <b>22.8</b> | 3.9000          | 16449794.9        | <b>81.1549</b>       |
| 0.001                 | 24.4        | 4.6500          | 16443151.5        | 86.0673              |
| 0.01                  | 24.6        | <b>3.8500</b>   | <b>16436257.2</b> | 83.4345              |
| <b>Општи VNS</b>      |             |                 |                   |                      |
| 0.0001                | 26.0        | 1.5333          | 16059857.3        | 96.7385              |
| 0.001                 | 26.4        | 2.0833          | 16056881.5        | 94.1094              |
| 0.01                  | <b>25.2</b> | <b>1.2833</b>   | <b>16053951.7</b> | <b>92.2575</b>       |
| <b>Редуковани VNS</b> |             |                 |                   |                      |
| 0.0001                | 29.4        | <b>1.5500</b>   | 15730112.7        | <b>85.9065</b>       |
| 0.001                 | 25.4        | 1.6000          | 15728284.5        | 98.8121              |
| 0.01                  | <b>24.7</b> | <b>1.5500</b>   | <b>15726546.1</b> | 116.8467             |
| <b>Vowel</b>          |             |                 |                   |                      |
| <b>Основни VNS</b>    |             |                 |                   |                      |
| 0.0001                | 33.8        | 184.3000        | 13532953.1        | <b>35015780.2573</b> |
| 0.001                 | 34.8        | 191.5000        | 13496147.6        | 35614202.7155        |
| 0.01                  | <b>28.8</b> | <b>155.4500</b> | <b>13459096.8</b> | 35611550.2359        |
| <b>Општи VNS</b>      |             |                 |                   |                      |
| 0.0001                | 36.2        | 324.7833        | 10264242.9        | 35601606.8927        |
| 0.001                 | 35.0        | 335.6667        | 10186847.3        | 35791007.9598        |
| 0.01                  | <b>31.3</b> | <b>290.1500</b> | <b>10114480.2</b> | <b>35107397.9319</b> |
| <b>Редуковани VNS</b> |             |                 |                   |                      |
| 0.0001                | 48.6        | 19.0000         | 8116939.8         | 37652412.8135        |
| 0.001                 | 37.2        | 14.4000         | 8113825.3         | <b>37282739.6946</b> |
| 0.01                  | <b>31.6</b> | <b>12.4500</b>  | <b>8111501.5</b>  | 38693920.6773        |
| <b>Wine</b>           |             |                 |                   |                      |
| <b>Основни VNS</b>    |             |                 |                   |                      |
| 0.0001                | 25.0        | 6.8500          | 7846245.5         | <b>2480461.6625</b>  |
| 0.001                 | 24.9        | 7.0500          | 7839278.6         | 2539837.8261         |
| 0.01                  | <b>23.7</b> | <b>6.2500</b>   | <b>7832430.3</b>  | 2546383.4994         |
| <b>Општи VNS</b>      |             |                 |                   |                      |
| 0.0001                | 25.8        | 2.4167          | 7464542.8         | 2775288.6472         |
| 0.001                 | 26.7        | 2.5833          | 7461634.8         | <b>2726421.6123</b>  |
| 0.01                  | <b>24.0</b> | <b>2.4333</b>   | <b>7458740.5</b>  | 2951866.7817         |
| <b>Редуковани VNS</b> |             |                 |                   |                      |
| 0.0001                | 24.8        | <b>1.6000</b>   | 7179653.7         | 2799019.7093         |
| 0.001                 | 26.2        | 1.6000          | 7177792.6         | <b>2739844.2356</b>  |
| 0.01                  | <b>23.1</b> | 1.8500          | <b>7175956.6</b>  | 2904010.5828         |

Табела 6.4 - Резултати модификација VNS-а за критеријум заустављања са варирањем очекиваног побољшања у 20 узастопних итерација

Функција циља 2 је функција циља 1 подељена сумом свих међусобних растојања центроида [14]:

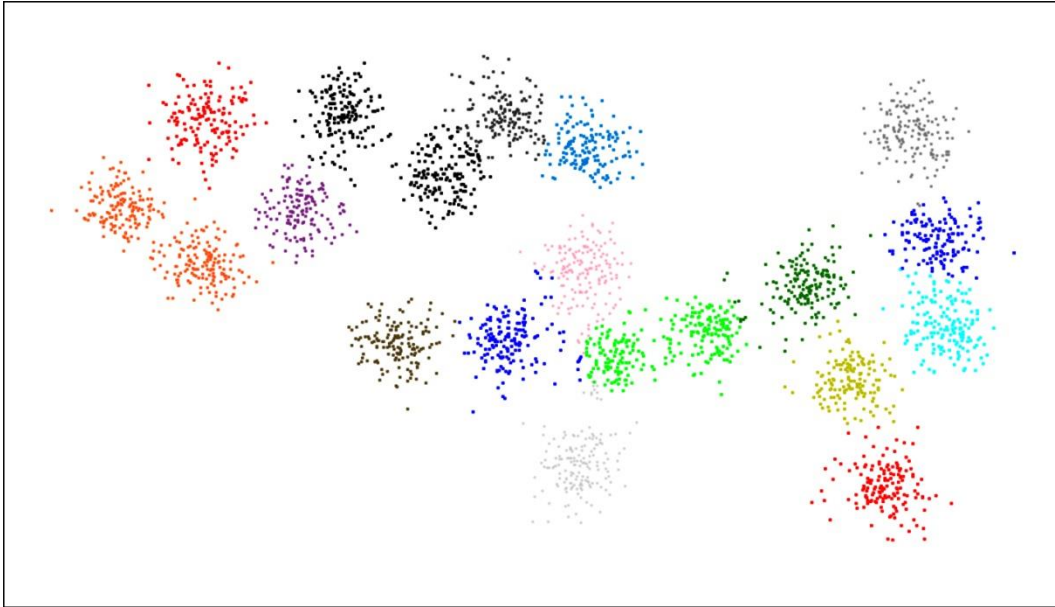
$$f(x) = \min \frac{\sum_{i=1}^m \sum_{x_l \in C_i} \|x_l - G_i\|^2}{\sum_{i < j} \|G_i - G_j\|}.$$

Функција циља 3 је сума просечних растојања парова тачака унутар кластера (видети 4.5) [14].

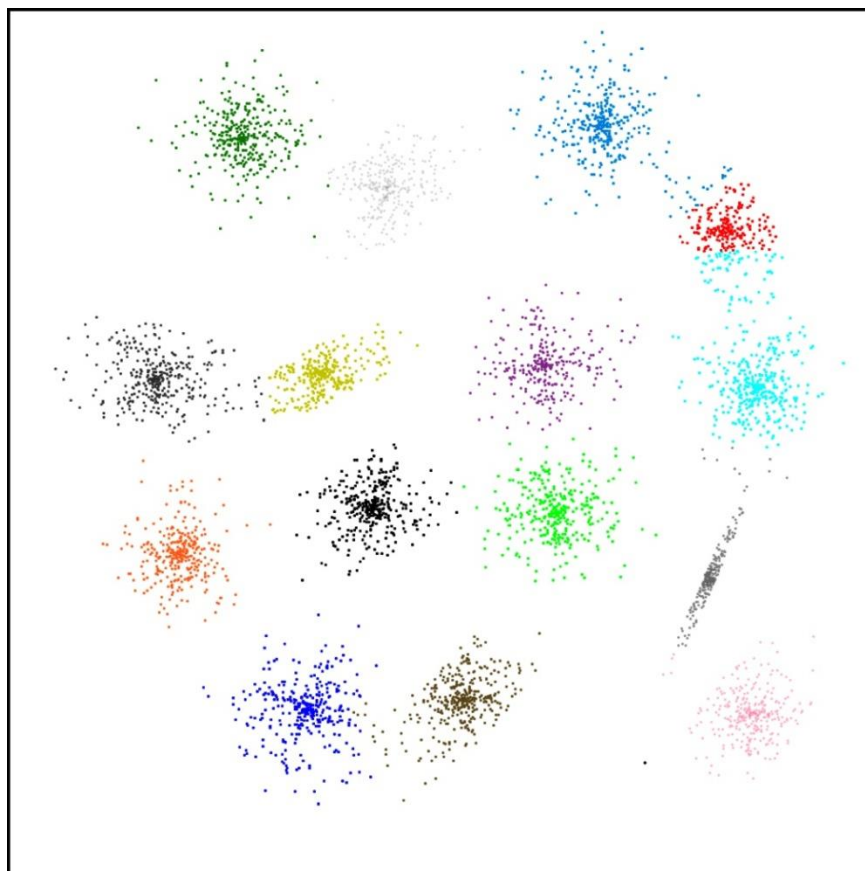
$$f(x) = \min \sum_{i=1}^m \frac{1}{n_i(n_i - 1)} \sum_{\substack{x_l \in C_i \\ x_{l'} \in C_i \\ l < l'}} \|x_l - x_{l'}\|^2.$$

У табели 6.3 су упоређени резултати које дају неке модификације VNS-а за различите критеријуме заустављања. Код првог критеријума заустављања број итерација је био ограничен на 20, као и у трећем где је тај број представљао број итерација без промене. Критеријум заустављања "мало побољшање" мери да ли је побољшање у последњих неколико, у овом случају 20, итерација дошло до захтеваног побољшања. Ако није било очекиваног побољшања, алгоритам стаје. Понашање модификација приликом варирања овог захтеваног побољшања у 20 итерација илустровано је резултатима у табели 6.4. Величине 0.01, 0.001 и 0.0001 говоре за колико процената најмање се морају разликовати посматрано решење и решење добијено 20 итерација раније: 0.01=1%, 0.001=0.1% и 0.0001=0.01%.

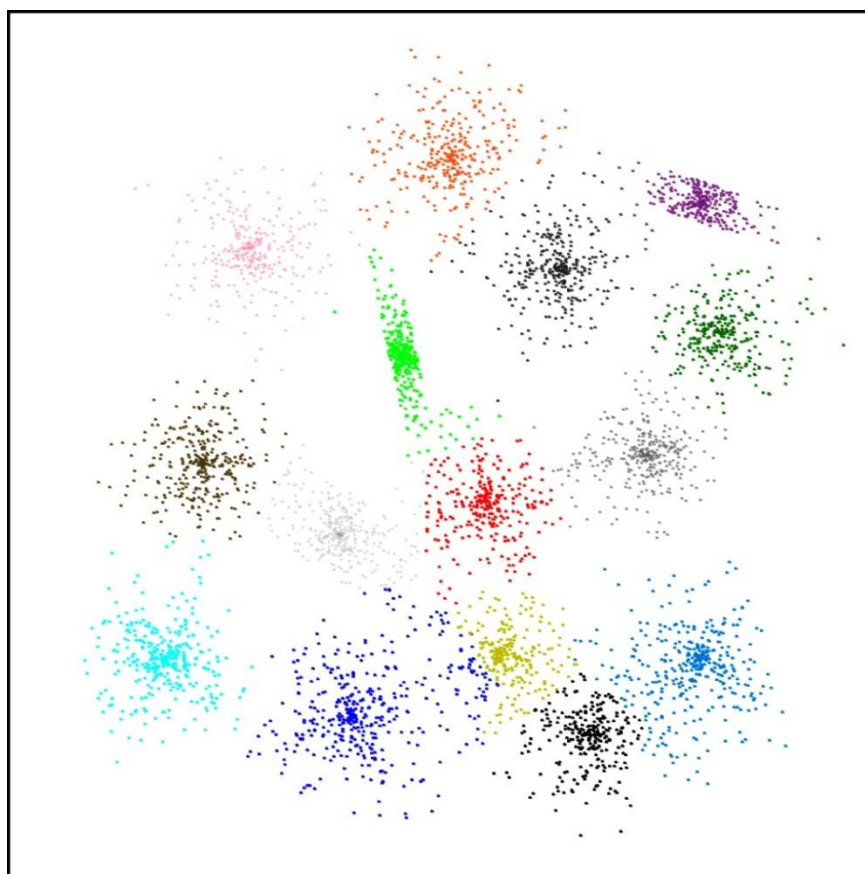
С друге стране, стварни кластери ових скупова података се преклапају, што их чини непогодним за примену ове методе кластеровања са поменутом функцијом циља. Стога су изабрана још 3 синтетичка скупа података и на њима тестиране основна и општа модификација VNS-а. Као што је већ поменуто, тестирани A- и S-скупови података су преузети са сајта <https://cs.joensuu.fi/sipu/datasets/> [22], [23], [24] а тачна имена инстанци које су коришћене су A1, S1 и S2. За кластеровање сва три скупа коришћене су основна и општа модификација VNS-а, а резултати су приказани следећим сликама и табелама.



Слика 6.1 - Резултат GVNS-а за A1 скуп података



Слика 6.3 - Резултат GVNS-а за  $S1$  скуп података



Слика 6.2 - Резултат GVNS-а за  $S2$  скуп података

| Скуп података | Време (ms ) | f(x)              | CS     | CH         | DB     | D      | S      |
|---------------|-------------|-------------------|--------|------------|--------|--------|--------|
| <b>A1</b>     |             |                   |        |            |        |        |        |
| Основни VNS   | 7225.9      | 17342080648.70    | 0.9543 | 9713.0881  | 0.6126 | 0.0069 | 0.4897 |
| Општи VNS     | 21345.6     | 16498728746.76    | 0.9438 | 10241.6310 | 0.6000 | 0.0083 | 0.5367 |
| <b>S1</b>     |             |                   |        |            |        |        |        |
| Основни VNS   | 4683.0      | 15290744711693.00 | 0.9282 | 13635.1379 | 0.4839 | 0.0043 | 0.6289 |
| Општи VNS     | 17475.5     | 13673791017264.90 | 0.8879 | 15323.5093 | 0.4576 | 0.0041 | 0.6460 |
| <b>S2</b>     |             |                   |        |            |        |        |        |
| Основни VNS   | 5838.9      | 20346279529989.90 | 1.0407 | 8804.2882  | 0.5872 | 0.0044 | 0.5447 |
| Општи VNS     | 24130.9     | 19014376006499.70 | 1.0223 | 9488.5461  | 0.5717 | 0.0055 | 0.5569 |

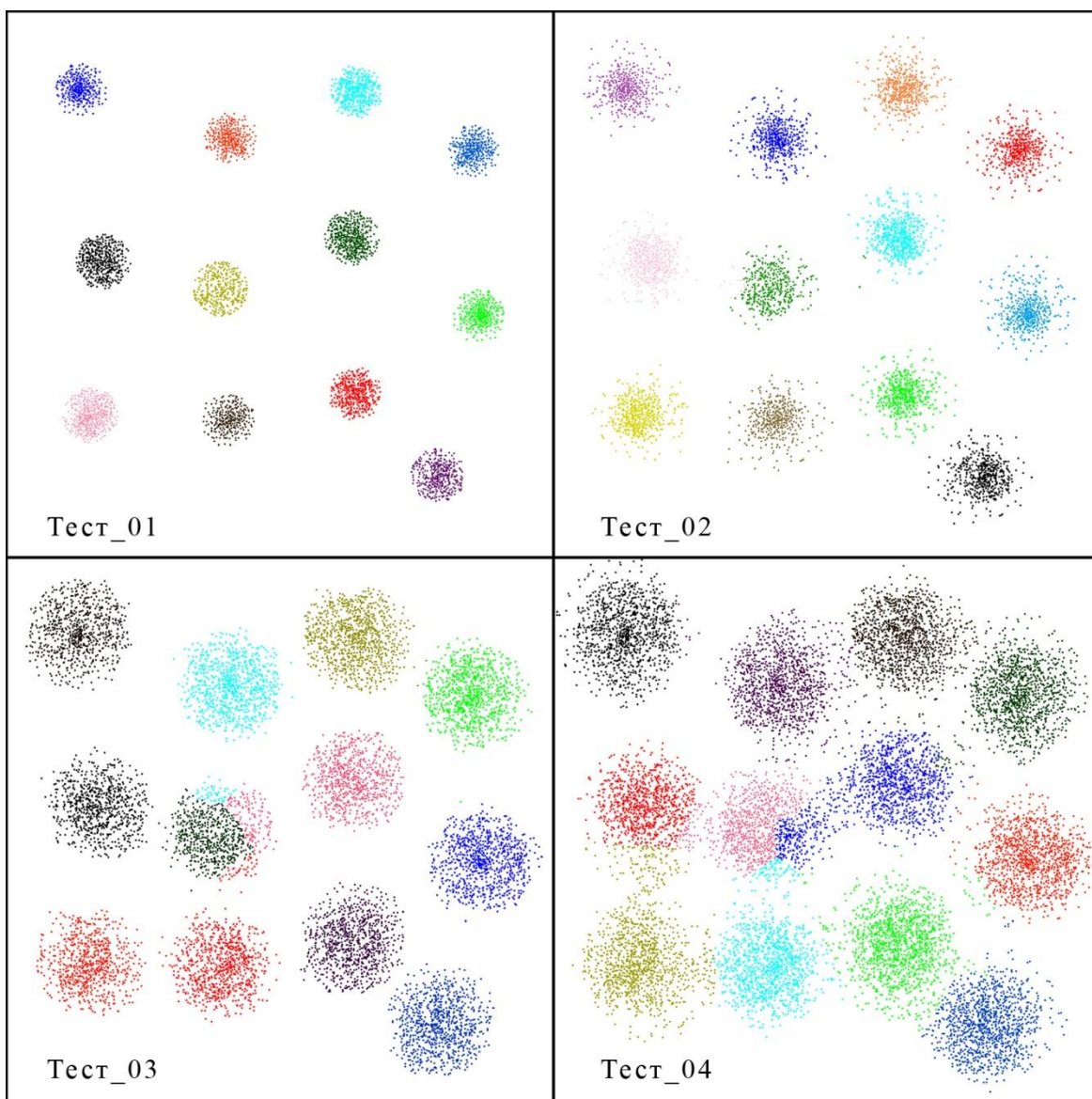
Табела 6.6 - Резултати модификација VNS-а за скупове података A1, S1 и S2

| Скуп података | K1    | K2    | K3    | K4    | K5    | K6    | K7    | K8    | K9    | K10   | K11   | K12   | K13   | K14   | K15   | K16   | K17   | K18   | K19   | K20   |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>A1</b>     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Основни VNS   | 44.4  | 80.2  | 103.2 | 119.2 | 132.3 | 140.1 | 145.1 | 147.8 | 149.5 | 150.7 | 152.8 | 155.4 | 158.2 | 160.8 | 165.2 | 171.7 | 179.5 | 190.4 | 209.6 | 244.2 |
| Општи VNS     | 50.5  | 83.5  | 113.1 | 128.2 | 136.7 | 141.8 | 145.1 | 147.5 | 149.2 | 151.0 | 152.3 | 154.4 | 156.7 | 159.7 | 163.6 | 167.9 | 175.4 | 185.2 | 205.2 | 233.3 |
| <b>S1</b>     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Основни VNS   | 102.5 | 230.6 | 282.2 | 306.5 | 317.5 | 326.3 | 333.3 | 338.7 | 343.4 | 348.9 | 354.2 | 362.3 | 379.4 | 422.6 | 551.9 |       |       |       |       |       |
| Општи VNS     | 136.6 | 250.2 | 289.6 | 308.5 | 319.6 | 328.4 | 333.7 | 338.0 | 342.3 | 347.9 | 352.3 | 359.1 | 373.4 | 407.9 | 512.6 |       |       |       |       |       |
| <b>S2</b>     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Основни VNS   | 67.5  | 194.1 | 265.8 | 292.6 | 312.3 | 324.3 | 332.6 | 340.7 | 350.2 | 358.0 | 370.3 | 384.1 | 407.7 | 452.3 | 547.5 |       |       |       |       |       |
| Општи VNS     | 79.1  | 218.0 | 273.9 | 303.0 | 317.5 | 327.1 | 335.5 | 342.8 | 349.1 | 355.0 | 362.9 | 372.7 | 391.0 | 431.0 | 541.6 |       |       |       |       |       |

Табела 6.5 - Величине кластера добијене модификацијама VNS-а за скупове података A1, S1 и S2



Поред приказаних јавно доступних тест инстанци за кластеровање, за потребе овог рада направљена су још 4 скупа података који илуструју везу између квалитета кластеровања и јасне раздвојености кластера. Свака наредна инстанца има више тачака и већи шум, што отежава кластеровање, захтева више времена за извршавање алгоритма и даје слабије резултате у смислу постигнутих вредности за индексе валидности кластеровања.



Слика 6.4 - Резултат GVNS-а за генерисане скупове података

|                    | Време (ms)       | Итерације   | Позиви мере      | CH                | DB            | S             |
|--------------------|------------------|-------------|------------------|-------------------|---------------|---------------|
| <b>Тест_01</b>     |                  |             |                  |                   |               |               |
| <b>Основни VNS</b> |                  |             |                  |                   |               |               |
| Најбољи сусед      |                  |             |                  |                   |               |               |
| VNS околина 2      | <b>1276.1000</b> | <b>29.0</b> | <b>1758419.8</b> | 44041.7365        | 0.2481        | 0.8239        |
| VNS околина 3      | 7041.3000        | 37.1        | 1853968.8        | <b>45791.3319</b> | <b>0.2425</b> | <b>0.8265</b> |
| Први бољи сусед    |                  |             |                  |                   |               |               |
| VNS околина 2      | 322.4000         | <b>30.4</b> | 1738699.5        | 44145.4142        | 0.2573        | 0.8236        |
| VNS околина 3      | 787.5000         | 27.0        | 1779729.4        | <b>47461.6098</b> | <b>0.2469</b> | <b>0.8291</b> |
| <b>Општи VNS</b>   |                  |             |                  |                   |               |               |
| VND                |                  |             |                  |                   |               |               |
| VNS околина 2      |                  |             |                  |                   |               |               |
| VND околина 1      | <b>65.4000</b>   | <b>31.0</b> | <b>807.4</b>     | 39012.3089        | 0.3121        | 0.7997        |
| VND околина 2      | 1841.5000        | 36.4        | 26264.3          | 47399.1332        | 0.2570        | 0.8238        |
| VND околина 3      | 3110.4000        | 37.0        | 81990.7          | <b>53394.2532</b> | <b>0.2286</b> | <b>0.8368</b> |
| VND околина 10     | 15245.4000       | 35.1        | 314404.2         | 53248.0655        | 0.2305        | 0.8367        |
| VNS околина 3      |                  |             |                  |                   |               |               |
| VND околина 1      | <b>89.4000</b>   | 35.5        | <b>446653.4</b>  | 37766.1666        | 0.3038        | 0.8038        |
| VND околина 2      | 3959.2000        | <b>32.4</b> | 493244.1         | 54638.1602        | 0.2277        | 0.8384        |
| VND околина 3      | 10076.3000       | 42.8        | 643844.1         | 54782.4507        | 0.2290        | <b>0.8385</b> |
| VND околина 10     | 54443.5000       | 35.6        | 1233807.3        | <b>54877.8277</b> | <b>0.2261</b> | <b>0.8385</b> |
| <b>Тест_02</b>     |                  |             |                  |                   |               |               |
| <b>Основни VNS</b> |                  |             |                  |                   |               |               |
| Најбољи сусед      |                  |             |                  |                   |               |               |
| VNS околина 2      | <b>1855.1000</b> | 34.9        | <b>3554822.2</b> | 21755.5946        | 0.3757        | 0.7218        |
| VNS околина 3      | 7697.7000        | <b>33.5</b> | 3651142.1        | <b>23693.5457</b> | <b>0.3637</b> | <b>0.7299</b> |
| Први бољи сусед    |                  |             |                  |                   |               |               |
| VNS околина 2      | <b>371.3000</b>  | <b>34.7</b> | <b>3532071.0</b> | 16315.2659        | 0.4590        | 0.6761        |
| VNS околина 3      | 1287.6000        | 40.3        | 3580084.7        | <b>19120.6923</b> | <b>0.4217</b> | <b>0.7074</b> |
| <b>Општи VNS</b>   |                  |             |                  |                   |               |               |
| VND                |                  |             |                  |                   |               |               |
| VNS околина 2      |                  |             |                  |                   |               |               |
| VND околина 1      | <b>108.0000</b>  | 56.6        | <b>1913349.4</b> | 14038.7949        | 0.4991        | 0.5081        |
| VND околина 2      | 2030.9000        | 29.9        | 1936213.1        | 21897.3374        | 0.3996        | 0.7163        |
| VND околина 3      | 2978.9000        | 29.2        | 1987556.3        | 24027.9636        | 0.3593        | 0.7332        |
| VND околина 10     | 15116.9000       | <b>25.5</b> | 2157497.1        | <b>24623.1334</b> | <b>0.3569</b> | <b>0.7349</b> |
| VNS околина 3      |                  |             |                  |                   |               |               |
| VND околина 1      | <b>117.6000</b>  | 38.6        | <b>2276935.8</b> | 15878.0503        | 0.4763        | 0.6740        |
| VND околина 2      | 5104.0000        | 33.3        | 2324620.5        | 24746.7995        | 0.3594        | 0.7357        |
| VND околина 3      | 9787.2000        | 35.1        | 2455676.7        | 24652.7507        | 0.3547        | 0.7360        |
| VND околина 10     | 66626.9000       | <b>32.0</b> | 3093590.7        | <b>25916.3513</b> | <b>0.3420</b> | <b>0.7414</b> |

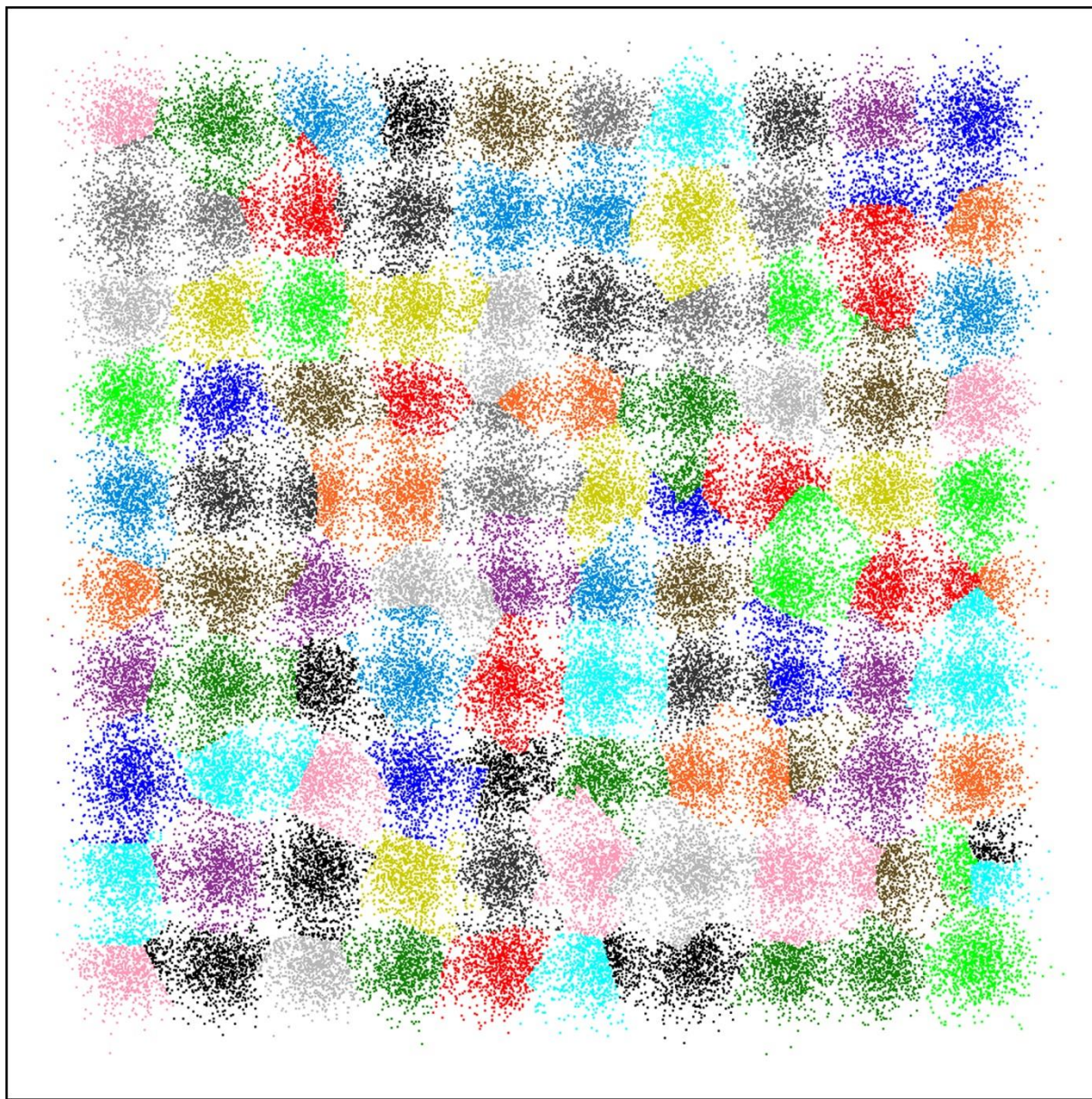
Табела 6.7 - Резултати модификација VNS-а за генерисане скупове података 01 и 02

|                    | Време (ms)        | Итерације   | Позиви мере      | CH                | DB            | S             |
|--------------------|-------------------|-------------|------------------|-------------------|---------------|---------------|
| <b>Тест_03</b>     |                   |             |                  |                   |               |               |
| <b>Основни VNS</b> |                   |             |                  |                   |               |               |
| Најбољи сусед      |                   |             |                  |                   |               |               |
| VNS околина 2      | <b>3194.2000</b>  | <b>34.3</b> | <b>5542657.3</b> | 18293.3349        | 0.5520        | 0.5834        |
| VNS околина 3      | 15549.0000        | 39.2        | 5648068.1        | <b>19722.7714</b> | <b>0.5303</b> | <b>0.5928</b> |
| Први бољи сусед    |                   |             |                  |                   |               |               |
| VNS околина 2      | <b>692.8000</b>   | 40.5        | <b>5520352.5</b> | 16388.5084        | 0.6154        | 0.5404        |
| VNS околина 3      | 2711.9000         | <b>38.4</b> | 5571092.5        | <b>18314.6548</b> | <b>0.5479</b> | <b>0.5778</b> |
| <b>Општи VNS</b>   |                   |             |                  |                   |               |               |
| VND                |                   |             |                  |                   |               |               |
| VNS околина 2      |                   |             |                  |                   |               |               |
| VND околина 1      | <b>151.8000</b>   | 49.2        | <b>3703291.2</b> | 16015.9609        | 0.6166        | 0.5412        |
| VND околина 2      | 4701.3000         | 43.4        | 3736997.6        | 21141.5481        | <b>0.5148</b> | 0.6007        |
| VND околина 3      | 7611.8000         | 42.1        | 3804234.3        | 20673.5083        | 0.5323        | 0.5988        |
| 10                 | 29223.8000        | <b>29.1</b> | 4001090.2        | <b>21162.4767</b> | 0.5165        | <b>0.5993</b> |
| VNS околина 3      |                   |             |                  |                   |               |               |
| VND околина 1      | <b>242.3000</b>   | 48.1        | <b>4145419.4</b> | 15703.2022        | 0.6164        | 0.5418        |
| VND околина 2      | 7489.8000         | 30.9        | 4188233.7        | 20087.2682        | 0.5271        | 0.5930        |
| VND околина 3      | 13838.8000        | <b>30.0</b> | 4295504.0        | 21025.9472        | 0.5210        | 0.5996        |
| VND околина 10     | 133606.6000       | 41.1        | 4988074.4        | <b>21202.4456</b> | <b>0.5118</b> | <b>0.6033</b> |
| <b>Тест_04</b>     |                   |             |                  |                   |               |               |
| <b>Основни VNS</b> |                   |             |                  |                   |               |               |
| Најбољи сусед      |                   |             |                  |                   |               |               |
| VNS околина 2      | <b>5094.1000</b>  | <b>39.0</b> | <b>7758376.7</b> | 20881.7573        | 0.6030        | 0.5200        |
| VNS околина 3      | 23823.0000        | 43.7        | 7871272.2        | <b>21417.3752</b> | <b>0.6100</b> | <b>0.5219</b> |
| Први бољи сусед    |                   |             |                  |                   |               |               |
| VNS околина 2      | 897.9000          | <b>43.3</b> | <b>7731724.9</b> | 18897.3405        | 0.6533        | 0.4941        |
| VNS околина 3      | 3609.2000         | 45.2        | 7787123.1        | <b>19037.6668</b> | <b>0.6498</b> | <b>0.5023</b> |
| <b>Општи VNS</b>   |                   |             |                  |                   |               |               |
| VND                |                   |             |                  |                   |               |               |
| VNS околина 2      |                   |             |                  |                   |               |               |
| VND околина 1      | <b>241.6000</b>   | 54.2        | <b>5713680.6</b> | 17118.7231        | 0.7120        | 0.4665        |
| VND околина 2      | 6006.2000         | 31.4        | 5747358.9        | 21327.6057        | 0.6154        | 0.5234        |
| VND околина 3      | 8225.9000         | <b>29.7</b> | 5800560.7        | 22072.5848        | 0.5834        | 0.5305        |
| VND околина 10     | <b>52650.9000</b> | 37.1        | 6040859.4        | <b>22701.9067</b> | <b>0.5825</b> | <b>0.5390</b> |
| VNS околина 3      |                   |             |                  |                   |               |               |
| VND околина 1      | <b>451.8000</b>   | 69.5        | <b>6213529.8</b> | 18596.2649        | 0.6760        | 0.4918        |
| VND околина 2      | 10718.0000        | <b>30.7</b> | 6260827.9        | 22150.3388        | 0.5886        | 0.5371        |
| VND околина 3      | 20799.1000        | 31.8        | 6362021.1        | 22372.9422        | 0.5836        | 0.5335        |
| VND околина 10     | 215018.0000       | 42.0        | 7112764.3        | <b>23152.4486</b> | <b>0.5762</b> | <b>0.5408</b> |

Табела 6.8 - Резултати модификација VNS-а за генерисане скупове података 03 и 04



Сви до сада коришћени скупови података имају између 150 (Ирис) и 13 000 (Тест\_04 са 13006) података. Видели смо да ако нема преклапања кластера, коришћене модификације VNS-а релативно брзо доводе до квалитетних подела скупа у кластере према задатој функцији циља. Ради истраживања граничних могућности VNS-а, алгоритам је тестиран на скупу који има 100 000 података које треба кластеровати у 100 кластера који се не преклапају. Коришћен је познати скуп података под називом Birch1 [25]. Резултат постигнут на овом скупу података коришћењем GVNS-а дат је на слици 6.4.



Слика 6.5 - Резултат GVNS-а за скуп података *birch1*

## 7. ЗАКЉУЧАК

У овом раду су представљене неке модификације методе променљивих околина и приказана примена истих у кластеровању вишедимензионих података. На почетку рада је дат преглед неких најпознатијих модификација методе променљивих околина (2. поглавље). Затим је уведен појам кластеровања и приказана основна подела на различите типове кластеровања и кластера у поглављу 3.

Модификације методе променљивих околина су примењене на партиционо кластеровање. Најпре је изложен проблем кластеровања као НП-тежак проблем са циљем минимизирања суме квадрата растојања тачака до њиховог центроида [8]. Онда су посебно изложени и размотрени сви битни делови методе променљивих околина у примени на кластеровање почевши од избора иницијалног решења и оптималног броја кластера, преко дефинисања околина, имплементације локалног претраживања и размрдавања, до избора критеријума заустављања. Приказана је нова идеја за формирање околина за овако дефинисан проблем, која је далеко мање сложености од, рецимо, пермутацијског представљања неког допустивог решења и стандардног формирања околина пермутовањем одређеног броја координата вектора решења. У петом поглављу је детаљно описана имплементација алгорита и дат псеудокод за основну и општу модификацију методе променљивих околина.

Резултати који су добијени применом описаних модификација за дефинисан проблем кластеровања приказани су у поглављу 6. За тестирање су коришћене јавно доступне тест инстанце и два скупа података направљена за потребе овог рада. Модификације су се показале ефикасним у кластеровању скупова података са јасно раздвојеним класама података, сфероидног облика и релативно сличних пречника. Назначен је утицај избора функције циља и критеријума заустављања на квалитет кластеровања и време извршавања алгорита.

На тему партиционог кластеровања написане су хиљаде радова, али могућности методе променљивих околина још увек у тој области нису детаљно истражене. Велики број

параметара који се у зависности од потребе могу варирати, могућности најразличитијих организовања и имплементирања околина и могућности комбиновања методе променљивих околина са другим метахеуристикама, чине да даља истраживања ове методе обећавају добре резултате, не само у области кластеровања, већ и у решавању други комплексних алгоритамских проблема данашњице.

## ЛИТЕРАТУРА

- [1] Mladenović N, Hansen P. Variable neighborhood search. *Computers and Operations Research*, 1997;24(11); pp 1097–1100.
- [2] Mladenović N, Hansen P. Variable Neighborhood Search. In: Burke E. K. Kendall G. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* : Boston, MA: Springer US; 2005. pp 211-238.
- [3] Mladenović N, Hansen P. Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research*. 2007;130: pp 449-467.
- [4] Everitt BS. *Cluster Analysis*, 3rd ed. New York, Toronto: Halsted Press; 1993.
- [5] Jain AK, Dubes RC. *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall, Inc; 1988
- [6] Das S, Abraham A, Konar A. *Metaheuristic Clustering*. Berlin, Heidelberg: Springer-Verlag; 2009. pp 22.
- [7] Tan P, Steinbach M, Kumar V. *Introduction to Data Mining*. San Francisco: Pearson Addison-Wesley; 2006. pp 490-496.
- [8] Aloise D, Deshpande A, Hansen P, and Popat P. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*. 2009 May;75(2) pp 245-248.
- [9] Дражић З, Модификације методе променљивих околина и њихове примене за решавање проблема распоређивања преноса датотека [докторска дисертација]. [Београд]: Математички факултет; 2014. 124 стр.
- [10] Das S, Abraham A, Konar A. *Metaheuristic Clustering*. Berlin, Heidelberg: Springer-Verlag; 2009. pp 218-222.
- [11] Wang X, Qiu W, Zamar R. H. CLUES: A Non-parametric Clustering Method Based on Local Shrinking. *Computational Statistics & Data Analysis*. 2007 Sep;52(1): pp 286-298
- [12] Kärkkäinen I, Fränti P. Dynamic local search algorithm for the clustering problem. Joensuu: University of Joensuu, Department of Computer Science; 2001. pp 11. Report No.: A-2002-6.
- [13] Fränti P, Virtajoki O. Iterative shrinking method for clustering problems. *Pattern Recognition*. 2006 May;39(5): pp 761-765.
- [14] Desgraupes B. *Clustering Indices*, University Paris Ouest, Lab Modal'X 2013 Apr
- [15] Saitta S, Raphael B, Smith, IFC. A Bounded Index for Cluster Validity. In: Perner P. *Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM; 2007 July 18-20; Leipzig, Germany*. Berlin Heidelberg: Springer; 2007. pp 174-187.
- [16] Thahira T, Sequeira JV, Sameema, Ansari Z. Performance Analysis of Self-Organizing Neural NetworkBased Clustering. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2015 Feb;5(2): pp 820-829.

- [17] Murty M. R, Naik A, Murthy J. V. R, Reddy P. V. G. D. P, Satapathy S. C, Parvathi K. Automatic Clustering Using Teaching Learning Based Optimization. Applied Mathematics [Internet]. 2014 May [cited 2016 Sep 10];5(8) pp.1202-1211. Available from: [http://file.scirp.org/pdf/AM\\_2014050617255935.pdf](http://file.scirp.org/pdf/AM_2014050617255935.pdf)
- [18] Machine Learning Repository [Internet]. Irvine: Center for Machine Learning and Intelligent Systems. 1988 – Iris Data Set; [Cited 2016 Sep 10]; Available from: <https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>
- [19] Machine Learning Repository [Internet]. Irvine: Center for Machine Learning and Intelligent Systems. 1993 – Glass Identification Data Set; [Cited 2016 Sep 10]; Available from: <https://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data>
- [20] Machine Learning Repository [Internet]. Irvine: Center for Machine Learning and Intelligent Systems. 1991 – Wine Data Set; [Cited 2016 Sep 10]; Available from: <https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data>
- [21] Machine Learning Repository [Internet]. Irvine: Center for Machine Learning and Intelligent Systems. N/A – Vowels Data Set; [Cited 2016 Sep 10]; Available from: <https://archive.ics.uci.edu/ml/machine-learning-databases/JapaneseVowels-mld/ae.test>
- [22] Machine Learning [Internet]. Joensuu: School of Computing University of Eastern Finland. 2006 – S-Sets S1; [Cited 2016 Sep 10]; Available from: <https://cs.joensuu.fi/sipu/datasets/s1.txt>
- [23] Machine Learning [Internet]. Joensuu: School of Computing University of Eastern Finland. 2006 – S-Sets S2; [Cited 2016 Sep 10]; Available from: <https://cs.joensuu.fi/sipu/datasets/s2.txt>
- [24] Machine Learning [Internet]. Joensuu: School of Computing University of Eastern Finland. 2002 – A-Sets A1; [Cited 2016 Sep 10]; Available from: <https://cs.joensuu.fi/sipu/datasets/a1.txt>
- [25] Machine Learning [Internet]. Joensuu: School of Computing University of Eastern Finland. 1997 – Birch-Sets Birch1; [Cited 2016 Sep 10]; Available from: <https://cs.joensuu.fi/sipu/datasets/birch1.txt>