アルゴリズム論
Theory of Algorithms

第6回講義
枝刈り探索法(続き)

アルゴリズム論
Theory of Algorithms

Lecture #6

Prune and Search (Cont'd)

枝刈り探索法の原理

与えられた問題を解くのに，構成要素を1つずつ吟味し，問題の解決に不必要だと思われるものを冗長なものとして削除し，縮小された問題を再帰的に解く.

サイズnの問題を解くのに要する計算時間を$T(n)$とする.
サイズnの問題の構成要素を吟味して，サイズをrn $(r < 1)$の問題に縮小できるものとする.
構成要素の吟味が線形のcn時間でできるとすると，漸化式

$$T(n) \leqq T(rn) + cn$$

を得る.

この漸化式は次のように解ける．ただし，$r^k n \leqq 1$とし，$T(1)=d$とする.

$$T(n) \leqq T(rn) + cn \leqq T(r^2 n) + crn + cn \leqq T(r^3 n) + cr^2 n + crn + cn$$
$$\leqq T(r^k n) + cr^{k-1} n + \cdots + cr^2 n + crn + cn \leqq d + cn/(1-r).$$

よって，$T(n) = O(n)$となり，線形時間であることがわかる.

Principle of Prune-and-Search

Solve a given problem by removing as many redundant elements as possible by examining every element and solving the reduced problem recursively.

$T(n)$: time to solve a problem of size n.

Assume that examination of every element of a problem of size n reduces the problem size into rn ($r<1$).

If we can examine all the elements in cn time, then we have the recurrence equation

$$T(n) \leqq T(rn) + cn.$$

The equation can be solved as follows: Here assume that $r^k n \leqq 1$ and $T(1)=d$, a constant. Then, we have

$$T(n) \leqq T(rn) + cn \leqq T(r^2 n) + crn + cn \leqq T(r^3 n) + cr^2 n + crn + cn$$
$$\leqq T(r^k n) + cr^{k-1} n + \cdots + cr^2 n + crn + cn \leqq d + cn/(1-r).$$

Thus, we have $T(n) = O(n)$, i.e., it runs in linear time.

2変数線形計画問題

問題P15:(2変数線形計画問題)
変数x, yに関する多数の線形不等式をすべて満たす(x, y)の中で，線形の目的関数cx+dyの値を最小(または最大)にするものを求めよ.

一般的な形式:
目的関数: $c_1x_1 + c_2x_2 \rightarrow \min$
制約条件:
$$
\begin{aligned}
d_{11}x_1 + d_{12}x_2 &\geqq e_1,\\
d_{21}x_1 + d_{22}x_2 &\geqq e_2,\\
&\cdots\cdots\\
d_{n1}x_1 + d_{n2}x_2 &\geqq e_n.
\end{aligned}
$$

$y=c_1x_1 + c_2x_2$, $x=x_1$とおくと，n個の制約式は，d_{i2}/c_2の符号により

$$y \geqq a_ix+b_i,$$

の形の下方制約式と

$$y \leqq a_ix+b_i,$$

の形の上方制約式に分類できる.

2-dimensional Linear Program

Problem P15:（2-dimensional Linear Program）
Find (x, y) to minimize (or maximize) a linear objective function cx+dy among those satisfy all the linear inequalities on variables x and y.

General form：

Objective: $c_1x_1 + c_2x_2 \rightarrow \min$

subject to :
$$\begin{aligned} d_{11}x_1 + d_{12}x_2 &\geqq e_1, \\ d_{21}x_1 + d_{22}x_2 &\geqq e_2, \\ &\cdots\cdots \\ d_{n1}x_1 + d_{n2}x_2 &\geqq e_n. \end{aligned}$$

If we set $y=c_1x_1 + c_2x_2$, $x=x_1$,
n inequalities are classified into lower constraints

$y \geqq a_ix+b_i$,

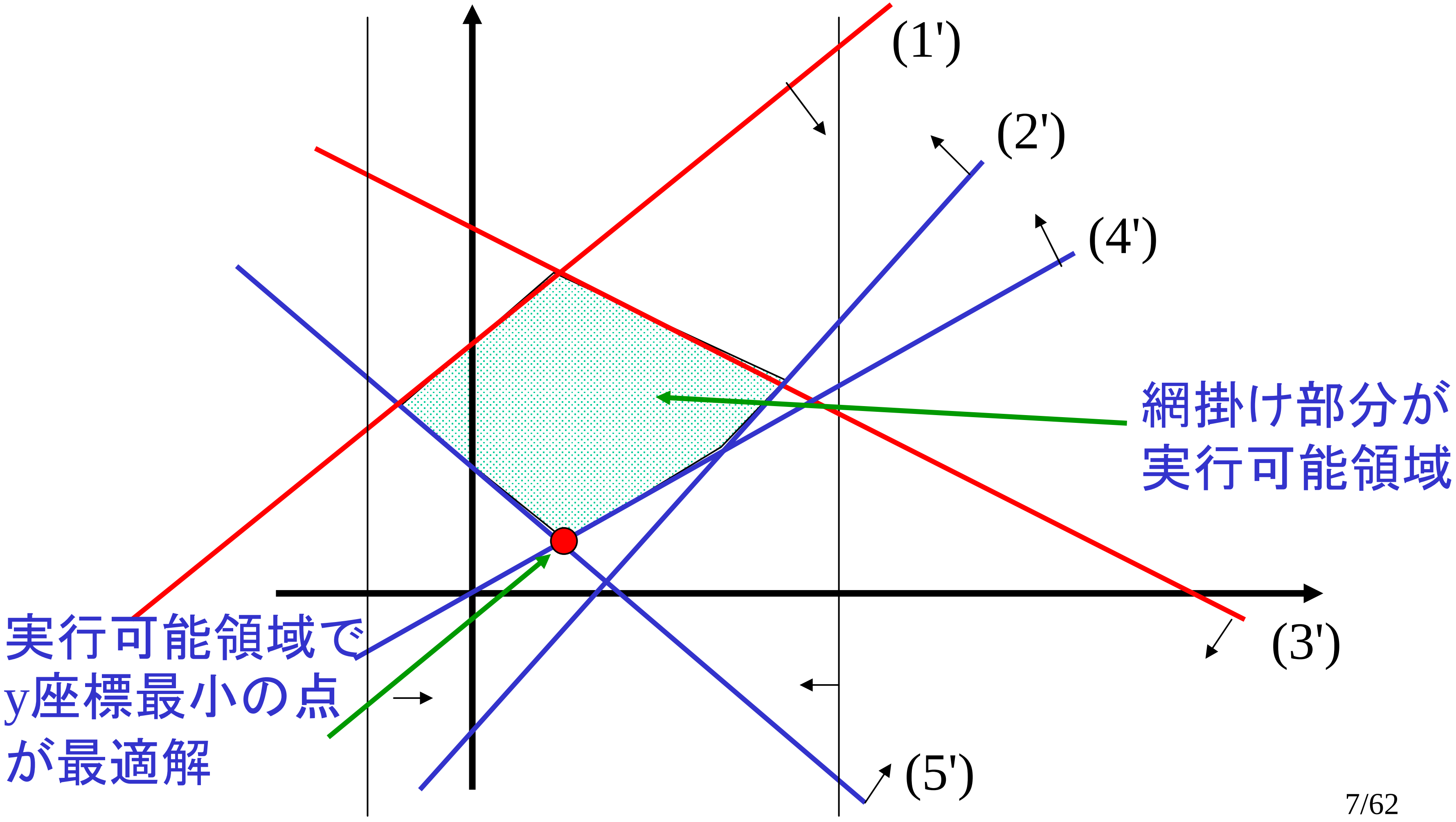
and upper constraints

$y \leqq a_ix+b_i$,

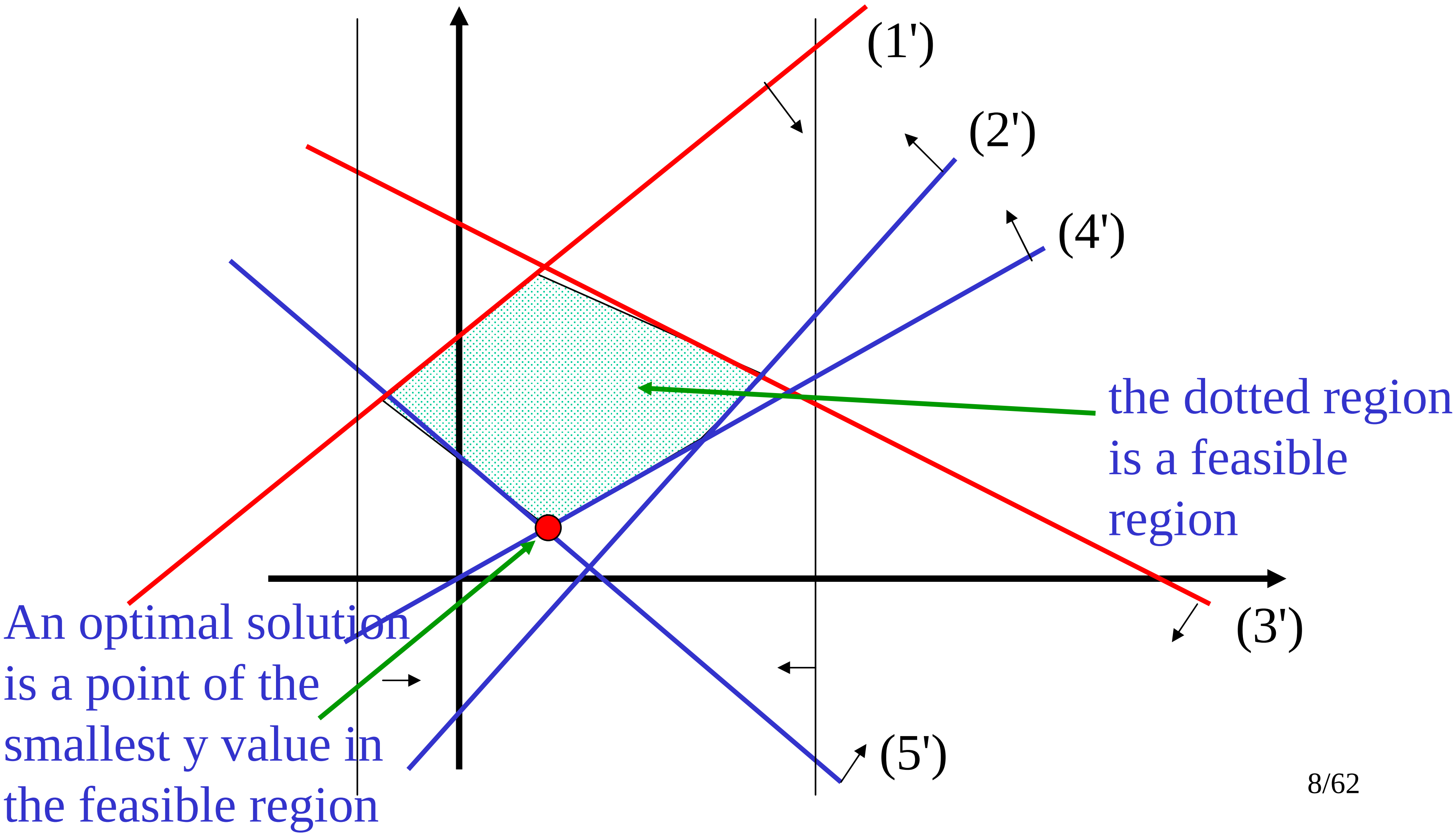
depending on signs of d_{i2}/c_2.

目的関数 : $y \to \min$

制約条件 : (2')$y \geqq x-3$, (4') $y \geqq (1/2)x$, (5') $y \geqq -x+3$,
(1')$y \leqq x+5$, (3')$y \leqq -(1/2)x+7$, (6'7') $-2 \leqq x \leqq 8$.



Objective: $y \to \min$
subject to: (2') $y \geqq x-3$, (4') $y \geqq (1/2)x$, (5') $y \geqq -x+3$,
(1') $y \leqq x+5$, (3') $y \leqq -(1/2)x+7$, (6'7') $-2 \leqq x \leqq 8$.



補題：実行可能領域Fは，（もし存在すれば）凸多角形を成す．

証明：実行可能領域は制約式を満たす領域
下方制約式を満たす領域は下に凸，
上方制約式を満たす領域は上に凸，
よって，それらの凸領域の共通部分も凸である．

２つの関数の定義
g(x): 下方制約式を満たす領域の境界を表す関数
h(x): 上方制約式を満たす領域の境界を表す関数
すなわち，

$$g(x) = \max\{a_i x + b_i \mid i \in I_1\}$$

$$h(x) = \min\{a_i x + b_i \mid i \in I_2\}$$

と定めるとき，実行可能領域Fは

$$F = \{(x, y) \mid g(x) \leqq y \leqq h(x) \text{ and } a \leqq x \leqq b\}$$

と表現できる．

Lemma: A feasible region forms a convex polygon (if it exists).

Proof: A feasible region is a region satisfying all the constraints
The region satisfying the lower constraints is convex downward.
The region satisfying the upper constraints is convex upward.
Thus, the intersection of these convex regions is also convex.

Definition of two functions

$g(x)$: function representing the boundary of the region satisfying the lower constraints

$h(x)$: function representing the boundary for the upper constraints

That is, if we define

$g(x) = \max\{a_i x + b_i \mid i \in I_1\}$, and

$h(x) = \min\{a_i x + b_i \mid i \in I_2\}$

then, the feasible region F is expressed by

$F = \{(x, y) \mid g(x) \leqq y \leqq h(x) \text{ and } a \leqq x \leqq b\}$.

n個の制約式が与えられたとき，対応する実行可能領域を陽に構成すれば，$O(n \log n)$時間かかる.

線形時間のアルゴリズムを得るには，実行可能領域全体を陽に構成せずに，最適解を与える頂点を求めることが必要.

枝刈り探索の考え方.

- 制約式を吟味して，一定割合の制約式を冗長な制約式として削除する.
- 最適解を与える頂点は2つの制約式に対応する直線の交点として与えられる.
- 関数$g(x)$と$h(x)$の性質を利用する.
  $g(x)$は下に凸，$h(x)$は上に凸
- ただし，関数$g(x)$と$h(x)$の全体を求めようとすると，$O(n \log n)$時間かかってしまうことに注意.

Given n constraints, if we construct the corresponding feasible region in an explicit manner, then it takes O(n log n) time.

To obtain a linear-time algorithm, we need to find an optimal vertex without constructing the whole feasible region explicitly.

Idea based on Prune-and-Search

- By examining the constraints, we remove constraints by a fixed ratio as redundant constraints.
- A vertex giving an optimal solution is specified as an intersection of two lines corresponding to constraints.
- Use properties of the functions g(x) and h(x)

  g(x) is convex downward while h(x) is convex upward.
- Note that if we compute the whole of the functions g(x) and h(x) then it takes O(n log n) time.

関数g(x)とh(x)の性質を用いた枝刈り探索

性質(1)下方制約に対応する関数g(x)は下に凸,
(2)上方制約に対応する関数h(x)は上に凸,
(3)xの値の範囲は[a,b].

区間[a,b]内の任意のx'を指定したとき,
直線x=x'と下方制約式との交点の中でy座標最大の点
$g(x')=\max\{a_i x' + b_i \mid i \in I_1\}$
直線x=x'と上方制約式との交点の中でy座標最小の点
$h(x') = \min\{a_i x' + b_i \mid i \in I_2\}$
ここで, $g(x') \leqq h(x')$ ならば, x=x'は実行可能領域と交わる.
(注意)上の関数値はO(n)時間で計算可能.

ケース1 : $g(x') > h(x')$のとき
ケース2 : $g(x') \leqq h(x')$のとき
いずれの場合にも, 交点を与える直線の傾きにより冗長な制約式を削除する.

Prune-and-Search based on the properties of g(x) and h(x)

Properties

(1) The function g(x) for the lower constraints is convex downward,

(2)The function h(x) for the upper constraints is convex upward, and

(3) The range of x is [a,b].

When we specify any x' in the interval [a,b],

among intersections of the line x=x' with lower constraints find the one of the largest y value gives g(x')

$$g(x')=\max\{a_i x' + b_i \mid i \in I_1 \}$$

among intersections of the line x=x' with upper constraints find the one of the smallest y value gives h(x')

$$h(x') = \min\{a_i x' + b_i \mid i \in I_2 \}$$

If $g(x') \leqq h(x')$ then the line x=x' really intersect a feasible region.

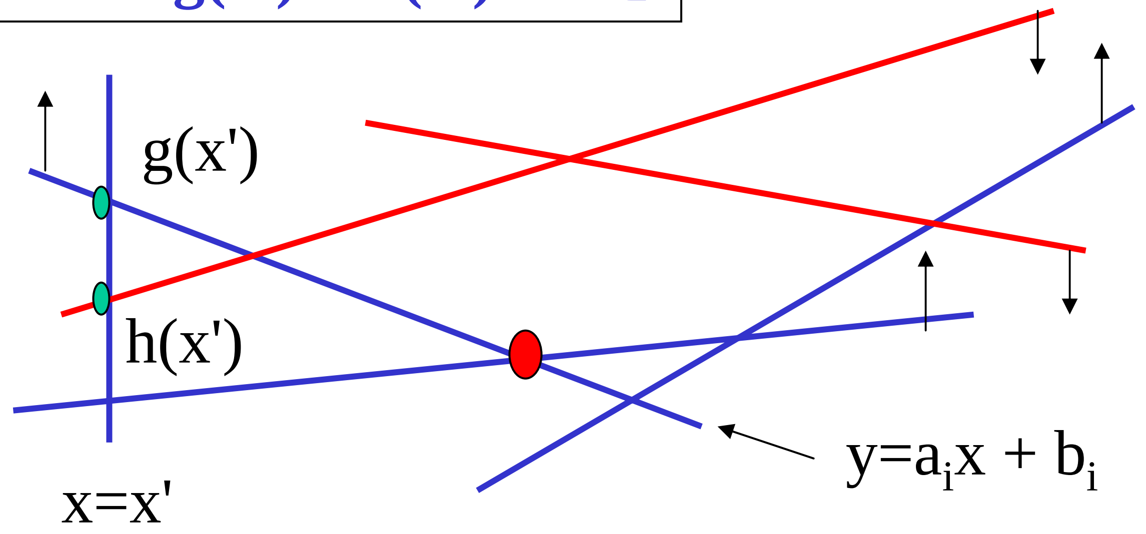
（Remark）The above function values care computed in O(n) time.

Case 1: $g(x') > h(x')$

Case 2: $g(x') \leqq h(x')$

In either case, we remove redundant constraints based on the slope of the line giving the intersection.

ケース1：g(x') > h(x')のとき



g(x')の値を与える制約式を$y \geqq a_ix + b_i$とする.

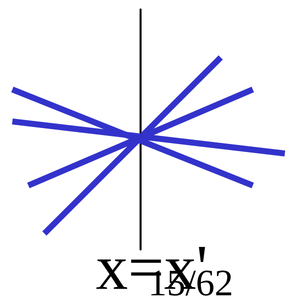
ケース1.1: g(x')を与える制約式がただ1つの場合

この場合，x=x'における関数y=g(x)の微分値g'(x')が
$g'(x')=a_i$として求まる.

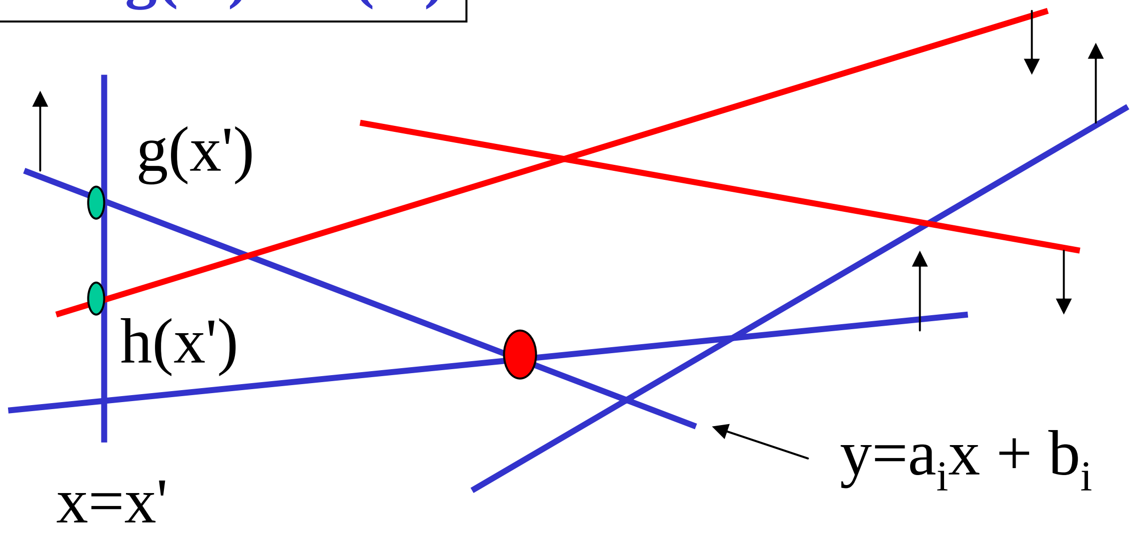
ケース1.2: g(x')を与える制約式が複数個ある場合

それらの関数の中の最大の傾きがx=x'の右
での微分値

それらの関数の中の最小の傾きがx=x'の左
での微分値



Case 1 : g(x') > h(x')



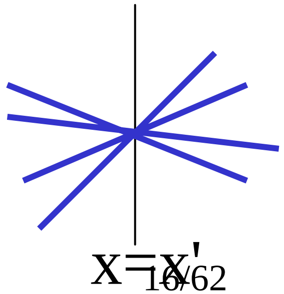
Suppose the constraint $y \geqq a_i x + b_i$ gives the value of g(x').

Case 1.1: Only one constraint gives g(x')

In this case, the differential value g'(x') of the function y=g(x) at x=x' is obtained as $g'(x')=a_i$.

Case 1.2: More than one constraint give the value of g(x')

The largest slope among those functions is the differential value at the left of x=x'.
The smallest slope among those functions is the differential value at the right of x=x'.



最適解がどちらの方向にあるかを知りたい

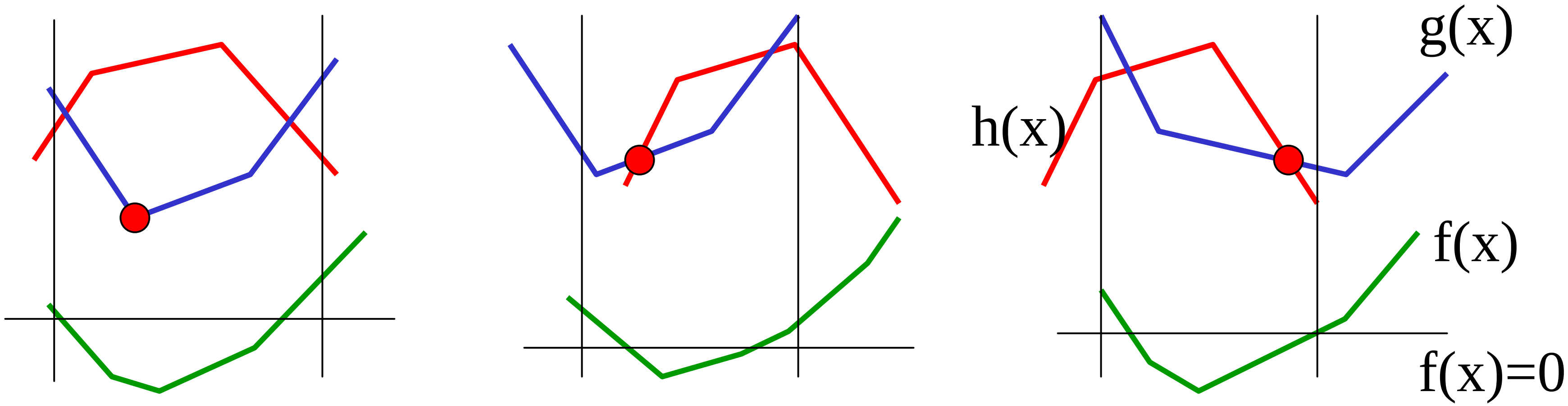
新たな関数 $f(x) = g(x) - h(x)$ の導入

$g(x)$: 下方制約式，下に凸

$h(x)$: 上方制約式，上に凸，$-h(x)$は下に凸

$f(x)=g(x)-h(x)$は下に凸.

(x,y)が実行可能<=> $g(x) \leqq y \leqq h(x)$ <=> $f(x) \leqq 0$



$h(x') \leqq g(x')$が成り立つ様々な状況(縦線で示す)

$x=x'$における$y=g(x)$の傾きだけでは最適解の方向を特定することはできない.

$y=f(x)$の傾きを考えると，最適解は$f(x)$の値が減少する方向にある

Want to know in which side an optimal solution lies.

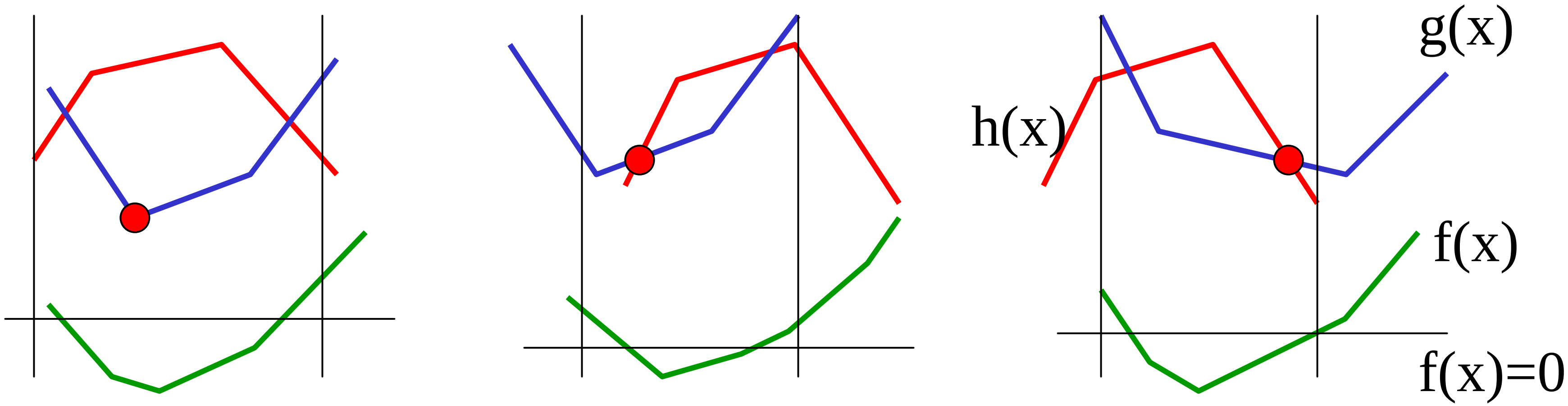
Introducing a new function f(x) = g(x) - h(x).

g(x): lower constraints, convex downward

h(x): upper constraints, convex upward, -h(x) is convex downward

f(x)=g(x)-h(x) is convex downward.

(x,y) is feasible <=> g(x) ≦y ≦h(x) <=> f(x) ≦ 0



Various situations (indicated by vertical lines) in h(x')≦g(x') holds

We cannot know the direction of an optimal solution using only slope of y=g(x) at x=x'.

Considering the slope of y=f(x), an optimal solution lies in the direction in which f(x) value decreases.

関数y=f(x)のx=x'における傾きを知りたい

　O(n)時間で，x=x'（の左右）におけるy=g(x)の傾き

　O(n)時間で，x=x'（の左右）におけるy=h(x)の傾き

これらを用いてx=x'（の左右）におけるy=f(x)の傾きを求める.

f(x')>0かつx=x'の右におけるy=f(x)の傾き<0のとき

　最適解はx=x'の右にある.

f(x')>0かつx=x'の左におけるy=f(x)の傾き>0のとき

　最適解はx=x'の左にある.

それ以外は

　解は存在しない.

最適解の方向が分かると，冗長な制約式を削除できる.

その方法は？

Want to know the slope of the function y=f(x) at x=x‘

O(n) time suffices to find slopes of y=g(x) at x=x'(and around it).

O(n) time suffices to find slopes of y=h(x) at x=x'(and around it).

Using them, find slopes of y=f(x) at x=x’ (and around it).

When f(x’)>0 and the slope of y=f(x) at the right of x=x’ < 0

optimal solution lies to the right of x=x’.

When f(x’)>0 and the slope of y=f(x) at the left of x=x’ > 0

optimal solution lies to the left of x=x’.

Otherwise,

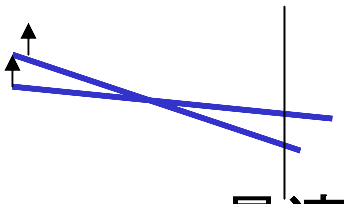
there is no solution.

Once the direction of an optimal solution is known, we can remove redundant constraints.

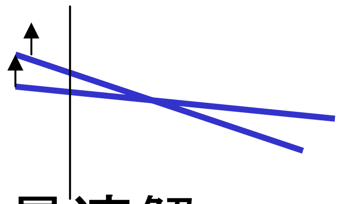
How?

x=x'の右に最適解がある場合

下方制約式を2つずつペアにする.
上方制約式も2つずつペアにする.
制約式が奇数個なら，最後の一つはペアにしない.



最適解

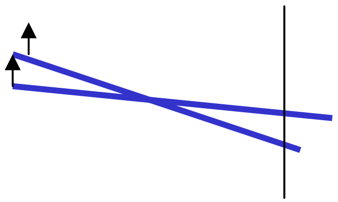


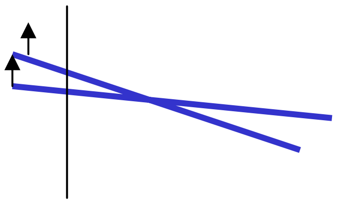
最適解

下方制約式のペア(i,j)　$y \geqq a_i x + b_i$, $y \geqq a_j x + b_j$
交点をx_{ij}とする.
実行可能領域は[a,b]の範囲にあった.
(1) $x_{ij} \leqq a$のとき,
　　最適解は右にある．よって，傾きの小さい制約は冗長
(2) $x_{ij} \geqq b$のとき,
　　最適解は左にある．よって，傾きの大きい制約は冗長
(3)それ以外のとき

When an optimal solution lies to the right of x=x'

Decompose lower constraints into pairs.
Decompose upper constraints into pairs.
If we have odd number of constraints, then leave the last one.





optimal solution   optimal solution

Pair (i, j) of lower constraints: $y \geqq a_i x + b_i$, $y \geqq a_j x + b_j$
Let their intersection be x_{ij}.
Note that the feasible region is in the region [a,b].
(1) Case 1: $x_{ij} \leqq a$,
Optimal solution lies to the right. So, the one of smaller slope in the pair is redundant.
(2) Case 2 $x_{ij} \geqq b$,
Optimal solution lies to the left. So, the one of larger slope in the pair is redundant.
(3)Case 3: otherwise.

交点が[a,b]の区間に入っているとき

　　そのようなペアについて交点x_{ij}を求め，それらの中央値x_mを求める.

$x=x_m$において，$g(x)$と$h(x)$の値を求めるなど，先と同じ計算を行う

このとき，下記のいずれかの結論を得る:

(1)この問題に解はない.

(2)x_mは最適解である.

(3)最適解は$x=x_m$の右にある.

(4)最適解は$x=x_m$の左にある.

(3)の場合，交点が$x=x_m$の左にあるペア(i,j)については，

　　一方の制約式を冗長な制約式として削除可能.

(4)の場合，交点が$x=x_m$の右にあるペア(i,j)については，

　　一方の制約式を冗長な制約式として削除可能.

したがって，約1/4の制約式を削除することができる.

　　=>枝刈り探索法により線形時間で最適解が求まる.

When the intersection lies in the interval [a,b]:

Find the intersection x_{ij} for each such pair and compute the median of those intersections.

Do the same calculations as before, say to compute g(x) and h(x) at $x=x_m$. Then , we have one of the followings as the result.

(1)This problem has no solution.

(2)x_m is the optimal solution.

(3)An optimal solution lies to the right of $x=x_m$.

(4)An optimal solution lies to the left of $x=x_m$.

For (3), we can remove one of the paired constraints whose intersection lies to the left of $x=x_m$ as a redundant constraint.

For (4), we can remove one of the paired constraints whose intersection lies to the right of $x=x_m$ as a redundant constraint.

Thus, we can remove about ¼ of all constraints.

➔ We can find an optimal solution in linear time using Prune-and-Search.

定理:2変数の線形計画問題は，制約式の個数nの線形時間で解ける.

証明:
簡単のため，どの制約式もxとyの両方に関係するものとする.
(1)n個の制約式を下方制約式と上方制約式に分け，それぞれでペアを作る. それぞれのペアに対して対応する直線の交点を求める. そのような交点の個数はほぼn/2である.
(2)上記の交点たちのx座標の中央値x_mを求める.
(3)直線$x=x_m$と下方制約式に対応する下方境界と上方制約式に対応する上方境界との交点と，交点を与える制約式を求めることにより，それらの交点における実行可能領域の傾きを求める.
(4)上記で求めた傾きにより最適解が直線$x=x_m$のどちら側にあるかを判定し，その反対側に交点をもつ制約式のペアについて一方の制約式を冗長な制約式として削除する. このとき削除される制約式は，少なくともn/4個ある.

Theorem: A 2-dimensional linear program can be solved in time linear in the number of constraints.

Proof:

For simplicity, we assume that every constraint is associated with both x and y.

(1) Divide the n constraints into lower and upper constraints and decompose them into pairs in each group. For each pair, compute the intersection of their associated lines. There are about n/2 such intersections in total.

(2)Find the median of those x-coordinates x_m.

(3)At the line $x=x_m$, we compute slopes of the boundary of the feasible region by finding the lower constraint giving the intersection with the lower boundary for lower constraints and the one for the upper boundary for upper constraints.

(4) Using the slopes obtained above, we check which side of the line $x=x_m$ contains an optimal solution, and then we remove as redundant constraints one of paired constraints having their corresponding intersection in the other side. Then, there are at least n/4 constraints to be removed.

したがって，n個の制約式で定義される2変数線形計画問題を解くための時間をT(n)とすると，
(1)制約式のペアに対する交点はO(n)時間で計算できる.
(2)もO(n)時間（最悪でも線形時間で中央値を求めることが可能）.
(3)境界との交点もすべての制約式との交点を調べればよいからO(n)時間で十分.
(4)中央値の位置を利用して制約式を削除する操作もO(n)時間.
これにより制約式は高々(3/4)n個に減る.
したがって，O(n)の時間を適当な定数cによりcnと表せば，

$$T(n) \leqq T((3/4)n) + cn$$

という漸化式を得る. これを解けば，T(n) = O(n)を得る.

演習問題：xだけの制約式やyだけの制約式があれば，それらはそれぞれどのように扱うべきか？

Therefore, if we denote by T(n) the time to solve a 2-dimensional linear program, then we have

(1)intersections for those pairs can be computed in O(n) time in total,

(2) is done in O(n) time（median can be found in linear time in the worst case).

(3)O(n) time suffices to compute intersections with all constraints to compute the intersection with the boundary.

(4)The operations to remove redundant constraints using the location of the median are done in O(n) time. Then, the number of constraints is reduced to at most (3/4)n.

Hence, if we represent O(n) time by cn for some constant c, then we have

$$T(n) \leqq T((3/4)n) + cn,$$

from which we obtain T(n) = O(n).

Exercise：How should we treat those constraints associated only with x or y?

アルゴリズム論
Theory of Algorithms

第7回講義
線形計画法

アルゴリズム論
Theory of Algorithms

Lecture #7

Linear Programming

線形計画問題と線形計画法

入力:線形不等式と線形の目的関数
出力:すべての線形不等式を満たす解があるかどうかを判定し, 解が存在する場合には, さらに目的関数を最大(または最小)にする解を求める.

n: 変数の個数,
m: 線形不等式の形で与えられる制約式の個数

各制約式はn個の変数の線形不等式の形で与えられるから, それぞれがn次元空間の半空間に対応している.
したがって, m個の半空間の共通部分が存在するかどうかを判定し, 存在するなら, その頂点の中で目的関数の値を最大(または最小)にするものを求めればよい.
実行可能領域:m個の半空間の共通部分

線形計画問題はnとmに関する多項式時間で解けることが知られている.

Linar Program and Linear Programming

Input：Linear inequalities and linear objective function
Output：Determine whether there is a solution satisfying all the linear inequalities, and if there exists, find a solution to maximize (or minimize) the objective function.

n: number of variables,
m: number of constraints given as linear inequalities

Since constraints are given as linear inequalities on n variables, they correspond to half spaces in the n-dimensional space.
Thus, it suffices to determine whether m half spaces have their intersection, and if it exists, to find a vertex at which the objective function is maximum(or minimum).

Feasible region：intersection of m half spaces

It is known that Linear Program can be solved in polynomial time in n and m.

２変数の線形計画問題

2次元平面における半平面の共通部分が実行可能領域

必ず凸多角形になる.

３変数の線形計画問題

3次元空間における半空間の共通部分が実行可能領域

必ず凸多面体になる.

線形計画問題の一般形

目的関数: $c_1x_1+c_2x_2+\dots+c_nx_n \rightarrow \min$

ただし, $c_1, c_2, \dots, c_n$は与えられた定数

制約式:

不等式制約 $a_{11}x_1+a_{12}x_2+\dots+a_{1n}x_n \leqq b_1$

$\dots\dots$

$a_{k1}x_1+a_{k2}x_2+\dots+a_{kn}x_n \leqq b_k$

等式制約 $a_{k+1,1}x_1+a_{k+1,2}x_2+\dots+a_{k+1,n}x_n = b_{k+1}$

$\dots\dots$

$a_{m1}x_1+a_{m2}x_2+\dots+a_{mn}x_n = b_m$

2-variable linear program

Feasible region is intersection of half planes in the plane, which is always a convex polygon.

3-variable linear porgram

Feasible region is intersection of half spaces in the space, which is always a convex polyhedron.

General form of Linear Program

Objective function: $c_1x_1+c_2x_2+\ldots+c_nx_n \rightarrow \min$

Here, $c_1, c_2, \ldots, c_n$ are given constants.

Constraints:

Inequalities $a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n \leqq b_1$

$\ldots\ldots$

$a_{k1}x_1+a_{k2}x_2+\ldots+a_{kn}x_n \leqq b_k$

Equalities $a_{k+1,1}x_1+a_{k+1,2}x_2+\ldots+a_{k+1,n}x_n = b_{k+1}$

$\ldots\ldots$

$a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n = b_m$

問題P16: 2種類の原材料AとBにより2種類の製品P_1とP_2を製造する場合，どのような生産計画を立てれば利益最大にできるか？

製品を1単位製造するのに必要な原材料の量
製品P_1では，Aを2, Bを4単位分だけ必要
製品P_2では，Aを3, Bを5単位分だけ必要
原材料の在庫は，Aが5単位，Bが9単位分
利益率：製品P_1は1単位当たり3万円，製品P_2は4万円
➔
製品P_1の生産量をx_1, 製品P_2の生産量をx_2とすると，
全体の利益は

$3x_1 + 4x_2$　　（最大化すべき）目的関数

で与えられる．一方，制約は

$2x_1 + 3x_2 \leqq 5$　在庫量の関係
$4x_1 + 5x_2 \leqq 9$　在庫量の関係
$x_1 \geqq 0,\ x_2 \geqq 0$　生産量は非負

となる．

Problem P16: Suppose 2 products P_1 and P_2 **are made from 2 materials A and B. How can we maximize the profit?**

Quantity required to make one unit of products

For products P_1, 2 units of A and 4 units of B are required.

For products P_2, 3 units of A and 5 units of B are required.

5 units of A and 9 units of B are available.

Profit: 30,000 yen per unit for product P_1, 40,000 yen for P_2

➔

Let x_1 and x_2 be quantities of P_1 and P_2. Then, the profit is given by

$3x_1 + 4x_2$ objective function (to be maximized)

On the other hand, constraints are

$2x_1 + 3x_2 \leqq 5$ constraints on quantities available

$4x_1 + 5x_2 \leqq 9$ constraints on quantities available

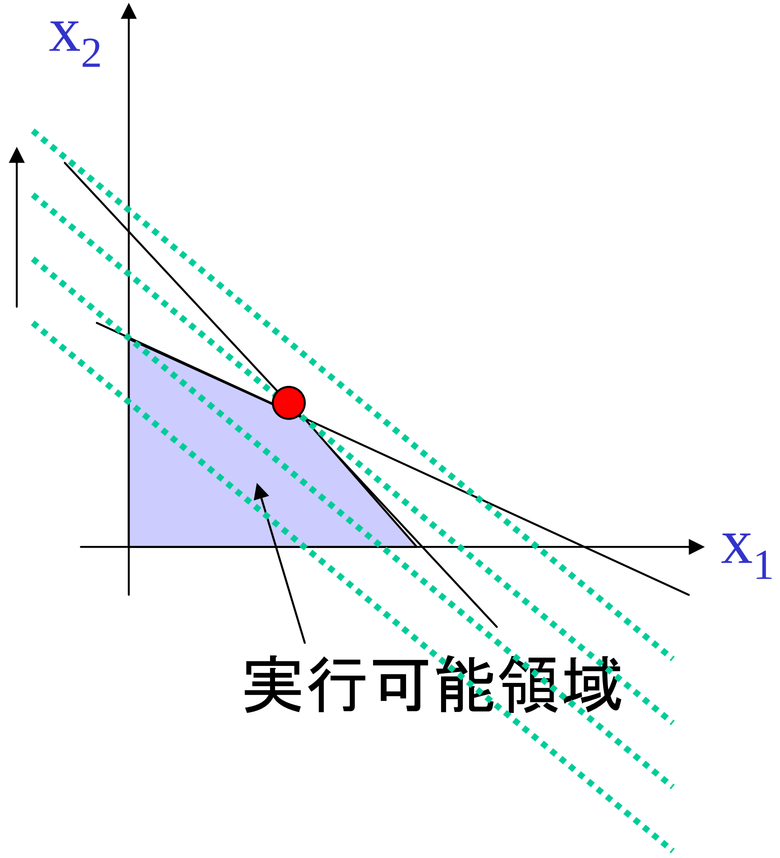
$x_1 \geqq 0, x_2 \geqq 0$ quantities are not negative.

線形計画問題

目的関数:　$3x_1 + 4x_2$→最大

制約条件:

$$2x_1 + 3\,x_2 \leqq 5$$
$$4x_1 + 5\,x_2 \leqq 9$$
$$x_1 \geqq 0,\ x_2 \geqq 0$$



目的関数 $3x_1 + 4x_2 = k$

→直線　$x_2 = -(3/4)x_1 + k/4$

2つの制約式に対応する直線の交点は(1, 1).

つまり，製品P_1を1単位，製品P_2も1単位だけ製造するのが最適.

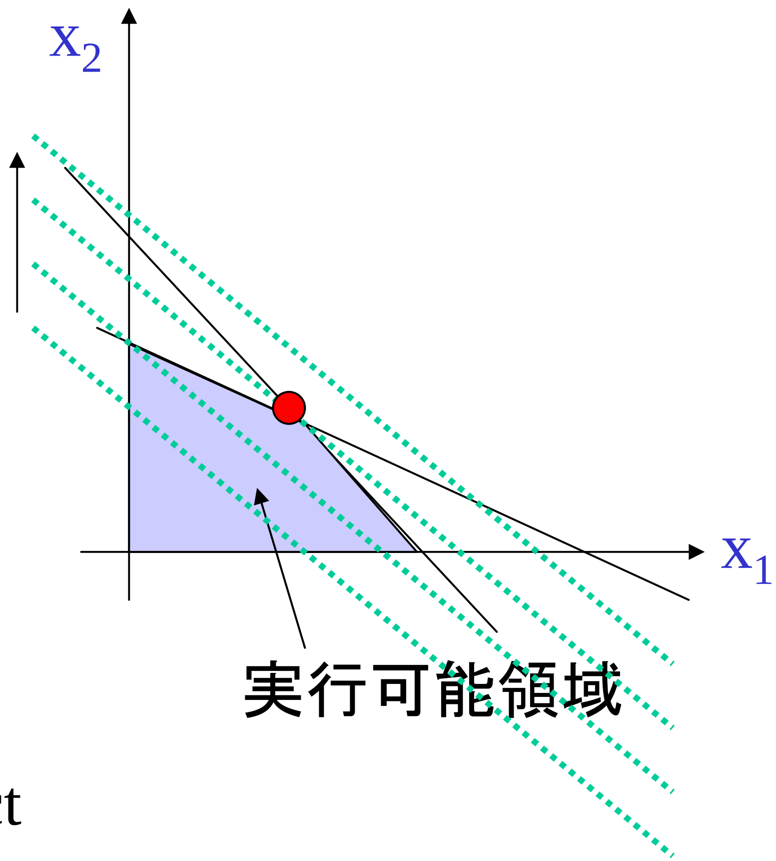
演習問題E7-1:上の例において，他に2個の変数x_3, x_4を導入して，不等式制約をすべて線形等式制約と変数≧0の形式に変更する方法を考えよ.

Linear Program

Objective function: $3x_1 + 4x_2 \rightarrow \max$

Constraints:

$$2x_1 + 3x_2 \leqq 5$$
$$4x_1 + 5x_2 \leqq 9$$
$$x_1 \geqq 0,\ x_2 \geqq 0$$



Objective function $3x_1 + 4x_2 = k$

$\rightarrow$ Line $x_2 = -(3/4)x_1 + k/4$

Intersection of lines corresponding to two constraints is (1,1).
That is, producing one unit of product P_1 and one unit of product P_2 is the best.

Exercise E7-1: Change the above constraints into constraints by linear equalities and those of the form variable $\geqq 0$ by introducing two variables x_3 and x_4.

線形計画問題の解法

n個の変数で定義される線形計画問題

➔n次元空間において制約式に対応する半空間によって定義される凸多面体の頂点の中で目的関数の値を最適化するものを求める問題.

ただし，凸多面体の頂点をすべて列挙すれば指数時間かかる.

シンプレックス法（Dantzig, 1947年）

凸多面体の1つの頂点から出発して，その隣接頂点の中で目的関数の値が改善される頂点に移動するという操作を繰り返し，移動できなくなったときに，その頂点を最適解とする.

シンプレックス法で必ず最適解が求まる.

∵凸多面体の性質から，局所的にだけ最適という場所はない.
目的関数を改善する方向にだけ移動すれば，必ず最適解に到達する.

Algorithm for Linear Program

Linear Program defined by n variables

➔the problem of finding a vertex among those vertices of a convex polyhedron corresponding to constraints in the n-dim space at which the objective function is optimized.

Here, notice that if we enumerate all the vertices then it takes exponential time.

Simplex Algorithm（Dantzig, 1947）

Starting from a vertex of a convex polyhedron, we repeat an operation of visiting a vertex among its adjacent ones to improve the value of the objective function. When we cannot move anymore, we are at an optimal vertex.

Simplex Algorithm always finds an optimal solution.

∵Due to the property of a convex polyhedron, there is no vertex that is only locally optimal. If we move only in the direction of improving the objective function, it always reaches an optimal solution.

シンプレックス法の効率

最悪の場合には指数時間を必要とする.
しかし, 実用的には効率は良い.

線形計画問題は多項式時間で解けるか?

Khachiyan(1979)の結果

$O(nm^3L)$時間の楕円体法(ellipsoid algorithm)

n: 変数の個数, m: 制約式の個数

L: 係数を指定するのに使われる最大のビット数

Karmarker(1984)の内点法(interior method)

$O(nm^{2.5}L\log L)$時間のアルゴリズム

ATTがアルゴリズム特許を申請したことで有名

MirzaianのDPA(Deepest Peak Algorithm)

計算時間は$O(m^3n^2)$と主張しているが, 真偽は不明

Megiddo(1984), Clarkson(1986), Dyer(1986)は

変数の個数に関しては指数時間かかるが,

制約式の個数に関しては線形のアルゴリズムを提案

Efficiency of Simplex Algorithm

It takes exponential time in the worst case. However, it is efficient in practice.

Can Linear Programs be solved in polynomial time?

Khachiyan's result (1979)

Ellipsoid algorithm: $O(nm^3L)$ time

n: number of variables, m : number of constraints

L: maximum number of bits used to specify coefficeints

Karmarker's interior method (1984)

$O(nm^{2.5}L\log L)$ time algorithm

Famous for the application of algorithm patent by ATT

Mirzaian's DPA(Deepest Peak Algorithm)

He claims $O(m^3n^2)$ time, but the truth is not known.

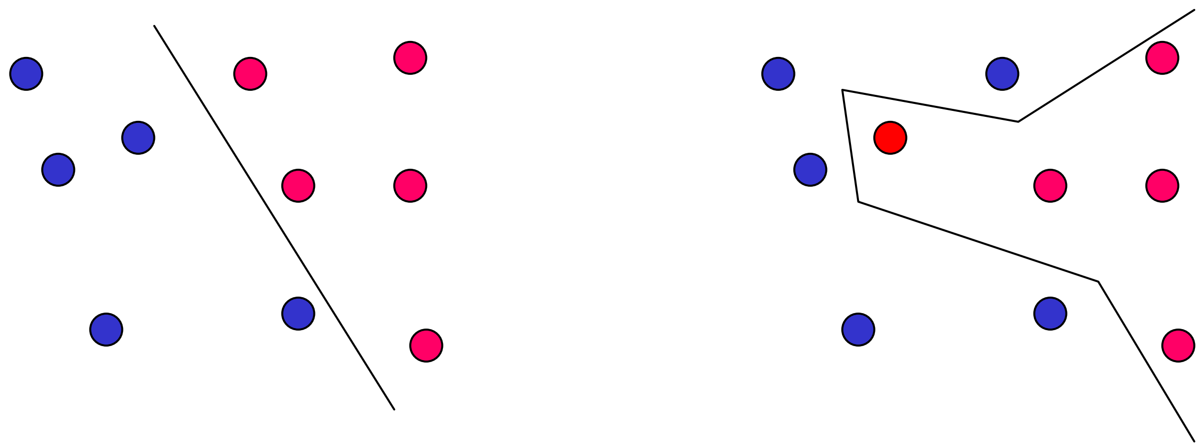
Megiddo(1984), Clarkson(1986), Dyer(1986):

They propose algorithms which take time exponential in the number of variables but linear in that of constraints.

線形計画問題として定式化できる問題

問題P17:（線形分離可能性問題）
n次元空間に2つの点集合が与えられたとき，それらを分離する超平面が存在するかどうかを判定せよ.

2次元平面では，2つの点集合を分離する直線が存在するかどうかを判定する問題となる.

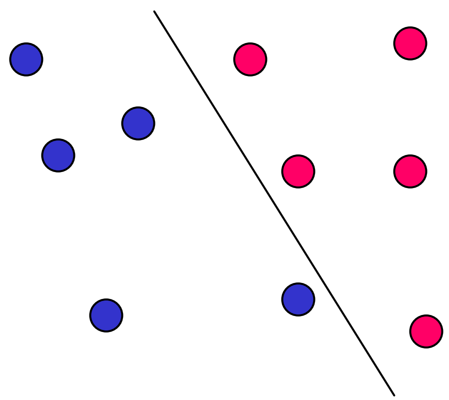


線形分離可能　　　　線形分離不可能

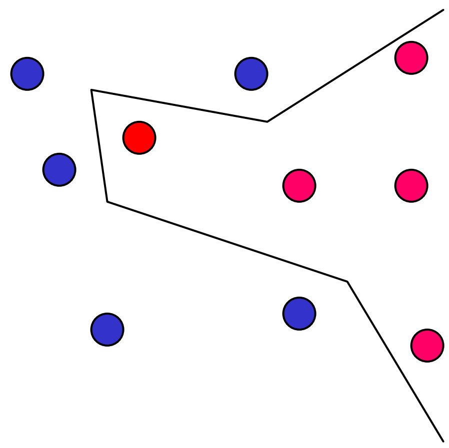
Problems formulated as Linear Programs

Problem P17: (Linear Separability)
Given two sets of points in the n-dim. space, determine whether there exists a hyperplane separating them.

In the 2-dim. plane, the problem is to determine whether there exists a line separating two sets of points.

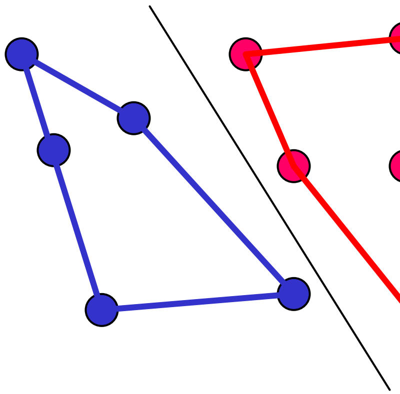


linearly separable

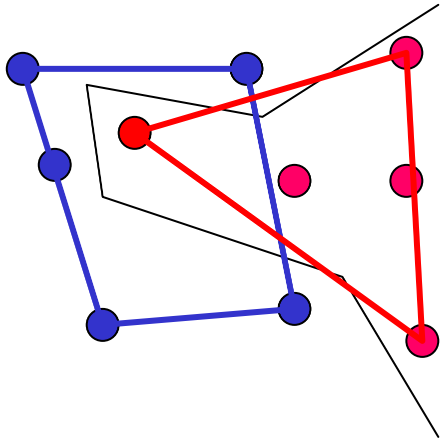


linearly nonseparable

２次元平面の場合，２つの点集合が線形分離可能であるのは
それぞれの点集合に対する凸包(最小包含凸多角形)が互いに
共通部分を持たないことである.



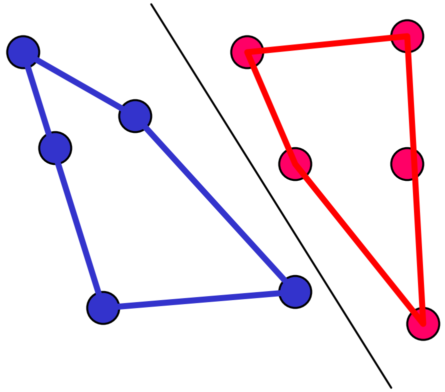
線形分離可能



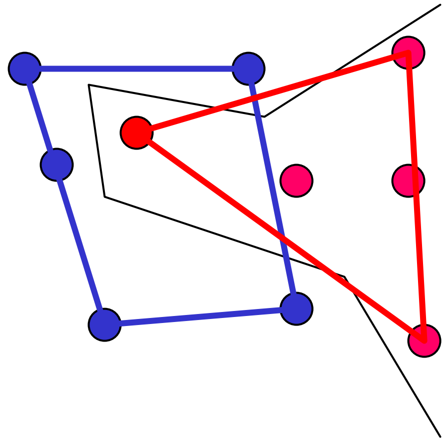
線形分離不可能

アルゴリズムP17-A0:
(1)２つの点集合RとBを入力する．ただし，n=|R|+|B|.
(2)各点集合に対する凸包CH(R)とCH(B)を求める.
(3)CH(R)とCH(B)に共通部分があるかどうかを判定.
もし共通部分があれば，解はないと出力.
そうでなければ，共通内接線を求めて，分離直線として出力.

In the 2-dim. plane, two sets of points are separable if their associated convex hulls (the smallest convex polygons containing them) have no intersection.



Linearly separable



Linearly nonseparable

Algorithm P17-A0:

(1) Input two sets R and B of points, where n=|R|+|B|.

(2) Construct convex hulls CH(R) and CH(B) for these sets of points.

(3) Determine whether CH(R) and CH(B) have intersection.
If there is any intersection, report that there is no solution.
Otherwise, find common inner tangents and report them as separating lines.

アルゴリズムP17-A0:
(1)2つの点集合RとBを入力する．ただし，n=|R|+|B|.
(2)各点集合に対する凸包CH(R)とCH(B)を求める．
(3)CH(R)とCH(B)に共通部分があるかどうかを判定．
もし共通部分があれば，解はないと出力．
そうでなければ，共通内接線を求めて，分離直線として出力．

アルゴリズムP17-A0の計算時間:
(1)は入力だけなので，O(n)時間．
(2)の凸包計算はO(n log n)時間．
(3)の共通部分の計算と共通内接線の計算はO(n)時間．
全体ではO(n log n)時間となる．

演習問題E7-2: 点集合RとBのサイズをそれぞれn, mとするとき，全体の計算時間をnとmを用いて表現せよ．nとmの値が大きく異なるとき，別の考え方ができるか？

もっと効率よく解くことは可能か？

Algorithm P17-A0:

(1) Input two point sets R and B, where n=|R|+|B|.

(2) Construct convex hulls CH(R) and CH(B) for these sets of points.

(3) Determine whether CH(R) and CH(B) have intersection.
If there is any intersection, report that there is no solution.
Otherwise, find common inner tangents and report them as separating lines.

Computation time of Algorithm P17-A0:

(1) takes O(n) time since it is only for input.

(2) takes O(n log n) time for convex hulls.

(3) takes O(n) time to compute intersection and inner tangent lines.

In total, it takes O(n log n) time.

Exercise E7-2: Let n and m be sizes of sets R and B. Represent the total computation time using n and m. If there is big difference between n and m, is there any other idea?

Is there more efficient algorithm?

線形計画法に基づくアルゴリズムP17-A1

入力の集合を

$R=\{(x_1,y_1), \ldots, (x_k, y_k)\}$, $B=\{(x_{k+1},y_{k+1}), \ldots, (x_n, y_n)\}$

とする．もしRとBを分離する直線$y=ax+b$が存在するなら，

$$y_i \leqq ax_i + b, \ i=1, \ldots, k,$$
$$y_i \geqq ax_i + b, \ i=k+1, \ldots, n$$

または，

$$y_i \geqq ax_i + b, \ i=1, \ldots, k,$$
$$y_i \leqq ax_i + b, \ i=k+1, \ldots, n$$

が成り立つはずである．

逆に，

$$b \geqq -ax_i + y_i, \ i=1, \ldots, k,$$
$$b \leqq -ax_i + y_i, \ i=k+1, \ldots, n$$

または，

$$b \leqq -ax_i + y_i, \ i=1, \ldots, k,$$
$$b \geqq -ax_i + y_i, \ i=k+1, \ldots, n$$

を満たす(a, b)の値が存在すれば，RとBは線形分離可能．
これは，２変数a, bに関する線形計画問題であるから，
$O(n)$時間で解ける．

Algorithm P17-A1 based on Linear Programming

Let input point sets be

$R=\{(x_1,y_1), \ldots, (x_k, y_k)\}$, and $B=\{(x_{k+1},y_{k+1}), \ldots, (x_n, y_n)\}$.

If there is a line separating R and B, then we must have

$$y_i \leqq ax_i + b, \ i=1, \ldots, k,$$
$$y_i \geqq ax_i + b, \ i=k+1, \ldots, n$$

or

$$y_i \geqq ax_i + b, \ i=1, \ldots, k,$$
$$y_i \leqq ax_i + b, \ i=k+1, \ldots, n.$$

Conversely, if there is (a,b) satisfying

$$b \geqq -ax_i + y_i, \ i=1, \ldots, k,$$
$$b \leqq -ax_i + y_i, \ i=k+1, \ldots, n$$

or

$$b \leqq -ax_i + y_i, \ i=1, \ldots, k,$$
$$b \geqq -ax_i + y_i, \ i=k+1, \ldots, n$$

then R and B are linearly separable.

This is a linear program for two variables, and thus it can be solved in O(n) time.

例: R={(1,2), (2,1), (3,1)}, B={(2,2), (3,3)}のとき,

線形計画問題1:

$b \geqq -1*a + 2,$
$b \geqq -2*a + 1,$
$b \geqq -3*a + 1,$
$b \leqq -2*a + 2,$
$b \leqq -3*a + 3$

線形計画問題2:

$b \leqq -1*a + 2,$
$b \leqq -2*a + 1,$
$b \leqq -3*a + 1,$
$b \geqq -2*a + 2,$
$b \geqq -3*a + 3$

演習問題E7-3:実際に実行可能領域を図示することにより,どちらの線形計画問題が実行可能解をもつかを判断せよ.

Example: Suppose R={(1,2), (2,1), (3,1)} and B={(2,2), (3,3)}.

Linear Program 1:

$b \geqq -1*a + 2,$

$b \geqq -2*a + 1,$

$b \geqq -3*a + 1,$

$b \leqq -2*a + 2,$

$b \leqq -3*a + 3$

Linear Program 2:

$b \leqq -1*a + 2,$

$b \leqq -2*a + 1,$

$b \leqq -3*a + 1,$

$b \geqq -2*a + 2,$

$b \geqq -3*a + 3$

Exercise E7-3: Determine which linear program has a feasible solution by drawing feasible regions in practice.

最短経路問題

問題P18: 辺に正の重みをもつグラフG=(V, E, c)と2頂点s, tが与えられたとき，sからtへの最小重み経路（最短経路）を求めよ.

この問題はダイクストラ法として知られる有名なアルゴリズムを用いて効率よく解けることが知られているが，線形計画問題としても定式化できる.

用意すべき変数： d_i = 頂点sから頂点v_iへの最短経路の長さ.

辺(v_i,v_j)の長さ（重み）を$c(v_i,v_j)$と表す.

このとき，制約式は

$d_1=0$ ($s=v_1$とする）

$d_j \leqq d_i + c(v_i,v_j)$ すべての辺(v_i,v_j)について，ただし，v_jはsとは異なること.

目的関数は

$\max d_n$ ただし，$v_n=t$とする.

多項式時間では解けるものの，変数の数が多いのでダイクストラ法の方が効率が良い.

Shortest Path Problem

Problem P18: Given a weighted graph G=(V,E,c) and two vertices s and t, find a shortest (minimum-weight) path from s to t.

It is known that this problem can be solved by a famous Dijkstra's algorithm. It is also formulated as a linear program.

Variables to be prepared:

d_i = length of a shortest path from s to a vertex v_i.

The length (weight) of an edge (v_i,v_j) is denoted by $c(v_i,v_j)$.

Then, the constraints become as follows:

$d_1=0$ (with $s=v_1$)

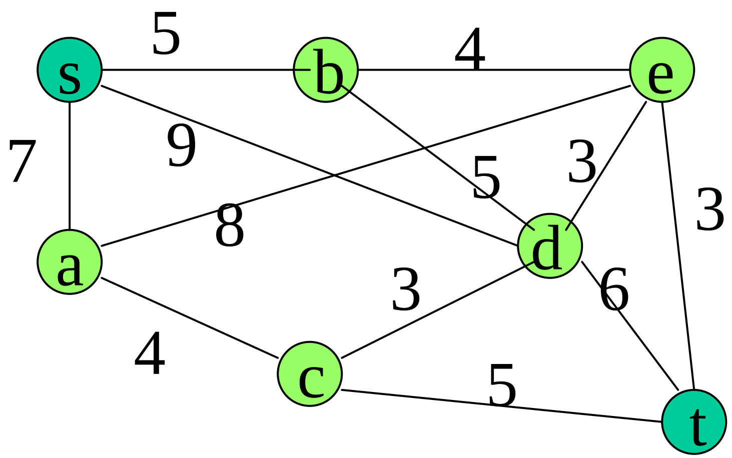
$d_j \leqq d_i + c(v_i,v_j)$ for each edge (v_i,v_j), where v_j must be different from s.

Objective function becomes

max d_n where $v_n=t$.

It can be solve in polynomial time, but Dijkstra's algorithm is more efficient since it has many variables.

演習問題 E7-4: 下記のグラフに対応する線形計画問題を実際に書き下せ.



(s, a, b, c, d, e, t)

$= (v_1, v_2, v_3, v_4, v_5, v_6, v_7)$

と番号付けること.

$d_1 = 0,$

$d_2 \leqq d_1 + 7,$

$d_2 \leqq d_6 + 8,$

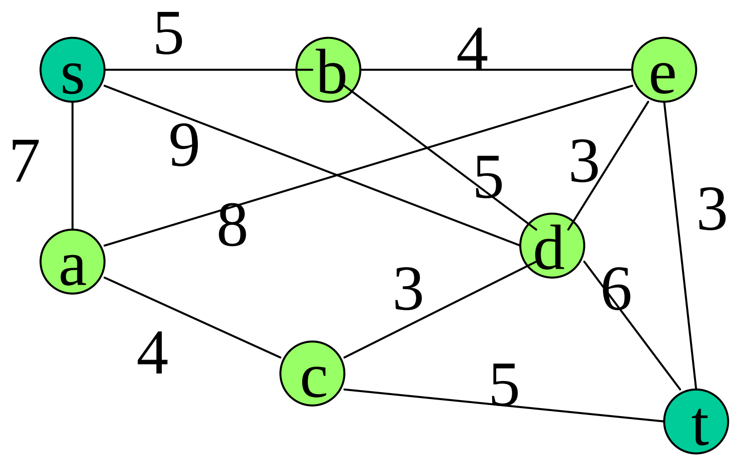
$d_2 \leqq d_4 + 4,$

$d_3 \leqq d_1 + 5,$

$d_3 \leqq d_5 + 5,$

……

Exercise E7-4: Write a linear program corresponding to the graph shown below.



Assume numbering:

(s, a, b, c, d, e, t)

= (v_1, v_2, v_3, v_4, v_5, v_6, v_7)

$d_1 = 0$,

$d_2 \leqq d_1 + 7$,

$d_2 \leqq d_6 + 8$,

$d_2 \leqq d_4 + 4$,

$d_3 \leqq d_1 + 5$,

$d_3 \leqq d_5 + 5$,

……

整数計画問題

正確には整数線形計画問題.
制約式と目的関数が線形式でなければならない点は線形計画問題と同じであるが，変数の値を整数に限定したもの.

様々な問題を整数計画問題として定式化できるという意味で非常に強力な方法であるが，残念ながら多項式時間のアルゴリズムは知られていない.

制約式と目的関数の係数は任意であるが，変数の値を0か1に限定したものを0-1整数計画問題と呼ぶが，0-1整数計画問題ですらNP完全であることが知られている.

Integer Program

Exactly, Integer Linear Program.
Constraints and objective function must be linear as in Linear Program, but variables must take integral values.

It is a very powerful scheme in the sense that various problems can be formulated as Integer Programs, but no polynomial time algorithm is known.

It is called a 0-1 Integer Program if we may be arbitrary number of constraints and any coefficients in an objective function, but values of variable are restricted to 0 or 1. It is known that even the 0-1 Integer Program is NP-complete. .

整数計画問題として定式化できる問題

n個の論理変数を$(x_1, x_2, \ldots, x_n)$とする.
論理変数x_nまたはその否定$\neg x_n$をリテラルという.
３個のリテラルをOR∨で結んだものを節という.
節をAND∧で結んだものを3SAT式という.

$$F(x_1, x_2, x_3)$$
$$= (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

真理値割り当て：各論理変数に真理値(0または1)を割り当てること
上の例で,

$$F(0,1,1) = (0 \vee \neg 1 \vee 1) \wedge (\neg 0 \vee 1 \vee \neg 1) \wedge (\neg 0 \vee 1 \vee 1) = 1$$
$$F(1,0,1) = (1 \vee \neg 0 \vee 1) \wedge (\neg 1 \vee 0 \vee \neg 1) \wedge (\neg 1 \vee 0 \vee 1) = 0$$

であるから, (0,1,1)という真理値割り当ては上式を充足するが, (1,0,1)は充足しない. 充足する真理値割り当てが存在するような3SAT式は充足可能であるという.

Problems formulated as Integre Programs

Let n logical variables be $(x_1, x_2, \ldots, x_n)$.
Logical variable x_n or its negation $\neg x_n$ is called a literal.
A clause is a connection of three literals by OR $\vee$.
3SAT expression is a combination of clauses by AND $\wedge$.

$$F(x_1, x_2, x_3)$$
$$= (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

Truth assignment : assignment of truth value (0 or 1) to each variable.
In the above example, we have

$$F(0,1,1) = (0 \vee \neg 1 \vee 1) \wedge (\neg 0 \vee 1 \vee \neg 1) \wedge (\neg 0 \vee 1 \vee 1) = 1,$$
$$F(1,0,1) = (1 \vee \neg 0 \vee 1) \wedge (\neg 1 \vee 0 \vee \neg 1) \wedge (\neg 1 \vee 0 \vee 1) = 0,$$

and so the truth assignment (0,1,1) satisfies the expression, but (1,0,1) does not.
A 3SAT expression is called satisfiable if there is a truth assignment satisfying it.

問題P19: (充足可能性問題3SAT)
n個の変数とm個の節からなる3SATの式が与えられたとき，
それが充足可能かどうかを判定し，充足可能なら，式を充足する
真理値割り当てを求めよ.

この問題は代表的なNP完全問題である.

整数計画問題としての定式化

各論理変数の値を整数値0, 1に限定する制約式

$$0 \leqq x_i \leqq 1, \text{ integer } x_i, \ \ i=1, 2, \dots , n$$

論理変数x_iの否定$\neg x_i$は $1-x_i$ と表現する.

各節に関する制約式

$$(x_1 \lor \neg x_2 \lor x_3) \Rightarrow x_1 + (1-x_2) + x_3 \geqq 1$$

後は各節に対応する制約式をANDの形で並べればよい.

$$(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \Rightarrow$$

$$(x_1 \lor \neg x_2 \lor x_3) \Rightarrow x_1 + (1-x_2) + x_3 \geqq 1,$$

$$(\neg x_1 \lor x_2 \lor \neg x_3) \Rightarrow (1-x_1) + x_2 + (1-x_3) \geqq 1,$$

$$(\neg x_1 \lor x_2 \lor x_3) \Rightarrow (1-x_1) + x_2 + x_3 \geqq 1.$$

Problem P19: (3SAT: 3-Satisfiablility Problem)
Given a 3SAT expression consisting of n variables and m clauses, determine whether it is satisfiable or not, and find a truth assignemt satisfying it if it is.

This problem is a typical NP-complete problem.

Formulation as an Integer Linear Program

Constraints for logical variables to take only 0 or 1

$$0 \leqq x_i \leqq 1, \text{ integer } x_i, \ \ i=1, 2, \dots, n$$

Represent the negation $\neg x_i$ of variable x_i as $1-x_i$.

Constraint associated with each clause

$$(x_1 \vee \neg x_2 \vee x_3) \Rightarrow x_1 + (1-x_2) + x_3 \geqq 1$$

Then, constraints for clauses are connected by putting AND

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \Rightarrow$$

$$(x_1 \vee \neg x_2 \vee x_3) \Rightarrow x_1 + (1-x_2) + x_3 \geqq 1,$$

$$(\neg x_1 \vee x_2 \vee \neg x_3) \Rightarrow (1-x_1) + x_2 + (1-x_3) \geqq 1,$$

$$(\neg x_1 \vee x_2 \vee x_3) \Rightarrow (1-x_1) + x_2 + x_3 \geqq 1.$$