# GAのスケジューリング問題への応用

平成30年　4月25日

富山県立大学　杉山桃香

# 発表内容

- 研究の背景・目的
- スケジューリング問題
- 遺伝的アルゴリズムの構成
- 初期収束と近傍モデル
- まとめ

# 1.研究背景・目的

## 研究背景

課題

組合せ最適化問題を解く

理論的には

厳密解を得ることは可能

問題

計算の手数と時間が必要

必要

近似解を求めること

## 研究目的

GAを適用して解く

# 2.スケジューリング問題とは

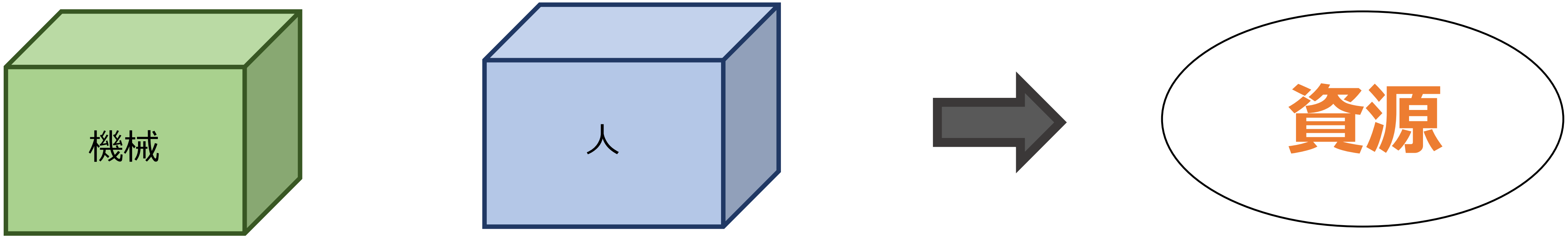
いくつかのやらなければならない仕事(作業)があるときにどういう順番で仕事を進めるのが最も速いか、あるいは都合が良いか、その順番を求める問題

(例)

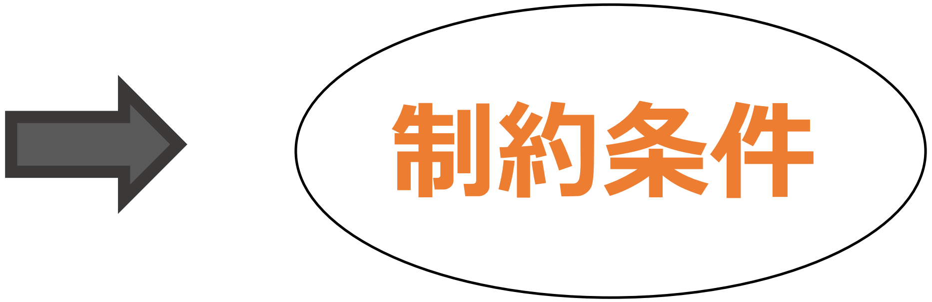
・授業の時間割を決める問題

・従業員の勤務表を作る問題

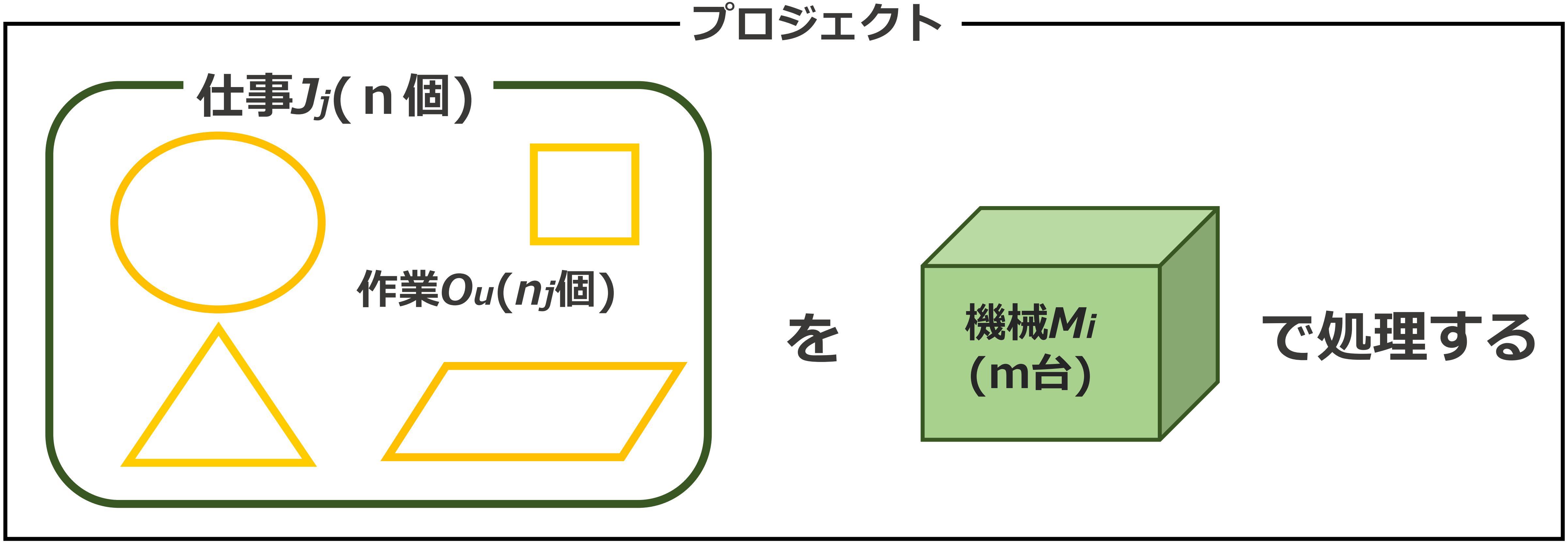
・製造工場の最適作業工程の自動作成




- ✓ 同時に使用してはいけない
- ✓ 機械の台数に限りがある
- ✓ 次の作業に入るまでに時間を空ける
- ✓ 次の作業まで時間を空けてはいけない



# 問題の記述




**制約条件**

- ✓ 作業$O_u$は$u$の増加順に処理されなければならない($O_1 \rightarrow O_2 \rightarrow O_3$)
- ✓ 各作業$O_u$は機械$M_{\mu n}$上で分割なしで所要時間$P_u$で処理される($\mu_n$は作業$O_u$が処理される機械番号)

このプロジェクトの完了時間$C_{max}$

最小となるように

各機械上で作業が開始される時刻を決定

このような問題を

<u>ジョブショップ型スケジューリング問題</u>と呼ぶ

# GAを適用する準備

明確にする

数理計画問題に定式化する

プロジェクト

⬇

混合整数計画問題(MIP)

決定変数

同じ機械で処理される
作業間の順序

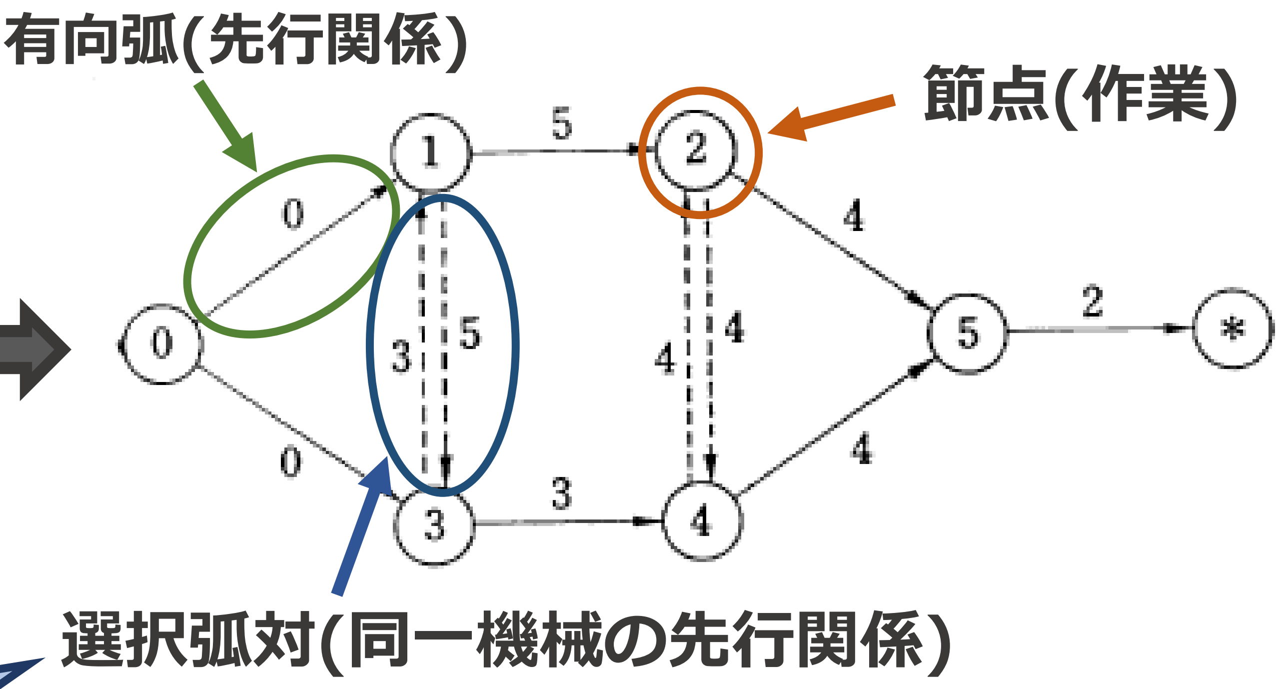
⬇

<u>各作業の開始時刻が決定</u>

# 選択グラフによる問題の表現

表 1　プロジェクトの例

| 仕事 | 作業（機械，処理時間） | | 先行仕事 |
|---|---|---|---|
| $J_1$ | $O_1(M_1, 5)$ | $O_2(M_2, 4)$ | — |
| $J_2$ | $O_3(M_1, 3)$ | $O_4(M_2, 4)$ | — |
| $J_3$ | $O_5(M_2, 2)$ | — | $J_1, J_2$ |




**選択弧対の解消**で**決定変数**が決まる

**全て解消→１つのスケジュール完成**

このときの０から＊までの
クリティカル・パスの長さ

‖

このプロジェクトの完了時間*Cmax*

注)このときグラフに閉路(サイクル)が
生じないようにする

# スケジューリング問題に対するGAの特徴

## 厳密解法

<目的>
質の良い解を得る

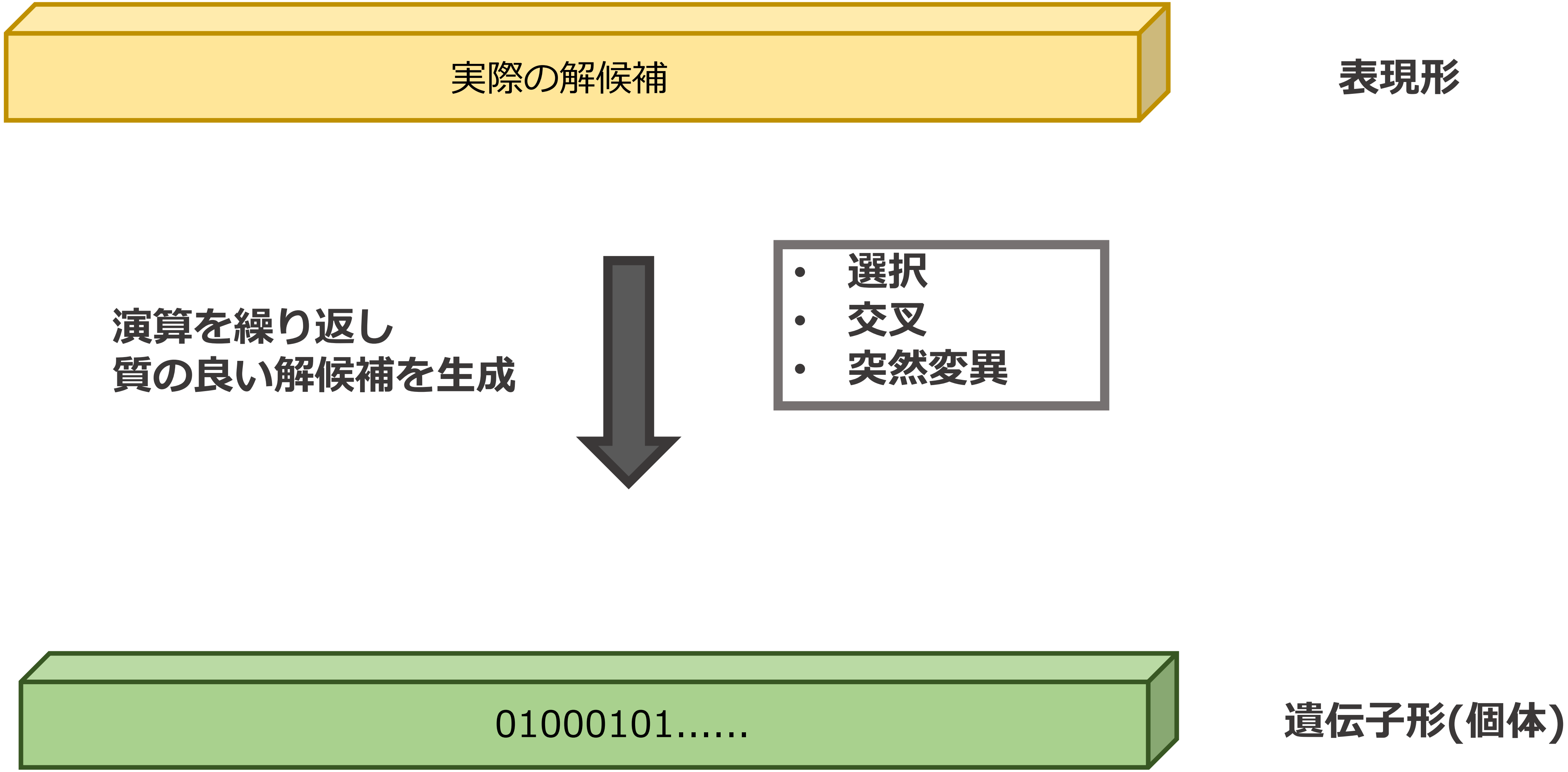
<特徴>
・広い解空間をくまなく探索
・時間がかかる

矛盾

## 近似解法

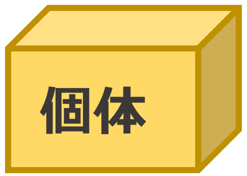
<目的>
短時間で近似解を求める

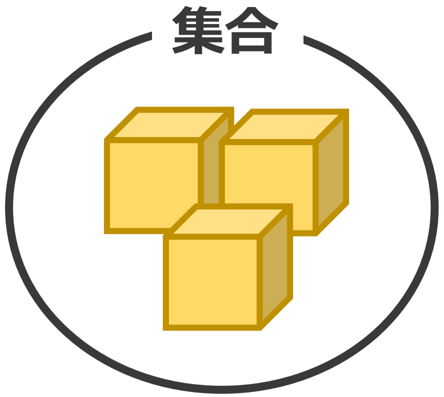
<特徴>
・先験情報によって
探索空間を絞り込む


実際の解候補
表現形
演算を繰り返し
質の良い解候補を生成
選択
交叉
突然変異
01000101……
遺伝子形(個体)

このように、GAは近似解法(逐次改善法など)の範疇に属する解法とみれる

# しかし




<GAの特徴>

◆ 個体の集合を対象として適応度を高めていく

◆ 個体(記号列)全体のみでなく部分列も評価できる

◆ 適応度を高めるのに役立つ部分列を同時並列的に増やせる

# 3.遺伝的アルゴリズムの構成

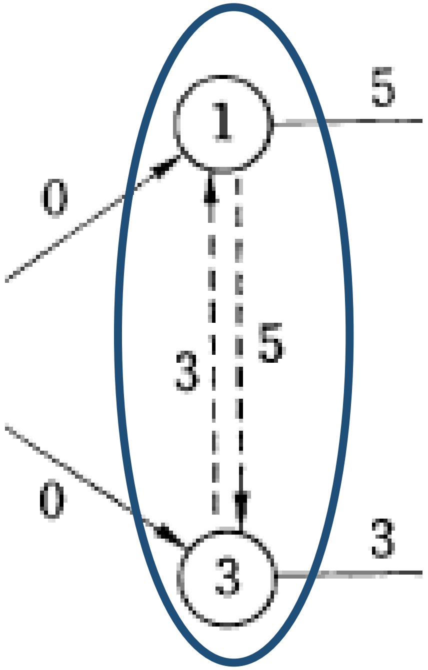
重要

問題をどのように表現するか

↓

選択グラフに基づいて個体の遺伝子形を定める方法について

# (1)個体の遺伝子形




選択弧対の解消方向

→0と１で表す

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| $d_{14}$ | $d_{17}$ | $d_{47}$ | $d_{25}$ | $d_{26}$ | $d_{56}$ | $d_{38}$ |

個体の遺伝子形

節点1と４の間の選択弧対

（0のとき）
番号の大きい節点から小さい節点への選択弧を消す

（1のとき）
番号の小さい節点から大きい節点への選択弧を消す

# (2)表現形への変換規則

<問題>

**選択弧対を独立に解消** → **閉路を含む個体**

**交叉や突然変異**

**そこで**

記号( 0 /1)の値を選択弧対の優先解消方向とみて

遺伝子形から表現形(実際のスケジュール)へ変換するときの規則を定める。

## 変換規則

0.すべての選択弧対を未解消にしておく

1.現在までの選択弧対の解消状況を示すグラフに対して

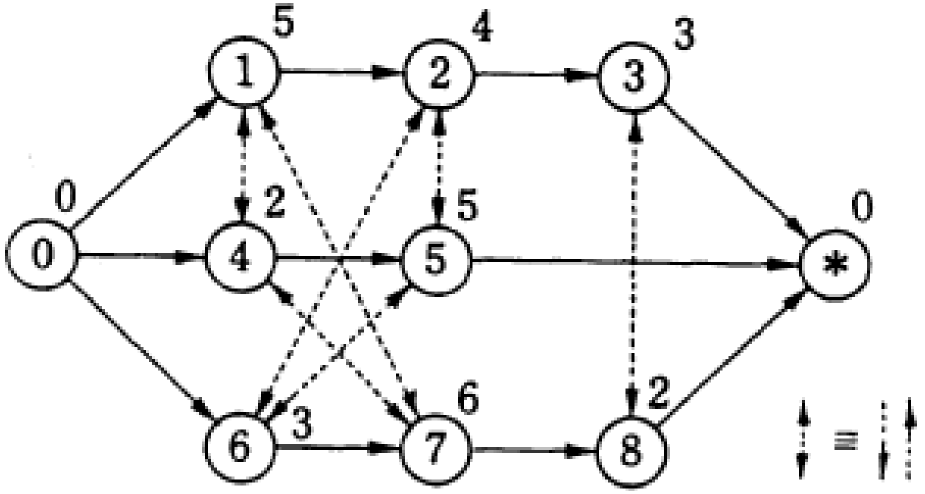
未解消の選択弧対をすべて除去したグラフで、クリティカルパスを実行

→1つのスケジュールが定まる

2.コンフリクトが発生していなければ、終了節点の最早開始時刻として*Cmax*を求める。

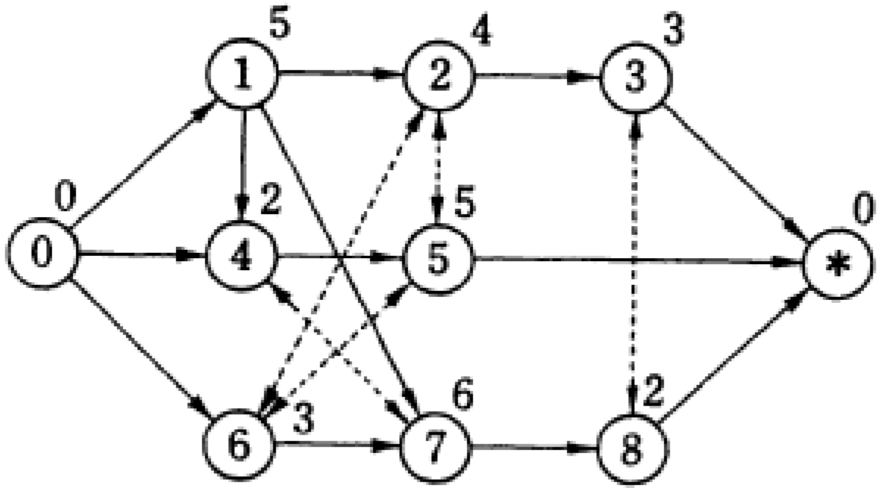
コンフリクトが発生していたら、最も早い時刻に生じている選択弧対を参照して解消

0



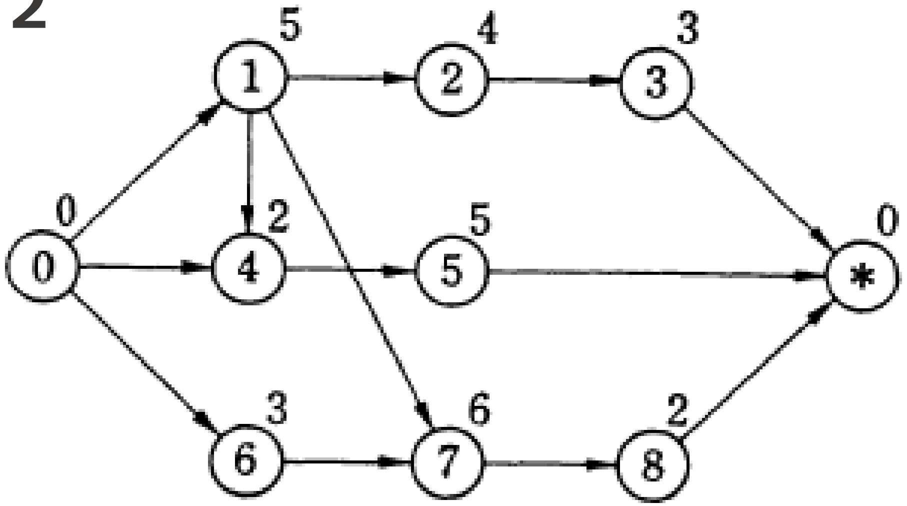
（a）選択グラフ

1



（b）現在の選択弧対解消状況を示すグラフ

2



（c）未解消の選択弧対が除去されたグラフ

１つのスケジュールが得られる

# (3)適応度の計算

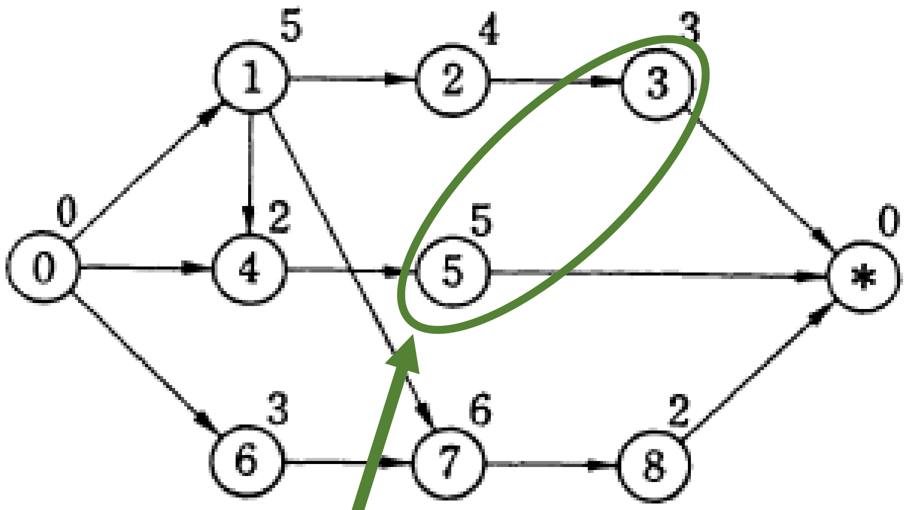
*Cmax*に対応した適応度*f*(*Cmax*)を求める。

$$f(C_{max})=\frac{C_{max}^{u}-C_{max}}{C_{max}^{u}-C_{max}^{l}}\times 100 \qquad (1)$$

$C_{max}^{u}$ → *Cmax*の上限値(全作業の処理時間の和)

$C_{max}^{l}$ →*Cmax*の下限値(選択弧対をすべて除去したグラフのクリティカルパスの長さ)

# (4)遺伝子形における冗長性




（c） 未解消の選択弧対が除去されたグラフ

参照されていない

遺伝子形として異なるものが
同じスケジュールに含まれることがある。

閉路を発生させない工夫の一つ

# (5)計算例と考察

**問題例**

- **先行関係のない８つの仕事**

✓各仕事に含まれる作業数：2～4

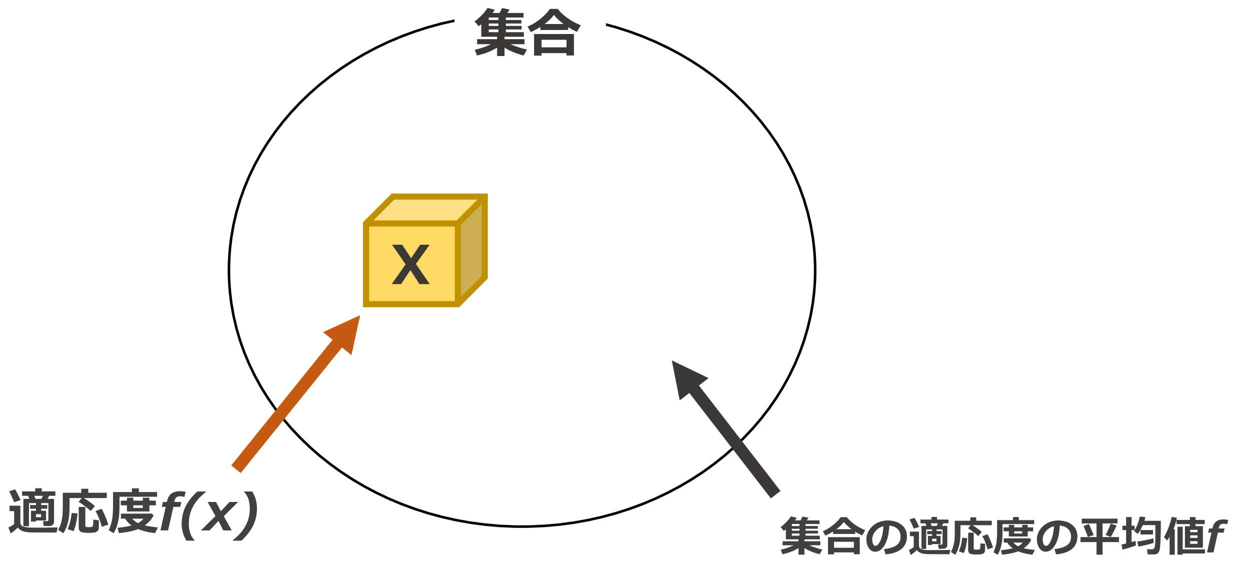
✓作業を処理する機械数：*$M_1$~$M_2$*の6台

✓各作業の処理時間：1～12

一様乱数によって決定した100個の問題に対してGAを適用

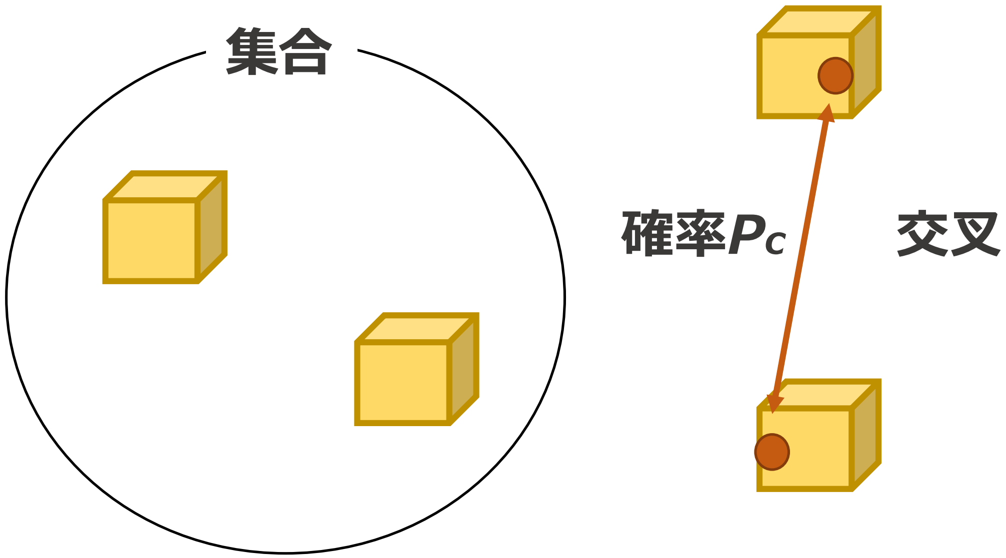
# 遺伝演算子の説明

●選択：個体Xの適応度と集合の適応度の平均値から期待値を求める




**期待値＝*f(x)/f***

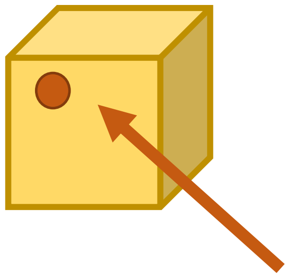
● 交叉：2個体選びランダムに2ヶ所組み換える




（確率$P_c$ ＝0.6）

● 突然変異：ランダムに記号を反転させる

1ビットランダムに選ぶ

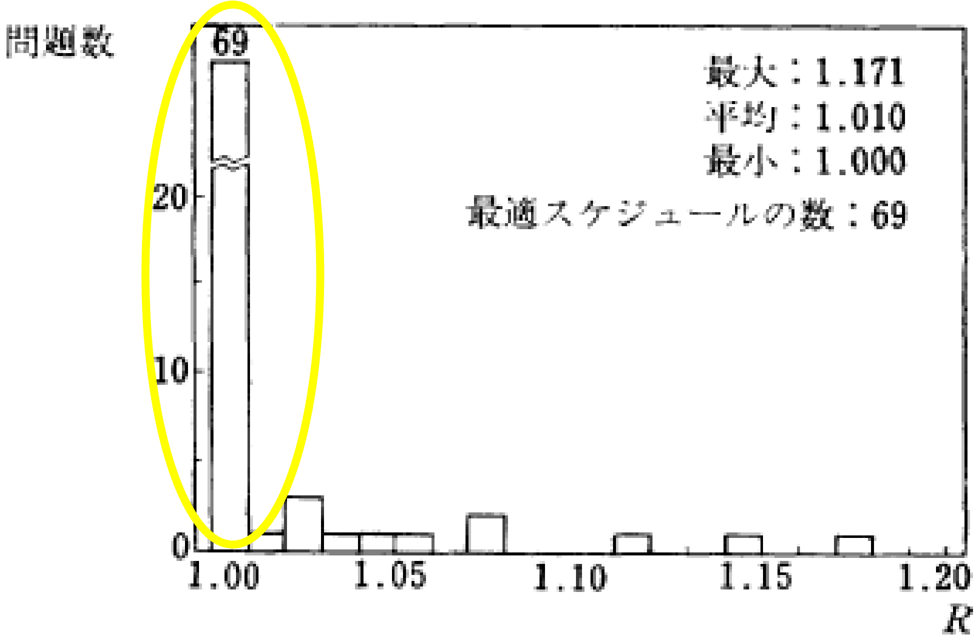


確率$P_m$ で
突然変異

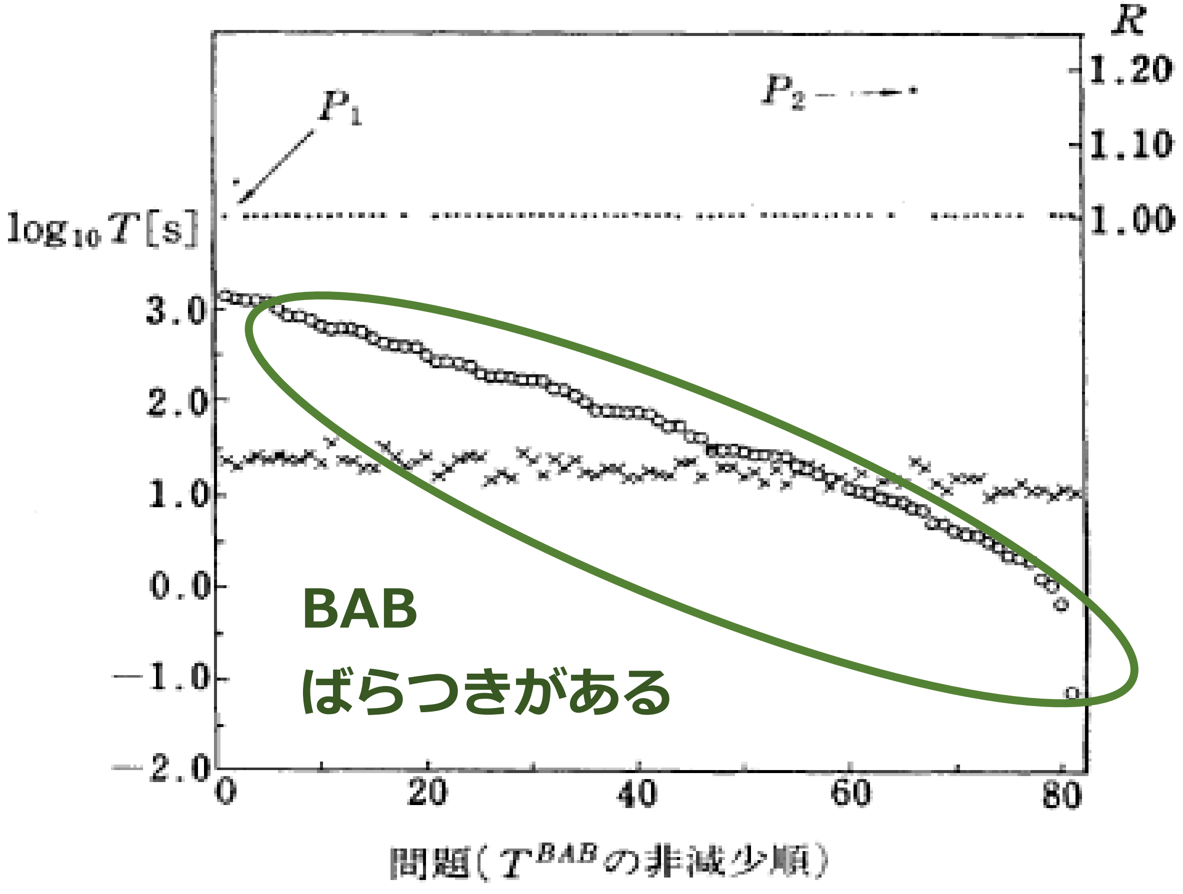
（確率$P_m$ ＝0.005）

## GAによるスケジュールの良さR

$$R=\frac{C_{\max}^{GA}}{C_{\max}^{BAB}}$$ （＝最小値に対する比）




(a) 解の良さ $R$ のヒストグラム

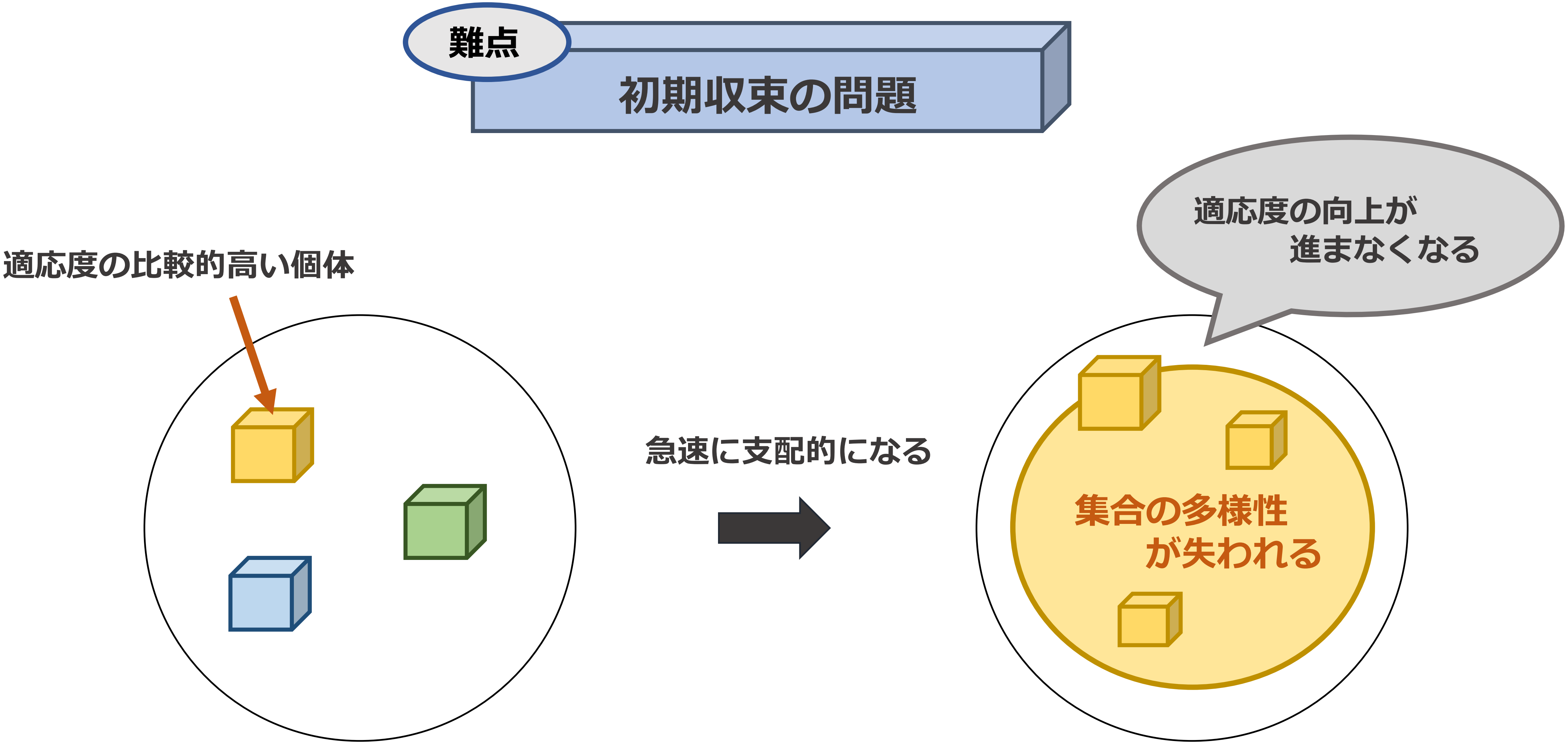



×および○は，それぞれGAおよびBABによる結果

(b) 問題ごとの計算時間と $R$

図 4　GA と BAB による結果の比較

# 4.初期収束と近傍モデル



対策

# 近傍モデルの概念を導入

個体

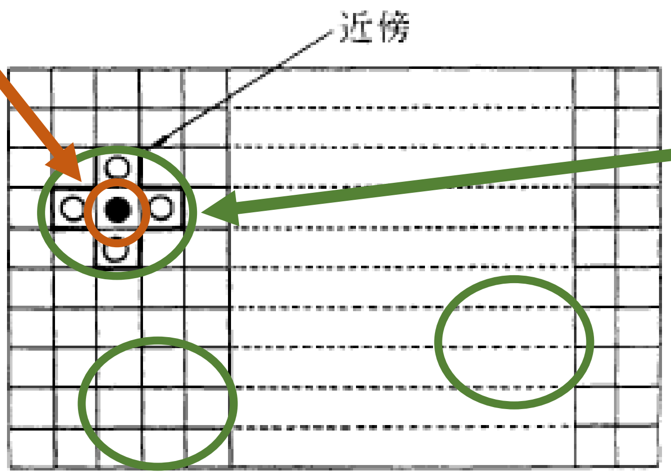



図 5　近傍モデルにおける個体配置と近傍

選択は近傍ごとに行う

並列に実行

並列計算機上で実行しやすい

アルゴリズムの高速化を実現

**近傍の定義**

各個体からのハミング距離が1以内であること

# まとめ

✓GAの特徴を簡単なジョブショップ問題を解いてみることで勉強できた

✓近傍モデルを導入することにより多様性を保持することができた

**考察**

個体の配列方法や近傍の定義の仕方を変えると結果にどの程度影響するのか

さまざまな問題に対する利点と欠点を見極めるための試行錯誤が必要