

Компьютерын ухаан

СҮЛЖЭЭГ ЗАДЛАХ БОДЛОГОД ЗОРИУЛСАН ДАХИН ОРУУЛАХ ТЕХНИКИЙГ САЙЖРУУЛАХ

Г.Гантулга, П.Далайжаргал*

МУИС, ХШУИС, Мэдээлэл, Компьютерийн Ухааны Тэнхим

Received on 2021.04.15; Revised on 2021.06.24; Accepted on 2021.07.01

* Холбоо баригч зохиогч: dalaijargal@seas.num.edu.mn

Хураангуй

Сүлжээнээс хасахад сүлжээ өөрөө жижиг хэмжээтэй холбоост бүрдлүүд болгон задардаг байхаар хамгийн цөөн оройг олохыг Сүлжээг Задлах Бодлого (Network Dismantling Problem) гэнэ. Уг бодлого нь нийгэм, биологи, технологийн сүлжээнүүдийн бат бөх чанарыг судлахаас гадна, дутагдалтай бүтцийг судлахад ач холбогдолтой. Сүүлийн жилүүдэд дахин оруулах аргыг сүлжээг задлах бодлогод ашиглах судалгаа хийгдэж эхлээд байна. Гэхдээ дахин оруулах аргын судалгаа дутмаг байна. Энэхүү ажлаар бид нэгэн шинэ дахин оруулах аргыг танилцууллаа. Дэвшүүлсэн аргаа нийгэм, биологи, технологийн төрлийн 13-н сүлжээний өгөгдлийг ашиглан туршив. Туршилтын үр дүнгээс харахад дэвшүүлсэн арга ихэнх сүлжээний хувьд бусад аргаас давуу үр дүн үзүүлээ.

Түлхүүр үг: Комплекс сүлжээ, сүлжээг задлах бодлого, дахин оруулах техник

1 Удиртгал

Агаарын тээвэр [1], эрчим хүчний сүлжээ [2], Интернет [3], биологийн сүлжээ [4, 5], нийгмийн сүлжээ [6, 7] зэрэг олон бодит систем (real-world system)-ийг комплекс сүлжээ (ирмэгээр холбогдсон зангилаануудын сүлжээ)-гээр төлөөлүүлж болно. Ихэнх тохиолдолд хамгийн том холбоост бүрдэл нь системийн үйл ажиллагааг хэвийн биелүүлэхэд голлох үүрэгтэй оролцдог [8]. Довтолгооны оновчтой стратеги, мэдээллийн оновчтой тархалт, дархлаажуулалтын бодлогыг ойлгох хэрэгцээ шаардлагад тулгуурлан бид сүлжээг задлах асуудлыг судалж байна (өөрөөр хэлбэл цөөн тооны чухал зангилааны олонлогийг устгасан (идэвхгүй болгох)-аар сүлжээг жижиг холбоост бүрдлүүдэд задлах зангилааны хамгийн бага дэд олонлогийг хайж олох).

Сүлжээ (математикийн хэлээр G граф) нь N ширхэг зангилаа (эсвэл орой)-ны олонлог V , зарим хос зангилааг холбосон ирмэгийн олонлог E -ээс тогтоно. Сүлжээний судалгааны хүрээнд хэд хэдэн чухал асуултууд нь зангилааныхаа S дэд хэсгийг сонгож, тодорхой байдлаар авч үзэх үед графын шинж чанарыг өөрчлөхтэй холбоотой юм. Жишээлбэл, S -ийн оройг (зэргэлдээ ирмэгийн хамт) хасвал графын хамгийн том холбоост бүрдэл хэсэг (giant connected component (gcc))-ийн хэмжээ хэр зэрэг буурдаг вэ? Цар тахлын халдварын үед анх S дэх зангилаанууд бохирдсон бол тахлын тархалт ямар үр дүнд хүрэх вэ? Эсрэгээр, S -ийн зангилааны вакцинжуулалт нь сүлжээг бүтнээр нь вакцинжуулахад хэр нөлөөтэй вэ? Зэрэг асуултын хариулт нь сүлжээг задлах бодлоготой нягт холбоотой.

Collective influencer [9], сүлжээнээс бүх циклийг

устгаад, үлдэх мод графыг задлах арга [8], тасралтын цэг (articulation point)-ээр сүлжээг задлах арга [10], спектр графын онол (spectral graph theory)-д суурилсан арга [11] зэрэг сүлжээг задлах бодлогыг бодох олон аргуудыг дэвшүүлээд байна. Сүлжээг задлах аргуудын тухай дэлгэрэнгүй мэдээллийг [12]-оос авах боломжтой. Сүлжээг задлах аргыг сайжруулах зорилгоор дахин оруулах техник [13] (re-insertion technique) сүүлийн жилүүдэд судлагдаж эхлээд байгаа хэдий ч нарийвчилсан шинжилгээ дутмаг байна.

Энэхүү өгүүлэлд сүлжээг задлах аргыг сайжруулах шинэ дахин оруулах аргыг танилцуулах ба үр дүнгээ “сэгсрэх” гэж нэрлэсэн техник ашиглан илүү сайжруулах боломжийг танилцуулах болно. Дэвшүүлсэн аргаа нийгэм, биологи, технологийн төрлийн 13-н сүлжээний өгөгдлийг ашиглан туршихад, 10-11 сүлжээг бусад аргаас илүү үр дүнтэй задалсан байна.

2 Дэвшүүлж буй арга

2.1 Дахин оруулах тэнцвэрт арга

Бидний дэвшүүлсэн арга нь тухайн оройг нэмсний дараа граф дахь бүрдлүүдийн хэмжээний квадрат нийлбэр буюу Тэгшитгэл 1-ыг аль болохоор бага байлгахыг зорьдог. Тиймээс үүсгэгдсэн бүрдлүүдийн хэмжээг аль болохоор бага байлгана. Мөн нэг бүрдлийг хэтэрхий том болгохгүй, бүрдлүүдийн хэмжээг тэнцвэртэйгээр нэмэгдүүлж өгөх тул бид энэхүү аргыг дахин оруулах тэнцвэр гарга гэж нэр-

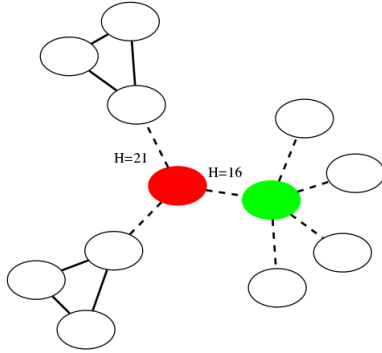
лэсэн.

$$H(u) = \sum_{C \in SCC(G')} \frac{|C|(|C| - 1)}{2} \quad (1)$$

Энд $|C|$ нь холбоост бүрдлийн хэмжээ, G' нь u оройг оруулсны дараа үүсэх граф, $SCC(G')$ нь G' графын холбоост бүрдлүүдийн олонлог болно. Тэгшитгэл 1 u оройг буцааж нэмсний дараах үүсгэгдсэн граф дахь бүрдлүүдийн хэмжээний квадрат нийлбэрүүдийг бодно. Устгасан бүх u оройны хувьд $H(u)$ (Тэгшитгэл 1) утга хамгийн бага байх u оройг эргүүлэн графт нэмнэ.

$$u = \arg \min_{v \in S} H(v) \quad (2)$$

Нэмэхдээ нэмж буй оройны, графт аль хэдийн байгаа оройтой холбогдох ирмэгүүдийг дахин сэргээнэ. Жишээ нь Зураг 1-д үзүүлсэн графт улаан, ногоон оройг устгасан ба тэдгээрийн ирмэгүүд нь мөн устгагдсан тул тасархай шугамаар дүрсэлсэн. $H = 21$ байх оройг оруулах юм бол хэт том бүрдэл хэсэг үүсэх бөгөөд бусад бүрдлүүд нь хэт жижиг хэвээр байна. Харин $H = 16$ байх оройг оруулбал үүсэх бүрдэл хэсгүүдийн хэмжээ ойролцоо байх бөгөөд тэнцвэртэйгээр оройнуудыг буцаан оруулж байна.



Зураг 1: Бүдсэн оройнууд нь графаас устгагдсан оройнууд, тасархай шугамаар тэмдэглэгдсэн ирмэгүүд нь тэдгээрийн устгагдсан ирмэг. Зурагт устгагдсан оройнуудын H функццийн (Тэгшитгэл 1) утгыг харуулав.

Дэвшүүлж буй дахин оруулах аргын баримжаа кодыг Алгоритм 1-д харуулав.

Algorithm 1 Дахин оруулах тэнцвэрт алгоритм

REINSERTION-BL(G, S)

- 1: **while** $S \neq \emptyset$
 - 2: $u = \arg \min_{i \in S} H(i)$
 - 3: u -г G графт буцааж нэм.
 - 4: $S = S/\{u\}$
 - 5: Одоогийн S -ын хувьд G графын хамгийн том холбоост бүрдлийг хэмжинэ.
-

2.2 Сэгсрэх техник

Графын хамгийн том бүрдлийг жижиглэн хуваасны дараа устгасан оройнуудыг нэг, нэгээр нь нэмж

орүүлж эхэлнэ. Сэгсрэх техник нь s ширхэг оройг оруулсны дараа b ($s > b$) ширхэг оройг эргүүлэн хасах арга юм. Оройг дахин хасахдаа өмнө нь ашигласан хасах аргаа хэрэглэнэ. Ингэснээр үр дүнг муутгасан зарим нэг оройг хасаж, оруулах оройнуудыг шигших боломж бүрдэж байгаа юм. Бидний дэвшүүлж буй оруулах аргыг, сэгсрэх техниктэй хослуулсан алгоритмыг Алгоритм 2-д харуулна.

Algorithm 2 Сэгсрэх техниктэй, дахин оруулах тэнцвэрт алгоритм

REINSERTION-BLS(G, S, s, b)

- 1: $step = 0$
 - 2: **while** $S \neq \emptyset$
 - 3: $step = step + 1$
 - 4: $u = \arg \min_{i \in S} H(i)$
 - 5: u -г G графт буцааж нэм.
 - 6: $S = S/\{u\}$
 - 7: Одоогийн S -ын хувьд G графын хамгийн том холбоост бүрдлийг хэмжинэ.
 - 8: **if** $step == s$
 - 9: $step = 0$
 - 10: G -с b ширхэг оройг буцааж хасаад, S -д оруулна.
-

Сэгсрэх арга S -ийн хэмжээг ихэсгэж, багасгаж байгаа учраас утгасан оройн тооны нэг утга дээр өөр өөр хамгийн том бүхэл хэсгийн хэмжээ бүртгэгдэх бөгөөд давхацсан утгуудын хувьд хамгийн багийг нь авсан.

3 Туршилт, үр дүн

3.1 Өгөгдөл

Бид туршилтдаа бодит ертөнцийн өгөгдлүүдийг ашигласан [14]. Зарим сонирхолтой сүлжээний талаар товчхон оруулав.

humanDiseaseome: Хүний өвчний сүлжээний хувьд оройнууд нь өвчнүүд ба хоёр өвчин ямар нэг ижил гений өөрчлөлтөөр үүсгэгддэг бол хооронд нь ирмэг татсан [15].

powergrid: Америкийн баруун мужуудад орших өндөр хүчдэлийн холболтын сүлжээ. Сүлжээний оройнууд нь генератор, хөрвүүлэгч, жижиг станцууд ба ирмэгүүд нь тэдгээрийн хоорондох өндөр хүчдэлийн шугамууд.

facebook: Фэйсбүүкийн найзуудын жагсаалтыг агуулна. Тусгай ашын тусламжтай судалгаанд оролцсон хүмүүсээс цуглуулж авсан. Өгөгдлийн оройнууд нь хэрэглэгч, circles, ego сүлжээнүүд байна. Өгөгдөлд хэрэглэгчийг нуухын тулд хэрэглэгчийн ID-ыг шинээр сольж тавьсан. Мөн өгөгдөлд байгаа онцлог шинжийн вектор тусгагдсан ч гэсэн тэдний утгыг нь далдлагдсан [16].

grqc, hepth, heppl, astroph, condmat: Arxiv дээр төрөл бүрийн салбарт хамтран ажилладаг эрдэмтдийн хамтын ажиллагааны сүлжээг агуулна. Хэрэв өгүүлэл хэвлүүлж буй зохиогч i өөр нэг j

Хүснэгт 1: Туршилтад ашиглах сүлжээнүүдийн товч мэдээлэл, энд V нь зангилааны тоо, E нь ирмэгийн тоо, $\langle d \rangle$ нь дундаж зэрэг, H нь бүрдэл хэсгийн тоо, C нь кластерын коэффициент.

Networks	V	E	$\langle d \rangle$	H	C
HumanDis	516	1,188	4.60	1	0.43
Tren_Rom	255	272	2.13	1	0.02
EU_flights	1,191	31,610	53.08	2	0.40
openflights	1,858	13,900	14.96	371	0.33
yeast1	2,018	2,705	2.68	185	0.02
powergrid	4,941	6,594	2.66	1	0.10
OClinks	1,899	13,838	14.57	4	0.06
facebook	4,039	88,234	43.69	1	0.52
grqc	5,242	14,484	5.52	355	0.63
hepth	9,877	25,973	5.25	429	0.28
hepph	12,008	118,489	19.73	278	0.65
astroph	18,772	198,050	21.10	290	0.32
condmat	23,133	93,439	8.07	567	0.26

хүнтэй хамтарч бичсэн бол граф i -с j чиглэлгүй ирмэгийг агуулах болно. Хэрэв нэг өгүүлэл k зохиогчтой бол k оройтой бүтэн дэд граф үүсгэнэ [16].

3.2 Туршилтын арга зүй

Дахин оруулах аргыг харьцуулахдаа **Random**, **HDA**, **CI** гэсэн задлах аргаар графын хамгийн том бүрдэл хэсгийг хамгийн ихдээ 5 оройтой бүрдэл хэсгүүдэд хуваан оройнуудыг устгасан.

- **Random**: Графаас санамсаргүйгээр нэг оройг сонгон устгах зарчмаар сүлжээг жижиглэн хуваасан.
- **HDA**: High Degree Adaptive арга нь хамгийн өндөр зэрэгтэй оройг устгах зарчмаар графыг жижиглэн хуваана. Нэг оройг устгасны дараагаар үлдсэн оройнуудын зэргийг шинэчилнэ.
- **CI**: Collective Influencer¹ [17] аргыг ашиглан хамгийн өндөр CI оноотой оройг хасах зарчмаар графыг жижиглэн хуваана. Нэг оройг хасаны дараа үлдсэн оройнуудыг CI оноог дахин тооцно.

Графыг задлах алгоритм бүр дээр дахин оруулах аргаа туршиж үзсэн. Ингэхдээ Хүснэгт 1 харуулсан бодит сүлжээнүүд дээр туршиж ажиллуулсан.

Дэвшүүлж буй аргаа [13]-д дурдагдсан дараах гурван дахин оруулах аргатай харьцуулсан.

- **Re-insertion I (R1)**: Нэмэгдэж буй орой хамгийн цөөн тооны бүрдэл хэсгүүдийг хооронд нь холбож байхаар оройг сонгоно.
- **Re-insertion II (R2)**: Нэмэгдэж буй оройн холбон үүсгэсэн бүрдэл хэсгийн хэмжээ хамгийн бага байхаар оройг сонгоно.

- **Re-insertion III (R3)**: Холбож буй бүрдэл хэсгийн тоо, үүсгэсэн бүрдэл хэсгийн хэмжээ хоёрын үржвэр хамгийн бага байхаар оройг сонгон оруулна. Re-insertion I, II-ын үржвэр гэсэн үг.

Өөрсдийн дэвшүүлсэн аргаа хоёр янзаар ажиллуулсан, нэг нь сэгсрэх техник ашиглаагүй (**R_BL**), нөгөө нь дэвшүүлсэн арга дээрээ сэгсрэх техник хэрэгжүүлсэн (**R_BLS**). Эдгээр таван дахин оруулах аргыг, гурван задлах аргатай хослуулан өгөгдөл дээрээ туршиж харуулсан болно.

Дахин оруулах аргыг хооронд нь харьцуулахдаа дараах хэмжигдэхүүнийг ашигласан.

$$R = \frac{1}{N} \sum_{Q=5}^N s(Q) \quad (3)$$

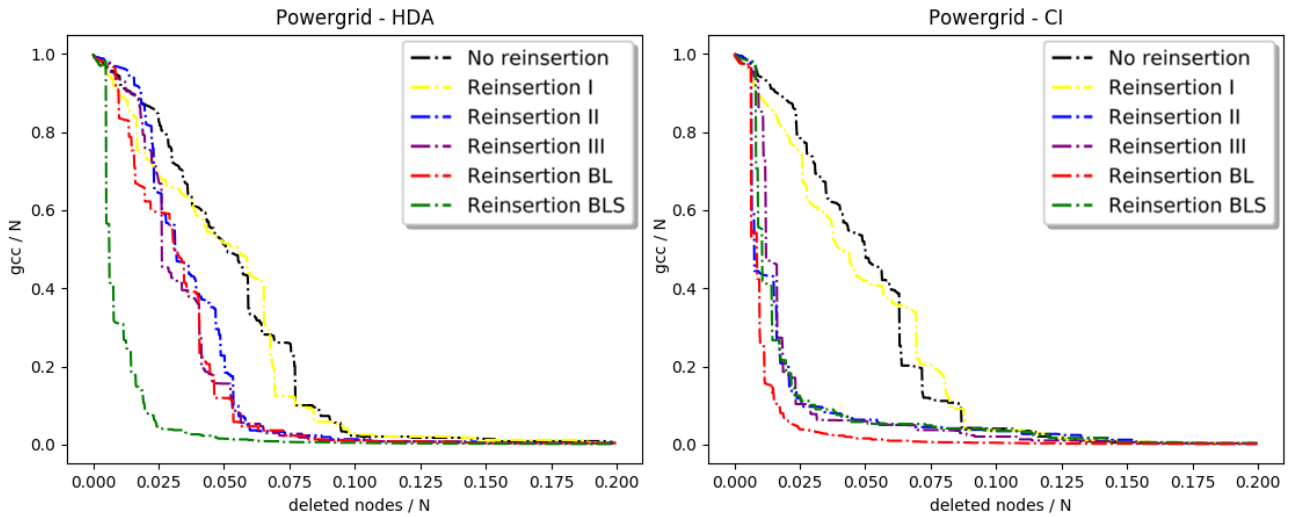
Энд $s(Q)$ нь Q ширхэг оройг буцааж нэмсний дараах хамгийн том бүрдэл хэсгийн хэмжээ. Устгасан оройнуудаас нэг нэгээр сонгон оруулахдаа хамгийн том бүрдэл хэсгийн хэмжээнүүдийн нийлбэрийн дунджийг Тэгшитгэл 3-аар олж байна. Энэ нь үндсэндээ Зураг 2-д харуулсан графикийн муруйн доорх хэсгийн талбайг бодож олж байгаа болно.

3.3 Дахин оруулах тэнцвэрт, сэгсрэх аргын үнэлгээ

Эхлээд бид дахин оруулах тэнцвэрт аргыг (**R_BL**), ямар ч дахин оруулах арга хэрэглээгүй задлах аргатай (**noR**) харьцуулсан. Энэ үр дүнг Хүснэгт 2-д тэгшитгэл 3-р бодон гарган харуулав. Сүлжээг задлах гурван арга тус бүрийн хувьд дахин оруулах тэнцвэрт аргыг хэрэглэсэн нь илүү давамгайл үр дүнг үзүүлж, нэг тохиолдлоос бусад бүх тохиолдолд дахин оруулах аргыг хэрэглэсэн нь илүү үр дүнтэй байна. Оройнуудыг санамсаргүй байдлаар устгасан үр дүн дээр дахин оруулах аргыг хэрэглэснээр дахин оруулах аргыг хэрэглээгүй ухаалаг алгоритмын үр дүнгээс илүү гарч байгаа нь энгийн дахин оруулах арга Сүлжээг Задлах Бодлогын үр дүнд ихээхэн нөлөөлж байгааг бататгаж байна [13]. HDA аргаар сүлжээг задлахад нөгөө хоёр аргаас ихэнх граф дээр илүү сайн задалж байгаа ч CI-ыг дахин оруулах тэнцвэрт аргатай хослуулахад HDA-ыг сайжруулснаас илүү давсан үзүүлэлт гаргасан. Иймд дахин оруулах тэнцвэрт арга нь CI-г илүү сайжруулж байгаа нь дундаж сайжруулалт 317.50-аар харагдаж байна. Дундаж сайжралтыг бодохдоо өгөгдөл тус бүрийн үр дүнгийн зөрүүний нийлбэрийг нийт өгөгдлийн тоонд хуваасан. Энд бодсон дундаж сайжралтын оноо нь задлах алгоритмаас ихээхэн хамаарах бөгөөд санамсаргүй аргын хувьд дундаж сайжралт хамгийн сайн байгаа нь задлах алгоритм анхнаасаа муу задласантай холбоотой.

Дараагаар нь дахин оруулах тэнцвэрт арга дээрээ сэгсрэх техникийг нэмсэн (**R_BLS**). Ингэхэд сэгсрэх техник нь дахин оруулах тэнцвэрт аргын үр дүнг илүү сайжруулсан. Хүснэгт 3-д сэгсрэх техникийг хэрэглэснээр дундаж сайжралт гурван задлах аргын хувьд ойролцоо байгаа бөгөөд сайжирсан

¹<https://github.com/zhfkt/ComplexCi>



Зураг 2: Powergrid өгөгдөл дээрх дахин оруулах аргуудын харьцуулалт. Хэвтээ тэнхлэг нь устгасан оройн тоог нийт оройн тоонд харьцуулсан хэмжээ. Босоо тэнхлэг нь хамгийн том бүрдлийн хэмжээг нийт оройн тоонд харьцуулсан хэмжээ.

үзүүлэлт Random-ын хувьд дунджаар 91.85, HDA-ын хувьд дунджаар 72.6, CI-ын хувьд дунджаар 85.19 байна. Random-ын хувьд дундаж сайжралт хамгийн их байгаа боловч үр дүнгийн хувьд бусдыгаа давж чадахгүй байна. Харин дахин оруулах тэнцвэрт аргатай HDA, CI-д өмнө нь ялагдаж байсан (Хүснэгт 2) бол сэгсрэх техниктэй, дахин оруулах тэнцвэрт аргыг хэрэглэснээр HDA-ын үр дүнг илүү сайжруулан 13 графын 8-д нь хамгийн сайн үр дүнг үзүүлсэн.

Туршилтаас үзэхэд R_BL нь CI аргыг илүү сайжруулж, R_BLS нь HDA аргын үр дүнг илүү сайжруулж байна. Хамгийн сайн үр дүнг HDA-г, сэгсрэх техниктэй дахин оруулах тэнцвэрт аргатай ашигласнаар гаргах боломжтой туршилт харуулж байна.

3.4 Бусад аргатай харьцуулсан үнэлгээ

Энэ хэсэгт дэвшүүлсэн сэгсрэх техниктэй дахин оруулах тэнцвэрт аргыг бусад дахин оруулах аргууд (R1, R2, R3)-тай харьцуулан үр дүнгийн үнэлгээг гүйцэтгэлээ. Ингэхэд туршилтын 13 өгөгдлийн хувьд R_BLS дахин оруулах арга нь Random-р задлах үед 11 удаа, HDA-р сүлжээг задлах үед 11 удаа, CI-р сүлжээг задлах үед 10 удаа тус тус ялсан үр дүн үзүүлсэн (Хүснэгт 4, 5, 6).

Хүснэгт 4, 5, 6 дахь тоон үр дүнгийн график дүрслэлийг зарим сүлжээний хувьд дүрслэв. powergrid өгөгдлийн хувьд таван дахин оруулах аргын үр дүнгийн графикийг Зураг 2-т харуулав. Хэвтээ тэнхлэгийн ихэнх хэсэгт уг тэнхлэгтэйгээ аль болох ойр байвал сайн үр дүнг үзүүлж байна гэж үзэж болно. Дахин оруулах арга ашиглаагүй (хар тасархай зураас) хувилбар болон дахин оруулах I (Re-insertion I [13]) аргууд хамгийн тааруу үр дүн үзүүлжээ. Ихэнх дахин оруулах аргууд (Re-insertion I -с бусад) сүлжээг илүү үр дүнтэйгээр задалж байна. Ямар ар-

гаар (HDA эсвэл CI) устгаснаас шалтгаалж дахин оруулах аргын үр нөлөө өөр байгаа нь харагдана. Тухайлбал, HDA устгах аргын (зүүн гар тал) хувьд сэгсрэх техниктэй дахин оруулах тэнцвэрт арга (Зураг 2-т ногоон тасархай зураасаар дүрсэлсэн) бусад аргаа илэрхий ялж байна. Харин CI (баруун гар тал) устгах аргын хувьд дахин оруулах тэнцвэрт арга (Зураг 2-т улаан өнгөөр дүрсэлсэн) бусад аргаа ялсан байна.

4 Дүгнэлт

Комплекс сүлжээг үр дүнтэйгээр задлах зорилготой дахин оруулах шинэ аргыг танилцууллаа. Дэвшүүлсэн аргаа нийгэм, биологи, технологийн төрлийн 13-н сүлжээний өгөгдлийг ашиглан туршиж, саяхан дэвшигдсэн бусад гурван аргатай үр дүнгээ харьцуулан үнэллээ. Бидний дэвшүүлсэн дахин оруулах тэнцвэрт аргыг сэгсрэх техник илүү сайжруулж байна. Туршилтын үр дүн HDA алгоритмаар сүлжээг задлаад, сэгсрэх техниктэй дахин оруулах тэнцвэрт аргыг хэрэглэснээр хамгийн сайн үр дүнг авах боломжтой бөгөөд бидний гаргасан үр дүн өмнө энэ салбарт хийгдсэн ажлын үр дүн илүү сайжруулсан байна. Дахин оруулах аргыг нэмснээр хамгийн өндөр зэргийг устгадаг энгийн алгоритм илүү нарийн ажилгаатай CI алгоритмаас сайн үр дүнг үзүүлж чадаж байна.

Зохиогчийн оролцоо

Уг судалгааг П.Далайжаргал дизайн хийж, алгоритмын хөгжүүлэлт, хэрэгжүүлэлт, туршилт Г.Гантулга гүйцэтгэлээ, өгүүллийн бичилтийг хамтран гүйцэтгэв.

Хүснэгт 2: Дахин оруулах тэнцвэрт аргыг ямар ч дахин оруулах аргыг хэрэглээгүй задлах аргатай харьцуулсан үр дүн. noR нь ямар ч дахин оруулах аргыг хэрэглээгүй үр дүн. R_{BL} нь дахин оруулах тэнцвэрт арга. Ялсан тоо нь нэг аргын хоёр баганыг харьцуулан тоолсон бол нийт ялсан тоо нь бүх баганыг харьцуулан тоолсон. Тухайн графын хувьд хамгийн бага үр дүнг тодоор харуулав.

Сүлжээ	Random		HDA		CI	
	noR	R _{BL}	noR	R _{BL}	noR	R _{BL}
humanDiseaseome	141.66	16.82	16.34	10.55	18.19	10.66
Treni_Roma	20.07	15.30	19.65	11.25	18.10	19.89
EU_flights	542.80	293.49	312.40	295.62	318.31	296.91
openflights	646.38	131.40	148.55	146.68	159.09	138.59
yeast1	577.86	79.18	72.64	65.29	72.26	63.38
powergrid	1,092.29	88.36	257.03	157.62	249.33	96.50
OClinks	835.63	297.00	276.25	271.98	278.34	268.12
facebook	1,783.65	539.21	1,075.84	391.92	1,136.85	790.25
grqc	1,475.21	258.17	440.53	256.27	477.82	251.14
hepth	3,384.68	828.29	997.25	781.33	1,037.64	781.01
hepnh	4,820.37	1,377.08	2,048.81	1,322.39	2,175.45	1,313.49
astroph	8,137.28	3,319.25	4,099.93	3,160.29	4,275.94	2,921.57
condmat	8,966.96	2,132.57	2,752.33	2,067.86	2,914.13	2,052.45
Ялсан тоо	0	13	0	13	1	12
Нийт ялсан тоо	0	3	0	3	0	7
Дундаж сайжралт	1,772.98		275.27		317.50	

Хүснэгт 3: Дахин оруулах тэнцвэрт аргыг дахин оруулах тэнцвэрт арга дээр сэгсрэх техник ашигласантай харьцуулсан үр дүн. R_{BL} нь дахин оруулах тэнцвэрт арга, R_{BLS} нь сэгсрэх техниктэй дахин оруулах тэнцвэрт арга. Тухайн графын хувьд хамгийн бага үр дүнг тодоор харуулав.

Сүлжээ	Random		HDA		CI	
	R _{BL}	R _{BLS}	R _{BL}	R _{BLS}	R _{BL}	R _{BLS}
humanDiseaseome	16.82	14.52	10.55	9.67	10.66	10.35
Treni_Roma	15.30	9.71	11.25	8.70	19.89	16.98
EU_flights	293.49	284.24	295.62	268.22	296.91	279.38
openflights	131.40	102.38	146.68	111.15	138.59	125.40
yeast1	79.18	58.32	65.29	52.33	63.38	54.97
powergrid	88.36	67.52	157.62	52.57	96.50	55.64
OClinks	297.00	263.54	271.98	252.32	268.12	254.57
facebook	539.21	419.95	391.92	398.88	790.25	311.75
grqc	258.17	231.09	256.27	218.79	251.14	219.11
hepth	828.29	727.20	781.33	695.65	781.01	697.49
hepnh	1,377.08	1,219.26	1,322.39	1,212.73	1,313.49	1,175.91
astroph	3,319.25	2,927.67	3,160.29	2,892.50	2,921.57	2,875.26
condmat	2,132.57	1,856.72	2,067.86	1,819.64	2,052.45	1,819.67
Ялсан тоо	0	13	1	12	0	13
Нийт ялсан тоо	0	2	0	8	0	3
Дундаж сайжралт	91.85		72.76		85.19	

Хүснэгт 4: Санамсаргүй оройг устган сүлжээг задлан R1, R2, R3, R_BLS дахин оруулах аргуудыг харьцуулав.

Сүлжээ	Random			
	R1	R2	R3	R_BLS
humanDisease	23.57	17.43	16.50	14.52
Treni_Roma	16.78	16.09	13.60	9.71
EU_flights	283.19	295.10	284.40	284.24
openflights	121.96	144.22	111.02	102.38
yeast1	93.92	78.05	77.66	58.32
powergrid	288.04	122.62	116.49	67.52
OClinks	297.50	301.30	291.31	263.54
facebook	799.59	464.18	263.45	419.95
grqc	401.72	259.55	254.21	231.09
hepth	928.04	829.74	783.03	727.20
hepph	1,573.41	1,387.83	1,300.08	1,219.26
astroph	3,383.85	3,325.69	3,193.14	2,927.67
condmat	2,495.34	2,115.95	2,078.61	1,856.72
Ялсан тоо	1	0	1	11

Хүснэгт 5: HDA аргаар сүлжээг задлан R1, R2, R3, R_BLS дахин оруулах аргуудыг харьцуулав.

Сүлжээ	HDA			
	R1	R2	R3	R_BLS
humanDisease	17.83	13.65	13.39	9.67
Treni_Roma	19.60	11.36	10.67	8.70
EU_flights	263.41	296.18	272.88	268.22
openflights	109.22	147.55	109.01	111.15
yeast1	72.63	64.65	61.94	52.33
powergrid	242.75	181.89	163.96	52.57
OClinks	258.23	270.11	261.65	252.32
facebook	869.79	658.02	604.11	398.88
grqc	339.59	267.12	261.06	218.79
hepth	896.86	790.33	784.21	695.65
hepph	1,559.41	1,330.92	1,291.31	1,212.73
astroph	3,359.73	3,237.00	3,073.13	2,892.50
condmat	2,456.12	2,064.25	2,022.59	1,819.64
Ялсан тоо	1	0	1	11

Хүснэгт 6: CI аргаар сүлжээг задлан R1, R2, R3, R_BLS дахин оруулах аргуудыг харьцуулав.

Сүлжээ	CI			
	R1	R2	R3	R_BLS
humanDisease	18.18	13.65	11.49	10.35
Treni_Roma	20.74	19.07	19.89	16.98
EU_flights	269.31	303.09	282.63	279.38
openflights	120.58	107.53	115.52	125.40
yeast1	69.77	63.43	61.41	54.97
powergrid	237.97	92.75	98.46	55.64
OClinks	252.16	266.43	255.02	254.57
facebook	1,049.71	923.39	641.01	311.75
grqc	300.24	257.49	248.60	219.11
hepth	816.87	781.60	754.49	697.49
hepph	1,364.16	1,322.79	1,278.56	1,175.91
astroph	3,068.38	2,943.97	2,936.36	2,875.26
condmat	2,214.22	2,065.00	1,998.21	1,819.67
Ялсан тоо	2	1	0	10

Санхүүжилт

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

Ашиг сонирхлын зөрчилгүйн баталгаа

Ашиг сонирхлын зөрчилгүй болно.

Ашигласан ном

- [1] Zanin M, Lillo F. Modelling the air transport with complex networks: A short review. The European Physical Journal Special Topics. 2013;215(1):5–21.
- [2] Cuadra L, Salcedo-Sanz S, Del Ser J, Jiménez-Fernández S, Geem ZW. A critical review of robustness in power grids using complex networks concepts. Energies. 2015;8(9):9211–9265.
- [3] Albert R, Barabási AL. Statistical mechanics of complex networks. Reviews of modern physics. 2002;74(1):47.
- [4] Sharan R, Ideker T. Modeling cellular machinery through biological network comparison. Nature biotechnology. 2006;24(4):427–433.
- [5] Rubinov M, Sporns O. Complex network measures of brain connectivity: uses and interpretations. Neuroimage. 2010;52(3):1059–1069.
- [6] Kelly ME, Duff H, Kelly S, Power JEM, Brennan S, Lawlor BA, et al. The impact of social activities, social networks, social support and social relationships on the cognitive functioning of healthy older adults: a systematic review. Systematic reviews. 2017;6(1):1–18.
- [7] Kong X, Shi Y, Yu S, Liu J, Xia F. Academic social networks: Modeling, analysis, mining and applications. Journal of Network and Computer Applications. 2019;132:86–103.
- [8] Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L. Network dismantling. Proceedings of the National Academy of Sciences. 2016;113(44):12368–12373.
- [9] Morone F, Makse HA. Influence maximization in complex networks through optimal percolation. Nature. 2015;524(7563):65–68.
- [10] Tian L, Bashan A, Shi DN, Liu YY. Articulation points in complex networks. Nature communications. 2017;8(1):1–9.
- [11] Ren XL, Gleinig N, Helbing D, Antulov-Fantulin N. Generalized network dismantling. Proceedings of the national academy of sciences. 2019;116(14):6554–6559.

- [12] Wandelt S, Sun X, Feng D, Zanin M, Havlin S. A comparative analysis of approaches to network-dismantling. *Scientific reports*. 2018;8(1):1–15.
- [13] Fan C, Zeng L, Feng Y, Xiu B, Huang J, Liu Z. Revisiting the power of reinsertion for optimal targets of network attack. *Journal of Cloud Computing*. 2020;9:1–13.
- [14] Aringhieri R, Grosso A, Hosteins P, Scatamacchia R. Local search metaheuristics for the critical node problem. *Networks*. 2016;67(3):209–221.
- [15] Goh KI, Cusick ME, Valle D, Childs B, Vidal M, Barabási AL. The human disease network. *Proceedings of the National Academy of Sciences*. 2007;104(21):8685–8690.
- [16] Leskovec J, Krevl A. SNAP Datasets: Stanford Large Network Dataset Collection; 2014. <http://snap.stanford.edu/data>.
- [17] Morone F, Min B, Bo L, Mari R, Makse HA. Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific reports*. 2016;6(1):1–11.