

1. Problem Statement

방향성이 없는 간선에 가중치가 부여된 단순 그래프(simple graph) $G = (V, E)$ 가 주어진다. V 와 E 는 각각 정점 집합과 간선 집합이다. 정점 집합 V 를 두 개의 집합 S 와 $S' (= V - S)$ 로 나누는 방법은 여러 종류가 있는데, 여러분은 S 와 S' 을 잇는 간선의 가중치의 총합을 최대화하도록 나눠야 한다. 그림 1은 10개의 정점과 14개의 간선으로 이뤄진 단순 그래프의 한 인스턴스다. 각 간선의 가중치는 1 또는 -1로 주어져있다. 그림 2는 그림 1의 그래프의 정점 집합을 둘로 나누는 방법 중 하나로, $S = \{1, 6, 8, 9\}$, $S' = \{2, 3, 4, 5, 7, 10\}$ 로 나눈 것이다. S 와 S' 에 속한 정점들을 각각 검은색과 흰색으로 표시하였다. 굵게 표시된 간선은 S 와 S' 을 잇는 간선으로 가중치의 총합은 8이 된다. 그림 3은 정점 집합을 둘로 나누는 다른 방법으로, 두 집합을 잇는 간선의 가중치의 총합은 7이다.

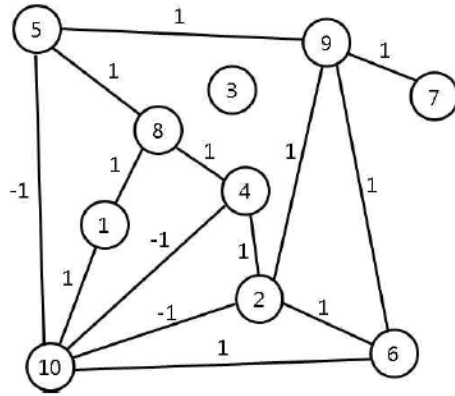


그림 1

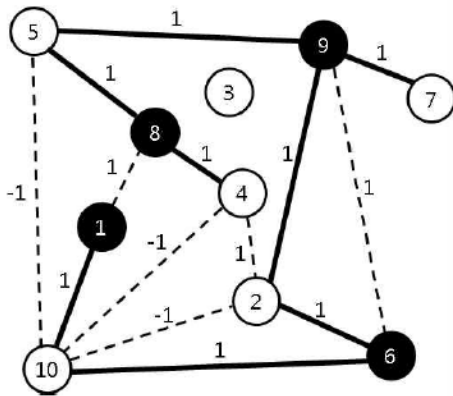


그림 2

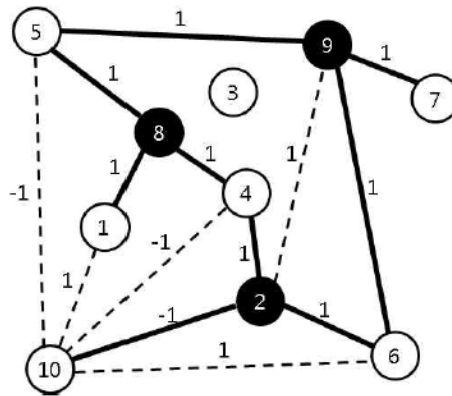


그림 3

주어진 단순 그래프 G 에서 정점 집합을 두 집합 S 와 $S' (= V - S)$ 로 나눌 때, 이 둘을 잇는 간선의 가중치의 총합을 제한 시간 내에 가능한 최대화하는 Genetic Algorithm을 작성하라.

- 인스턴스의 크기: 정점의 개수 $|V| \leq 500$, 간선의 개수 $|E| \leq 5000$
- 제한 시간: 한 인스턴스 당 180초

[입력]

입력 파일의 이름은 maxcut.in이다. 입력 파일의 각 항목은 공백과 줄바꿈으로 구분된다. 첫 줄에는 정점의 개수 $|V|$ 와 간선의 개수 $|E|$ 가 주어진다. 둘째 줄부터 차례대로 각 간선의 양 끝 정점의 번호와 간선의 가중치가 주어진다. 정점의 번호는 1 이상 $|V|$ 이하의 정수로 주어지고, 간선의 가중치는 정수로 주어진다. 시작점과 끝점이 같은 간선(self-loop)이나 동일한 두 정점을 잇는 복수 개의 간선(multiple edges)는 주어지지 않는다.

[출력]

출력 파일의 이름은 maxcut.out이다. 여러분이 구한 최적 집합 S 내의 모든 정점 번호를 한 줄에 공백으로 구분하여 출력한다. 출력 파일에 가중치 총합 수치를 포함하지 않는다.

[입출력 예]

입력 (그림 1의 그래프)

```
10 14
1 10 1
1 8 1
2 10 -1
2 9 1
2 6 1
2 4 1
4 8 1
4 10 -1
5 9 1
5 10 -1
5 8 1
6 10 1
6 9 1
7 9 1
```

출력(그림 2의 $S = \{1, 6, 8, 9\}$)

```
1 6 8 9
```

2. Restriction

이번 과제에서는 순수 GA만 사용하도록 한다. 지역 최적화(local optimization)의 성격을 갖는 과정이 어떤 식으로든 포함되지 않도록 한다. 이번 과제는 순수 GA가 갖는 한계를 경험하는 것을 목표로 하며, 이를 위해 crossover, mutation 및 기타 preprocessing에서 지역 최적화 과정을 금지하는 것으로 이해하면 된다. 즉, 강의 후반에 배운 병렬 GA(island method, elite GA 등), 국부

최적화(local optimization)을 사용할 수 없다.

이번 과제를 포함, 알고리즘 강좌의 나머지 숙제에서 제출할 프로그램은 반드시 CPU 상에서 single thread로 수행되어야 한다. 멀티코어 혹은 멀티쓰레드 컴퓨팅은 허용하지 않으며, gpu 활용 역시 허용하지 않는다. 프로그램은 외부와 통신 없이 단독으로 수행되어야 한다. 앞의 제약을 위반한 경우 큰 패널티를 받는다.

3. Language

프로그램 언어는 C++을 사용한다. 다른 언어를 사용하는 경우 미리 교수와 상의하여 승인된 경우에 한해 허용한다. 제한 시간(180초)을 두고 수행하는 채점 방식 상, 수행속도가 빠른 C++을 사용하는 것이 바람직하다. C++이 기본적으로 제공하는 standard library 이외의 다른 라이브러리를 사용하는 경우에는 미리 교수와 상의해야 한다. 교수와 상의 없이 다른 언어나 라이브러리를 사용하는 경우, 큰 패널티를 받을 수 있다. Standard library는 standard container를 포함한다.

4. Compilation and Execution

~~제출 시 반드시 Makefile을 포함해야 하며, make all 명령으로 컴파일이 가능하도록 만들어야 한다. C++을 사용하는 경우 컴파일러의 최적화 옵션은 O3까지 사용할 수 있다. 또한 make run 명령으로 프로그램이 수행되도록 한다. 아래 예시와 같이 명령을 입력했을 때, 여러분이 구한 해가 maxcut.out에 기록되어야 한다.~~

```
$ ls
Makefile yourfile1.cpp yourfile2.cpp ...
$ make all
...
$ cp /home/ga_Ta/instance1 ./maxcut.in
$ make run
...
$ cat ./maxcut.out
1 6 8 9
```

~~컴파일과 수행은 리눅스에서 가능해야 한다. 채점 컴퓨터는 Intel Xeon Scalable 4210 이며, OS는 Ubuntu Server 22.04.2 LTS다. 컴퓨터 사양 차이를 고려하여 알고리즘을 작성하는 것을 권장한다.~~

실행은 제출한 .exe 파일을 이용해 윈도우에서 배치로 진행한다. 실행 시 콘솔창에 어떤 문자열도 출력하지 않는다. 채점 컴퓨터는 Intel i7-11700 @ 3.60 GHz 이며, OS는 Windows 11 Professional 이다. 컴퓨터 사양 차이를 고려하여 알고리즘을 작성하는 것을 권장한다.

5. Team

팀을 구성하여 문제를 해결할 수 있다. 팀의 구성 인원은 최소 1인, 최대 5인이다.

6. 제출

제출 파일은 아래 3파일을 학번.zip 으로 압축하여 제출한다. 폴더 형태로 압축하지 않는다.

- 학번.cpp
- 학번.exe
- 학번.pdf

학번은 팀원 중 가장 빠른(가장 작은) 숫자이며, LMS 제출은 해당 학번 학생만 제출하고, 다른 팀원은 제출하지 않는다. 학번 숫자에 반드시 하이픈을 포함한다.

예시.

수하루

2021-11111	2022-11110	2021-11113	2021-11114	2021-11115
김서아	김이서	김지안	김하윤	김지아

제출 파일: 2021-1111.zip

압축 해제 시 다음 3파일이 .zip 파일과 같은 경로에 위치한다.

- 2021-1111.cpp
- 2021-1111.exe
- 2021-1111.pdf

7. Report

프로젝트에 대해 보고서를 작성하여 제출한다. 분량은 A4 4페이지 이하이며, 팀 이름과 멤버의 학번, 이름을 적고 주어진 형식으로 작성한다. 보고서 내에 소스 코드를 그대로 포함해서는 안된다. 포함 시 큰 패널티를 받는다. 보고서 본문에는 최소한 다음 내용을 포함시킨다.

1. 사용한 GA 구조(팀원 별 의견 표를 포함할 것)
2. 해의 표현(chromosome design, 팀원 별 의견 표를 포함할 것)
3. 사용한 연산자에 대한 설명(selection, crossover, mutation, replacement 등, 팀원 별 의견 표를 포함할 것)
4. 함께 제공하는 세 개의 샘플 인스턴스 인스턴스에 대해 GA를 각각 최소 30번씩 수행하여 가장 좋은 결과, 평균 결과, 표준편차를 테이블로 기록(최소 90번의 GA run)
5. 1~3을 거치며 팀원의 의견을 하나로 모은 뒤, 최고의 GA에 대한 팀 의견
6. 위에서 수행한 GA run들 중 하나를 선택, 세대 진행에 따른 population 분석(해들의 평균 품질, 최고 품질 등)
7. Discussion(느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등)
8. 9강 강의 중 생각해 볼 내용에 대한 팀원 별 답변 정리 표

8. Grading

- 평가는 보고서와 코드를 1:1 비율로 평가한다. 제출한 코드 중 일부는 공개되며, 제출 후

팀의 코드를 본인의 포트폴리오에 올려도 무방하다. 둘 모두 상대평가에 의해 점수가 부여된다.

- 채점 컴퓨터는 별도로 준비한 인스턴스에 대해 제출한 .exe 파일을 180초 동안 실행하고 강제 종료한 뒤, 출력 파일의 점수를 확인하는 과정을 3회 반복하여 평가한다. 출력 형식이나 내용이 잘못되어 채점이 불가능하거나, 채점을 방해하는 행위(폴더 삭제 등)가 발생할 경우 0점을 받는다. 별도로 준비한 인스턴스에 대해 가장 좋은 품질의 해가 만점을 받고, 다른 해는 그 품질의 차이에 따라 부분 점수를 받는다. 다른 해는 그 품질의 차이에 따라 부분점수를 받는다. 부분점수는 품질 차이의 절대치와 상대치(비율)에 따라 부여된다. 외부 프로그램도 경쟁에 포함될 수 있다.
- 보고서는 위의 최소 내용을 모두 포함할 경우 기본 점수가 부여되며, 나머지 성적은 보고서 내용이 충실한 정도에 따라 평가한다. 보고서에 불필요한 내용(표지, 소스코드, 별도의 분석 없는 raw data 등)을 포함하는 경우 감점 요인이 된다.
- Copy detection program에 의해 카피 판별 후, 원본과 복사본 공히 0점 처리하며 최종 학점에서 두 grade 강등함. 일부든 전부든 모두 카피로 처리함.
- 보고서와 프로그램을 따로 제출할 수 없다.