AI알고리즘

비선형계획법

수퍼스타를 키우는 곳

전주대학교

비선형계획법 개요

비선형계획모형 소개

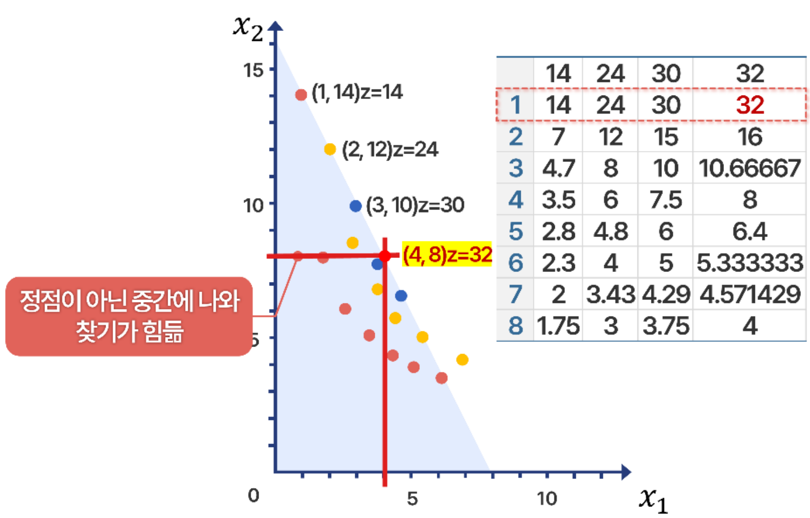
비선형계획모형 소개

❖ 비선형계획문제(Nonlinear Programming Problem)

- 선형계획법이 아니라면 최적해를 찾기 어려움
- 앞서 소개한 최적화 기법들은 목적 함수 및 제약식 모두 선형 함수(Linear Function)를 가정
- 정수계획법에서 다룬 모형 역시 정수해 조건을 제외한 나머지는 모두 선형 함수를 가정
- 현실 문제에서는 다양한 비선형 관계(Non-linear Relation)가 존재
- 선형이 아닌 비선형으로 표현된 최적화문제를 비선형계획문제라고 함
- 최적해를 찾는 가장 좋은 해법이 없음

선형이 아니라는 의미

$$
\begin{aligned}
&Max\ \ x_1 x_2 \\
&s.t. \\
&\qquad 2x_1 + \ x_2\ \le 16 \\
&\qquad\ x_1,\, x_2 \ge 0
\end{aligned}
$$



	14	24	30	32
1	14	24	30	32
2	7	12	15	16
3	4.7	8	10	10.66667
4	3.5	6	7.5	8
5	2.8	4.8	6	6.4
6	2.3	4	5	5.333333
7	2	3.43	4.29	4.571429
8	1.75	3	3.75	4

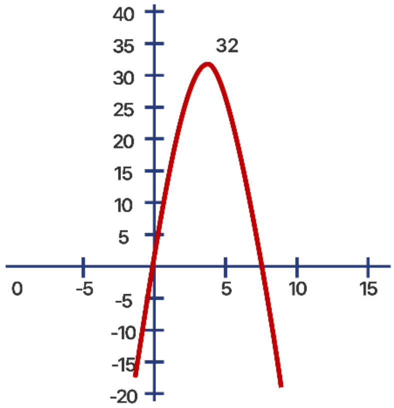
- 실수를 이어 놓으면 $1/x$ 형태의 함수의 형태가 쭉 그려질 것
- 점점 밖으로 나가는 형태
- 반원 형태를 가지고 있음
- 실행 가능 영역 내에서 가장 좋은 걸 찾는 것

직선을 만족하는 x_1, x_2에서 최적해가 일어나기 때문

$$
\begin{aligned}
2x_1 + \ x_2 &= 16 \\
x_2 &= 16 - 2x_1
\end{aligned}
$$

비선형계획모형 소개

❖ 비선형계획문제(Nonlinear Programming Problem)

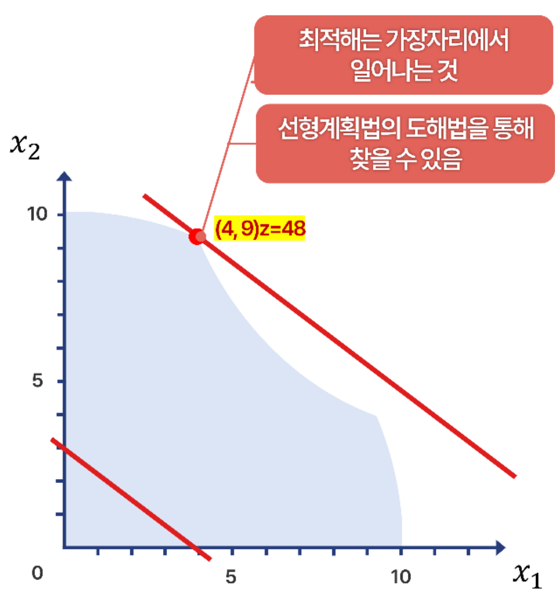


$$Max\ \ x_1\, x_2$$

$$\begin{aligned} x_1\, x_2 &= x_1(16-2x_1) \\ &= -2x_1^2 + 16x_1 \\ &= -2(x_1^2 - 8x_1) \\ &= -2[(x_1-4)^2 - 16] \\ &= -2(x_1-4)^2 + 32 \end{aligned}$$

- 경계선상에 있다는 것을 알면 다른 함수 형태로 문제를 풀 수 있음
  → 문제를 이해하고, 최적해를 찾으려 노력
  → 기존의 알고리즘을 적용, 수정하고 새로운 알고리즘을 제작

$$\begin{aligned} &Max\ \ 3x_1 + 4x_2 \\ &s.t. \\ &\qquad x_1^2 + x_2^2 \ \le 97 \\ &\qquad x_1 x_2 \le\ 36 \\ &\qquad x_1,\, x_2 \ge 0 \end{aligned}$$



- 변수가 3개 이상인 경우
  - 실행 가능 영역을 그릴 수 없음
  - 경계를 실제로 알 수 없음
  - 심플렉스법처럼 정점을 찾으면서 최적해를 찾을 수 없음

비선형계획모형 소개

❖ 선형계획문제와의 차이점

- 볼록집합(Convex Set)
  - 선형계획문제의 실행가능영역은 볼록집합
  - 목적함수가 선형 함수이기 때문에, 선형계획모형에서 최적해는 실행가능해영역의 꼭짓점(Extreme Point)에 존재
- 최적화 알고리즘의 부재
  - 휴리스틱 알고리즘
    - 최적해는 만들어내지 못하지만 최적해에 가까운 해 도출 가능
    - 비선형계획법에는 가장 좋은 알고리즘은 없음

비선형계획법 개요

볼록성과 오목성

볼록성과 오목성

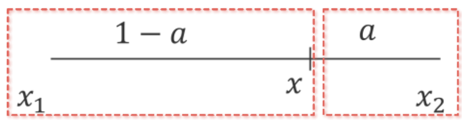
❖ 선분

- 두 점을 이은 곧은 선
- 직선: 두 점을 곧게 이은 선을 한없이 늘인 모든 선

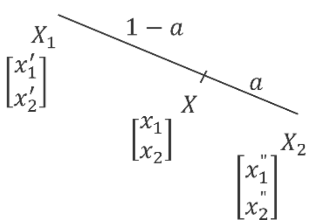
❖ 1차원

$$x = ax_1 + (1-a)x_2, 0 \le a \le 1$$

$$a = 0 \text{ 일 때}, x = x_2$$



❖ 2차원

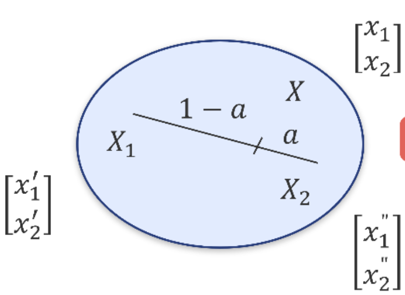


$$\begin{bmatrix}x_1\\x_2\end{bmatrix} = a\begin{bmatrix}x_1'\\x_2'\end{bmatrix} + (1-a)\begin{bmatrix}x_1''\\x_2''\end{bmatrix}$$

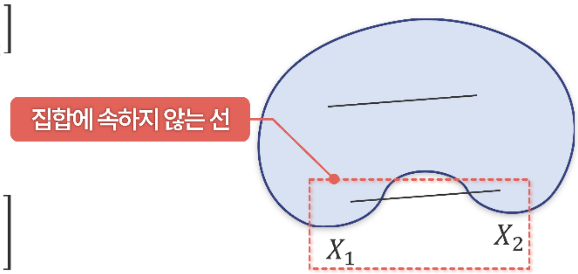
$$= \begin{bmatrix}ax_1' + (1-a)x_1''\\ax_2' + (1-a)x_2''\end{bmatrix}$$

❖ 볼록집합(Convex Set)

- 집합 내 두 점을 연결한 선분이 항상 집합에 포함되면 볼록집합
  - 어떤 집합의 두 점을 선분으로 이용 → 선분상의 모든 점들이 집합에 속하면 블록집합



convex set

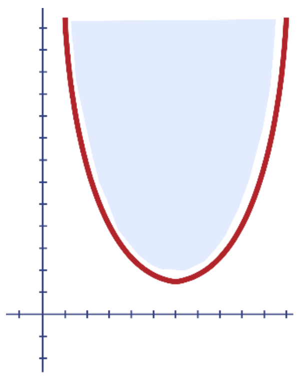


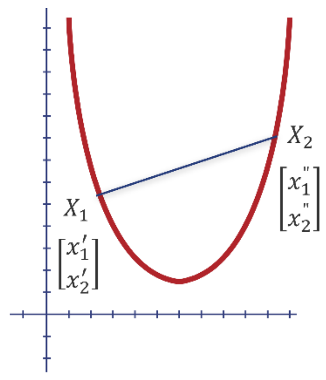
Non-convex set

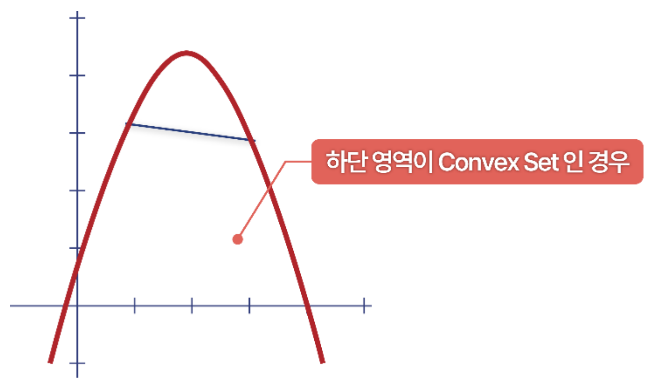
볼록성과 오목성

❖ 볼록함수(Convex Function)

- 그래프의 위쪽 영역이 볼록집합(Convex Set)인 함수





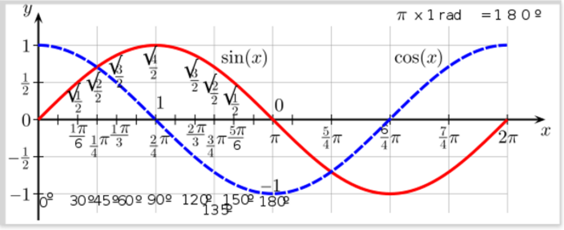


Concave function

볼록성과 오목성

❖ 테일러 전개(Tayor Expansion)

- 공간이동을 통해 최적의 방법을 구하는 것



- 특정 점을 기준으로 함수를 근사하는 방법
- $f(x) = f(a) + f'(a)(x - a)$

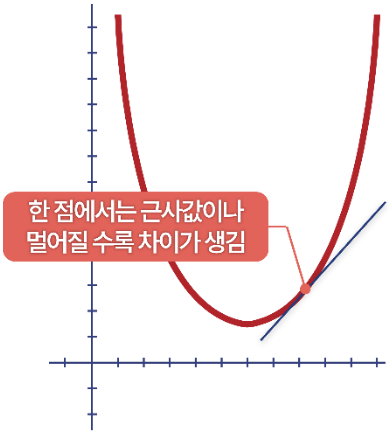
$$f(x) = (x-6)^2 + 2$$

$a = 9$에서 접선

$$f(x) = f(9) + f'(9)(x-9) = 6x - 43$$

$$f(x_2) \geq f(x_1) + f'(x_1)(x_2 - x_1)$$

$$f''(x) \geq 0$$



Concave function

- 미분(기울기)
- 적분(면적)
- 거리/시간=속도
- 속도/시간=가속도
  - 2차 도함수