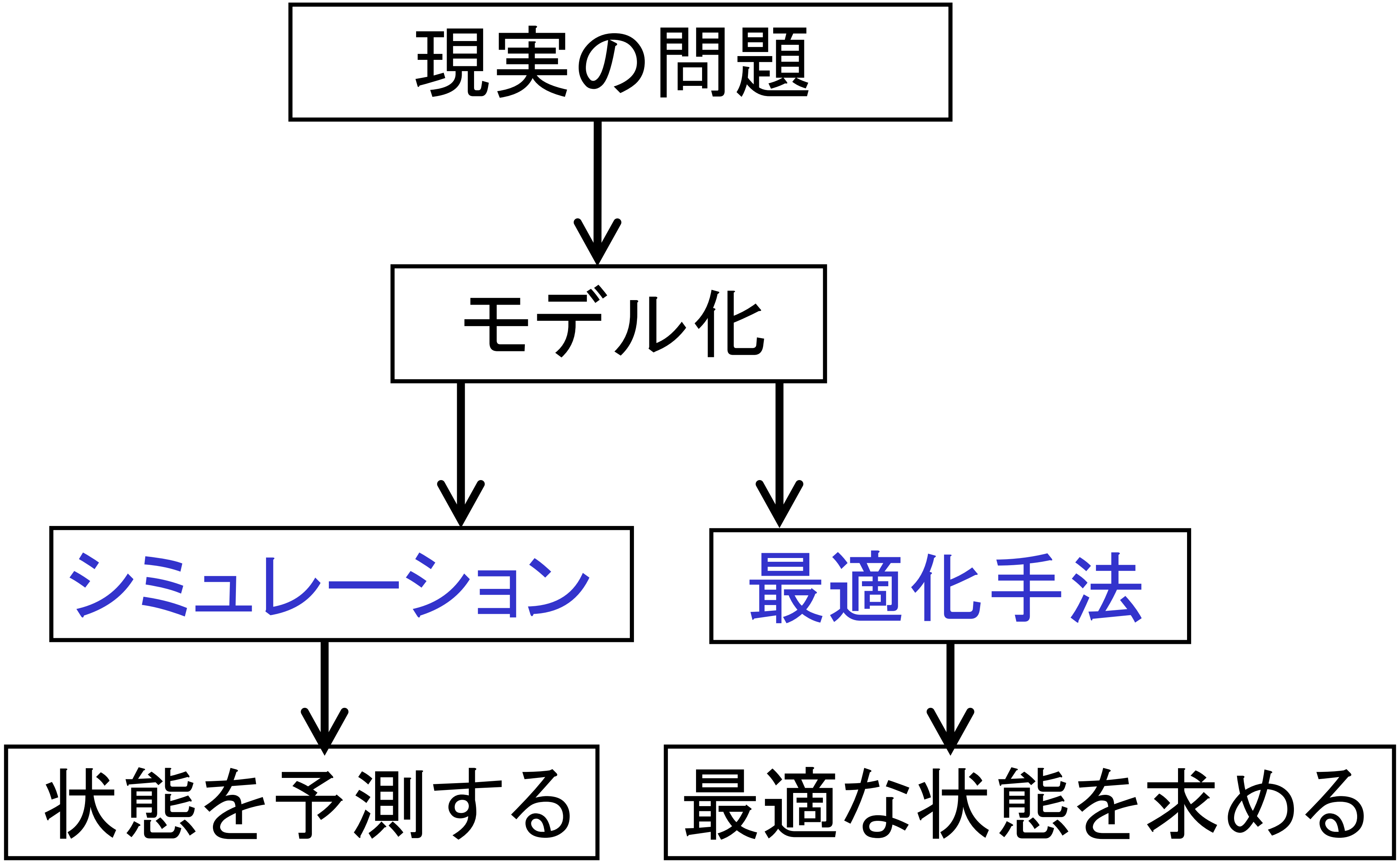
＜モデリングとシミュレーション＞



＜シミュレーション＞

・モンテカルロ・シミュレーション
  ・確率的要因・・・乱数発生
・連続型シミュレーション
  ・システムダイナミックス・・・ストックとフロー
  ・差分方程式、微分方程式
・離散型シミュレーション
  ・待ち行列
  ・ネットワークモデル（PERT型）

② 乱数とシミュレーション

1―― 一様乱数

　確率的モデルの動作を調べるコンピュータシミュレーションでは，**乱数**①を使用する。乱数のなかで基本的なものは，どの数値も等しい確率で出てくる**一様乱数**（いちようらんすう）②である。

■表4　乱数表（JIS-9031）の一部

行										
1	67	11	09	48	96	29	94	59	84	41
2	67	41	90	15	23	62	54	49	02	06
3	78	26	74	41	76	43	35	32	07	59
4	32	19	10	89	41	50	09	06	16	28
5	45	72	14	75	08	16	48	99	17	64
6	74	93	17	80	38	45	17	17	73	11
7	54	32	82	40	74	47	94	68	61	71
8	34	18	43	76	96	49	68	55	22	20
9	04	70	61	78	89	70	52	36	26	04
10	38	69	83	65	75	38	85	58	51	23

2—— モンテカルロ法

ここでは，乱数を用いたシミュレーションである**モンテカルロ法**の代表的な例について述べる。

例題5 円周率を求める

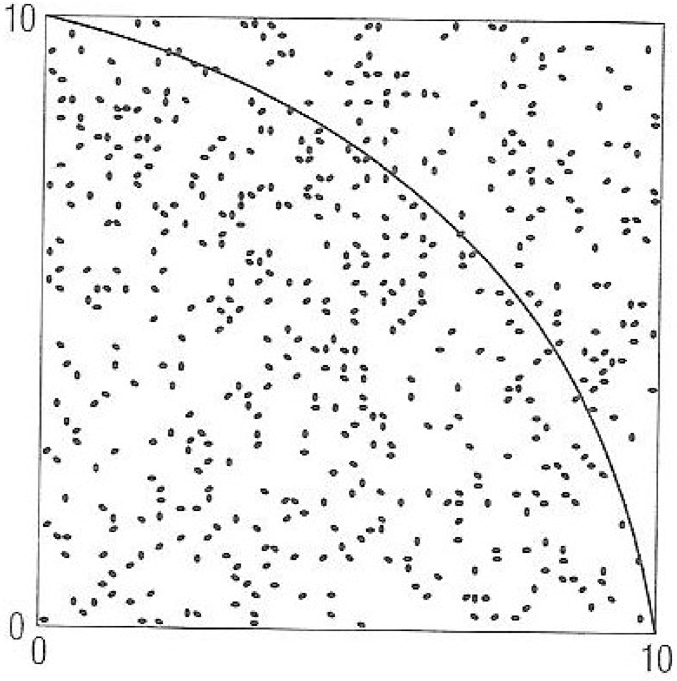
一様乱数を100個発生させ，円の面積から円周率を求めてみよう。また，1000個の場合，どのようになるか調べてみよう。

一辺10cmの正方形の方眼紙を用意して，そこに黒ゴマをできるだけランダムにまいたとしよう。このとき，正方形内にある黒ゴマの数をN，また，正方形内に接する$\frac{1}{4}$円のなかにある黒ゴマの数をMとする。

正方形の面積は，100（cm^2）

$\frac{1}{4}$円の面積は，$\frac{(\pi\times10\times10)}{4}$（$cm^2$）

である。もし，黒ゴマが一様にまかれているならば，N，Mの数は，それぞれの面積に比例するので，$\frac{M}{N}$は$\frac{\pi}{4}$になるはずである。これより円周率の値を求めることができる。



1 システムダイナミックスとは

図１に示すように，シミュレーションの型は，時間に対して静的なものと動的なものに分類される。さらに，動的なものは連続的なものと離散的なものに分類される。ここで，連続的とは，時間に関して連続的に変化することをあらわし，離散的とは，１年や１か月ごとなどの離散時間ごとにとらえることができることをあらわす。確定的とは，条件が設定されると一意的に結果が導き出されるものをいう。**システムダイナミックス**は，図１の網が掛かっている部分の性質をもつ。

シミュレーション
- 静的
- 動的
  - 連続的 ―― 確定的
  - 離散的 ―― 統計的

■図1　システムダイナミックスの性質

2 ストックとフローによるモデルの構成

身のまわりの自然現象や社会現象などのシステムの時間的変化は，ストックとフローの組み合わせで表現することができる。

■表1　自然現象の例

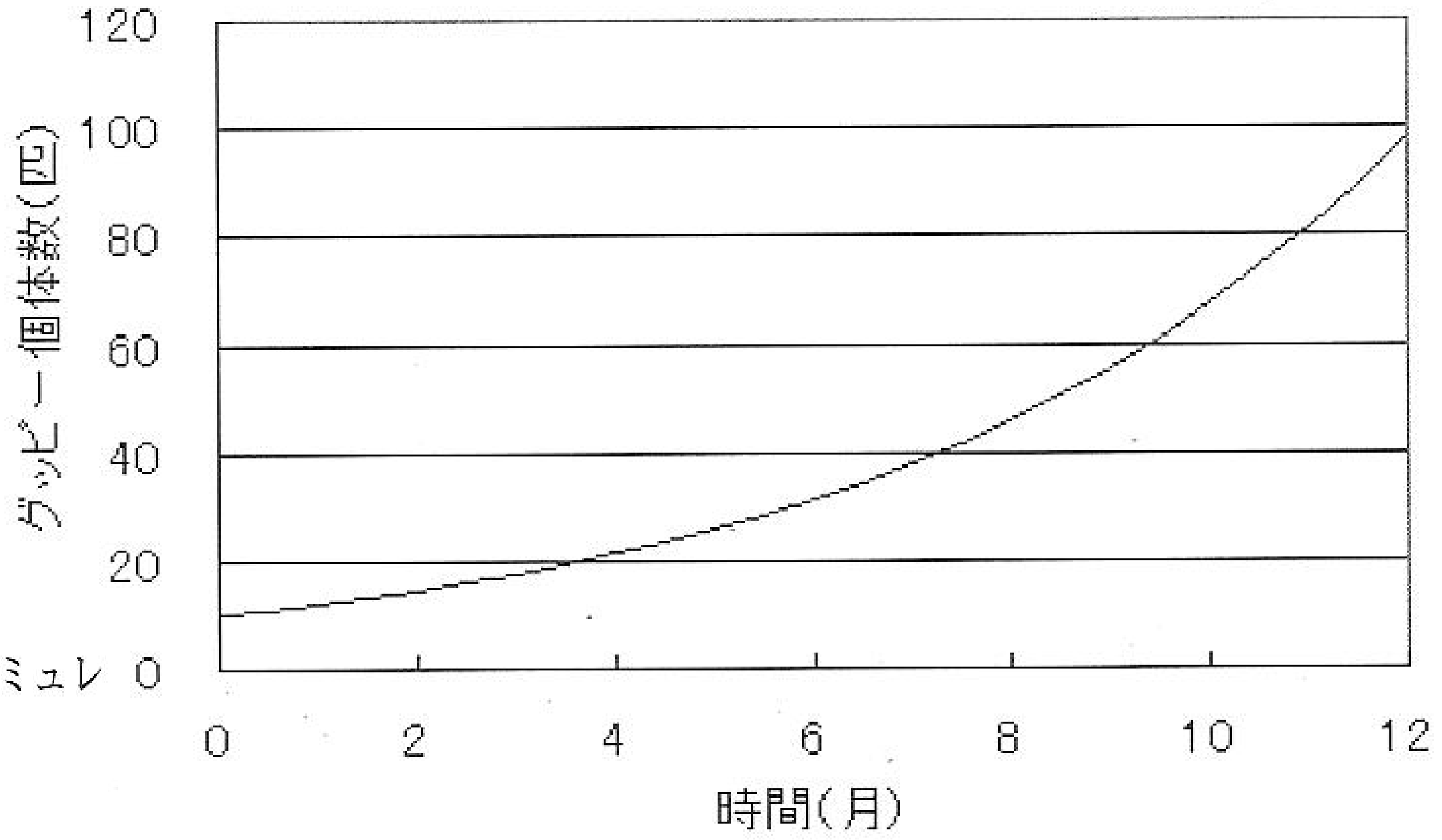
現象	インフロー①	ストック	アウトフロー①
生物の個体数	出生	個体数	死亡
ダムの貯水量	流入	貯水量	流出
水温の変化	加熱	水温	放熱

■表2　社会現象の例

現象	インフロー	ストック	アウトフロー
駐車場の管理	入庫	駐車台数	出庫
預金額の変化	入金	残高	出金
リサイクル	回収	保管	再生

	1	2	3	4	5
1	初期値	グッピー個体数	10		
2	設定値	増加率	0.2		
3	時間設定	終了時間	12	時間間隔	0.5
4					
5	経過時間	グッピー個体数			
6	0	10			
7	0.5	11			
8	1	12			
9	1.5	13			
10	2	15			
11	2.5	16			

■図6　グッピーの繁殖のシミュレーション結果（一部）



■図7　グッピーの繁殖のグラフ

問2 例題２で，グッピーの増加率が25%になった場合のシミュレーションを行い，結果をグラフで表示しなさい。

１．（１）グッピー個体数のストック･フローモデル

時間 t か月後のグッピー個体数の計算式を考える。

増加速度(t)＝グッピー個体数(t)×増加率

グッピー個体数(t＋Δt)＝グッピー個体数(t)＋増加速度(t)×Δt

（２）

初期値	グッピー個体数	20		
設定値	増加率	0.25		
時間設定	終了時間	3	時間間隔	1
経過時間	グッピー個体数			
0	20			
1	25			
2	31	(31.25)		
3	39	(39.06)		

例題1　ダム洪水対策問題

あるダムで，洪水期（6，7月とする）のダム管理における洪水対策①として放流量を決めたい。シミュレーションによる解決案を提案するためにモデルをつくってみよう。なお，ダムの容量は，満水で350万m^3の大きさとする（この値を超えるとダムから水があふれ大変危険な状態になる）。1日あたりのダムへの水の流入量は不規則で，洪水期については表のように与えられているものとする。

1日あたりの流入量（万m^3）	確率	累積確率
0.5	0.1	0.10
1	0.7	0.80
5	0.1	0.90
10	0.07	0.97
20	0.02	0.99
50	0.01	1.00

①大雨による洪水をなくすために，ダムで一時的に水をためて安全に放流すること。

▶ 3．モデルを数式で表現する

1日あたりの「流入量」の式は，乱数の値を累積確率の範囲に当てはめる方法を用いる。(→p.48)流入量の値を取り出す関数は，図3の数式表①のように与えられる。

1日あたりの「流出量」は，ダム貯水量＜310のときは「日常放流量」の値，それ以外のときは「最大放流量」の値を出力する関数として数式表②のようにする。

変化後のダム貯水量は数式表③のようになる。

```
①流入量＝ if RAND<=0.1 then 0.5
          else if RAND<=0.8  then 1
          else if RAND<=0.9  then 5
          else if RAND<=0.97 then 10
          else if RAND<=0.99 then 20
          else 50
②流出量＝ if（ダム貯水量<310）then 日常放流量
          else 最大放流量
③変化後のダム貯水量＝現在のダム貯水量
          ＋（流入量－流出量）×時間間隔
```

■図3　数式表

初期値	貯水量	300万m^3		
設定値	日常放流量	1万m^3	最大放流量	30万m^3
	310万未満		310万以上	
時間設定	終了時間	5	時間間隔	1
経過時間	一様乱数	貯水量	流入量	流出量
1	0.77	300	1	1
2	0.43	300	1	1
3	0.09	299.5	0.5	1
4	0.82	303.5	5	1
5	0.15	303.5	1	1

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＋Δt)－f(t)＝Δyとおくと，

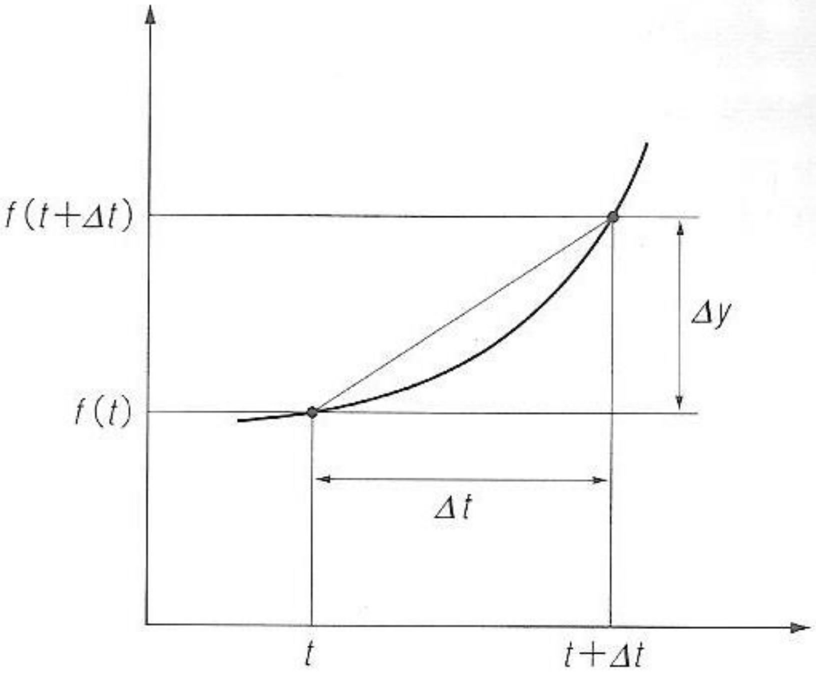
$$\frac{f(t+\Delta t)-f(t)}{\Delta t}=\frac{\Delta y}{\Delta t}=4t$$

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。



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。

次の条件で，このモデルのシミュレーションを行う。

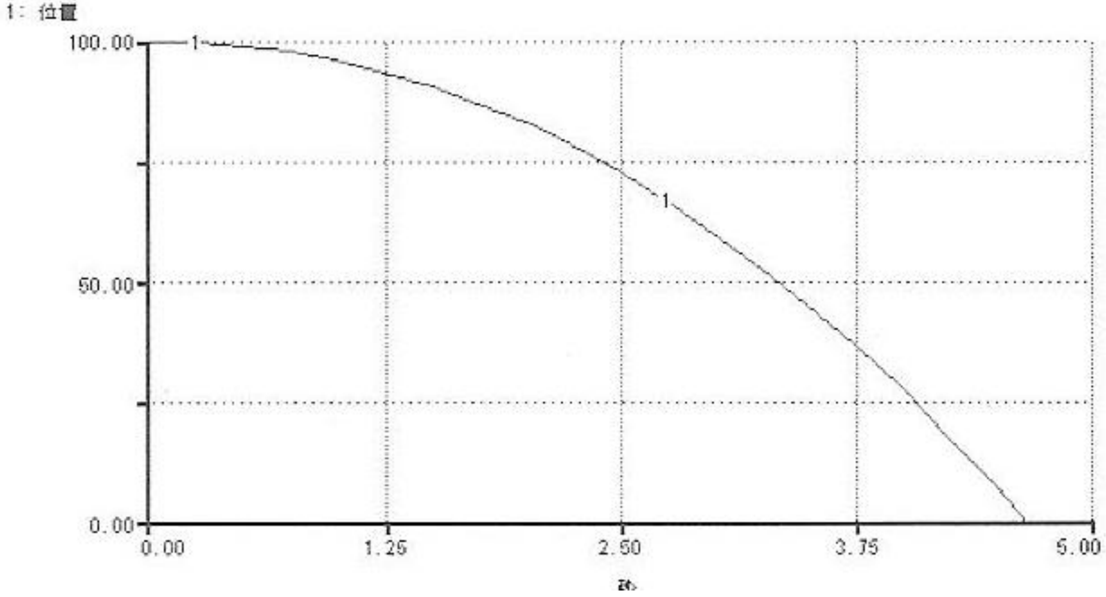
時間の単位：秒　　期間：開始0秒　終了5秒

時間間隔：0.25　　計算方法：オイラー法

加速度α：ー10（m/s²）
位置（高さ）y：1000（m）
速さv：　v ＝α× t

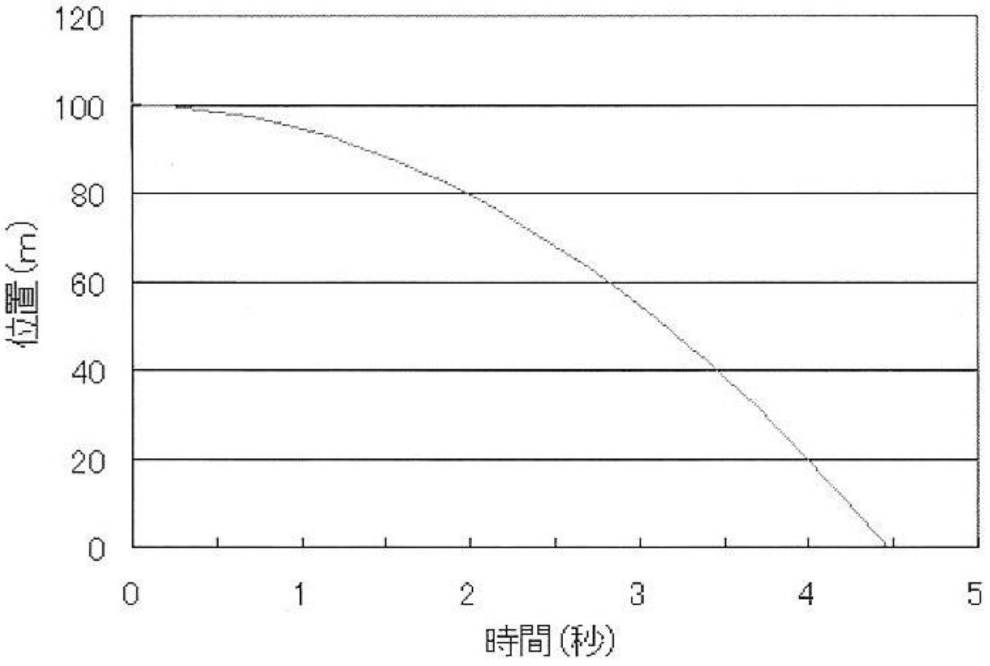
$v(t+\Delta t) = v(t) + \alpha \times \Delta t$

$Y(t+\Delta t) = Y(t) + v \times \Delta t$



	1	2	3	4	5
1	初期値	位置	100	運動量	0
2	設定値	質量	5	加速度	9.8
3	時間設定	終了時間	5	時間間隔	0.1
4					
5	時間 t	位置			
6	0	100.00			
7	0.1	99.90			
8	0.2	99.71			
9	0.3	99.41			
10	0.4	99.02			
11	0.5	98.53			
12	0.6	97.94			
13	0.7	97.26			
14	0.8	96.47			
15	0.9	95.59			
16	1	94.61			

■図7　自由落下のシミュレーション結果（一部）



■図8　自由落下のグラフ

初期値	位置	1000m		
設定値	加速度	$10m/s^2$		
時間設定	終了時間	5	時間間隔	1
経過時間	位置	速度	加速度	
0	1000	0	10	
1	1000	10	10	
2	990	20	10	
3	970	30	10	
4	940	40	10	
5	900	50	10	

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

運動方程式

F＝mg＋kx＝mα

　＝20x－700 ＝ 70α

重力加速度g：－10（m/s^2）

フック定数k：20(kg/s^2）

体重m： 70(kg）

加速度α

α＝2x/7－10

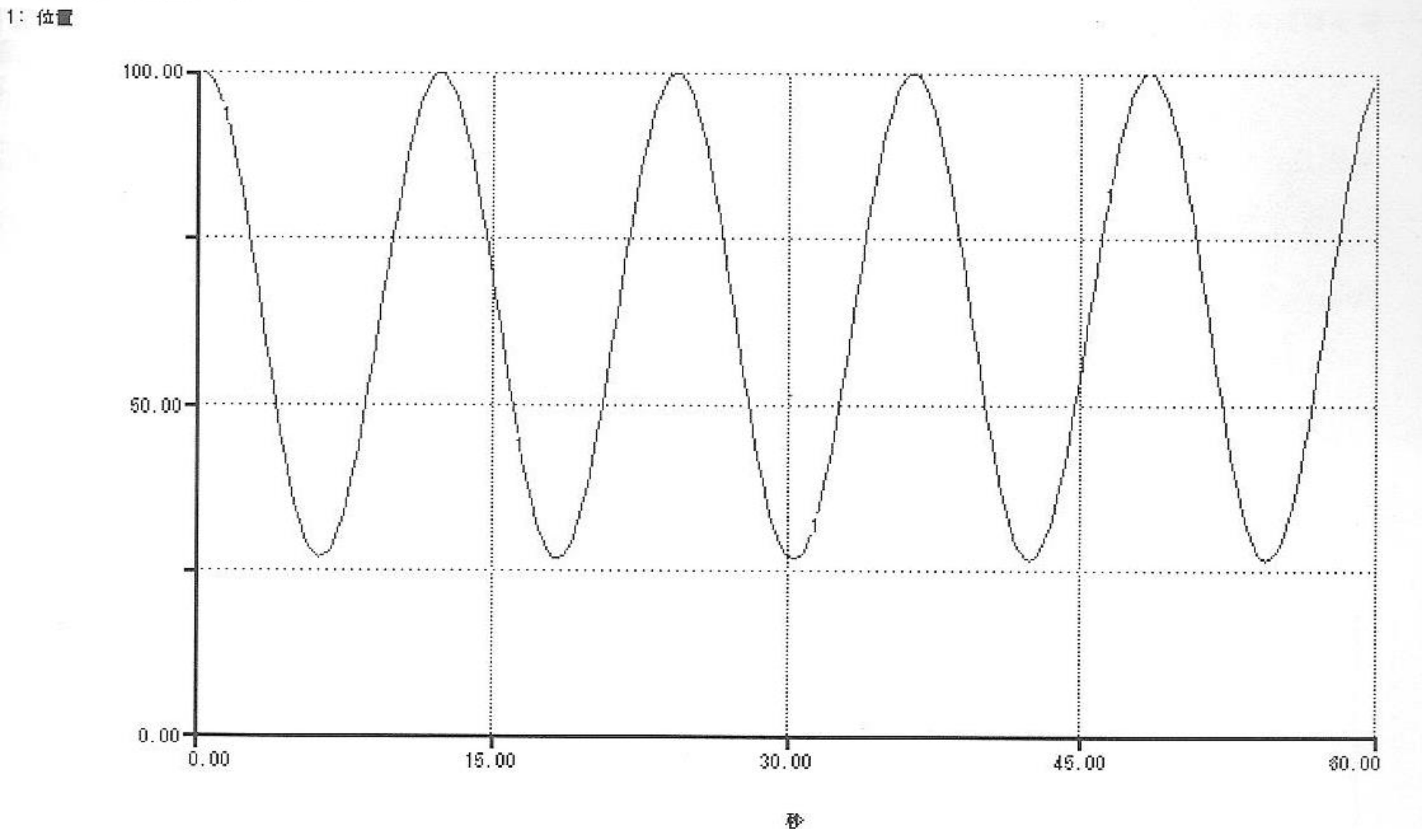
x＝300－y

位置（高さ）y：300（m)

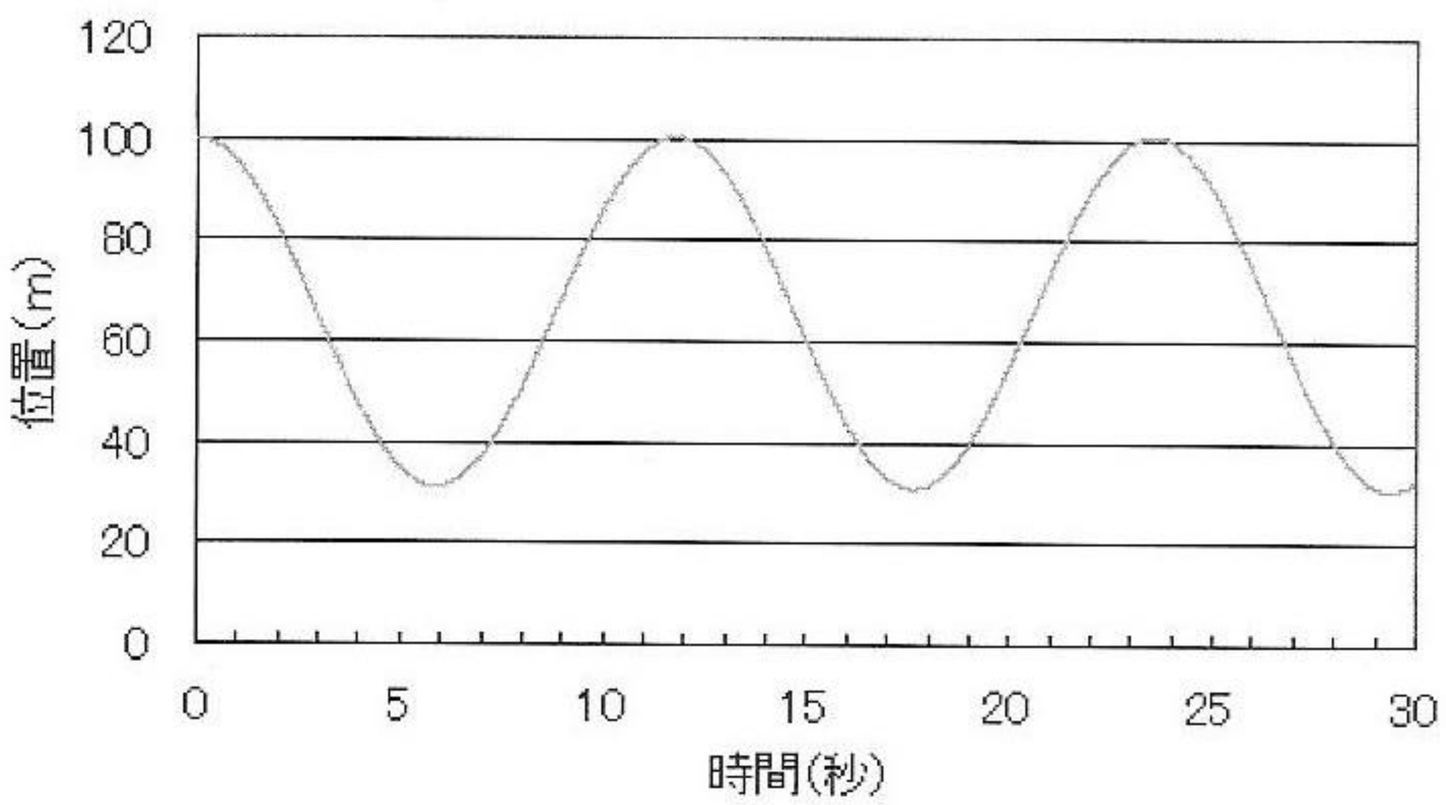
速さv：

$v(t+\Delta t) = v(t) + \alpha \times \Delta t$

$Y(t+\Delta t) = Y(t) + v \times \Delta t$



	1	2	3	4	5	6	7
1	初期値	運動量	0	位置	100		
2	設定	体重	70	重力加速度	9.8	フック定数	20
3	時間設定	終了時間	30	時間間隔	0.005		
4							
5	時間 t	位置					
6	0	100.00					
7	0.005	100.00					
8	0.01	100.00					
9	0.015	100.00					
10	0.02	100.00					
11	0.025	100.00					
12	0.03	100.00					
13	0.035	99.99					
14	0.04	99.99					
15	0.045	99.99					
16	0.05	99.99					
17	0.055	99.99					



■図12　バンジージャンプのシミュレーション結果（一部）

初期値	位置	300m		
設定値	体重	70kg	重力加速度	$10m/s^2$
	フック定数	$20kg/s^2$		
時間設定	終了時間	6	時間間隔	1
経過時間	位置	ひもの伸び	速度	加速度
0	300	0	0	10
1	300	0	10	10
2	290	10	20	7.1
3	270	30	27.1	1.4
4	242.9	57.1	28.5	−6.3
5	214.4	85.6	22.2	−14.5
6	192.2	107.8	7.7	−20.8
7	184.5	115.5	−13.1	−23
8	197.6	102.4	−36.1	−19.3
9	233.7	66.3	−55.4	−8.9
10	289.1	10.9	−64.3	6.9

2.2 車両の選定

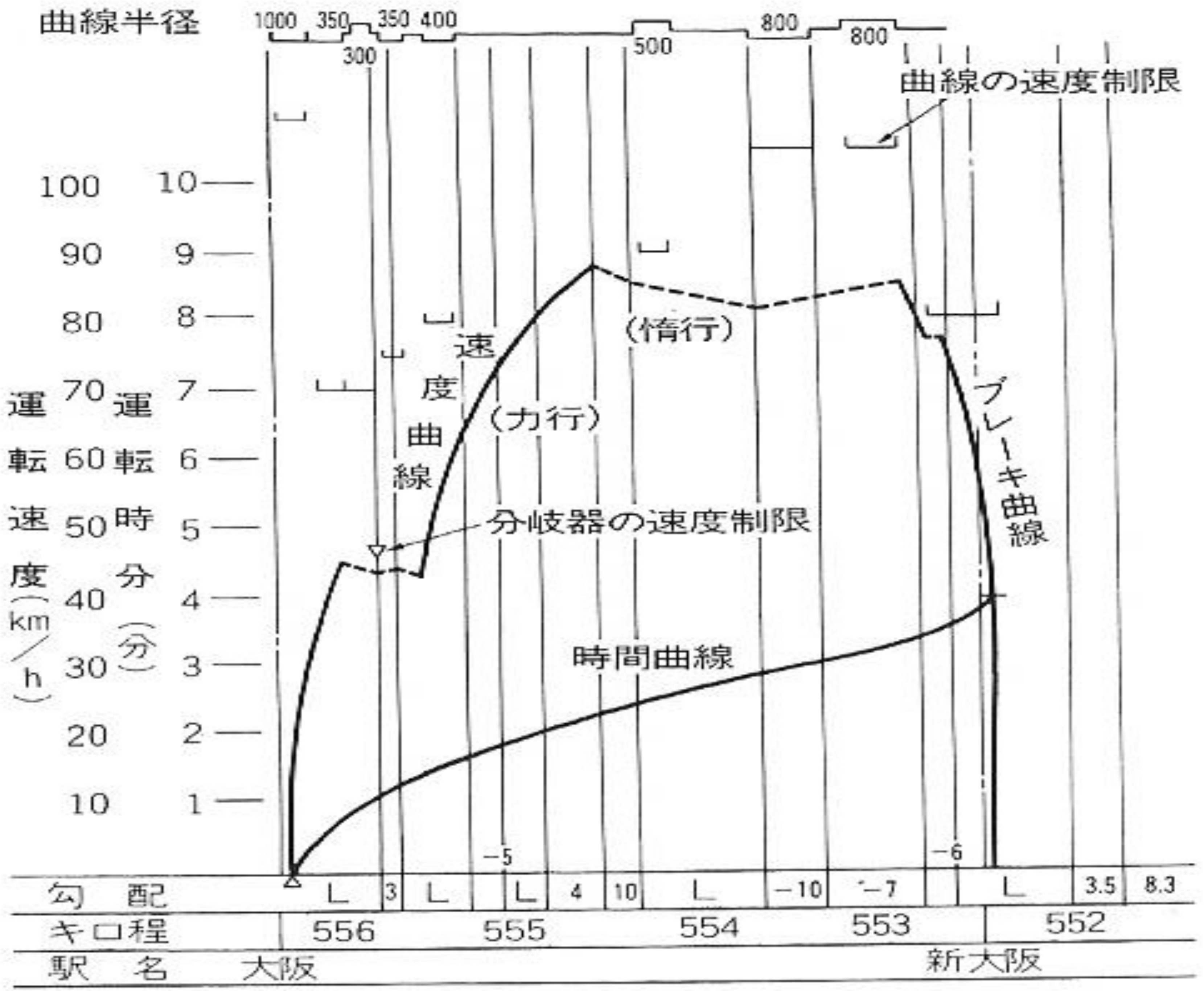
需要予測等により輸送量が想定されると、次に線区の運転区間、時間帯の必要輸送力、お客様サービスなどを考慮して、使用する車種・形式、編成両数を決定します。これもシステムのデータベース等を利用しますが、経験豊かな人での仕事が主となります。

2.3 運転時分の算定

列車ダイヤを作成するためには、すでに決定された車両によって各々の駅間を何分で運転できるか分らなければなりません。この時分のことを基準運転時分といいます。基準運転時分は、列車を運転する場合の駅間における計画上の最小所要時分のことです。

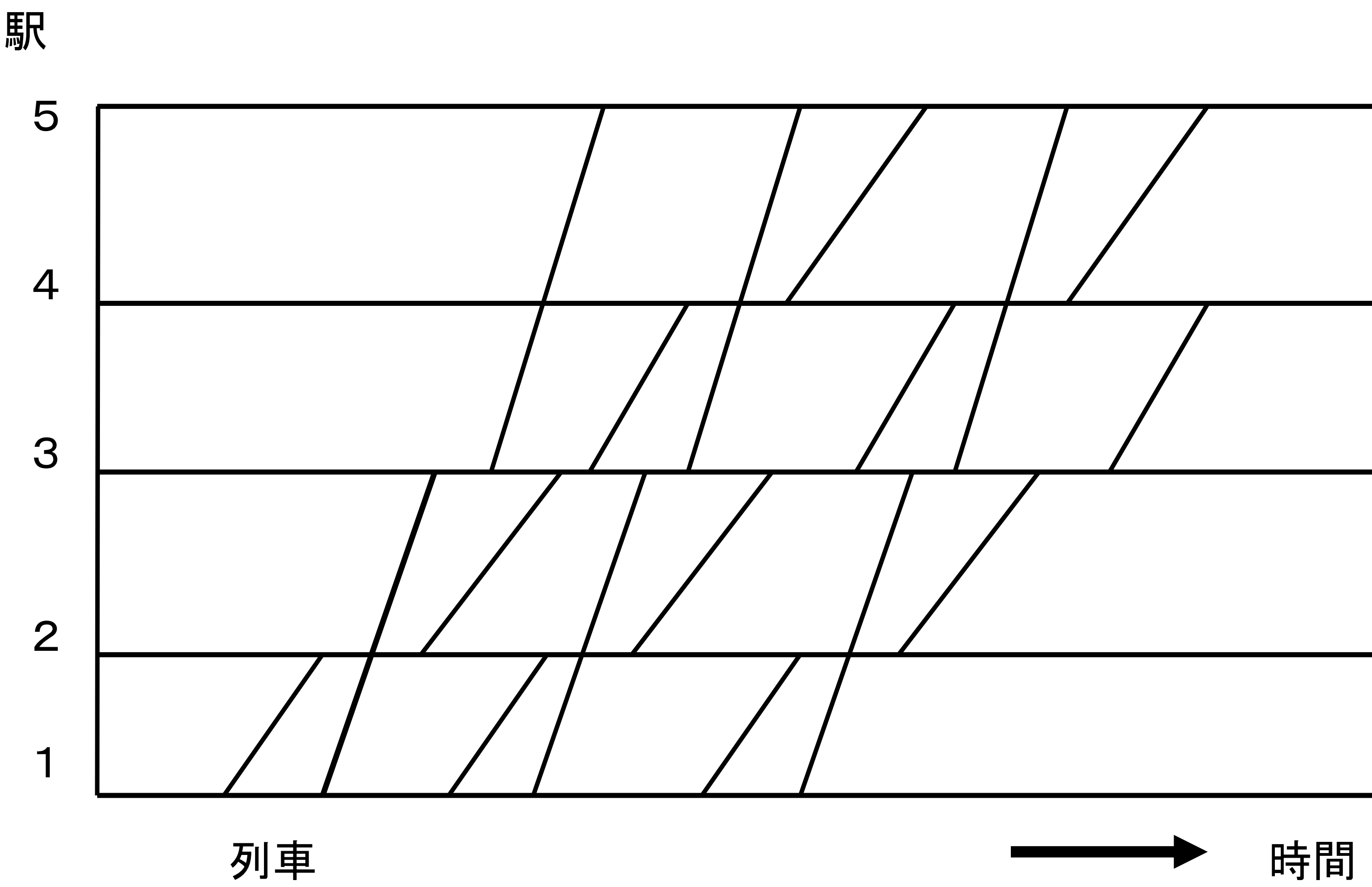
　この時分を算出するために必要となるのが、運転曲線図と呼ばれるグラフです。運転曲線図は、列車の走行状態をグラフ化したもので、列車の距離と速度の関係を表しています。一般的には、「ラン・カーブ」と呼ばれています。

　R東日本では、この運転曲線を作成するシステムとして、1992年に運転曲線作成システム「ヘラクレス」を開発しました。ヘラクレスでは、図1のように、線路データ（距離、勾配、曲線等）と車両データ（加減速性能等）を基礎データとして管理し、与えられた条件下のもとで最小運転時分の運転曲線を作成します。また、この運転曲線をもとに、基準運転時分表を作成します。

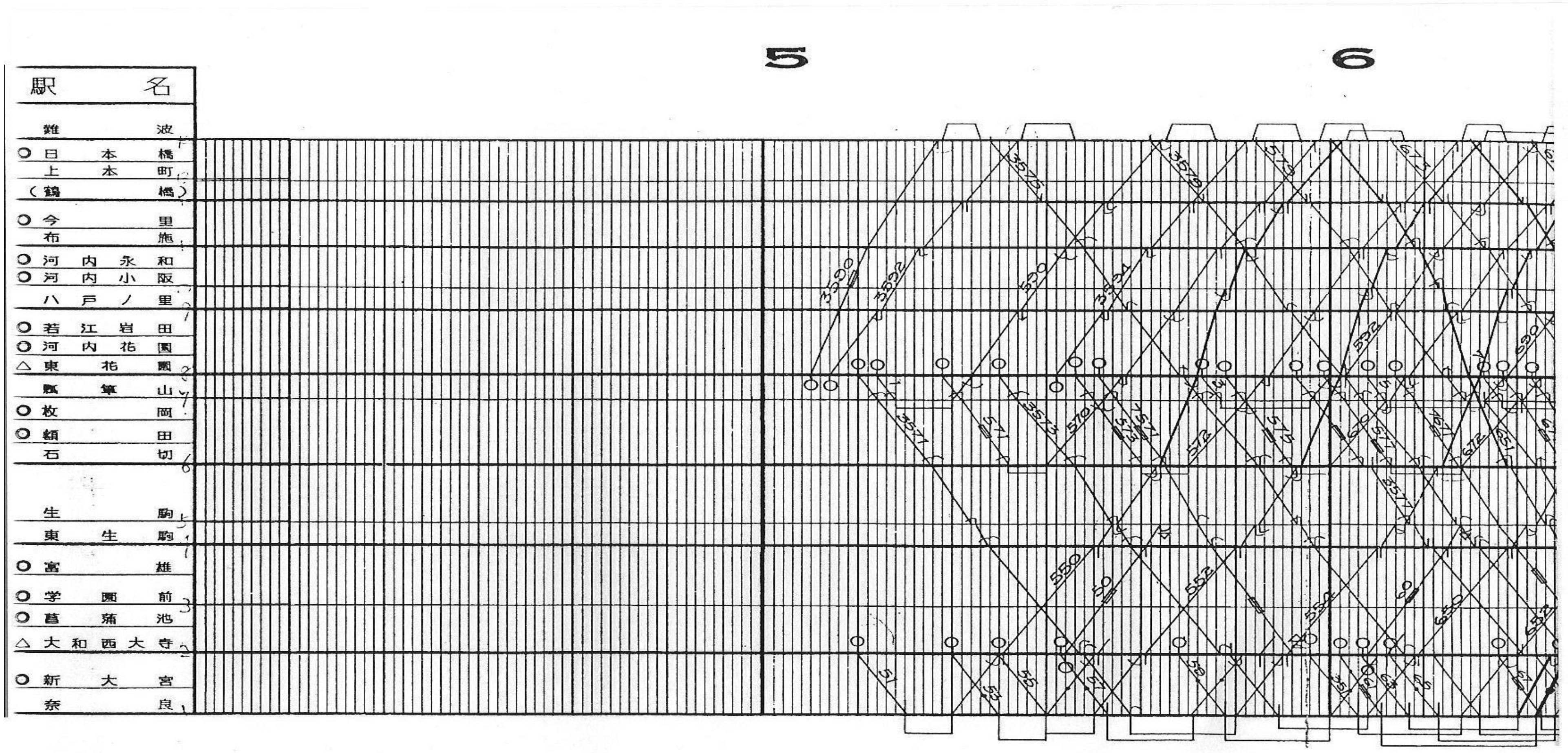


曲線半径
1000
350
300
350
400
500
800
800
曲線の速度制限
100
90
80
70
60
50
40
30
20
10
10
9
8
7
6
5
4
3
2
1
運転速度（km/h）
運転時分（分）
速度曲線
（力行）
（惰行）
ブレーキ曲線
分岐器の速度制限
時間曲線
−5
−6
勾配
L
3
L
L
4
10
L
−10
−7
L
3.5
8.3
キロ程
556
555
554
553
552
駅名
大阪
新大阪

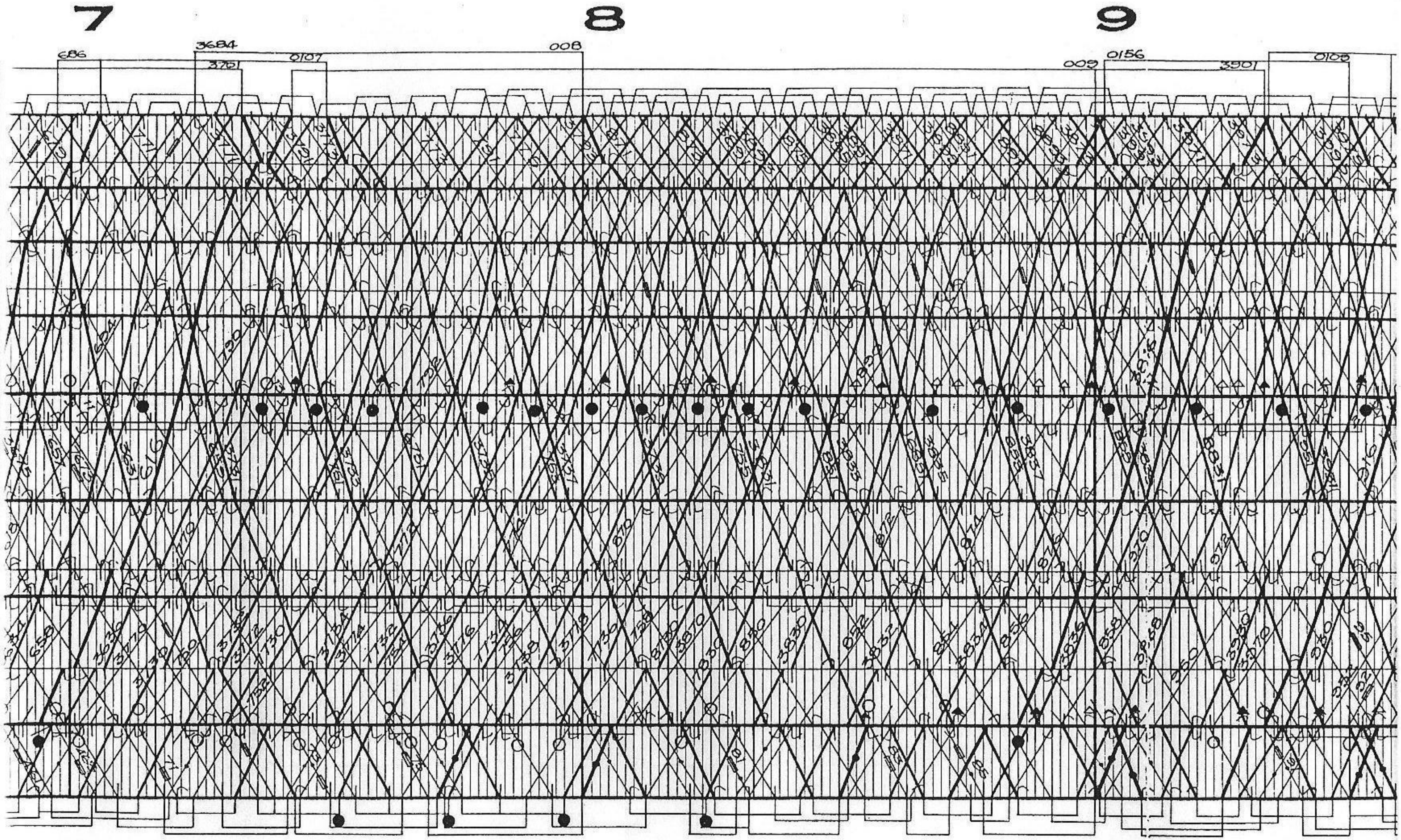
列車ダイヤ



実際の列車ダイヤの例



駅名
難波
日本橋
上本町
（鶴橋）
今里
布施
河内永和
河内小阪
八戸ノ里
若江岩田
河内花園
東花園
瓢箪山
枚岡
額田
石切
生駒
東生駒
富雄
学園前
菖蒲池
大和西大寺
新大宮
奈良
5
6


7
8
9
686
3684
3751
0107
008
009
0156
3501

3・1 列車自動運転システムとは

このシステムは，運転士に代り，出発から次の駅への停止までを計算機制御により行う。ここでは，安全性を確保し，乗り心地良く，制限速度を守り，正確に停止し，エネルギーを節約，短時間で運転することがシステムの制御目的である。このファジィ制御システムは，これらの制御目的に基づく制御則を制御知識としてもち，走行速度，制限速度，地点信号，走行指令に基づき，加速・減速指令である力行指令，ブレーキ指令をファジィ推論により決定する（図1）。

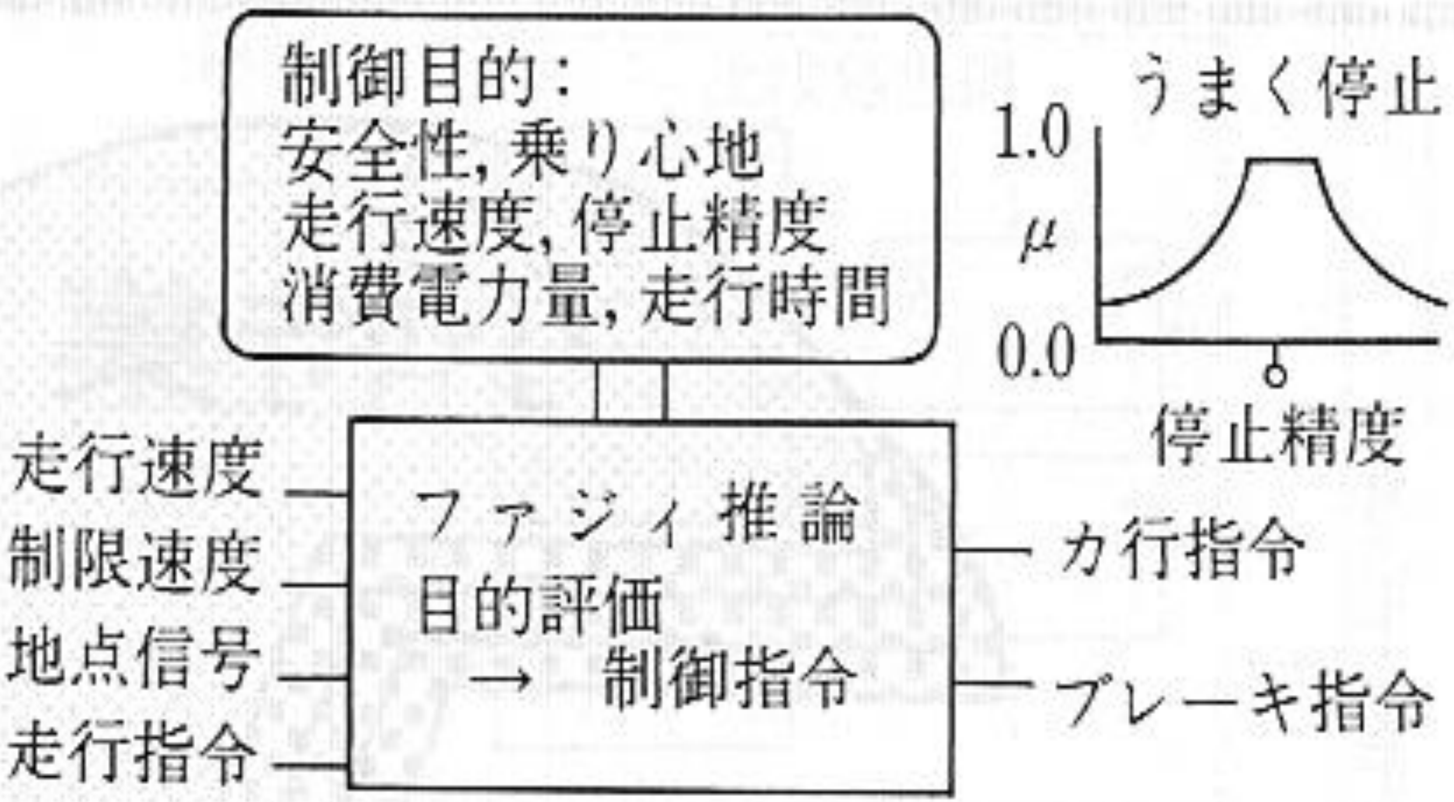


図 1　列車自動運転のファジィ制御

ここでは，「加速指令を少し増やしたとき，制御速度を守った運転ができそうか？」，「今のブレーキのままだと，どこに停止しそうか？」，「ブレーキを変化させれば，どこに停止しそうか，乗り心地はどうか？」という制御目的の評価を行う。例えば，駅に停止する間際には「ブレーキを少し強くして，乗り心地良く，正確に駅に止まれそうならば，ブレーキを少し強くする」といった熟練者制御則群に基づき最もうまく制御できる制御指令を選択する。

図 2　予見ファジィ制御の推論過程

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，

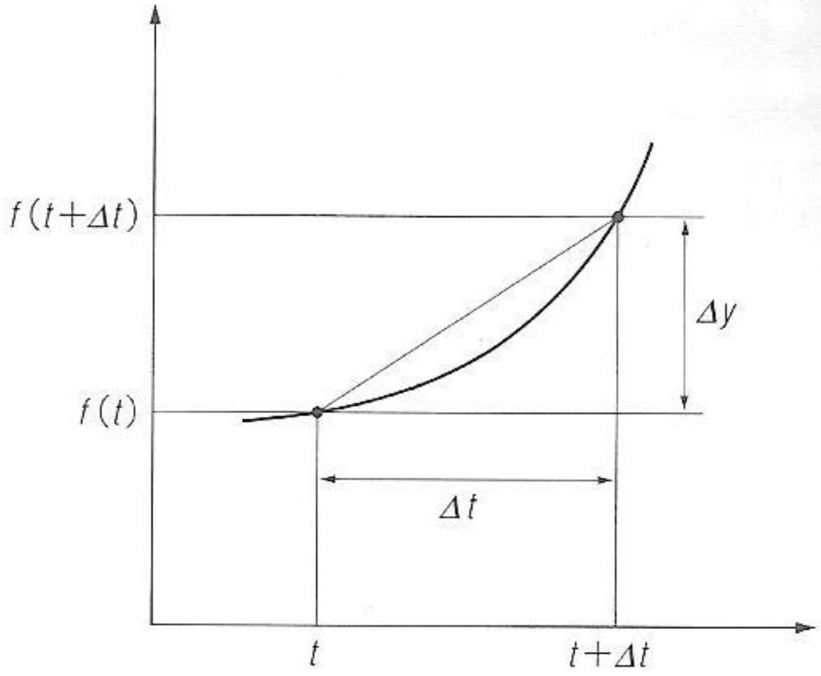
$$\frac{f(t+\Delta t)-f(t)}{\Delta t}=\frac{\Delta y}{\Delta t}=4t$$

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，

$$f'(t)=4t,\ あるいは\frac{dy}{dt}=4t$$

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。



列車運行シミュレーション

＜シミュレーション方式＞

- 連続型シミュレーション

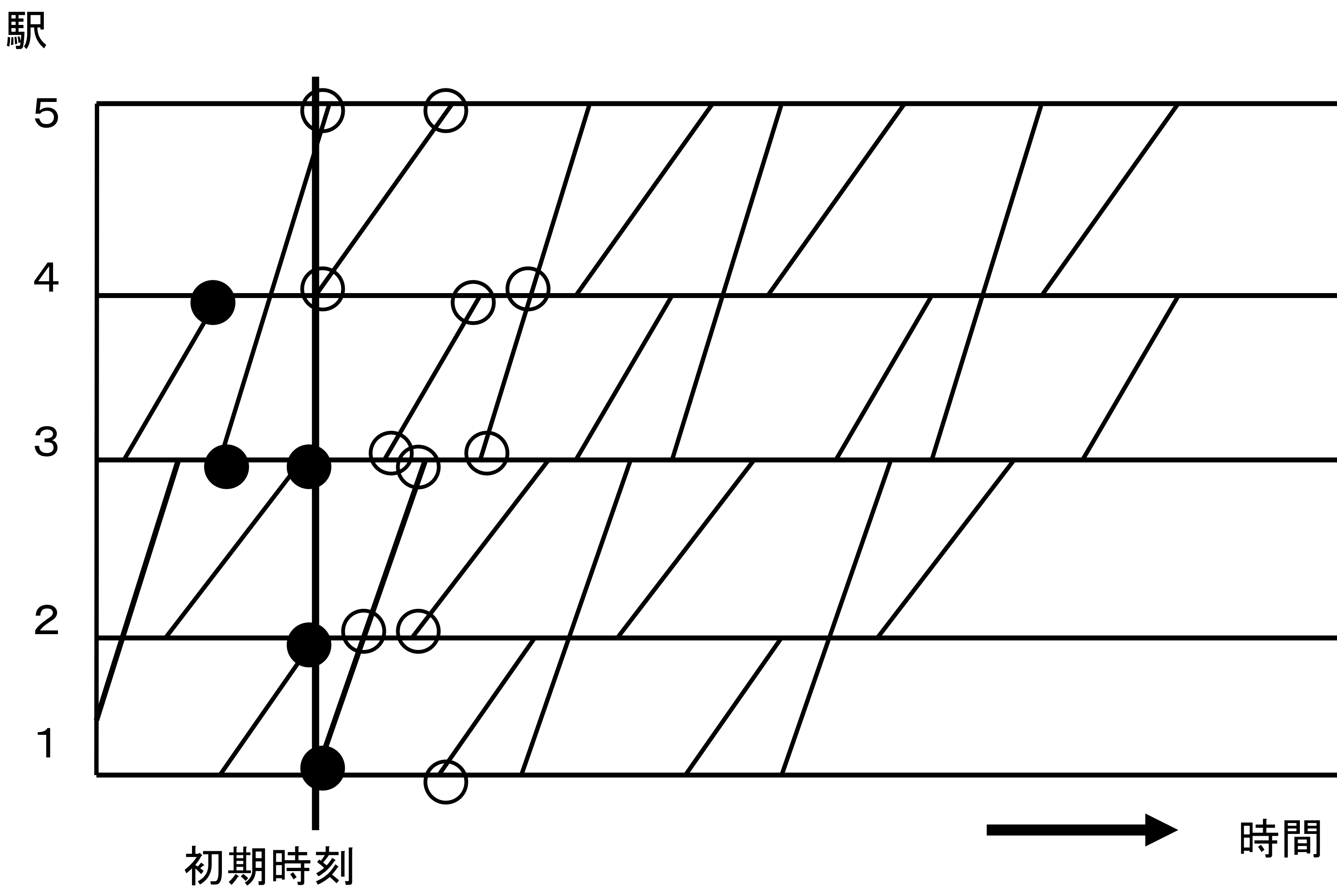
  実際の列車走行

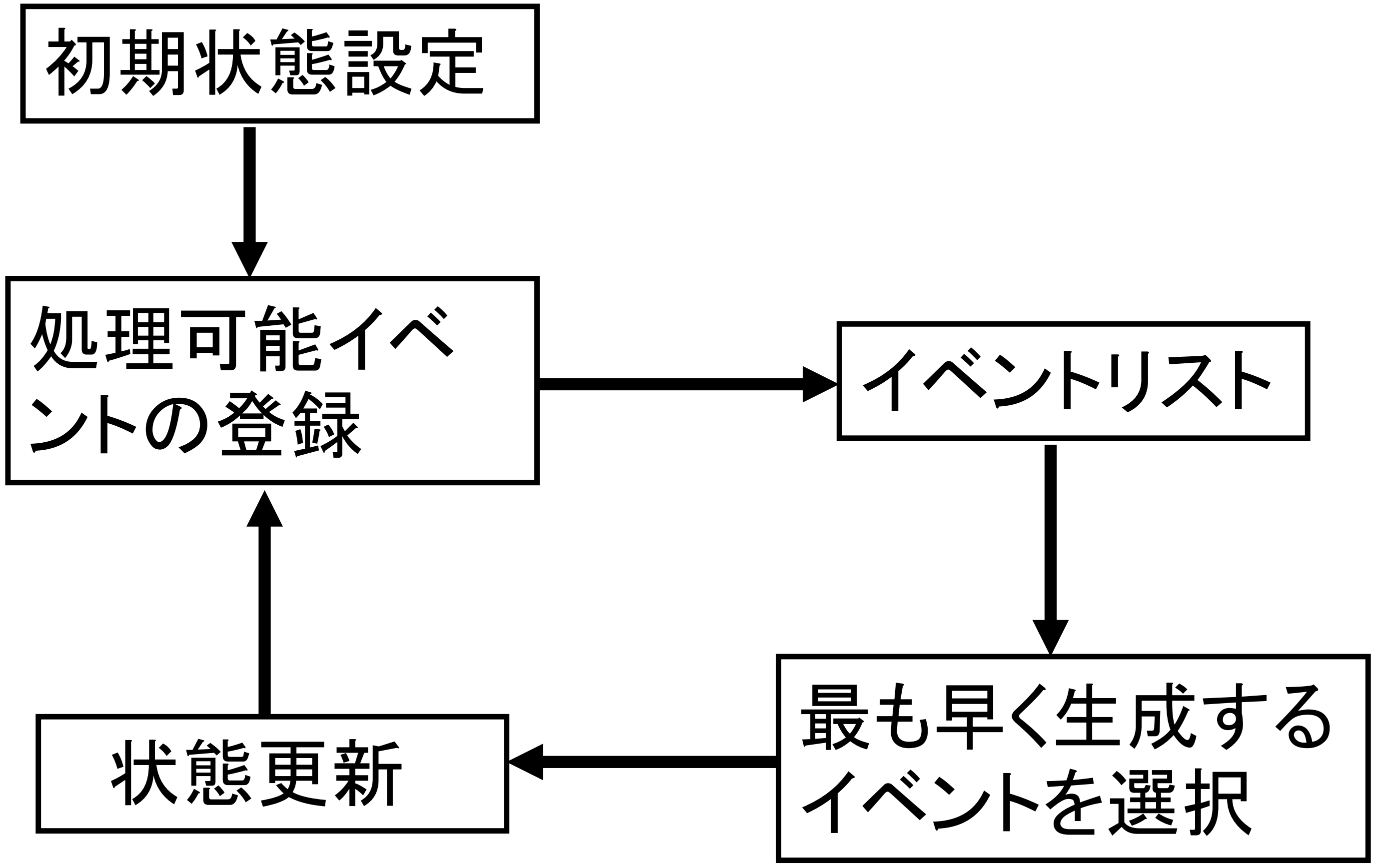
- 離散型シミュレーション

  ー＞　イベントシミュレーション

  列車の駅着発単位

イベントシミュレーション




初期状態設定
処理可能イベントの登録
イベントリスト
最も早く生成するイベントを選択
状態更新

列車運行のネットワークモデル

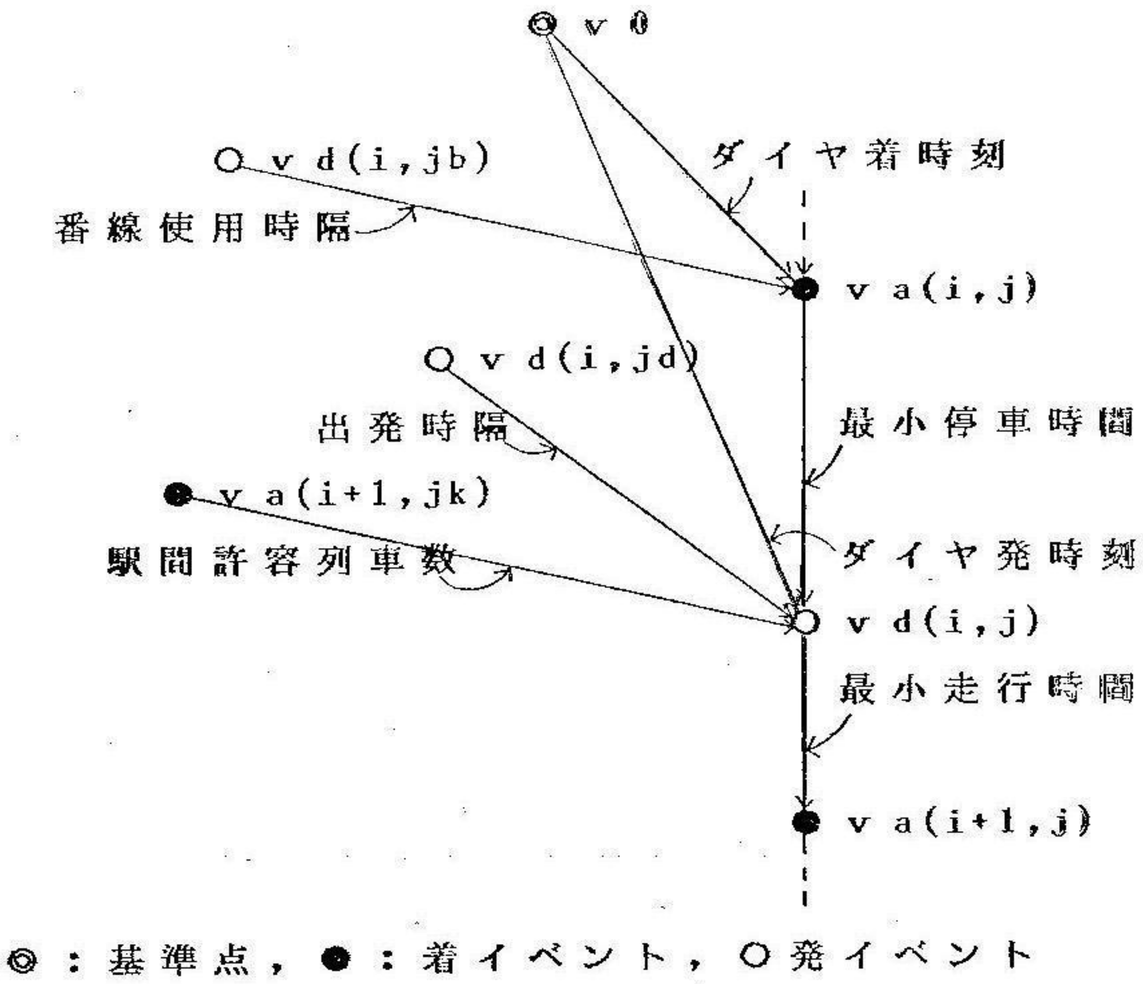
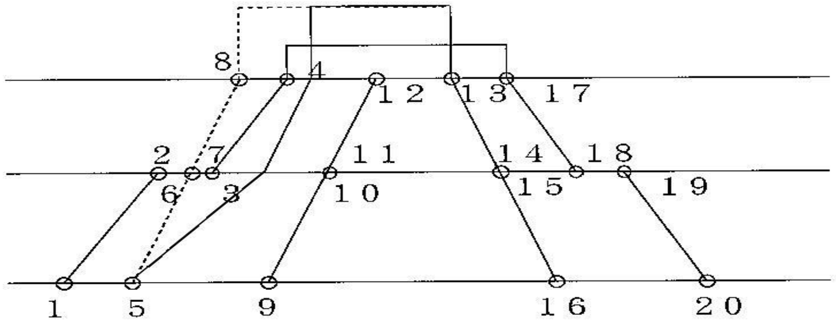
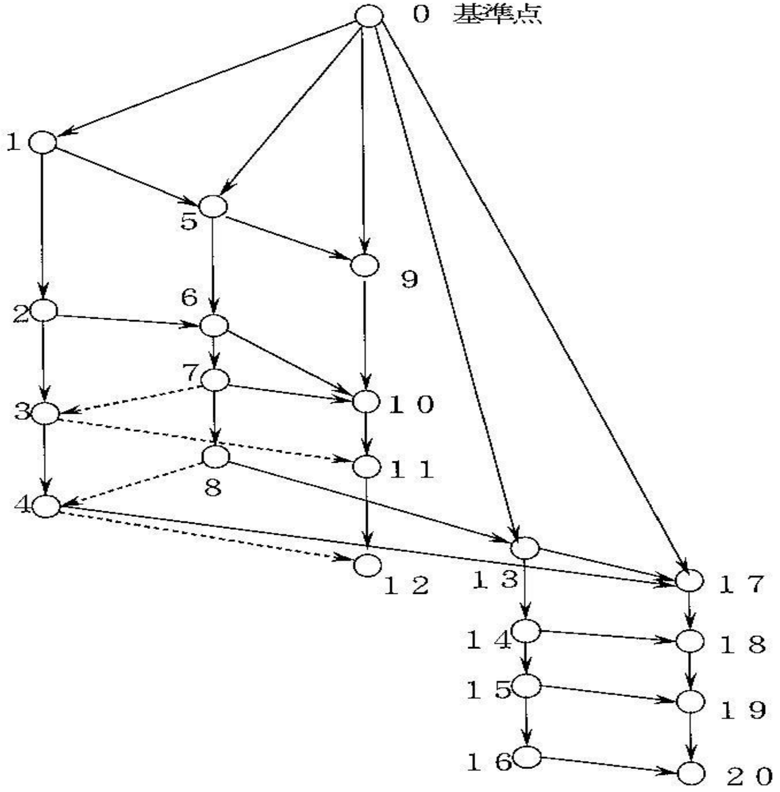


図１．列車運行のネットワーク表現

列車運行のネットワークモデル



⇓



奈多駅
和白駅
福岡CC
原上
立花口
新宮霊園
雁ノ巣駅
平山
立花山
唐ノ原駅
東区
鹿児島本線
遠見岳
牧ノ鼻
三日月温泉
香椎花園前駅
東区
シオヤ鼻
福岡湾
海ノ中道駅
香椎花園
香住ヶ丘1
猪野
久山町
山陽新幹線
西戸崎ｼｰｻｲﾄﾞCC
マリンワールド海の中道
海の中道海浜公園
香椎線
カシイGC
香椎
城ノ越山
猪野川
山の神
香椎参道
久山CC
池上池
城浜
香椎神宮駅
谷口
西戸崎駅
大岳GC
名香野駅
新建川
城山
久山町役場
西戸崎
端島
名島駅
森江山
深井
久大薬用植物園
八田
東区
土井駅
江辻山
蒲田池
極楽池
江辻池
福岡IC
篠栗北
松島
和田橋
福岡湾
都市高速1号線
大隅
粕屋町
多々良川
篠栗駅
篠栗町役場
土手崎
伊賀駅
長者原
門松
尾仲
馬手池
乙犬
西浦池
博多港
東区役所
篠栗線
柚須駅
原町駅
古渡船場
須恵川
乙犬山
切通池
若杉
鵜来島
博多港ﾌｪﾘｰのりば
福岡県庁
粕屋町役場
柿元池
福岡市
別府
酒殿駅
植木
大谷池
岳城山
姪浜港ﾌｪﾘｰのりば
西公園
須崎公園
福岡空港
須恵
須恵駅
都市高速道路
中央区
博多区役所
榎田
青木
志免町役場
須恵PA
室見川
天神
上須恵
福岡ﾀﾜｰ
樋井川
中央区役所
福岡市役所
博多駅
博多区
須恵中央駅
大的
東平尾公園
須恵町役場
大濠公園
早良区役所
樋井川
美術館
薬院駅
比恵
博多の森陸上競技場
須恵町
九州自動車道
西区役所
早良口
脇山口
鹿原山
城南区役所
南公園
宇美川
新原駅
新原
月隈
志免町
香椎線
柳原

地図全体の情報

↓

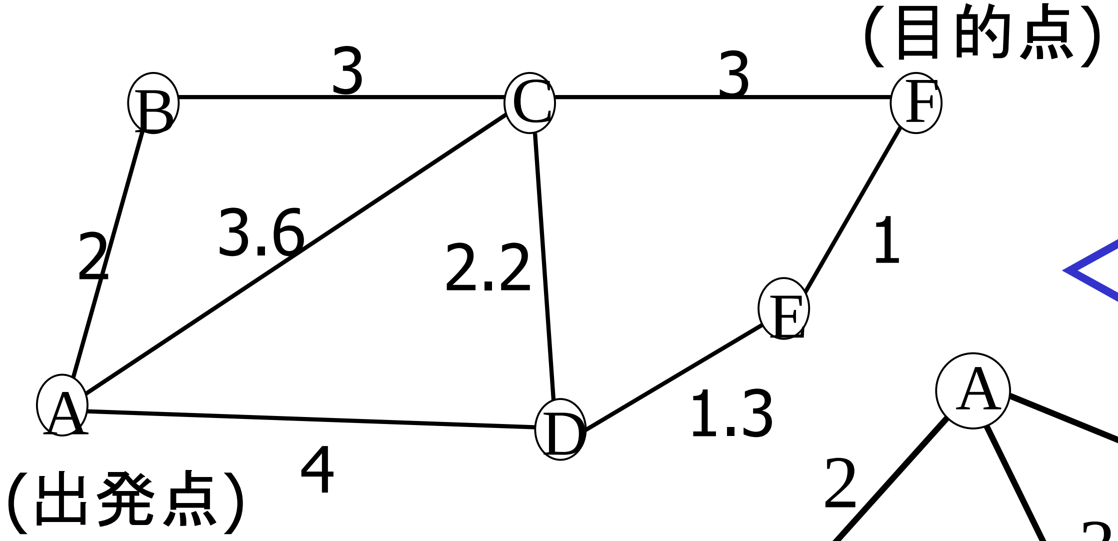
道路に関する情報

↓

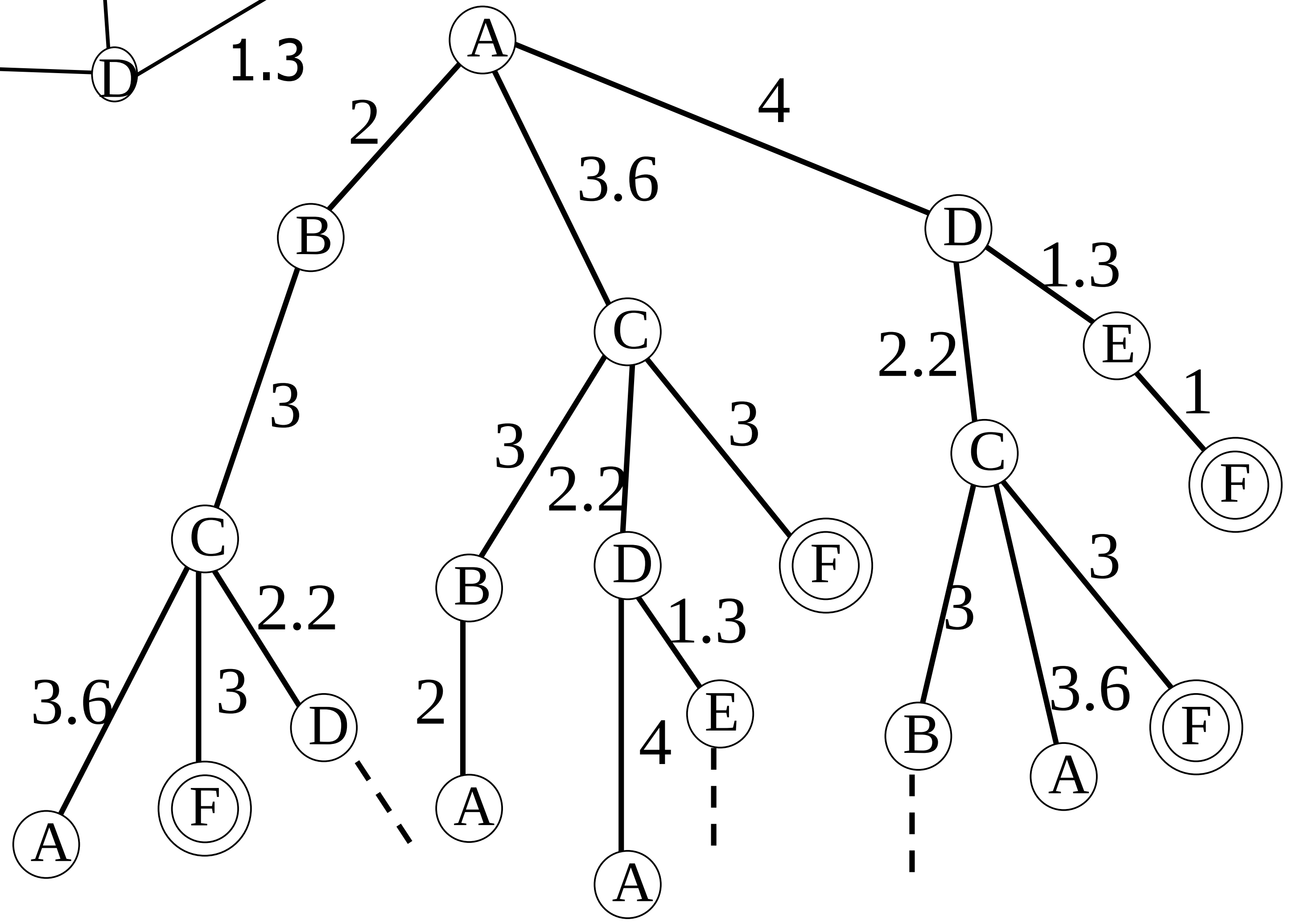
グラフ化

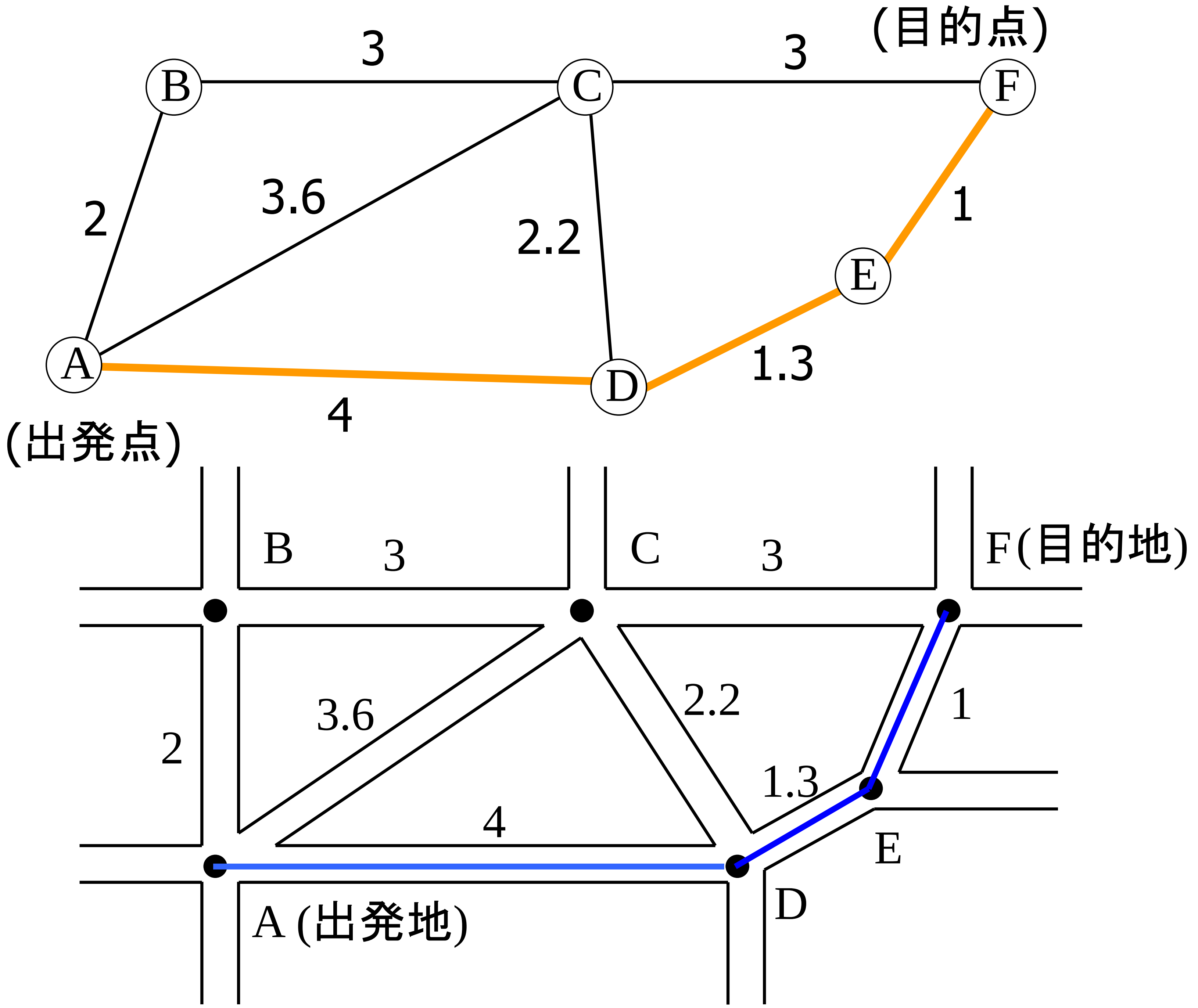
解決するのに適した形で問題を表現する

⇒ モデル化


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

＜経路探索木＞

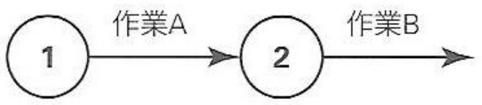

A
2
3.6
4
B
C
D
3
3
2.2
3
2.2
1.3
E
1
F
C
B
D
F
C
3.6
3
2.2
2
1.3
3
3
3.6
A
F
D
A
A
E
4
B
A
F



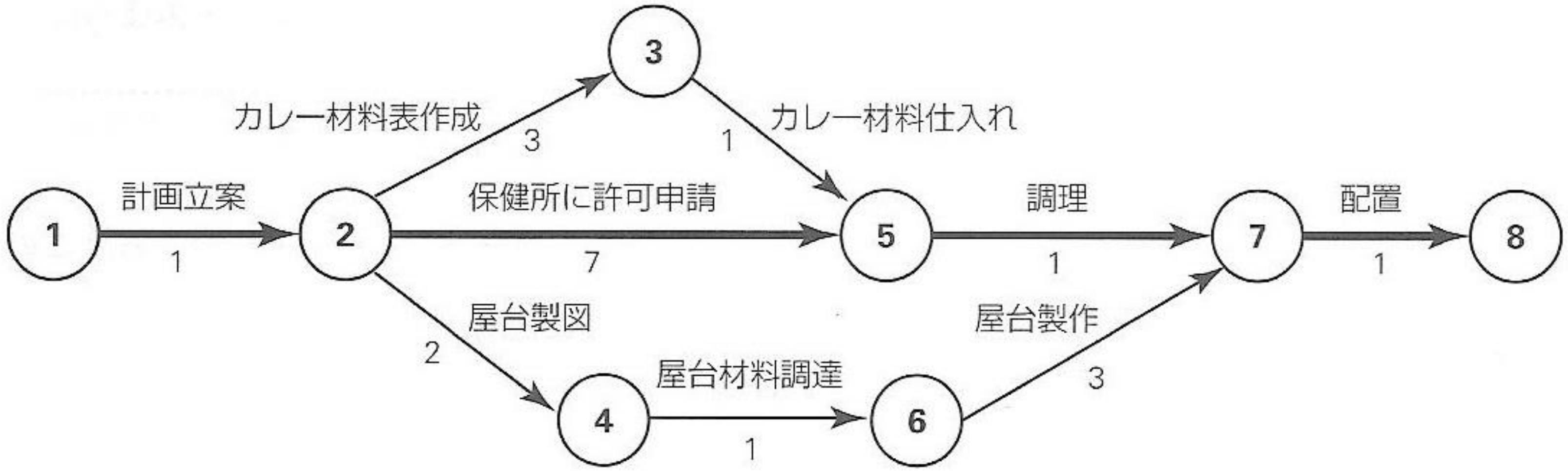
(目的点)
A
B
C
D
E
F
3
3
2
3.6
2.2
1
1.3
4
(出発点)
B
3
C
3
F(目的地)
2
3.6
2.2
1
1.3
4
E
D
A (出発地)

■表1　基準作業リスト

作業内容	先行作業	必要日数	必要人数
計画立案	なし	1	3
保健所に許可申請	計画立案	7	1
カレー材料表作成	計画立案	3	2
カレー材料仕入れ	材料表作成	1	2
調理	仕入れ，許可申請	1	3
配置	調理，屋台製作	1	3
屋台製図	計画立案	2	2
屋台材料調達	屋台製図	1	2
屋台製作	屋台材料調達	3	3



■図5　進行過程のあらわしかた



■図6　カレー模擬店の作業工程のネットワークモデル

時刻	イベント	生成時刻	終了時刻
0	計画立案	0	1
1	保健所許可申請	1	8
	カレー材料表	1	4
	屋台製図	1	3
3	屋台材料調達	3	4
4	カレー材料仕入れ	4	5
	屋台製作	4	7
8	調理	8	9
9	配置	9	10

ワインの保管料と送料
何カ月おきの注文が得か

在庫管理

問題

フランス料理店を経営する山崎さんは、お気に入りのシャトーからワインを直輸入しようと考えています。ワインは1月に1ケース消費します。ワインの送料は、ケースの数に関わらず、1回あたり10,000円です。

ワインを保管するためのワインセラーのレンタル費用は、1ケースあたり月に1,000円です。たとえば半年おきに注文する場合、ワインセラーは6ケース分確保しておく必要があり、月に6,000円のレンタル費用が必要になります。何カ月おきに注文するのが最も得でしょうか？

解決

注文の間隔を短くすると、セラーのレンタル費は少なくてすみますが、送料が余分にかかります。注文の間隔を長くすると、送料は少なくてすみますが、レンタル費が余分にかかります。

これをまとめたのが下の表です。費用をもっとも安くするには、3カ月に1回注文すればいいことがわかります。これが経済的発注量（EOQ）です。

でも実際は、よく売れる月も売れない月もあるので、これよりは少し在庫を余分に持っておくのが良いでしょう。

	ワインの送料	セラーのレンタル料	合計（円）
1カ月に1回	120,000	12,000	132,000
2カ月に1回	60,000	24,000	84,000
3カ月に1回	40,000	36,000	76,000
4カ月に1回	30,000	48,000	78,000
5カ月に1回	24,000	60,000	84,000
6カ月に1回	20,000	72,000	92,000
…	…	…	…
1年に1回	10,000	144,000	154,000

・経済的発注量（EOQ)

在庫費と発注費の和が最小となるロットサイズ（一度に注文する量）

・在庫の補充方法

発注点法：現在の在庫量を基準

定期発注法：一定日数おきに発注

・ABC分析

優先順位をつけて管理

・リードタイム

ジャストインタイム、かんばん方式

例題1　入園券販売窓口の待ち行列（サービス時間が一定の場合）

ある遊園地には入園券販売窓口が１つある。客の到着間隔の状況が表１のような累積確率となっている。今後くる５人の客について，どのような待ち行列ができるか調べてみよう。なお，サービス時間は30秒で一定とする。

手順

▶１．考え方

過去の客50人の到着間隔は表１のとおりであり，サービス時間は30秒で一定である。

最初の客の到着時間を０とする。客２～５の到着時間は，一様乱数の値が累積確率の範囲のどこに入るか[①]を求め，到着間隔を決定する。つまり，乱数の値と「累積確率」列の数値を比較して，累積確率の数値が乱数の値より大きく最も近い累積確率の行に着目して，その行の中央値を到着間隔とする。

■表1　客の到着間隔（過去の状況）

到着間隔（秒）	中央値	度数	確率	累積確率
0以上10未満	5	13	0.26	0.26
10以上20未満	15	19	0.38	0.64
20以上30未満	25	9	0.18	0.82
30以上40未満	35	6	0.12	0.94
40以上50未満	45	2	0.04	0.98
50以上60未満	55	1	0.02	1.00
60以上		0	0.00	
	合計	50	1.00	

次に，客の到着時間に窓口があいているかどうかを調べ，あいていれば窓口の状態を「あき」とし，そうでなければ「待ち」とする。

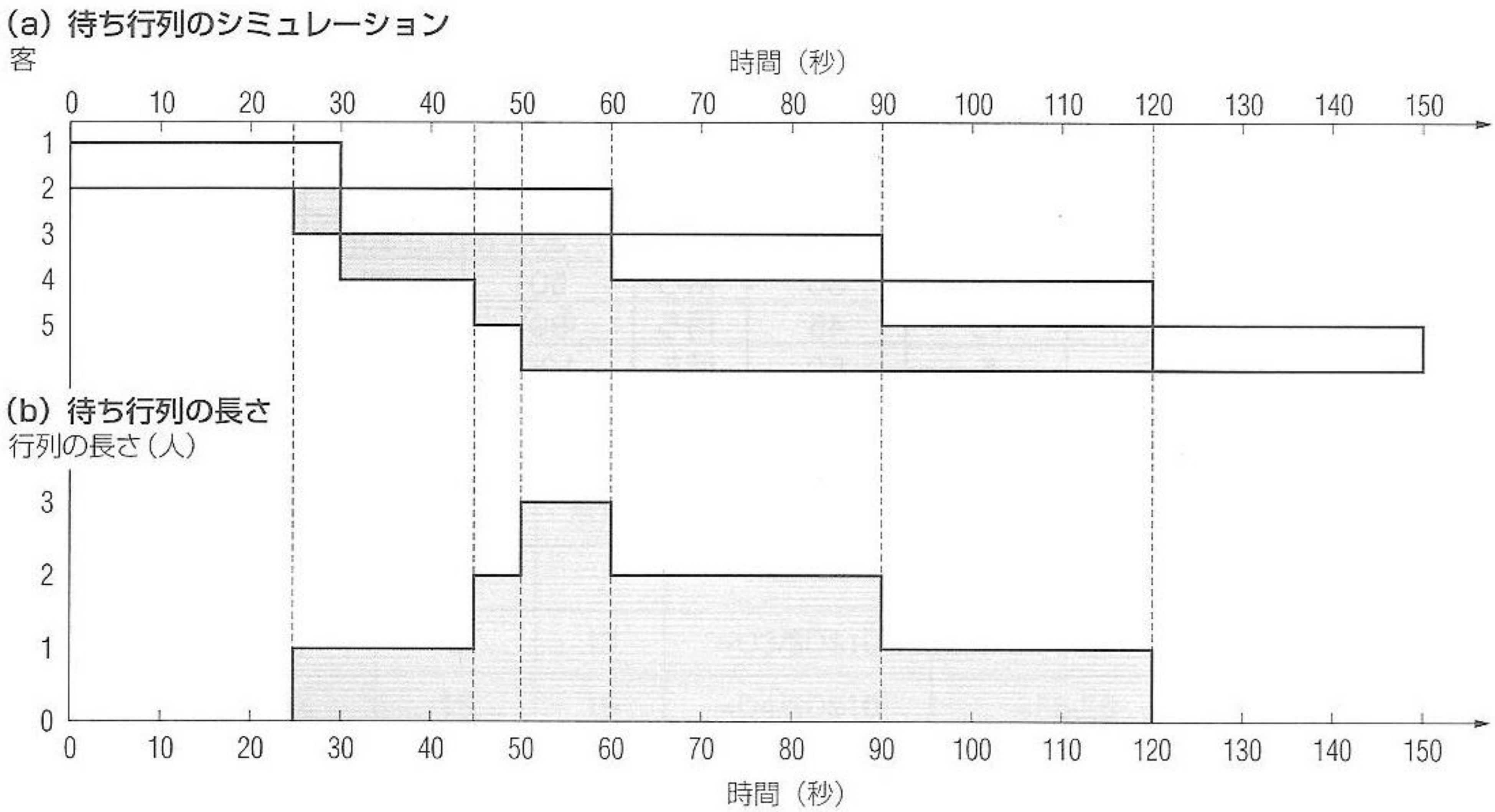
サービス開始時刻は，窓口が「あき」ならば「到着時刻」，そうでなければ「前に並んでいる客のサービス終了時刻」とする。

サービス終了時刻は，サービス開始時刻にサービス時間をたした値とする。

①指定したセルの値に対応するデータを表示するLOOKUP関数を用いることにより，求めることができる。

	A	B	C	D	E	F	G	H	I
1		到着間隔(秒)	中央値	度数	確率	累積確率			
2						−1			
3		0以上10未満	5	13	0.26	0.26			
4		10以上20未満	15	19	0.38	0.64			
5		20以上30未満	25	9	0.18	0.82			
6		30以上40未満	35	6	0.12	0.94			
7		40以上50未満	45	2	0.04	0.98			
8		50以上60未満	55	1	0.02	1.00			
9		60以上		0	0.00				
10			合計	50	1.00				
11									
12	客	一様乱数	到着間隔	到着時刻	窓口	サービス開始時刻	サービス時間	サービス終了時刻	待ち時間
13	1			0	あき	0	30	30	0
14	2	0.7725	25	25	待ち	30	30	60	5
15	3	0.1078	5	30	待ち	60	30	90	30
16	4	0.4283	15	45	待ち	90	30	120	45
17	5	0.1098	5	50	待ち	120	30	150	70
18							平均待ち時間		30

■図1　客の到着時間と窓口の状況（今後の予想）


(a) 待ち行列のシミュレーション
客
時間（秒）
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
1
2
3
4
5
(b) 待ち行列の長さ
行列の長さ（人）
3
2
1
0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
時間（秒）

■図3　待ち行列のグラフ

客	一様乱数	到着間隔	到着時刻	窓口	サービス開始時刻	サービス時間	サービス終了時刻	待ち時間
1			0	あき	0	30	30	0
1	0.77	25	25	待ち	30	30	60	5
2	0.43	15	40	待ち	60	30	90	20
3	0.23	5	45	待ち	90	30	120	45
4	0.85	35	80	待ち	120	30	150	40
5	0.35	15	95	待ち	150	30	180	55
						平均待ち時間		33

＜エージェントシミュレーション＞

一定のルールに基づいて、自律的に行動するエージェントの振る舞いや、それらの相互作用から現れる、複雑な社会現象をシミュレーションする。

私たちの生活している社会においては、例えばエージェントは「ヒト」になりますし、道路交通網において は「自動車」に対応します。これらエージェントの個々の行動や移動、振る舞い、状態変化をコンピュータ上 で同時にシミュレーションする事で、エージェントが活動している社会全体の振る舞いを分析する事が出来 ます。

また、エージェントシミュレーションでは、エージェントの個々の振る舞いからでは、予測できなかったような社会システムの現象も予測する事ができます。このような現象は「創発」と呼ばれます。

エージェントシミュレーションの例として、ツイッター上に新製品発売に関する情報が広がる様子をを考えます。これをエージェントシミュレーションでモデル化すると、エージェントはツイッターユーザであり、エージェントの行動は情報を目にしてツイートする、リツーイトする、何も行動しない、とパターン化されます。

また、エージェントが活動する社会は、ユーザ間のフォロー、フォロワー関係のネットワーク構造であり、口コミは構築されているユーザ同士のネットワーク上を伝播していきます。このようにコンピュータ上で ユーザの個々の行動をシミュレーションしてみると、ツイッター上で新製品に関する情報を目にするユーザ数の推移を予測できます。

あるいは、エージェントの移動を行動ルールとしてモデル化すると、火災発生時の避難の様子などもシミュレーションする事が出来、最適な避難経路の設計や、非常灯の配置などに役立てる事ができます。同様に、商用施設や空港、駅などの施設に適用すると、混雑緩和策の検討、施設の要員計画にも活用できます。

道路交通網においては、自動車の走行をエージェントシミュレーションする事で、渋滞緩和策として、信号制御方法や、道路建設などを検討する事が出来ます。

このように様々な場面でエージェントシミュレーションは有効ですが、利点として、エージェントの状態や 行動ルールを複雑にする事で、より現実に近いモデル化も可能となる点も挙げられます。

＜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

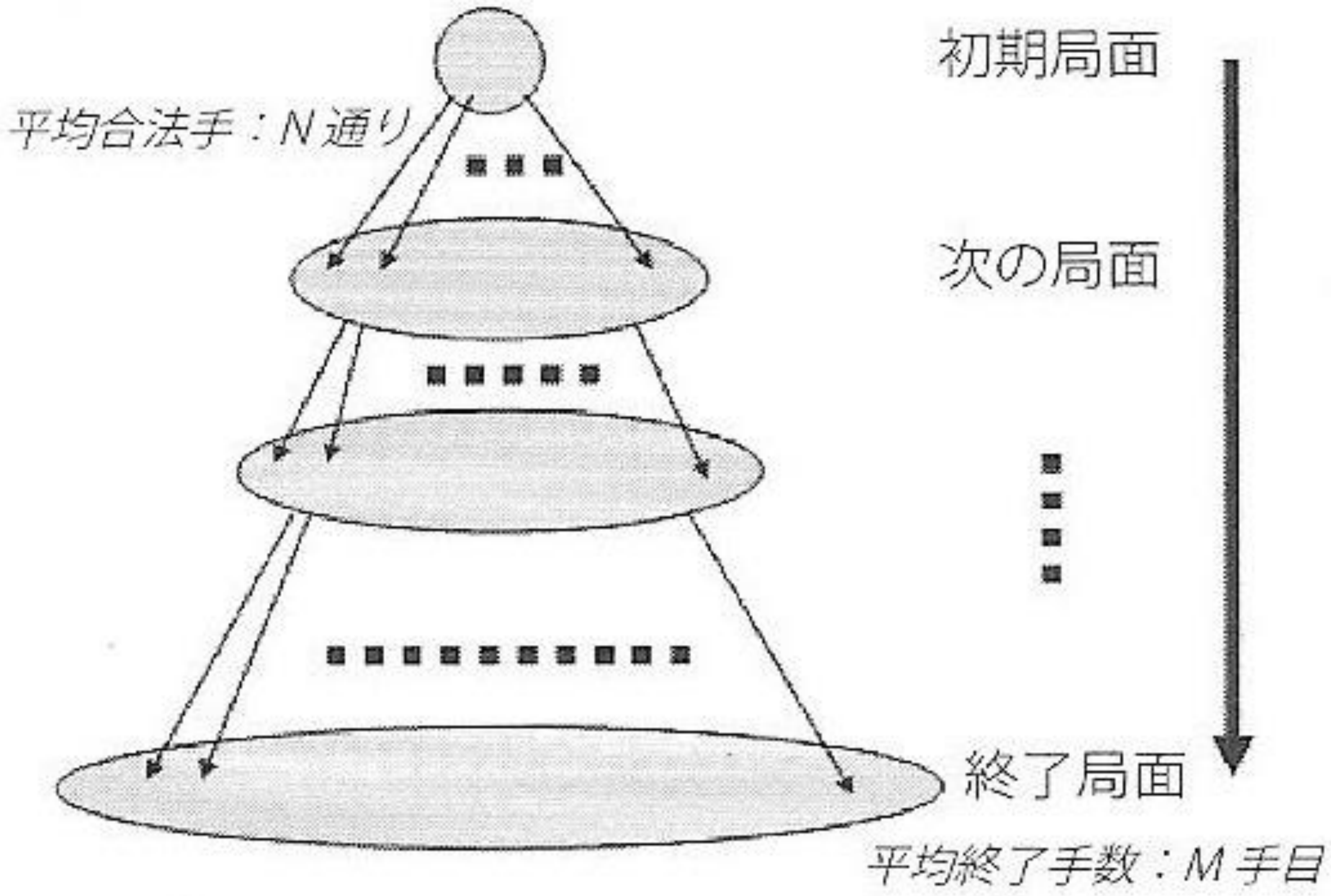


図３　ゲーム木探索によるゲームの複雑さ

コンピュータ将棋の技術

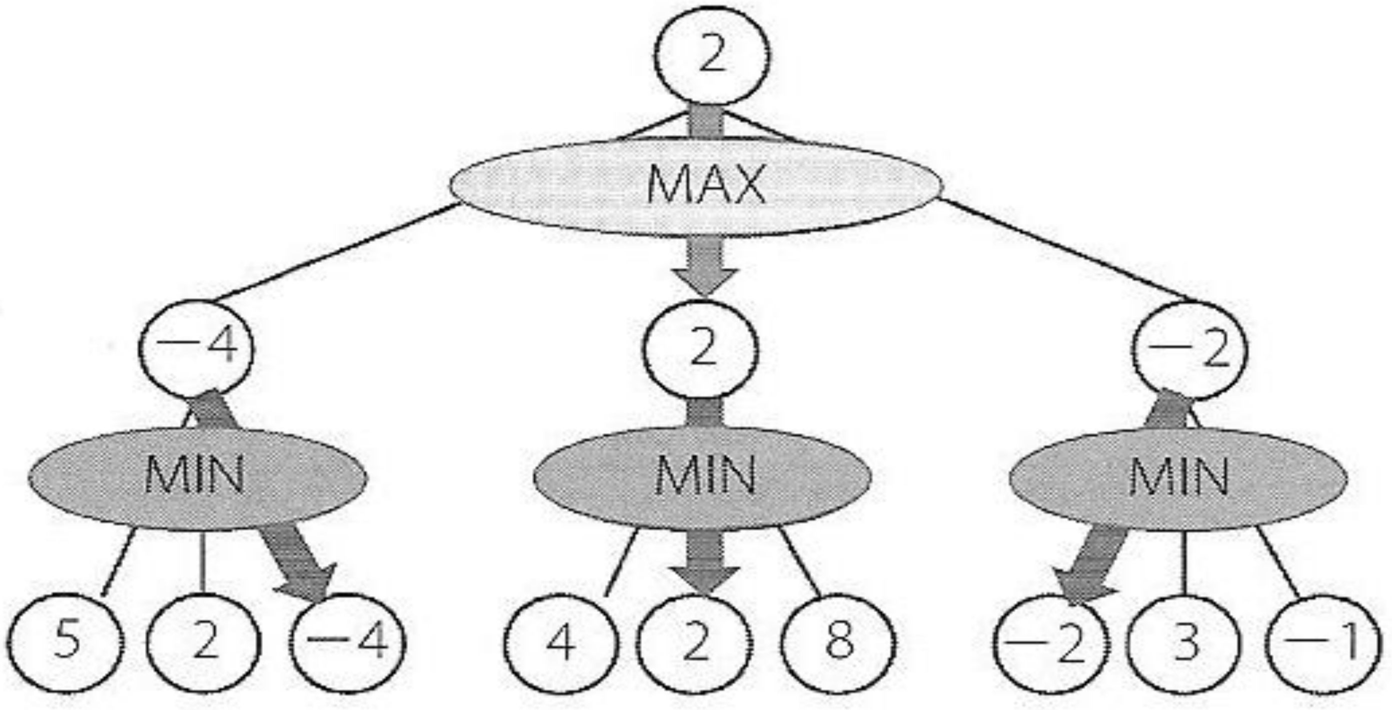


図４　ゲーム木のミニマックス探索

小野田博一：

人工知能はいかにして強くなるのか？

対戦型AIで学ぶ基本の仕組み

講談社、2017

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。

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、

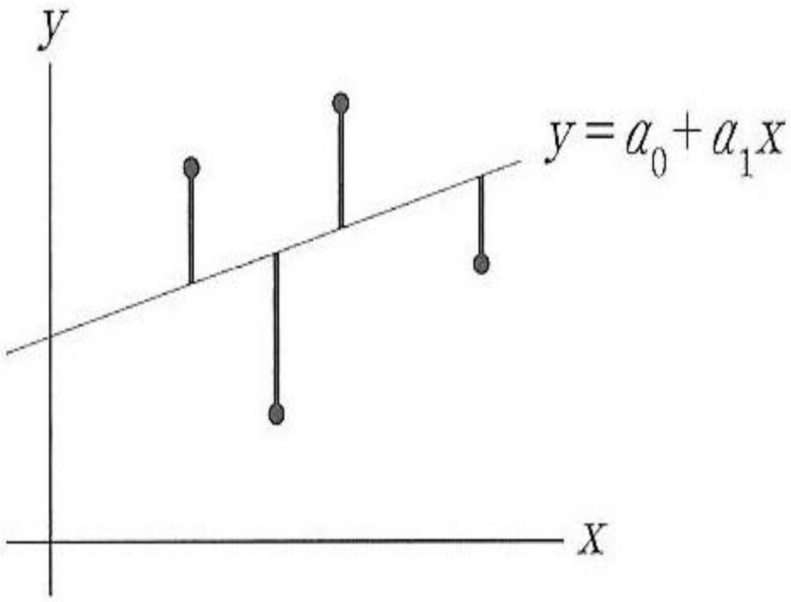
$$y=a_0+a_1x_1+a_2x_2$$

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。

X, Yを以下の行列として（Xの第１列に１がずらっと並んでいるのは、定数項a_0を求めるためです）、各係数a_0, a_1, a_2は、$(X'X)^{-1}X'Y$で得られます（X'はXの転置行列、Z^{-1}はZの逆行列です）。

$$X = \begin{pmatrix} 1 & 50 & 60 \\ 1 & 60 & 55 \\ 1 & 65 & 65 \\ 1 & 70 & 63 \\ 1 & 75 & 70 \\ 1 & 80 & 72 \end{pmatrix} \qquad Y = \begin{pmatrix} 51 \\ 59 \\ 57 \\ 63 \\ 66 \\ 70 \end{pmatrix}$$

まず、$X'X$は

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 50 & 60 & 65 & 70 & 75 & 80 \\ 60 & 55 & 65 & 63 & 70 & 72 \end{pmatrix} \begin{pmatrix} 1 & 50 & 60 \\ 1 & 60 & 55 \\ 1 & 65 & 65 \\ 1 & 70 & 63 \\ 1 & 75 & 70 \\ 1 & 80 & 72 \end{pmatrix} = \begin{pmatrix} 6 & 400 & 385 \\ 400 & 27250 & 25945 \\ 385 & 25945 & 24903 \end{pmatrix}$$

この逆行列は、

$$\frac{1}{231100} \begin{pmatrix} 5463725 & 27625 & -113250 \\ 27625 & 1193 & -1670 \\ -113250 & -1670 & 3500 \end{pmatrix}$$

これに右から

$$X'Y = \begin{pmatrix} 366 \\ 24755 \\ 23639 \end{pmatrix}$$

を掛けて、

$$\begin{pmatrix} \frac{258539}{9244} \\ \frac{33267}{46220} \\ -\frac{1077}{4622} \end{pmatrix}$$

となります。上から順に、a_0, a_1, a_2の値です$\left(\frac{258539}{9244} \fallingdotseq 28.0,\ \frac{33267}{46220} \fallingdotseq 0.720,\ -\frac{1077}{4622} \fallingdotseq -0.233\right)$。

$$y = a_0 + a_1 x_1 + a_2 x_2$$

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

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

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

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

$$\begin{aligned} f(\boldsymbol{a}) &= {}^t(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{a})(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{a}) \\ &= {}^t\boldsymbol{Y}\boldsymbol{Y} - 2\,{}^t\boldsymbol{Y}\boldsymbol{X}\boldsymbol{a} + {}^t\boldsymbol{a}\,{}^t\boldsymbol{X}\boldsymbol{X}\boldsymbol{a} \end{aligned}$$

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

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

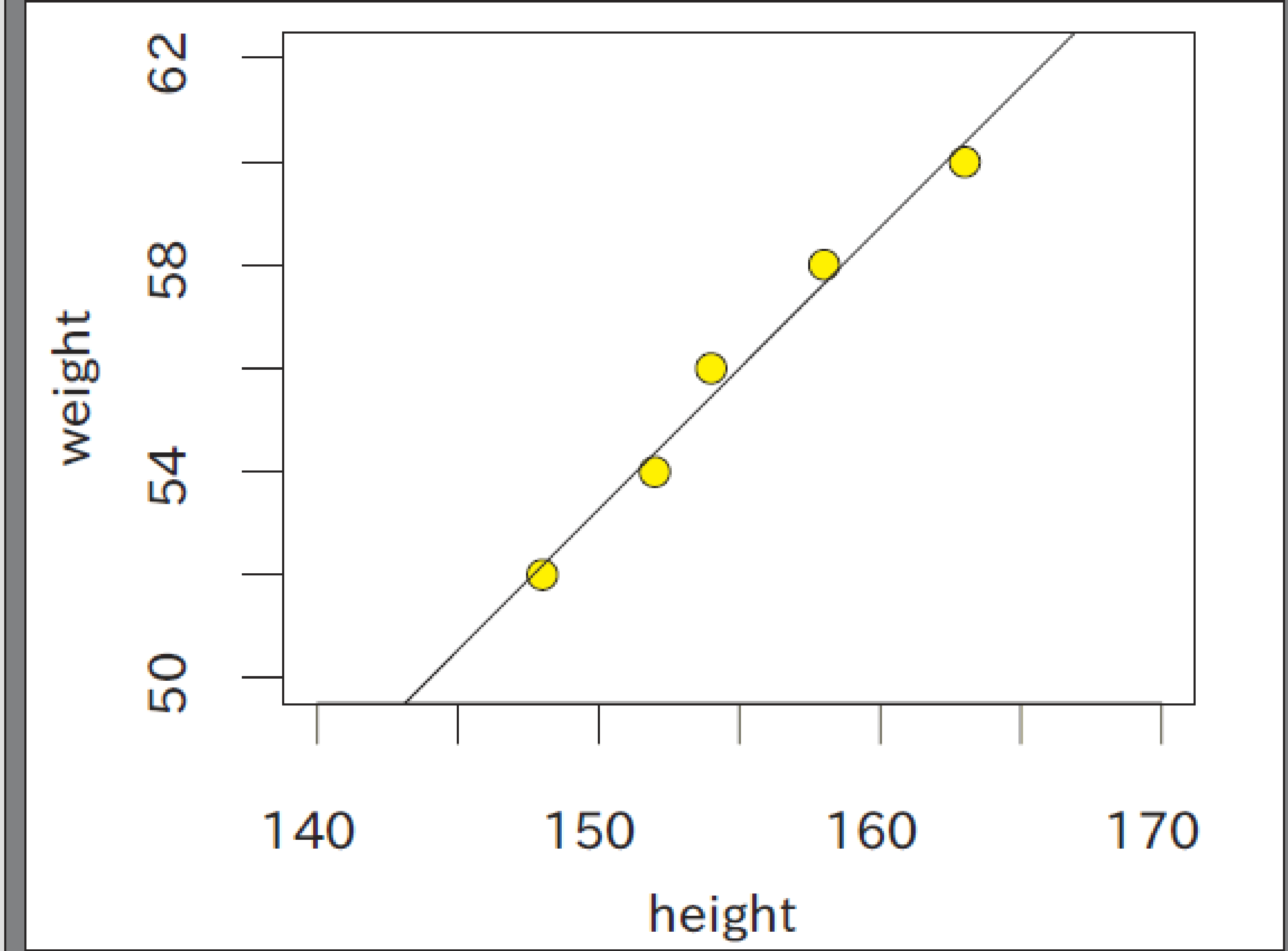
$$\boxed{\boldsymbol{a} = ({}^t\boldsymbol{X}\boldsymbol{X})^{-1}\,{}^t\boldsymbol{X}\boldsymbol{Y}}$$

放送大学

[データの分析と知識発見]

回帰分析

	身長	体重
A01	148	52
A02	152	54
A03	154	56
A04	158	58
A05	163	60



回帰分析

（線形）回帰分析とは、ある変数（目的変数）をいくつかの変数（説明変数）の（線形）結合で表現する手法である。

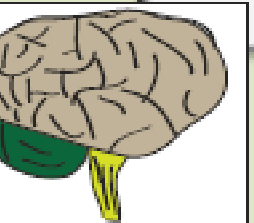
説明変数が一つの時を単回帰分析
説明変数が複数ある時を重回帰分析という。

```
---------------身長一体重----------
h1 <- read.csv("w1.csv", header=T, row.names=1)
View(`h1`)
h1

h2 <-lm( weight~height, data=h1)
h2
plot(h1, pch=16 )

abline( h2 )

--------------科目成績------------------------------
p1 <- read.csv("kamoku.csv", header=T, row.names=1)
p2 <-lm( 数学~理科 + 社会, data=p1)
```

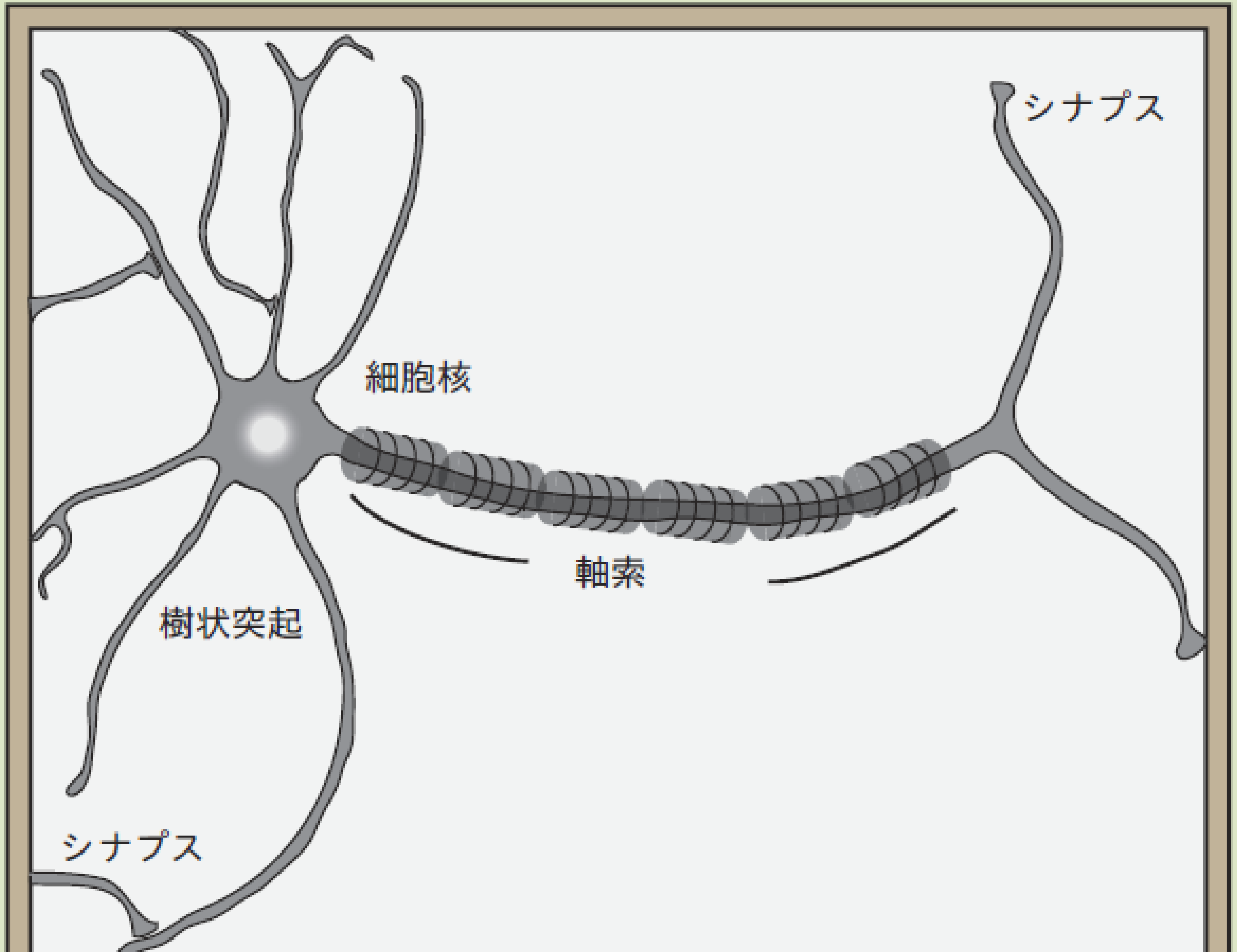
ニューラルネットワーク

ニューラルネットワークとは
脳の神経回路網のことであり,
それを模したもの。

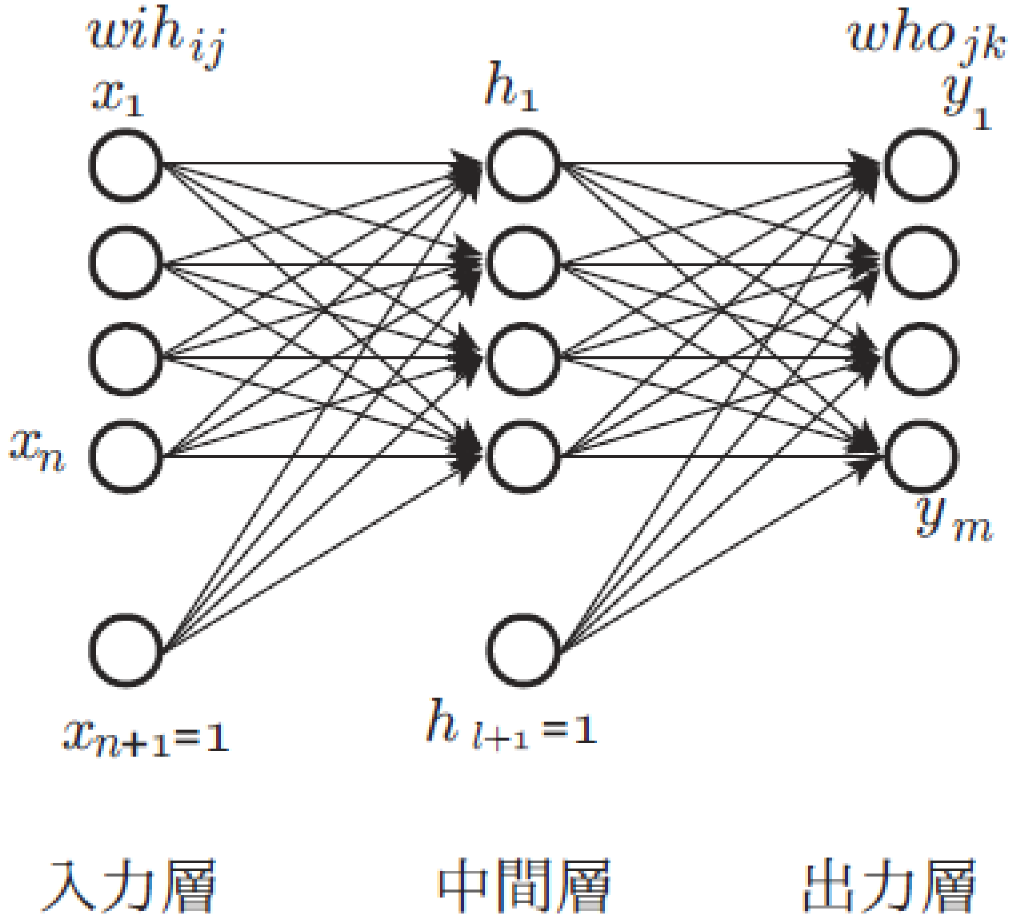
脳を真似ることで、生体が持つ機能を
機械に持たせることを目的とする。



神経細胞



バックプロパゲーション



入力

$$\boldsymbol{x}^{(p)} = \begin{pmatrix} x_1^{(p)} \\ x_2^{(p)} \\ \vdots \\ x_n^{(p)} \\ x_{n+1}^{(p)}(=1) \end{pmatrix}$$

出力 教師信号

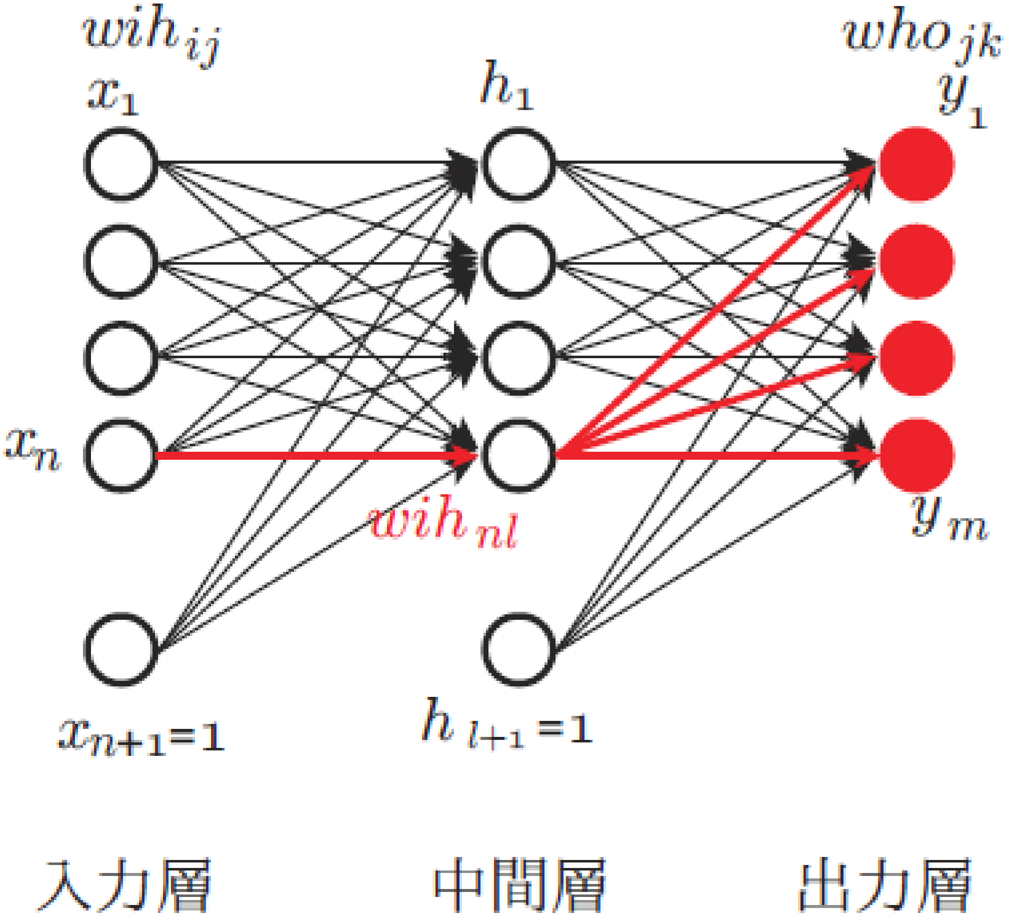
$$\boldsymbol{y}^{(p)} = \begin{pmatrix} y_1^{(p)} \\ y_2^{(p)} \\ \vdots \\ y_m^{(p)} \end{pmatrix}, \hat{\boldsymbol{y}}^{(p)} = \begin{pmatrix} \hat{y_1}^{(p)} \\ \hat{y_2}^{(p)} \\ \vdots \\ \hat{y_m}^{(p)} \end{pmatrix}$$

ルールを抽出するとは？

・電話帳からいくつかの電話番号と名前を覚える
→ 学習していない人の電話番号はわからない

・気温や湿度から海の家の売上を予測する
・過去数日の天気から次の日の天気を予測する
→ 何らかのルールに従って変動している
と考えることができるものに対して用いる

バックプロパゲーション



誤差の計算

$$
\boxed{
\begin{aligned}
E_p &= \sum_{k=1}^{m}(y_k^{(p)} - \hat{y}_k^{(p)})^2 \\
E &= \sum_{p=1}^{N} E_p
\end{aligned}
}
$$

個々の結合荷重が
誤差に与える影響

$$
\boxed{\Delta w = -\epsilon \frac{\partial E}{\partial w}}
$$

制約なし問題の最適性条件

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

$\boxed{x^* \text{が局所的最適解であるための必要条件}}$

x^* : 関数fの停留点　　　1次の必要条件

$\boxed{\text{凸関数} f \text{の任意の停留点} x^* \text{は(大域的)最適解}}$

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

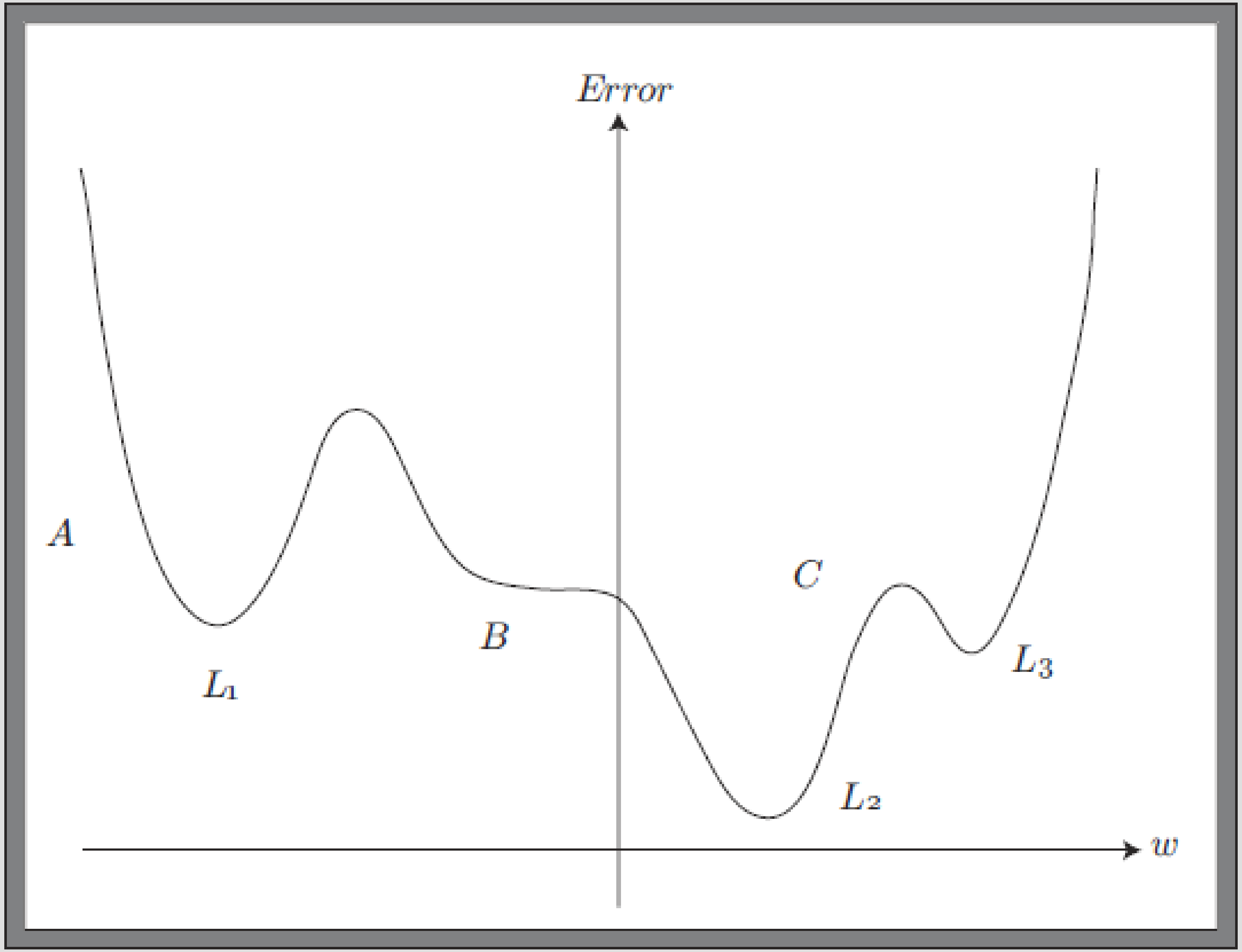
$\boldsymbol{d}^{(k)} := -\nabla f(\boldsymbol{x}^{(k)})$ とおいてステップ(2)へ.

(2) ステップ幅$\alpha^{(k)}$を求め，次の点

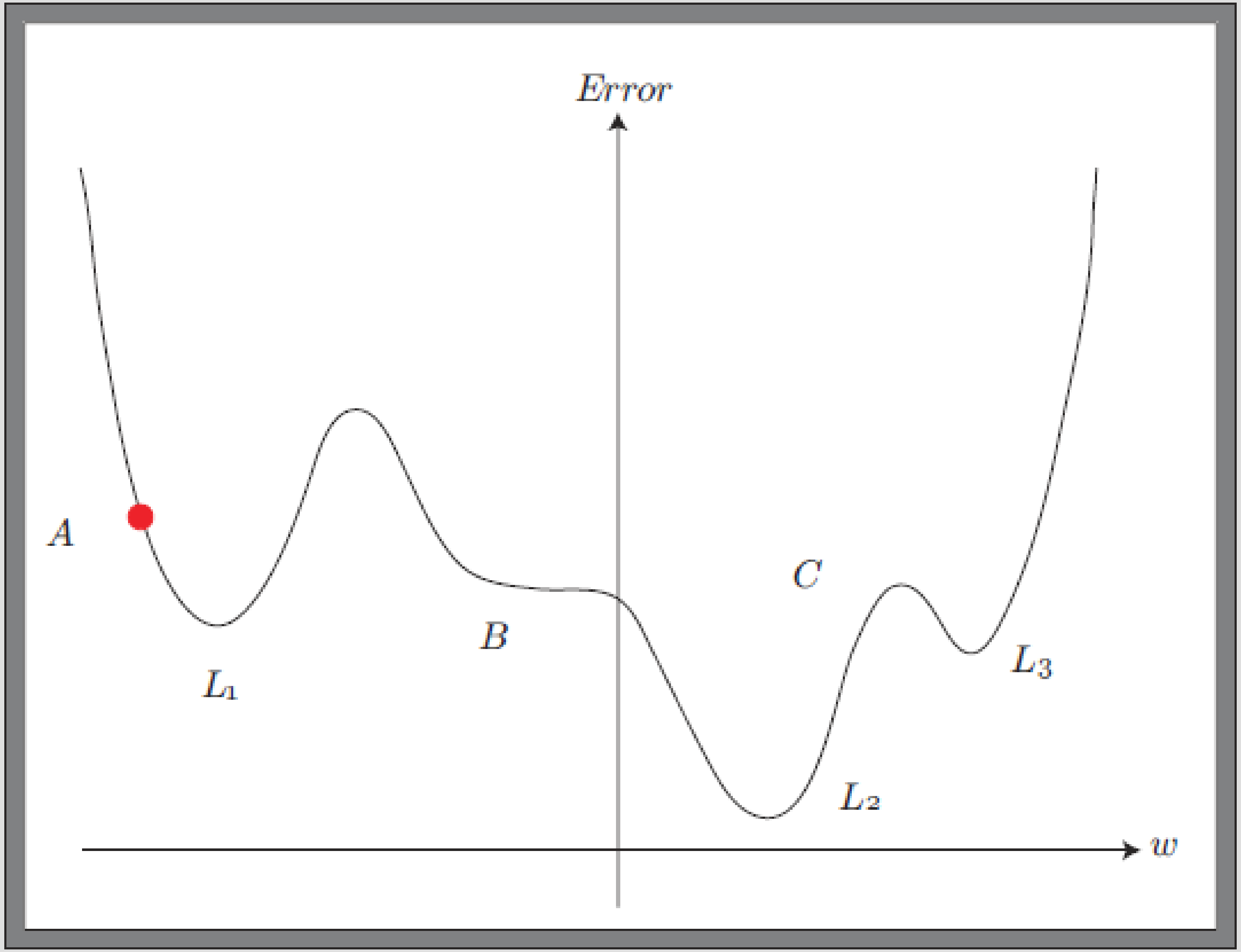
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \alpha^{(k)} \boldsymbol{d}^{(k)}$　を定める.

k:= k +1とおいてステップ(1)へ戻る.

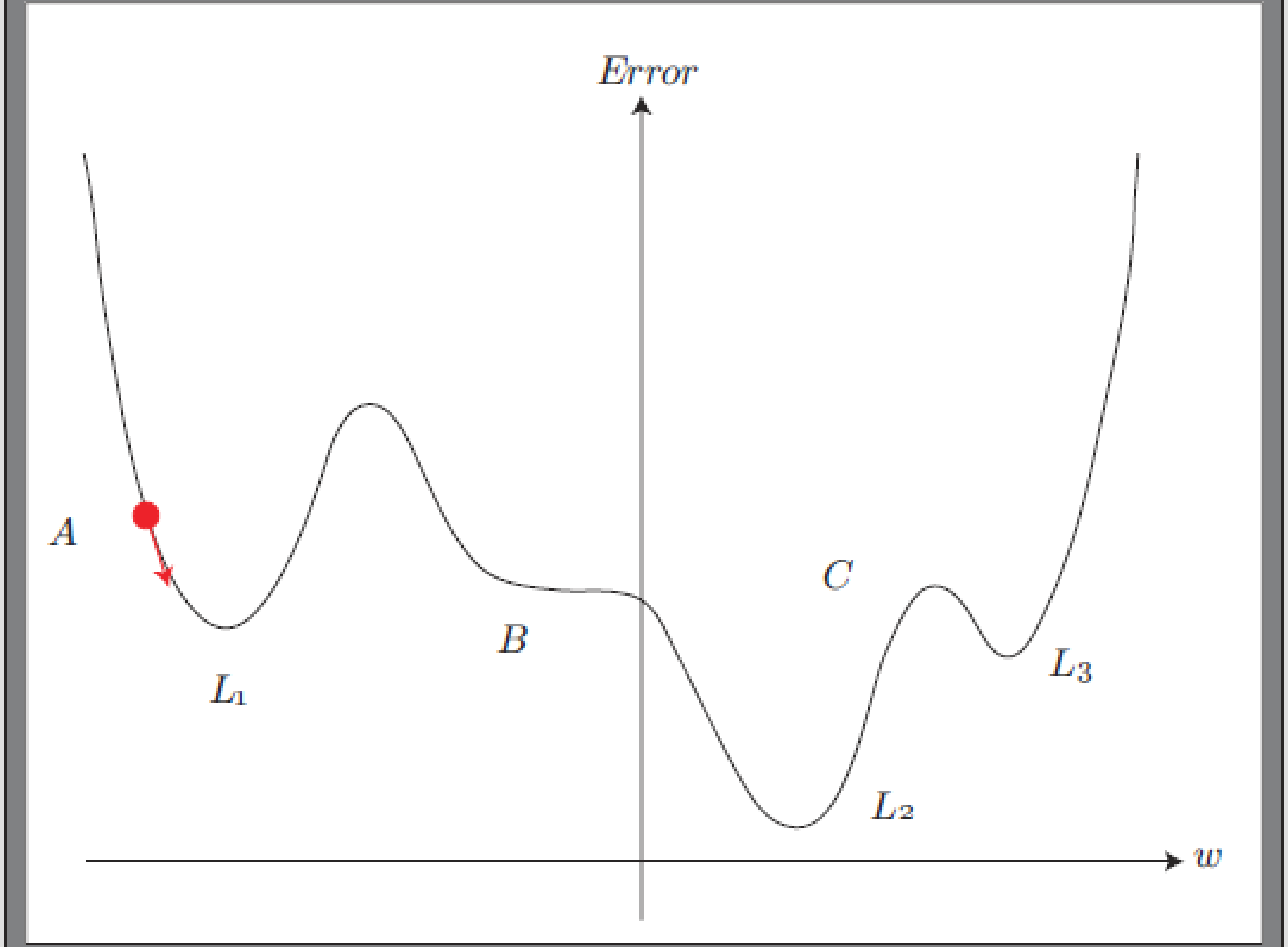
最急降下法



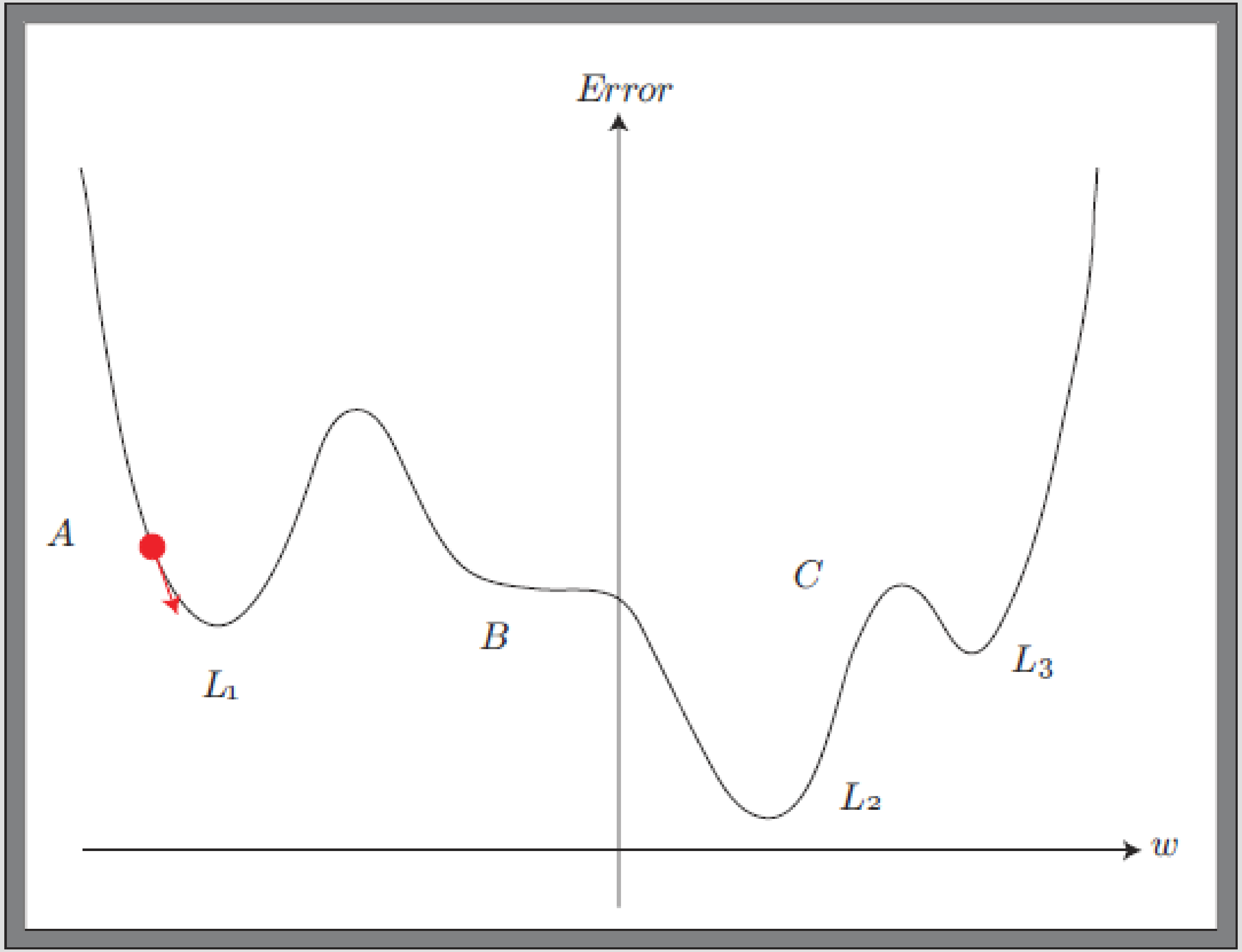
最急降下法



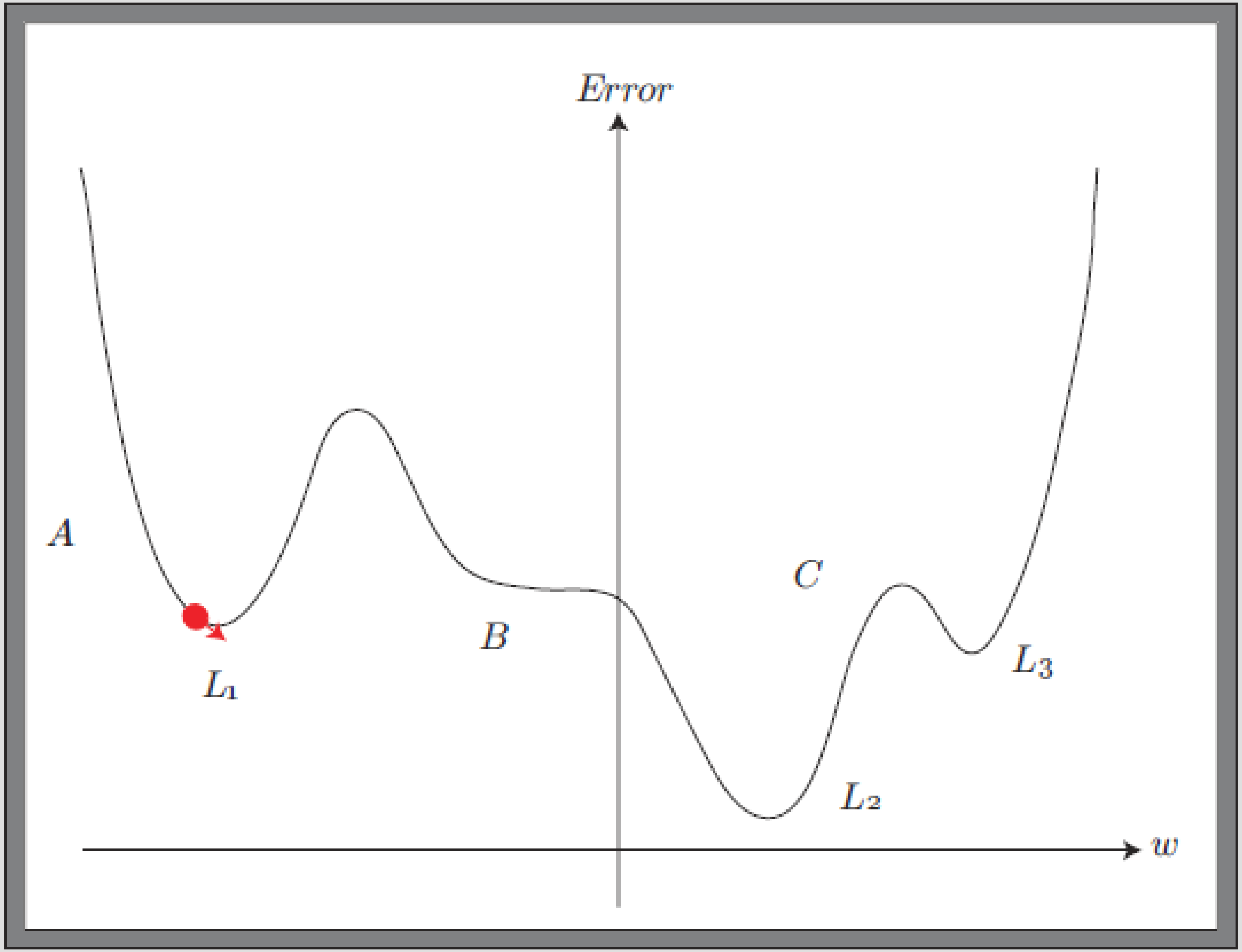
最急降下法



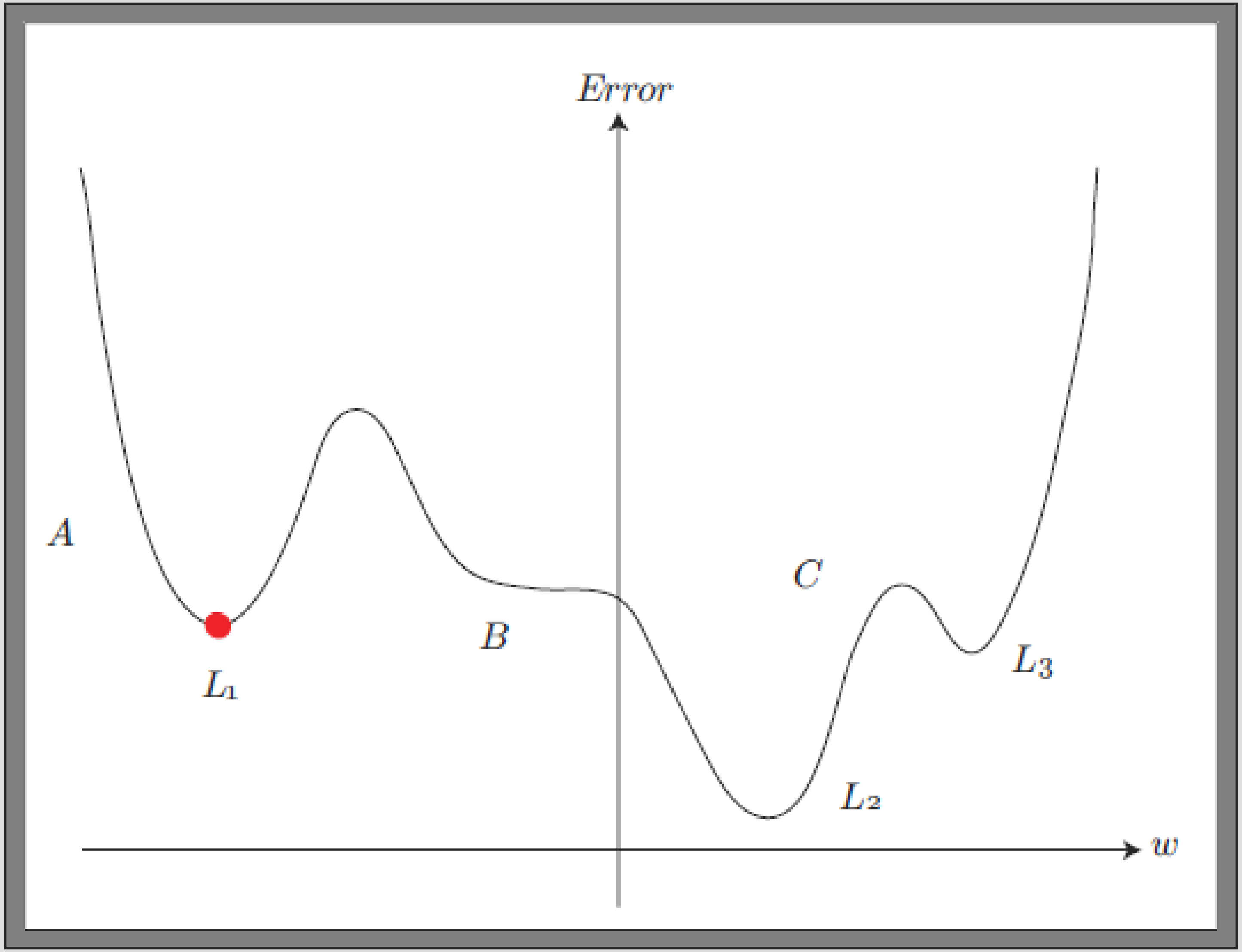
最急降下法



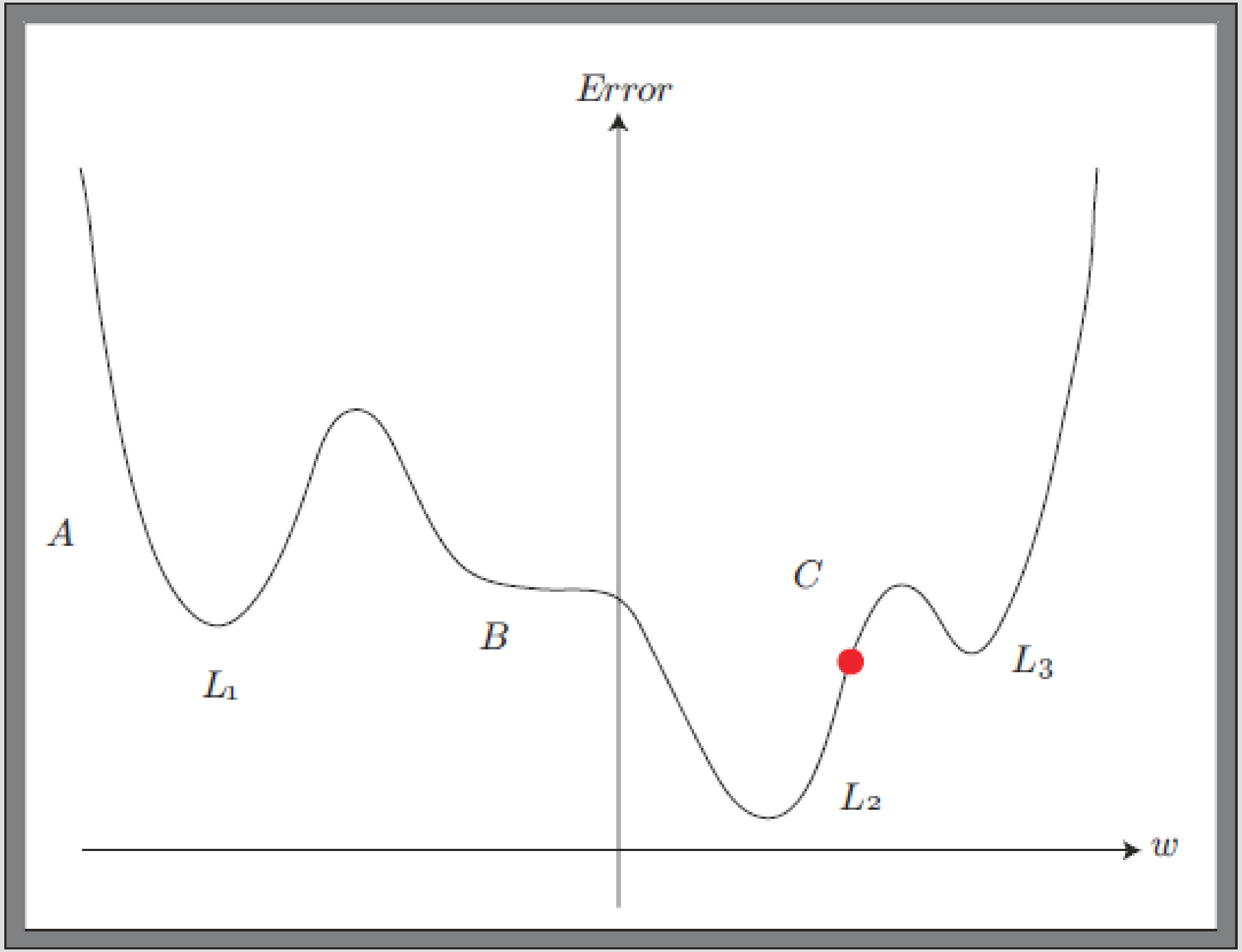
最急降下法



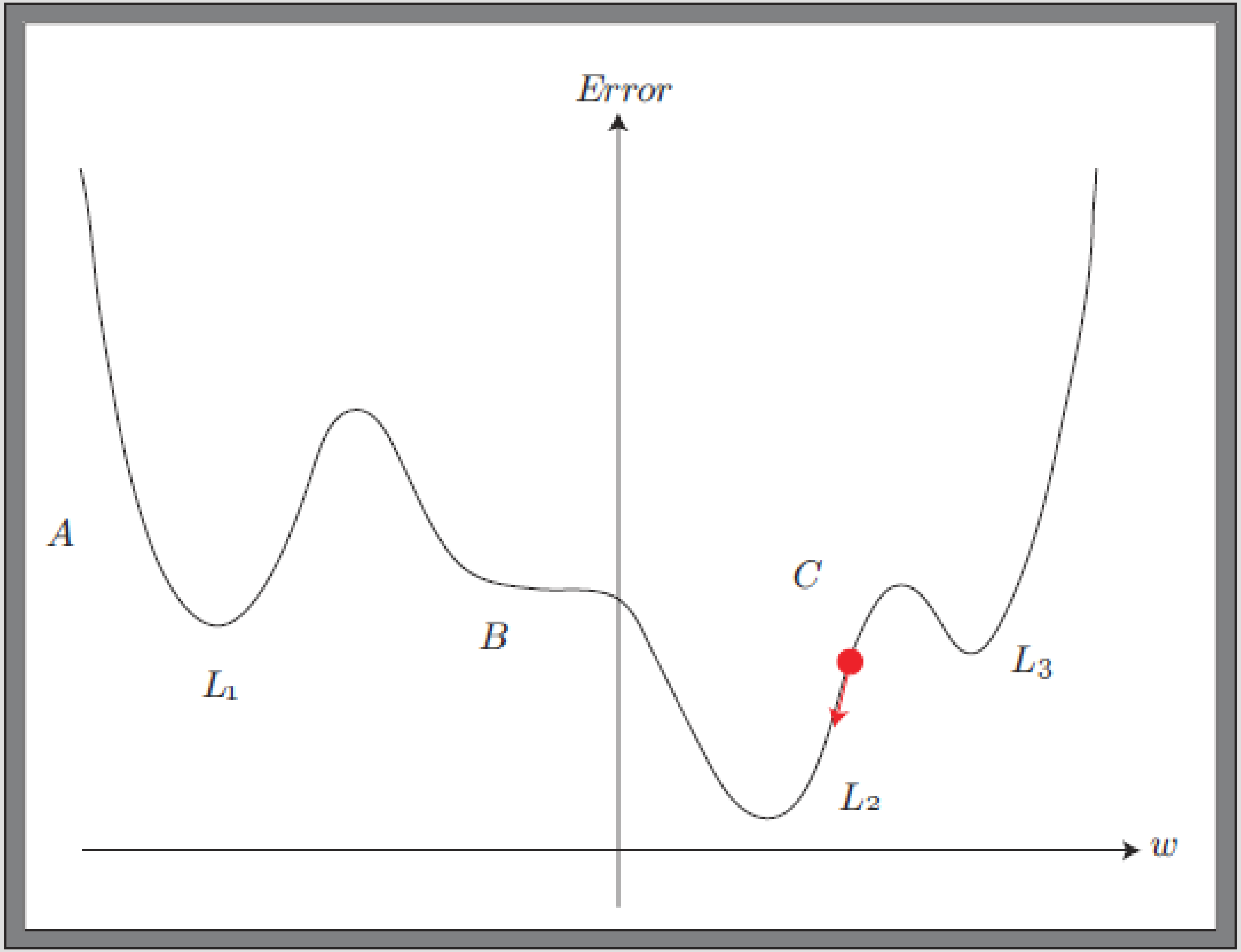
最急降下法



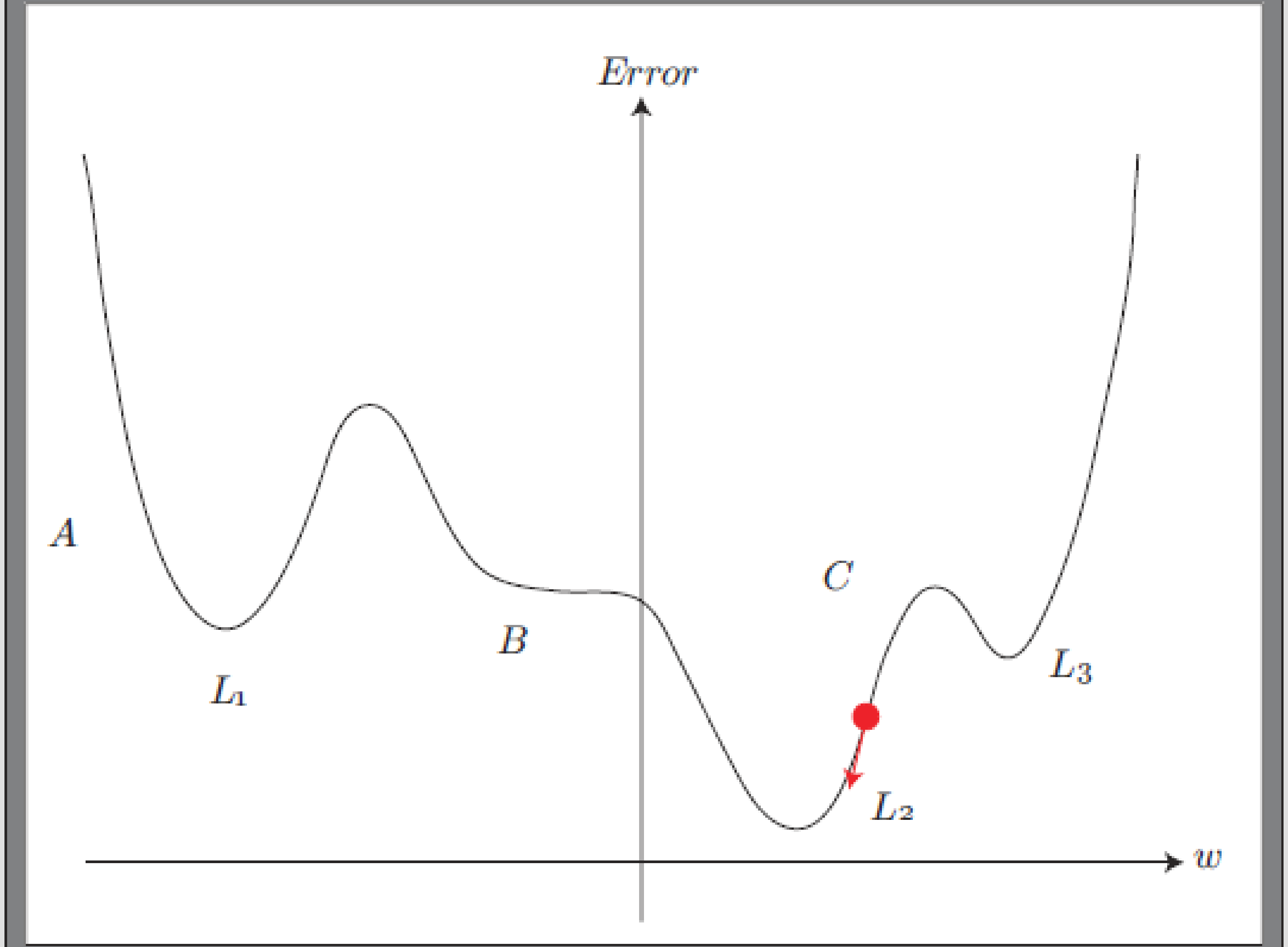
最急降下法



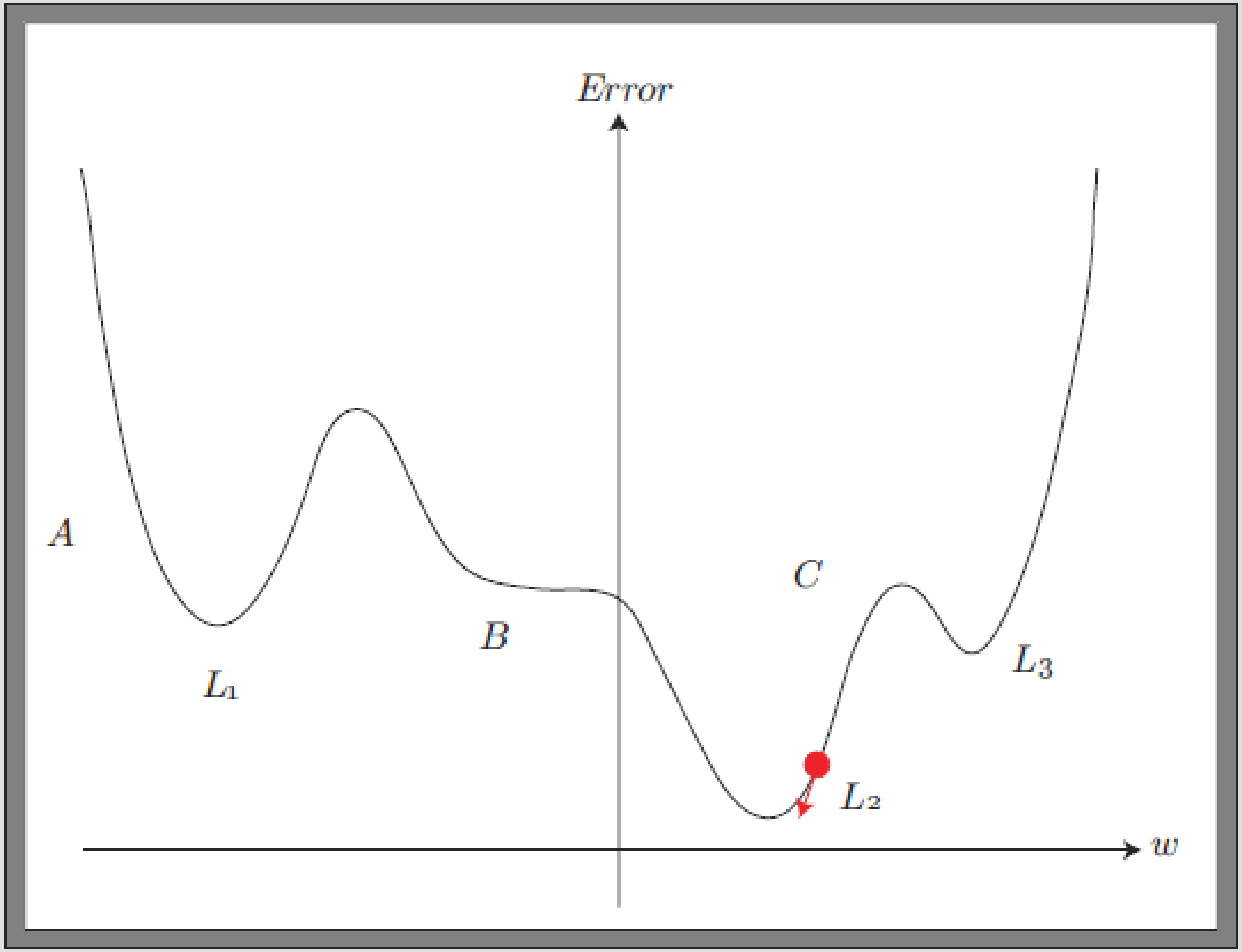
最急降下法



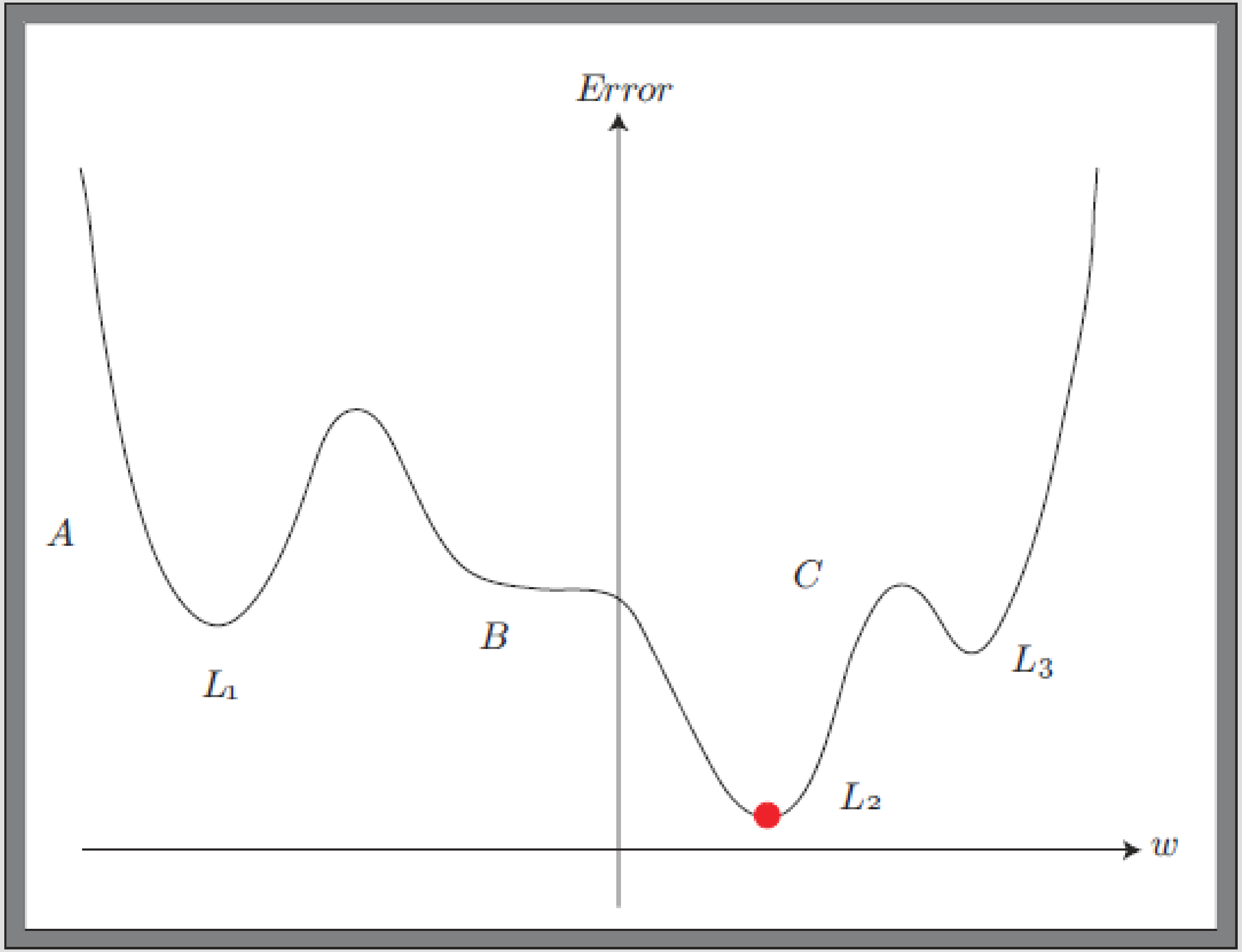
最急降下法



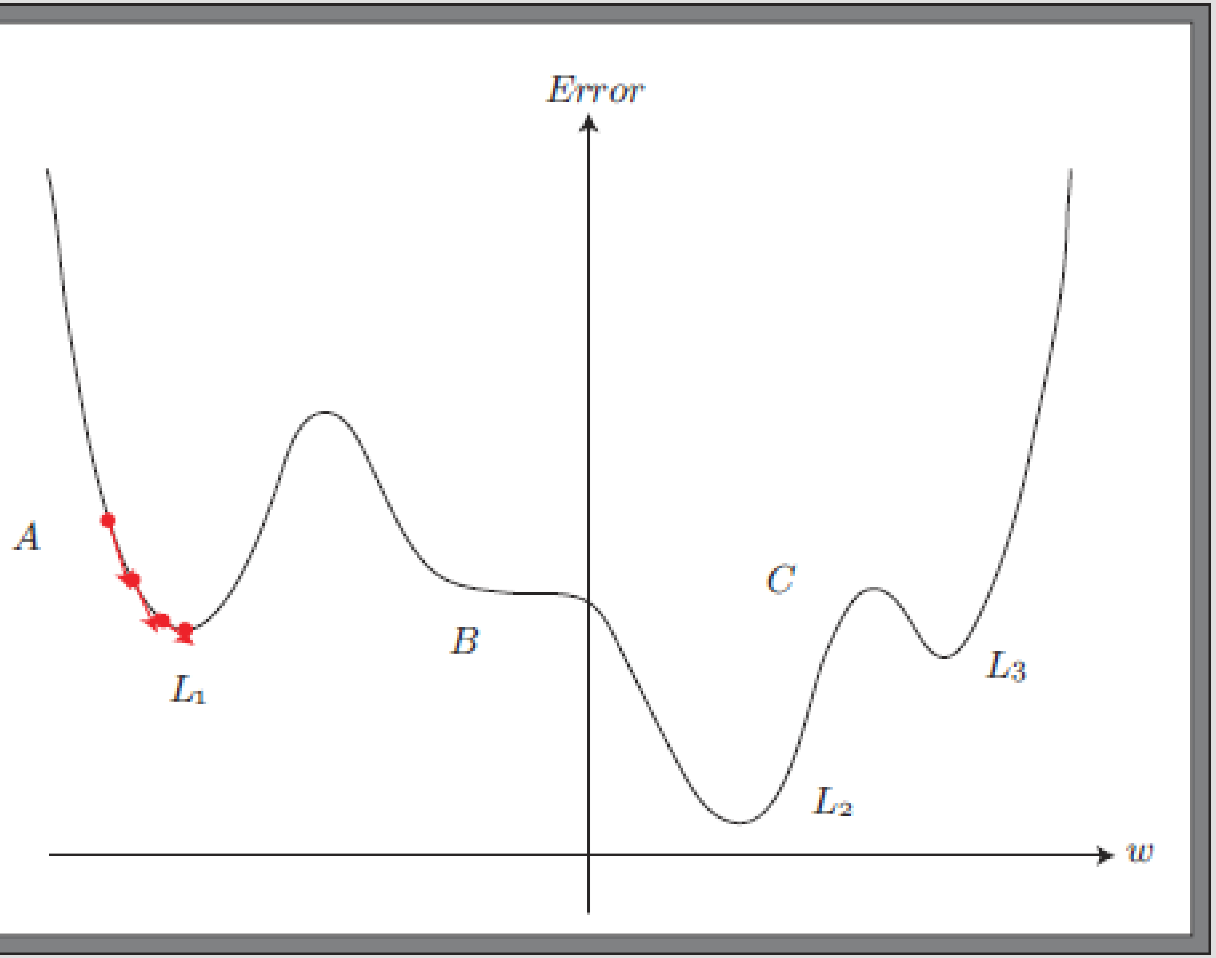
最急降下法



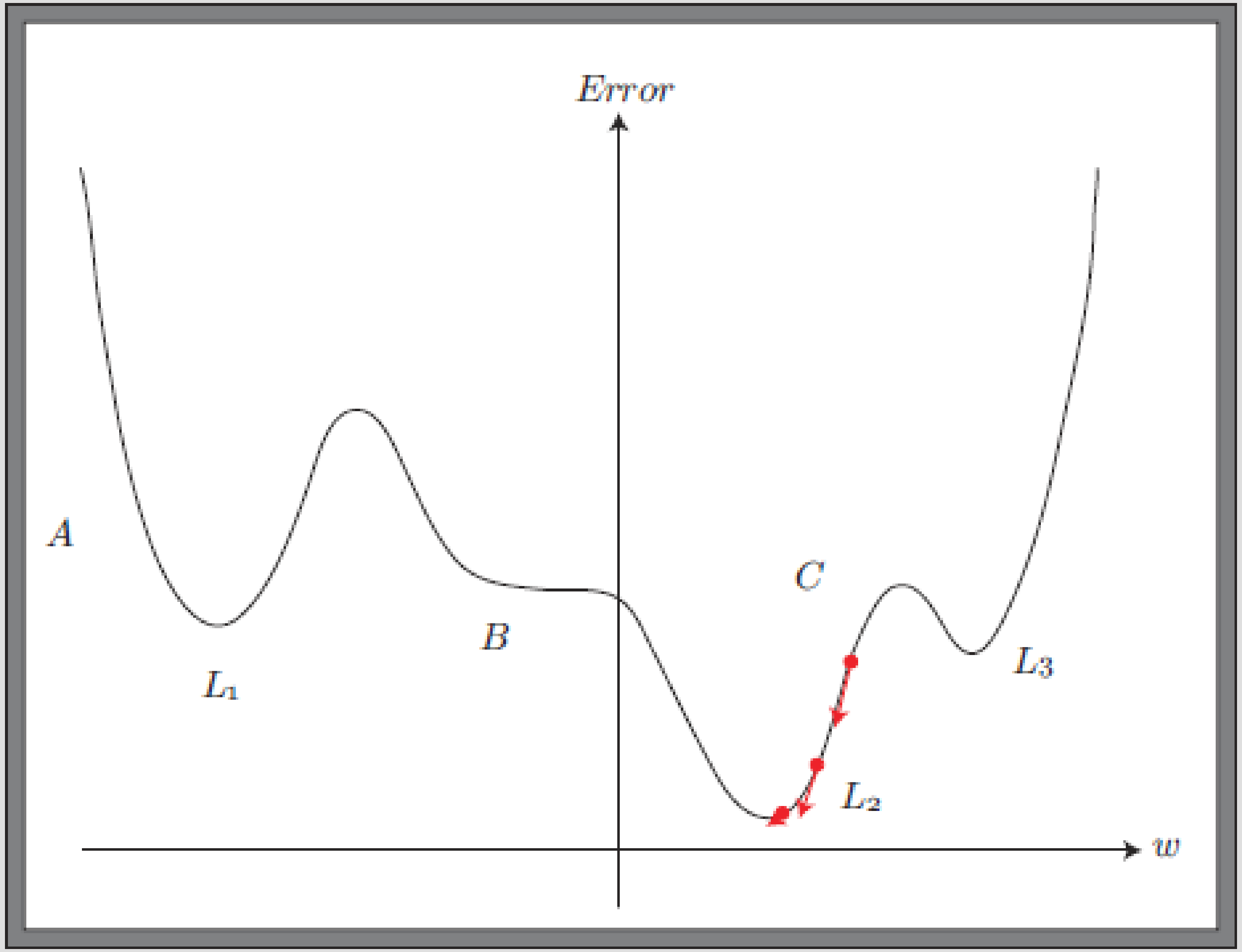
最急降下法



最急降下法

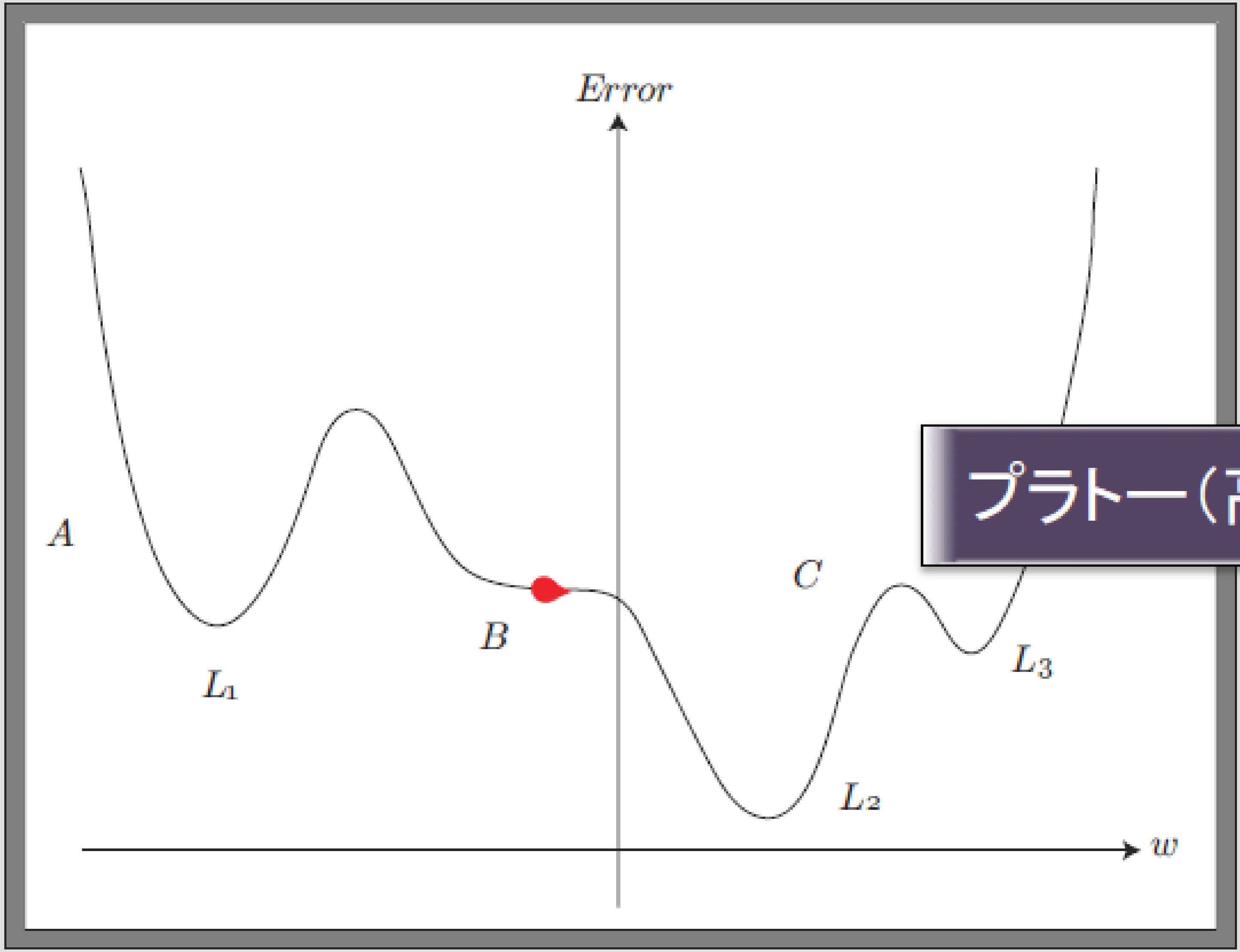


局所解（ローカルミニマム）に収束



初期値によって結果が異なる

最急降下法



プラトー（高原状態）

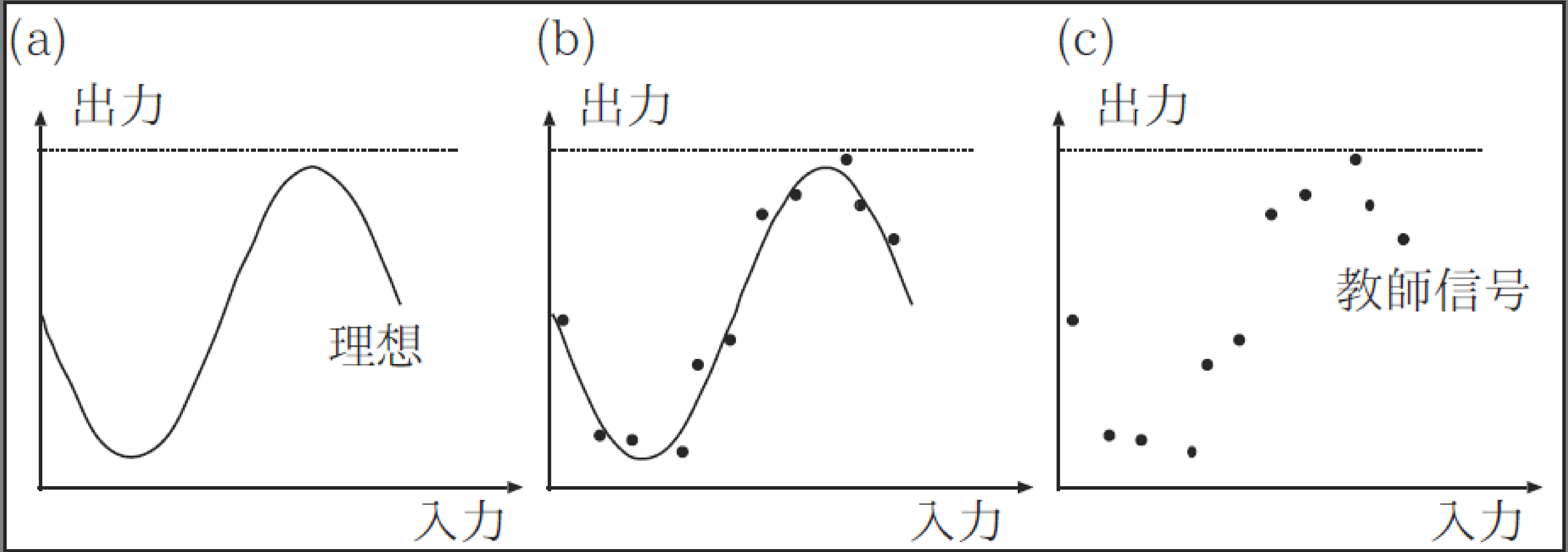
バックプロパゲーション

最急降下法とは

　結合荷重が誤差に与える影響を計算し、
　誤差が減る方向に結合荷重を修正する。

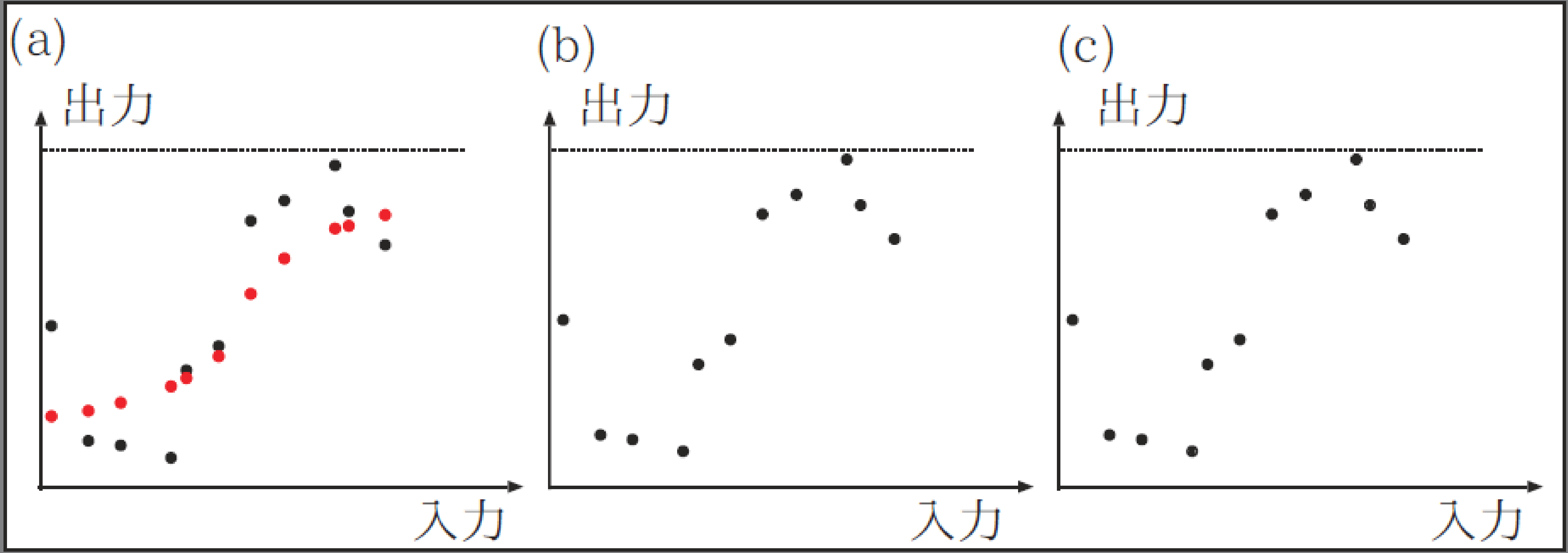
・必ずしも最小値を取るとは限らない。
・初期値によって異なった結果になることもある。
・プラトーによって学習が停滞することもある。

汎化と過学習ー教師信号

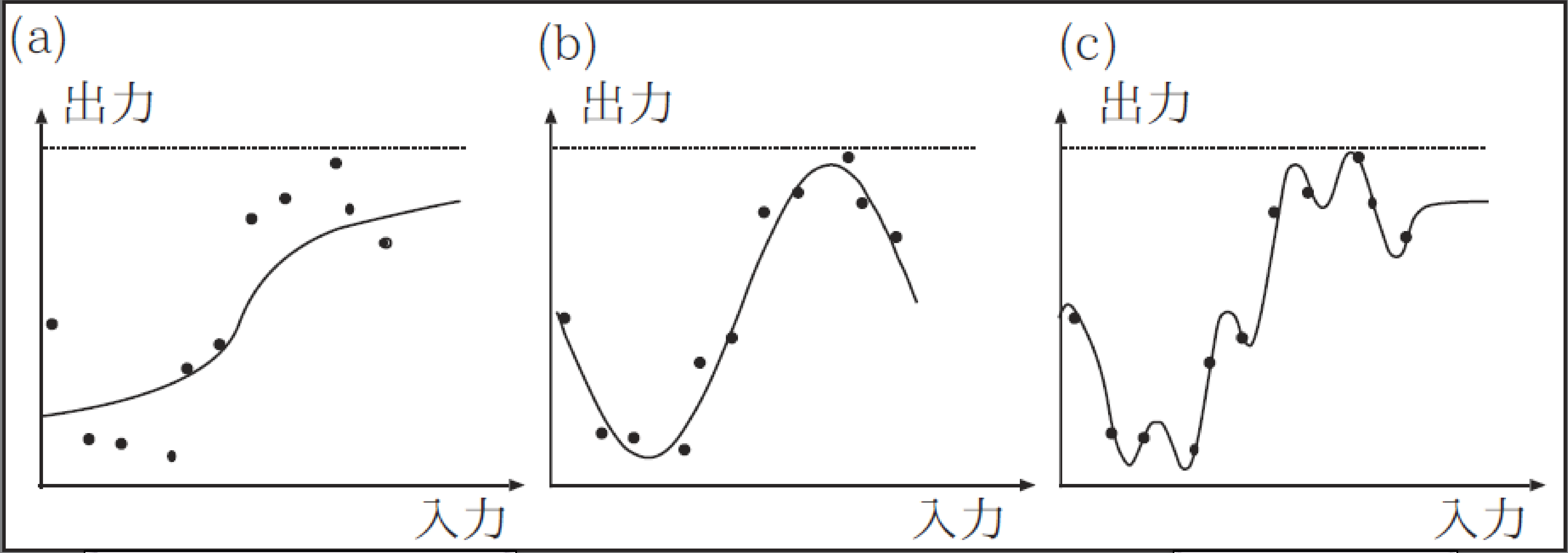


教師信号が誤差を含むことがある

汎化と過学習ー学習結果(1)



汎化と過学習—学習結果



学習が不十分

過学習

データについて

訓練用データ

input,	output
0.015,	0.500912724
0.105,	0.304927043
0.195,	0.223999883
0.285,	0.185455886
0.375,	0.301246846
0.465,	0.40991515
0.555,	0.621385785
0.645,	0.732367781
0.735,	0.854840194
0.825,	0.753472695
0.915,	0.657707115

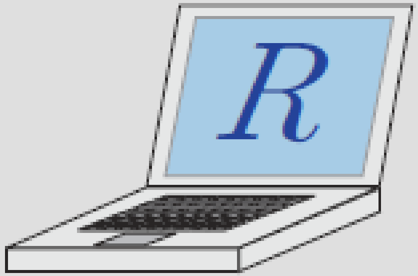
検証用

input,	output
0.025,	0.45306966
0.05,	0.407294902
0.075,	0.36380285
0.1,	0.323664424
0.125,	0.287867966
0.15,	0.257294902
0.175,	0.232698043
0.2,	0.214683045
0.225,	0.203693498
0.25,	0.2
	⋮

バックプロパゲーションによる学習

```
> train <- read.csv("sin1.csv",header=T)

> valid <- read.csv("sin2.csv",header=T)
```

```
> sin1  <- nnet(output ~ input, data=train, size=1,maxit=1000)

> yosoku1  <- predict(sin1,valid)
```



バックプロパゲーションによる学習

```
> plot ( valid, type="b", pch=16, ylim=c(0,1))
> points (train, pch=16, col="2")
```

```
>points ( valid$input, yosoku1,pch=16, col="4")
> lines ( valid$input, yosoku1)
```