AIP

HW 07 – REPORT

소속 : 정보컴퓨터공학부 컴퓨터공학전공

학번 : 202355570

이름 : 이유진

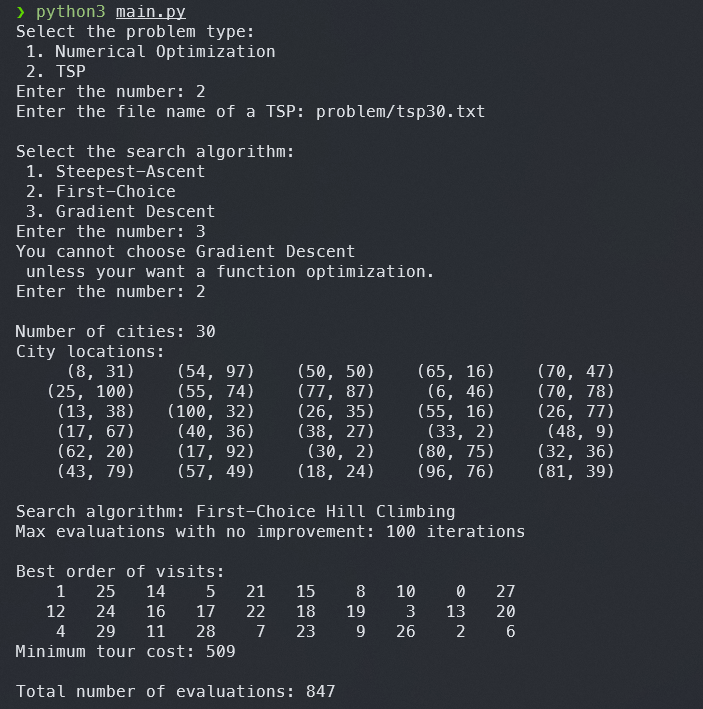
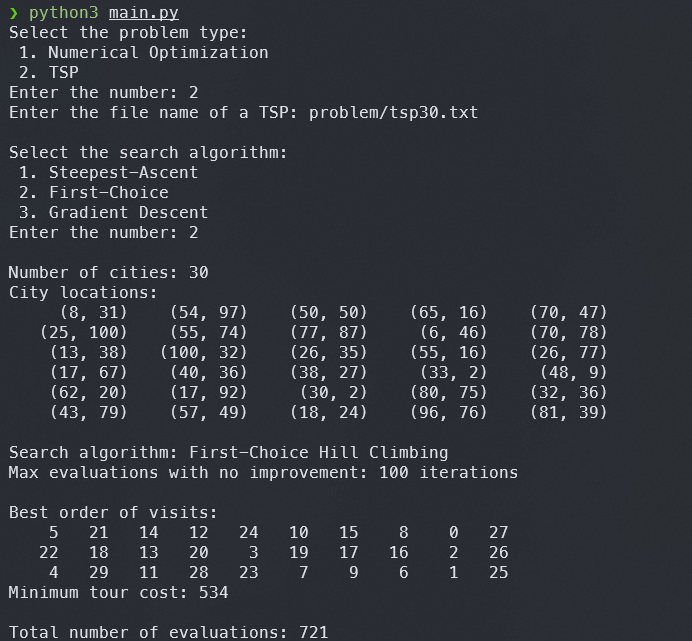
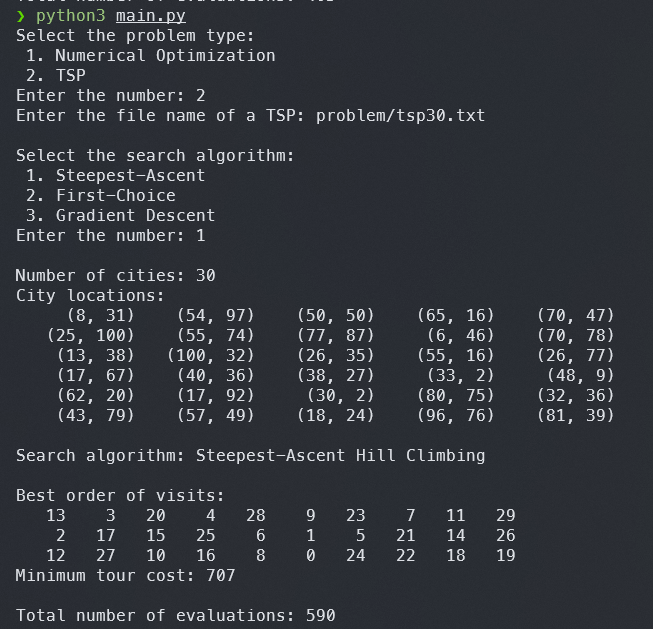
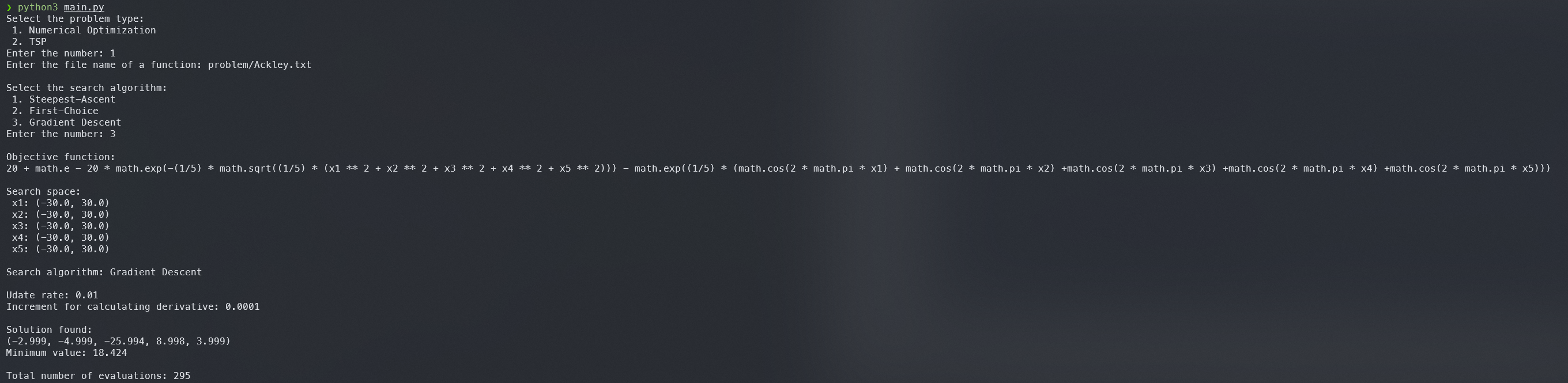
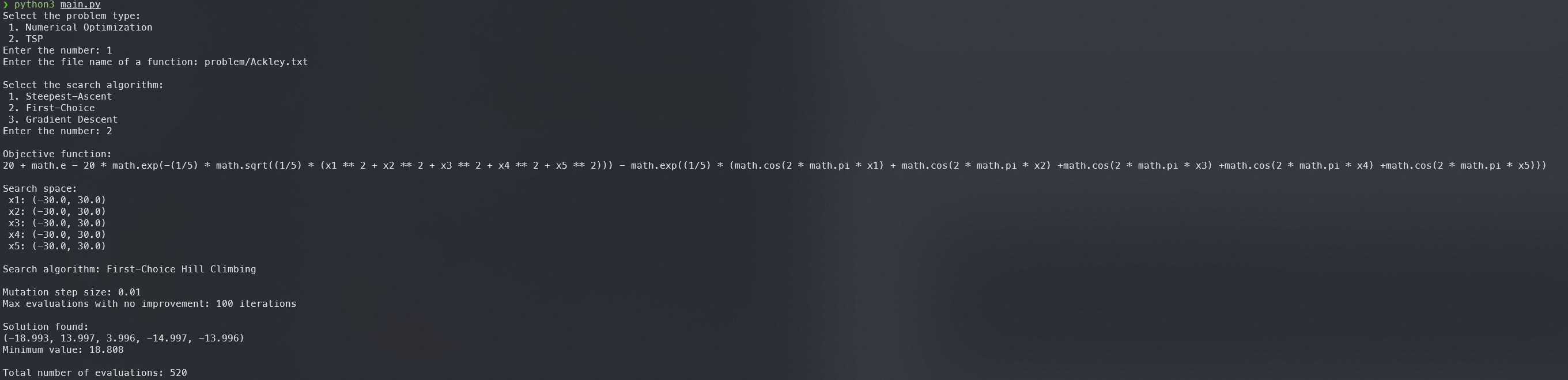
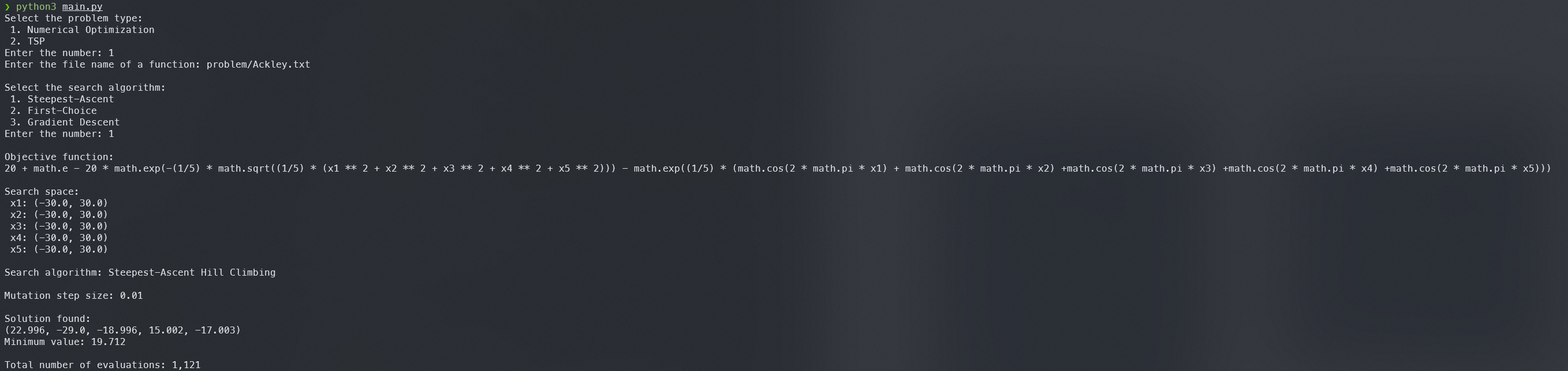
1. 서론

이전 과제에서 만든 class들을 하나의 프로그램으로 통합하여 한 프로그램 (main.py) 만을 실행시켜도 여러가지 문제를 여러 방식으로 해결할 수 있도록 구현해보자.

Setup라는 슈퍼클래스를 구현하여 HillClimbing 클래스와 Problem 클래스에서 모두 필요한 정보를 공유할 수 있도록 만들어보자.

SteepestAcent, FirstChoice, GradientDescent 세가지 검색 알고리즘을 통합시켜 HillClimbing class를 구현해보자.

2. 본론

1. 먼저 Setup.py에서 Setup class를 구현하여 HillClimbing 클래스와 Problem 클래스에서 모두 필요한 정보를 공유할 수 있도록 self.\_delta, self.\_alpha, self.\_dx를 초기화할 수 있또록 만든다.
2. Problem.py 에서 `from setup import Setup`로 setup 클래스를 import해주고, Problem(Setup)으로 해당 클래스를 상속하도록 만들어준다. 슈퍼클래스를 호출하여 delta, alpha, dx를 초기화해주고 기존의 따로 초기화하던 부분을 없애준다.
3. Optimizer.py 에서 Setup부분을 import하고 HillClimbing class를 만들어 Setup을 상속받도록 만들어준다. \_\_init\_\_에서 Setup에서 정의된 delta, alpha,dx에 접근할 수 있도록 슈퍼클래스의 생성자를 호출해주고 문제 Type을 구분하기위한 pType을 정의해준다.
4. pType을 초기화할 수 있는 setVariables 메서드와 공통적으로 출력되는 부분을 displaySetting 메서드를 만들어준다.
5. FristChoice, SteepestAscent. GradientDescent 서브클래스를 만들어 HillClimbing class를 상속받게 만들어주고 저번 과제에서만든 displaySetting을 가져와 각각 따로 구현해준다. 공통으로 출력되는 부분은 super().displaySetting()를 호출하여 출력해준다. 슈퍼클래스에서 선언만 해놓은 run부분을 만들어줌. (이전 과제 만든 FirstChoice(), SteepestAscent(), GradientDescent() 부분의 내용을 가져와서 구현한다.
6. SteepestAscent의 경우 bestOf를 추가적으로 메서드로 만들어 구현해준다. 파라미터에 self를 추가해주고, run 메서드 부분에 self.bestOf(neighbors,p)를 호출하도록 만들어주었다.
7. GradientDescent의 경우 출력방식이 약간달라 self.\_alpha, self.\_dx로 따로 출력해주었따.
8. Main.py 로 여러가지 문제를 여러가지 방식으로 풀 수 있도록 문제타입과 알고리즘을 선택할 수 있는 함수를 만들어 준다. 만약 problemType이 ==2 즉, TSP인 경우에는 Gradient Descent 를 사용할 수 없으므로 알고리즘 Type로 Gradient Descent 선택시 예외처리를 해줘야함
9. Invalid 함수를 구현하여 aType==2 && pType==3 인 경우 True를 리턴하게 하여 올바른 값이 나올때까지 입력을 받게 만들어준다.
10. 
11. 

3. 결론

토의 및 결론 (1페이지 이내)

본론의 결과를 바탕으로 실험 결과를 해석 및 분석