



## 심플렉스 방법 (Simplex Method)

선형 계획법 문제를 푸는 한가지 방법

2019년 03월 02일

### Standard Form 과 Slack Form

선형 계획법 문제에서 Standard form 은 다음과 같은 조건을 만족하는 문제를 일컫습니다.

- 목적함수(Objective function)를 최대화하는 문제입니다.
- 조건식은  $\alpha_1 x_1 + \dots + \alpha_n x_n \leq \beta$  와 같은 형태를 가집니다.
- 결정변수(Decision variable)들은 음수가 아닙니다.

Standard Form을 만족하는 한가지 예시를 살펴보겠습니다.

$$\begin{aligned}
 \max_{x_1, x_2} \quad & x_1 + 3x_2 \\
 \text{subject to} \quad & x_1 + 2x_2 \leq 4 \\
 & 4x_1 + x_2 \leq 4 \\
 & -x_1 + x_2 \leq \frac{3}{2} \\
 & x_1 \geq 0, x_2 \geq 0
 \end{aligned}$$

Slack form은 부등식을 등식으로 만들기 위해 느슨한 변수(Slack variable)를 도입하는 형식입니다. 위의 최적화 문제에 Slack variable  $s_1, s_2, s_3$ 을 추가해 Slack form 형태의 문제로 만들어 보겠습니다.



$$\begin{aligned}
 4x_1 + x_2 + s_2 &= 4 \\
 -x_1 + x_2 + s_3 &= \frac{3}{2} \\
 x_1 \geq 0, x_2 \geq 0, \\
 s_1 \geq 0, s_2 \geq 0, s_3 \geq 0
 \end{aligned}$$

## 심플렉스 방법

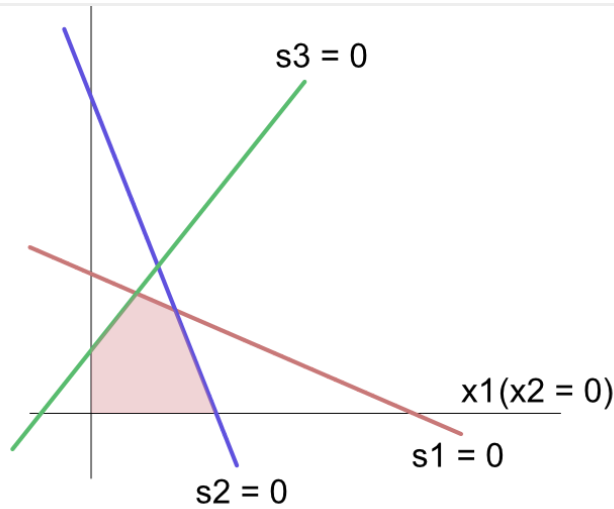
심플렉스는 LPP (Linear programming problem)을 푸는 한가지 방법입니다. 심플렉스 방법의 원리와 그 풀이법을 아래 최적화 문제 예시와 함께 살펴보겠습니다.

$$\begin{aligned}
 \max_{x_1, x_2} \quad & x_1 + 3x_2 \\
 \text{subject to} \quad & x_1 + 2x_2 \leq 4 \\
 & 4x_1 + x_2 \leq 4 \\
 & -x_1 + x_2 \leq \frac{3}{2} \\
 & x_1 \geq 0, x_2 \geq 0
 \end{aligned}$$

심플렉스 방법은 Standard form 으로 제시된 최적화 문제를 Slack form 으로 고치는 것부터 시작합니다.

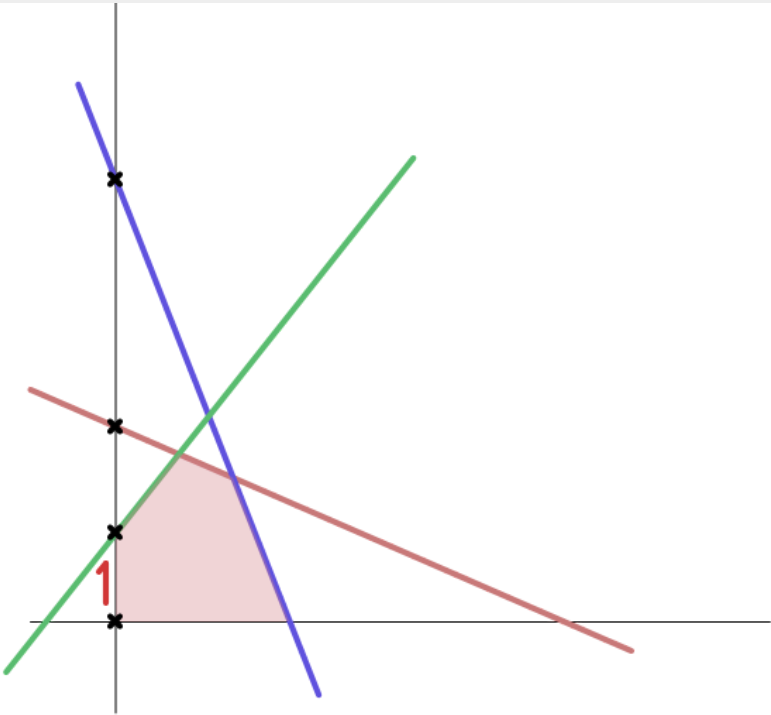
$$\begin{aligned}
 \max_{x_1, x_2} \quad & x_1 + 3x_2 \\
 \text{subject to} \quad & x_1 + 2x_2 + s_1 = 4 \\
 & 4x_1 + x_2 + s_2 = 4 \\
 & -x_1 + x_2 + s_3 = \frac{3}{2} \\
 & x_1 \geq 0, x_2 \geq 0, \\
 & s_1 \geq 0, s_2 \geq 0, s_3 \geq 0
 \end{aligned}$$

위 문제는 아래와 같이 좌표 평면에 그려볼 수 있습니다.

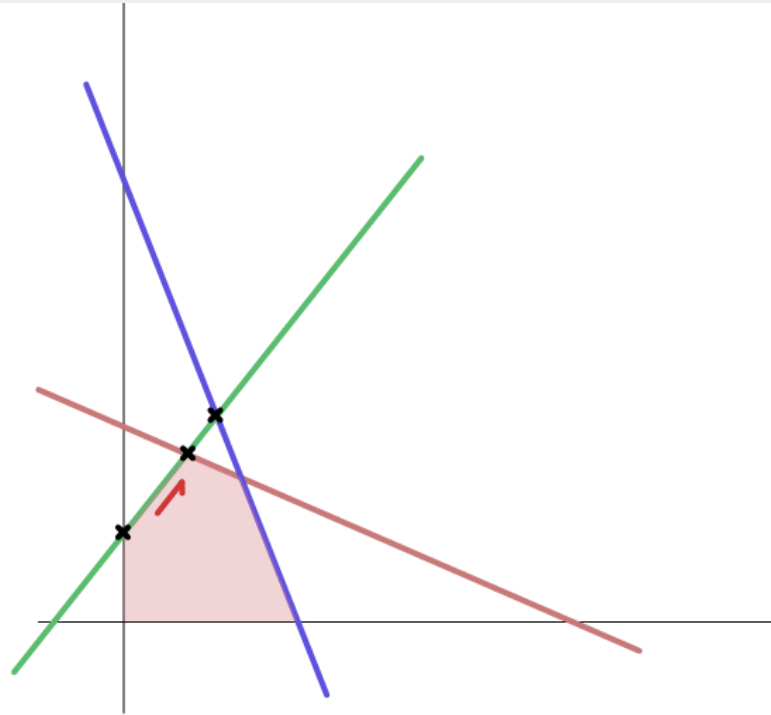


좌표 평면에서  $s_i$ 는 Slack variable 을 의미합니다. 붉게 칠해진 다각형은 제약식을 만족하는 범위입니다. 직관적으로 목적함수  $x_1 + 3x_2$  는 다각형의 어느 꼭지점을 지날 때 최대값을 가집니다. 심플렉스 방법은 최댓값을 찾기 위해  $(0, 0)$ 에서 시작해 제약식 범위를 만족하는 꼭지점을 찾아 반복적으로 이동합니다. 반복의 첫번째 회차에서 시작점은  $(0, 0)$ 이고 알고리즘은  $x_1$  축 또는  $x_2$  축 방향 중 하나를 선택해야 합니다. 이 때 선택의 기준은 목적 함수  $x_1 + 3x_2$ 의 계수입니다.  $x_1$ 의 계수는 1이고  $x_2$ 의 계수는 3이므로 계수가 큰  $x_2$  방향으로 이동하는 것이 최댓값을 찾는 데 더 효율적입니다.

방향은 결정했지만 얼마나 이동할지 아직 정하지 않았습니다.  $x_2$  축이  $s_3 = 0$ 와 만나는 점,  $s_1 = 0$ 과 만나는 점,  $s_2 = 0$ 과 만나는 점이 후보이고 이 중 거리가 적은 꼭지점이 제약식을 만족하는 점입니다. 만약 거리가 먼  $s_1 = 0$  이  $x_2$ 축과 만나는 점을 선택하면  $s_3 < 0$ 이 되고 제약식을 만족하지 않습니다. ( $\because$  식  $x_2 = x_1 + \frac{3}{2} - s_3$ 에서  $s_3 < 0$ 는  $x_2$  절편이 증가함을 의미하므로)  
이를 좌표 평면에 표현하면 다음과 같습니다.



두번째 이터레이션에서  $s_3 = 0$  직선을 따라 이동하고 그 거리는  $s_3 = 0$ 과  $s_1 = 0$ 이 만나는 꼭지점까지입니다.  
아래 그림은 두번째 이터레이션을 표현한 것입니다.



알고리즘은 두번째 이터레이션에서 도착한 꼭지점을  $x_1 + 3x_2$ 의 최적의 솔루션으로 판정하고 종료합니다. 그림을 통해 직관적으로 이해한 것을 직접 계산하면서 살펴보겠습니다. 우선 제약식과 목적 함수를 아래와 같이 정리합니다.

$$\begin{aligned}x_1 + 2x_2 + s_1 &= 4 \\4x_1 + x_2 + s_2 &= 4 \\-x_1 + x_2 + s_3 &= \frac{3}{2} \\p - x_1 - 3x_2 &= 0\end{aligned}$$

변수  $p$ 는 목적 함수의 최댓값을 뜻합니다.

위 식을 아래와 같이 표로 정리합니다.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$p$	
1	2	1	0	0	0	4
4	1	0	1	0	0	4
-1	1	0	0	1	0	3/2



-1	-3	0	0	0	1	0
----	----	---	---	---	---	---

이는 아래와 같은 단순한 행렬-벡터 곱셈과 같습니다. 이 때 곱해지는 벡터의 첫번째, 두번째 엔트리를 보면 알고리즘이 (0,0)부터 시작함을 알 수 있습니다.

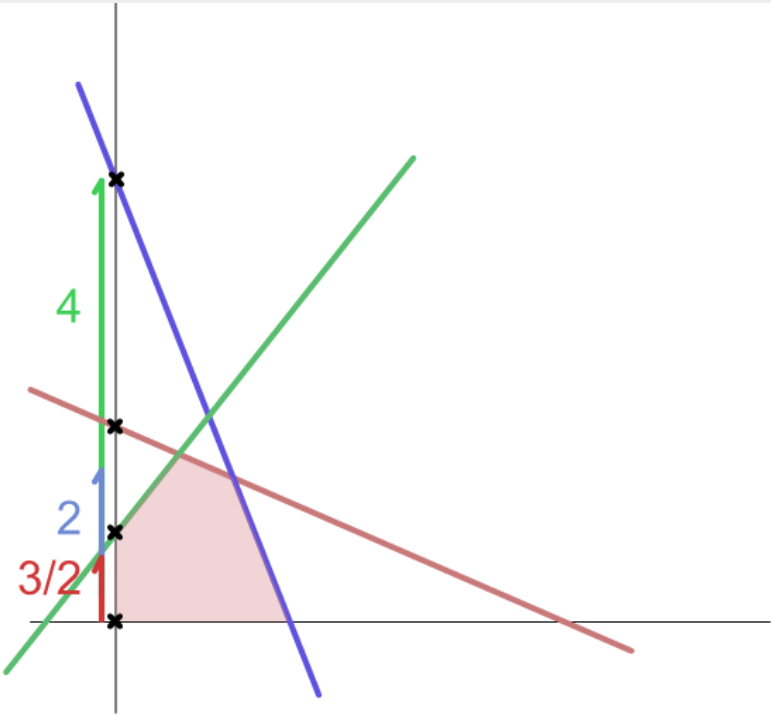
$$\begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ -1 & -3 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 = 0 \\ x_2 = 0 \\ s_1 = 4 \\ s_2 = 4 \\ s_3 = 3/2 \\ p = 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 3/2 \\ 0 \end{bmatrix}$$

알고리즘의 반복은 언제 종료될까요? 결론부터 말씀드리면 표의 마지막 행의 엔트리가 모두 음수가 아닐 때까지 반복합니다. 음수 엔트리가 존재한다는 것은 아직 최댓값을 만족하지 못했고 가야할 꼭지점이 있다는 것을 직관적으로 의미합니다.

목적 함수의 최댓값을 찾기 위해 첫번째 이터레이션을 수행해 보겠습니다. 이전에 알고리즘이 목적 함수  $x_1 + 3x_2$  에서 계수가 큰  $x_2$ 를 다음 꼭지점을 찾아갈 방향으로 선택함을 보았습니다. 이는 표의 마지막 행에서 두번째 엔트리인 -3을 선택함을 의미합니다. 이제  $x_2$  축 방향으로 가장 가까운 꼭지점을 찾아가는 과정이 남았습니다. 이는  $x_2$  열의 엔트리들을 대응하는 우변 상수 즉, 마지막 열에 대해 나눈 값으로 판단합니다. 계산 결과는 다음과 같습니다.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$p$		ratio
1	2	1	0	0	0	4	$4 / 2 = 2$
4	1	0	1	0	0	4	$4 / 1 = 4$
-1	1	0	0	1	0	3/2	$3/2 / 1 = 3 / 2$
-1	-3	0	0	0	1	0	

ratio 값을 좌표 평면에 표현하면 아래와 같습니다.



가장 가까운 꼭지점을 찾아가기 위해 ratio 가 가장 작은 행을 선택합니다. 이 행을 피벗행이라 합니다. 나머지 행들을 피벗행에 대해 기본행연산해 아래와 같은 표로 재구성합니다.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$p$	
3	0	1	0	-2	0	1
5	0	0	1	-1	0	5/2
-1	1	0	0	1	0	3/2
-4	0	0	0	3	1	9/2

이로써 첫번째 이터레이션 수행이 끝났습니다. 위 표를 아래와 같은 행렬-벡터 방정식으로 다시 쓸 수 있습니다.



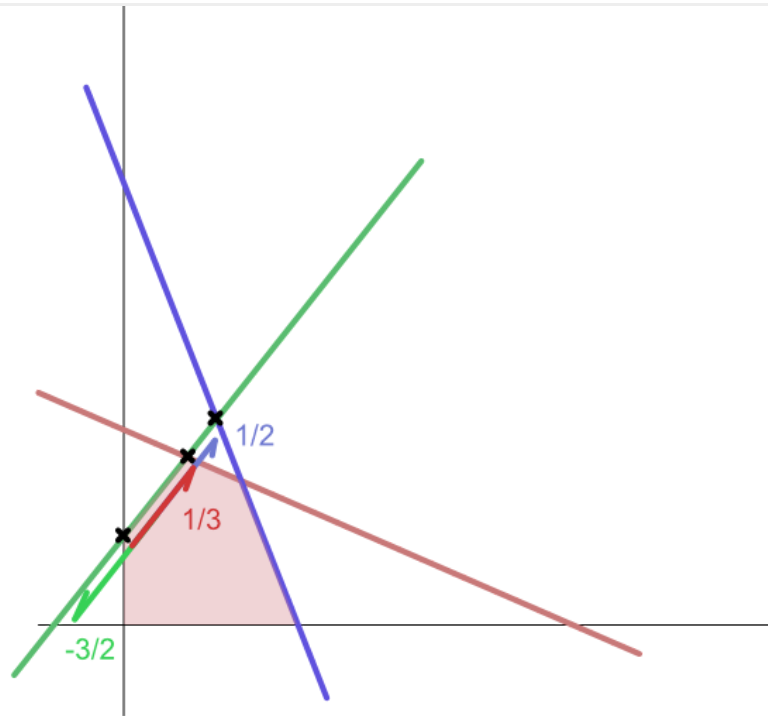
$$\begin{bmatrix} -1 & 1 & 0 & 0 & 1 & 0 \\ -4 & 0 & 0 & 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} s_2 = 5/2 \\ s_3 = 0 \\ p = 9/2 \end{bmatrix} \begin{bmatrix} 3/2 \\ 9/2 \end{bmatrix}$$

두번째 이터레이션을 위해 다시 마지막행을 살펴보겠습니다. 첫번째 엔트리가 -4로 음수이고 각 행들의 ratio 를 구하면 다음과 같이 정리할 수 있습니다.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$p$		ratio
3	0	1	0	-2	0	1	1/3
5	0	0	1	-1	0	5/2	5/2 / 5 = 1/2
-1	1	0	0	1	0	3/2	3/2 / (-1) = -3/2
-4	0	0	0	3	1	9/2	

ratio 값을 좌표 평면에 표현하면 아래와 같습니다.





ratio 값이 음수가 나온 것은 위 그림에서 볼 수 있듯 제약식 조건에 역행하는 방향으로 감을 의미합니다. 따라서 음수인 ratio 를 가진 행은 제외합니다. ratio 값이 가장 작은 첫번째 행을 피벗행으로 선택하고 기본행연산하면 다음 표를 얻습니다.

$x_1$	$x_2$	$s_1$	$s_2$	$s_3$	$p$	
1	0	1/3	0	-2/3	0	1/3
0	0	-5/3	1	7/3	0	5/6
0	1	1/3	0	1/3	0	11/6
0	0	4/3	0	1/3	1	35/6

마지막 행의 엔트리가 모두 양수이므로 알고리즘을 종료합니다. 이를 행렬-벡터 곱셈으로 표현하면 아래와 같습니다.



$$\left[ \begin{array}{cccccc} 0 & 1 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 4/3 & 0 & 1/3 & 1 \end{array} \right] \left[ \begin{array}{l} s_2 = 5/6 \\ s_3 = 0 \\ p \end{array} \right] \left[ \begin{array}{l} 11/6 \\ 35/6 \end{array} \right]$$

위 식을 계산하면  $x_1 = \frac{1}{3}$ ,  $x_2 = \frac{11}{6}$ ,  $p = \frac{35}{6}$  을 얻습니다. 두 직선의 교점을 이용해 이를 계산해 보겠습니다. 제약식  $x_1 + 2x_2 = 4$ 와  $-x_1 + x_2 = \frac{3}{2}$  의 교점은  $(\frac{1}{3}, \frac{11}{6})$  입니다. 따라서 목적 함수  $x_1 + 3x_2$  의 최댓값은  $\frac{35}{6}$  이고 이 때  $x_1 = \frac{1}{3}$ ,  $x_2 = \frac{11}{6}$  입니다.

## 참고

- [www.youtube.com/watch?v=ORn1MVC2gg4](https://www.youtube.com/watch?v=ORn1MVC2gg4)
- [www.youtube.com/watch?v=2eCdWJ59iI4](https://www.youtube.com/watch?v=2eCdWJ59iI4)



최적화 문제



# 선형대수학



G

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

2

Share

Best Newest Oldest

Be the first to comment.

Subscribe

Privacy

Do Not Sell My Data