＜数理計画モデル＞

・線形計画問題

・ネットワーク計画問題

簡単に解ける

・非線形計画問題

・組合せ計画問題

難しい
（厳密な最適化は困難な場合が多い）

非線形計画モデル

＜ポートフォリオ選択問題＞

資産w円を3種類の株式A_1, A_2, A_3に分散して投資する。

株式の現在価格：p_1, p_2, p_3円

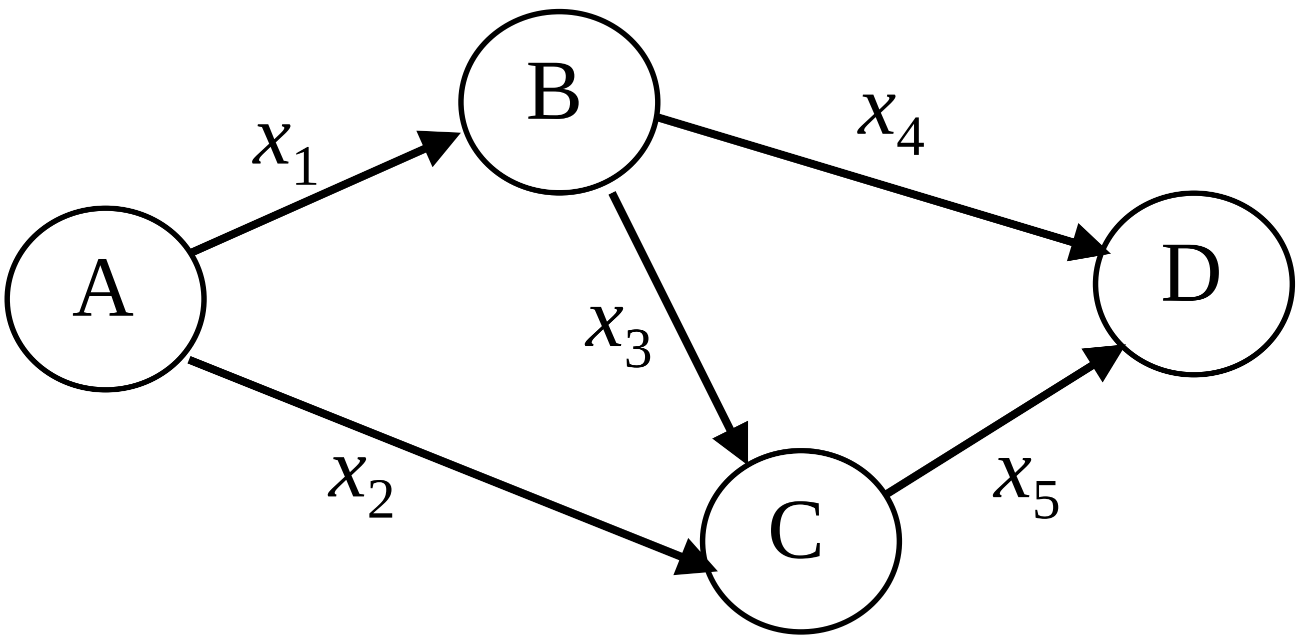
1カ月後の株式の価格：P_1, P_2, P_3円

$\longrightarrow$ 確率変数

＜交通流割当問題＞

W台の車がAからDへ向かう

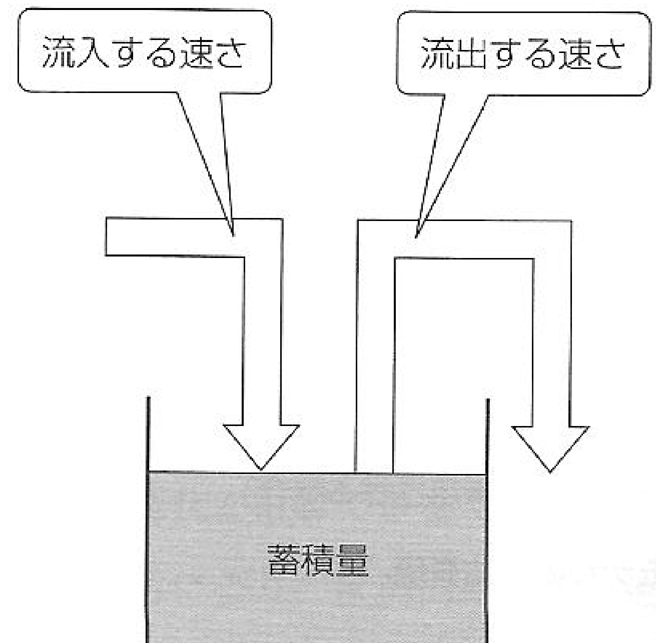
各道路を通る車の台数: x_1, x_2, x_3, x_4, x_5



② 動的モデルの作成

1―― 蓄積量と変化の速さ

時間的に変化する現象は，**動的システム**ともよばれる。動的モデル作成の目的は，動的なシステムの動作をとらえて図や数式に表現することである。動的システムの内部には，物，エネルギーや情報の蓄積量があり，これによってシステムの動作が決まる。この蓄積量の時間的な変化は，物，エネルギーや情報がシステムに流入する速さやシステムから流出する速さによって決定される。単位時間あたりの蓄積量の変化量を「変化の速さ」とよぶことにすると，1章3節例題2で学習した(→p.31)ように，変化の速さが一定である場合，ある「時間間隔」が経過したときの蓄積量の変化量は次のようになる。



変化量＝変化の速さ×時間間隔

変化後の蓄積量＝現在の蓄積量＋変化量

つまり，数式モデルは次のようにあらわされる[①]。

（1）「変化の速さ」をあらわす式

（2）変化後の蓄積量＝現在の蓄積量＋変化の速さ×時間間隔

■表2　蓄積量と変化の速さの例

蓄積量	変化の速さ
水量	流入，流出
人口	出生，死亡
預金額	預け入れ・利息の発生，引き出し
在庫量	仕入れ，販売
速度	加速，減速

例題3　ロケットの高度を求める

あるロケットを打ち上げたとき，ロケットの高度は時間（秒）の2乗を2倍したものであった。1～5秒後の高度の変化を求めてみよう。

ロケットの高度は，

1秒後には，$2 \times (1)^2 = 2\,\mathrm{m}$

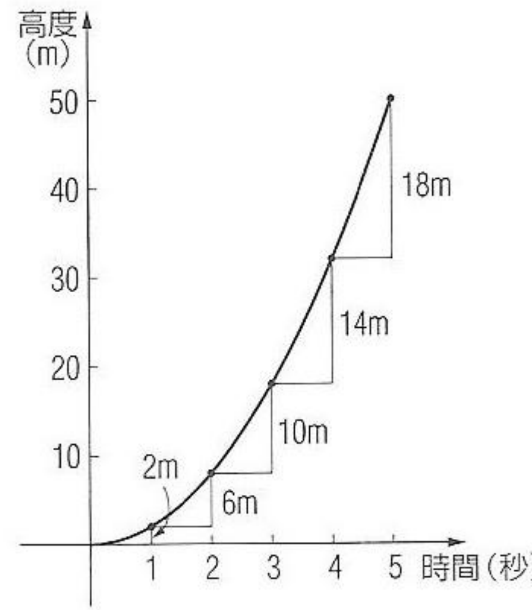
2秒後には，$2 \times (2)^2 = 8\,\mathrm{m}$　となる（表3）。

表3からわかるように，ロケットの1秒後ごとの上昇距離は全区間で一定でなく，しだいに大きくなっていくことがわかる。

すなわち，ロケットの高度は，

変化後の高度＝現在の高度＋上昇距離×時間間隔

となる。



■図6　ロケットの高度

■表3　ロケットの高度

時間（秒）	0	1	2	3	4	5
変化後の高度（m）	0	2	8	18	32	50
上昇距離（m/秒）		2	6	10	14	18

参考にしよう　差分方程式と微分方程式

平均変化率は2点の直線の傾きであり，例題4のロケットの高度のように，最初の高度から各時間に対する高度を近似的に求めていくことができる。

平均変化率の時間間隔Δtがじゅうぶん小さい場合，関数$y=f(t)$の変化を$f(t+\Delta t)-f(t)=\Delta y$とおくと，

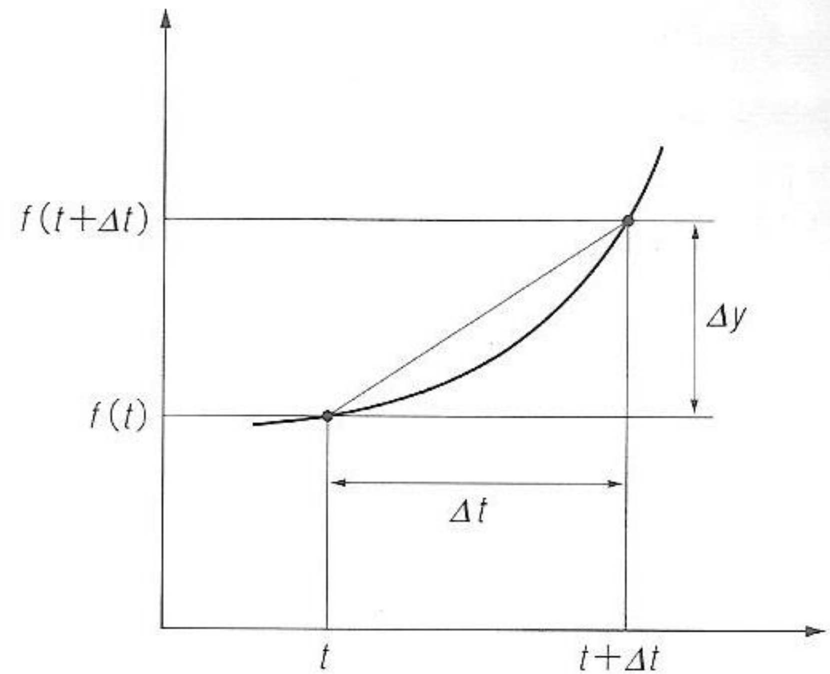
$$\frac{f(t+\Delta t)-f(t)}{\Delta t}=\frac{\Delta y}{\Delta t}=4t$$

であり，このような関係式を**差分方程式**という。**オイラー法**は，差分方程式を数値的にといていく方法である。

また，時間間隔Δtをかぎりなく0に近づけたとき，2点の直線の傾きは関数$f(t)$の接線となる。接線の傾きを$f'(t)$としたとき，

$f'(t)=4t$，あるいは$\dfrac{dy}{dt}=4t$

のような関係式を**微分方程式**という。微分方程式を近似したものが差分方程式である。微分方程式の数値解法には，オイラー法や**ルンゲクッタ法**とよばれる方法がある。オイラー法は誤差が大きいため，精度が求められる場合は，ルンゲクッタ法が利用される。

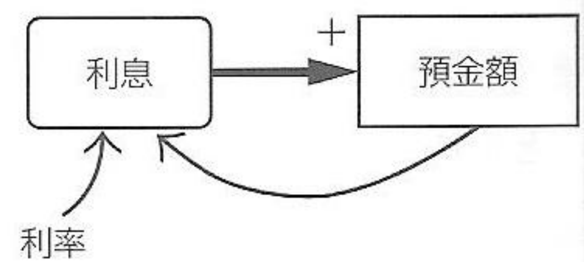


2——変化の速さが蓄積量に比例する現象

銀行に，利率が年５％の１年複利の複利法[①]で10000円を預けたとすると，５％の利率によりはじめの年は500円の利息がつき，預金額が10500円となる。次の年は，元金が10500円となるので，利息はその５％で525円になる。そして10年目には約775円の利息がつき，預金額は16289円になる。このように複利法の銀行預金では，預金額を「蓄積量」とすると利息が「変化の速さ」をあらわす。利息は，利率を比例定数として蓄積量に比例して決まる。銀行預金のモデルを図であらわすと図10のようになる。

このほかに，変化の速さが蓄積量に比例する現象には，人口の増加，生物の個体数の増加，携帯電話利用者数の増加などがある。

①利息を元金に加え，これを次の期間の新元金として利息を計算する方法。



■図10　銀行預金のモデル

1 運動の法則

①速度＝運動量／質量

②運動量＝速度×質量
または，
運動量＝力×時間

ここでは，物の動きをモデル化し，物の動きを支配する約束事を学習しよう。

運動のモデルを表現する基本構造は，「位置」「速度」「運動量」「力」「質量」の５つの要素で成り立っている。それぞれの要素の意味は，表１のようにとらえることにする。

■表1　運動のモデルを表現する要素

要素	意味
位置	基準点からみて，どこにあるのかをあらわす。位置は，ストックとして「速度」の蓄積により求める。
速度①	物体の位置が変化する速さ。速度には大きさと方向がある。バイフロー(→p.106)としてあらわす。
運動量②	物体に加わるあらゆる「力」の蓄積。「力」の蓄積なので，ストックであらわす。
力	運動量を増減する，つまり物体におよぼされる「押す力」をあらわす。「力」にも向きがあり，バイフローであらわす。
質量	物体がもつ「速度が変化しにくい性質」の大きさ。同じ運動量の場合，質量が少ない（軽い）場合は速くなり，逆に質量が多い（重い）場合はおそくなる。

バンジージャンプやスカイダイビングを行った場合に，足につけたゴムやパラシュートの影響を受ける前の人が落下するようすをモデル化してみよう。

加速度：－9.8（m／s^2）

力＝質量×加速度　…………（1）

位置（高さ）：100（m）

質量：75（kg）

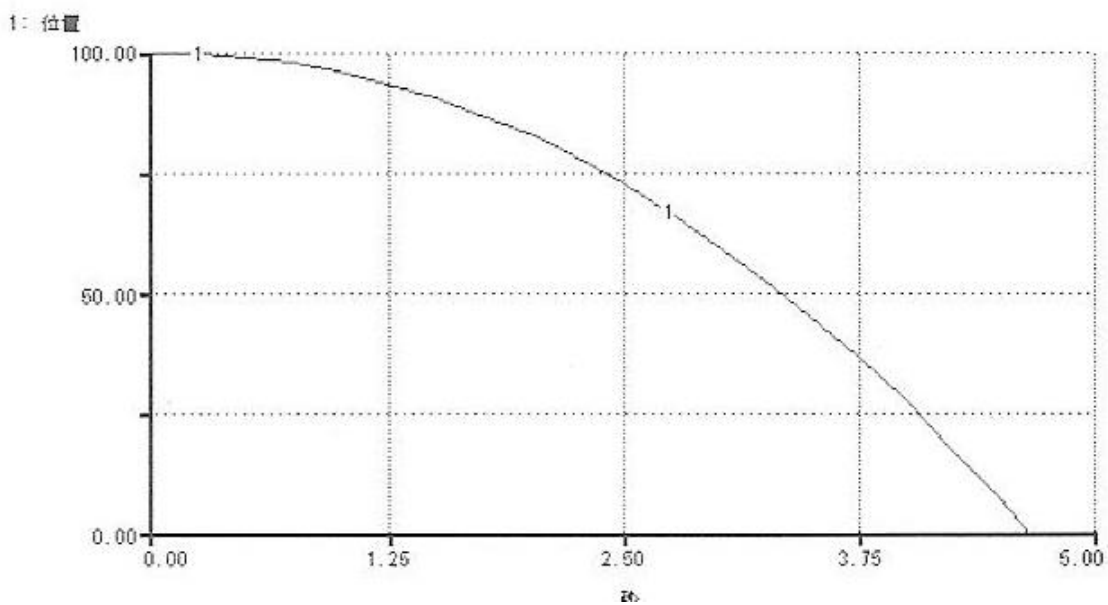
運動量：0（kg m／s）

速度＝運動量×1／質量

次の条件で，このモデルのシミュレーションを行う。

時間の単位：秒　　期間：開始0秒　終了5秒

時間間隔：0.25　　計算方法：オイラー法



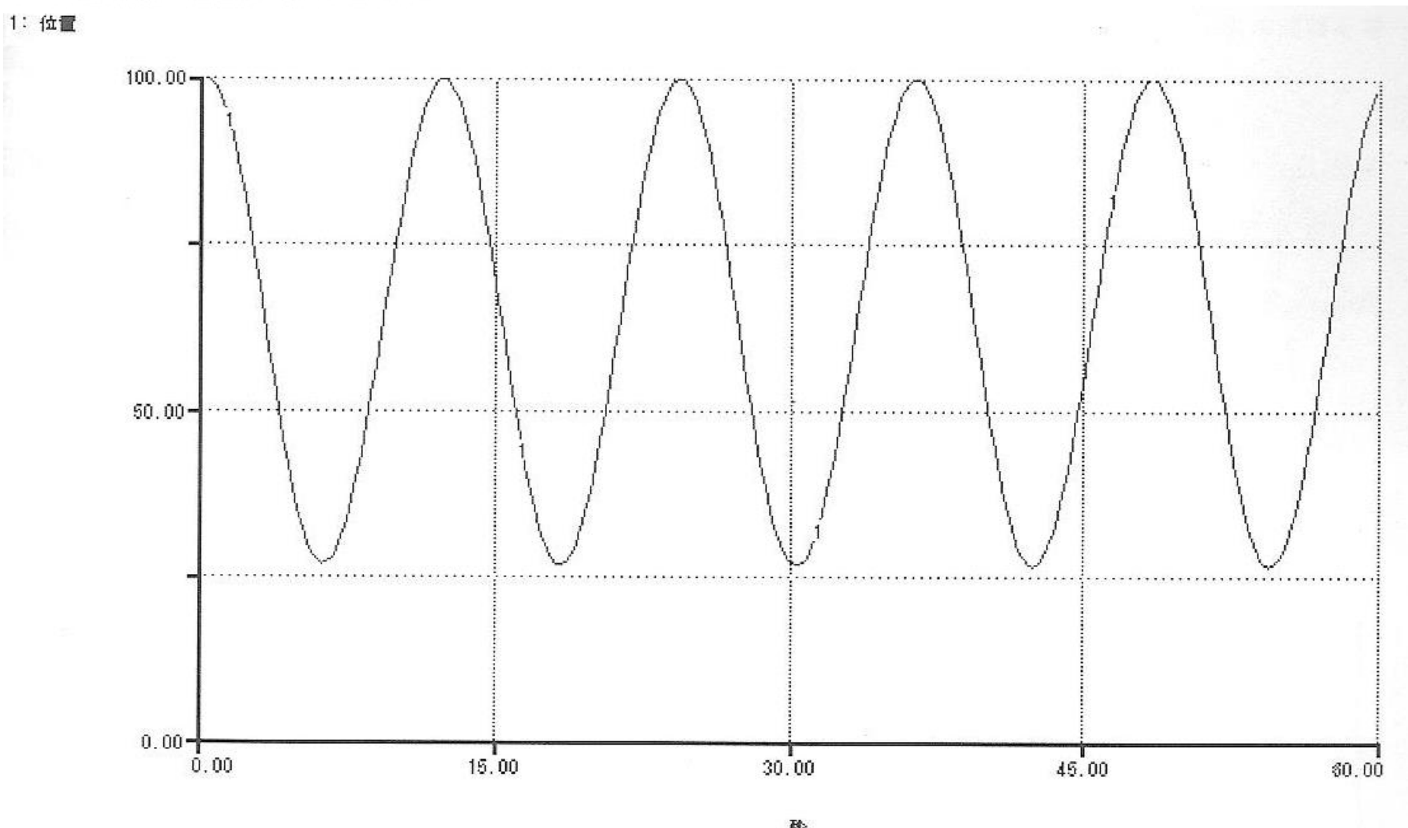
例題3　バンジージャンプのモデルをつくる

バンジージャンプは，足につけたゴムひもがのび縮みすることにより，上下運動をくりかえす。このようすをシミュレーションしてみよう。ただし，簡単にするため，ジャンプ台の高さからゴムひもの長さまでの自由落下については考慮しないものとする。

ジャンプ台の高さ：100（m）
フック定数：20（kg/s^2）
台からの位置＝位置－ジャンプ台の高さ
復元力＝フック定数×台からの位置

次の条件で，このモデルのシミュレーションを行う。

時間の単位：秒　　期間：開始0秒　終了60秒
時間間隔：0.25　　計算方法：ルンゲクッタ法



非線形計画問題

- 制約なし問題
- 制約つき問題

＜非線形計画問題＞

目的関数：　$f(\boldsymbol{x}) \longrightarrow$ 最小

制約条件：　$\boldsymbol{x} \in S$

＜制約なし問題＞

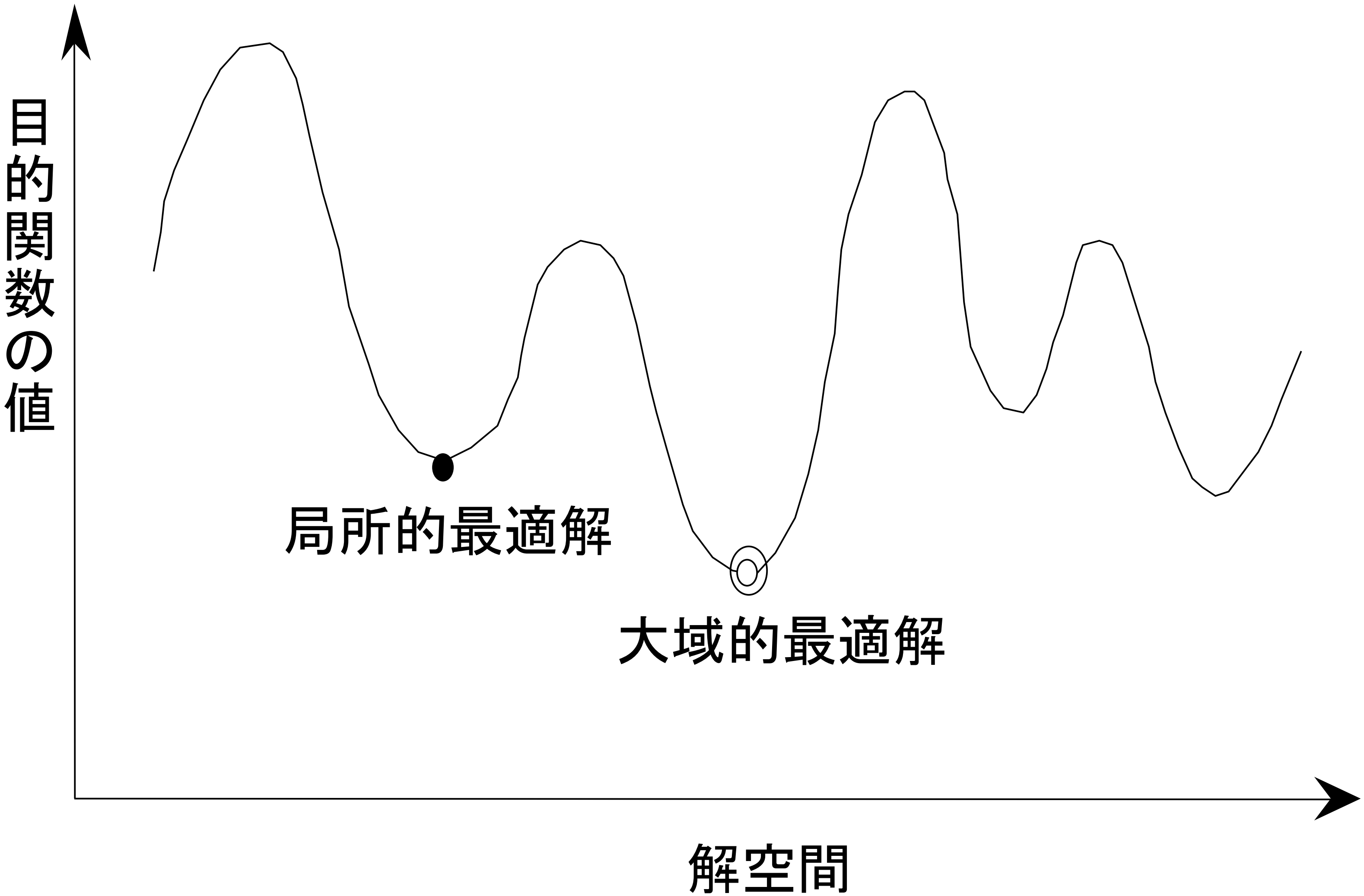
目的関数：　$f(\boldsymbol{x}) \longrightarrow$ 最小

局所的最適解と大域的最適解

大域的最適解：実行可能領域S全体において目的関数fが最小となる点

局所的最適解：十分近くのどの実行可能解よりも目的関数fの値が小さい点

＜最適化の概念＞



＜凸計画問題＞

目的関数f : 凸関数

実行可能領域 : 凸集合

局所的最適解＝大域的最適解

凸関数f : $\boldsymbol{x},\boldsymbol{y} \in \mathrm{R}^n,\ 0 \leqq \alpha \leqq 1 \Rightarrow f(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}) \leqq \alpha f(\boldsymbol{x})+(1-\alpha)f(\boldsymbol{y})$

凸集合S : $\boldsymbol{x},\boldsymbol{y} \in \mathrm{S},\ 0 \leqq \alpha \leqq 1 \Rightarrow \alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y} \in \mathrm{S}$

＜一般的な非線形計画問題＞

いくつもの局所的最適解のなかから大域的最適解を見つけることは非常に困難

→ 局所的最適解を求めることが当面の目標

関数の勾配とヘッセ行列

$\nabla f(\boldsymbol{x})$: 点$\boldsymbol{x}$における関数fの勾配

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix}$$

例）

$$f(\boldsymbol{x}) = 5x_1{}^2 - 6x_1x_2 + 5x_2{}^2 - 10x_1 + 6x_2$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = 10x_1 - 6x_2 - 10$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_2} = -6x_1 + 10x_2 + 6$$

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} 10x_1 - 6x_2 - 10 \\ -6x_1 + 10x_2 + 6 \end{bmatrix}$$

点$\boldsymbol{a}=(0,0)^{\mathrm{T}}$, $\boldsymbol{b}=(2,0)^{\mathrm{T}}$, $\boldsymbol{c}=(3,1)^{\mathrm{T}}$における関数$f$の勾配

$$\nabla f(\boldsymbol{a}) = \begin{bmatrix} -10 \\ 6 \end{bmatrix},\ \nabla f(\boldsymbol{b}) = \begin{bmatrix} 10 \\ -6 \end{bmatrix}, \nabla f(\boldsymbol{c}) = \begin{bmatrix} 14 \\ -2 \end{bmatrix}$$

$\nabla^2 f(\boldsymbol{x})$：点$\boldsymbol{x}$における関数$f$のヘッセ行列

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_1{}^2} & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_1 x_2} & \cdots & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_1 x_n} \\ \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_2{}^2} & \cdots & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f(\boldsymbol{x})}{\partial x_n{}^2} \end{bmatrix}$$

対称行列

例）

$$f(\boldsymbol{x}) = 5x_1^{\,2} - 6x_1x_2 + 5x_2^{\,2} - 10x_1 + 6x_2$$

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} 10x_1 - 6x_2 - 10 \\ -6x_1 + 10x_2 + 6 \end{bmatrix}$$

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix}$$

固有値：$\lambda_1=4,\ \lambda_2=16$

固有ベクトル：$x_1=(1,1)^T,\ x_2=(1,-1)^T$

制約なし問題の最適性条件

$$\nabla f(x^*) = \mathbf{0}$$

x^*が局所的最適解であるための必要条件

x^* : 関数fの停留点　　1次の必要条件

凸関数fの任意の停留点x^*は（大域的）最適解

例）

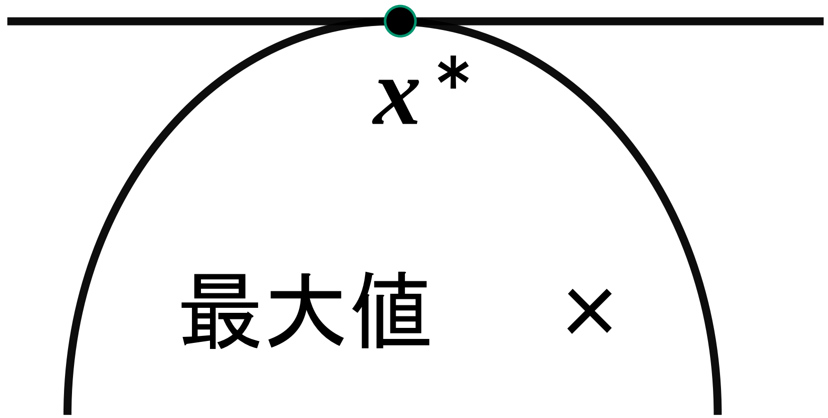
$$f(\boldsymbol{x}) = 5x_1^2 - 6x_1x_2 + 5x_2^2 - 10x_1 + 6x_2$$

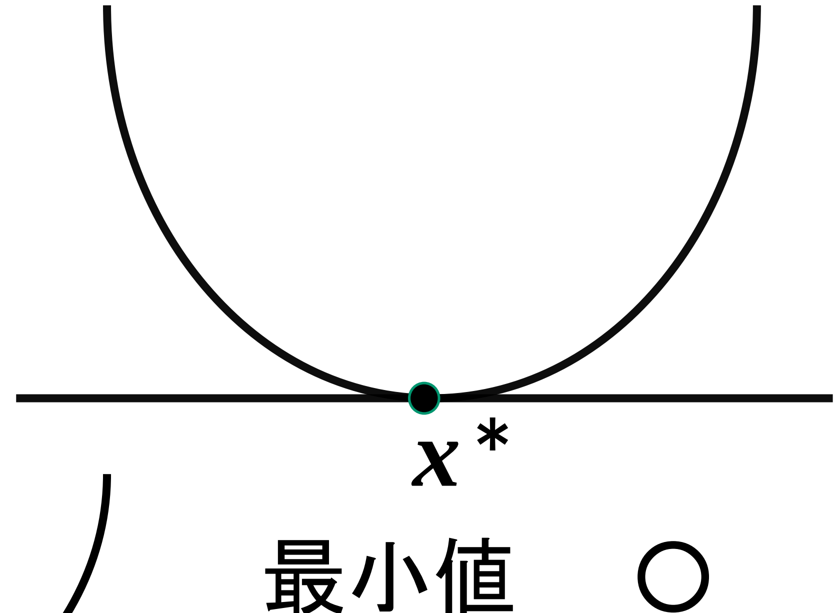
$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} 10x_1 - 6x_2 - 10 \\ -6x_1 + 10x_2 + 6 \end{bmatrix}$$

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$$

$$\begin{cases} 10x_1 - 6x_2 - 10 = 0 \\ -6x_1 + 10x_2 + 6 = 0 \end{cases} \qquad \begin{cases} x_1 = 1 \\ x_2 = 0 \end{cases}$$

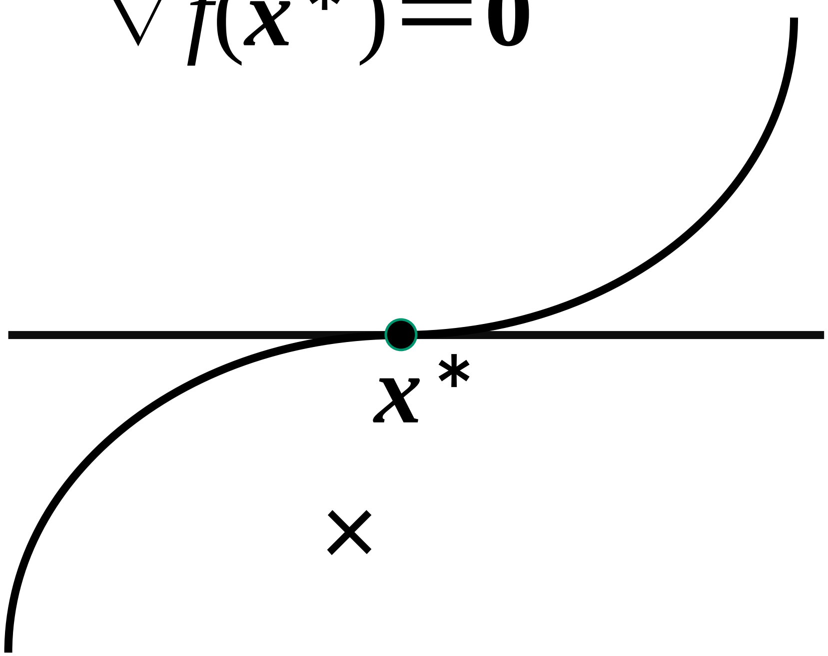
$$\boldsymbol{x}^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



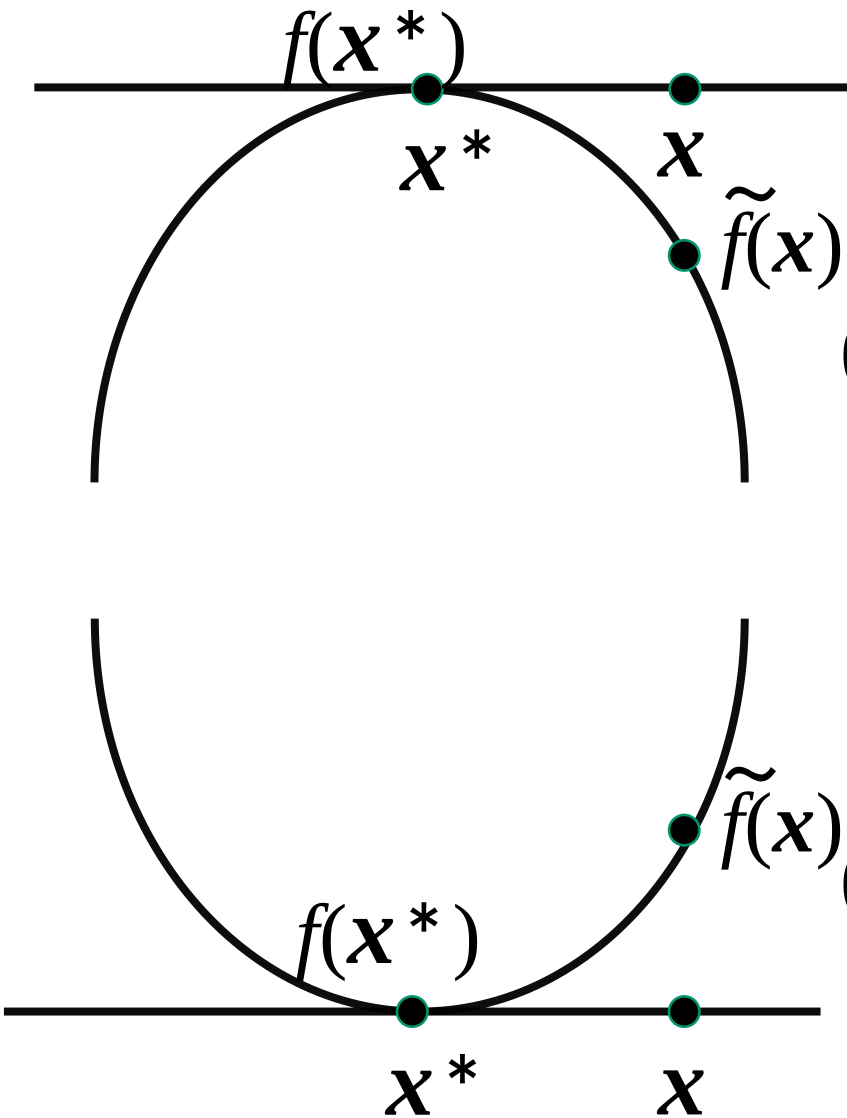


1次の必要条件

$$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$$



$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^*) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x}-\boldsymbol{x}^*)$$



$$\tilde{f}(\boldsymbol{x}) < f(\boldsymbol{x}^*)$$

$$(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x}-\boldsymbol{x}^*) < 0$$

$$\tilde{f}(\boldsymbol{x}) > f(\boldsymbol{x}^*)$$

$$(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x}-\boldsymbol{x}^*) > 0$$

$\nabla^2 f(\boldsymbol{x}^*)$ は正定値

A: 半正定値行列

$\boldsymbol{x}^{\mathrm{T}} \mathrm{A} \boldsymbol{x} \geqq 0$ （すべての$\boldsymbol{x}$について）

$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$

$\nabla^2 f(\boldsymbol{x}^*)$ は半正定値

最適性の**2次の必要条件**

$\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$

$\nabla^2 f(\boldsymbol{x}^*)$ は正定値

最適性の**2次の十分条件**

＜制約なし問題の解法＞

非線形計画問題の最適解を有限回の演算で厳密に求めることは困難

一般には，最適解に収束するような点列 $\{\boldsymbol{x}^{(k)}\}$ を次々と生成する反復法が用いられる.

最急降下法

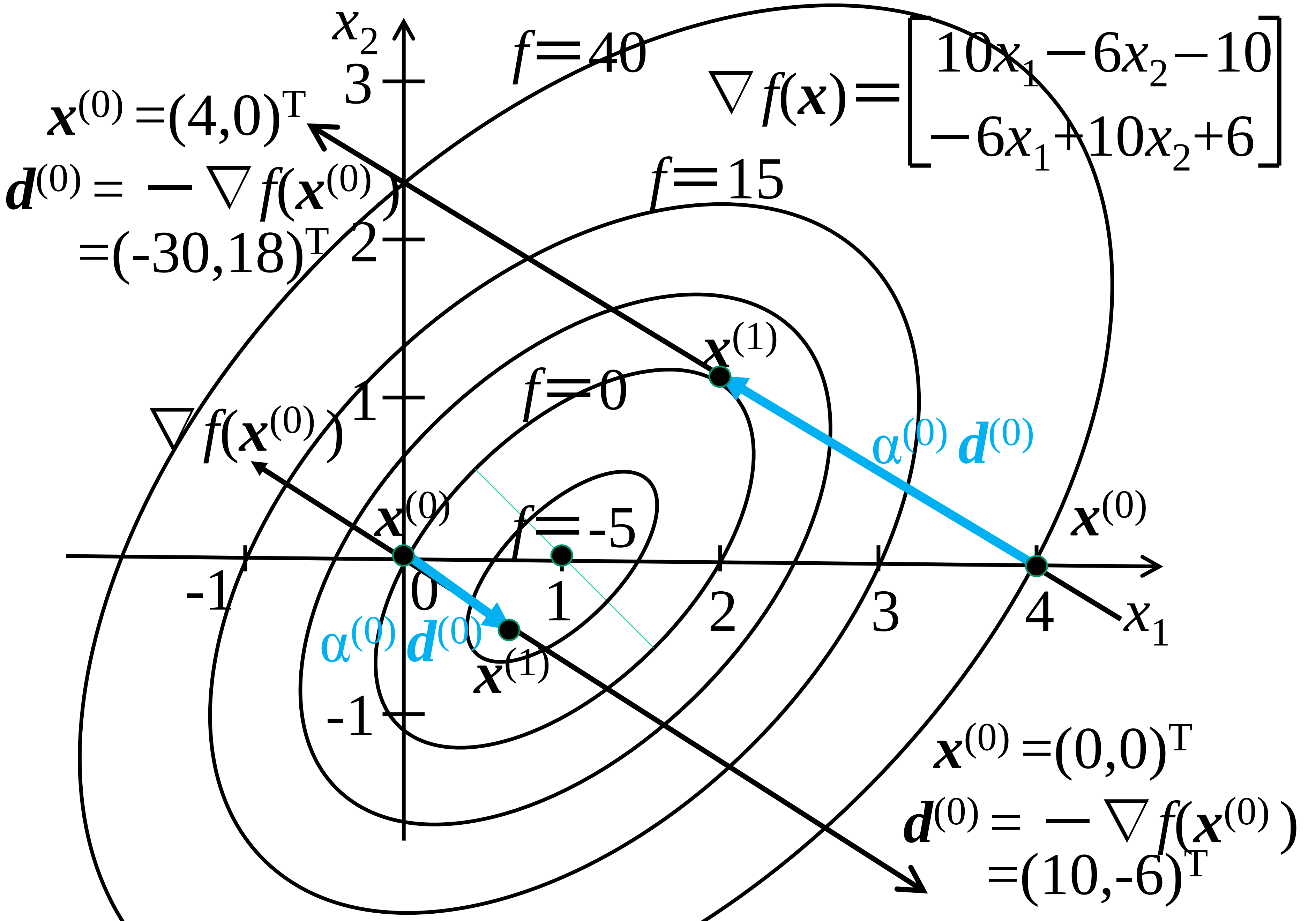
＜最急降下法＞

(0) 出発点$\boldsymbol{x}^{(0)}$を選び，k:=0 とおく.

(1) $\nabla f(\boldsymbol{x}^{(k)})=\mathbf{0}$ ならば計算終了．さもなければ
$\boldsymbol{d}^{(k)} := -\nabla f(\boldsymbol{x}^{(k)})$ とおいてステップ(2)へ.

(2) ステップ幅$\alpha^{(k)}$を求め，次の点
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \alpha^{(k)} \boldsymbol{d}^{(k)}$ を定める.
k:= k +1とおいてステップ(1)へ戻る.

$f(\boldsymbol{x}) = 5x_1^2 - 6x_1x_2 + 5x_2^2 - 10x_1 + 6x_2$ の等高線



ニュートン法と準ニュートン法

＜ニュートン法＞

(0) 出発点$\boldsymbol{x}^{(0)}$を選び，k:=0 とおく.

(1) $\nabla f(\boldsymbol{x}^{(k)})=\mathbf{0}$ ならば計算終了. さもなければ

$$\nabla^2 f(\boldsymbol{x}^{(k)})\boldsymbol{d} = -\nabla f(\boldsymbol{x}^{(k)})$$

の解$\boldsymbol{d}^{(k)}$を求め，ステップ(2)へ.

(2) 次の点を $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k)}$ とする.

k:= k +1とおいてステップ(1)へ戻る.

$$f(\boldsymbol{x}) = 5x_1^2 - 6x_1x_2 + 5x_2^2 - 10x_1 + 6x_2$$

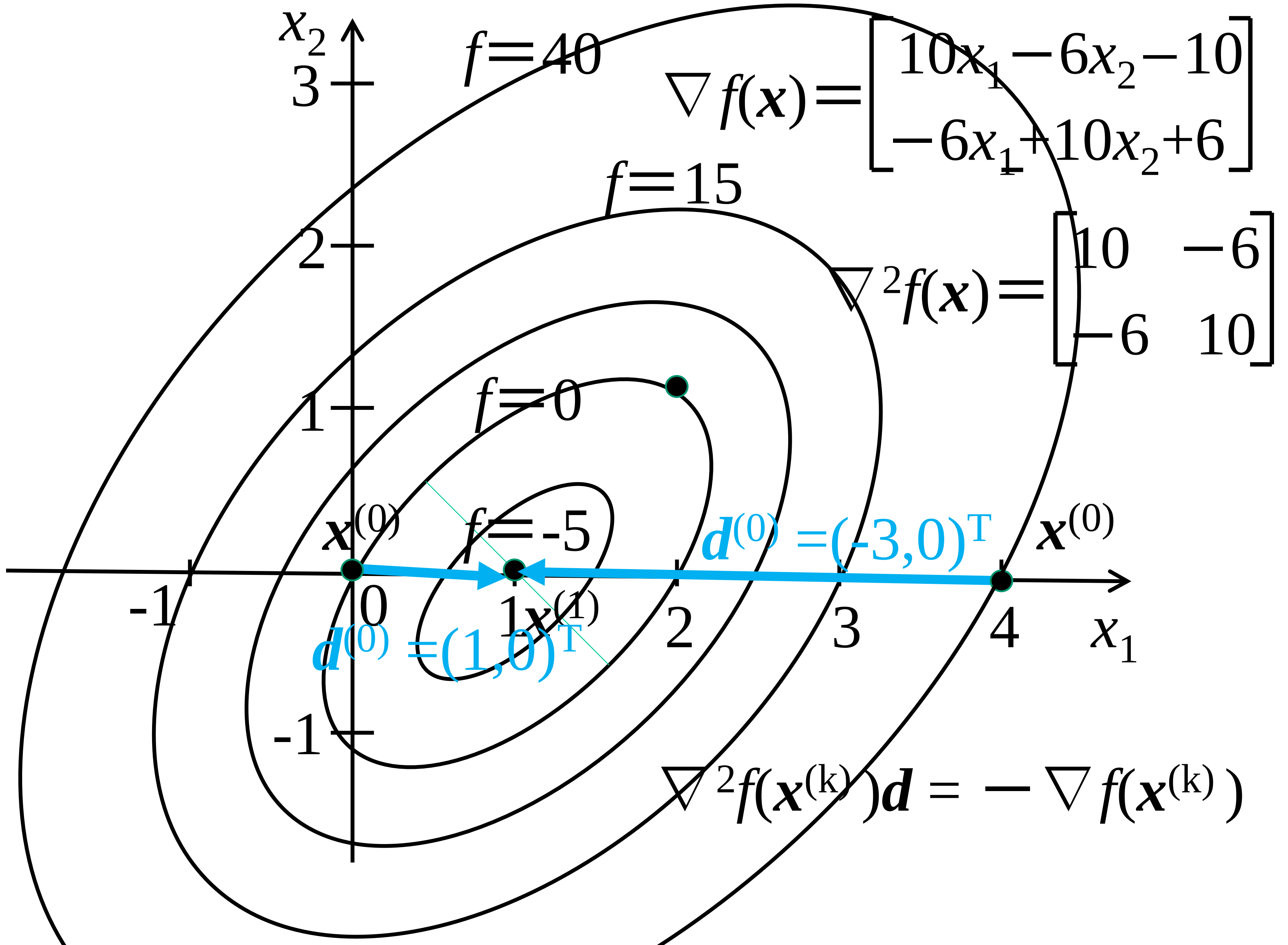
$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} 10x_1 - 6x_2 - 10 \\ -6x_1 + 10x_2 + 6 \end{bmatrix} \quad \nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix}$$

$$\nabla^2 f(\boldsymbol{x}^{(k)})\boldsymbol{d} = -\nabla f(\boldsymbol{x}^{(k)})$$

$$\boldsymbol{x}^{(0)} = (0,0)^{\mathrm{T}} \quad \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix} \boldsymbol{d} = \begin{bmatrix} 10 \\ -6 \end{bmatrix} \quad \begin{aligned} \boldsymbol{d}^{(0)} &= (1,0)^{\mathrm{T}} \\ \boldsymbol{x}^{(1)} &= \boldsymbol{x}^{(0)} + \boldsymbol{d}^{(0)} \\ &= (1,0)^{\mathrm{T}} \end{aligned}$$

$$\boldsymbol{x}^{(0)} = (4,0)^{\mathrm{T}} \quad \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix} \boldsymbol{d} = \begin{bmatrix} -30 \\ 18 \end{bmatrix} \quad \begin{aligned} \boldsymbol{d}^{(0)} &= (-3,0)^{\mathrm{T}} \\ \boldsymbol{x}^{(1)} &= \boldsymbol{x}^{(0)} + \boldsymbol{d}^{(0)} \\ &= (1,0)^{\mathrm{T}} \end{aligned}$$

$f(\boldsymbol{x}) = 5x_1^2 - 6x_1x_2 + 5x_2^2 - 10x_1 + 6x_2$ の等高線



組合せ計画モデル

<生産計画問題>

目的関数：　$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \longrightarrow$ 最大

制約条件：　$\mathrm{A}\boldsymbol{x} \leqq \boldsymbol{b},\ \ \boldsymbol{x} \geqq 0$

$\boldsymbol{x}$ の各要素は整数

整数計画問題

組合せ計画問題

有限個の要素からなる実行可能領域のなかで目的関数が最小となる解を見つける問題

例）　・最短路問題

・線形計画問題

　有限個の実行可能基底解から最適解を見つける

・ネットワーク計画問題

　特殊な線形計画問題

＜組合せ計画問題＞

実行可能解の数は有限 → すべての実行可能解の目的関数を計算すれば最適解が求まる

n個の0-1変数$x_i (i=1,..,n)$ の組

$\boldsymbol{x}=(x_1,\ldots, x_n)$

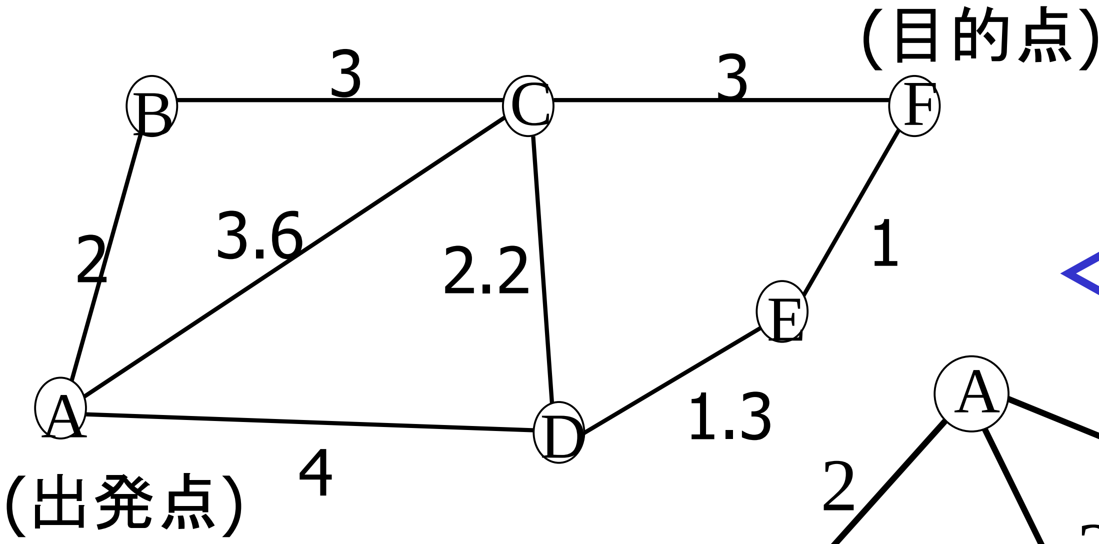
のとりうる値　2^n 個

変数が数十個程度の問題ですら現実には取り扱えない

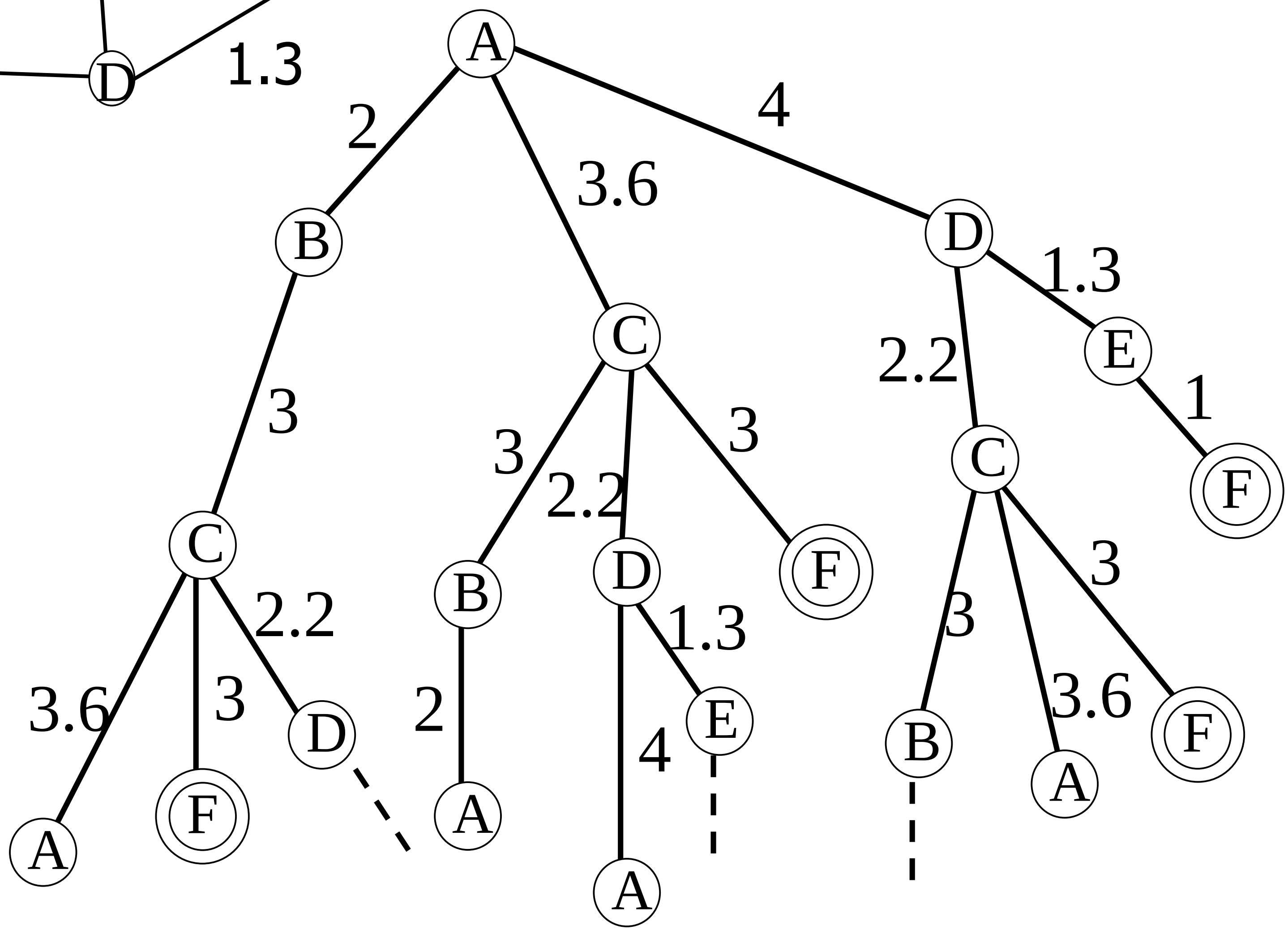
分枝限定法

実行可能解を列挙するために場合分けを行っていく過程で，最適解が得られる見込みのない不必要な場合分けをできるだけ省略して，探索する範囲を絞り込むことにより計算時間を短縮する方法.

様々な問題に対して用いることのできる一般的な計算原理



＜経路探索木＞



出発点からの経路が最も短い点を展開する

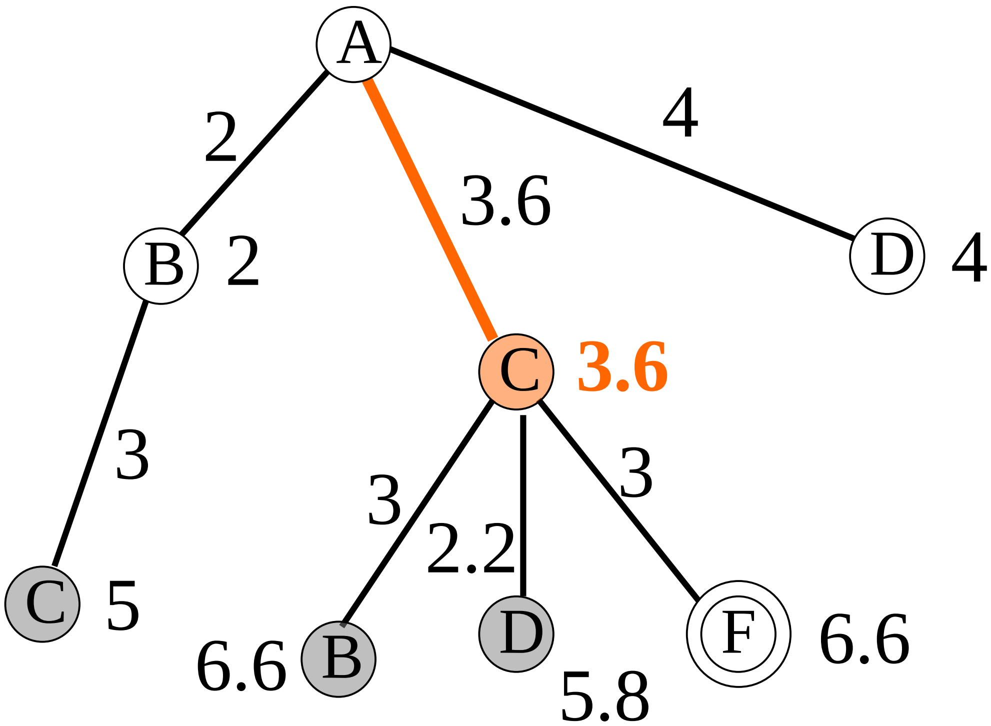
⇒ 最良優先探索

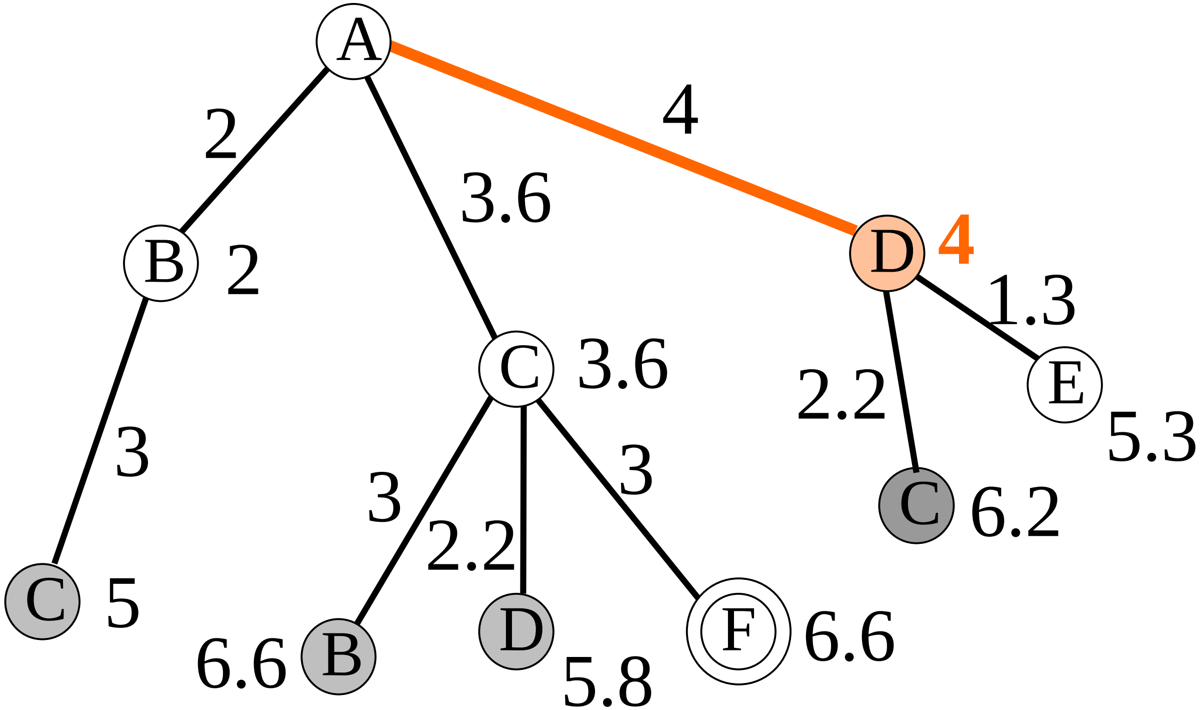
より短い経路が得られている点は展開しない

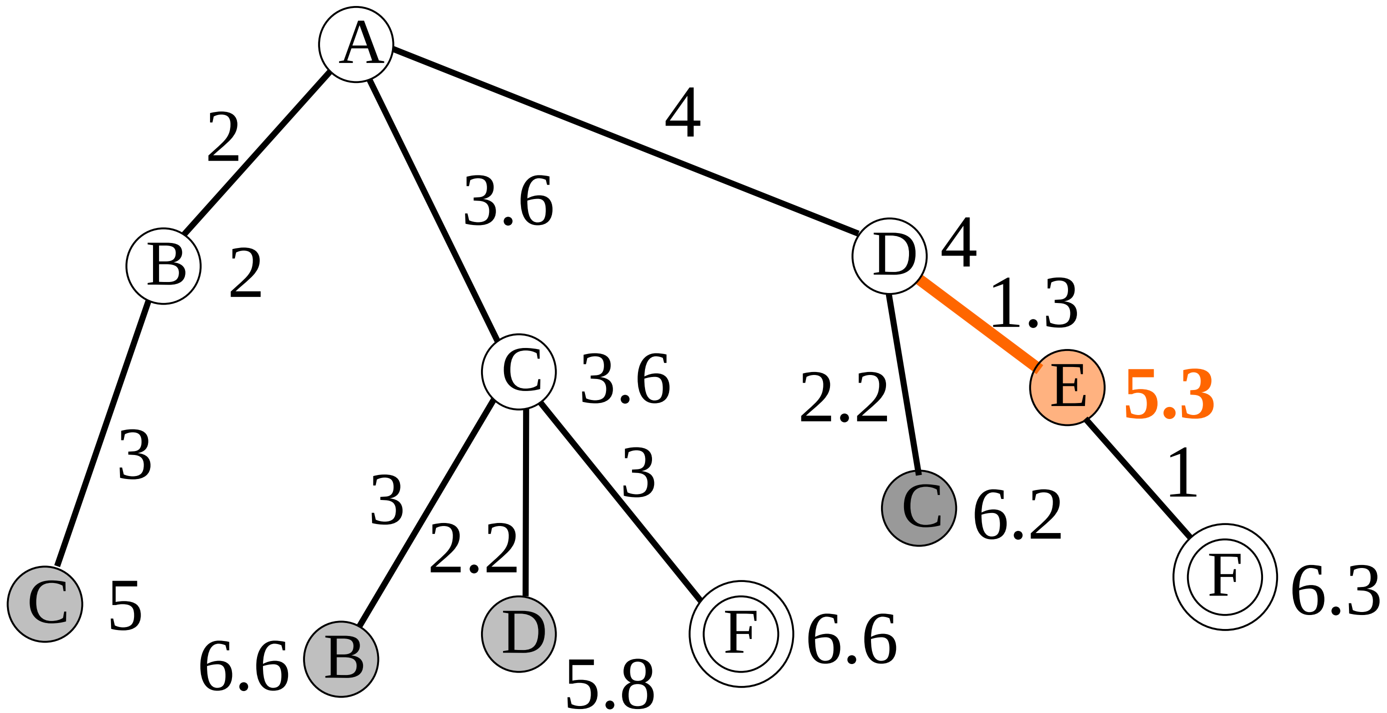
⇓

出発点から近い順に、各点の最短経路が求まる

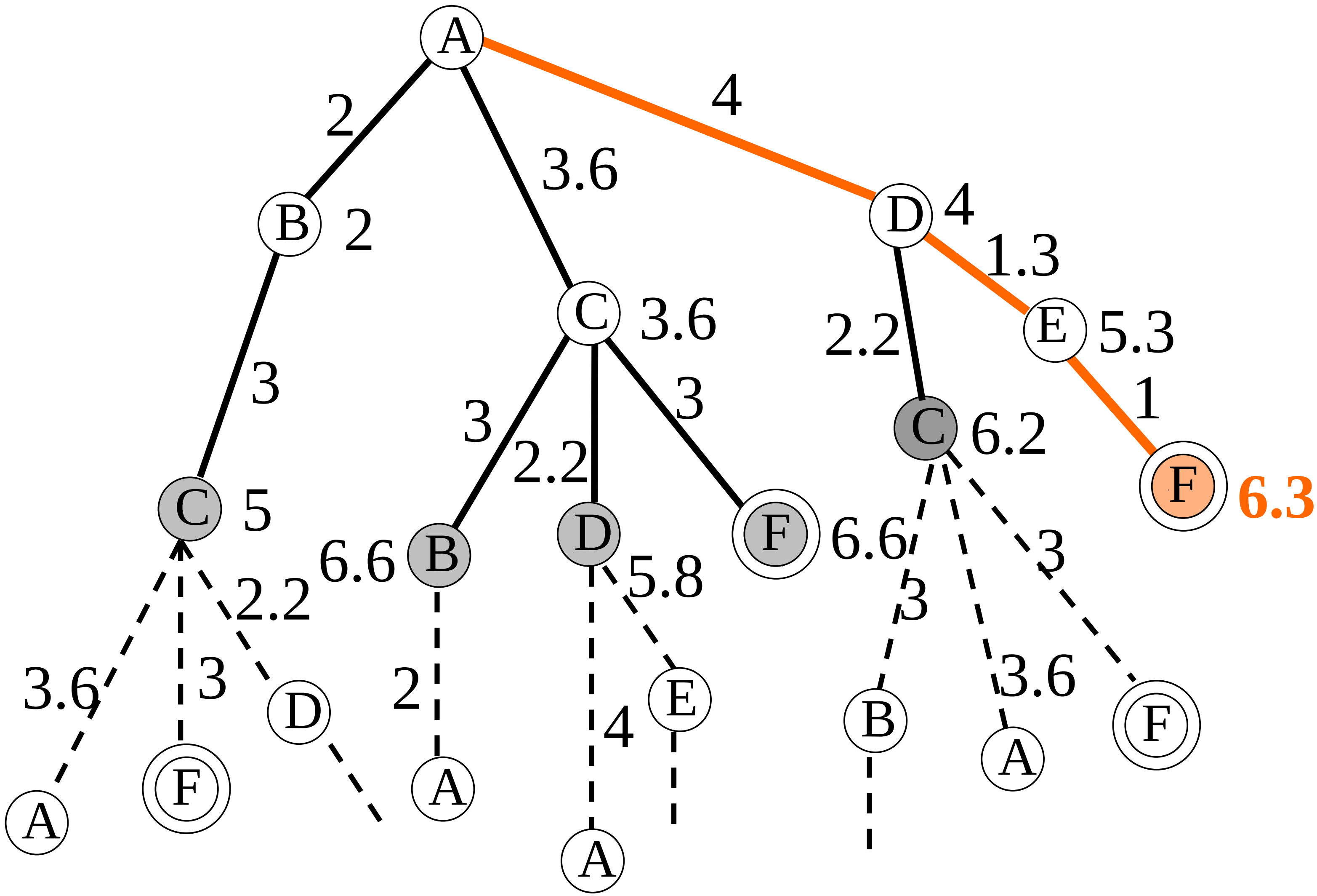
⇒ ダイクストラ法

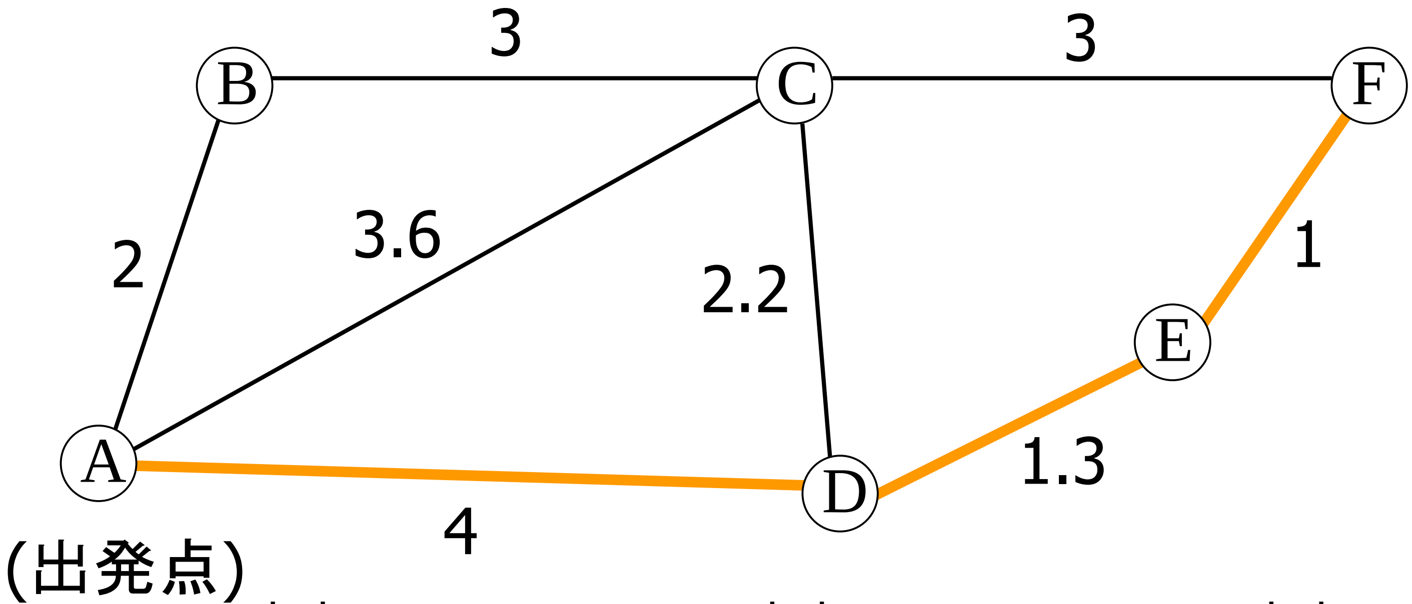

A
2
4
3.6
B
2
D
4
C
3.6
3
3
3
2.2
C
5
6.6
B
D
F
6.6
5.8

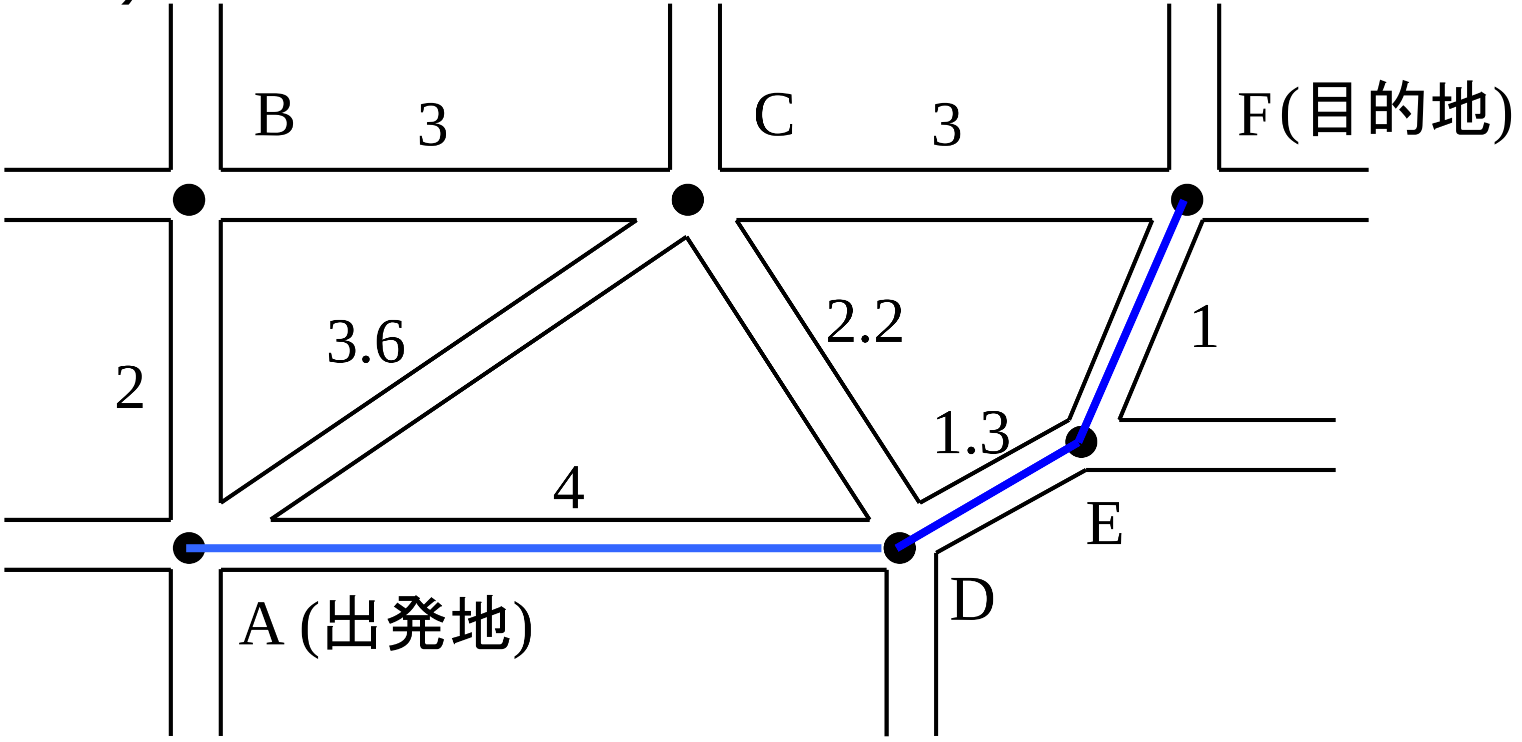

A
2
4
3.6
B
2
D
4
1.3
C
3.6
2.2
E
5.3
3
3
3
C
6.2
2.2
C
5
6.6
B
D
F
6.6
5.8



A
2
3.6
4
B
2
D
4
1.3
C
3.6
2.2
E
5.3
3
3
3
1
C
6.2
2.2
F
6.3
C
5
D
F
6.6
6.6
B
5.8


A
2
3.6
4
B
2
D
4
1.3
C
3.6
2.2
E
5.3
3
3
1
3
C
6.2
F
6.3
C
5
2.2
B
6.6
D
F
6.6
3
6.6
5.8
2.2
3
3
3.6
3
D
2
E
3.6
4
B
A
F
A
F
A
A


3
3
B
C
F
2
3.6
2.2
1
E
1.3
A
D
4
(出発点)


B
3
C
3
F(目的地)
3.6
2.2
1
2
1.3
4
E
D
A (出発地)

もっと効率化できないか？

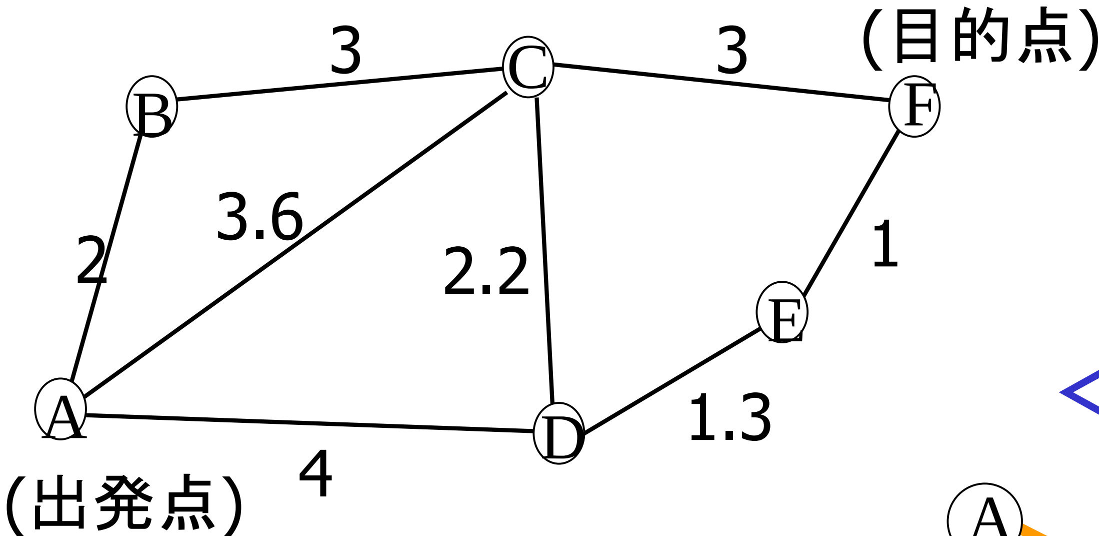
実際には目的地から遠ざかる方には行かない場合が多い

→ 目的地に近づく方向を優先する

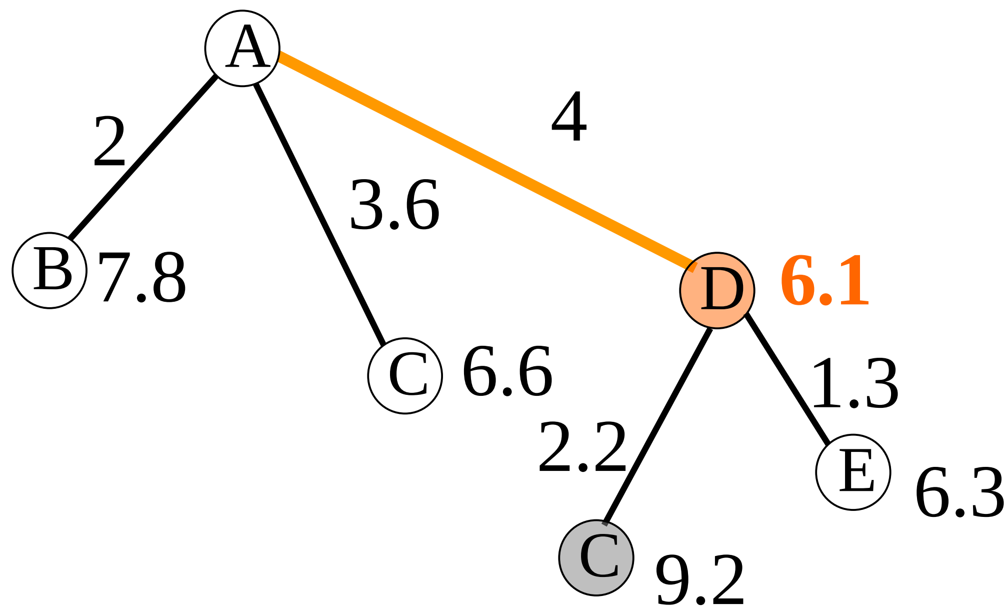
↓

目的地までの距離も考慮した最良優先探索

→ A*アルゴリズム

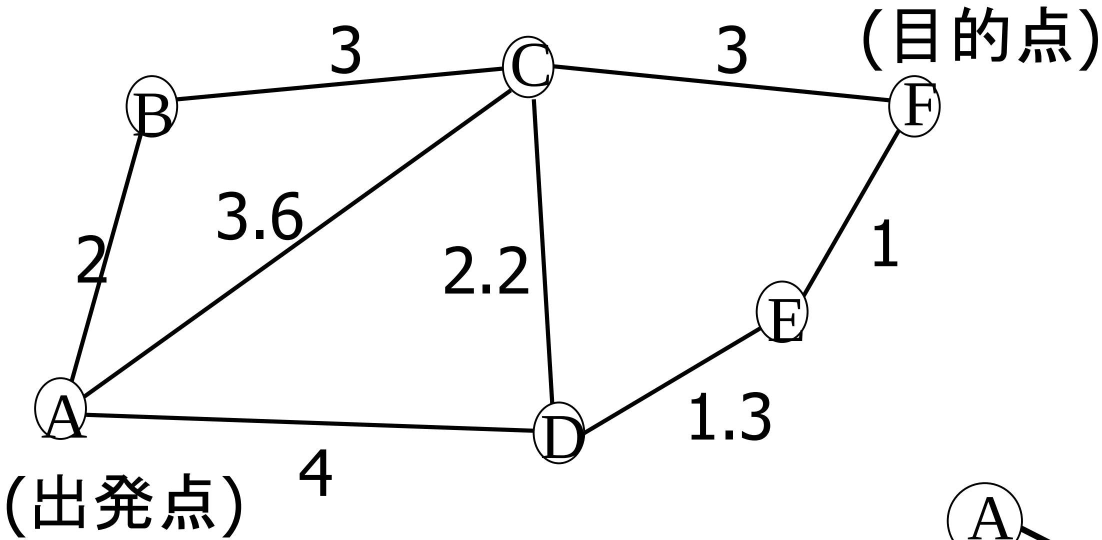


＜経路探索木＞



＜直線距離＞

BF間：5.8
CF間：3
DF間：2.1
EF間：1



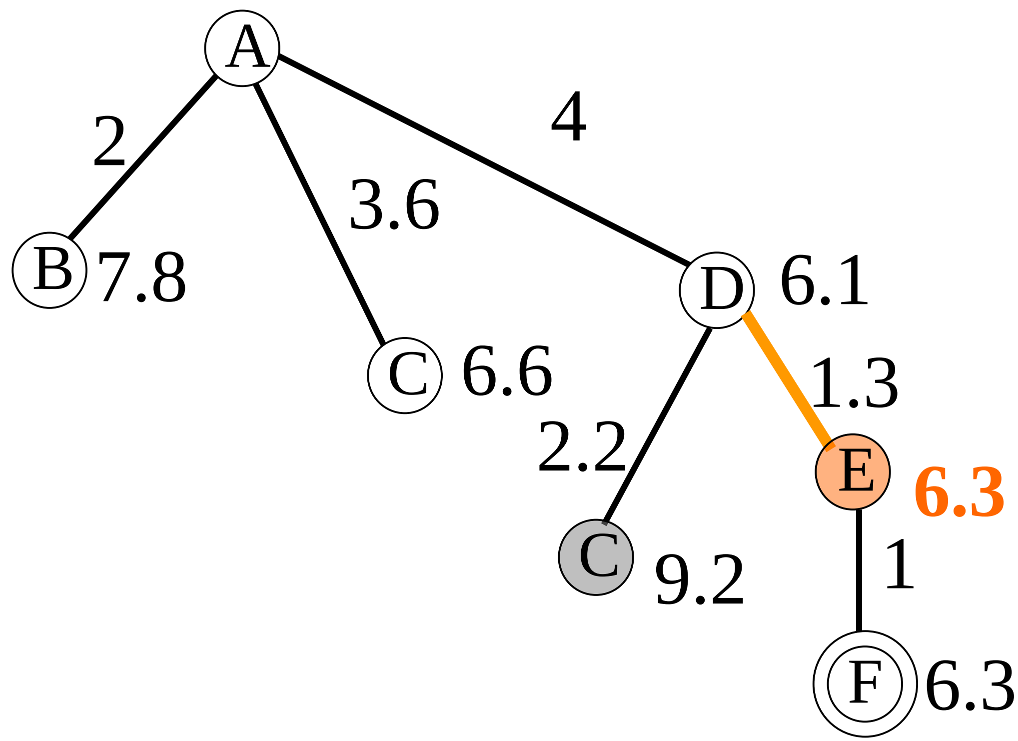
＜直線距離＞

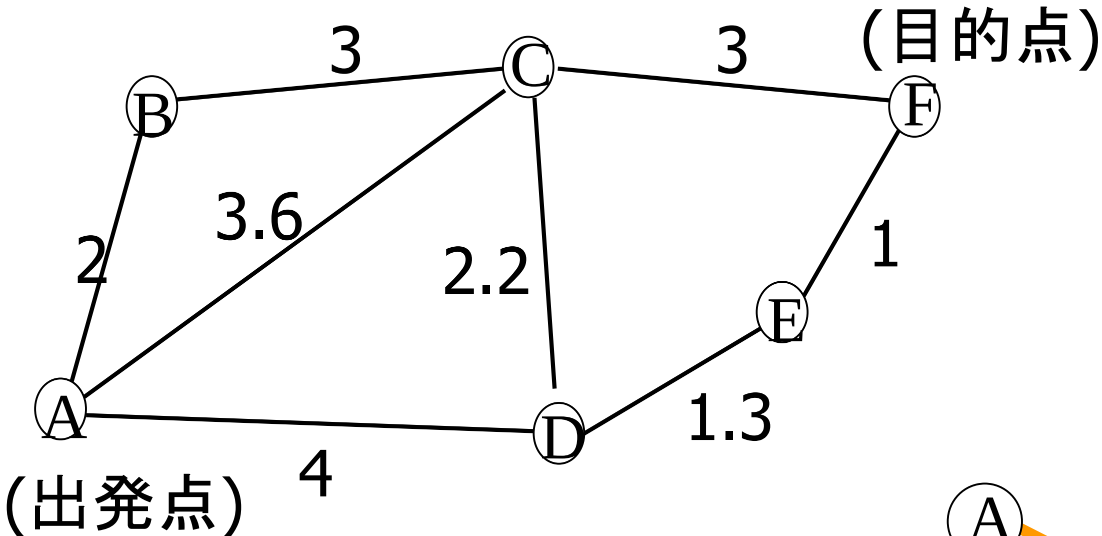
BF間：5.8

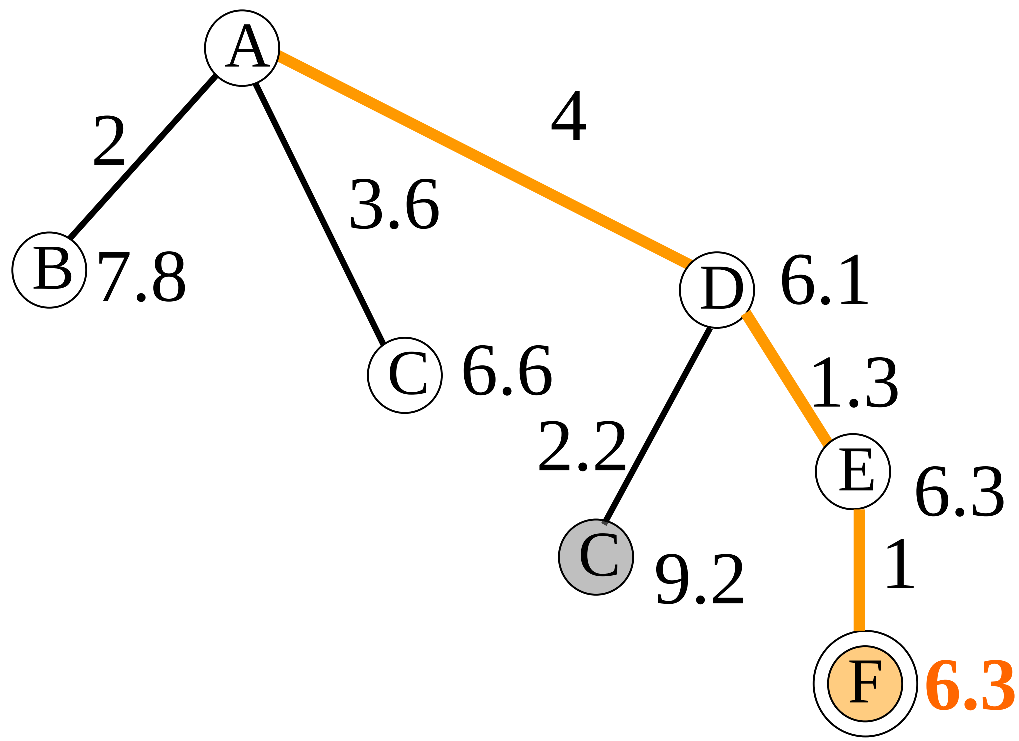
CF間：3

DF間：2.1

EF間：1







＜直線距離＞

BF間：5.8

CF間：3

DF間：2.1

EF間：1

＜人工知能＞

コンピュータの進歩

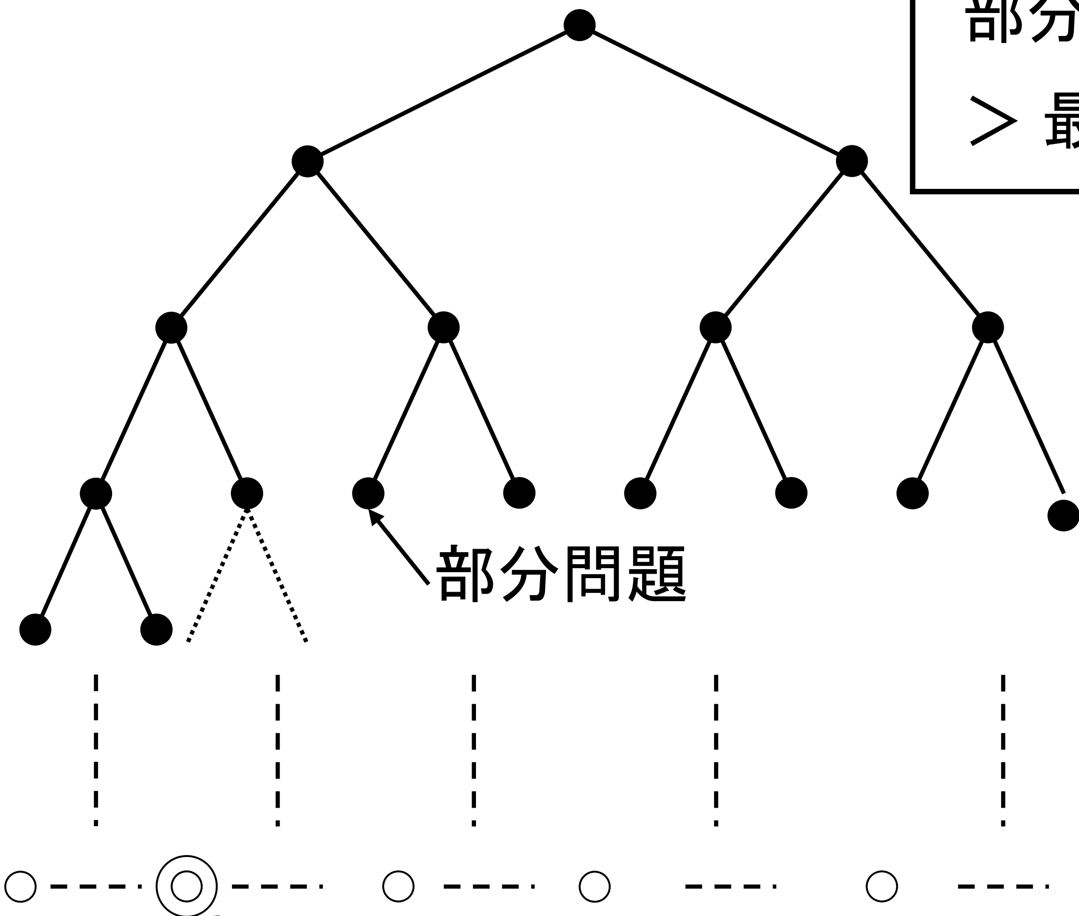
・チェスの世界チャンピオンに勝利

・全米クイズ王に勝利

・将棋でプロ棋士（高段者）に勝利

・車の自動運転

列挙木



＜分枝限定法＞

部分問題の解の下界値
＞ 最適解の上界値（暫定解の値）

↓

その部分問題から最適解は得られないので探索打ち切り（限定操作）

A*アルゴリズム

最短路問題の解法

最良優先探索による分枝限定法

部分問題の評価値

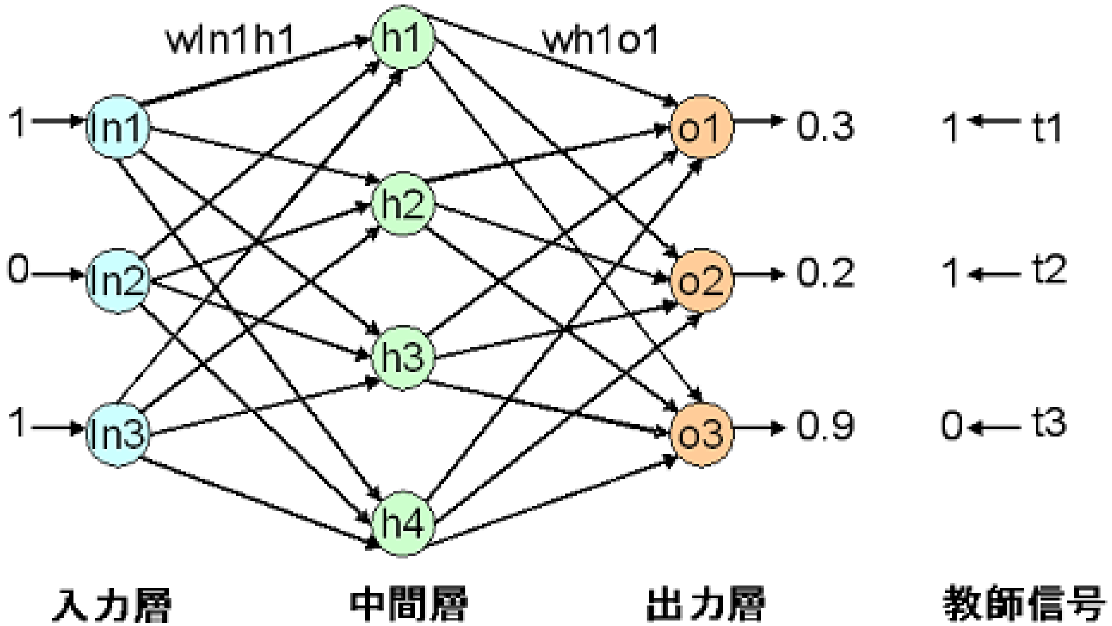
ダイクストラ法:

出発地からの最短距離

A*アルゴリズム:

出発地からの最短距離＋目的地までの直線距離

ニューラルネットワークによる学習



$$E = \frac{1}{2}\sum_{i=1}^{3}(oi - ti)^2 = \frac{1}{2}((o1 - t1)^2 + (o2 - t2)^2 + (o3 - t3)^2)$$

誤差逆伝播法　最急降下法の適用

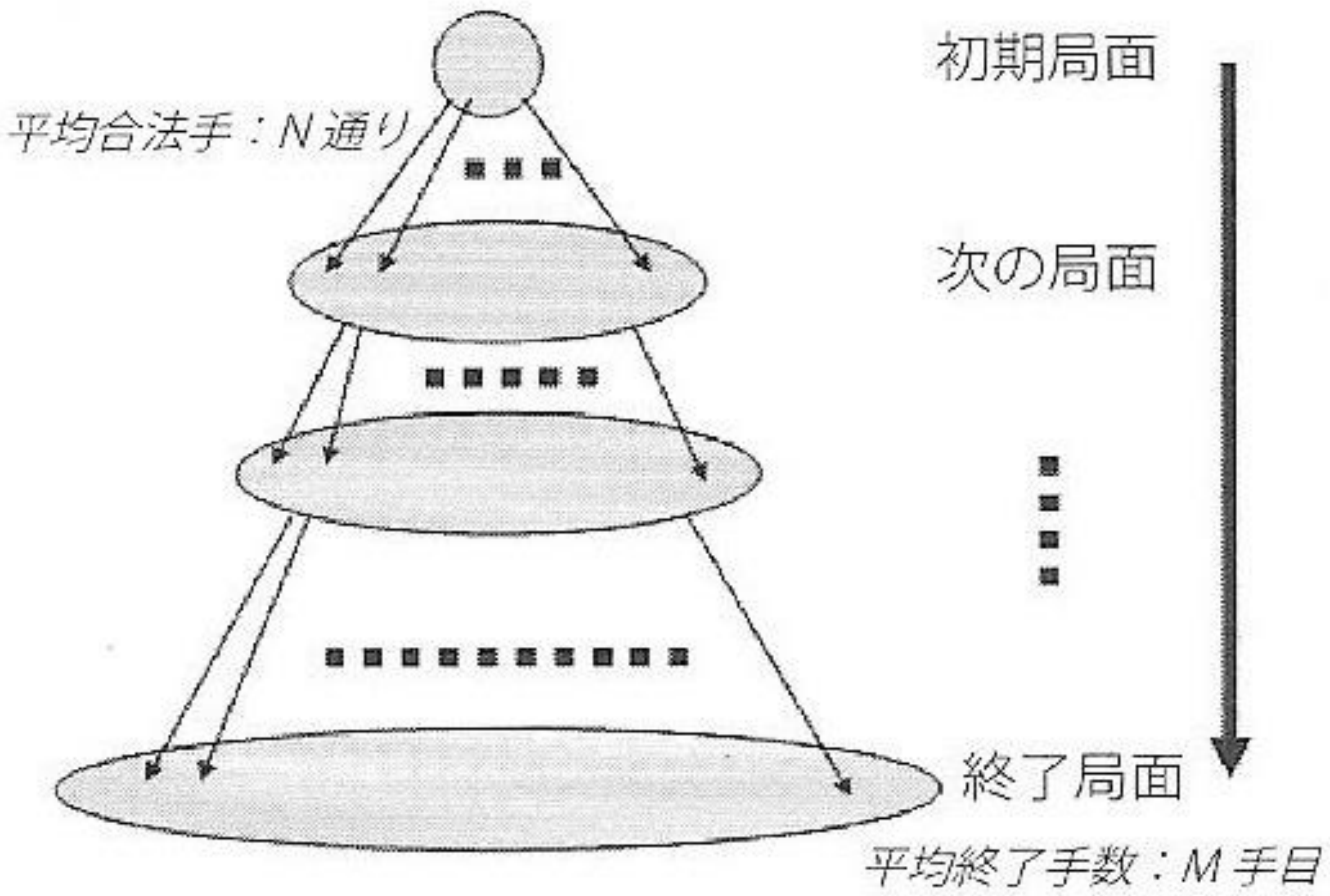


図３　ゲーム木探索によるゲームの複雑さ

・チェッカー：10の30乗

・オセロ：10の60乗

・チェス：10の120乗

・将棋：10の220乗

・囲碁：10の360乗

・チェッカー、オセロ、チェスは20世紀中に人間のトップに勝利

コンピュータ将棋の技術

ゲーム木探索:アルゴリズムの基本

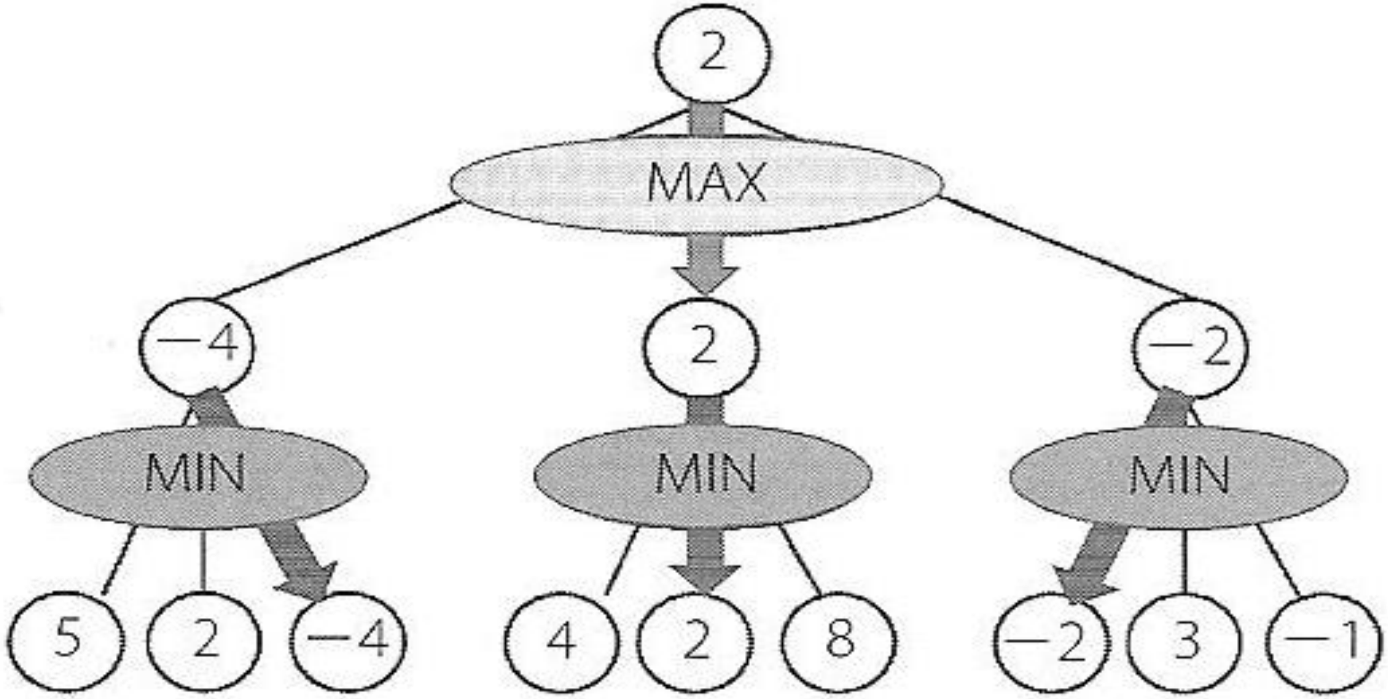


図４　ゲーム木のミニマックス探索

局面評価値:「静的評価関数」と呼ばれる形勢を数値化したもの

ミニマックス探索:「相手は自分にとって最も嫌な手を選択するはず」という基本的な考え方

「探索の効率化」と「評価関数の精緻化」が両輪

Bonanzaの登場と効果

評価関数の機械学習を導入

それまでのコンピュータ将棋では，評価関数の設計はプログラマの職人的な技によって点数付けがされていて，パラメータを少し変えては自己対戦を繰り返し，効果があるかどうかを評価するということを行っていた。従って，プログラマにはある程度の将棋の知識が不可欠であり，強豪プログラマは，将棋のアマチュア有段者という人が多かった。

しかし，Bonanzaでは，プロ棋士やトップアマチュアの実践棋譜を教師データにして，盤上すべての駒の位置関係のパラメータを設定し，そのパラメータを学習によって調整し，熟達者ならでは駒の配置（持ち駒も含む）を学習させることを試みた。具体的には，各盤面の判断基準となる評価関数が熟達者のそれと同じようになるようにパラメータを求めていく。さらに，保木氏は，これらの技術を包み隠さずソースコードごと公開した[6]。

回帰分析はどんなものなのか

回帰分析とは、大ざっぱに言えば、複数の種類のデータを用いてある値を予測する計算式を得ることです。たとえば、「理科のテストの得点（x_1）と社会のテストの得点（x_2）を使って、数学のテストの得点（y）を予想する計算式を作る」のは、回帰分析です。

数学	理科	社会
51	50	60
59	60	55
57	65	65
63	70	63
66	75	70
70	80	72

計算式（予測式）を、

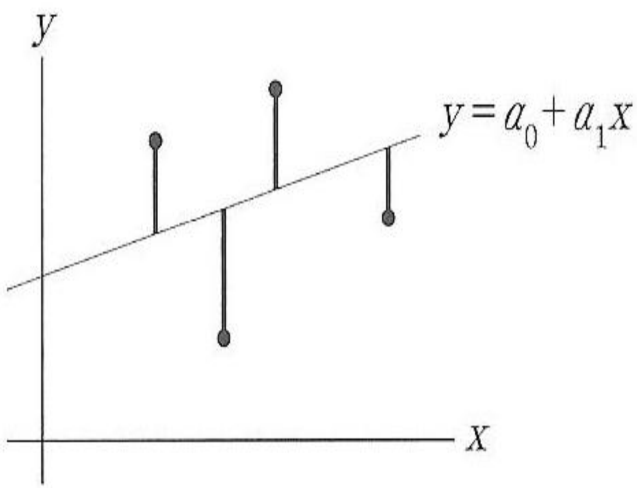
$$y = a_0 + a_1x_1 + a_2x_2$$

のように、1次式とするものを線形回帰分析といいます。

$$y = 28.0 + 0.720x_1 - 0.233x_2$$

この結果によれば、たとえば、理科89点、社会82点の人は、数学が73点だろうと予想できます。

わかりやすく視覚的に理解できるように、単純な例を挙げておきます。4つのデータ（下図の若干大きく描いてある4点）を使って$y=a_0+a_1x$という回帰直線を求める例です。この場合、「回帰直線を最小2乗法で求める」とは、下図の太線の長さの2乗の総和が最小になるような直線（の方程式）を求めることを意味します。



なお、ずっと後で詳しく説明しますが、基本的にはチェスやチェッカーでは、形勢判断に評価関数を用いていて、その関数は、本項冒頭にあるような線形式です。名人の棋譜を集めて回帰分析を行なうことで、コンピューターはより正確な評価関数を獲得します。

ちなみに、DEEP BLUEや初期のCHINOOKでは、線形回帰分析が使われました。CHINOOKは世界最強のチェッカー・プログラムです。これらについても後で詳しく説明します。

$y = a_0 + a_1x_1 + a_2x_2$

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_{11} & X_{21} \\ 1 & X_{12} & X_{22} \\ & \vdots & \vdots \\ 1 & X_{1n} & X_{2n} \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad \boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

$\boxed{\text{最小2乗法}}$

$f(\boldsymbol{a}) = {}^{t}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{a})(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{a})$　を最小化

$\nabla f(\boldsymbol{a}^*) = \boldsymbol{0}$　となる $\boldsymbol{a}^*$ を求める

$$f(\boldsymbol{a}) = {}^{t}(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{a})(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{a})$$

$$= {}^{t}\boldsymbol{Y}\boldsymbol{Y} - 2\,{}^{t}\boldsymbol{Y}\boldsymbol{X}\boldsymbol{a} + {}^{t}\boldsymbol{a}\,{}^{t}\boldsymbol{X}\boldsymbol{X}\boldsymbol{a}$$

$$\nabla f(\boldsymbol{a}) = -2\,{}^{t}\boldsymbol{X}\boldsymbol{Y} + 2\,{}^{t}\boldsymbol{X}\boldsymbol{X}\boldsymbol{a} = \boldsymbol{0}$$

$$ {}^{t}\boldsymbol{X}\boldsymbol{X}\boldsymbol{a} = {}^{t}\boldsymbol{X}\boldsymbol{Y}$$

$$\boxed{\boldsymbol{a} = ({}^{t}\boldsymbol{X}\boldsymbol{X})^{-1}\,{}^{t}\boldsymbol{X}\boldsymbol{Y}}$$