

## Генетик Алгоритм ашиглан Critical Node Detection Problem-ыг бодох нь

Багийн Гишүүд

- Нямдаваа
- Мичидмаа
- Мөнхдэлгэр
- Сарнай

### 0 Хураангуй

CNDP нь хэрэглэгчийн тодорхойлсон  $k$  утганд  $G = (V, E)$  графаас  $R$  ширхэг дэд оройн ( $R \subseteq V, |R| \leq k$ ) олонлогийг устгахад графын компонентуудын холбоосын тоо хамгийн бага байх  $R$  дэд оройн олонлогыг олох бодлого юм. Энэ өгүүлэлд бид CNDP-ийг генетик алгоритм (GA) ашиглан шийдвэрлэх болно. Генетикийн алгоритм нь байгалийн хувьслын зарчмуудад тулгуурлагдсан сүрэгт суурилсан бодлогын үр дүнг сайжруулах арга юм. Генетик алгоритмд суурилсан CNDP-г олох алгоритм нь  $R$  оройн дэд олонлогыг хувьсан өөрчлөх, боломжит сайн шийдлүүдийг удамшуулах маягаар бодлогын боломжит шийдлүүдээс оновчтой эсвэл бараг-оновчтой шийдийг олох зорилготой болно.

Түлхүүр үг – генетик алгоритмт, Critical Node Detection Problem (CNDP)

### I Удиртгал

Critical Node Detection Problem (CNDP) нь сүлжээн дэх хамгийн чухал зангилаа (орой)-нуудыг тодорхойлоход чиглэсэн бодлого юм. Хэрэв тэдгээр оройнууд устгавал сүлжээний бүтэц, үйл ажиллагаанд ихээхэн эвдрэл, доголдол үүсгэх аюултай. Critical node-ыг тодорхойлсноор тээврийн сүлжээ, цахилгаан сүлжээ, харилцаа холбооны сүлжээ, нийгмийн сүлжээ зэрэг янз бүрийн салбаруудын үйл ажиллагаанд чухал нөлөө үзүүлдэг.

CNDP нь NP-hard төрлийн бодлого юм. Энэ нь нь оновчтой шийдлийг олох энгийн алгоритм байх магадлал багатай гэсэн үг. Гэсэн хэдий ч CNDP-ийн сайн шийдлийг олохын тулд ашиглаж болох хэд хэдэн арга байдаг.

CNDP-ийг шийдвэрлэхэд ашиглаж болох нэг арга бол генетик алгоритм юм. Генетик алгоритм нь байгалийн шалгарлын үйл явцаас сэдэвлэгдэн бүтээгдсэн бодлогын үр дүнг сайжруулах алгоритм юм.

Генетик алгоритмын үндсэн бүрэлдэхүүн хэсгүүдэд санамсаргүйгээр боломжит шийдлүүдийг илэрхийлэх сүрэг үүсгэх, шийд болгоны чанарыг үнэлэх зорилгын функц, selection, crossover, mutation гэх мэт операторуудыг давтан ашиглах замаар шийдлүүдийг сайжруулан оновчтой шийдлийг олно.

- Initialization: Сүргийг анхлан байгуулах буюу Generation 0-ийг үүснэ. Энэ нь individual эсвэл хромосом гэж нэрлэгддэг боломжит шийдлийн сүргийг санамсаргүй байдлаар үүсгэнэ. Нэг individual бүр бодлогыг шийдэх нэг боломжит шийдлийг төлөөлдөг.
- Evaluation: Зорилгын функцийг сүрэгт байгаа individual болгоны fitness value-г тооцоолж дараагийн үед өвлөгдөн үлдэх шийдлүүдийг үнэлэнэ.
- Selection: Сайн fitness value-тэй individual-уудыг сүргээс сонгогдоно.
- Reproduction: Сонгогдсон individual-уудаас шинэ үр удам үүсгэнэ. Үүнийг ихэхдээ crossover, mutation гэсэн генетикийн операторуудыг ашиглан хийдэг.
  - o Crossover: Эцэг эх хоёр individual-ын хооронд удамшлын мэдээлэл солилцох байдлаар үр удмаа бий болгох явдал юм. Crossover-ыг хийхдээ нэг гений кроссовер, олон гений кроссовер, жигд кроссовер гэсэн техникүүдээс ашиглаж болно.
  - o Mutation: Шинэ шийдэл, үр дүн олохын тулд зарим individual хромсомууд дээр санамсаргүйгээр өөрчлөлт оруулдаг.
- Replacement: Шинээр үүссэн төл болон өмнөх үеийн зарим individual-уудыг нэгтгэн шинэ үе (generation) үүсгэнэ. Selection болон Reproduction шатуудыг дараагийн үед давтан хийнэ.
- Termination: Алгоритмыг дуусгавар болох шалгуур хангах хүртэл дээрх шатуудыг үе дамжин давтагдана. Дуусгавар болгох шалгуур нь generation-ы тоогоор хязгаарлах, хүссэн fitness value хүртэл давтах, эсвэл хугацааны зааг тавих байдлаар тодорхойлж болно.
- Solution Extraction: Алгоритм нь fitness value дээр тулгуурлан хамгийн сүүлийн generation-ы хамгийн шилдэг individual-ыг олно. Тухайн individual нь бодлогын оновчтой эсвэл бараг оновчтой шийдлийг илэрхийлнэ.

GA нь олон төрлийн optimization асуудлыг шийдвэрлэхэд үр дүнтэй байдаг. Гэсэн хэдий ч, GA-ийн гүйцэтгэл нь асуудал, бодлого болгоноос хамаарч өөр өөр байж болно. Зарим тохиолдолд GA оновчтой шийдлийг олж чаддаг бол бусад тохиолдолд зөвхөн сайн ойролцоо шийдийг олж чадна.

## II Ажиглалт

1.  $G = (V, E)$  графаас хасах  $R$  оройн олонлогын чадал  $k$ -тай тэнцүү байвал хамгийн сайн шийдэлд дөхнө.

2.  $G = (V, E)$  графаас хасах  $R$  оройн олонлогыг хассаны дараах гарах үр дүнд үүсэх компонентуудын оройн тоо аль болох хоорондоо тэнцүү буюу компонентуудын оройн тоон зөрүү бага байх тусам хамгийн сайн шийдэлд дөхнө.

### III Шийдэл

Генетик алгоритм (GA) ашиглан critical node detection problem (CNDP)-ыг шийдвэрлэхийн тулд бид доор дурдсан алхмуудыг дагаж болно.

1. Representation: Сүргийн individual бүр CNDP-ийн боломжит шийдлийг төлөөлнө.  $G = (V, E)$  графаас  $k$  тоогоор  $R$  оройн олонлогийг санамсаргүйгээр үүсгэнэ. Санамсаргүйгээр үүсгэсэн олонлог нь  $G$  графт харьяалагдах оройн дугааруудаас  $k$  ширхэг байна.
2. Initialization: Санамсаргүйгээр нэг сүрэгт 100 ширхэг individual-тай байхаар generation 0-ийг үүсгэнэ. Individual бүрт  $G$  графт харьяалагдах оройн дугааруудаас  $k$  ширхэг байна.
3. Evaluation: Сүргийн individual бүр дээр Objective Function-оор тус бүрчлэн Fitness value-г тооцоолно. Objective Function нь графын компонент бүрийн нийт оройн хосын тооны нийлбэр болно. Fitness value утга бага байвал сайн.

*for component in components*

*fitness += len(component) \* len(component - 1) / 2*

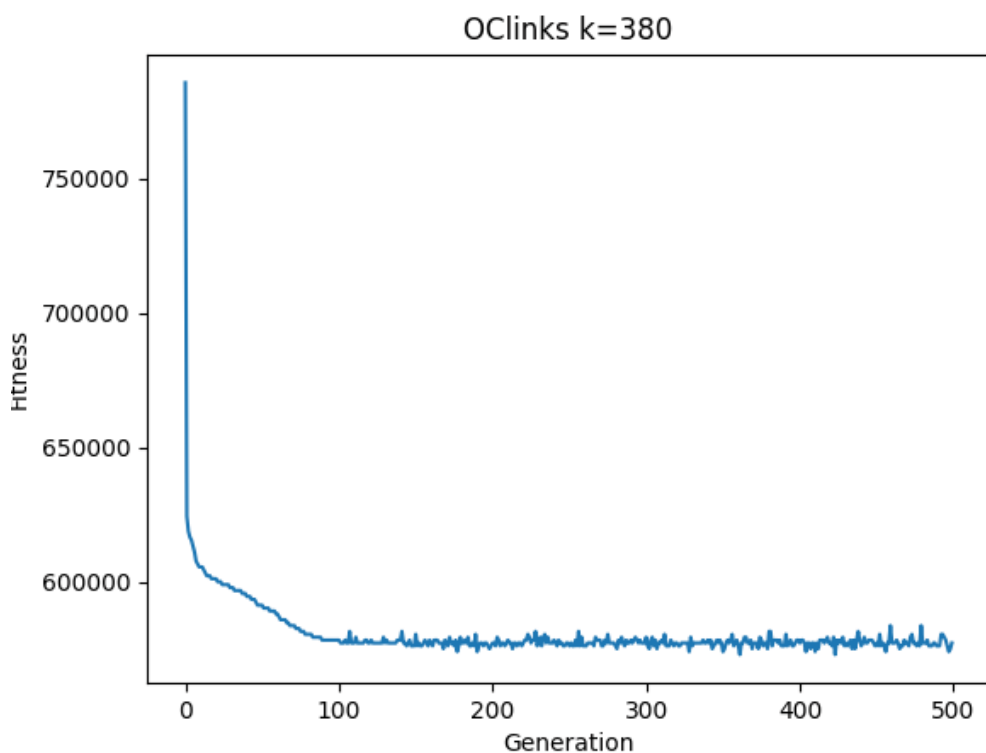
4. Selection: Crossover үйлдлийг хийхийн тулд тооцоолж гаргасан Fitness value array-гаас Tournament Selection аргыг ашиглан эцэг болон эх хромсомыг сонгоно. Tournament Selection арга нь өмнөх generation-аас  $a$  (жишээ нь:  $a = 3$ ) ширхэг хромсомыг санамсаргүйгээр сонгож аваад, тэдгээрээс хамгийн их Fitness value-тай хромсом нь parent болгон сонгох арга юм.
5. Reproduction: Сонгогдсон эцэг болон эх хромсомдаас үр удам үүсгэхийн тулд кроссовер, мутаци зэрэг генетикийн операторуудыг ашиглана.
  - a. Crossover: Сонгогдсон эцэг болон эх генүүдийг голоор нь тэнцүү хуваагаад эцгийн эхлэл хэсгийн гений цуваан дээр эх хромсомын төгсгөл хэсгийн цувааг, харин эх хромсомын эхлэл хэсгийн гений цуваан дээр эцгийн төгсгөл хэсгийн цувааг нэмээд хоёр шинэ хүүхэд хромсомууд үүсгэнэ.
  - b. Mutation: Crossover оператороор үүссэн нийт шинэ хүүхэд хромсомуудаас 10% магадлалтайгаар мутаци үйлдэл хийнэ. Мутаци үйлдэл нь санамсаргүй хромсомын санамсаргүй ген дээр санамсаргүйгээр графын оройн дугаарыг сонгоод орлуулж тавина.
6. Replacement: Өмнөх generation-ны хромсомууд болон Reproduction шатаас үүссэн шинэ хромсомуудын дундаас дараагийн generation-ыг үүсгэнэ. Шинэ generation-нд хамгийн сайн Fitness value-тай хромсомууд орно.

7. Termination: Хэрэв maximum generation-ны өгөгдсөн тоонд хүрээгүй бол 3-р алхам болох Evaluation оператороос ахин эхлэнэ. Үгүй бол дараагийн алхам руу шилжилнэ.
8. Solution Extraction: Эцсийн generation дахь хамгийн сайн Fitness value-тай хромсом нь CNDP-ийн хамгийн оновчтой шийд болно.

## IV Дүгнэлт

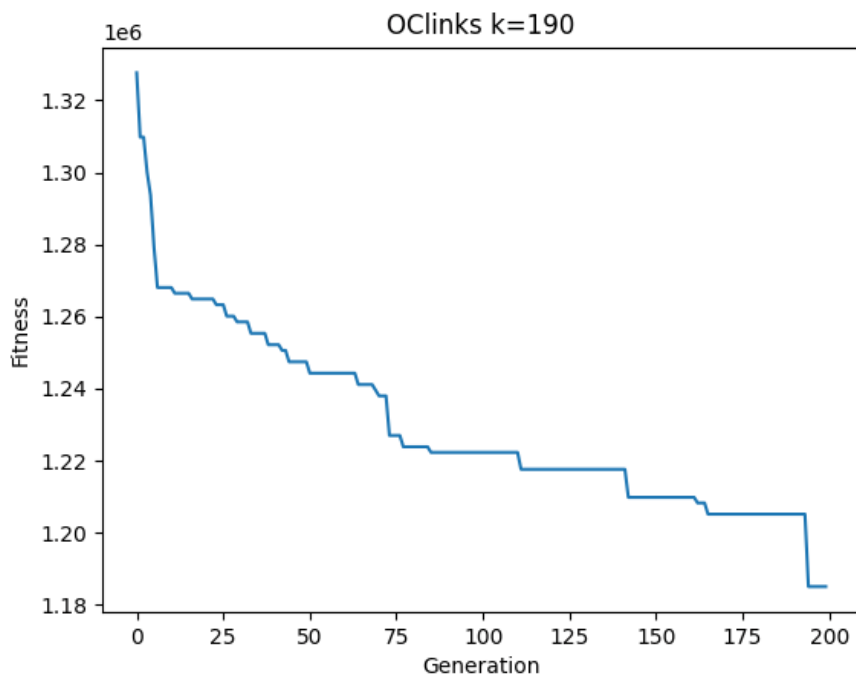
### Алгоритм 1

Oclinks графаас maximum generation-ий тоо 300, устгагдах R олонлогийн чадал  $k = 380$  ширхэг оройн тоо байхад гарах үр дүн. 1 generation = 5 seconds, 40 minutes = 500 generations



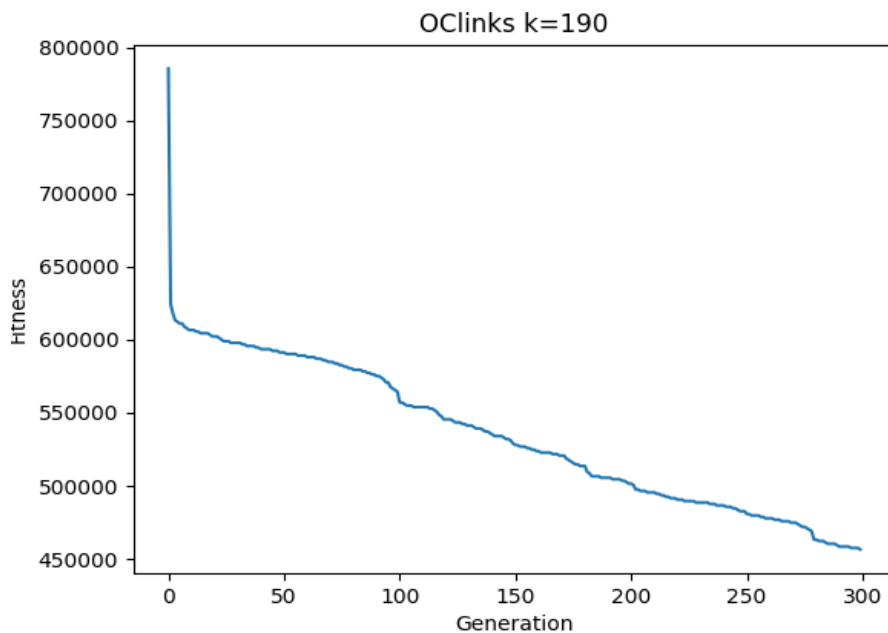
Дээрх үүссэн графаас харахад эхний 100 generation-ий дотор бараг-optimal шийдийг олсон харагдаж байна. Харин дараагийн 400 generation-ний турш том өөрчлөлт үүсгэхгүйгээр бага зэрэг хэлбэлзсэн байна.

Харин maximum generation-ий тоо 200, устгагдах R олонлогийн чадал  $k = 190$  ширхэг оройн тоо байхад гарах үр дүн дараах байдлаар харагдаж байна.



**Алгоритм 2** – Generation 0 үүсгэхдээ Degree centrality хамгийн их байдаг k оройг population-д нэмэх. 1 generation = 3 seconds

1. K = 190, Gen = 300, Population\_size = 300 үед Oclinks сүлжээн дээр гарсан үр дүн.



2.  $K = 380$ ,  $Gen = 150$ ,  $Population\_size = 300$  үед Oclinks сүлжээн дээр гарсан үр дүн.

