최적화 공학문제2

생성자	재 재환 김
태그	엔지니어링

1. 최적화 문제와 근 구하기 문제의 차이

- **근 구하기 문제**:
  - 근 구하기는 함수 f(x)의 값이 0이 되는 지점을 찾는 문제입니다. 예를 들어, 방정식 $f(x) = 0$의 해를 찾는 것과 같습니다.
  - 이 문제는 주로 수학적 방정식을 해결할 때 많이 사용되며, 다양한 수치적 기법(뉴튼법, 이분법 등)이 사용됩니다.
- **최적화 문제**:
  - 최적화는 함수 $f(x)$의 최소값 또는 최대값을 찾는 문제입니다. 즉, 주어진 조건 하에서 함수가 가장 작은 값(최소) 또는 가장 큰 값(최대)을 찾는 것입니다.
  - 최적화 문제는 주로 자원의 효율적 사용, 비용 절감, 성능 향상 등과 관련된 공학적 문제에서 많이 등장합니다.
- **공통점**:
  - 근 구하기 문제와 최적화 문제는 모두 주어진 함수에서 특정한 지점을 찾는다는 공통점이 있습니다.
  - 근 구하기 문제에서는 함수가 0이 되는 지점을 찾고, 최적화 문제에서는 도함수가 0이 되는 지점(극점)을 찾습니다. 이 점에서 유사하지만, 최적화 문제에서는 그 지점이 극대값인지 극소값인지 추가적인 검증이 필요합니다.
- **차이점**:
  - 근 구하기 문제에서는 $f(x) = 0$이 되는 x를 찾습니다.
  - 최적화 문제에서는 $f\prime(x) = 0$이 되는 x를 찾아서, 그 값이 최소값 또는 최대값인지 확인합니다.
  - 미분이 가능하지 않거나, 미분값을 구할 수 없는 경우도 많기 때문에, 근 구하기 문제처럼 단순하게 풀리지 않는 경우도 많습니다.

2. 최적화 문제의 제약 사항

최적화 문제는 근 구하기 문제보다 복잡할 수 있는 이유가 **제약 조건** 때문입니다. 예를 들어, 공학적 문제에서는 다양한 물리적, 자원적 제약 조건이 존재합니다.

- **제약 조건의 예**:
    - 예를 들어, 물질의 양은 0 이상이어야 하고, 비용은 일정한 한도를 넘지 않아야 하는 제약이 있을 수 있습니다.
    - 이처럼 최적화 문제는 함수 자체만을 고려하는 것이 아니라, 현실적인 제약 조건을 고려하여 해를 찾아야 합니다.
- **제약 조건을 포함한 최적화 문제**:
    - 최적화 문제에서 제약 조건이 포함되면, 단순히 도함수가 0이 되는 지점을 찾는 것이 아니라 제약 조건을 만족시키면서 그 지점을 찾아야 합니다.
    - 또한, 그 지점이 극대값인지 극소값인지 구분하는 과정도 필요합니다. 이를 위해 두 번째 미분을 사용하거나, 제약 조건을 함께 고려한 최적화 기법이 필요합니다.

3. 최적화 문제의 실제 예시: Traveling Salesman Problem (TSP)

- **문제 설명**:
    - 여러 도시를 방문해야 하는 상황에서, 주어진 모든 도시를 가장 짧은 경로로 방문하고 다시 출발점으로 돌아오는 문제입니다.
    - 이 문제는 **NP 문제**로 분류되며, 도시의 수가 증가할수록 가능한 경로의 수가 기하급수적으로 증가합니다.
- **가능한 해결법**:
    - 도시가 5개일 때는 순열을 계산하여 모든 경로를 나열한 후 가장 짧은 경로를 찾을 수 있습니다. 하지만 도시가 50개로 증가하면, 50!개의 경로를 모두 계산해야 하므로, 이 방식으로는 해결할 수 없습니다.
    - 따라서, 최적화 문제에서는 모든 경로를 계산하기보다는 주어진 시간 내에 근사해를 찾는 방법이 더 현실적입니다.
    - 예를 들어, **유전 알고리즘**, **탐욕 알고리즘**, **시뮬레이티드 어닐링(Simulated Annealing)** 등은 시간 내에 근사해를 찾는 대표적인 기법들입니다.

4. 낙하산 문제: 최적화의 구체적 사례

- **문제 설명**:
    - 구호 물자를 낙하산으로 전달하는 상황에서, 물자가 지면에 도착할 때의 충격 속도를 20m/s 이하로 유지하면서 비용을 최소화하는 문제입니다.

    - 이 문제에서 고려해야 할 변수들은 낙하산의 크기, 낙하산의 개수, 줄의 길이, 항력 계수 등이 있으며, 이들은 상호 연관되어 있습니다.
- **목적 함수**:

$$\text{Cost} = n(c_0 + c_1\ell + c_2A^2)$$

  여기서:
    - 최적화 문제에서 목적 함수는 비용을 최소화하는 것입니다. 이는 낙하산의 크기 R와 개수 N에 의존합니다.
    - c_0, c_1, c_2는 가격 상수입니다.
    - ℓ은 낙하산의 줄 길이, A는 낙하산의 표면적입니다.
- **제약 조건**:
    - 낙하산이 지면에 도착할 때 속도가 20 m/s 이하가 되어야 한다는 제약 조건이 있습니다.
    - 낙하산의 크기와 개수는 속도와 비용에 모두 영향을 미칩니다. 따라서, 이 둘 간의 상충 관계를 고려해야 합니다.

5. 최적화 기법의 종류

- **비제약 최적화**: 제약 조건이 없는 최적화 문제를 해결하는 기법입니다.
    - **황금분할법 (Golden Section Method)**: 1차원 문제에서 사용되는 구간을 줄여나가는 방법입니다.
    - **2차 보간법 (Quadratic Interpolation Method)**: 함수의 2차 근사를 통해 최적점을 찾는 방법입니다.
    - **뉴튼법 (Newton's Method)**: 도함수를 이용해 빠르게 수렴하는 최적화 방법입니다. 도함수와 이차 도함수를 계산해야 하므로 계산 비용이 큽니다.
- **제약 최적화**: 제약 조건을 포함하는 최적화 문제를 해결하는 기법입니다.
    - **Simplex Method**: 선형 계획법(Linear Programming, LP)에서 많이 사용되며, 제약 조건을 만족시키면서 최적해를 찾는 기법입니다.
- **비선형 최적화**: 목적 함수나 제약 조건이 비선형인 경우 사용됩니다.
    - 비선형 문제는 선형 문제보다 해결이 어렵기 때문에 다양한 수치적 기법이나 근사 방법을 사용하여 해결합니다.

6. 계산량 문제와 근사 해법

최적화 문제에서 중요한 것은 정확한 해를 찾는 것뿐만 아니라, **주어진 시간 내에 해를 찾는 것**입니다.

- 예를 들어, 기상 예측에서는 내일의 날씨를 예측하기 위해서 내일이 지나기 전에 결과를 도출해야 합니다. 그렇지 않으면, 아무리 정확한 예측이라도 의미가 없어집니다.
- 이러한 이유로, 계산 시간이 중요한 경우 근사해를 찾아야 하며, 이 과정에서 다양한 **휴리스틱 알고리즘**이나 **메타휴리스틱 기법**이 사용됩니다.

7. 최적화 문제의 설계 변수와 함수 관계

- 최적화 문제에서는 목적 함수와 제약 조건이 설계 변수와 상호 복잡한 관계를 가집니다.
- 예를 들어, 낙하산의 크기 R는 줄의 길이 ℓ, 항력 계수 c, 낙하산의 개수 N 등에 모두 영향을 미칩니다. 따라서, 이 상호 의존적인 관계를 고려하여 설계 변수를 조정해야 합니다.
- **변수 최소화**: 최적화 문제에서는 변수의 수를 최소화하는 것이 중요한데, 이는 문제를 더 단순하게 만들어 효율적으로 해결할 수 있게 해줍니다.