

2023 Algorithm HW 8 : MAX-CUT

due: 6월 9일(금) 오후 23시 59분

1. Problem Statement

방향성이 없는 간선에 가중치가 부여된 단순 그래프(simple graph) $G = (V, E)$ 가 주어진다. 정점 집합 S 와 $S' (= V - S)$ 로 나눌 때, 이 둘을 잇는 간선의 가중치의 총합을 최대화하는 Hybrid GA를 작성하라. 단, 아래(2. Local Search)에 주어진 지역최적화 알고리즘을 사용할 수 있다. 입출력 형식과 파일명, 제출 방식은 지난 프로젝트와 동일하다.

- 인스턴스의 크기: 정점의 개수 $|V| \leq 1000$, 간선의 개수 $|E| \leq 10000$
- 제한 시간: 한 인스턴스 당 180초

2. Local Search

해 x 에서 정점 하나를 선택하여 반대편 집합으로 옮겨서 얻을 수 있는 해들의 집합을 $N(x)$ 라 하고(1-bit flip) $N(x)$ 의 원소는 ' x 에 이웃한 해'라 부르자. 이 때 임의의 해 x 에 이웃한 해의 개수 $|N(x)|$ 는 항상 $|N(x)| = |V|$ 를 만족한다. 아래의 알고리즘은 시작점으로 주어진 해 x 에서 출발하여, 이웃한 해들 중 더 좋은 해로 이동하는 과정을 반복한다. 도중에 더 좋은 이웃해가 없다면 이 알고리즘 local optimum에 도달한 것을 의미하고, 이 때 알고리즘은 종료된다.

Input: initial solution x	
1	generate a random permutation σ of $\{1, \dots, V \}$;
2	$improved \leftarrow \text{true}$;
3	while $improved$ do
4	$improved \leftarrow \text{false}$;
5	for $i \leftarrow 1$ to $ V $ do
6	if $\Delta f_x(\sigma[i]) > 0$ then
7	$x[\sigma[i]] \leftarrow 1 - x[\sigma[i]]$;
8	$improved \leftarrow \text{true}$;
9	end if
10	end for
11	end while

Line 6의 $\Delta f_x(v)$ 는 현재 해 x 에서 정점 v 를 반대편 집합으로 옮겼을 때 목적함수 값(가중치 합)의 변화량을 의미한다. 즉, 이 알고리즘은 이웃한 해들 중 개선되는 방향으로만 이동한다. Line 5-6에서 임의의 순서()로 정점을 순회하는 이유는 정점 순서에 따른 잠재적인 편향을 제거하기 위함이다. Line 7에서는 해 x 가 binary string $\{0, 1\}^{|V|}$ 로 표현되었음을 가정하고 있다. 만약 다른 표현형을 사용한다면 그에 맞게 Line 7을 바꾸면 된다.

3. Dynamic Programming

Dynamic Programming을 이용해 GA 구조를 변경할 수 있다(선택 사항).

4. Compilation and Execution

컴파일은 제출한 소스 파일을 이용해 채점 컴퓨터가 LINUX에서 컴파일한다. 채점 컴퓨터는 Intel Xeon Scalable 4210 이며, OS는 Ubuntu Server 22.04.2 LTS다.

5. Team

팀을 구성하여 문제를 해결할 수 있다. 팀의 구성 인원은 최소 1인, 최대 5인이며, 지난 숙제와 다른 친구들과 팀을 구성해도 무방하다.

6. 제출

제출 파일은 아래 3파일 2파일을 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 전체 구조
- 2) Dynamic programming을 활용한 방안
- 3) 사용한 GA 연산자에 대한 설명과 DP 알고리즘 설명
- 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) Discussion(느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등)

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 소스코드를 컴파일하여 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.