


PENGAMBILAN KEPUTUSAN MULTIKRITERIA

Program linier hanya mempunyai satu tujuan, apakah maksimasi atau minimisasi. Namun, seringkali perusahaan atau organisasi mempunyai lebih dari satu tujuan di samping maksimisasi laba atau minimisasi biaya itu. Pada kenyataannya, perusahaan atau organisasi sering mempunyai beberapa kriteria untuk pengambilan keputusan. Kasus-kasus seperti ini digolongkan sebagai kasus-kasus pengambilan keputusan multikriteria.



Sebagai contoh, di samping memaksimumkan laba, sebuah perusahaan bisa saja mempunyai tujuan mencegah pengurangan tenaga kerja atau mengurangi polusi. Model-model pengambilan keputusan multikriteria ini dapat diselesaikan dengan teknik *goal programming* ataupun *analytical hierarchy process* (AHP).

GOAL PROGRAMMING

Goal Programming adalah salah satu variasi dari program linier yang mempunyai lebih dari satu tujuan pada fungsi objektifnya. Model *goal programming* dibentuk dengan format umum yang sama dengan program linier dengan sebuah fungsi objektif dan kendala-kendala linier. Penyelesaiannya juga sangat mirip dengan program linier.

FORMULASI MODEL

Misalkan sebuah perusahaan mempunyai sebuah program linier sebagai berikut:

$$\text{Max } Z = 40x_1 + 50x_2$$

Dengan kendala:

$$x_1 + 2x_2 \leq 40 \rightarrow \text{kendala tenaga kerja}$$

$$4x_1 + 3x_2 \leq 120 \rightarrow \text{kendala bahan baku}$$

$$x_1, x_2 \geq 0$$


x_1 adalah jumlah produk 1 yang akan diproduksi

x_2 adalah jumlah produk 2 yang akan diproduksi

- Fungsi objektif Z menunjukkan laba (dalam \$) yang akan diperoleh dari produk 1 dan produk 2, yakni \$40 per unit untuk produk 1, dan \$50 per unit untuk produk 2
- Kendala pertama menunjukkan bahwa untuk memproduksi 1 unit produk 1 diperlukan 1 jam tenaga kerja dan 4 kg bahan baku, sementara untuk memproduksi 1 unit produk 2 diperlukan 2 jam tenaga kerja dan 3 kg bahan baku. Tenaga kerja yang tersedia terbatas sebanyak 40 jam dan bahan baku terbatas sebanyak 120 kg.
- Ini adalah standar model program linier.

Sekarang, andaikan perusahaan juga mempunyai tujuan lain di samping memaksimalkan laba itu, dengan urutan kepentingan sebagai berikut:

1. Untuk mencegah pengangguran perusahaan tidak ingin menggunakan tenaga kerja kurang dari 40 jam per hari.
2. Perusahaan ingin memperoleh laba yang memuaskan sebesar \$1600 per hari.
3. Oleh karena bahan baku harus disimpan di tempat khusus, maka perusahaan menghendaki untuk tidak menyimpan persediaan bahan baku lebih dari 120 kg per hari.
4. Perusahaan juga ingin agar jam lembur seminimal mungkin.



Tujuan-tujuan perusahaan yang berbeda-beda ini disebut sebagai goal perusahaan dalam konteks *goal programming*. Tentu saja perusahaan ingin mencapai sebanyak mungkin goal ini.

Goal Tenaga Kerja

Untuk mencegah pengangguran tenaga kerja, kendala perusahaan dirumuskan sebagai berikut:

$$x_1 + 2x_2 + d_1^- - d_1^+ = 40 \text{ jam}$$

kendala ini disebut dengan kendala goal.

d_1^- dan d_1^+ disebut variabel penyimpangan (*deviational variables*)

d_1^- = jumlah jam kerja kurang dari 40 (pengangguran)

d_1^+ = jumlah jam kerja lebih dari 40 jam (lembur)

Goal tercapai apabila kendala = 40 jam

Sebagai contoh, jika $x_1 = 5$ dan $x_2 = 10$, maka sejumlah 25 jam kerja telah dipakai. Substitusi harga-harga ini ke dalam .. kendala *goal* menghasilkan:

$$(5) + 2(10) + d_1^- - d_1^+ = 40$$

$$25 + d_1^- - d_1^+ = 40$$

Oleh karena hanya 25 jam yang dipakai untuk produksi, maka tenaga kerja menganggur sebanyak 15 jam ($40 - 25 = 15$). Jadi, bila kita buat $d_1^- = 15$ jam dan $d_1^+ = 0$ (karena jelas tidak ada lembur), kita peroleh:

$$25 + d_1^- - d_1^+ = 40$$

$$25 + 15 - 0 = 40$$

$$40 = 40$$

Sekarang, perhatikan kasus di mana $x_1 = 10$ dan $x_2 = 20$. Ini berarti bahwa jam kerja yang dibutuhkan adalah 50 jam. Artinya, 10 jam di atas level *goal* yang 40 jam itu. Kelebihan 10 jam ini adalah lembur. Jadi $d_1^- = 0$ (karena tidak ada pengangguran) dan $d_1^+ = 10$ jam.

Dari kedua contoh ini, paling tidak satu variabel penyimpangan adalah 0. Pada contoh pertama $d_1^+ = 0$, dan pada contoh kedua $d_1^- = 0$. Hal ini disebabkan oleh kenyataan bahwa tidak mungkin menggunakan jam kerja kurang dari 40 jam dan lebih dari 40 jam pada waktu yang bersamaan. Tentu saja kedua variabel d_1^- dan d_1^+ dapat sama dengan nol bila jam yang dipakai untuk produksi persis 40 jam.

Artinya, salah satu karakteristik fundamental dari *goal programming* adalah: paling tidak satu dari variabel penyimpangan di dalam kendala *goal* mestilah berharga sama dengan nol.

Langkah berikutnya dalam memformulasikan model *goal programming* ini adalah untuk menyatakan *goal* untuk tidak menggunakan kurang dari 40 jam kerja. Untuk ini kita membuat sebuah bentuk baru fungsi objektif, yakni:

Minimumkan $P_1 d_1^-$

Baca : minimumkan d_1^-

Fungsi objektif pada semua model *goal programming* adalah meminimumkan deviasi dari level kendala *goal*. Pada fungsi objektif ini *goal*-nya adalah untuk meminimumkan d_1^- , yakni kekurangterpakaian pekerja. Jika d_1^- sama dengan nol, maka artinya kita tidak akan menggunakan kurang dari 40 jam tenaga kerja. Jadi, tujuan kita adalah membuat d_1^- sama dengan nol atau seminimum mungkin. Simbol P_1 dalam fungsi objektif menunjukkan bahwa meminimumkan harga d_1^- adalah *goal* dengan prioritas pertama. Hal ini berarti bahwa bila model ini diselesaikan, langkah pertama adalah meminimumkan harga d_1^- sebelum *goal* lainnya ditangani.

Goal prioritas keempat dalam masalah ini juga berkaitan dengan kendala tenaga kerja. Prioritas keempat, P_4 , menunjukkan keinginan untuk meminimumkan jam lembur, yang telah dinyatakan di atas dengan variabel d_1^+ , oleh sebab itu fungsi objektif menjadi:

$$\text{Min } P_1 d_1^-, P_4 d_1^+$$

Baca: minimalkan d_1^- dan d_1^+

Seperti sebelumnya, fungsi objektif adalah meminimumkan variabel penyimpangan yakni d_1^+ . Dengan kata lain, bila d_1^+ sama dengan nol, tidak ada jam lembur yang diperlukan sama sekali. Dalam memecahkan masalah ini, goal jajaran keempat ini tidak akan diselesaikan sebelum goal pertama, kedua, dan ketiga diperhatikan.

Goal Laba

Goal kedua dalam model *goal programming* ini adalah untuk mencapai laba \$1600. Ingat bahwa fungsi objektif asli pada program linier adalah $Z = 40x_1 + 50x_2$. Sekarang kita reformulasi fungsi objektif ini sebagai sebuah kendala goal dengan level goal sebagai berikut:

$$40x_1 + 50x_2 + d_2^- - d_2^+ = 1600$$

Fungsi objektif meminimumkan d_2^-

Variabel penyimpangan d_2^- menyatakan jumlah laba kurang dari \$1600 dan d_2^+ menyatakan jumlah laba lebih dari \$1600. Goal perusahaan untuk mencapai laba sebesar \$1600 dinyatakan dalam fungsi objektif sebagai:

$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_4 d_1^+$$

Perhatikan bahwa hanya d_2^- yang diminimumkan, tidak d_2^+ , karena logis sekali untuk menganggap bahwa perusahaan pasti akan menerima bila laba melebihi \$1600, artinya tidak ada keinginan untuk meminimumkan kelebihan laba.

Dengan meminimumkan d_2^- pada level prioritas kedua, perusahaan berharap d_2^- akan menjadi nol, yang berarti paling sedikit laba sebesar \$1600 akan diperoleh.

Goal Mengenai Bahan Baku

Goal ketiga dari perusahaan adalah mencegah tersimpannya lebih dari 120 kg bahan baku setiap hari. Kendala goal adalah:

$$4 x_1 + 3 x_2 + d_3^- - d_3^+ = 120 \text{ kg}$$

Oleh karena variabel penyimpangan d_3^- menyatakan jumlah bahan baku kurang dari 120 kg per hari dan d_3^+ menyatakan jumlah melebihi 120 kg, goal ini dapat ditunjukkan dalam bentuk fungsi objektif sebagai berikut:

$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_3 d_3^+, P_4 d_1^+$$

Suku $P_3 d_3^+$ menunjukkan keinginan perusahaan untuk meminimumkan d_3^+ , yakni jumlah kelebihan dari 120 kg. P_3 menyatakan bahwa goal ini adalah prioritas ketiga untuk dicapai oleh perusahaan.

Dengan demikian, secara lengkap model *goal programming* contoh ini adalah:

$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_3 d_3^+, P_4 d_1^+$$

dengan kendala:

$$x_1 + 2x_2 + d_1^- - d_1^+ = 40$$

$$40x_1 + 50x_2 + d_2^- - d_2^+ = 1600$$

$$4x_1 + 3x_2 + d_3^- - d_3^+ = 120$$

$$x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ \geq 0$$

Bentuk-Bentuk Alternatif *Goal Programming*

Misalkan sekarang kita merubah model *goal programming* sebelum ini sedemikian rupa sehingga goal prioritas keempat kita adalah membatasi lembur hingga 10 jam, bukan meminimisasikan lembur. Kendala untuk tenaga kerja pada model *goal programming* sebelumnya adalah:

$$x_1 + 2x_2 + d_1^- - d_1^+ = 40 \text{ jam}$$

Pada kendala goal ini, d_1^+ menyatakan lembur. Oleh karena goal prioritas keempat yang baru adalah membatasi lembur hingga 10 jam, maka kita buat kendala goal sebagai berikut:

$$d_1^+ + d_4^- - d_4^+ = 10$$

Meskipun kendala goal ini kelihatannya tak biasa, pada *goal programming* mempunyai sebuah persamaan yang semuanya terdiri dari variabel penyimpangan adalah hal yang dapat diterima. Pada persamaan ini d_4^- adalah jumlah jam lembur yang kurang dari 10 jam, dan d_4^+ adalah jumlah jam lembur yang melebihi 10 jam. Oleh karena perusahaan membatasi jam lembur hingga 10 jam, d_4^+ diminimumkan pada fungsi objektif:

$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_3 d_3^+, P_4 d_4^+$$

Sekarang mari kita perhatikan tambahan goal prioritas kelima untuk contoh ini. Misalkan perusahaan mempunyai ruangan yang terbatas sehingga hanya dapat memproduksi 30 produk-1 dan 20 produk-2 per hari. Jika memungkinkan, perusahaan ingin memproduksi sebanyak ini. Namun, karena laba bagi produk-2 lebih besar daripada produk-1 (\$50 banding \$40), tentulah lebih penting untuk mencapai goal bagi produk-2. Goal kelima ini memerlukan dua kendala goal baru dalam formulasi, yakni:

$$x_1 + d_5^- = 30$$

$$x_2 + d_6^- = 20$$

Perhatikan bahwa variabel penyimpangan d_5^+ dan d_6^+ dihapus dari kendala-kendala goal ini. Hal ini diakibatkan oleh pernyataan dari goal kelima, yakni "tidak lebih dari 30 produk-1 dan 20 produk-2" yang dapat diproduksi. Dengan kata lain, penyimpangan positif, atau over produksi tidaklah memungkinkan.

Oleh karena goal sebenarnya dari perusahaan adalah mencapai level produksi yang ditunjukkan oleh dua kendala goal ini, variabel penyimpangan negatif d_5^- dan d_6^- haruslah diminimalkan pada fungsi objektif sebagai berikut:

$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_3 d_3^+, P_4 d_4^+, 4 P_5 d_5^- + 5 P_5 d_6^-$$

Oleh karena goal untuk produk-2 lebih penting daripada goal untuk produk-1, maka derajat kepentingan haruslah sebanding dengan jumlah laba (\$50 untuk produk-2 dan \$40 untuk produk-1). Maka goal untuk produk-2 lebih penting daripada goal untuk produk-1 dengan perbandingan 5 banding 4.

Koefisien 5 untuk $P_5 d_6$ dan 4 untuk $P_5 d_5$ disebut dengan pemberat. Dengan kata lain, minimisasi dari d_6 diberi bobot yang lebih tinggi daripada minimisasi d_5 pada level prioritas kelima. Ketika model diselesaikan, pencapaian goal untuk meminimumkan d_6 (produk-2) akan dicoba lebih dahulu sebelum pencapaian goal untuk meminimisasikan, meskipun kedua goal mempunyai level prioritas yang sama. Perhatikan pula bahwa kedua goal ini dijumlahkan karena keduanya berada pada level prioritas yang sama.

Goal programming yang lengkap untuk kasus ini adalah:

$$\text{Min } P_1d_1^-, P_2d_2^-, P_3d_3^+, P_4d_4^+, 4P_5d_5^- + 5P_5d_6^-$$

Dengan kendala:

$$\begin{aligned}x_1 + 2x_2 + d_1^- - d_1^+ &= 40 \\40x_1 + 50x_2 + d_2^- - d_2^+ &= 1600 \\4x_1 + 3x_2 + d_3^- - d_3^+ &= 120 \\d_1^+ + d_4^- - d_4^+ &= 10 \\x_1 + d_5^- &= 30 \\x_2 + d_6^- &= 20\end{aligned}$$

$$x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_4^+, d_5^-, d_6^- \geq 0$$

Interpretasi Grafis Goal Programming

- Untuk memahami lebih jauh mengenai goal programming, mari kita lihat bagaimana interpretasi grafis dari sebuah goal programming.

Perhatikan model *goal programming* berikut ini:

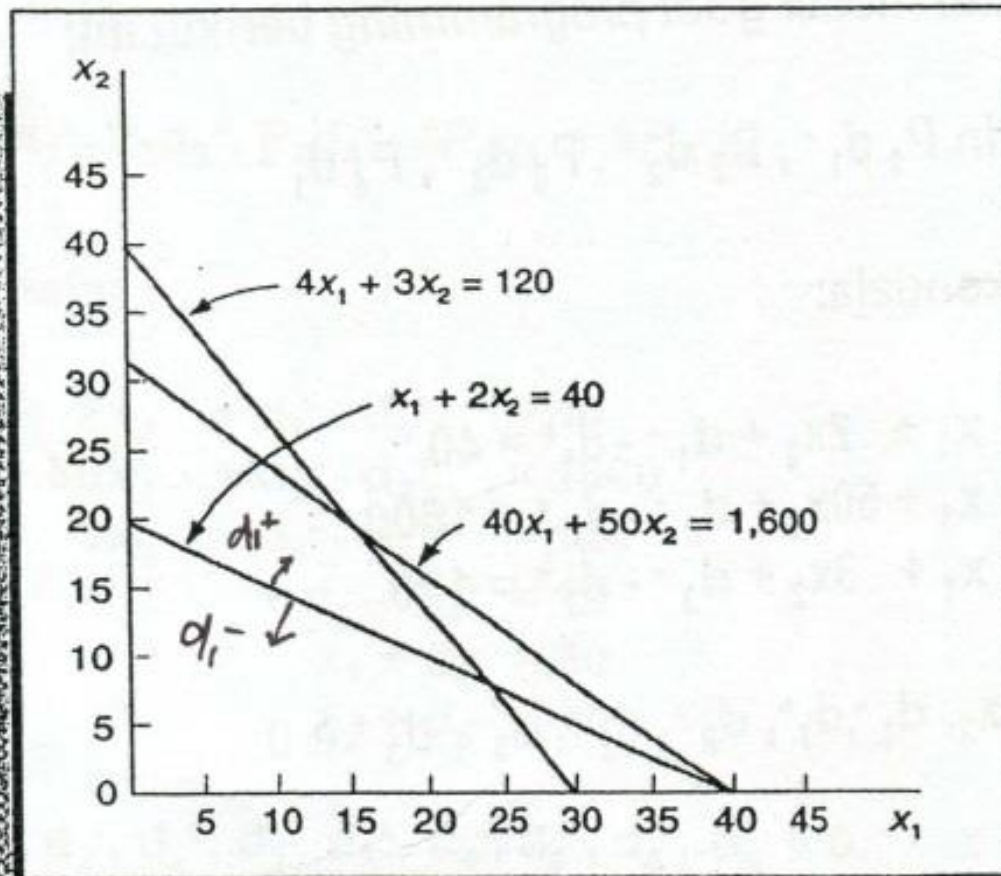
$$\text{Min } P_1 d_1^-, P_2 d_2^-, P_3 d_3^+, P_4 d_1^+$$

dengan kendala:

$$\begin{aligned}x_1 + 2x_2 + d_1^- - d_1^+ &= 40 \\40x_1 + 50x_2 + d_2^- - d_2^+ &= 1600 \\4x_1 + 3x_2 + d_3^- - d_3^+ &= 120\end{aligned}$$

$$x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+ \geq 0$$

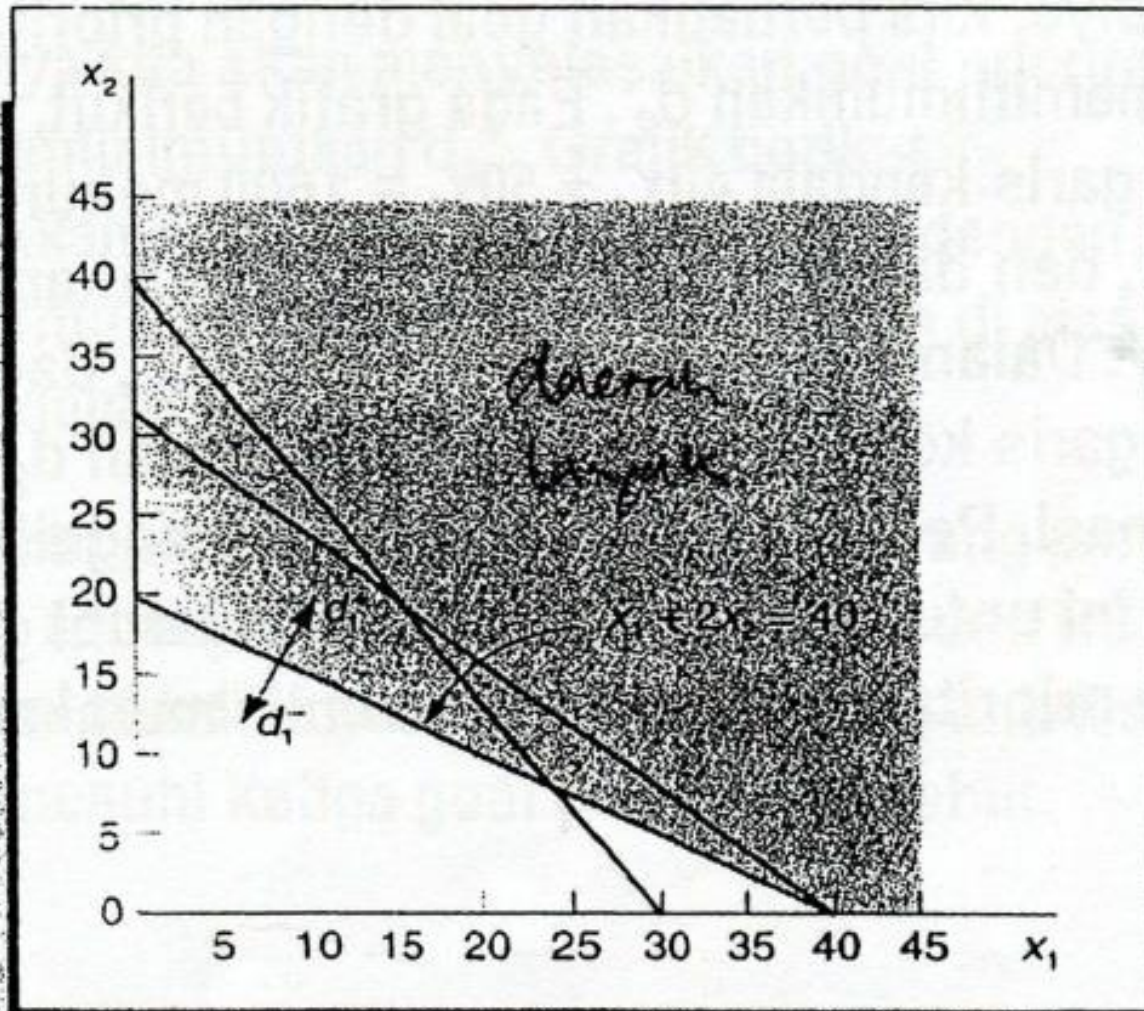
Untuk membuat grafik model ini, variabel penyimpangan di setiap kendala goal dijadikan nol sehingga masing-masing persamaan dapat digrafikkan pada satu set koordinat seperti gambar berikut:



Perhatikan bahwa dari grafik tersebut tidak ada daerah solusi layak karena ketiga kendala goal adalah persamaan sehingga semua titik solusi ada di garis kendala.

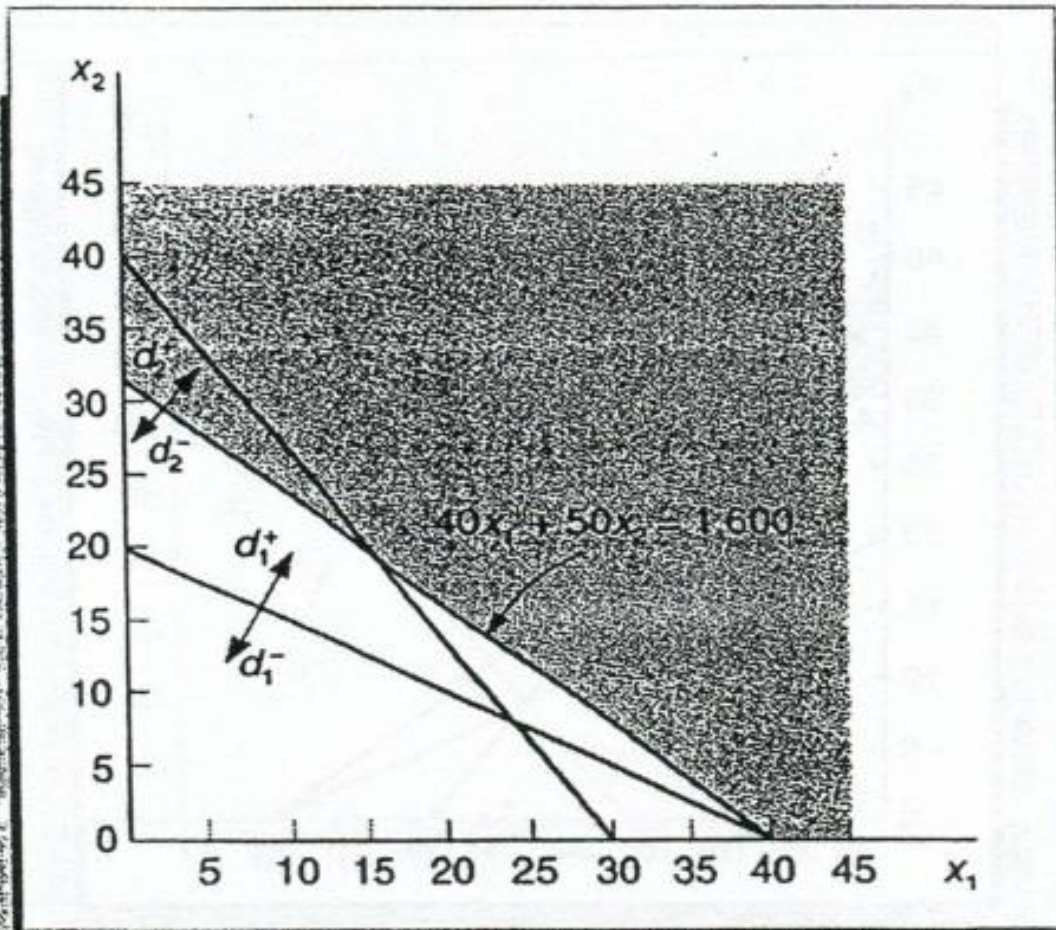
Logika solusi pada goal programming adalah sebuah upaya untuk mencapai sasaran atau goal yang terdapat di fungsi objektif sesuai dengan prioritasnya. Bila sebuah goal tercapai, perhatian kemudian ditujukan pada goal dengan prioritas tertinggi berikutnya. Namun, goal dengan prioritas yang lebih tinggi yang telah dicapai tidak boleh dibatalkan oleh goal dengan prioritas yang lebih rendah.

Pada contoh ini, kita perhatikan lebih dahulu goal untuk meminimumkan d_1^- . Hubungan antara d_1^- dan d_1^+ terhadap kendala goal ditunjukkan pada grafik berikut



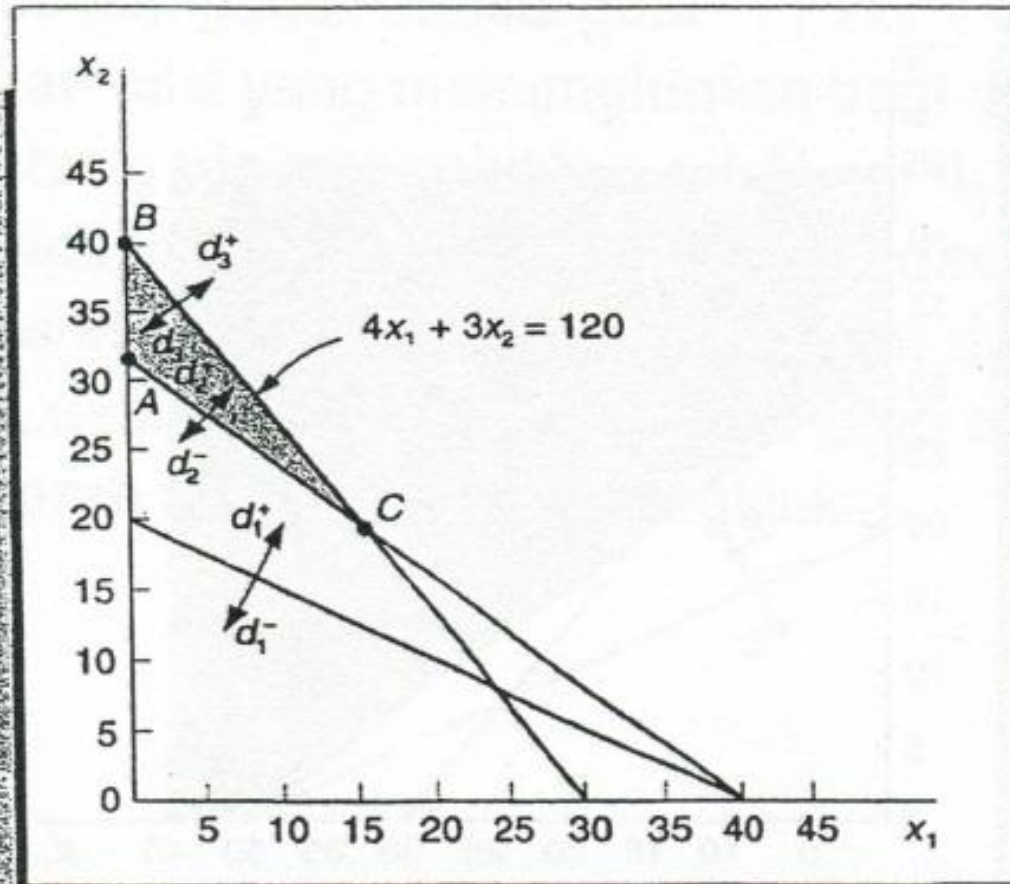
Daerah di bawah garis kendala goal $x_1 + 2x_2 = 40$ menunjukkan nilai yang memungkinkan bagi d_1^- , dan daerah di atas garis menunjukkan nilai bagi d_1^+ . Dalam rangka mencapai goal meminimumkan d_1^- , daerah di bawah garis kendala yang berkaitan dengan d_1^- dieliminasi, sehingga tinggalah daerah yang diarsir sebagai daerah solusi yang memungkinkan.

Berikutnya, kita perhatikan goal dengan prioritas kedua yakni meminimumkan d_2^- . Pada grafik berikut, daerah di bawah garis kendala $40x_1 + 50x_2 = 1600$ menunjukkan nilai d_2^- , dan daerah di atas garis menunjukkan nilai bagi d_2^+ . Dalam rangka meminimumkan d_2^- , daerah di bawah garis kendala yang berkaitan dengan d_2^- dieliminasi. Perhatikan bahwa dengan mengeliminasi daerah ini untuk d_2^- , kita tidak mempengaruhi goal dengan prioritas pertama untuk meminimumkan d_1^- .



Selanjutnya kita akan menyelesaikan goal prioritas ketiga, meminimumkan d_3^+ . Grafik berikut menunjukkan daerah yang berhubungan dengan d_3^- dan d_3^+ . Untuk meminimumkan d_3^+ daerah di atas garis kendala $4x_1 + 3x_2 = 120$ dieliminasi.

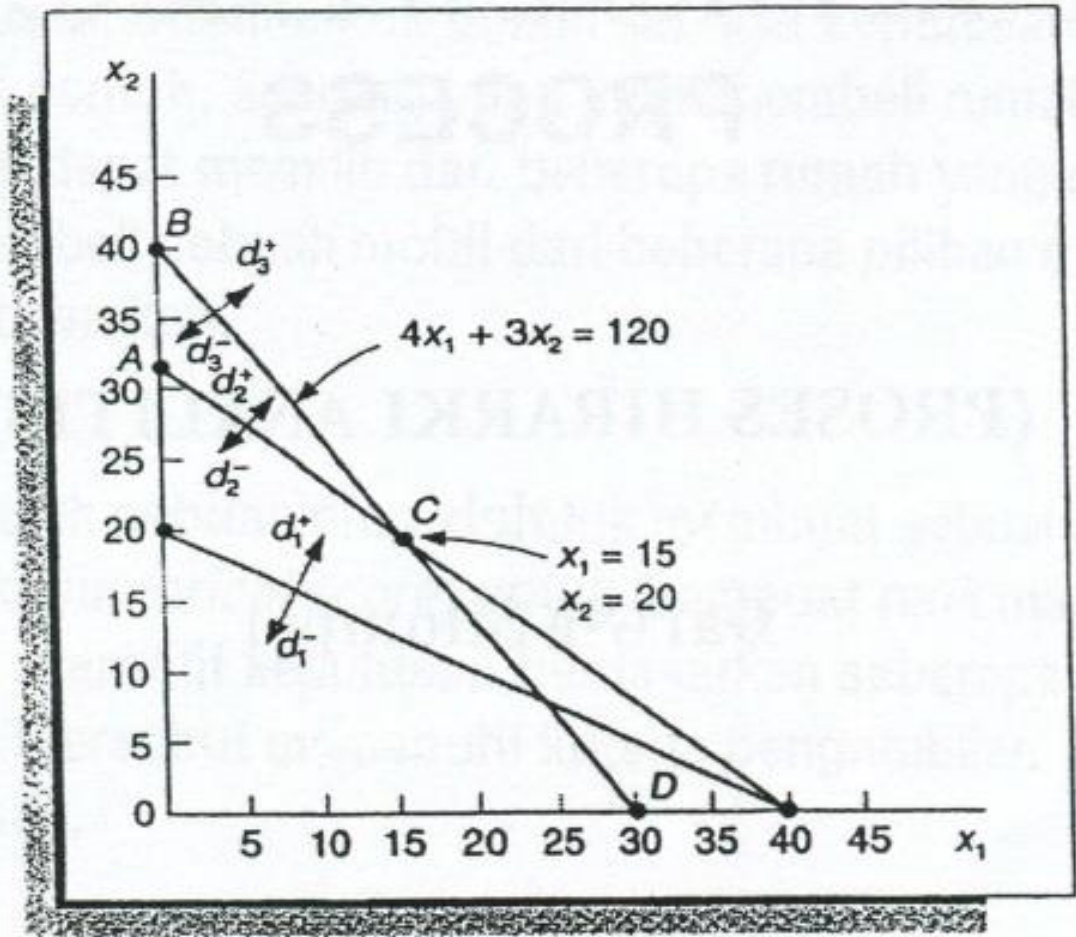
Setelah mempertimbangkan ketiga goal yang pertama, diperoleh sebuah daerah antara garis AC dan BC, yang berisi titik-titik yang memungkinkan untuk solusi yang memenuhi ketiga goal pertama tersebut.



Terakhir, kita harus memperhatikan goal prioritas keempat, meminimumkan d_1^+ . Untuk mencapai goal terakhir ini, daerah di atas garis kendala $x_1 + 2x_2 = 40$ harus dieliminasi. Namun, bila daerah ini kita eliminasi maka d_2^- dan d_3^- akan terganggu. Dengan kata lain, kita tidak dapat meminimumkan d_1^+ sepenuhnya tanpa mengganggu prioritas pertama dan kedua.

Oleh karena itu, kita mengusahakan memperoleh titik solusi yang memenuhi ketiga goal pertama tetapi berusaha memenuhi sebanyak mungkin goal dengan prioritas keempat ini.

Titik C menunjukkan solusi yang memungkinkan semua kondisi ini.



Perhatikan bahwa jika kita bergerak ke bawah garis kendala goal $4x_1 + 3x_2 = 120$ menuju titik D, d_1^+ dapat lebih diminimumkan; namun, harga d_2^- akan bertambah ketika kita bergerak menuju titik C. Jadi minimisasi d_1^+ hanya dapat diperoleh dengan mengorbankan goal yang berprioritas lebih tinggi.

Solusi pada titik C dihitung dengan menyelesaikan dua persamaan yang saling memotong di titik tersebut. Diperoleh:

$$x_1 = 15 \text{ cawan}$$

$$x_2 = 20 \text{ mangkuk}$$

$$d_1^+ = 15 \text{ jam}$$