최적제어문제(optimal control problem)

2017103580 사회학과 김정운

1.개요(Introduction)

최적제어문제(Optimal control problem)이란 state x와 이에 영향을 미치는 control u에 대한 미분방정식이 존재할 때, 특정 기준 하에서 state가 최적이 되도록 하는 control를 찾는 문제를 말한다. 이를 추가적인 조건이 없다는 가정하에서 수식으로 작성하면 다음과 같다.

Min J(x)subject to , =a,

이때 J(x)를 cost function이라고 하며, “=a, “를 경계조건(boundary condition)이라고 한다(를 initial time, 를 final time). J(x)는 사용자의 필요에 따라 다양하게 설정할 수 있는데, 대표적으로 와 가 있다. 전자는 mayor problem이라고 하며, 최종결과를 최적화하는데 초점을 둔다. 대표적인 예시로 공정에서 생산량을 극대화하는 것이 있다. 후자는 Bolza problem이라고 하며, mayor와 달리 주어진 구간 전체에서의 최적을 목표로 한다. 특정 시간 동안 움직이는 거리를 최소화하는 문제를 예시로 들 수 있다.

Bolza problem에 대한 접근 방식 중 하나는 Dynamic programming(이하 DP 알고리즘)이 있는데, DP 알고리즘이란 어떤 최적화 문제를 풀 때 보다 작은 규모에서의 최적해를 구하고 이를 활용하여 최종적인 해를 얻는 것을 말한다. 예를 들어, 위치 a에서 출발하여 위치 b로 도착하는데 거리를 최소화한다고 하자. 그러면 a와 b 사이에 있는 임의의 위치 , … 을 설정하고, a와 (k=1…n)의 최적 거리를 얻을 때 a와 lk-1의 정보를 이용한다. 이러한 문제 풀이방법을 DP 알고리즘이라고 한다. 이는 중간에 값을 저장하고 구간을 나눠서 푼다는 점에서, 연산량과 메모리 사용량이 많다. 특히, 최적에 영향을 미치는 변수가 많으면 이를 모두 고려해야 하기 때문에 기하급수적으로 연산량이 높아지는데, 이를 차원의 저주(Curse of dimension)라고 한다.

하지만 DP 알고리즘을 사용해야 하는 상황도 있는데, 그 중 하나는 bolza problem이 불확실성을 포함하는 경우가 있다. 가령 주식에 대한 확률미분방정식과 cost function으로써 포트폴리오가 있으며, t시점 이후의 수익을 최대화하겠다고 가정하자. 그러면 현재 시점을 기준으로 미래에 발생할 수 있는 경우 수를 모두 고려하여 최적의 의사결정을 얻을 것이다. 그러나 이는 미래의 정보를 알지 못한 상황에서 설정된 것이기 때문에, t보다 작은 시점에서 주식과 관련된 정보가 주어지면 이를 바탕으로 의사결정을 갱신해야 한다. DP 알고리즘은 보다 작은 문제에 대한 최적해를 구하기 때문에, 이러한 문제를 쉽게 해결할 수 있다. 이처럼 DP 알고리즘은 시간 구간에서 state의 정보를 지속적으로 수정할 때 사용될 수 있다.

또한 DP 알고리즘을 쓰면 좋은 상황이 있는데, 대표적인 것이 principal of optimality가 있다. 이는 전체 구간에서 최적인 해는 부분 구간에서도 최적이라는 것을 의미하며, 가령 x(t)가 최적해라고 한다면, 그것은 부분 문제에 대해서도 최적이라는 것을 의미한다. 이는 DP 알고리즘의 연산량을 줄일 수 있는데, 어떤 문제가 principal of optimality를 만족한다고 하면 DP 알고리즘에서 저장할 값은 오직 부분 최적해이기 때문이다. principal of optimality의 대우가 “부분해가 최적이 아니면 전체 해도 최적이 아니다”인데, 이는 부분 해 중에서 최적이 아닌 것들을 배제할 수 있음을 의미한다. 이러한 특징은 저장해야 값과 관련된 메모리를 줄일 수 있기 때문에 효율적이다.

또한 Principal of optimality는 markov property와 관련이 있는데, markov property는 미래에 대한 예측은 과거에 상관없이 현재 상태에 의해 결정된다는 것을 의미한다. 이를 수식으로 쓰면 P(x(t+1)|x(t), x(t-1) … x(t0))=P(x(t+1)|x(t))가 된다. 이는 시간 구간이 겹치지 않은 두 부분해 와 가 있을 때, 를 구할 때 의 최종결과만을 고려하면 된다는 것을 의미한다. 즉, t가 final time이 아닐 때의 는 의미가 없기 때문에 고려하지 않아도 된다. 이는 해당문제가 Markov property를 만족한다면, DP 알고리즘의 연산량을 크게 줄일 수 있음을 의미한다. Markov property가 성립하지 않는다면, 고려해야 하는 경우 수가 크게 증가할 수 있기 때문이다. 또한 principal optimality이면 Markov property가 성립한다고 할 수 있는데, 전자는 시간 구간이 겹치지 않으면 초기 조건만을 고려하기 때문이다.