前回の続き

【組合せ最適化問題の解法(1)】 まずこれらの方法で試す

●腕ずくの方法(brute force method):

コンピュータの高速な計算力に頼って、**起こりうる全ての場合を調べて**、
その中で最も良いものを選ぶ方法。(列挙法・全数探索とも呼ばれる)
〔特徴〕
◎ 原理が単純でわかりやすい
◎ 実行が可能であるならば、必ず最適解が見つかる
(組合せ最適化問題の解候補は有限な離散集合なので)
小規模問題では実用的
× 場合の数が組合せ爆発を起こす場合は実行不能

●欲張り法(greedy method):

各選択の時点で、**目先の利益が最大**になるよう選び続ける方法
〔特徴〕
◎ 原理が単純で分かりやすい
× 局所的な目先の最適化を行い続けることが全体の最適化になるとは
一般にはいえない…最適解が見つかる保障は一般にはない
◎ 多くの場合、最適ではないがそれに近いものが求まる
◎ グラフ最短経路探索問題など一部の問題では、最適解が得られる
(ダイクストラの方法) また、後述の動的計画法とも関連

【例】 ナップサック問題に対する欲張り法

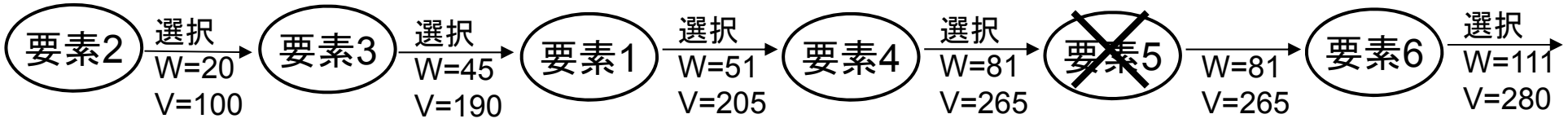
ナップサックの容量： 112

要素	1	2	3	4	5	6	7	8
重さ	6	20	25	30	40	30	60	10
価値	15	100	90	60	40	15	10	3
単位重さあたりの価値	2.5	5	3.6	2	1	0.5	0.166	0.3

（1）要素を**単位重さあたりの価値の高い順に並べ替える**。 i ＝1 とする。

（2）もし i 番目の要素を選択したら、選択した要素全体の重さがナップサック容量を超えない場合はその i 番目の要素を選択する。さもなければその要素を選択しない。

（3）i が要素数 n と等しくなったら終了。さもなければ i を i+1 として（2）へもどる。

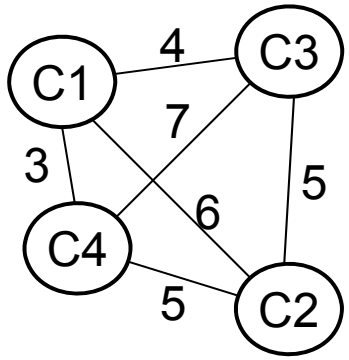


要素数10, 容量100, 価値上限100の問題で欲張り法により最適解を得る割合16～17%　だが最適解と欲張り法の解の違いは1割程度

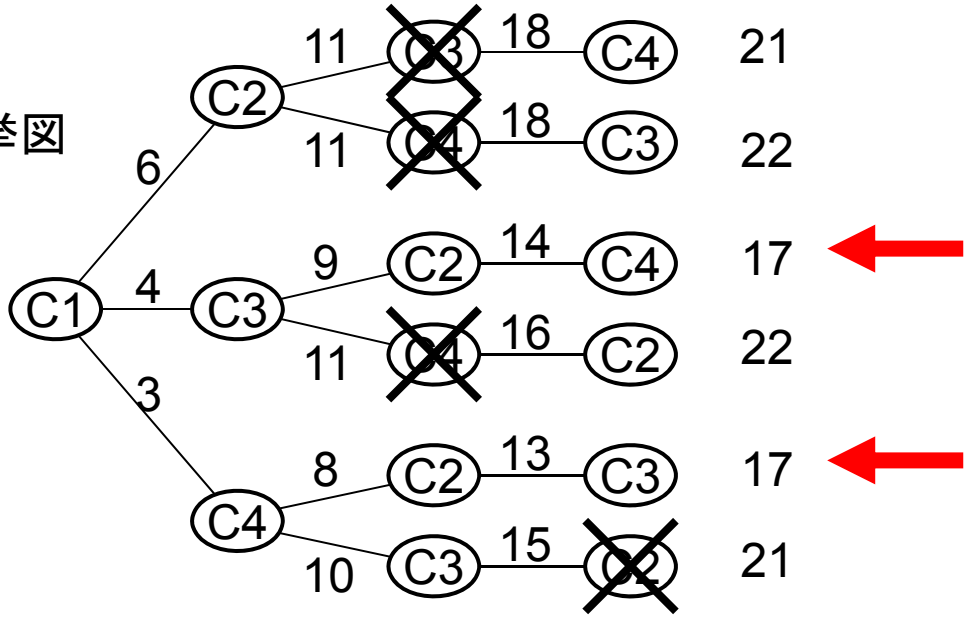
【組合せ最適化の解法(2)】　分枝限定法

全ての解候補を列挙して，その中で評価値最大(最小)のものを見出す：腕ずくの方法

例） 4都市TSP



列挙図



問題の規模が大きくなると、組合せ爆発のため実行不能！そこで…

→ **列挙図の木の生成途中で、**

分枝限定法 (branch and bound method)

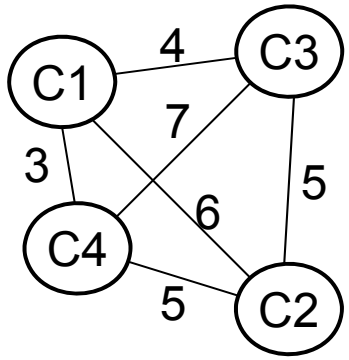
◎…「腕ずく法」と「欲張り法」の中間的なバランスの良い方法

△…**解の可能性が無い枝だけを刈れば最適解が保障**されるが、
大規模問題では実行不能になるので、最適解をあきらめて可能性の小さい枝を刈ることが多い

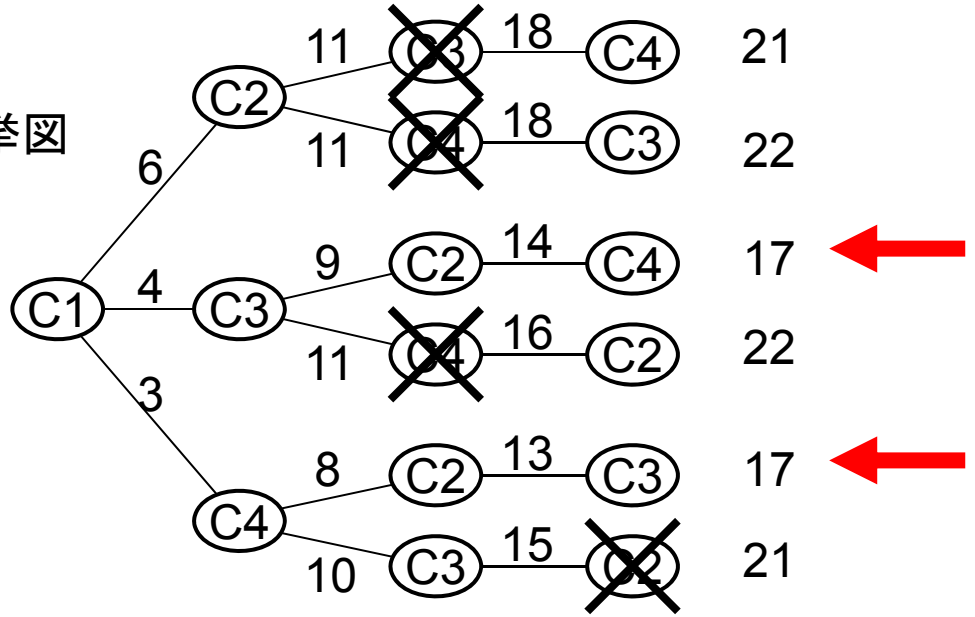
【組合せ最適化の解法(2)】　分枝限定法

全ての解候補を列挙して，その中で評価値最大(最小)のものを見出す:腕ずくの方法

例） 4都市TSP



列挙図



問題の規模が大きくなると、組合せ爆発のため実行不能！そこで…

→ **列挙図の木の生成途中で、解となる可能性の無い(または小さい)枝を刈る**

分枝限定法 (branch and bound method)

◎…「腕ずく法」と「欲張り法」の中間的なバランスの良い方法

△…**解の可能性が無い枝だけを刈れば最適解が保障**されるが、
大規模問題では実行不能になるので、最適解をあきらめて可能性の小さい枝を刈ることが多い

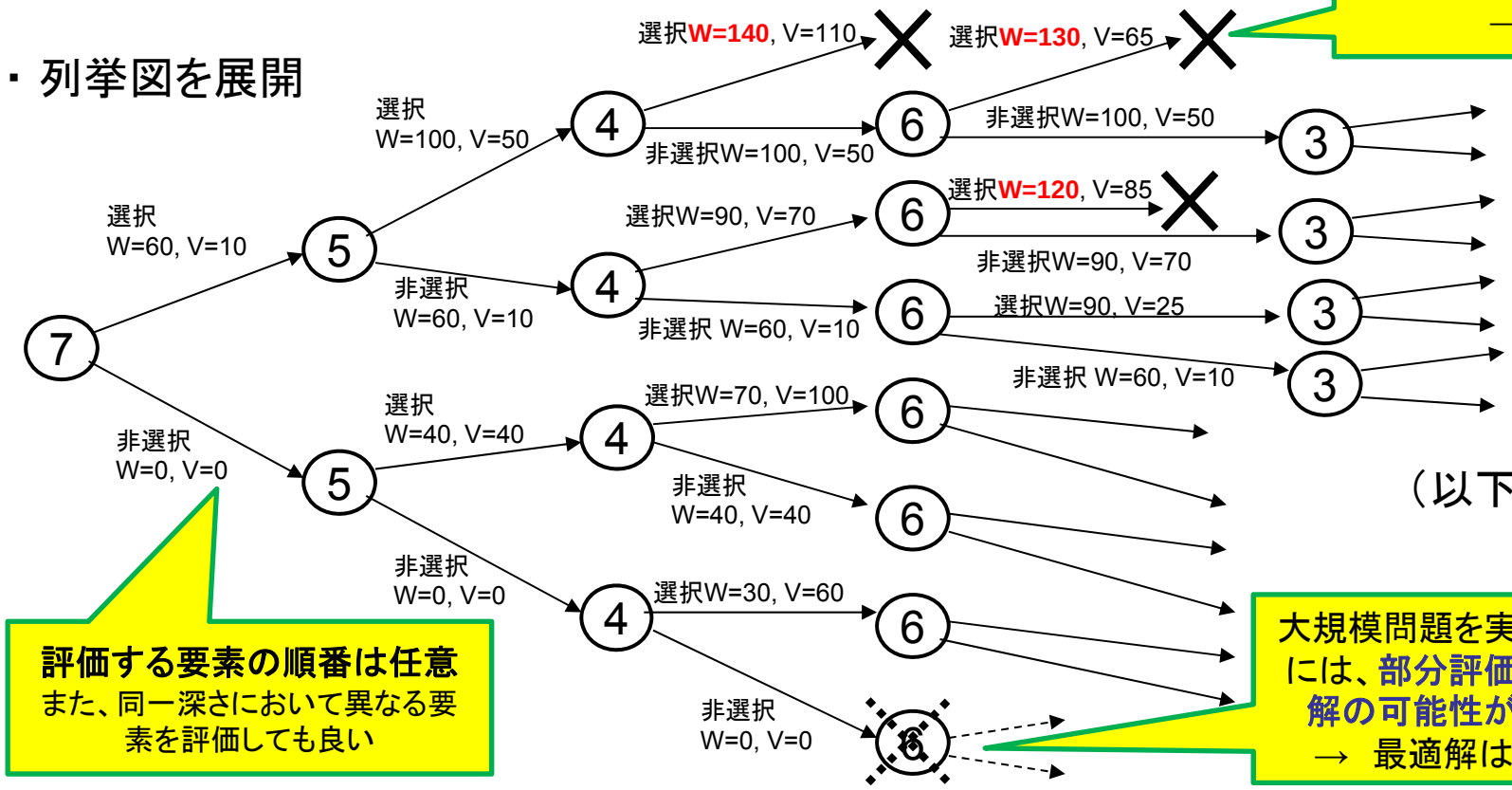
【例】 ナップサック問題に対する分枝限定法

ナップサックの容量W_{max} = 112

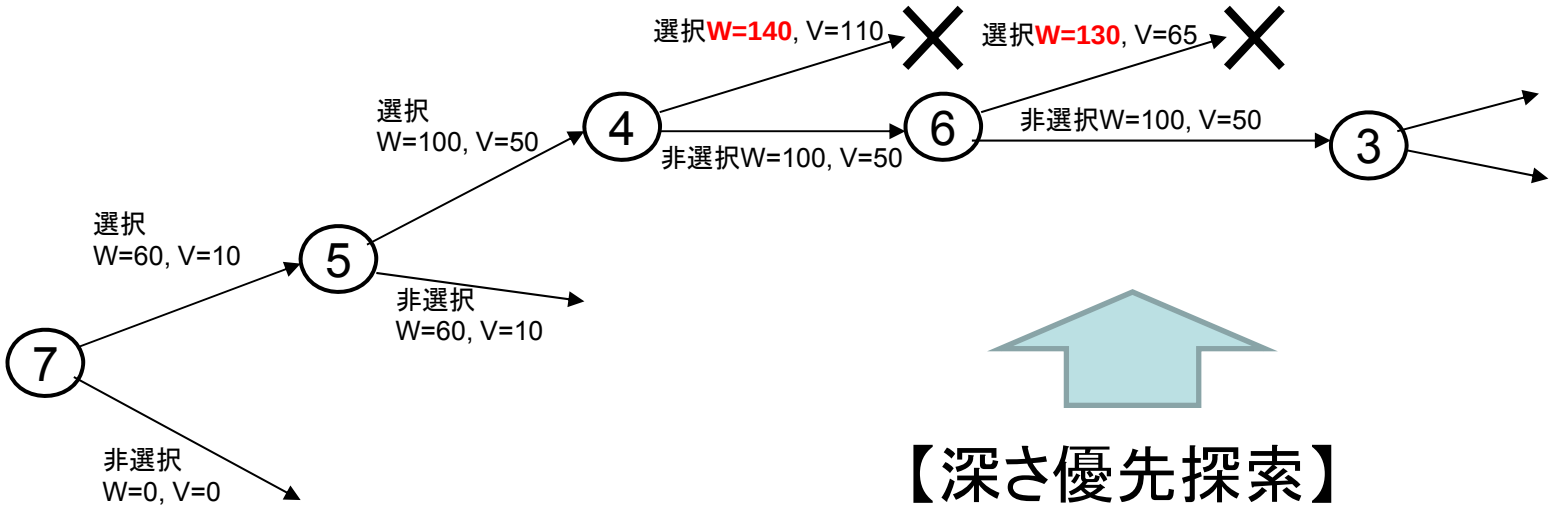
要素	1	2	3	4	5	6	7	8
重さ	6	20	25	30	40	30	60	10
価値	15	100	90	60	40	15	10	3

・腕づくの方法（全数探索）の場合、検討する解候補数は2^8＝256

・列挙図を展開



分枝限定法における枝の展開:「深さ優先探索」と「幅優先探索」



【深さ優先探索】

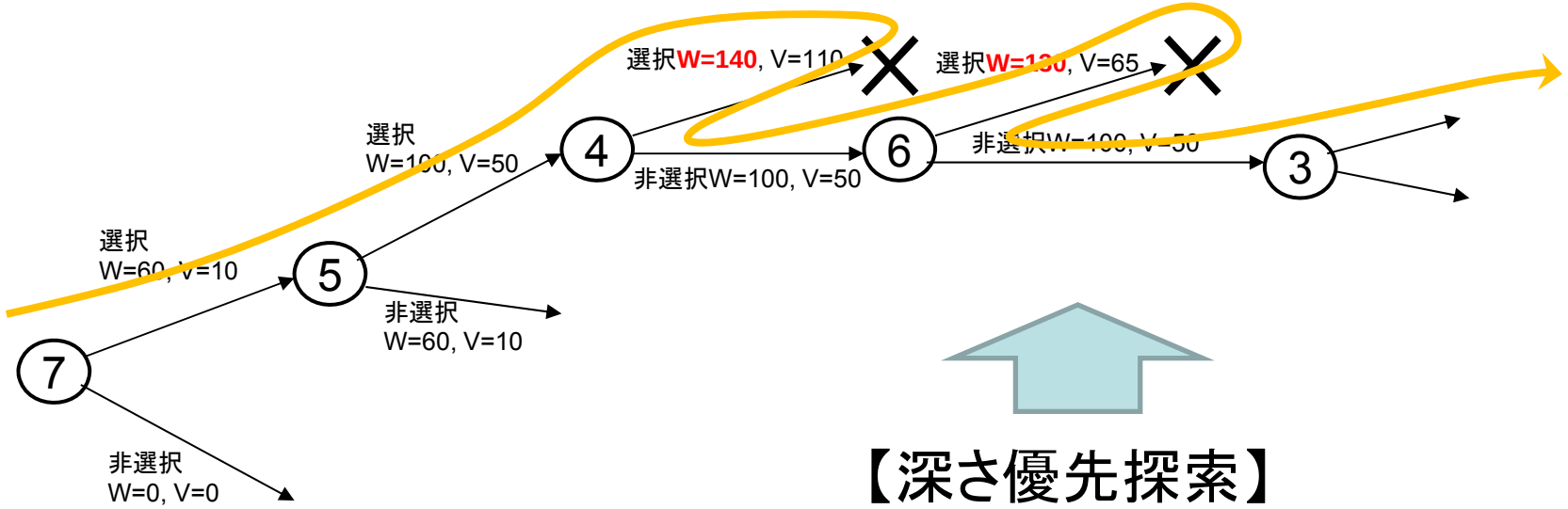
探索の各時点で、最大の深さの部分問題から優先的に探索する

特徴

○ 実行可能解を素早く見つける

○ 探索の過程で必要となる記憶容量は小さい

分枝限定法における枝の展開:「深さ優先探索」と「幅優先探索」



【深さ優先探索】

探索の各時点で、最大の深さの部分問題から優先的に探索する

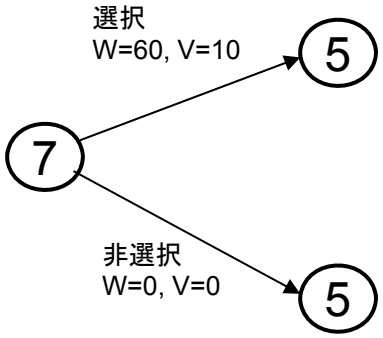
特徴

○ **実行可能解**を素早く見つける

○ 探索の過程で必要となる記憶容量は小さい

分枝限定法における枝の展開:「深さ優先探索」と「幅優先探索」

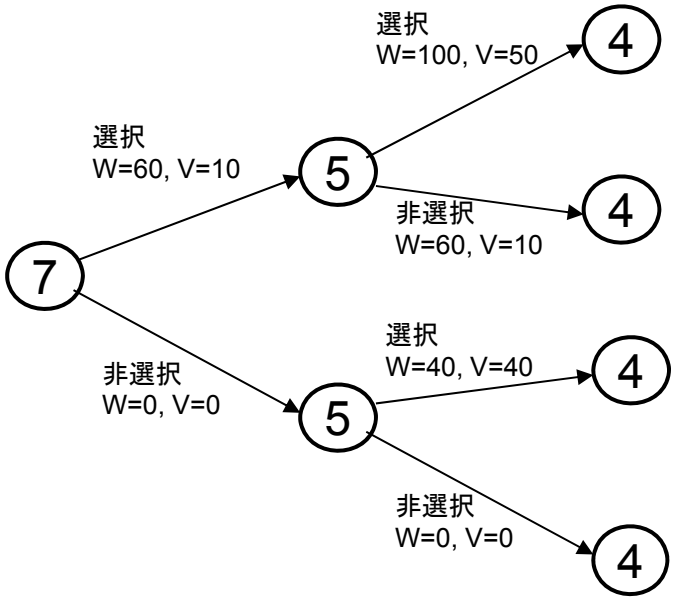
【幅優先探索】



探索の各時点で、
深さが最小の部分問題
から優先的に探索する

分枝限定法における枝の展開:「深さ優先探索」と「幅優先探索」

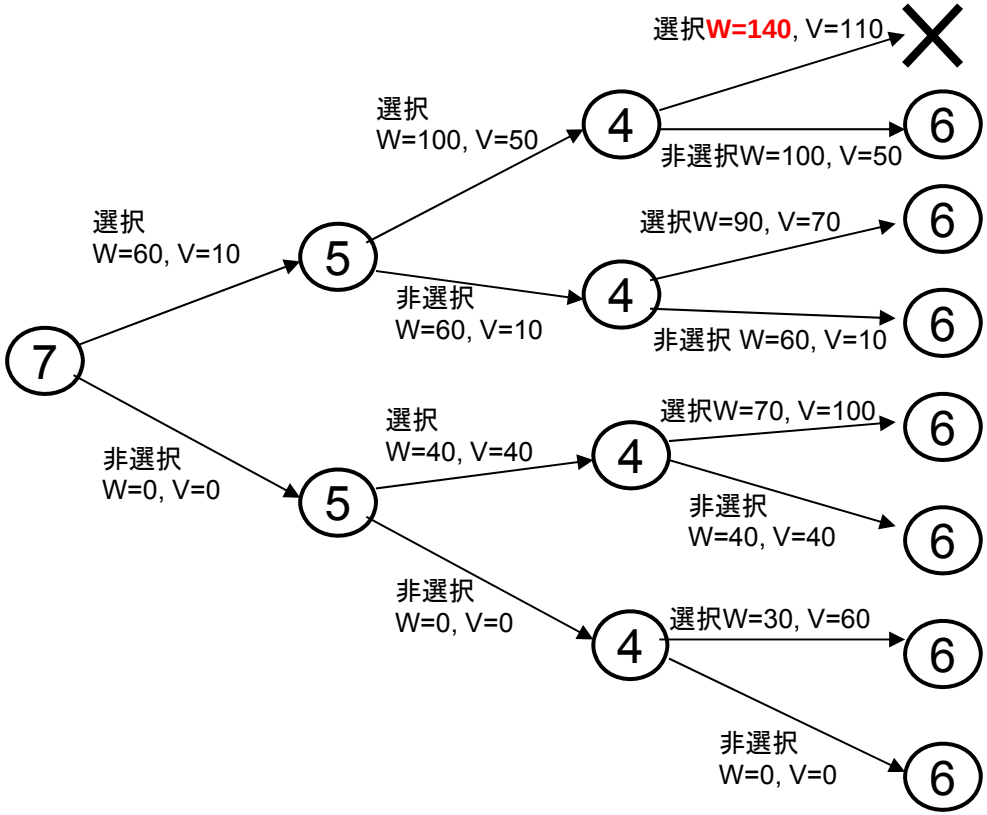
【幅優先探索】



探索の各時点で、
深さが最小の部分問題
から優先的に探索する

分枝限定法における枝の展開:「深さ優先探索」と「幅優先探索」

【幅優先探索】



探索の各時点で、
深さが最小の部分問題
から優先的に探索する

経路探索アルゴリズム「ダイクストラ法」は、
「深さ＝コスト」とした幅優先探索の一種でもある

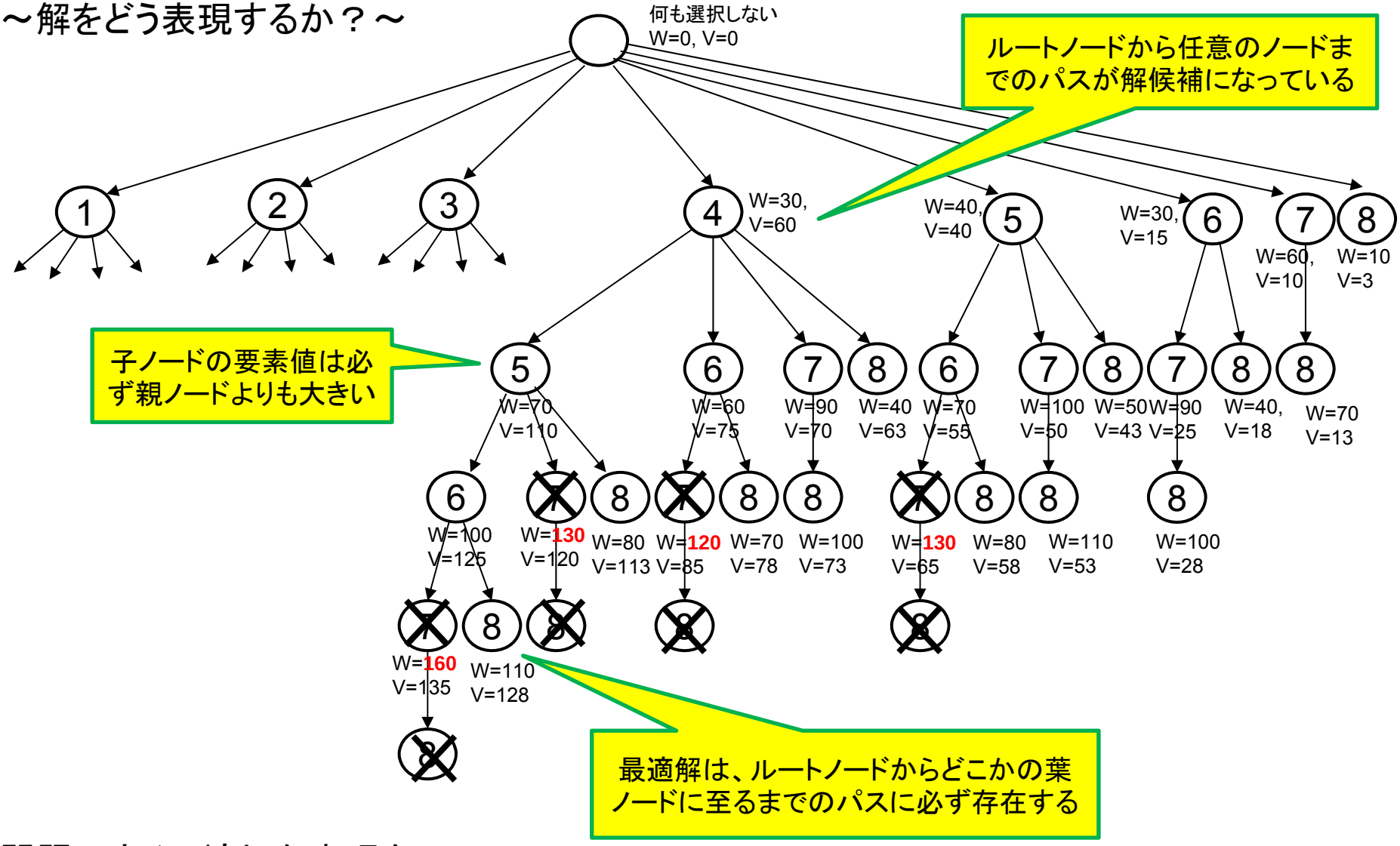
特徴

○ **最適解**を素早く見つける

X 大きな記憶容量を要す

分枝限定法における様々な「列挙図」表現

～解をどう表現するか？～



問題に応じて適切な表現を
選択・考案せよ

まとめ

組合わせ最適化　と　分枝限定法

【分枝限定法とは？】

1）解候補をツリー状の列挙図のように列挙・評価していくが、ツリー生成途中で
　　解の可能性の無い（あるいは可能性が低い）枝を刈ることで探索空間を減らす
2）解の可能性の無い枝だけを刈れば最適解を得る

【深さ優先探索】と【幅優先探索】

・分枝限定法における枝の展開方法　…　深さ優先のほうが省メモリ

【解候補および列挙図の表現方法】

・問題に応じて（あるいは好みに応じて）適切な表現を選択・考案せよ
　→　実問題においては、この「表現方法」を見つけることこそエンジニアの仕事

【レポート課題】　2017.10.27

下記のナップサック問題を「欲張り法」および「分枝限定法」を用いてそれぞれ解を求めよ。

レポートにはそれぞれの方法での導出過程を明記すること。

ナップサック容量：100

要素1	weight =32	value=67
要素2	weight =22	value=40
要素3	weight =41	value=93
要素4	weight =44	value=57
要素5	weight = 7	value=25
要素6	weight =47	value=97
要素7	weight =10	value= 2
要素8	weight = 1	value=10
要素9	weight =15	value= 6
要素10	weight = 5	value=17

【提出期限】 2017年11月10日（金）午後5時
【提出先】　講義後木村に直接手渡し
　　または　W2号館6階634号室

九州大学 工学部地球環境工学科
船舶海洋システム工学コース

システム設計工学 （担当：木村）

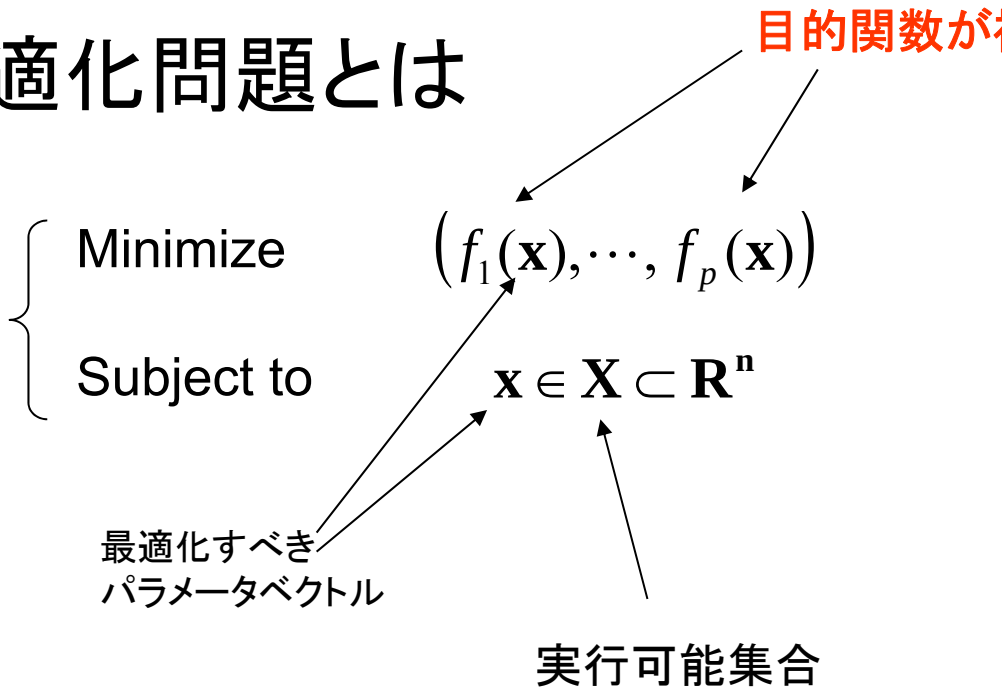
(8) 遺伝的アルゴリズムによる
多目的最適化（**MOGA**）

場所： 船1講義室

授業の資料等は

http://sysplan.nams.kyushu-u.ac.jp/gen/index.html

多目的最適化問題とは



実際の設計問題のほとんどは多目的最適化になっている

設計案の評価項目が複数あり、それらを**単純に足し合わせることができず**、
しかも**トレードオフ比が一意に決められない**問題

例1） 配管設計問題：設計変数＝管の配置、目的関数＝管路長・操作性・メンテナンス性
例2） レンズ系設計問題：設計変数＝レンズ曲率、目的関数＝ピント・歪曲・色収差

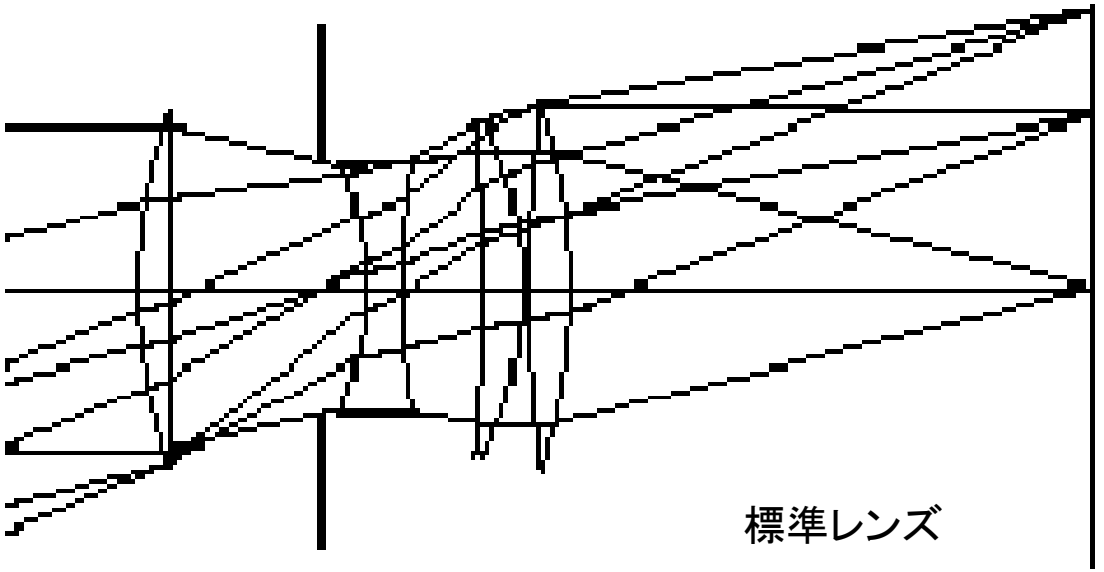
【準備】

$\mathbf{a}=(a_1,\cdots,a_p)^T$, $\mathbf{b}=(b_1,\cdots,b_p)^T\in\mathbf{R}^p$ のベクトルに対し、以下の不等号を定義：

$$\mathbf{a}\le\mathbf{b} \quad \overset{def}{\longleftrightarrow} \quad a_i\le b_i,\quad \forall i=1,\cdots,p_i \quad \text{かつ} \quad \mathbf{a}\neq\mathbf{b}$$

多目的最適化の例：レンズ系設計問題

例）4枚組み
レンズ系



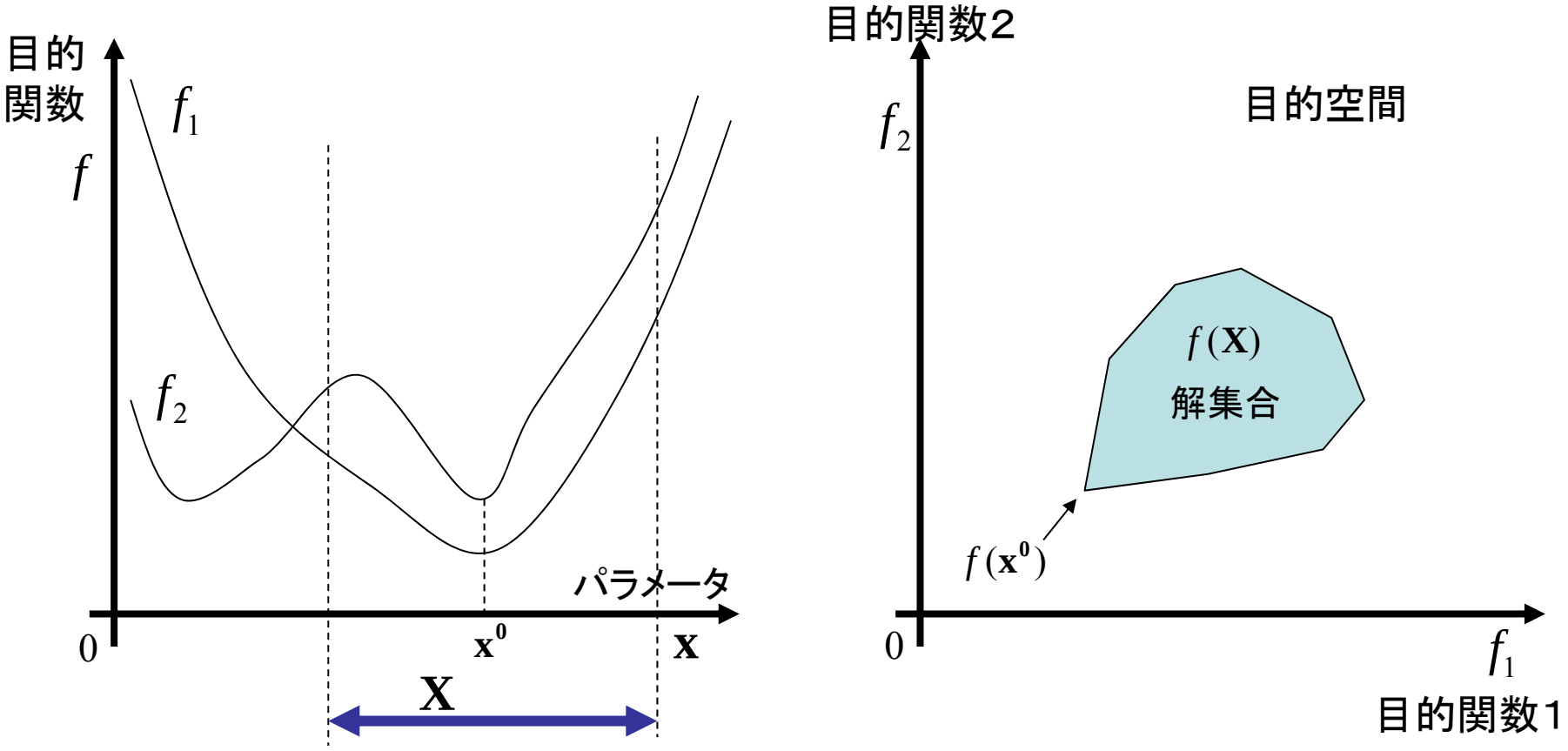
●レンズ系を特徴付けるパラメータ：
各レンズの**曲率**、**レンズ間距離**など（数十個） → **設計変数** $\mathbf{x}$

●レンズ系の評価：
（1）歪曲（distortion） f_1
（2）焦点（focus） f_2
（3）色収差（色のにじみ） f_3
（4）その他（像の明るさ、製造コスト、丈夫さなど）

同時に多様な評価を満足させなければならない

多目的

多目的最適化問題の完全最適解

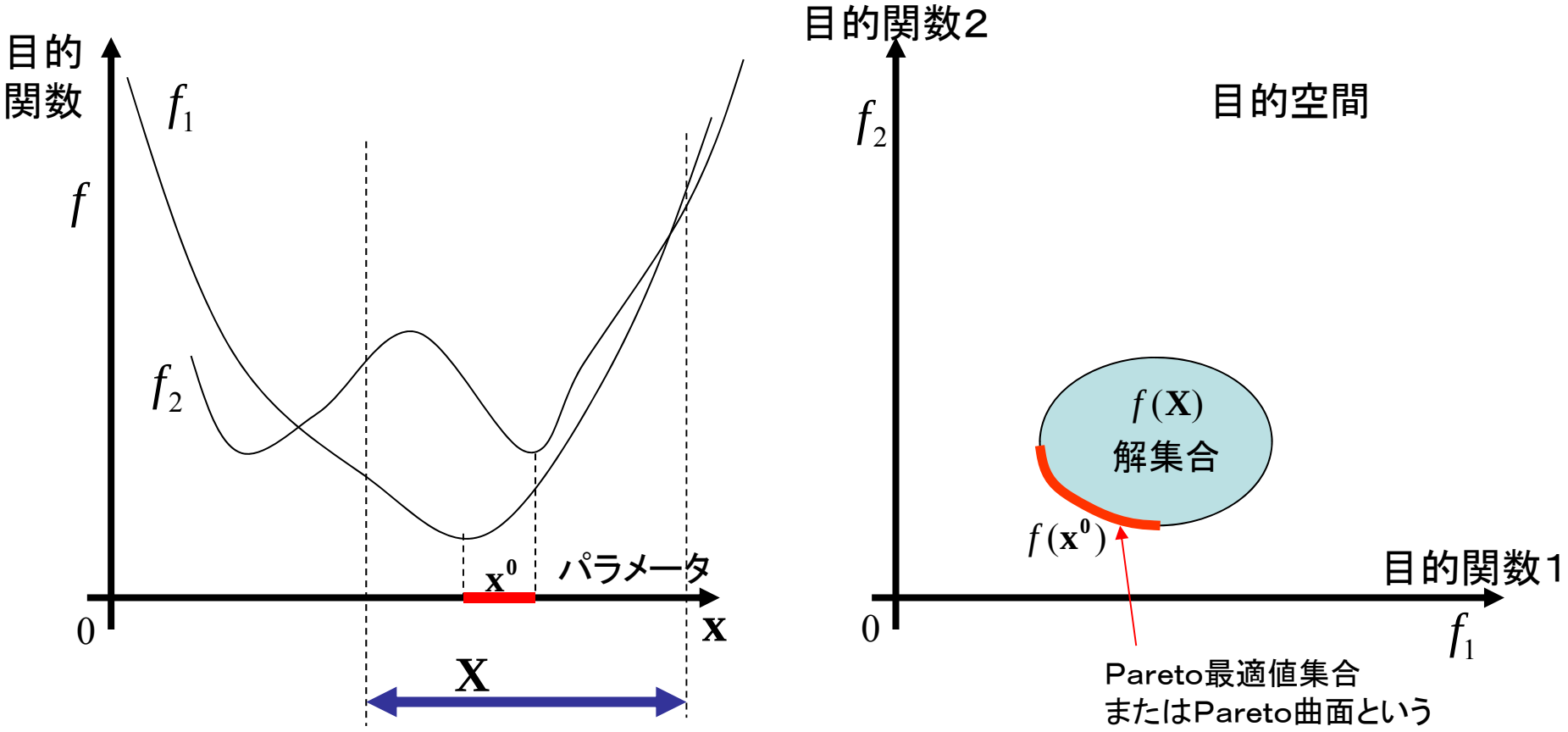


【定義】 多目的最適化問題(P)において、$\mathbf{x}^0 \in \mathbf{X}$ が以下の条件を満足するとき、$\mathbf{x}^0$を(P)の**完全最適解**という。

$$\forall \mathbf{x} \in \mathbf{X}: \quad f(\mathbf{x}^0) \leq f(\mathbf{x})$$

ただし、完全最適解が存在するケースは、実際には稀

多目的最適化問題のパレート解集合



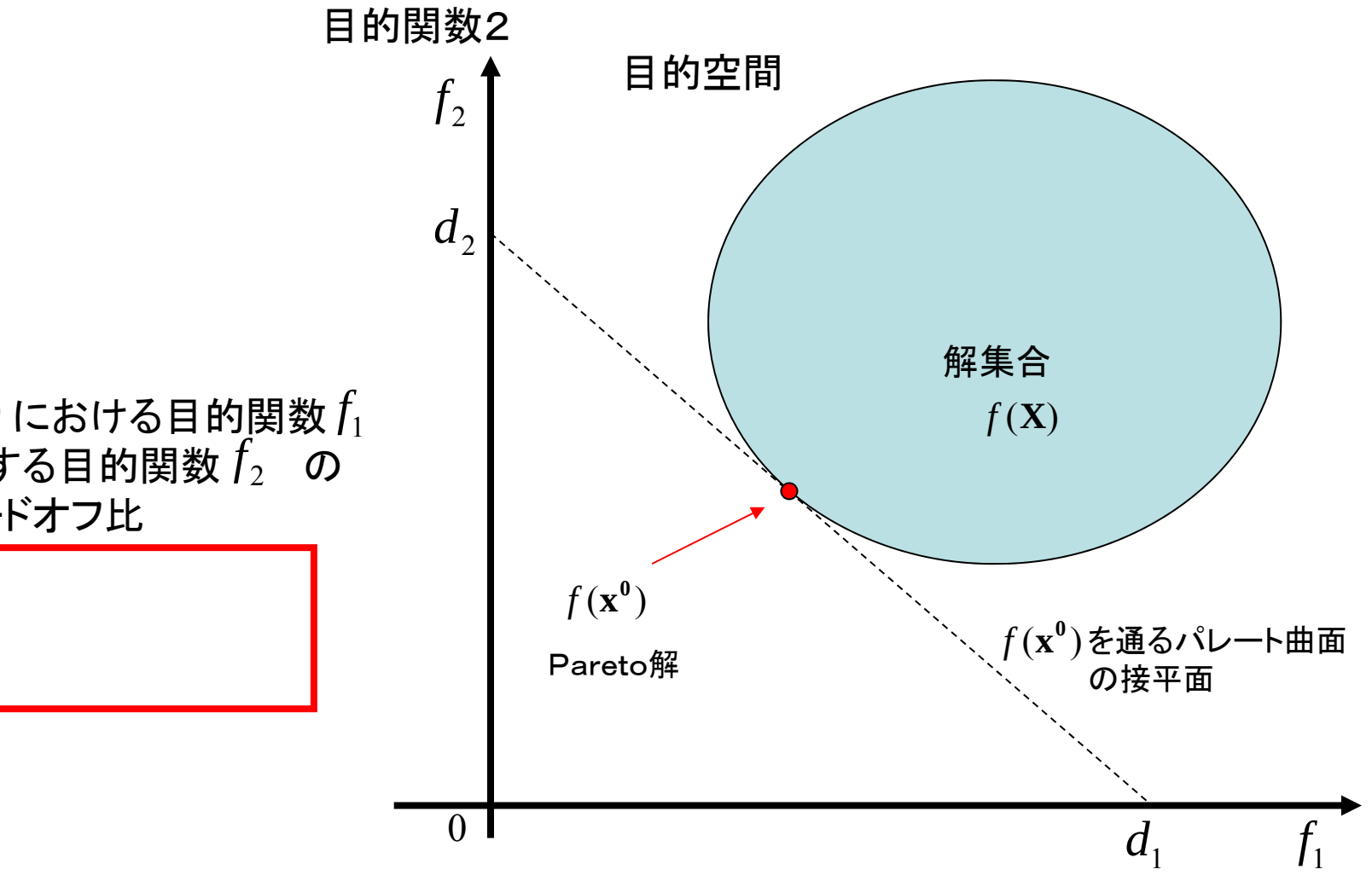
【定義】 多目的最適化問題(P)において、$\mathbf{x}^0 \in \mathbf{X}$ が以下の条件を満足するとき、
$\mathbf{x}^0$ を(P)のPareto(最適)解あるいは非劣解という。

「存在しない」
を意味する

$$\nexists \mathbf{x} \in \mathbf{X}: \quad f(\mathbf{x}) \leq f(\mathbf{x}^0)$$

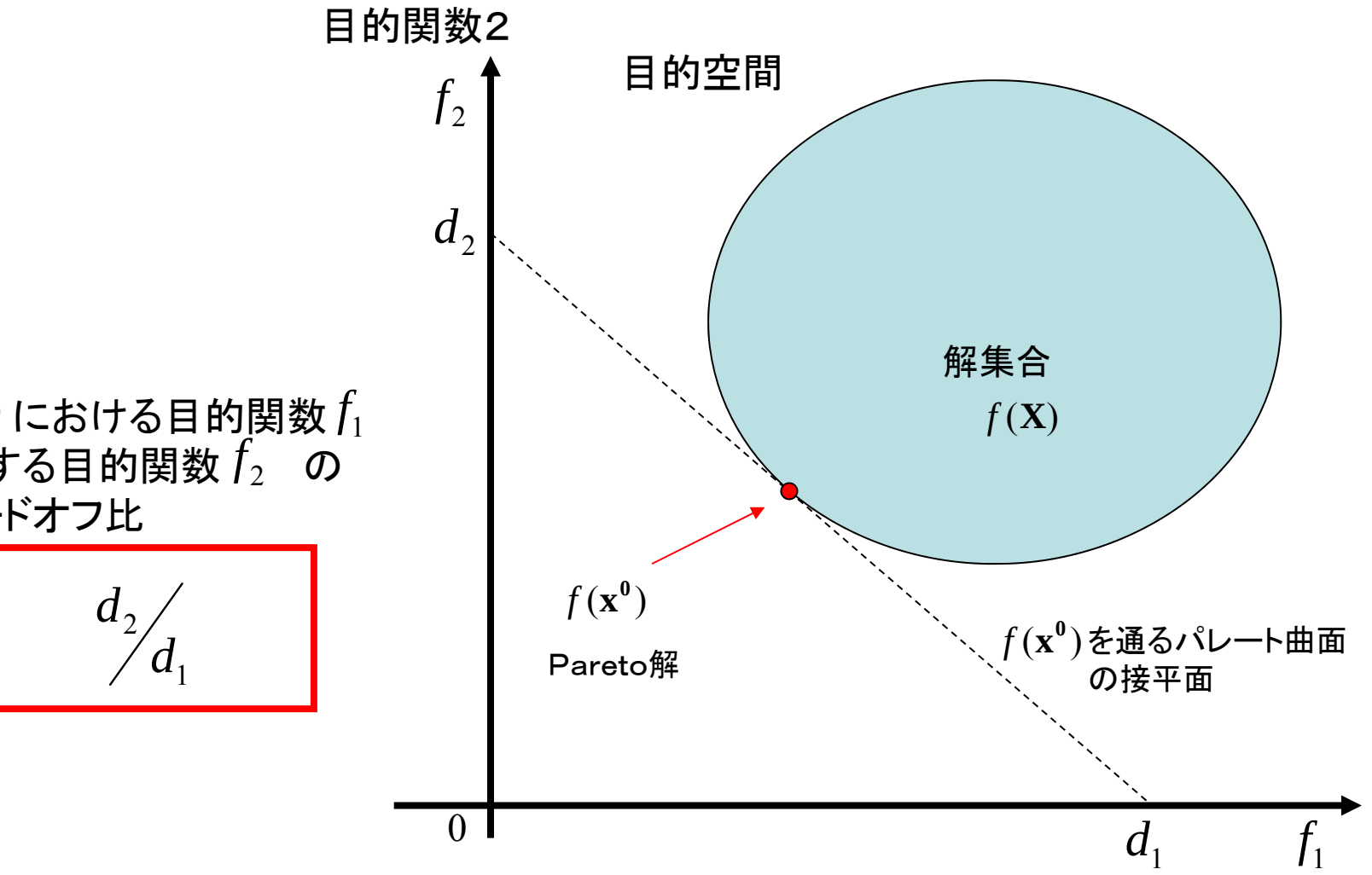
つまり、最適解は一般に無数に存在しうる！

パレート解のトレードオフ比



トレードオフ比が予め決まっている場合、
目的関数がただ1つの（単目的）最適化問題に帰着される。

パレート解のトレードオフ比



$f(\mathbf{x}^0)$ における目的関数 f_1
に対する目的関数 f_2 の
トレードオフ比

$$d_2 / d_1$$

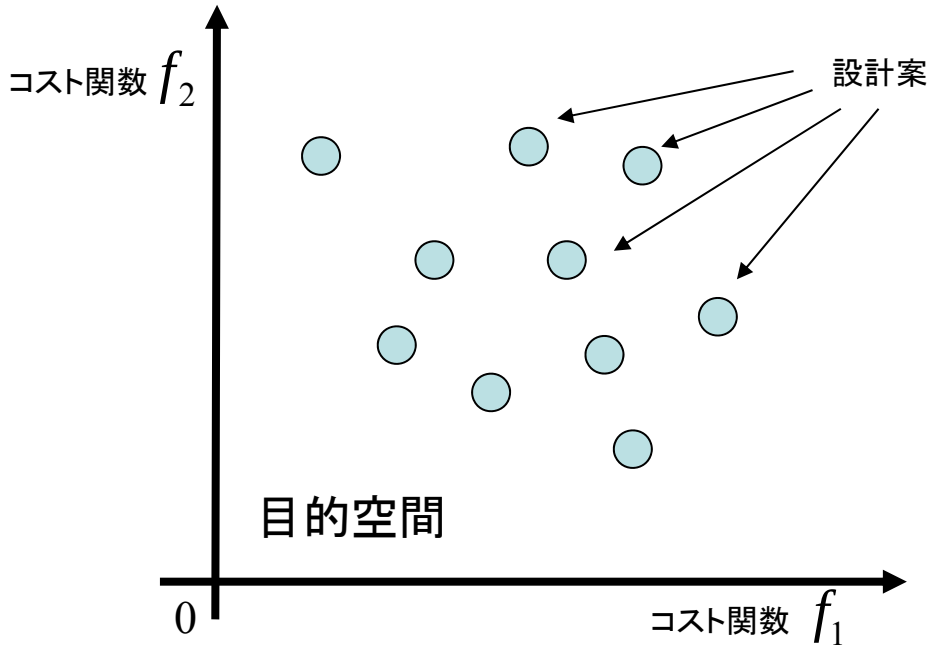
トレードオフ比が予め決まっている場合、
目的関数がただ1つの（単目的）最適化問題に帰着される。

ベテラン技術者による設計手順の技能伝承と多目的最適化の考え方

ベテラン技術者が扱っている設計問題が「難しい」と考えられている大きな原因：

1）

2）



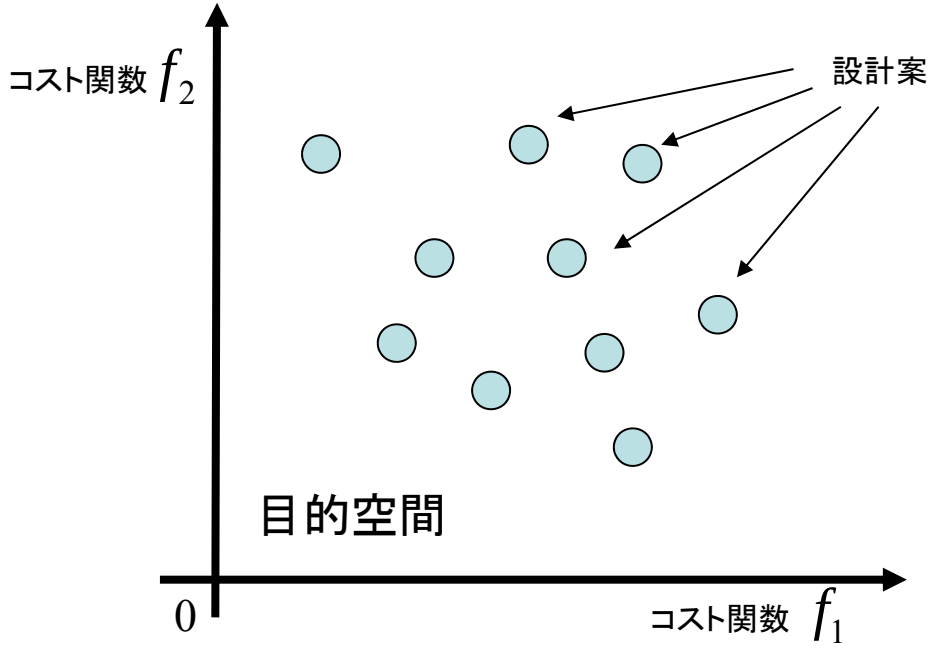
ベテラン技術者による設計手順の技能伝承と多目的最適化の考え方

ベテラン技術者が扱っている設計問題が「難しい」と考えられている大きな原因：

1） **実は多目的最適化問題**となっているのだが、設計者がそれに気付いていない

2） **評価項目（目的関数）の定式化があいまいで、きちんと数値化されていない**

→設計案の優劣の判定などはベテラン技術者の勘や経験で行われている**評価を数値化**し設計問題を**多目的最適化で定式化**すれば単純になる



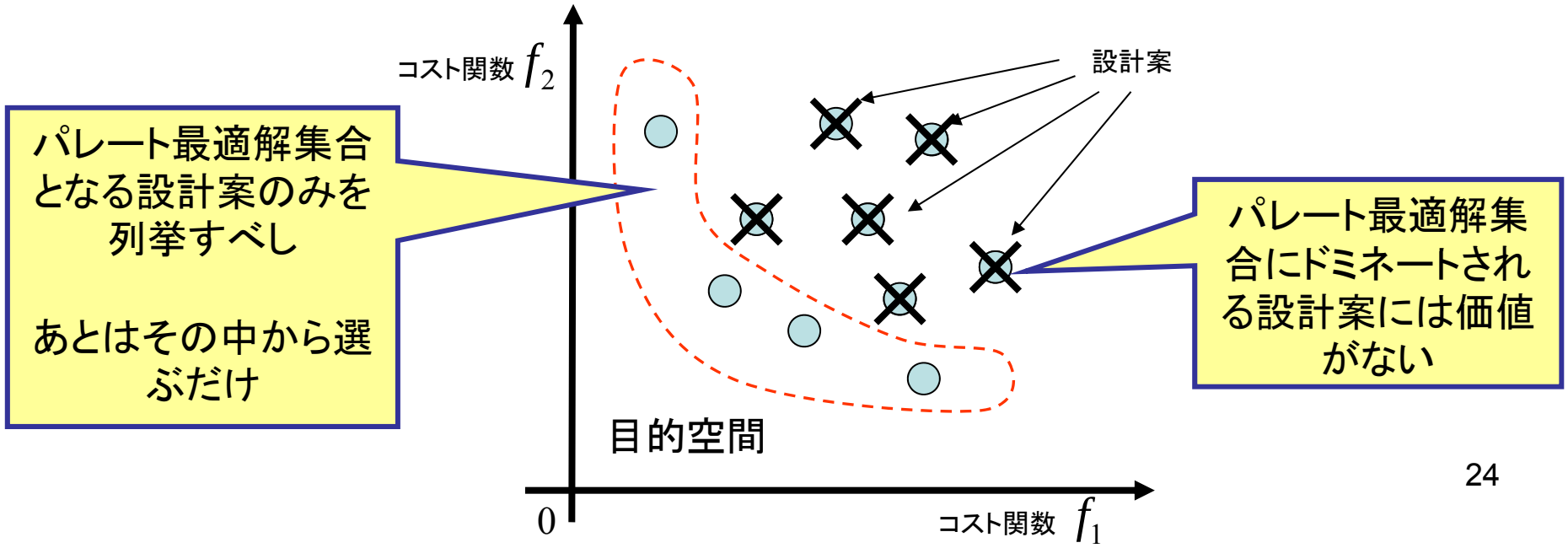
ベテラン技術者による設計手順の技能伝承と多目的最適化の考え方

ベテラン技術者が扱っている設計問題が「難しい」と考えられている大きな原因：

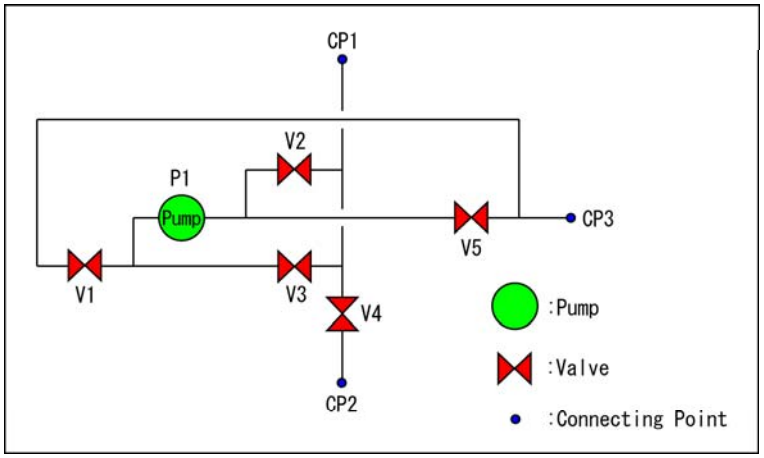
1） **実は多目的最適化問題**となっているのだが、設計者がそれに気付いていない

2） **評価項目（目的関数）の定式化があいまいで、きちんと数値化されていない**

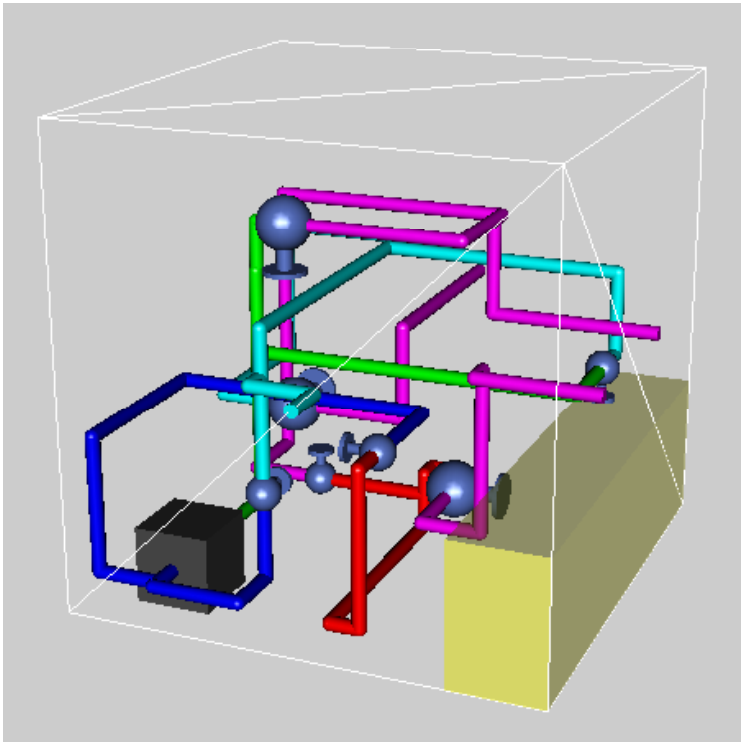
→設計案の優劣の判定などはベテラン技術者の勘や経験で行われている**評価を数値化**し設計問題を**多目的最適化で定式化**すれば単純になる



多目的最適化の例：配管設計問題



【系統図】



●最適化すべきパラメータ：
各パイプの位置・長さなど（数十個） → **設計変数 x**

●配管設計案の評価：
（1）製造コスト（管路径×管路長の合計） f_1
（2）バルブの操作性 f_2
（3）パイプのメンテナンス性 f_3

同時に多様な評価を満足させなければならない

→ **多目的**

【多目的最適化の解法】

（1） 単目的最適化手法を利用する方法

多目的最適化問題は、目的関数のトレードオフ比を決めれば
単目的最適化問題に帰着される

様々なトレードオフ比の組合せに対して単目的最適化手法を適用

場合によってはかなり強力だが、目的関数の次元数は2～3程度まで

（2） 多目的遺伝的最適化手法

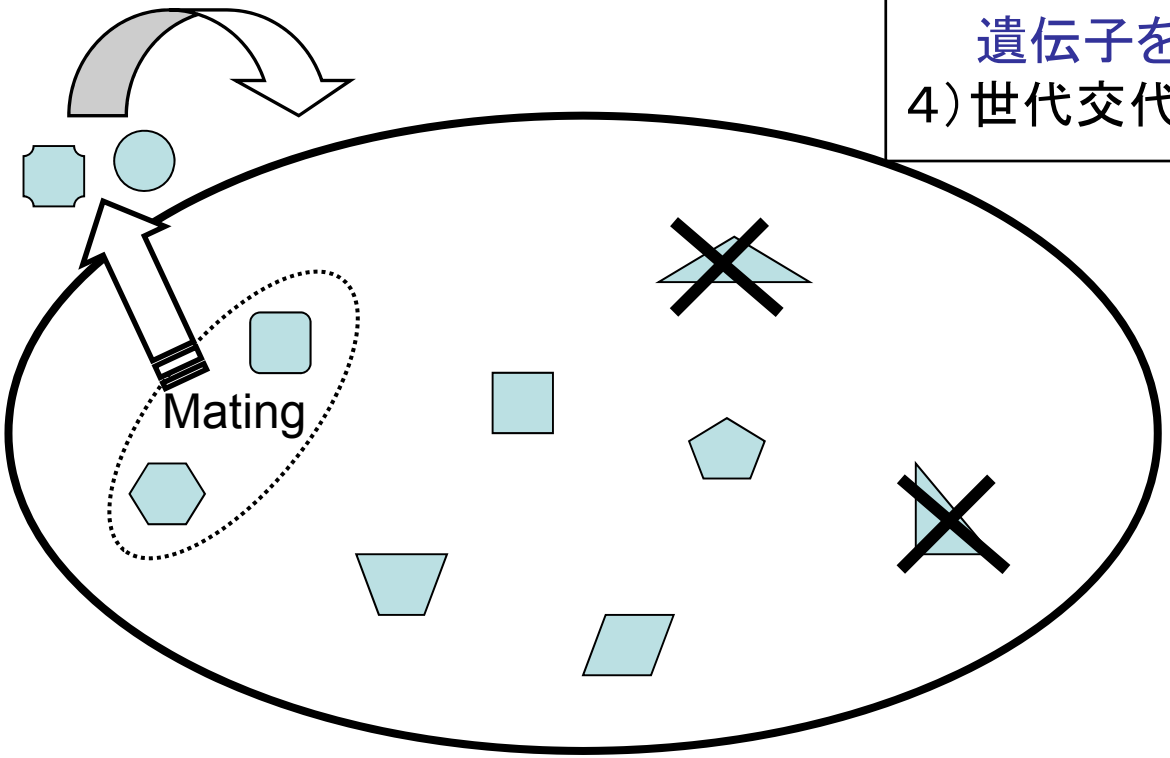
交叉方法：　単目的の場合と同じ
　　ただし交叉における親個体の選択にはノウハウあり
世代交代方法：　NSGA2（ドミネート淘汰戦略＋ニッチング）

【復習：遺伝的手法による最適化】

生物の進化の仕組みを模倣

生物個体　→　遺伝子によるコード化
優れた個体　→　高い評価値・生存率

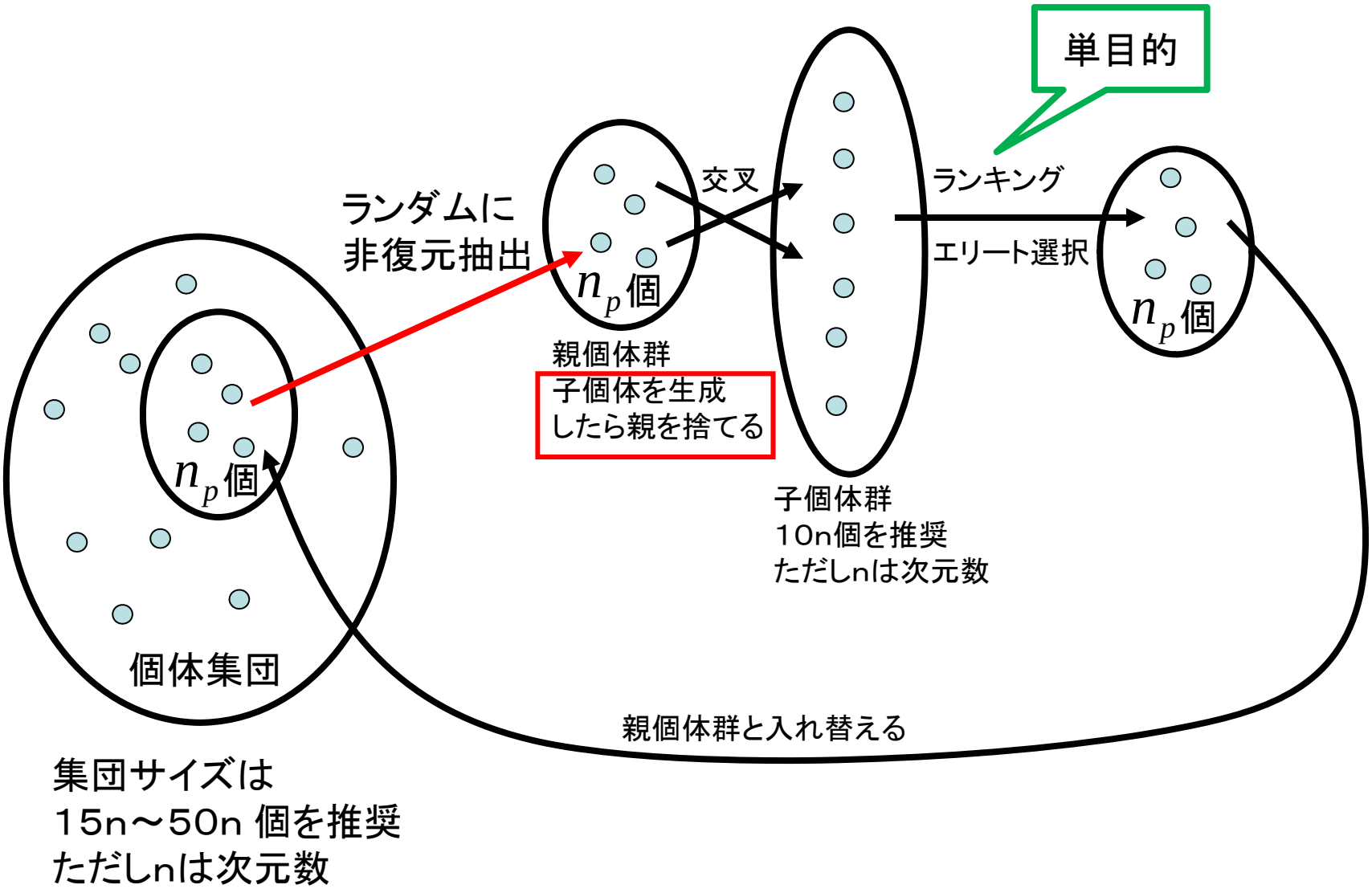
【処理手順】
1）設計したい物をパラメータ化　→　遺伝子
2）いろいろな個体を集めて集団を作り、
　個体を評価に応じて自然淘汰
3）生き残った個体同士で
　遺伝子を交換　→　新しい個体生成
4）世代交代を繰り返し、集団を進化



【探索の特徴】
・評価値の悪い個体でも
　生き残るチャンスが与えられる
・優れた親同士を掛け合わせる
　ことで、両親の優れた形質を
　受け継いだ子個体が生成される
・集団全体が進化していく

たった1つの解候補を改善していくよりも、
より優れた解や多様な評価を持つ複数の解を発見できる

多親世代交代モデルJGG ［小林2007］

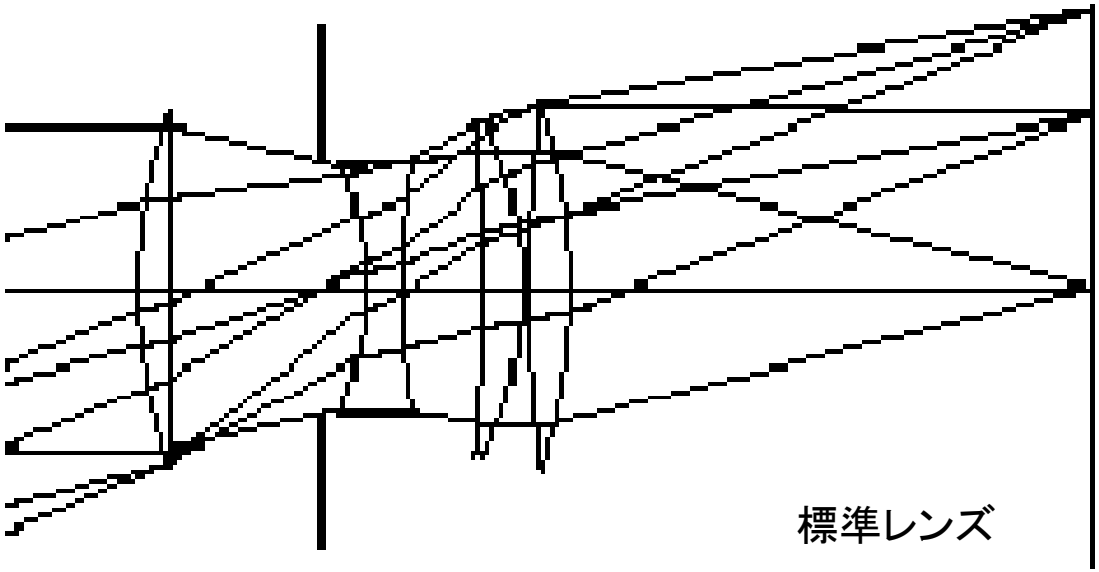


集団サイズは
15n～50n 個を推奨
ただしnは次元数

MGGと比較すると 2.5～13倍の性能アップ

多目的最適化と遺伝的手法

例）レンズ系の自動設計



●レンズ系を特徴付けるパラメータ：
各レンズの**曲率**、**レンズ間距離**など（数十個） → **遺伝子**

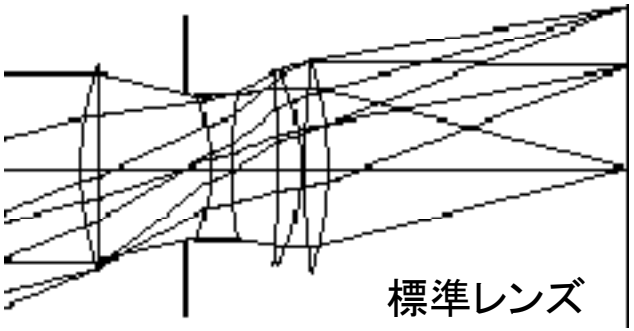
●レンズ系の評価：
（1）歪曲（distortion）
（2）焦点（focus）
（3）色収差（色のにじみ）
（4）その他（像の明るさ、製造コスト、丈夫さなど）

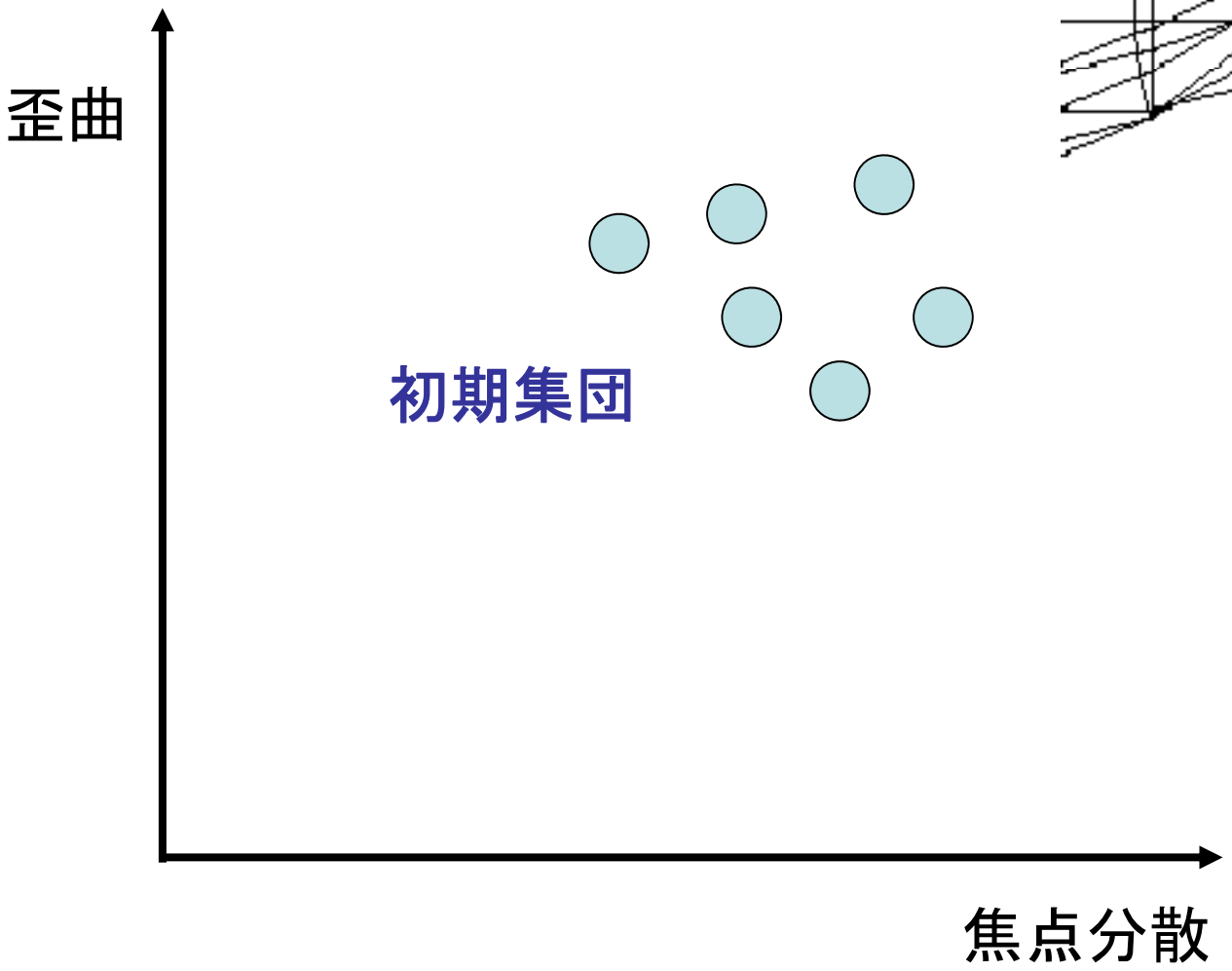
同時に多様な評価を満足させなければならない

多目的

GAによる多目的最適化の例:レンズ系の自動設計

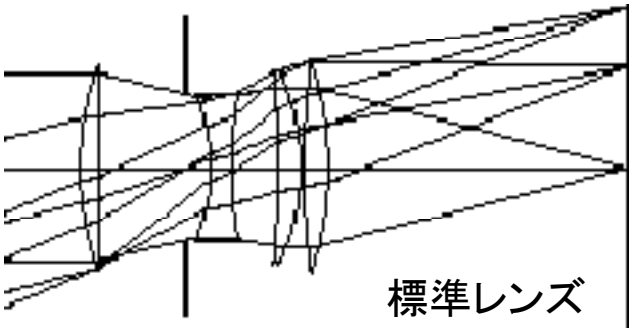
世代交代方法を多目的に合わせる

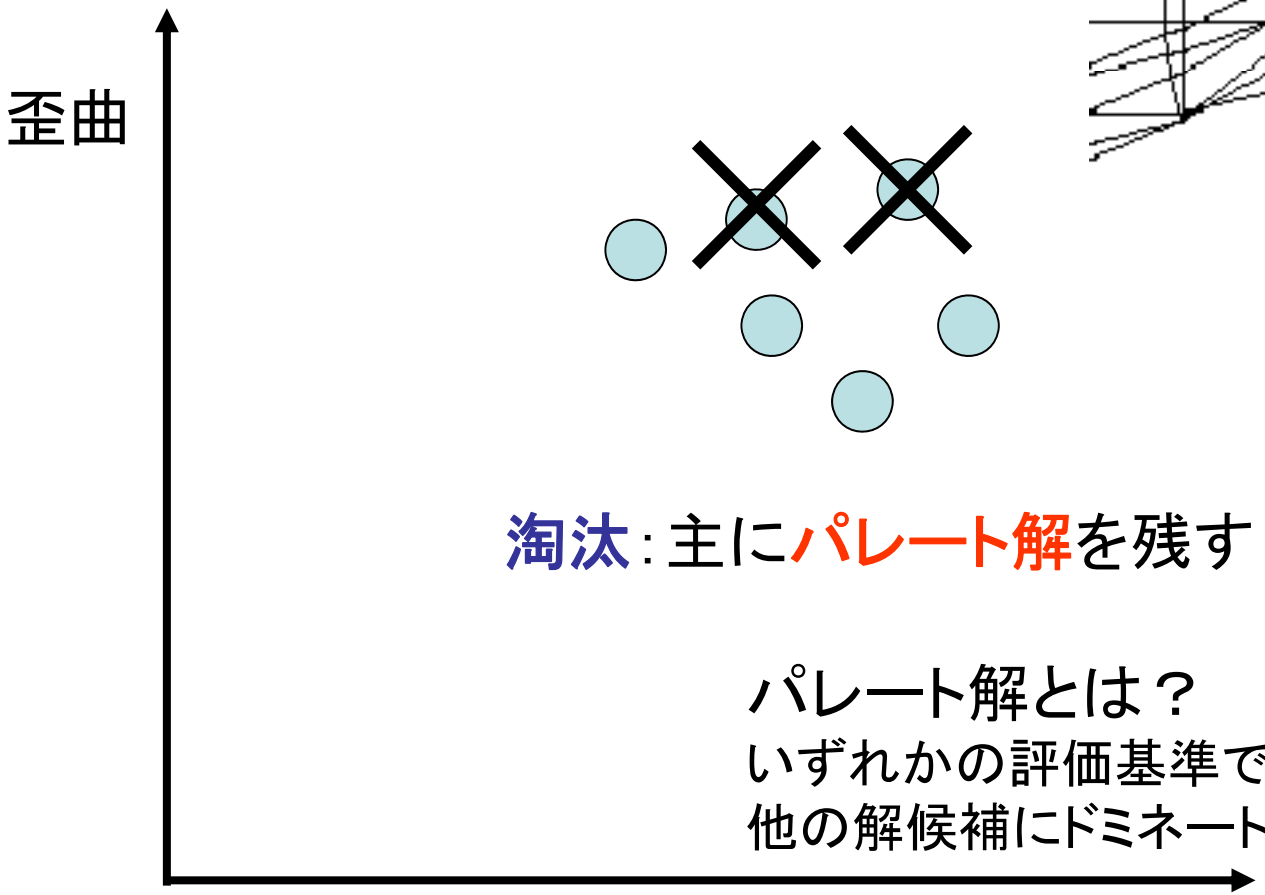




GAによる多目的最適化の例：レンズ系の自動設計

世代交代方法を多目的に合わせる



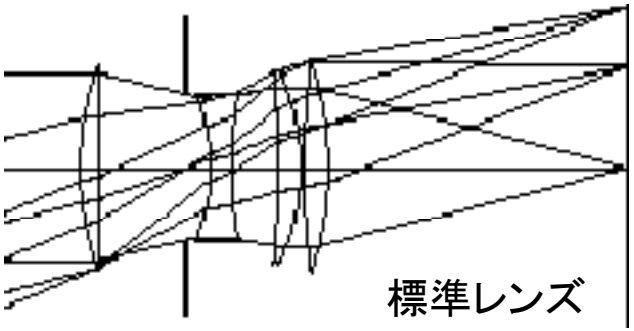


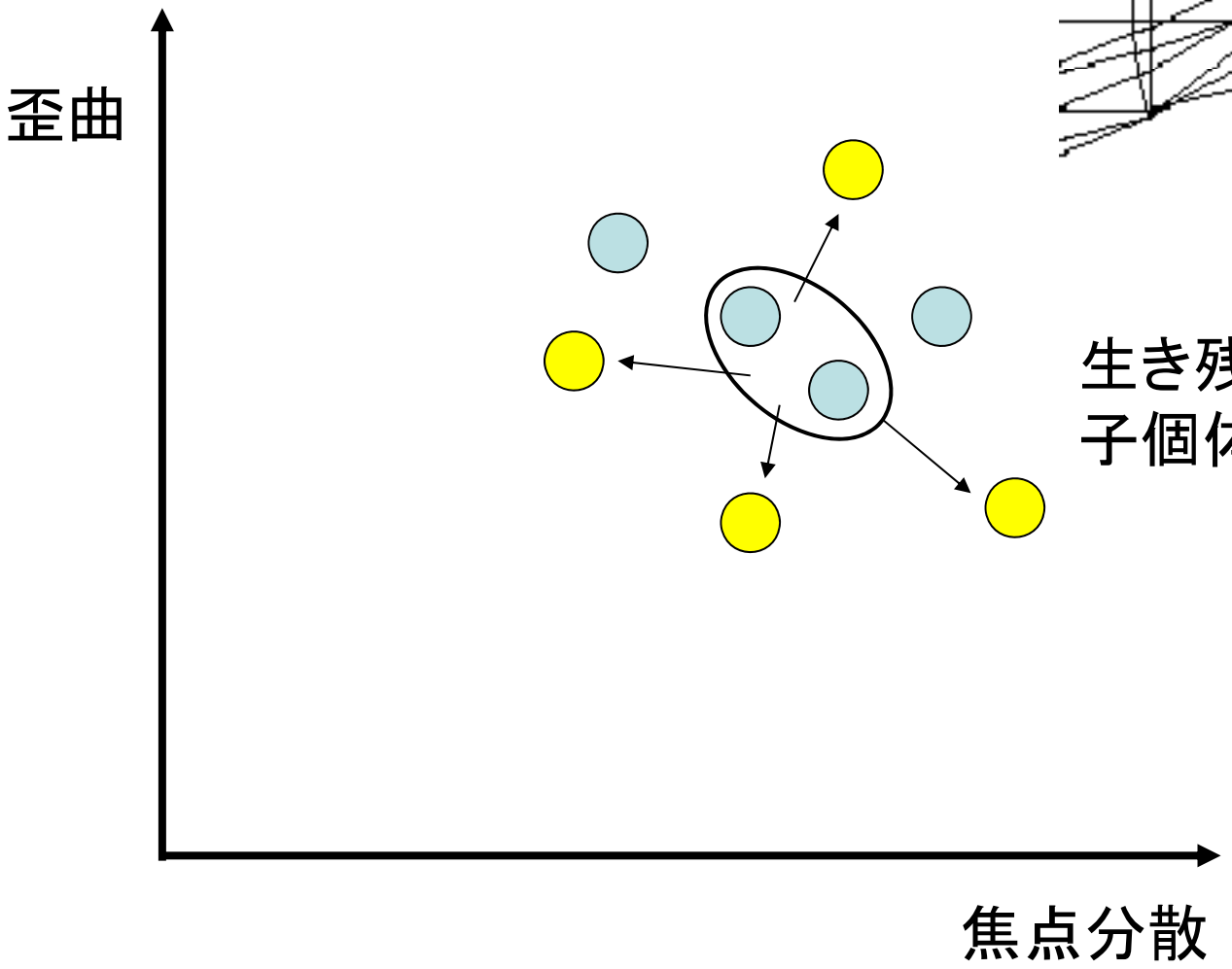
淘汰：主に**パレート解**を残す

パレート解とは？
いずれかの評価基準で評価が最大であり、
他の解候補にドミネートされない解（**非劣解**）

GAによる多目的最適化の例：レンズ系の自動設計

世代交代方法を多目的に合わせる

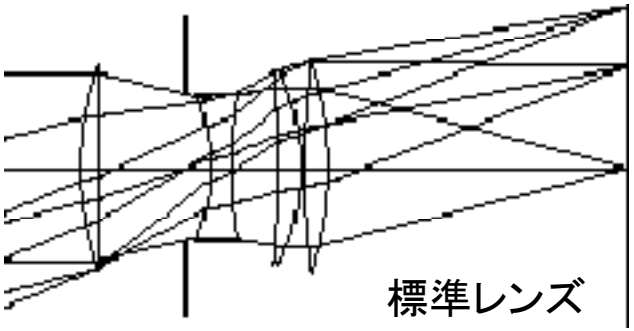


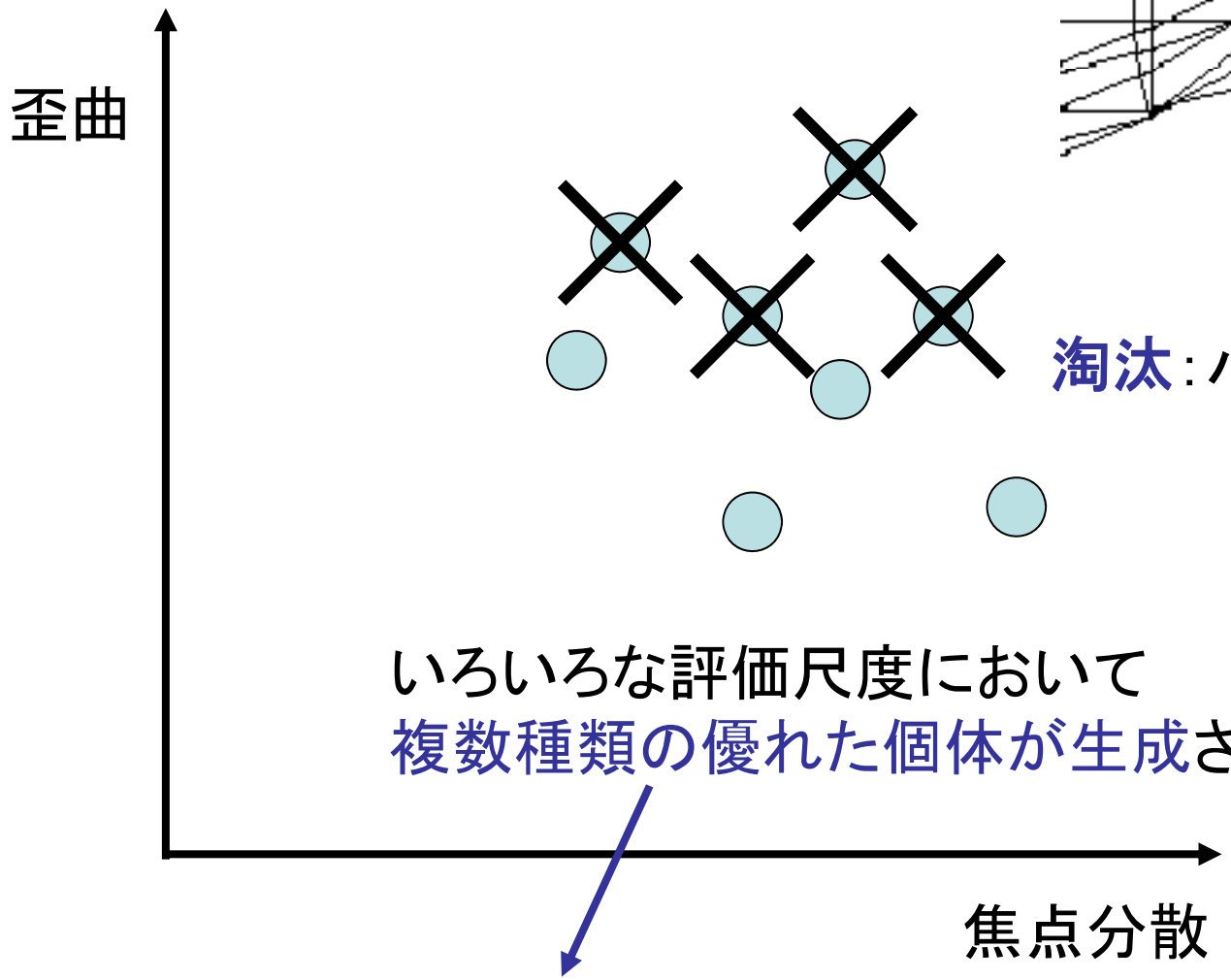


生き残った個体で
子個体生成

GAによる多目的最適化の例：レンズ系の自動設計

世代交代方法を多目的に合わせる



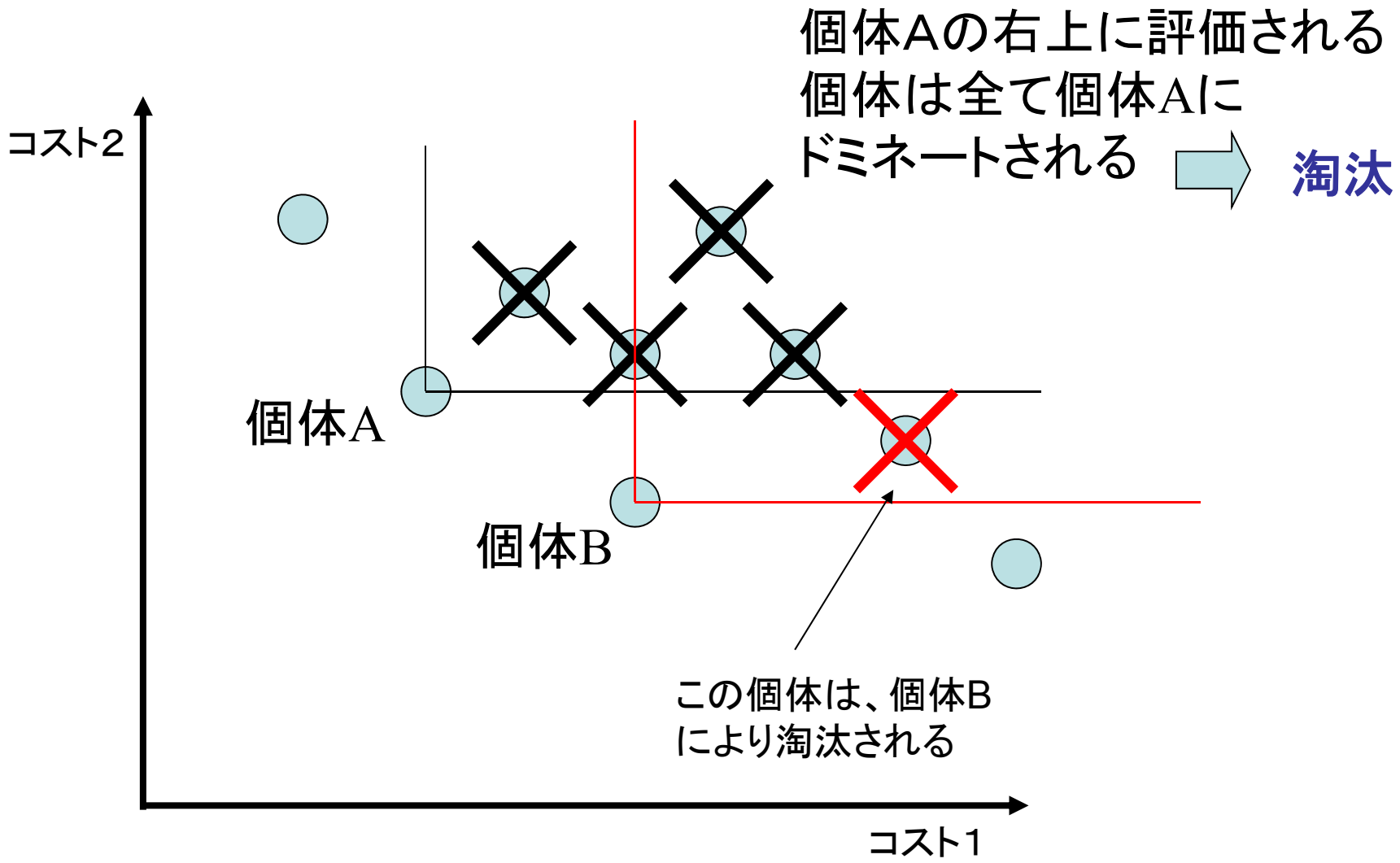


淘汰：パレート解を残す

いろいろな評価尺度において
複数種類の優れた個体が生成されていく

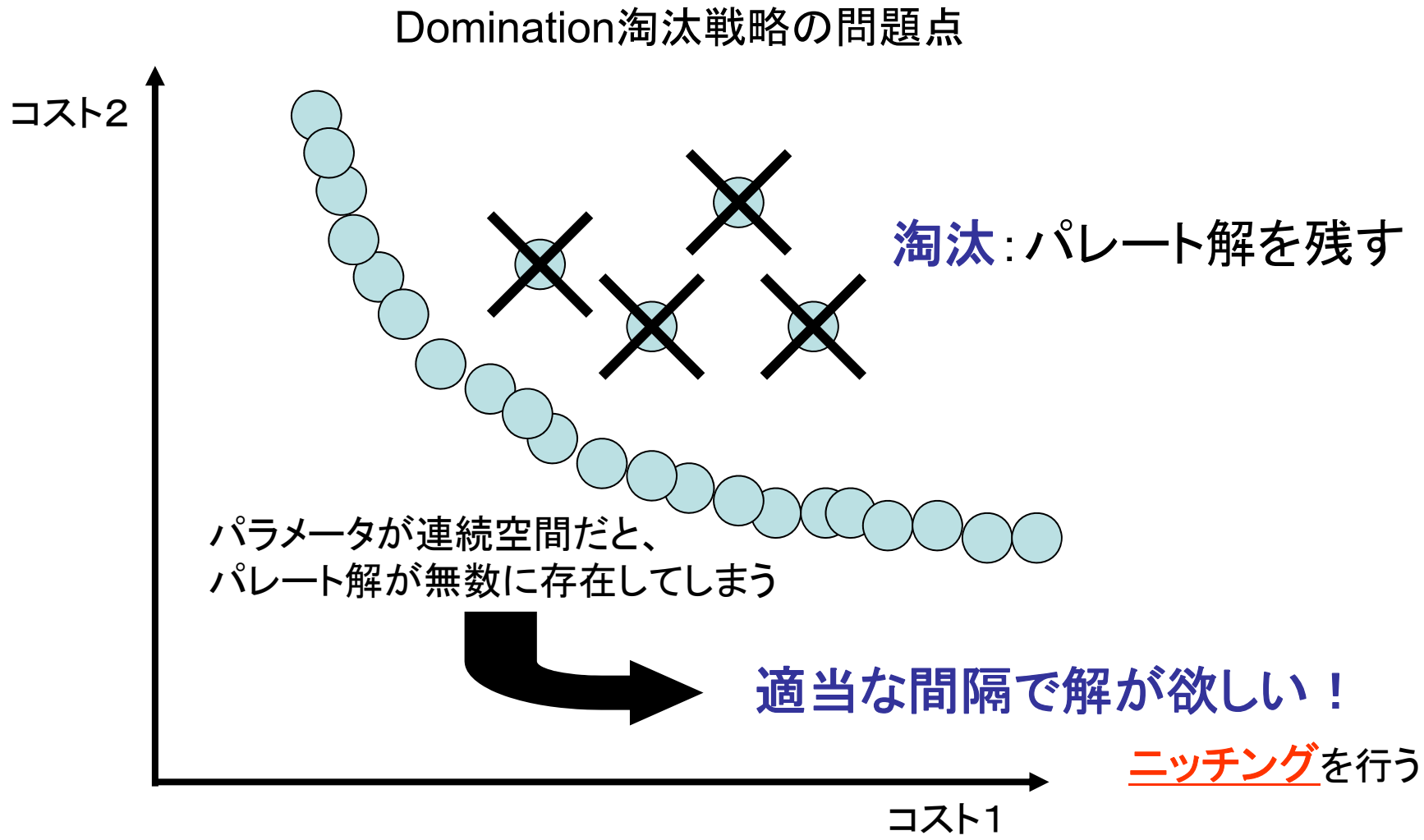
提示された複数の解候補から選択して設計できる

GAによる多目的最適化における淘汰戦略



他の解候補にドミネートされないパレート解
のみ残していく → Domination淘汰戦略

GAによる多目的最適化における淘汰戦略

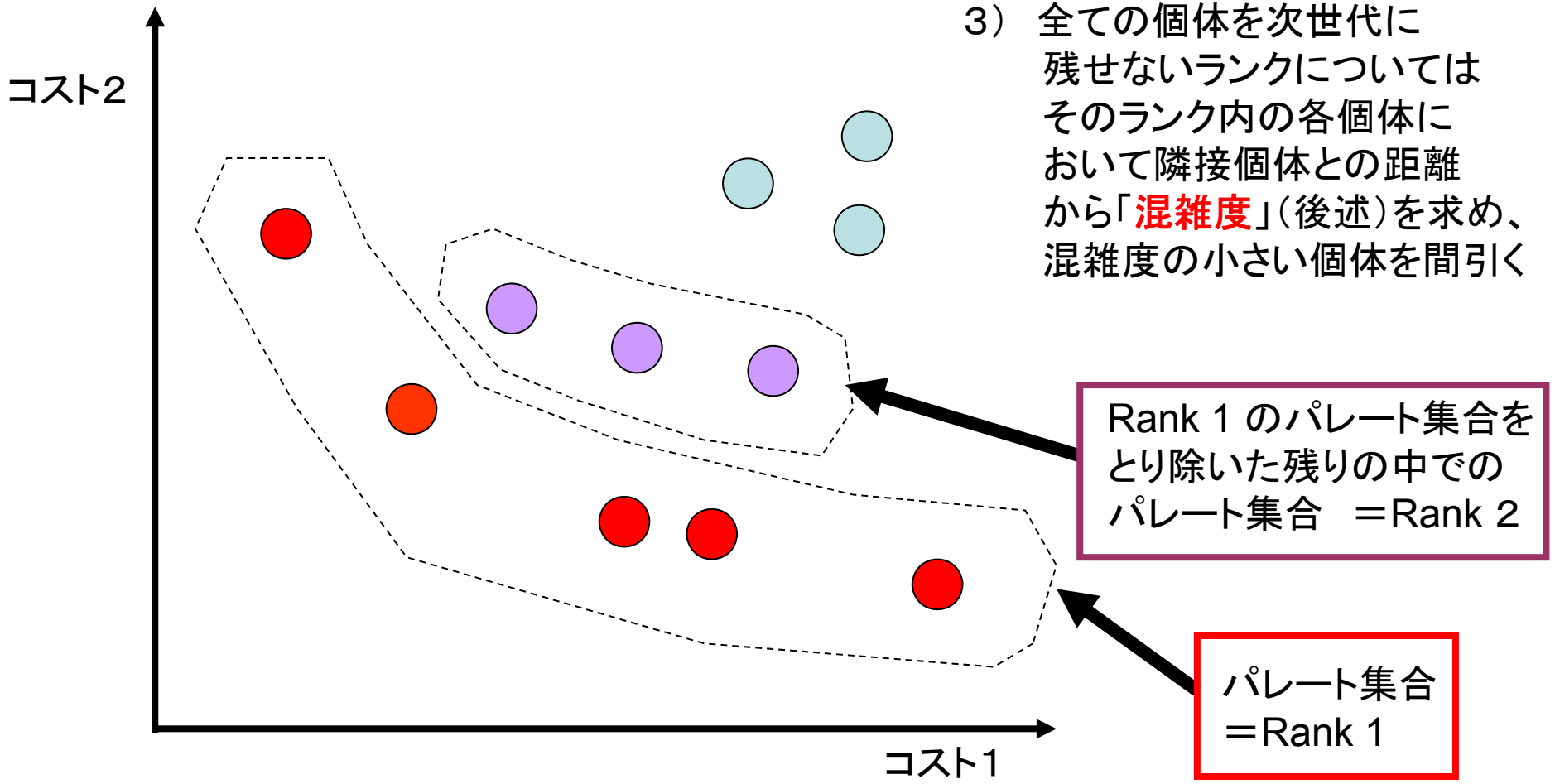


他の解候補にドミネートされないパレート解
のみ残していく → Domination淘汰戦略

GAによる多目的最適化における淘汰戦略： NSGA2

NSGA: Non-dominated Sorting Genetic Algorithm
［K.Deb and T.Goel, 2001］

1） ランキングを行う（図を参照）
2） 次世代に残す世代を
ランクの高い順に選択
3） 全ての個体を次世代に
残せないランクについては
そのランク内の各個体に
おいて隣接個体との距離
から「**混雑度**」（後述）を求め、
混雑度の小さい個体を間引く

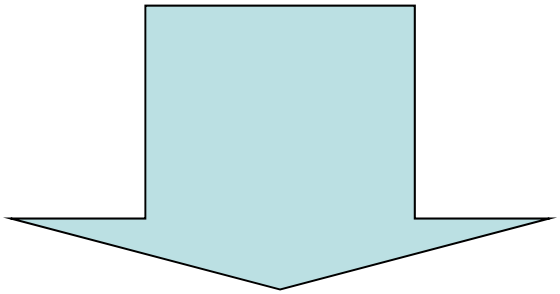


2007年現在、多目的GAではこのNSGA2が世界的にスタンダード

【補足】 NSGA2での「混雑度」の求め方

目的関数が2つの場合は、各ランク内で「隣接する個体間の距離」を求めるのは簡単だが、
目的関数がもっと高次元だったらどうするか？

各ランク内の個体集合を**目的関数別にソート**し、全個体それぞれの隣接する個体との
距離を求め、全ての目的関数における距離の和をその個体の混雑度とする。
（ある目的関数において**端にある個体は、その目的関数おける距離の最大値**を与える）

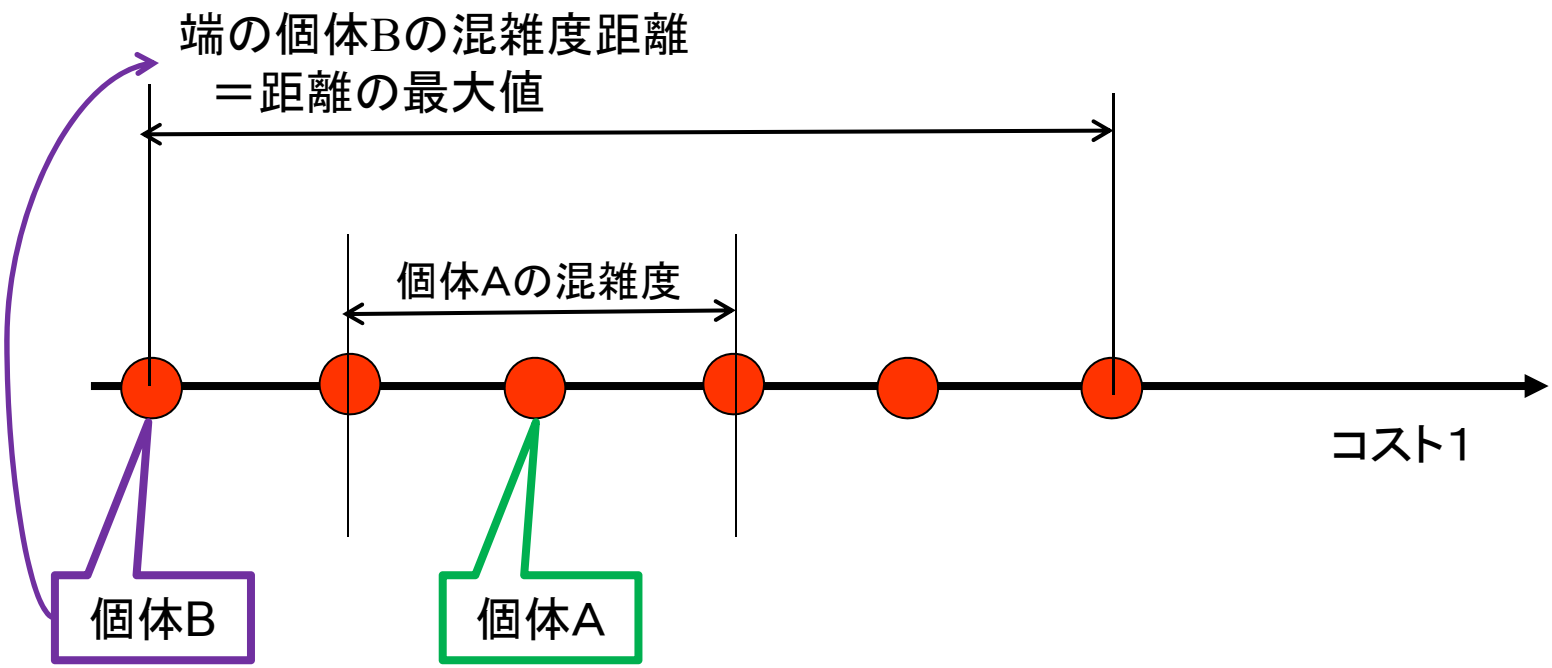


パラメータを必要としない優れた方法

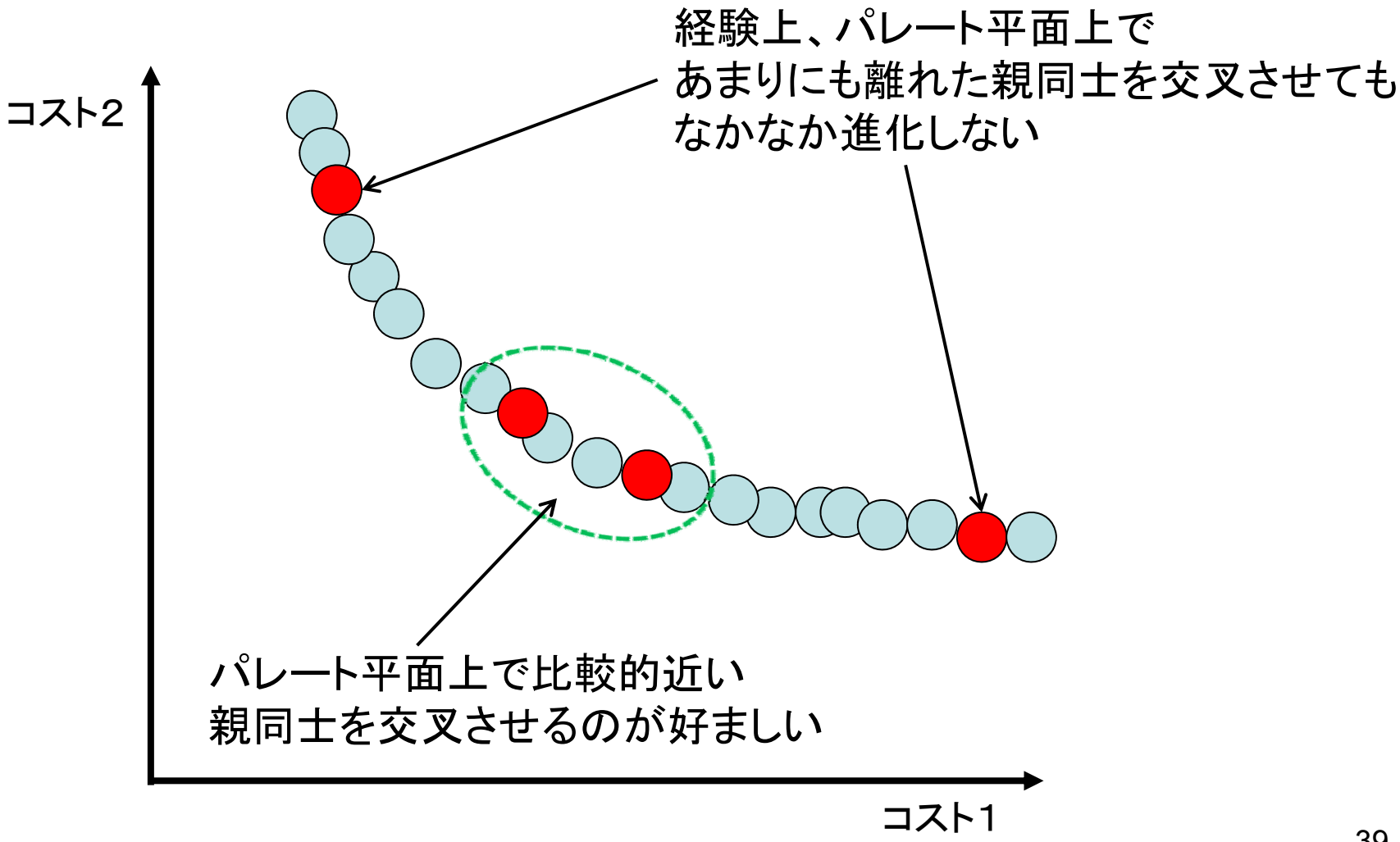
パラメータ空間でソートして多様な解を残す方法も提案されている

注目している個体が「無い」場合と、「追加した」場合のハイパーボリュームの差の大きさ
を混雑度とする場合もあるが、目的関数が高次元になると計算コストがかかる

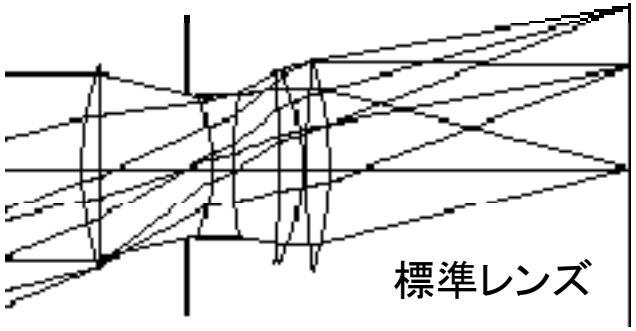
「混雑度」の計算例　＠目的関数空間

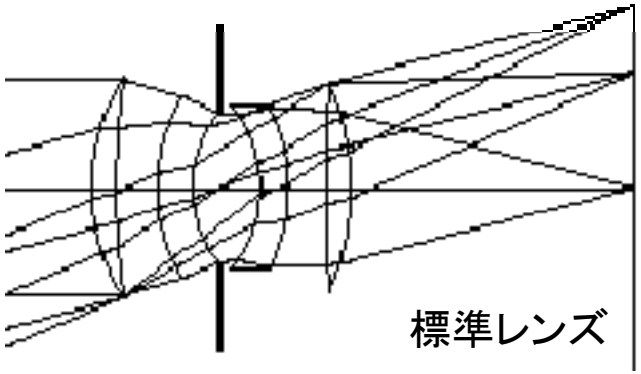


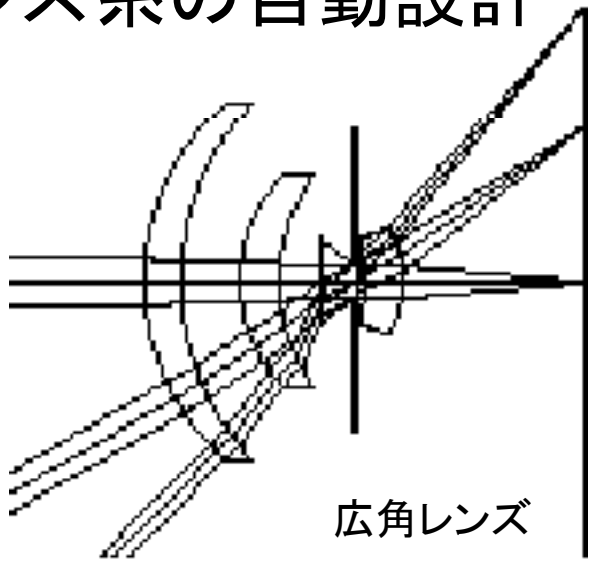
【参考】 多目的の連続関数最適化における親個体の選択について

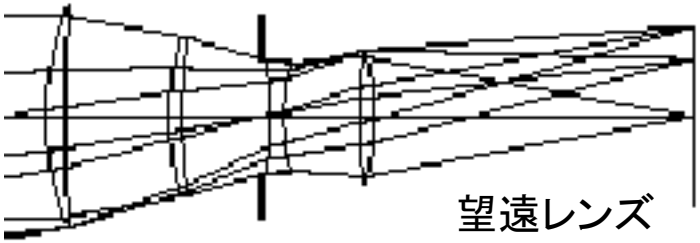


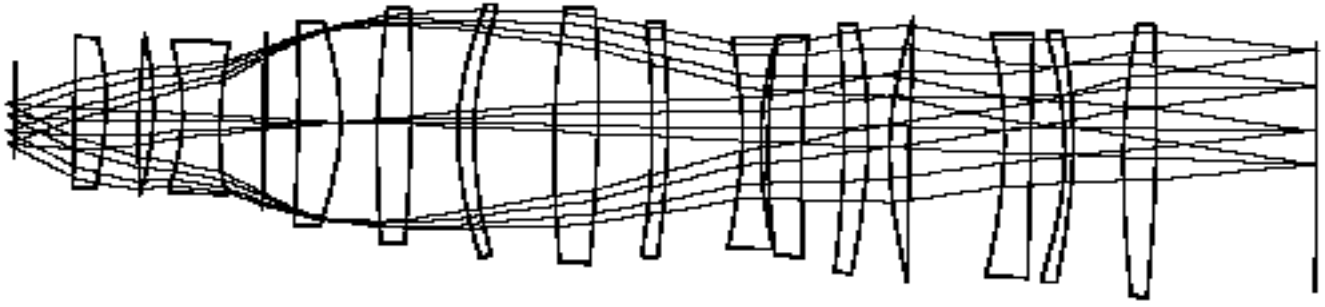
遺伝的アルゴリズム（GA）の応用：レンズ系の自動設計









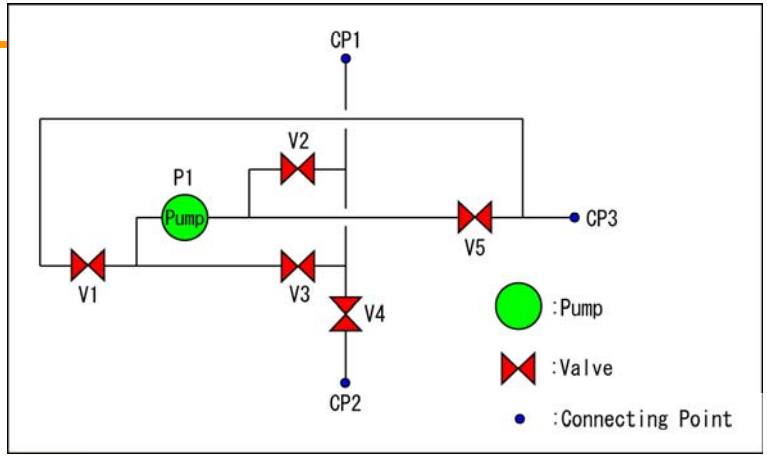


15枚組投影レンズ

評価値が最大になるようなレンズパラメータを探索
→新しいレンズの組合せを発見

計算機シミュレーション設定

Pipeline list



Input : From-to list

LINE NO	FLUID	SIZE	CLASS	FROM-TO					
P-001	S	150	-	V1	P1	V3	-	-	-
P-002	S	150	-	V1	V5	CP3	-	-	-
P-003	P	150	-	P1	V2	V5	-	-	-
P-004	P	150	-	V2	CP1	V3	V4	-	-
P-005	D	150	-	V4	CP2	-	-	-	-

VALVE NO	SIZE L	SIZE D	SIZE H	CLASS	AFTER			FORWARD		
V1	0.3	0.3	0.5	-	V5	CP3	-	P1	V3	-
V2	0.3	0.3	0.5	-	P1	V5	-	V3	V4	CP1
V3	0.3	0.3	0.5	-	P1	V1	-	V2	V4	CP1
V4	0.5	0.5	0.8	-	CP2	-	-	V2	V3	CP1
V5	0.5	0.5	0.8	-	P1	V2	-	V1	CP3	-

Equipment arrangement list

Input : Equipment list

EQUIP-NO	CATEGORY	TYPE	X	Y	Z	DIR	AFTER			FORWARD		
P1	PUMP	RK2	1.5	2.0	0.0	90.0	V1	V3	-	V2	V5	-

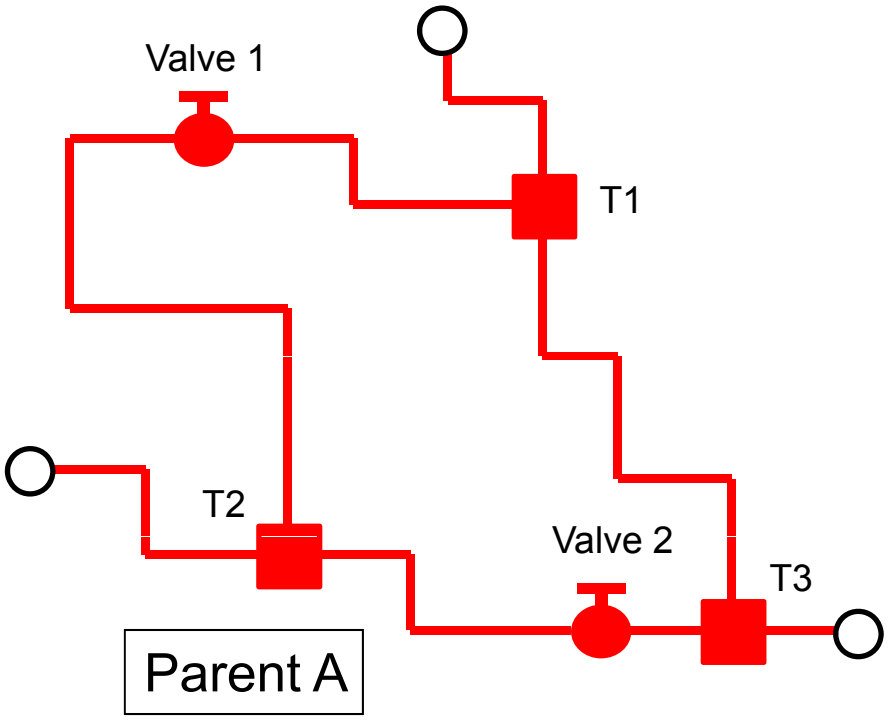
CATEGORY	TYPE	SIZE X	SIZE Y	SIZE Z	V1			V2		
					X	Y	Z	X	Y	Z
PUMP	RK2	0.8	0.8	0.8	0.4	0.0	0.4	0.4	0.8	0.4
					V3			V4		
					X	Y	Z	X	Y	Z
					0.4	0.0	0.4	-	-	-
					V5			CP1		
					X	Y	Z	X	Y	Z
					0.4	0.8	0.4	-	-	-
					CP2			CP3		
					X	Y	Z	X	Y	Z
					-	-	-	-	-	-

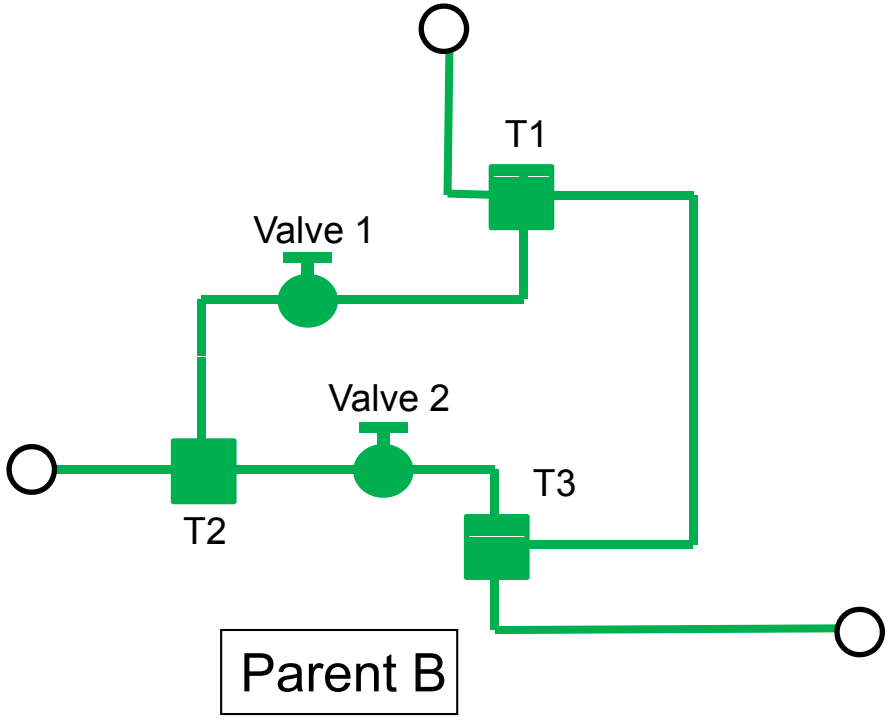
Total

Valves:	5
Pump(s):	
Connections:	3
T-branches:	5
Pipes:	15
Parameters for equipments:	40
Combination	over 10^{40}

機器配置図の交叉オペレーション

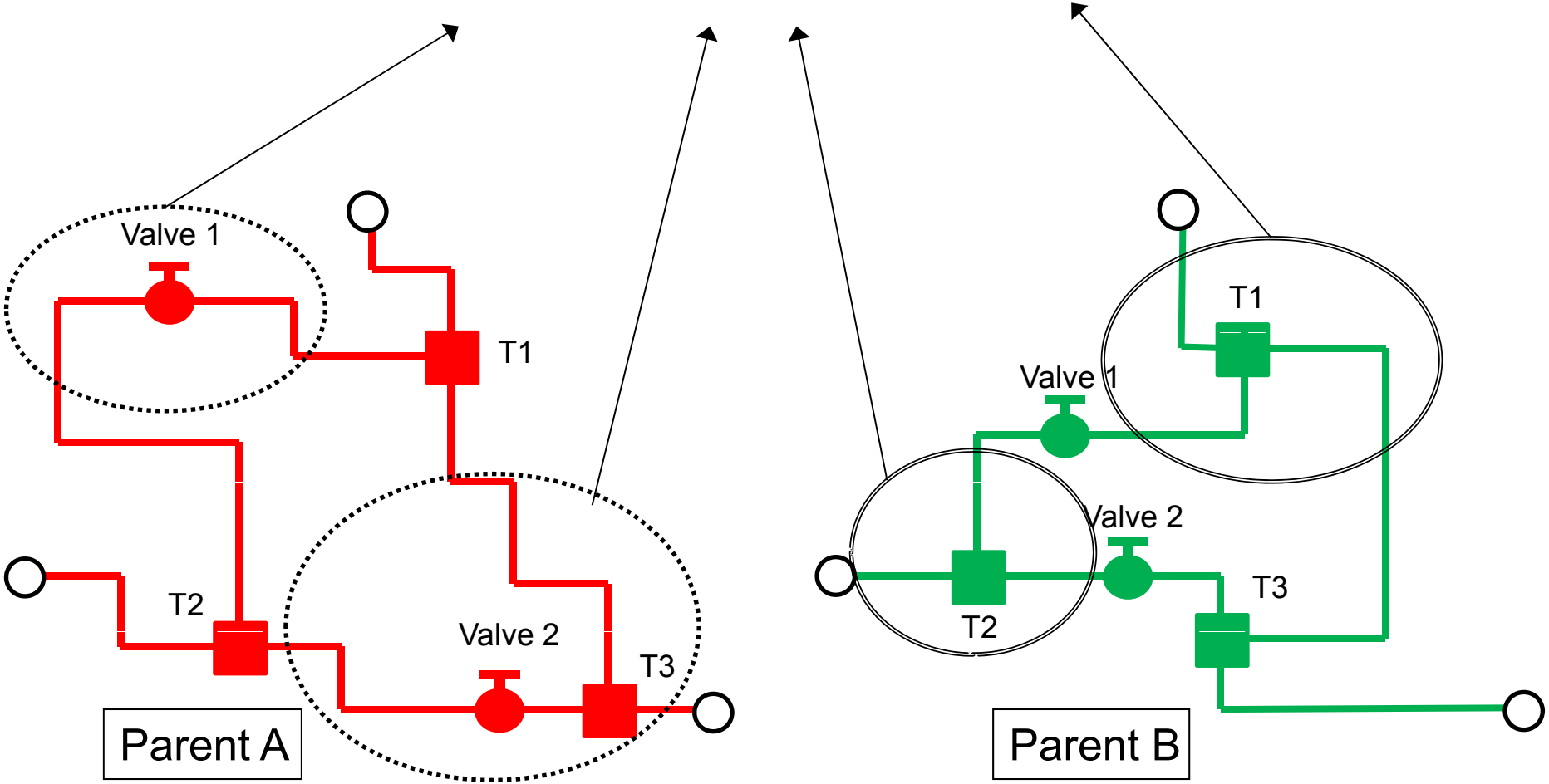
親個体から機器の位置と方向を選んで子個体を形成

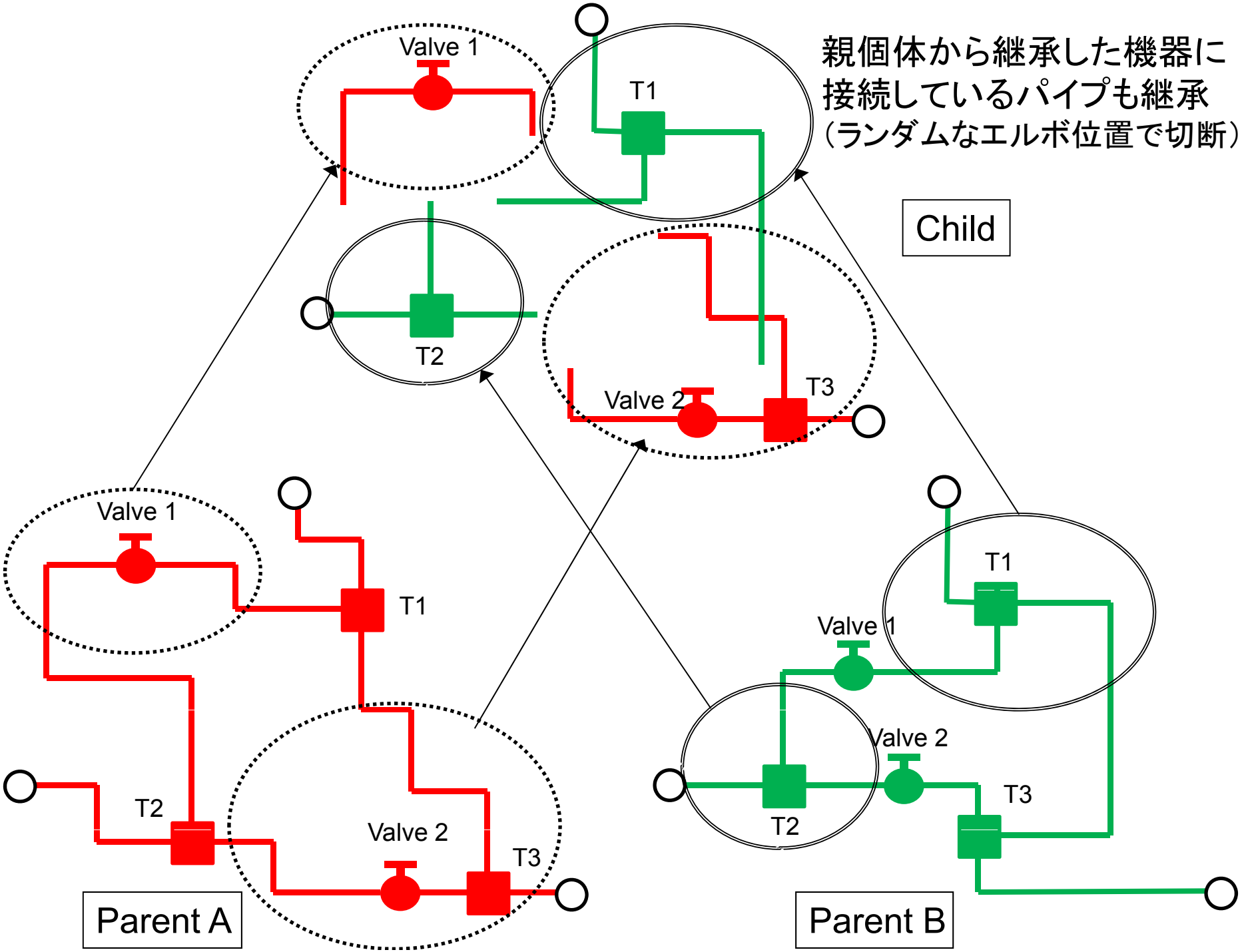


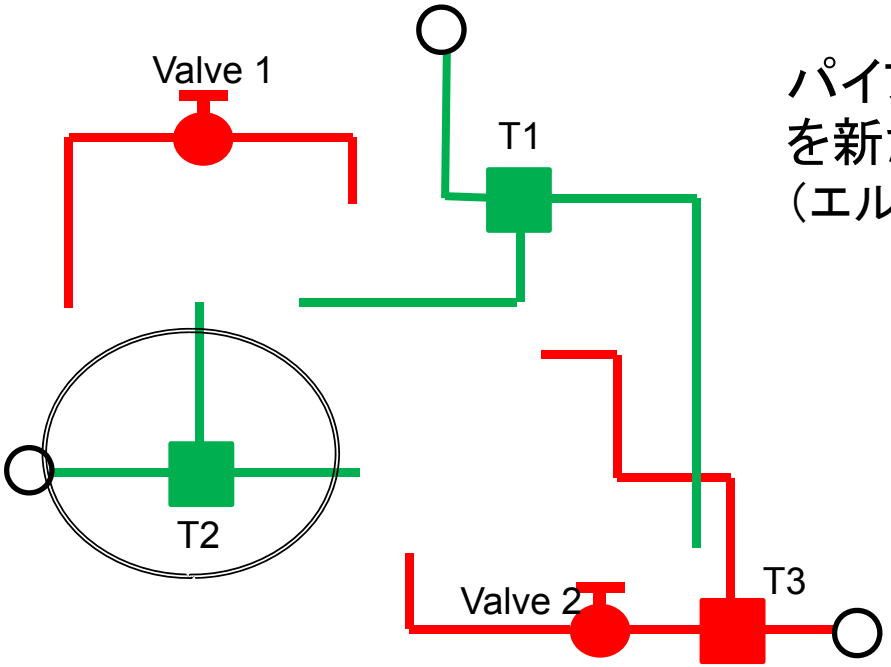


機器配置図の交叉オペレーション

親個体から機器の位置と方向を選んで子個体を形成

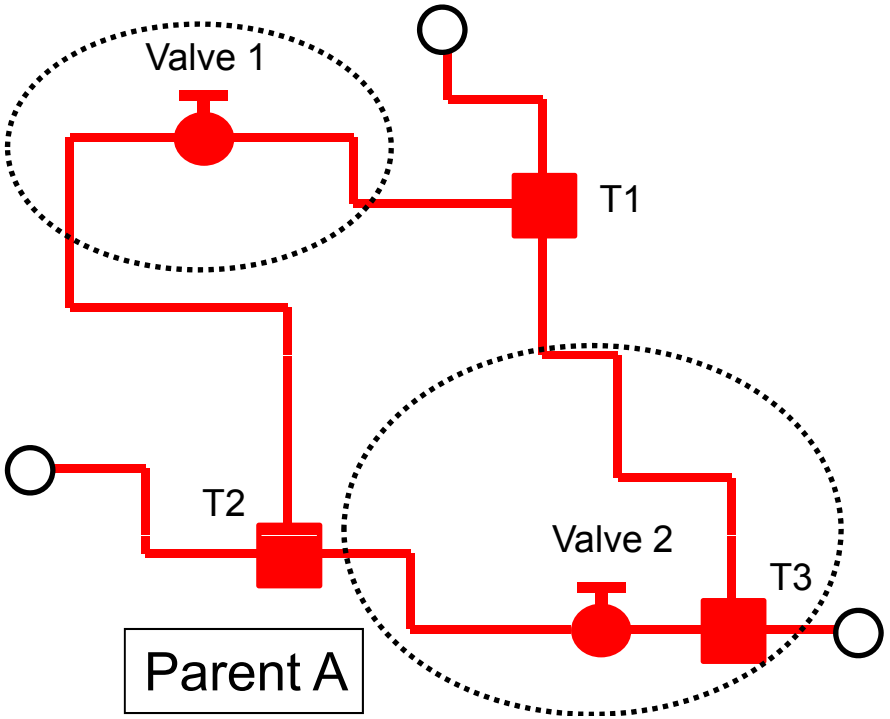



親個体から継承した機器に
接続しているパイプも継承
（ランダムなエルボ位置で切断）
Child
Valve 1
T1
T2
Valve 2
T3
Parent A
Valve 1
T1
T2
Valve 2
T3
Parent B
T1
Valve 1
Valve 2
T2
T3

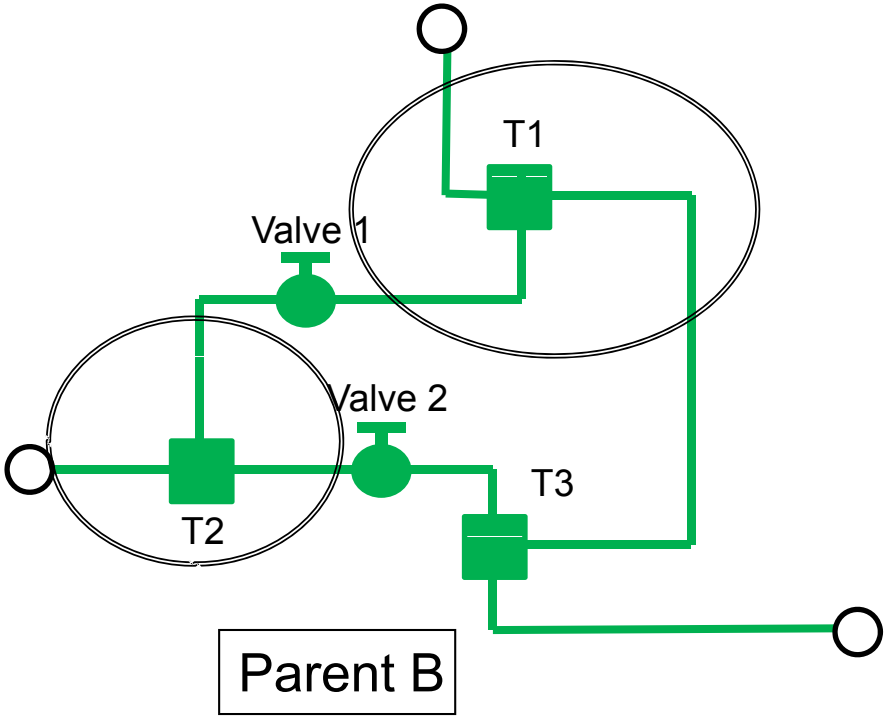


パイプが途切れている部分を新たに繋ぎ直す
（エルボ3個以下で）

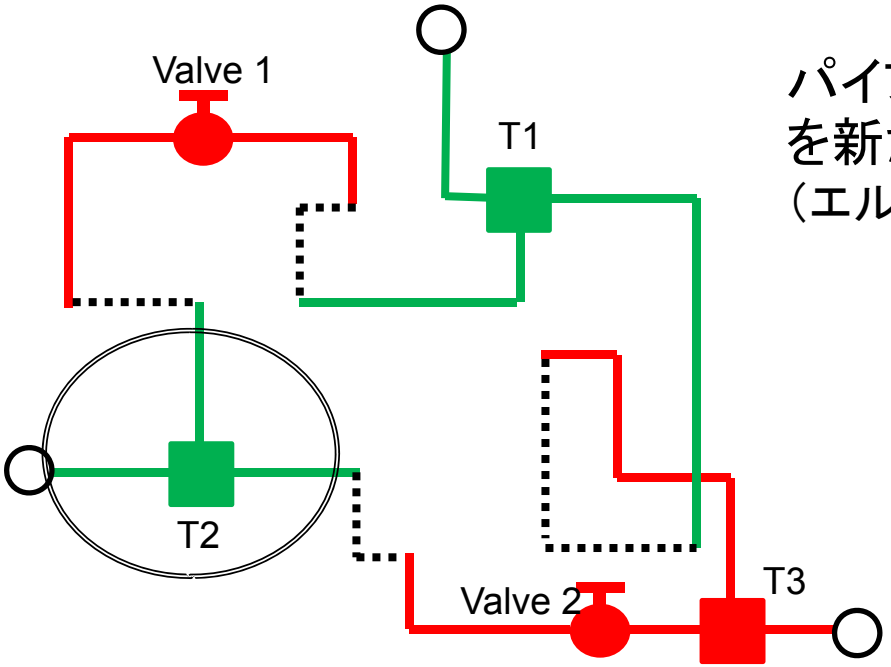
Child



Parent A

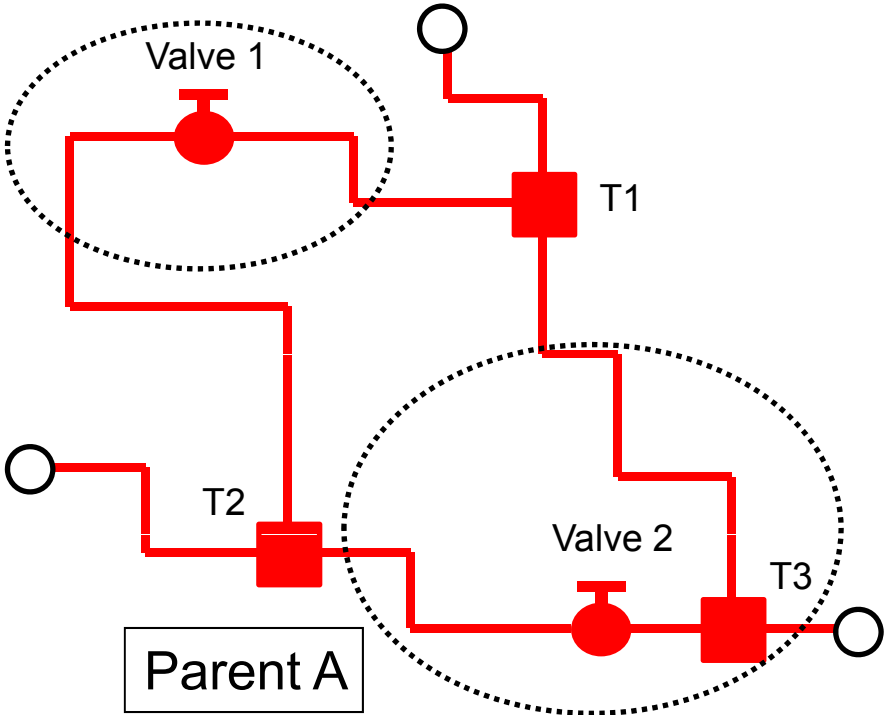


Parent B

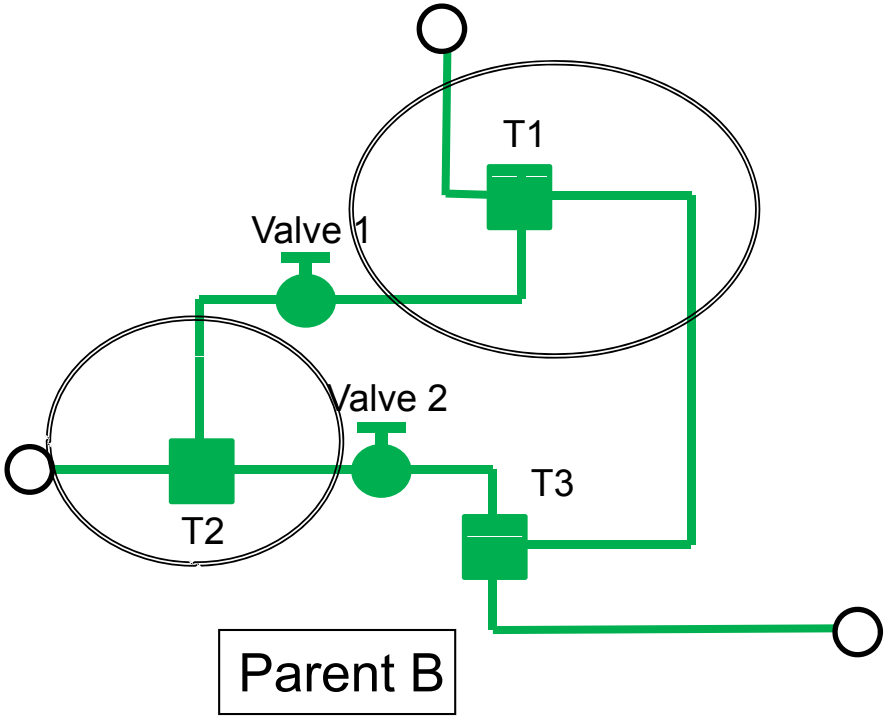


パイプが途切れている部分を新たに繋ぎ直す
（エルボ3個以下で）

Child



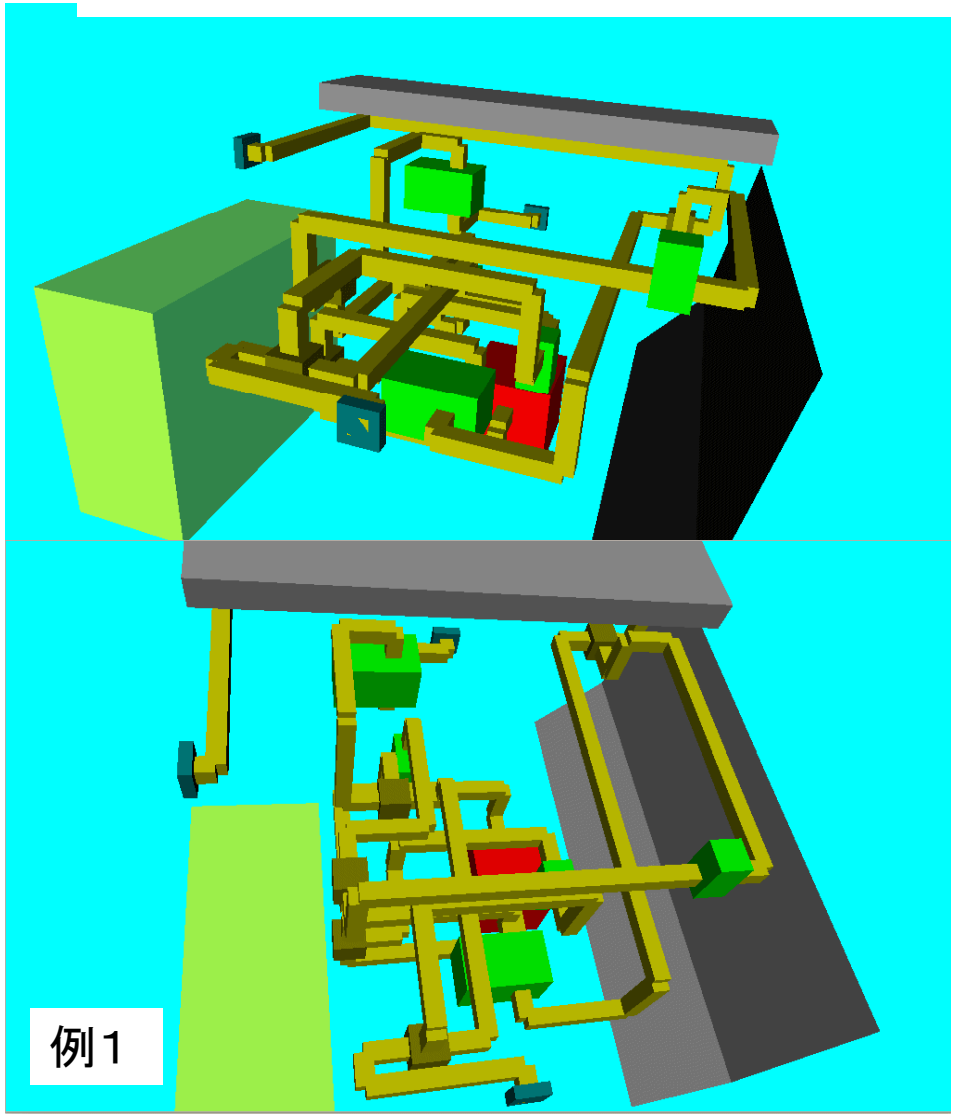
Parent A

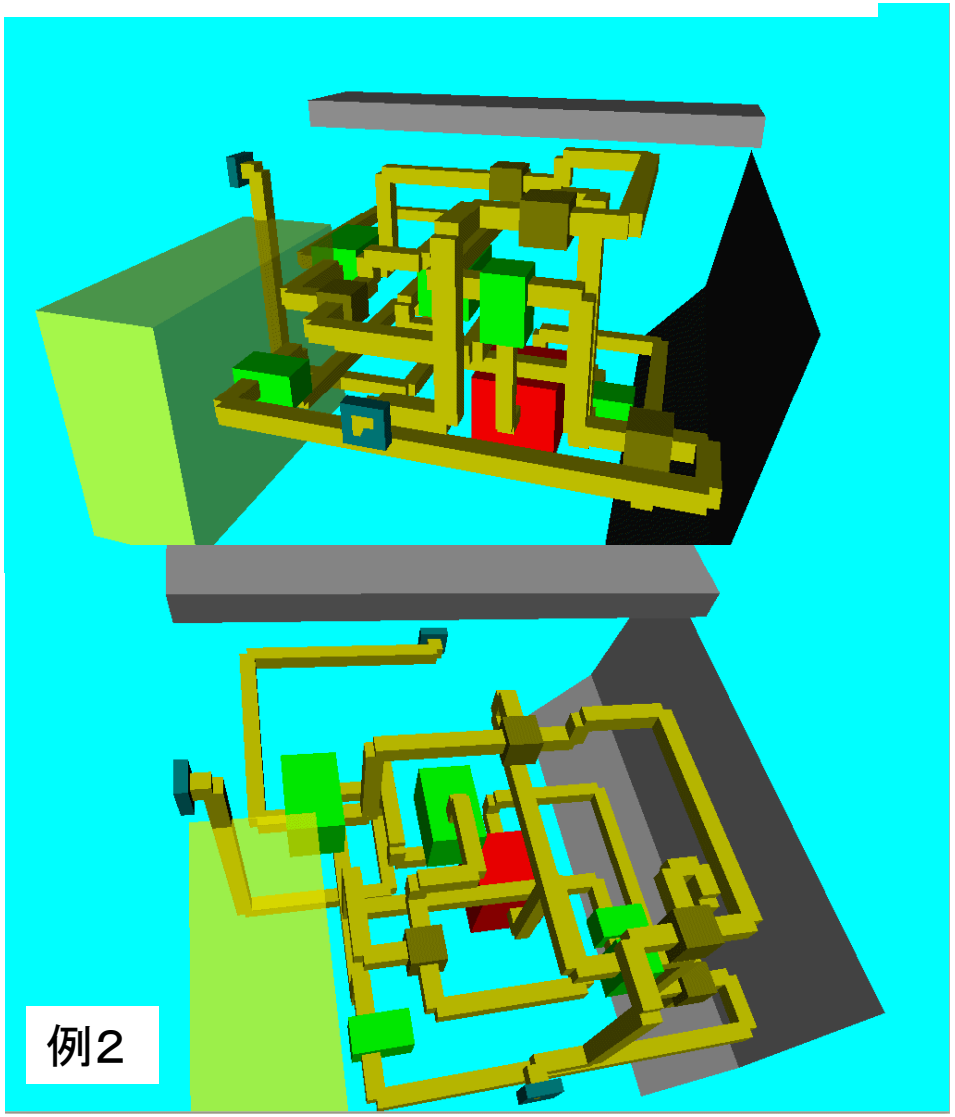


Parent B

遺伝的アルゴリズムで解候補を改善するには、
実行可能な**初期解集団が必要**

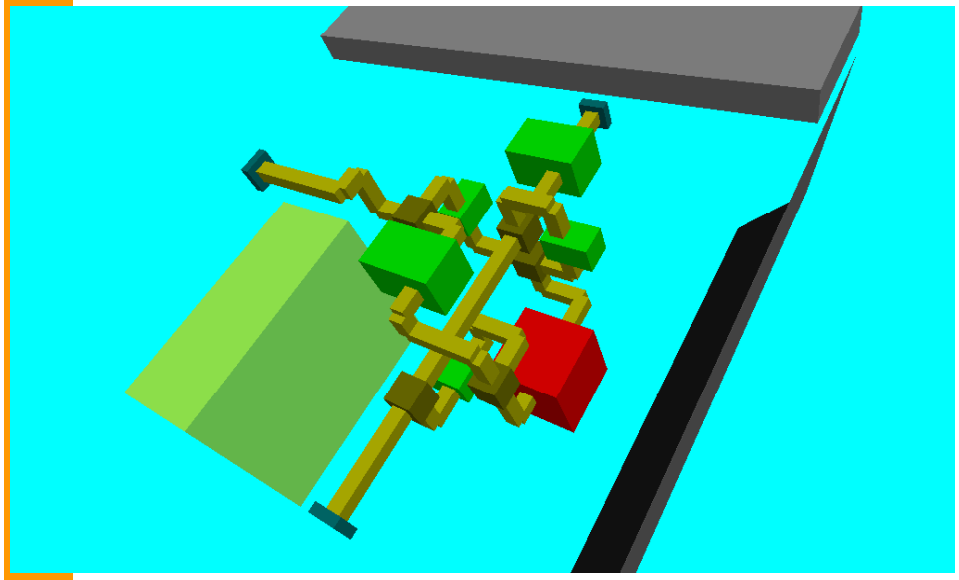
しかし、**ランダムな機器配置**で初期解を生成すると…

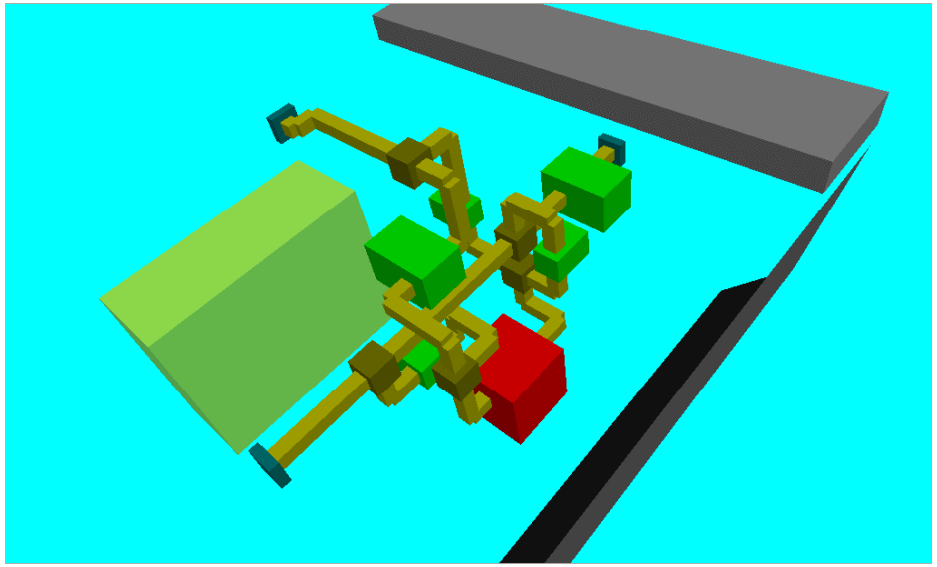




本手法によって得られた設計案

解候補の評価回数 6000
計算時間:ノートPCで約47時間

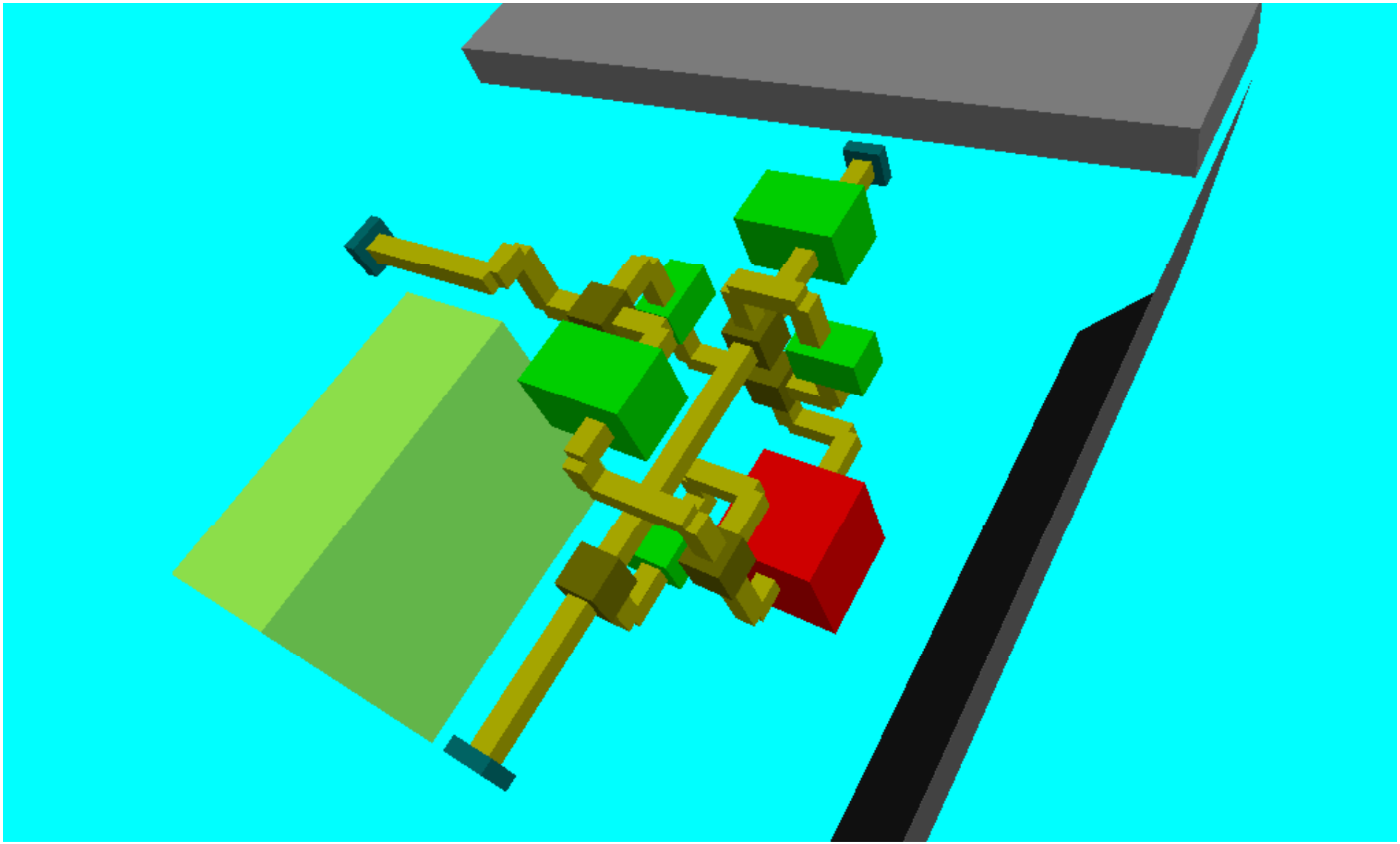




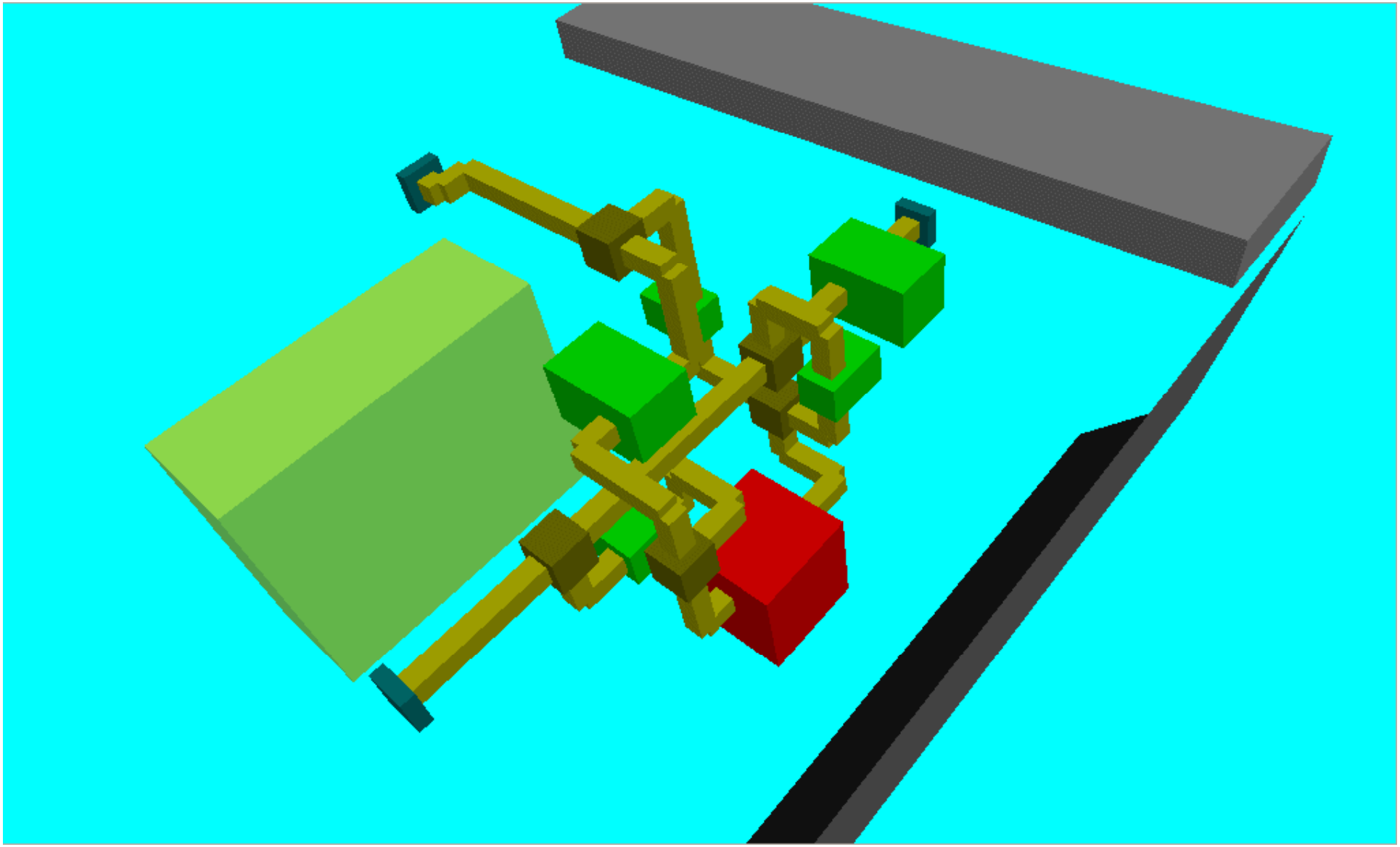
Material Cost	= 2.445
Number of elbows	= 19
Cost of Valve Operationality	= 0

Material Cost	= 2.52
Number of Elbows	= 18
Cost of Valve Operationality	= 0

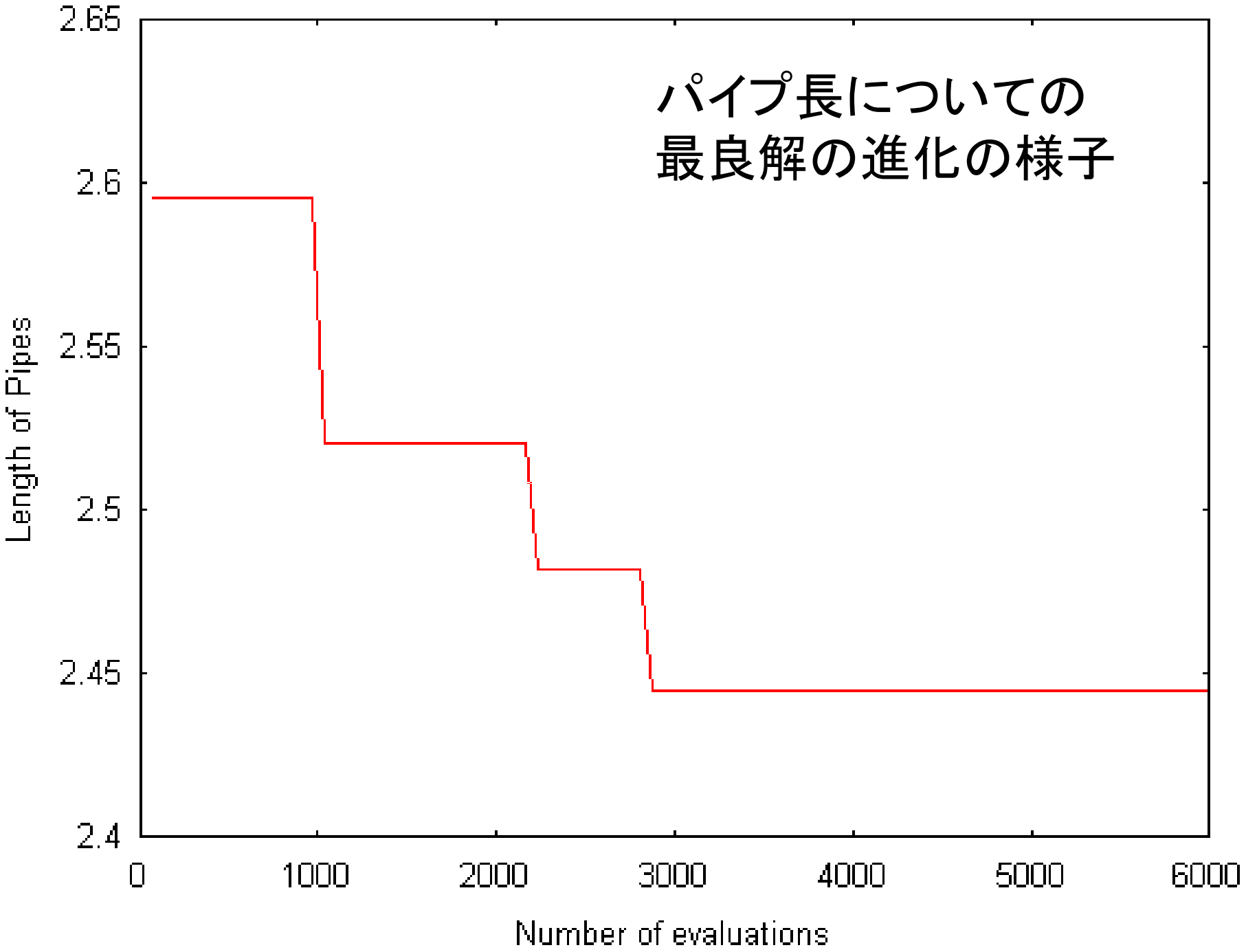
目的関数は3種類: パイプ長・エルボ数・バルブ操作性

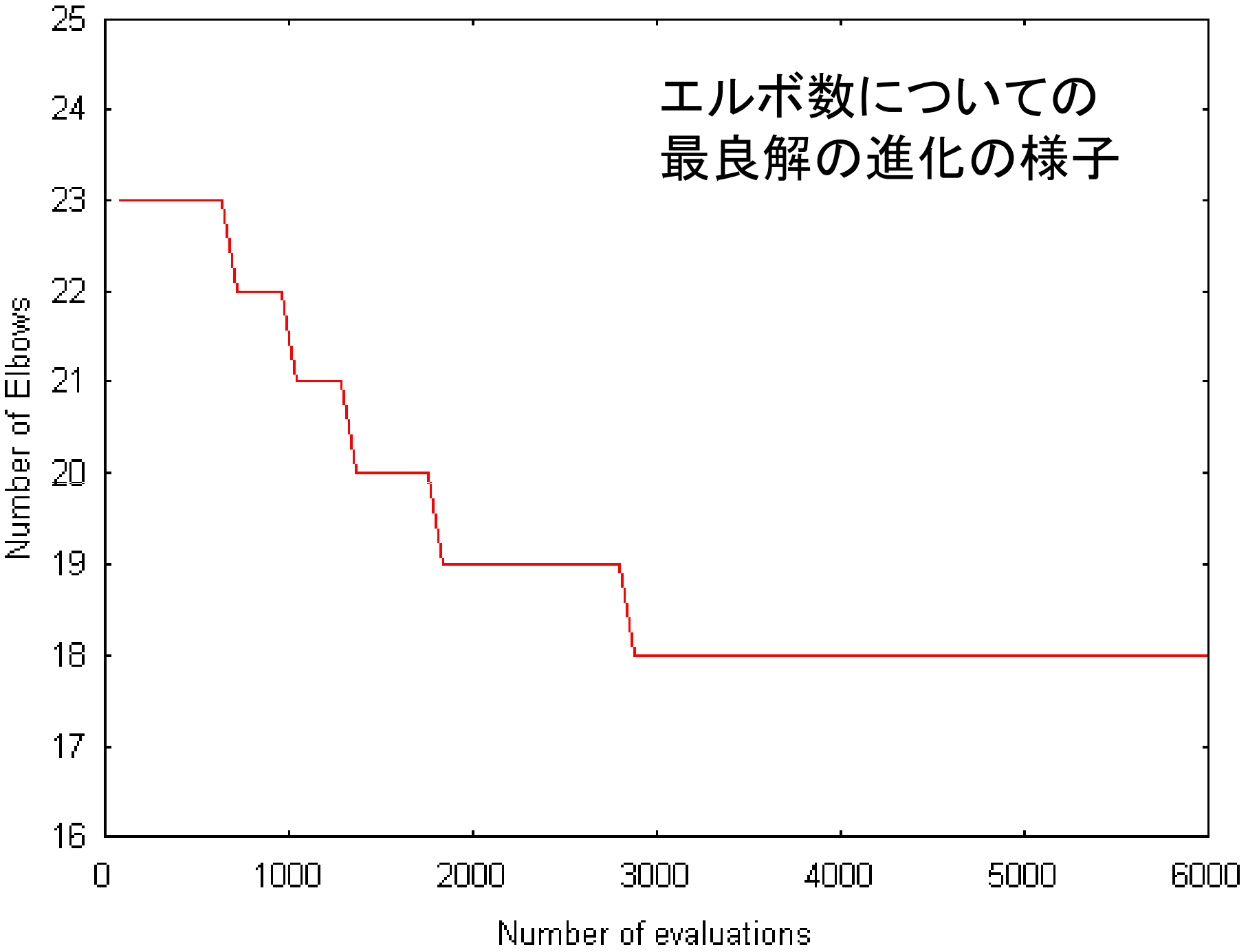


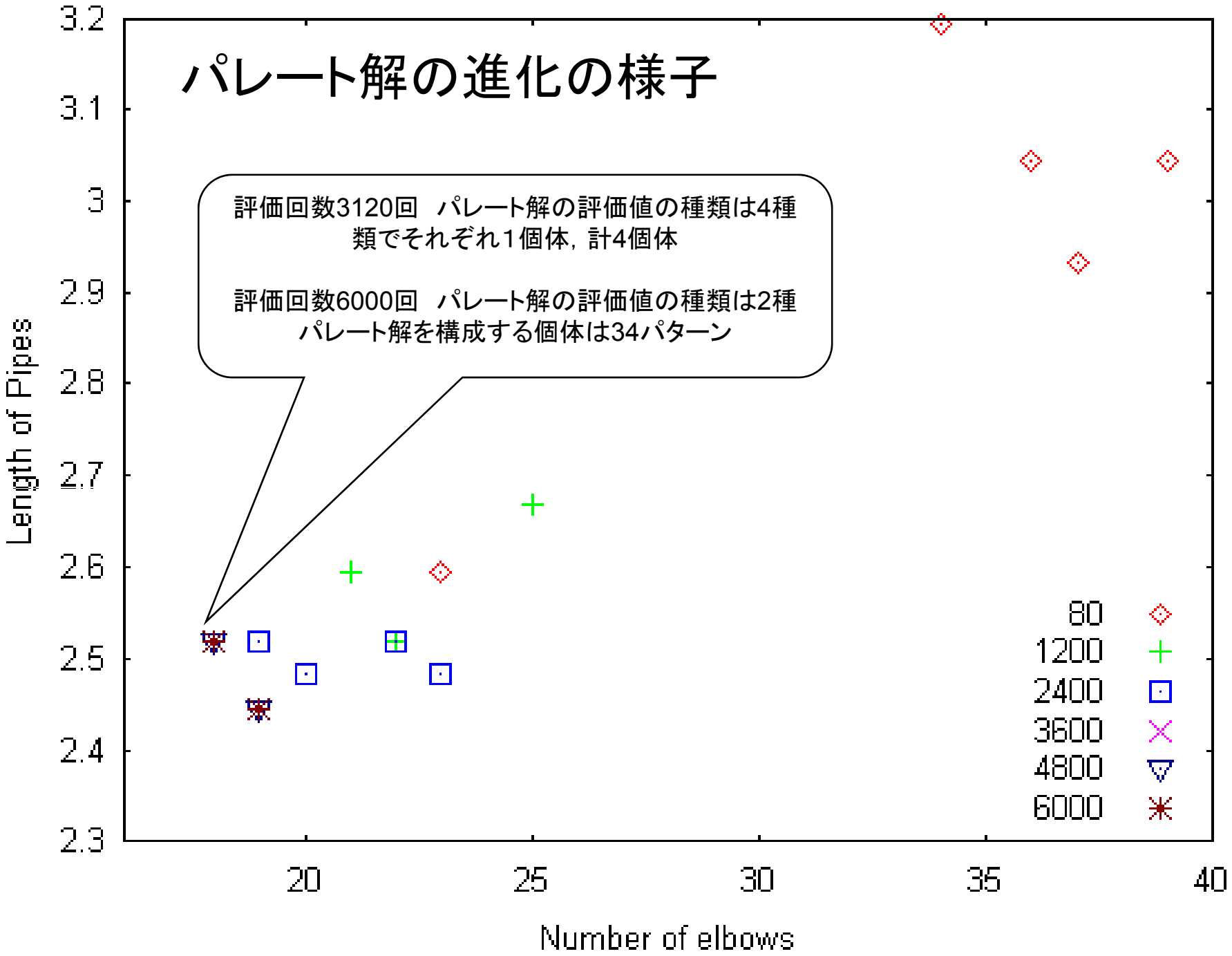
Material Cost = 2.445
Number of elbows = 19
Cost of Valve Operationality = 0



Material Cost = 2.52
Number of Elbows = 18
Cost of Valve Operationality = 0


パイプ長についての
最良解の進化の様子
Length of Pipes
2.65
2.6
2.55
2.5
2.45
2.4
0
1000
2000
3000
4000
5000
6000
Number of evaluations


エルボ数についての
最良解の進化の様子
Number of Elbows
25
24
23
22
21
20
19
18
17
16
0
1000
2000
3000
4000
5000
6000
Number of evaluations


パレート解の進化の様子
評価回数3120回　パレート解の評価値の種類は4種
類でそれぞれ1個体，計4個体
評価回数6000回　パレート解の評価値の種類は2種
パレート解を構成する個体は34パターン
Length of Pipes
Number of elbows
3.2
3.1
3
2.9
2.8
2.7
2.6
2.5
2.4
2.3
20
25
30
35
40
80
1200
2400
3600
4800
6000

参考文献

1）長尾　智晴： 最適化アルゴリズム，昭晃堂 (2000).

2）小林重信：実数値GAのフロンティア
人工知能学会誌Vol.24, No.1, pp.128--143, (2009).

3）小野　功，佐藤　浩，小林　重信：
単峰性正規分布交叉UNDXを用いた実数値GAによる関数最適化，
人工知能学会誌 Vol.14, No.6, pp.1146—1155 (1999).

4）K.Deb and T.Goel. Evolutionary Multi-Criterion Optimization,
Springer, pp.67--81 (2001)

ニッチングの戦略 NSGA2 の説明は以下のURLを参考にした
同志社大学　知的システムデザイン研究室
http://mikilab.doshisha.ac.jp/dia/research/mop_ga/moga/3/3-5-5.html

まとめ

多目的最適化　と　遺伝的多目的最適化手法

【多目的最適化とは？】
1）評価項目（コスト関数）が複数存在する
2）評価値同士を足し合わせることができない
3）評価値のトレードオフ比を決めることも困難

【多目的最適化問題の特徴】
1）最適解は複数（場合によっては無数に）存在し得る　→　パレート解

【多目的最適化問題の解法】
1）トレードオフ比を変えて単目的最適化を繰り返す　←　目的関数2次元程度
2）多目的遺伝的手法：
　・交叉方法：　　単目的の場合と同じ（ただし親個体の選択方法にノウハウ）
　・世代交代方法：　NSGA2　（パレート淘汰戦略＋ニッチング）

「あいまいな評価の数値化」と「多目的最適化問題への定式化」　により
ベテラン技術者の扱う設計作業を自動化できる可能性
ただし過度な多目的化はほとんどの解がパレート解となってしまうため避けるべき

【練習問題】2017.10.27

以下のようにA～Pに示される
LppとBが全て同じ船型の候補が存在するとき、
NSGA2を用いて**6個の解候補**を選択せよ

学籍番号
氏　名