

Компьютерын ухаан

# Сүлжээнээс бүлэг илрүүлэх Girvan-Newman алгоритмын өргөтгөл

П.Батцэнгэл<sup>1</sup>, Г.Гантулга<sup>1</sup>, Б.Доржнамжирмаа<sup>1</sup>, П.Далайжаргал<sup>1,\*</sup>

<sup>1</sup>МУИС, МКУТ

Хүлээн авсан 2023.05.01; Хянагдсан 2023.08.06; Зөвшөөрөгдсөн 2023.08.28

\*Холбоо баригч зохиогч: dalajargal@seas.num.edu.mn

## Хураангуй

Сүлжээнээс бүлэг илрүүлэхэд (community detection) өргөн хэрэглэгддэг, стандарт алгоритм болох Girvan-Newman (цаашид GN гэж товчлох) алгоритм, түүний хурдыг сайжруулах боломжуудыг энэ өгүүлэлд авч үзнэ. GN алгоритм нь сүлжээний ирмэгүүдийн хоорондын төв (betweenness centrality)-ийг бодож хамгийн өндөр оноо авсан нэг ирмэгийг устгах, дахин хоорондын төв (цаашид ХТ гэж товчлох) бодож өндөр оноотой ирмэгийг устгах гэх мэтээр ирмэгүүдийг устгаж, бүлгүүдийг илрүүлдэг. GN алгоритмын ажиллах хугацаа нь  $O(n \cdot m^2)$  (m-нийт ирмэгийн тоо, n-нийт оройн тоо) учир олон ирмэгтэй том сүлжээнд ашиглахад хүндрэлтэй юм. Алгоритмын хурдыг сайжруулах зорилгоор нэг удаа бодсон ХТ-ийн оноог ашиглан нэгэн зэрэг олон ирмэг устгах, ХТ оноог ойролцоогоор бодох гэсэн хоёр санааг ашиглан GN алгоритмын өргөтгөл дөрвөн хувилбарыг үүсгэв. Сүлжээний ирмэгүүдийг ХТ-ийн оноогоор эрэмбэлж эхний  $\sqrt{m}$  оройг нэгэн зэрэг устгах хоёр алгоритм (цаашид GN-2A, GN-2B гэж нэрлэсэн), GN-2A, GN-2B алгоритмуудыг ойролцоо ХТ-ийн оноог бодох аргатай хослуулсан (цаашид GN-3A, GN-3B гэж нэрлэсэн) мөн хоёр хувилбар зохиомжлов. Өргөн хэрэглэгддэг таван сүлжээний өгөгдлийг ашиглан GN-2A, GN-2B, GN-3A, GN-3B алгоритмыг тестэлж, GN алгоритмтай харьцуулан гүйцэтгэлийг үнэллээ. Том сүлжээний хувьд GN алгоритмтай харьцуулахад хурдыг 8-33 дахин сайжруулсан үр дүн харуулж байна.

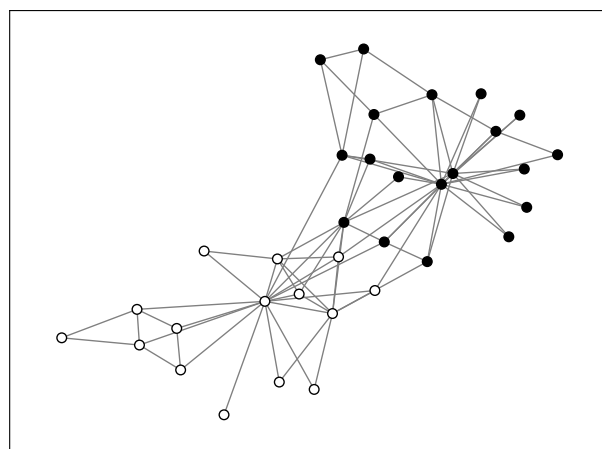
**Түлхүүр үг:** бүлэг бүтэц, комплекс сүлжээ, сүлжээнээс бүлэг илрүүлэх, Girvan-Newman алгоритм, хоорондын төв

## 1 Удиртгал

Хүмүүс, амьтад, юмс хоорондын харилцааг сүлжээгээр загварчлан судлах арга сүүлийн жилүүдэд эрчимтэй хөгжиж байна. Ихэнх сүлжээ бүлгүүдээс тогтдог [1, 2]. Нэг бүлгийн оройнууд хоорондоо холбоо ихтэй, харин тухайн нэг бүлгийн оройн бусад бүлгийн оройнуудтай холбоо багатай байна. Сүлжээний энэ шинж чанарыг бүлэг бүтэц (community structure) гэж нэрлэдэг [1]. Сүлжээнээс бүлэг илрүүлэх (community detection) нь хоорондоо харилцаа холбоотой юмсыг мэдэх, харилцан үйлдлүүд нь адилхан объектуудыг тодорхойлох, ирээдүйд бий болох холбоог таамаглах [3], тэдгээр бүлгүүдэд тохирсон бодлого хэрэгжүүлэх, хот сууриныг төлөвлөлт хийх, томоохон дэд бүтцийг сайжруулах [4] гэх мэт практик ач холбогдолтой.

Зураг 1-т туршилтад ашигласан karate өгөгдөл дээр илэрсэн бүлэг бүтцийг дүрсэлж харуулав. Уг сүлжээг судлаач Zachary их сургуулийн каратэ клубийн гишүүд хоорондын найзын харилцаанд тулгуурлан сүлжээг загварчилсан [5] бөгөөд тус клубийн “John A” болон “Mr. Hi” нарын хоорондын маргааны улмаас клуб хоёр хуваагдан салж цаашид тусдаа үйл ажиллагаа явуулах болсон. Энэхүү

сүлжээнээс GN алгоритмаар бүлэг илрүүлэхэд үүсэх бүтцийг зурагт харуулав. Хараар будсан нь нэг бүлэг харин цагаанаар будсан нөгөө бүлэг бөгөөд энэ хоёр бүлэг тус тусдаа үйл ажиллагаа явуулах болсон байна.



Зураг 1: Karate сүлжээн дээр илэрч буй бүлэг бүтэц

Хоорондын төв (betweenness centrality) нь сүл-

жээний оролцогч хооронд мэдээлэл дамжуулах үйл ажиллагааны оролцоог үнэлэн ирмэг бүрд эрэмбэ (оноо) өгөх арга юм [6, 7]. Боломжит хос оройн хоорондын хамгийн богино замд тухайн ирмэг хэдэн удаа орж байгаа тоог нийт богино замын тоонд харьцуулсан харьцааг хоорондын төв (ХТ)-ийн оноо гэнэ. ХТ-ийн оноо нь өндөр бол тэр ирмэг сүлжээний гол төв буюу бүлгүүдийг хооронд нь холбож буй ирмэг гэж үзнэ. ХТ оноог тооцоох  $O(n \cdot m)$  хугацааны алгоритмыг Brandes 2001 онд зохиосон [8], энд  $n$  нь сүлжээний оройн тоо,  $m$  нь сүлжээний ирмэгийн тоо юм. ХТ-ийн оноог Томьёо 1-ийг ашиглан тооцоолно. Томьёо 1-д  $\sigma_{uv(e)}$  нь  $e$  ирмэгийг дайрсан  $u$  оройгоос  $v$  орой хүртэлх богино замын тоо.  $\sigma_{uv}$  нь  $u$  оройгоос  $v$  орой хүртэлх нийт богино замын тоо юм.

$$BC(e) = \sum_{u \in V} \sum_{v \in V/u} \frac{\sigma_{uv(e)}}{\sigma_{uv}} \quad (1)$$

Сүлжээнээс бүлэг илрүүлэх стандарт аргуудын нэг болох Girvan-Newman (GN) алгоритмыг [1] судалцажээ. GN алгоритм бүлэг илрүүлэхдээ ХТ-ийн оноо хамгийн өндөр байх 1 ирмэг устгаад үлдсэн сүлжээний ирмэгүүдийн ХТ-ийн оноог бодож ХТ-ийн оноо өндөр 1 ирмэгийг дахин устгана. Энэ үйлдлийг зорилтот хэмжээний бүлэг илрүүлэх хүртэл үргэлжлүүлэн давтана [1]. GN алгоритм нь  $m$  удаа ХТ-ийг тооцоолох бөгөөд ажиллах хугацаа нь  $O(n \cdot m^2)$  болно. GN-ны  $O(n \cdot m^2)$  хугацаа нь том сүлжээний хувьд удаан ажиллах шалтгаан болдог.

GN алгоритмын сайжруулах асуудлыг [9–11] зэрэг ажлуудад авч үзсэн байна. Өндөр ХТ-ийн оноотой хэд хэдэн ирмэг байвал зөвхөн нэгийг нь устгахын оронд тэдгээр ирмэгүүдийг бүгдийг устгах санааг [10] дэвшүүлжээ. Онолын хугацааны ээдрээ нь GN-тэй адил  $O(n \cdot m^2)$  бөгөөд туршилтаар ойролцоо хугацаанд ажилласан байна. GN алгоритмыг Cuscoо хайлтын аргатай хослуулан, бүлэг илрүүлэх чанарыг сайжруулж, сүлжээнээс нөлөөтэй тоглогчийг илрүүлэх бодлогод ашиглах судалгаа [11]-г гүйцэтгэсэн байна. GN алгоритмд ашигласан бүлэг бүтцийн тодорхойлолтыг сайжруулах зорилго бүхий судалгааг [9]-д хийжээ. Судалгааны ажил [9, 11] дэх судалгаанууд нь бидний ажлаас өөр зорилготойг энд тэмдэглэе.

GN алгоритмын хурдыг сайжруулах зорилгоор дараах хоёр шинэ санааг дэвшүүлж байна. Үүнд, олон ирмэг нэгэн зэрэг устгах болон ХТ-ийн ойролцоо оноог ашиглах юм. Эхний санаагаар ХТ-ийн тооцооллоос нэг ирмэг устгахын оронд олон (тухайлбал, 10 ирмэг нэгэн зэрэг) ирмэг устгах GN-2A алгоритмыг зохиомжлов. Энэ санаа нь алгоритмын нийт давталтын тоог  $m$ -ээс  $\sqrt{m}$  болгон багасгана. Иймд GN-2A алгоритм нь  $O(n \cdot m^{1.5})$  хугацаанд ажиллана.

Өнгөрсөн арван жилд ХТ оноог ойролцоогоор бодох (approximate betweenness) судалгаа эрчимтэй хөгжиж байна [12, 13]. Алгоритмын давталт бүрд

$O(nm)$  хугацаагаар ХТ-ийн оноог бодож эрэмбэлэх хугацааг хэмнэхийн тулд ХТ-ийн ойролцоо оноог ашиглан ирмэгүүдийг эрэмбэлж, топ  $\sqrt{m}$  ирмэгийг нэгэн зэрэг устгах GN-3A алгоритмыг зохиомжлов.

Нэгэн зэрэг олон ирмэг устгах үед нэг нэгээр устгах аргад илэрдэггүй сул тал шинээр үүсэв. Тодруулбал, олон тооны жижиг (1-2 оройтой) хэмжээтэй компонентүүд үүсэж байв. Энэ асуудлыг шийдэх зорилгоор ирмэг бүрийг устгасны дараа шинээр үүсэх компонент нь хэт жижиг (өгөгдсөн босго хэмжээнээс бага байх) байвал тухайн ирмэгийг устгахгүй үлдэнэ. Үүсэх компонентийн хэмжээ (оройн тоо)-д босго тавьж, тухайн ирмэгийг устгах эсхийг шийддэг GN-2A, GN-3A алгоритмуудын өөрчилсөн хувилбаруудыг тус бүр GN-2B, GN-3B гэж нэрлэв. Өргөн хэрэглэгддэг таван сүлжээний өгөгдлийг ашиглан дэвшүүлсэн дөрвөн алгоритмыг зүгшрүүлэн, параметр тохируулга хийж, GN алгоритмын үр дүнтэй харьцуулав. Дэвшүүлсэн алгоритмууд GN-н ажиллах хурдыг олон дахин сайжруулаа. Гүйцэтгэлийн чанар нь үндсэн алгоритмын үр дүнтэй харьцуулахад 75%-аас дээш давхцаж байна.

## 2 Арга зүй

Энэ бүлэгт дэвшүүлж буй GN-2A, GN-2B, GN-3A, GN-3B алгоритмуудыг тодорхойлно. Эхлээд түүврийн аргад суурилсан ХТ-н ойролцоо оноо бодох Riondato болон Kornaropoulos [12] нарын танилцуулсан алгоритмыг (цаашид RK гэж товчлох) товч авч үзье. ХТ-ийн оноог ойролцоолох RK алгоритм нь таамгаар  $(u, v)$  хос оройг сонгож,  $u$  оройгоос  $v$  хүрэх богино замуудаас санамсаргүйгээр нэгийг сонгоно. Сонгогдсон замд харьяалагдах оройнуудын ХТ оноог  $1/r$ -р нэмэгдүүлнэ, энд  $r = (1/\epsilon^2) \cdot \lceil \log_2(\text{diam} - 2) + \ln(1/\delta) \rceil$ ,  $\epsilon$  нь ХТ оноо, түүний ойролцоо утгын зөрөөний хязгаарыг илэрхийлнэ,  $\delta$  нь энэ ойролцооллын итгэх завсар юм. Өөрөөр хэлбэл  $P(|bc(v) - bc(v)| > \epsilon) < \delta$  байна, энд  $bc(v)$  нь ойролцоо ХТ-н оноог,  $bc(v)$  нь ХТ-н оноог илэрхийлнэ. Таамгаар зам сонгож, түүнд харьяалагдах оройнуудын ХТ-н оноог  $1/r$  нэмэгдүүлэх процессыг  $r$  удаа давтан гүйцэтгэнэ. Иймд  $r$  нь алгоритмын хурд, ХТ-н онооны чанартай шууд хамааралтай. Өгөгдсөн  $\epsilon$ ,  $\delta$ -н хувьд сүлжээний хэмжээнээс үл хамааран  $r$  тогтмол байдаг. Энэ алгоритмын хурд нь  $(r \cdot (n + m))$  байна. Параметруудийн зарим утгын хувьд  $r$  -н тооцоолж үзье. Тухайлбал,  $\epsilon = 0.05$ ,  $\delta = 0.1$ , диаметр 6 байх үед  $r = 1720$  байна. RK алгоритмын хуурмаг кодыг Алгоритм 1-д оруулав. Алгоритм 1 дахь `get_approximate_diameter(G)` функц графын ойролцоо диаметрийг  $O(n + m)$  хугацаанд буцаана. Таамгаар нэг орой сонгож, тэр оройноос бусад бүх орой хүртэлх богино замыг төвшний нэвтрэлтээр тооцоолно, эдгээрээс хамгийн урт хоёр замын нийлбэрийг ойролцоо диаметрээр буцаана.

Алгоритм 2-д Girvan-Newman алгоритм [1]-н хуурмаг кодыг сийрүүлэв. GN-н өргөтгөл GN-2A,

**Алгоритм 1** RK алгоритм

---

```

RK(G,  $\epsilon$ )
1:  $bc(v) \leftarrow 0$ 
2:  $diam \leftarrow get\_approximate\_diameter(G)$ 
3:  $\delta \leftarrow 0.9$ 
4:  $r \leftarrow (1/\epsilon^2) \cdot [\log_2(diam - 2) + \ln(1/\delta)]$ 
5: for  $i = 1$  to  $r$ 
6:    $(u, v) \leftarrow$  таамгаар хос орой сонгох.
7:    $P_{u,v} \leftarrow u$  оройгоос  $v$  хүрэх богино замуудаас таамгаар нэгийг сонгох.
8:    $w \in P_{u,v}$  бүрийн хувьд:  $bc(w) \leftarrow bc(w) + 1/r$ 

```

---

GN-3A, GN-2B, GN-3B алгоритмуудын хуурмаг кодыг Алгоритм 3, Алгоритм 4, Алгоритм 5, Алгоритм 6-д тус тус оруулав. Эдгээрт ашигласан  $top\_edges(bc, k)$  функц нь өгөгдсөн ХТ оноогоор эрэмбэлэн топ-к ирмэгийг олж,  $topK\_edges$  хувьсагчид буцаана.  $remove\_edge(G, topK\_edges, threshold)$  функц нь  $G$  графаас  $topK\_edges$ -д байх ирмэгүүдийг нэг нэгээр устгана. Ирмэгийг устгах эсэхийг тогтоохдоо нөхцөл шалгаж шийднэ. Тухайн ирмэгийг устгасны дараа  $threshold$  параметрээс жижиг хэмжээтэй компонент (оройн тоо) шинээр үүсэж байвал тэр ирмэгийг устгахгүй үлдэнэ.  $threshold = 1$  байх нь нөхцөл шалгахгүй байхтай ижил юм. GN-2A болон GN-3A алгоритмууд нь  $threshold = 1$  буюу ирмэг устгахад нөхцөл шалгадаггүй. Харин GN-2B болон GN-3B алгоритмууд нь  $threshold = \log_2(n + m)$  босго оноотойгоор ирмэгийг устгадаг.

GN-2A: Графын ирмэгийн ХТ-ийн оноог бодож эрэмбэлээд ХТ-ийн оноо хамгийн өндөр  $\sqrt{m}$  оройг устгана. Тодорхойлж өгсөн хэмжээний бүлэг илрүүлэх хүртэл давтаж ажиллана. Үүсэх бүлгийн хэмжээ (оройн тоо)-д босго тавихгүй учир олон тооны жижиг (1-2 оройтой) хэмжээтэй бүлгүүд үүсгэх сул талтай.

GN-3A: Графын ирмэгийн ХТ-ийн оноог ойролцоолох алгоритм (Алгоритм 1) ашиглан эрэмбэлнэ. ХТ-ийн оноо хамгийн өндөр  $\sqrt{m}$  оройг устгана. Тодорхойлж өгсөн тооны бүлэг илрүүлэх хүртэл үйлдлээ давтаж ажиллана. Мөн үүсэх бүлгийн хэмжээ (оройн тоо)-д босго тавихгүй учир олон тооны жижиг (1-2 оройтой) хэмжээтэй бүлгүүд үүсгэх сул талтай.

GN-2B: Графын ирмэгийн ХТ-ийн оноог бодож эрэмбэлээд ХТ-ийн оноо хамгийн өндөр  $\sqrt{m}$  оройг устгана. Шинээр үүссэн бүлгийн хэмжээ (оройн тоо) хэрэглэгчийн тодорхойлж өгсөн босго хэмжээнээс бага байвал үүссэн бүлгийг буцаан нэгтгэх зорилгоор устгасан ирмэгээ буцаан нэмнэ. Тодорхойлж өгсөн тооны бүлэг илрүүлэх хүртэл үйлдлээ давтаж ажиллана (Алгоритм 5).

GN-3B: Графын ирмэгийн ХТ-ийн оноог ойролцоолох алгоритм ашиглан эрэмбэлнэ (Алгоритм 1). ХТ-ийн оноо хамгийн өндөр  $\sqrt{m}$  оройг устгана. Шинээр үүссэн бүлгийн хэмжээ (оройн тоо) хэрэглэг-

чийн тодорхойлж өгсөн босго хэмжээнээс бага байвал үүссэн бүлгийг буцаан нэгтгэх зорилгоор устгасан ирмэгээ буцаан нэмнэ. Хэрэглэгчийн тодорхойлж өгсөн тооны бүлэг үүсгэх хүртэл үйлдлээ давтаж ажиллана (Алгоритм 6).

**Алгоритм 2** GN алгоритм

---

```

GN(G, num_community)
1: threshold  $\leftarrow 1$ 
2: while # of communities in  $G < num\_community$ 
3:    $bc\_score \leftarrow exact\_edge\_betweenness(G)$ 
4:    $e \leftarrow top\_edges(bc\_score, 1)$ 
5:    $G \leftarrow remove\_edge(G, edges, threshold)$ 

```

---

**Алгоритм 3** GN-2A алгоритм

---

```

GN-2A(G, num_community,  $\alpha$ )
1: threshold  $\leftarrow 1$ 
2: while # of communities in  $G < num\_community$ 
3:    $bc\_score \leftarrow exact\_edge\_betweenness(G)$ 
4:    $topK\_edges \leftarrow top\_edges(bc\_score, \alpha \cdot \sqrt{m})$ 
5:    $G \leftarrow remove\_edge(G, topK\_edges, threshold)$ 

```

---

**Алгоритм 4** GN-3A алгоритм

---

```

GN-3A(G, num_community,  $\alpha$ )
1: threshold  $\leftarrow 1$ 
2: while # of communities in  $G < num\_community$ 
3:    $bc\_score \leftarrow exact\_edge\_betweenness(G, \epsilon)$ 
4:    $topK\_edges \leftarrow top\_edges(bc\_score, \alpha \cdot \sqrt{m})$ 
5:    $G \leftarrow remove\_edge(G, topK\_edges, threshold)$ 

```

---

## 3 Туршилт, үр дүн

Бүлэг илрүүлэх стандарт GN алгоритм болон тодорхой тооны ирмэг устгах замаар бүлэг илрүүлэх GN-2A, GN-2B, GN-3A, GN-3B алгоритмуудын хугацаа болон гүйцэтгэлийг харьцуулан энэ бүлэгт харуулав. Туршилтыг Intel(R) Core(TM) i7-3610 CPU @ 2.30GHz компьютер ашиглан python 3 хэл дээр Networkx графын санг ашиглан хэрэгжүүлэв.

### 3.1 Туршилтын өгөгдөл

Туршилтад ашигласан сүлжээнүүдийг Хүснэгт 1-д харуулав. *Karate club* сүлжээ нь их сургуулийн каратэ клубийн гишүүн хоорондын харилцааны сүлжээ [5]. Сүлжээний оройнууд нь клубийн гишүүдийг, ирмэгүүд нь хүмүүс хоорондын найзын холбоог илэрхийлнэ. *Dolphine* сүлжээ нь Шинэ Зеландад байрлах хэсэг бүлэг дельфиний харилцаа хамаарлын сүлжээ [5]. Сүлжээний орой нь дельфин, ирмэгүүд нь тэдгээрийн хоорондын нөхөрлөлийн холбоо хамаарал юм. *Polbooks* сүлжээ нь Amazon.com сайтаар зарагдсан АНУ-ын улс төрийн тухай номын сүлжээ.

**Алгоритм 5** GN-2B алгоритм

```

GN-2B(G, num_community,  $\alpha$ ,  $\epsilon$ )
1: threshold  $\leftarrow \log_2(n + m)$ 
2: while # of communities in G < num_community
3:   bc_score  $\leftarrow$  approximate_edge_betweenness(G,
    $\epsilon$ )
4:   topK_edges  $\leftarrow$  top_edges(bc_score,  $\alpha \cdot \sqrt{m}$ )
5:   G  $\leftarrow$  remove_edge(G, topK_edges, threshold)

```

**Алгоритм 6** GN-3B алгоритм

```

GN-3B(G, num_community,  $\alpha$ ,  $\epsilon$ )
1: threshold  $\leftarrow \log_2(n + m)$ 
2: while # of communities in G < num_community
3:   bc_score  $\leftarrow$  approximate_edge_betweenness(G,
    $\epsilon$ )
4:   topK_edges  $\leftarrow$  top_edges(bc_score,  $\alpha \cdot \sqrt{m}$ )
5:   G  $\leftarrow$  remove_edge(G, topK_edges, threshold)

```

Сүлжээний орой нь улс төрийн ном, ирмэгүүд нь тухайн хоёр номыг хамтад нь худалдан авсныг харуулна. *Power* сүлжээ нь АНУ-ын баруун мужуудын эрчим хүчний сүлжээний талаарх мэдээллийг агуулдаг. Ирмэг нь цахилгаан хангамжийн шугамыг илэрхийлдэг. Орой нь генератор, трансформатор болон дэд станц юм. *Facebook page* сүлжээ нь Facebook-ын итгэмжлэгдсэн хоолны хуудаснуудын сүлжээ ба орой нь хуудас, ирмэгүүд нь дундын дагачдыг илэрхийлнэ [5]. *Facebook* сүлжээ нь судалгаанд оролцсон бүлэг хүмүүсийн найзуудын жагсаалтаас бүтэх граф [14] юм.

*Хүснэгт 1: Туршиланд ашигласан өгөгдлүүд. Оройн тоо (n), ирмэгийн тоо (m), сүлжээний дундаж зэрэг (deg), сүлжээний диаметр(dia)*

Сүлжээ	n	m	deg	dia
Karate club	34	78	4	5
Dolphine	62	159	5	8
Polbooks	105	441	8	7
Power	4941	6594	2	19
Facebook	4039	88234	0	8

Алгоритмуудыг харьцуулахдаа ажилласан хугацаа болон илрүүлсэн бүлгүүдийн оройнуудыг GN алгоритмын илрүүлсэн бүлгүүдийн оройнуудтай харьцуулж давхцалыг хувиар илэрхийллээ. Энэ давхцалын хувийг алгоритмын шийдийн төстэй чанар (эсвэл X% -ийн төстэй үр дүн гэж хэрэглэв) гэж нэрлэв.

### 3.2 Алгоритмын параметр тохируулга

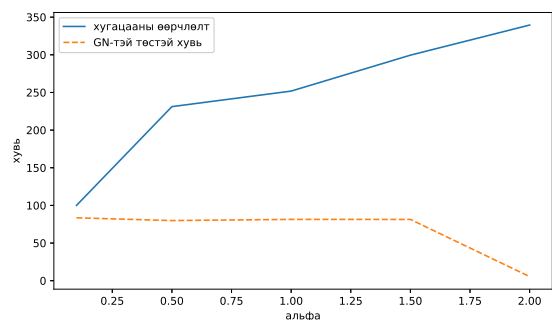
Дэвшүүлж буй алгоритмууд  $\alpha$ ,  $\epsilon$  гэсэн хоёр параметрээс хамааралтай.  $\alpha$  нь сүлжээнээс нэг удаа устгахдаа хэдэн ширхэг ирмэг устгахыг тохируулах параметр бол  $\epsilon$  нь сүлжээний ХТ-ийг бодоход зөвшөөрөгдөх алдааны хэмжээ болно.

Дэвшүүлж буй 4 алгоритм сүлжээнээс бүлэг илрүүлэхдээ нэг удаа устгах ирмэгийн тоог  $\alpha\sqrt{m}$  хэмжээгээр сонгодог. Туршилтыг *Karate club*, *Polbooks*, *Power*, *Facebook* гэсэн дөрвөн сүлжээн дээр гүйцэтгэв.

#### 3.2.1 Ирмэгийн устгалын хэмжээг тохируулах параметр

Эхлээд  $\alpha$  параметрийг 0.1, 0.5, 1, 1.5 гэсэн 4 утгаар GN-2B, GN-3B алгоритмууд дээр туршиж, GN алгоритмтай харьцуулсан үр дүнг Хүснэгт 2, 3-т харуулав. GN-3B алгоритмын хувьд ХТ-ийн оноо бодоход зөвшөөрөгдөх алдааны коэффициент болох  $\epsilon$ -ы утгыг 0.05-аар сонгож туршилтыг гүйцэтгэв.

GN-2B алгоритмын хувьд  $\alpha$  параметрийн хамаарлыг Хүснэгт 2-с харахад  $\alpha = 0.1$  үед жижиг сүлжээ болох *karate club* сүлжээн дээр GN-ээс удаан ажилласан боловч бусад 3 сүлжээн дээр 1.6–3 дахин хурдан ажилласан. Харин  $\alpha$ -ийн бусад утгын хувьд бүх сүлжээнүүд дээр GN-ээс хурдан ажилласан бөгөөд  $\alpha$ -ийн утга өсөх тусам хурдан ажиллах хандлага ажиглагдаж байна. Харин чанарын хувьд бүх  $\alpha$ -ийн утгын хувьд хоорондоо хол зөрүүгүй буюу GN-тэй дунджаар 79.86–83.54 % төстэй үр дүн гарсан. Хамгийн төстэй үр дүнг  $\alpha = 0.1$  үед үзүүлсэн бол  $\alpha$  ихсэхэд чанарын үр дүн бага зэрэг муудан, тогтвортой хэсэг яваад,  $\alpha = 2$  дээр огцом буурч байгаа үр дүн ажиглагдлаа. Уг ажиглалтыг Зураг 2-ээс харах боломжтой. Хугацааны хувьд  $\alpha = 0.1$  байх үед GN-ээс хэд дахин хурдан байсан үзүүлэлтийг 100 % гэж үзэн, уг үзүүлэлттэй хамааруулж хугацааны хувийг гаргасан.  $\alpha$  ихсэх тусам алгоритмын хурд нэмэгдэж байгаа учраас битүү шугам дээшээ үргэлж өссөн үзүүлэлттэй харагдаж байна. Харин шийдийн чанарын хувьд тасархай шугам буурч байгаа нь  $\alpha$  ихсэх тусам GN-тэй төстэй байдал алдагдаж байна.



*Зураг 2: GN-2B алгоритмын  $\alpha$  параметр ажиглах хугацаа болон, GN-тэй төстэй байдалд нөлөөлөх нөлөөлөл*

Хүснэгт 3-т GN-3B алгоритмын  $\alpha$ -ийн хамаарлыг туршсан үр дүнг харуулсан. Хүснэгтээс харахад  $\alpha$ -ын туршилтын утгуудад GN-3B алгоритм жижиг сүлжээнүүд дээр GN-ээс удаан байна. Өө

Хүснэгт 2: GN-2B алгоритмын  $\alpha$  параметрийн утгын туршилт. GN алгоритмын ажилласан хугацаа ( $t_1$ ), GN-2B алгоритмын ажилласан хугацаа ( $t_2$ ), GN-3B алгоритмын үүсгэсэн 2 бүлгийг GN алгоритмын үүсгэсэн 2 бүлэгтэй хэр төстэй хувь (төс)

Сүлжээ	$\alpha = 0.1$				$\alpha = 0.5$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	0.056	0.96	100	0.043	1.26	93.25
Polbooks	1.385	0.859	1.61	100	0.29	4.78	91.95
Power	2030.53	493.33	4.12	66.75	410.32	4.95	66.75
Facebook	4798.75	1572.75	3.05	67.4	549.7	8.73	67.5
<b>Дундаж</b>	1707.68	516.75	2.44	83.54	240.09	4.93	79.86
$\alpha = 1$							
Karate club	0.054	0.019	2.84	100	0.019	2.84	100
Polbooks	1.385	0.18	7.69	91.95	0.19	7.29	91.95
Power	2030.53	337.74	6.01	66.05	343.03	5.92	66.05
Facebook	4798.75	569.06	8.43	67.7	550.66	8.71	67.5
<b>Дундаж</b>	1707.68	226.75	6.25	81.43	223.47	6.19	81.38
$\alpha = 1.5$							
Karate club	0.054	0.019	2.84	100	0.019	2.84	100
Polbooks	1.385	0.18	7.69	91.95	0.19	7.29	91.95
Power	2030.53	337.74	6.01	66.05	343.03	5.92	66.05
Facebook	4798.75	569.06	8.43	67.7	550.66	8.71	67.5
<b>Дундаж</b>	1707.68	226.75	6.25	81.43	223.47	6.19	81.38

рөөр хэлбэл жижиг сүлжээний хувьд GN-3B-ийн техник хүнддэж байна гэсэн үг. Харин том сүлжээний хувьд  $\alpha$  томрох тусам алгоритмын хурд нэмэгдэж байна. Шийдийн GN-тэй төстэй байдлын хувьд жижиг сүлжээнүүд дээр  $\alpha$ -ийн бүх утгуудад 90-ээс дээш хувьтай бол том хоёр сүлжээнүүдийн хувьд  $\alpha = 0.1$  үед харгалзан 66, 99 % -тай төс харуулсан ч,  $\alpha$ -ийн бусад утгууд дээр энэ үзүүлэлт 66, 67 болж тогтворжиж байна.

Эдгээр  $\alpha$  параметрийн туршилтаар хугацаа болон шийдийн чанарын аль алийг нь харгалзан үзэж  $\alpha = 1$  байх үед GN-2B алгоритм хурдан бөгөөд чанартай шийдийг гаргаж байна. GN-3B алгоритмын хувьд хугацаа болон шийдийн чанарыг аль алийг нь харгалзан  $\alpha = 1.5$  хамгийн тохиромжтой байгаа бөгөөд удаах нь  $\alpha = 1$  байна. Харин бид алгоритмын үр дүнг харьцуулахдаа нэгэн жигд параметрээр ажиллуулах үүднээс бүх алгоритмуудын хувьд  $\alpha = 1$  утгаар сонгон авсан.

### 3.2.2 Ойролцооллын алдааны параметр

Дэвшүүлж буюу GN-3A, GN-3B алгоритмууд нь ХТ-ийг ойролцоогоор бодохдоо алдааны зөвшөөрөгдөх хязгаарыг илэрхийлэх  $\epsilon$  параметрээс хамаарна.  $\epsilon$  утгыг дөрвөн сүлжээн дээр 0.01, 0.03, 0.05, 0.07 гэсэн утгаар сонгон авч ажиллуулан, ажиллах хугацаа, давхцалын хувь (X%-ийн төс)-ийг Хүснэгт 4, 5-д харуулав. Нэгэн зэрэг устгах ирмэгийн тоог  $\alpha = 1$  утгаар сонгон авсан. Ойролцооллын алдаа буюу  $\epsilon$ -ийг томоор сонгох үед ХТ-ыг илүү хурдан бодох бөгөөд ингэснээр алгоритмын хурд нэмэгдэнэ. Тиймээс уг туршилтад  $\epsilon$  томроход алгоритм хурдан ажиллах нь илэрхий бөгөөд сонирхолтой үр дүн нь алгоритмын шийдийн төстэй чанар юм.

Хоёр алгоритмын хувьд шийдийн GN-тэй төстэй чанар нь  $\epsilon$ -ийн бүх утгуудад огцом өөрчлөлтгүй байна. GN-3A алгоритмын хувьд  $\epsilon$  томроход шийдийн төс 2, 1 гэсэн хувиар буурч байгаа үр дүн ажиглагдсан. Харин GN-3A алгоритмын хувьд чанар ихсэж

байснаа буурах хандлага ажиглагдаж байна. Тухайлбал  $\epsilon$ -ий 0.01, 0.03, 0.05, 0, 07 утгуудад шийдийн төс харгалзан дунджаар 80.51, 80.6, 81.39, 79.95 % -тай байна. GN-3A алгоритмын хувьд хамгийн төстэй чанарыг  $\epsilon = 0.01$  үед үзүүлсэн бол GN-3B алгоритмын хувьд  $\epsilon = 0.05$  үед байна. Иймд GN-3A, GN-3B алгоритмын ажиллах хурд, GN алгоритмын үр дүнтэй давхцаж буй хувь зэргийг харгалзан үзээд ойролцоолох параметрийн утгыг  $\epsilon = 0.05$  байхаар сонгон авлаа.

### 3.3 Харьцуулалт

GN-2A, GN-2B, GN-3A, GN-3B алгоритмуудыг өмнөх бүлэгт сонгосон оновчтой параметрийн утгаар ажиллуулж тус бүрээр 2 бүлэг үүсгэн тэдгээрийг GN алгоритмын илрүүлсэн 2 бүлэгтэй харьцуулсан үр дүнг Хүснэгт 6-д харуулав. Facebook сүлжээнээс 2 бүлэг илрүүлэхэд GN алгоритм ашиглавал 4, 798.75 секунд хугацаа зарцуулна. GN-2A алгоритм ашиглавал 8.79 дахин бага хугацаанд 66.37-ийн төстэй үр дүн, GN-2B алгоритм ашиглавал 8.75 дахин бага хугацаанд 67.52% -ийн үр дүн, GN-3A алгоритм ашиглавал 33.8 дахин бага хугацаанд 66.86% -ийн үр дүн, GN-3B алгоритм ашиглавал 36.74 дахин бага хугацаанд 67.52% -ийн төстэй үр дүнг тус тус авч болохыг хүснэгт харуулж байна. Хүснэгтээс харахад хамгийн сайн шийдийг GN-2B харуулсан нь дунджаар 171.01 секундэд 83.13% -тай, хамгийн богино хугацааг GN-3A харуулсан нь дунджаар 40.5 секундэд 74.09% -тай шийдийг гаргасан. GN-3A бусдаасаа хамгийн хурдан нь ч шийдийн чанарын хувьд бусдаасаа бага байна. Харин GN-3B алгоритмын хурд нь GN-3A-ийн дараа боловч шийдийн чанарын хувьд GN-2B-ийн дараа орж байна. Karate club, Dolphin, Polbooks зэрэг жижиг сүлжээний хувьд GN хурдан ажиллаж байна. GN алгоритм жижиг сүлжээтэй ажиллах үед асуудал үүсэхгүй. Том сүлжээний анализ хийх үед хугацааны өндөр өртөгтэй байдаг. GN-2A, GN-2B жижиг сүлжээнүүд дээр ч

Хүснэгт 3: GN-3B алгоритмын  $\alpha$  параметрийн туршилт. GN алгоритмын ажилласан хугацаа( $t_1$ ), GN-3B алгоритмын ажилласан хугацаа( $t_2$ ), GN-3B алгоритмын үүсгэсэн 2 бүлгийг GN алгоритмын үүсгэсэн 2 бүлэгтэй хэр төстэй хувь (төс)

Сүлжээ	$\alpha = 0.1$				$\alpha = 0.5$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	2.474	0.02	93.25	1.6	0.03	100
Polbooks	1.385	12.53	0.11	100	4.08	0.34	91.95
Power	2030.53	277.14	7.33	66.75	144.34	14.07	66.75
Facebook	4798.75	256.82	18.69	99.8	206.2	23.27	67.5
<b>Дундаж</b>	<b>1707.67</b>	<b>137.24</b>	<b>6.54</b>	<b>89.95</b>	<b>89.055</b>	<b>9.43</b>	<b>81.55</b>
Сүлжээ	$\alpha = 1$				$\alpha = 1.5$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	0.979	0.06	93.25	0.82	0.07	100
Polbooks	1.385	2.25	0.62	91.95	3.04	0.46	95.85
Power	2030.53	85.82	23.66	66.55	61.58	32.97	66.1
Facebook	4798.75	164.35	29.20	67.5	139.67	34.36	67.5
<b>Дундаж</b>	<b>1707.67</b>	<b>63.34</b>	<b>13.38</b>	<b>79.81</b>	<b>51.27</b>	<b>16.96</b>	<b>82.36</b>

Хүснэгт 4: GN-3A алгоритмын  $\epsilon$  параметрийн утгын туршилт. GN алгоритмын ажилласан хугацаа ( $t_1$ ), GN-2B алгоритмын ажилласан хугацаа( $t_2$ ), GN-3B алгоритмын үүсгэсэн 2 бүлгийг GN алгоритмын үүсгэсэн 2 бүлэгтэй хэр төстэй хувь (төс)

Сүлжээ	$\epsilon = 0.01$				$\epsilon = 0.03$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	8.47	0.01	55.1	1.01	0.05	56.36
Polbooks	1.385	35.9	0.04	90.1	4.11	0.34	91.95
Power	2030.53	989.18	2.05	66.5	109.7	18.51	66.45
Facebook	4798.75	3654.58	1.31	66.85	197.25	24.33	53.475
<b>Дундаж</b>	<b>1707.68</b>	<b>1172.03</b>	<b>0.85</b>	<b>69.64</b>	<b>78.02</b>	<b>10.81</b>	<b>67.06</b>
Сүлжээ	$\epsilon = 0.05$				$\epsilon = 0.07$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	0.32	0.17	56.35	0.18	0.30	55.1
Polbooks	1.385	1.42	0.98	91.45	0.73	1.90	92.45
Power	2030.53	42.2	48.12	66.45	21.1	96.23	66.45
Facebook	4798.75	610.04	7.87	53.05	77.99	61.53	53.05
<b>Дундаж</b>	<b>1707.68</b>	<b>163.50</b>	<b>14.28</b>	<b>66.83</b>	<b>25.00</b>	<b>39.99</b>	<b>66.76</b>

том сүлжээнүүд дээр ч GN-ээс хурдан байна. Шийдийн чанарын хувьд GN-2B илүү бөгөөд хугацааны хувьд GN-2A-аас дундаж нь 5 секундийн зөрөөтэй байна. Бидний судалгааны зорилго GN алгоритм ашиглан том сүлжээний анализ хийхэд шаардагдаж байгаа хугацааг багасгах бөгөөд GN-2A, GN-2B, GN-3A, GN-3B алгоритмууд бүгд хугацааны хувьд том сүлжээн дээр GN-ээс 8-33 дахин хурдан ажиллаж байна. Шийдийн чанарын хувьд GN-тэй дунджаар 83 % ойролцоо байна.

Дэвшүүлж буй алгоритмууд болон GN илрүүлсэн 4 бүлгийг хэмжээгээр нь харьцуулж Хүснэгт 7-д харуулав. Хүснэгтээс харахад GN-3B-ийн илрүүлсэн бүлгүүдийн хэмжээ нь хоорондоо хамгийн ойр байгаа нь GN-3B-ийн илрүүлсэн бүлэг GN-ий илрүүлсэн бүлгээс чанартай болохыг харж байна. GN-3B алгоритм GN-ээс 1.7 дахин бага хугацаа зарцуулсан байна. GN-3A алгоритм хугацаа хамгийн бага, GN-2B алгоритм 3 дахин бага хугацаа зарцуулж GN алгоритмтай ойролцоо үр дүн харуулсан байна. GN-2A, GN-2B, GN-3A алгоритмуудын илрүүлсэн хамгийн

том бүлгийн хэмжээ нь GN-ий илрүүлсэн хамгийн том бүлгийн хэмжээтэй ойролцоо байгаа нь эдгээр алгоритмууд төстэй бүлэг илрүүлснийг илтгэж байна.

GN-2B, GN-3B зэрэг алгоритм нь үүссэн бүлгийн оройн тоог шалгаж хэрэглэгчийн тодорхойлсон оройн тооноос цөөн оройтой бүлэг үүсгэсэн тохиолдолд устгасан ирмэгээ буцаан нэмж байгаа учир үндсэн алгоритмтай ойролцоо үр дүн үзүүлж байна. Гүйцэтгэх хугацаа нь багассан байгаа нь судалгааны зорилгод нийцсэн сайн үр дүн гарлаа.

Сүлжээний бүлгүүд хоорондоо ихэвчлэн олон тооны ирмэгүүдээр холбогдсон байдаг. Эдгээр ирмэгүүдийг хот доторх гүүртэй адилтгаж төсөөлж болох юм. Нэг гүүр хэд хэдэн эгнээ зам (ирмэг)-аас тогтож болно. Тухайн нэг гүүрний нэг эгнээг хаах (нэг ирмэг устгах) нь тэр гүүрний бусад эгнээний ачаалал, урсгалыг нэмэгдүүлнэ (бусад ирмэгийн ХТ-ийн оноог нэмэгдүүлнэ). Иймд нэг ирмэгийн устгал нь түүний хөрш ирмэгийг өндөр ХТ оноо авах шалтгаан болно. ХТ-ийн тооцооллоос нэгэн зэрэг олон

Хүснэгт 5: GN-3B алгоритмын  $\epsilon$  параметрийн утгын туршилт. GN алгоритмын ажигласан хугацаа ( $t_1$ ), GN-2B алгоритмын ажигласан хугацаа ( $t_2$ ), GN-3B алгоритмын үүсгэсэн 2 бүлгийг GN алгоритмын үүсгэсэн 2 бүлэгтэй хэр төстэй хувь (төс)

Сүлжээ	$\epsilon = 0.01$				$\epsilon = 0.03$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	11.81	0.00	96.55	1.31	0.04	96.9
Polbooks	1.385	36.5	0.04	91.95	3.96	0.35	91.95
Power	2030.53	1518.28	1.34	66.05	173.4	11.71	66.05
Facebook	4798.75	2672.1	1.80	67.5	299.05	16.05	67.5
<b>Дундаж</b>	1707.68	1059.67	0.79	80.51	119.43	7.04	80.60
Сүлжээ	$\epsilon = 0.05$				$\epsilon = 0.07$		
	$t_1$ (сек)	$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate club	0.054	0.48	0.11	100	0.26	0.21	93.25
Polbooks	1.385	1.42	0.98	91.95	0.74	1.87	92.95
Power	2030.53	63.4	32.03	66.1	32	63.45	66.1
Facebook	4798.75	239.78	20.01	67.5	110.74	43.33	67.5
<b>Дундаж</b>	1707.68	76.27	13.28	81.39	35.94	27.22	79.95

Хүснэгт 6: GN-2A, GN-2B, GN-3A, GN-3B алгоритмыг  $\alpha = 1, \epsilon = 0.05$  параметртэй ажиглуулсан үр дүн. GN-ы ажигласан хугацаа ( $t_1$ ), дэвшүүлж буй алгоритмын ажигласан хугацаа ( $t_2$ ), илрүүлсэн бүлэг GN-ийнхээ хэр төстэй байгаа хувь (төс)

Сүлжээ	$t_1$ (сек)	GN-2A			GN-2B		
		$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate	0.05	0.017	3.18	83.23	0.019	2.84	89.99
dolphin	0.10	0.045	2.16	76.37	0.18	0.54	100
Polbooks	1.38	0.19	7.26	88.31	0.19	7.26	92.04
Power	2030.53	284.06	7.15	65.43	306.38	6.63	66.12
Facebook	4798.75	546.2	8.79	66.37	548.29	8.75	67.52
<b>Дундаж</b>	1366.16	166.10	5.71	75.95	171.01	5.20	83.13
Сүлжээ	$t_1$ (сек)	GN-3A			GN-3B		
		$t_2$ (сек)	$t_1/t_2$	төс(%)	$t_2$ (сек)	$t_1/t_2$	төс(%)
Karate	0.05	0.02	3.18	83.24	16.34	0.00	89.99
dolphine	0.10	0.04	2.16	76.38	16.34	0.01	100
Polbooks	1.38	1.44	0.96	90.95	1.44	0.96	91.05
Power	2030.53	58.99	34.42	53.06	63.62	31.92	66.13
Facebook	4798.75	141.99	33.80	66.86	130.6	36.74	67.52
<b>Дундаж</b>	1366.16	40.50	14.90	74.09	45.67	13.93	82.94

ирмэг устгах нь нэг нэгээр устгахтай төстэй үр дүнд хүрч байгааг тайлбарлах нэг үндэслэл байх талтай.

## 4 Дүгнэлт

Сүлжээнээс бүлэг илрүүлэх стандарт аргуудын нэг Girvan-Newman алгоритмыг хурдыг сайжруулах хоёр санааг дэвшүүлж, хэрэгжүүлэв. Алгоритмын нэг давталтад хоорондын төвийн тооцооллоос  $\sqrt{m}$  ирмэг устгасан нь давталтын тоог  $m$ -ээс  $\sqrt{m}$  болгон багасгав. Хоёрдугаарт хоорондын төвийн ойролцоо оноог ашиглах нь топ-к оройг тодорхойлоход хангалттай байна. Энэ хоёр санааг тусгасан GN алгоритмын өргөтгөл алгоритм том сүлжээний хувьд олон дахин хурдан ажиллав. Илрүүлсэн бүлэг нь үндсэн алгоритмын үр дүнтэй 75% дээш давхцалтай байв.

## Зохиогчийн оролцоо

Уг судалгааны дизайныг П.Далайжаргал гаргаж, П.Батцэнгэл алгоритмын хэрэгжүүлэлт, туршилтыг гүйцэтгэв. Туршилтын үр дүнг нэгтгэх, дүгнэх, өгүүллийн бичилтийг бүх зохиогчид хамтран гүйцэтгэв.

## Санхүүжилт

Энэхүү судалгааны ажил нь Монгол Улсын Их Сургуулийн өндөр түвшний судалгааны төслийн (P2020-3978) санхүүжилтээр хэрэгжсэн судалгааны ажлын нэг хэсэг болно.

Хүснэгт 7: Facebook өгөгдлөөс илрүүлсэн 4 бүлгийн хэмжээ ( $\alpha = 1, \epsilon = 0.05$ )

Алгоритм	GN(сек)	GN-2A(сек)	GN-2B(сек)	GN-3A(сек)	GN-3B(сек)
Ажилласан хугацаа	25,462.91	1,004.06	8,743.95	278.83	15,202.36
Үүсгэсэн бүлгийн тоо болон бүлэг бүрийн оройн тоо	1-р бүлэг 1092, 2-р бүлэг 2193, 3-р бүлэг 548, 4-р бүлэг 20 оройтой	1-р бүлэг 2091, 2-р бүлэг 745, 3-р бүлэг 531, 4-р бүлэг 326 5-р бүлэг 199, 6-р бүлэг 59 оройтой. Үүнээс гадна 1 оройтой 79 бүлэг үүсгэсэн	1-р бүлэг 343, 2-р бүлэг 2131, 3-р бүлэг 753, 4-р бүлэг 753 5-р бүлэг 59 оройтой.	1-р бүлэг 2091, 2-р бүлэг 745, 3-р бүлэг 531, 4-р бүлэг 326 5-р бүлэг 199, 6-р бүлэг 59 оройтой. Үүнээс гадна 1 оройтой 79 бүлэг үүсгэсэн	1-р бүлэг 341, 2-р бүлэг 1348, 3-р бүлэг 752, 4-р бүлэг 779 5-р бүлэг 547, 6-р бүлэг 206 оройтой. Үүнээс гадна 1 оройтой 3 бүлэг үүсгэсэн

## Ашиг сонирхлын зөрчилгүйн баталгаа

Ашиг сонирхлын зөрчилгүй болохыг баталж байна.

## Ашигласан ном

- [1] Girvan M, Newman ME. Community structure in social and biological networks. Proceedings of the national academy of sciences. 2002;99(12):7821-6.
- [2] Fortunato S, Newman ME. 20 years of network community detection. Nature Physics. 2022;18(8):848-50.
- [3] Yang J, McAuley J, Leskovec J. Community detection in networks with node attributes. In: 2013 IEEE 13th international conference on data mining. IEEE; 2013. p. 1151-6.
- [4] Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proceedings of the national academy of sciences. 2004;101(9):2658-63.
- [5] Rossi R, Ahmed N. The network data repository with interactive graph analytics and visualization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29; 2015. .
- [6] Freeman LC. A set of measures of centrality based on betweenness. Sociometry. 1977:35-41.
- [7] Tsalouchidou I, Baeza-Yates R, Bonchi F, Liao K, Sellis T. Temporal betweenness centrality in dynamic graphs. International Journal of Data Science and Analytics. 2020;9:257-72.
- [8] Brandes U. A faster algorithm for betweenness centrality. Journal of mathematical sociology. 2001;25(2):163-77.
- [9] Kong B, Zhou L, Liu W. Improved modularity based on Girvan-Newman modularity. In: 2012 Second International Conference on Intelligent System Design and Engineering Application. IEEE; 2012. p. 293-6.
- [10] Despalatović L, Vojković T, Vukicević D. Community structure in networks: Girvan-Newman algorithm improvement. In: 2014 37th international convention on information and communication technology, electronics and microelectronics (MIPRO). IEEE; 2014. p. 997-1002.
- [11] Devi S, Rajalakshmi M. Community Detection by Node Betweenness Using Optimized Girvan-Newman Cuckoo Search Algorithm. Information Technology and Control. 2023;52(1):53-67.
- [12] Riondato M, Kornaropoulos EM. Fast approximation of betweenness centrality through sampling. In: Proceedings of the 7th ACM international conference on Web search and data mining; 2014. p. 413-22.
- [13] Riondato M, Upfal E. Abra: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. ACM Transactions on Knowledge Discovery from Data (TKDD). 2018;12(5):1-38.
- [14] Leskovec J, Mcauley J. Learning to discover social circles in ego networks. Advances in neural information processing systems. 2012;25.