

仮想サーバ配置問題のための 最適解近似アルゴリズム

工学研究科基盤工学専攻
情報システム工学コース
福本研究室

1175085 播田直紀

背景：サーバ統合

- × 企業で増えすぎたサーバ群を仮想化
- × 1／3以下の高性能な大容量サーバに集約
- × プライベートクラウドを実現

サーバ統合を行う際には、コストなどを見積もる上で必要な大容量サーバ台数を予め見極めておく必要がある



データセンターなどの大容量サーバ



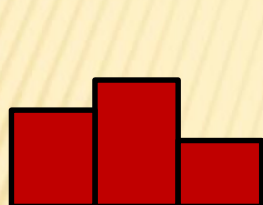
仮想化されたサーバを大容量サーバにどのように配置するかが重要

仮想サーバ配置問題

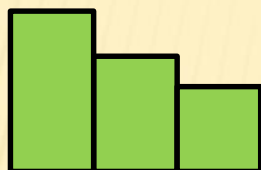
- × 各仮想サーバは計算資源使用量が見積もられる
 - + CPU
 - + メモリ
 - + ディスク
 - + ネットワーク など
- × 統合される仮想サーバの計算資源使用量の合計値が大容量サーバの閾値を超過しないように配置
- × 使用する大容量サーバの台数が最小となる組み合わせを求める組合せ最適化問題

仮想サーバ配置問題の例題

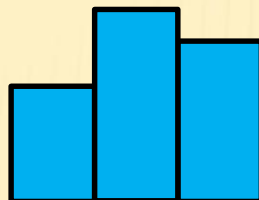
- ✖ 計算資源数 3、統合対象の仮想化サーバ数 5



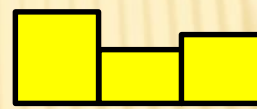
サーバ1



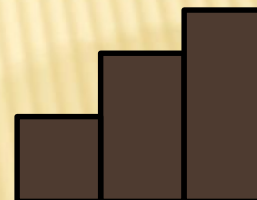
サーバ2



サーバ3

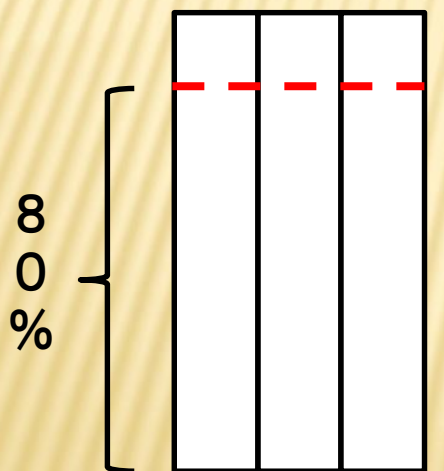


サーバ4



サーバ5

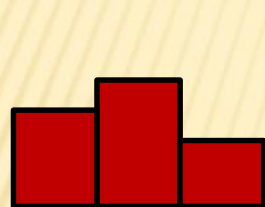
- ✖ 統合先となる大容量サーバ各計算資源の閾値を 80%とした場合



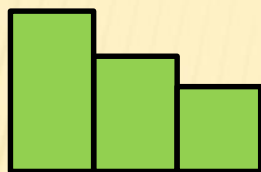
大容量サーバ
1台目

仮想サーバ配置問題の例題

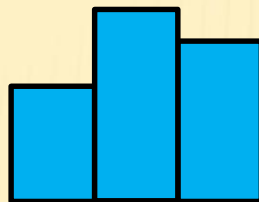
- ✖ 計算資源数 3、統合対象の仮想化サーバ数 5



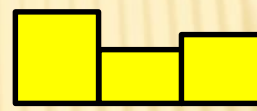
サーバ1



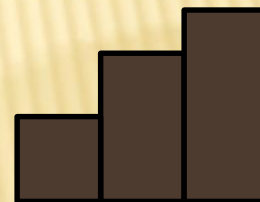
サーバ2



サーバ3

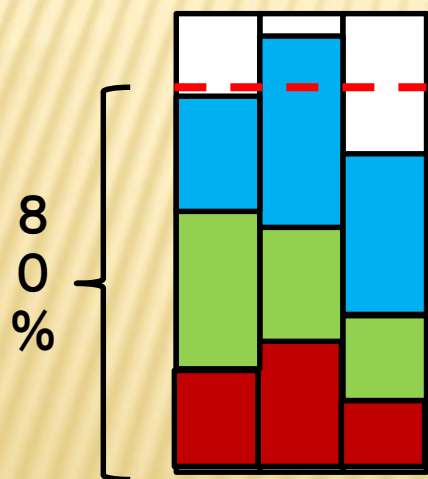


サーバ4

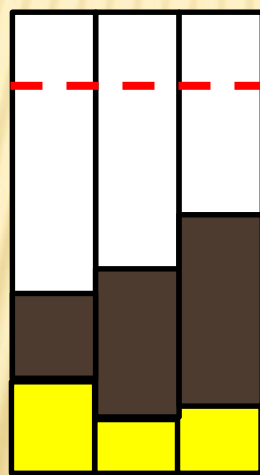


サーバ5

- ✖ 統合先となる大容量サーバ各計算資源の閾値を80%とした場合



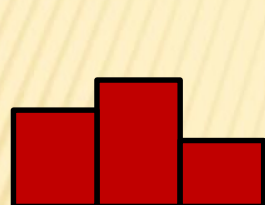
大容量サーバ
1 台目



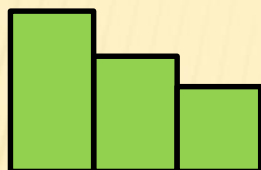
大容量サーバ
2 台目

仮想サーバ配置問題の例題

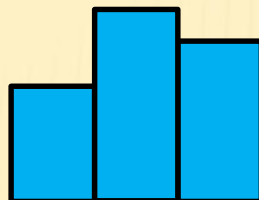
- ✖ 計算資源数 3、統合対象の仮想化サーバ数 5



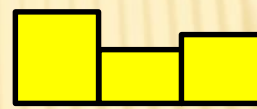
サーバ1



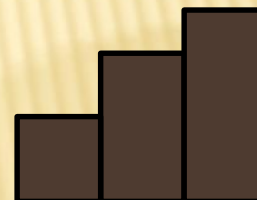
サーバ2



サーバ3

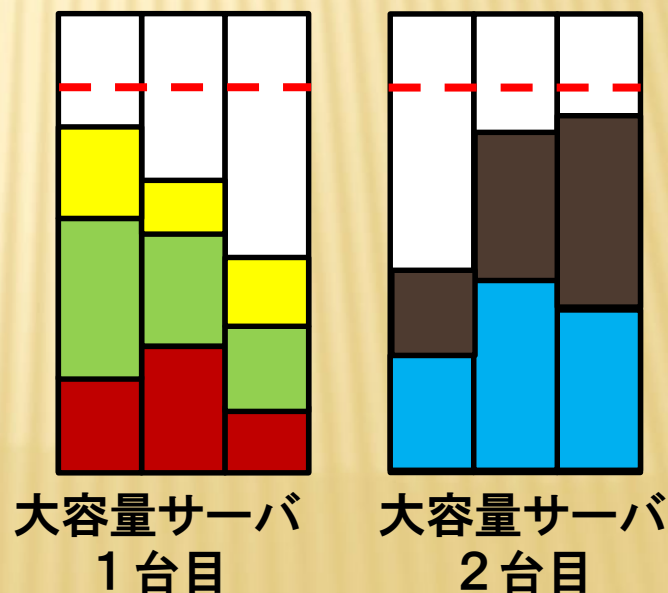
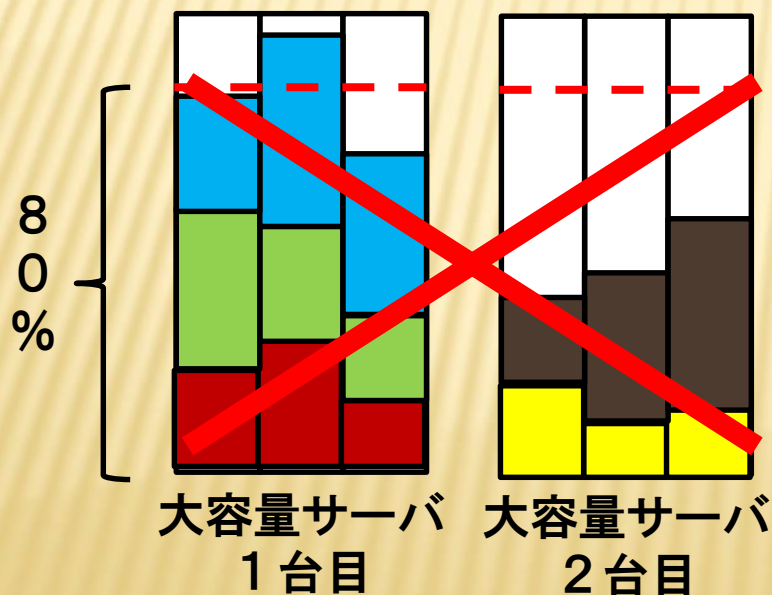


サーバ4



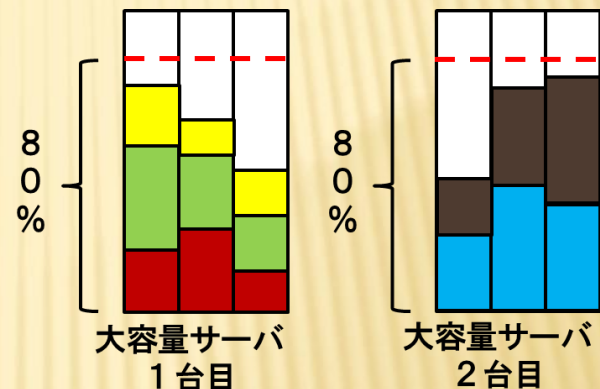
サーバ5

- ✖ 統合先となる大容量サーバ各計算資源の閾値を80%とした場合



解の表現方法

- × 例：[[1, 2, 4], [3, 5]]
 - 1 台目に1, 2, 4の仮想サーバを配置
 - 2 台目に3, 5の仮想サーバを配置
 - 必要な大容量サーバ台数は2



- × 解を仮想サーバを詰め込む順番として順列で表現
 - + 解個体 [1, 2, 4, 3, 5] → [[1, 2, 4], [3, 5]]
 - + リストの順に、F F 法を用いて詰め込み評価
- × F F 法
 - + アイテムが箱に詰められるかどうかを添字番号順に調べ、可能な箱に詰める動作を繰り返す
 - + 詰められなければ箱を増やしそれに詰める

厳密解法による計算の困難性

- × 仮想サーバ配置問題で求められる要件
 - + 仮想サーバの計算資源数は4～6前後
 - + 統合対象となる仮想サーバ数は数百から千台前後
 - + 計算時間はエンジニアによる現場での設計や提案書の作成、複数回実行することを考えると数分以内が望ましい
- × 仮想サーバ配置問題はNP完全な問題

数百から千台前後の統合対象を持つ仮想サーバ配置問題の場合、厳密解法では時間内に解を得ることができない



より短時間で有効な解を算出できるヒューリスティックなアルゴリズムにより近似解を求める

ヒューリスティックなアルゴリズム

計算資源数の少ない仮想サーバ配置問題で良い結果を示した改良型 F F D

局所探索による独自の改善法を持つランダムサンプリングや
遺伝的アルゴリズム

問題サイズが大きな問題でも改善法による効果が得られているか
確かめるために大域探索のみを行うランダム探索

× 今回実装したアルゴリズム

- + 改良型ランダムサンプリング (R A M 2)
- + 改良型 F F D (F F D R)
- + ランダム探索 (R S)
- + 遺伝的アルゴリズム (G A)

この 4 つの解法を適用し、比較を行う

ランダムサンプリング

✕ 局所探索アルゴリズムの一つ

[4, 3, 2, 1, 5, 6, 0]

ランダムサンプリング

× 局所探索アルゴリズムの一つ

[4, 3, 2, 1, 5, 6, 0]

選択した要素から順列パターンを作成



[4, 3, 2, 1, 5, 6, 0] = ?

[4, 3, 2, 6, 5, 1, 0] = ?

[4, 3, 1, 2, 5, 6, 0] = ?

[4, 3, 1, 6, 5, 2, 0] = ?

[4, 3, 6, 2, 5, 1, 0] = ?

[4, 3, 6, 1, 5, 2, 0] = ?

最も評価の高い
個体を残す

局所探索アルゴリズムは問題サイズの大きな問題では近似解や
厳密解にたどり着くために多くの時間がかかる

ランダムサンプリング改良案

× ランダム探索による解候補群の抽出を行う

はじめにランダム探索を行い、その時点で得られた同じ最良値を持つ解候補群を作成

〔解候補 1, 解候補 2, 解候補 3, ...〕

それぞれにランダムサンプリング手法を適用

最良値が更新されれば、またその時点で得られた同じ最良値を持つ解候補群を作成

〔解候補 1, 解候補 2, 解候補 3, ...〕

それぞれにランダムサンプリング手法を適用

終了条件を満たすまで繰り返す

各アルゴリズムの特徴

	特徴
RAM2	ランダム探索で解候補群として抽出した複数箇所に対して、 順列組み合わせパターンを用いた局所探索を行う
FFDR	アイテムを詰める順番が解の評価に大きく関わるため、箱に入らなかったアイテムを箱の増加要因となるアイテムとして保持し、もう一度詰め直す際に優先的に詰めることで解の改善を行う
RS	ランダムに解個体を生成し、最も評価の良い個体を残す
GA	生成した個体集団に対して選択、交叉、突然変異などの遺伝的操作を用いて解の改善を行う 交叉により1世代で複数箇所の局所探索を同時に行うことができる

比較・検証（１）

- × 統合対象サーバ数を500、1000の問題に以下の解法を適用して
試行平均と試行標準偏差の推移で比較を行った
 - + 改良型ランダムサンプリング（RAM2）
 - + 改良型FFD（FFDR）
 - + ランダム探索（RS）
 - + 遺伝的アルゴリズム（GA）
- × 計算資源を以下の5つとする
 - + CPU性能使用量
 - + メモリ使用量
 - + ディスク性能使用量
 - + ディスク容量使用量
 - + ネットワーク性能使用量
- × 閾値はすべて80%
- × 乱数の種の数と実行時間
 - + 乱数の種：100種類
 - + 実行時間：300秒間

比較・検証用パラメータ

× RAM2

- + ランダム探索回数：1000
- + 選択される要素数：3

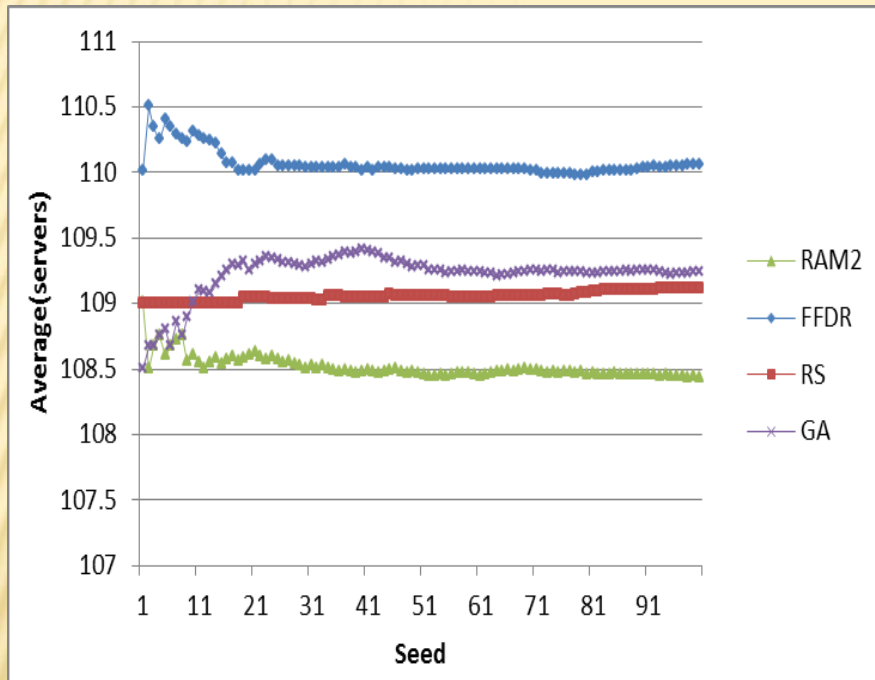
× FFD R

- + MAX r：統合対象サーバ数と同数

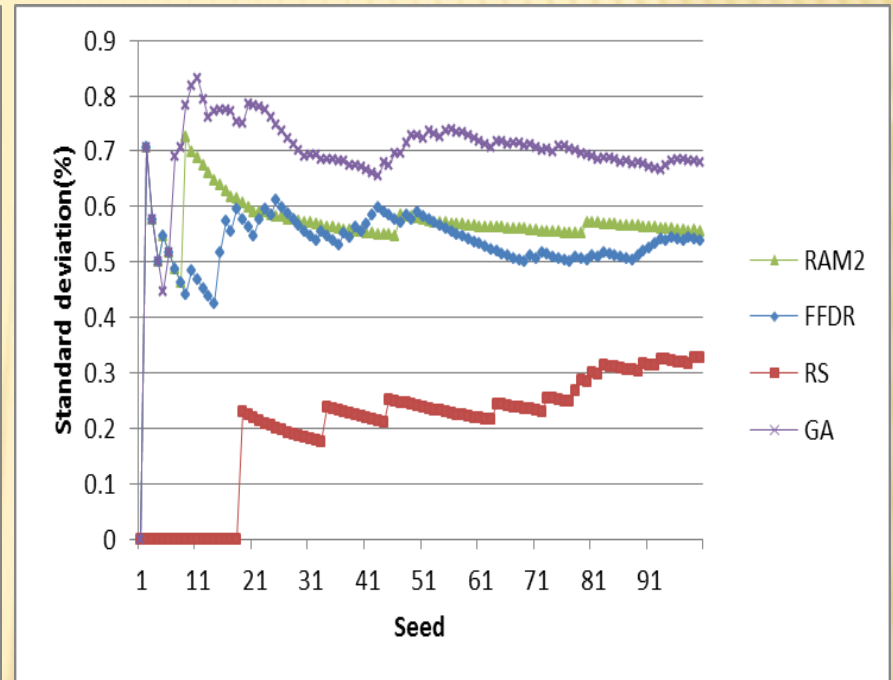
× GA

- + 集団サイズ：100
- + 初期集団生成：FF法
- + 選択方法：ルーレット選択
- + 交叉方法：PMX
- + 突然変異：ランダムな2点間をシャッフル
- + 交叉確率：90%
- + 突然変異確率：10%

比較検証結果（問題サイズ500）



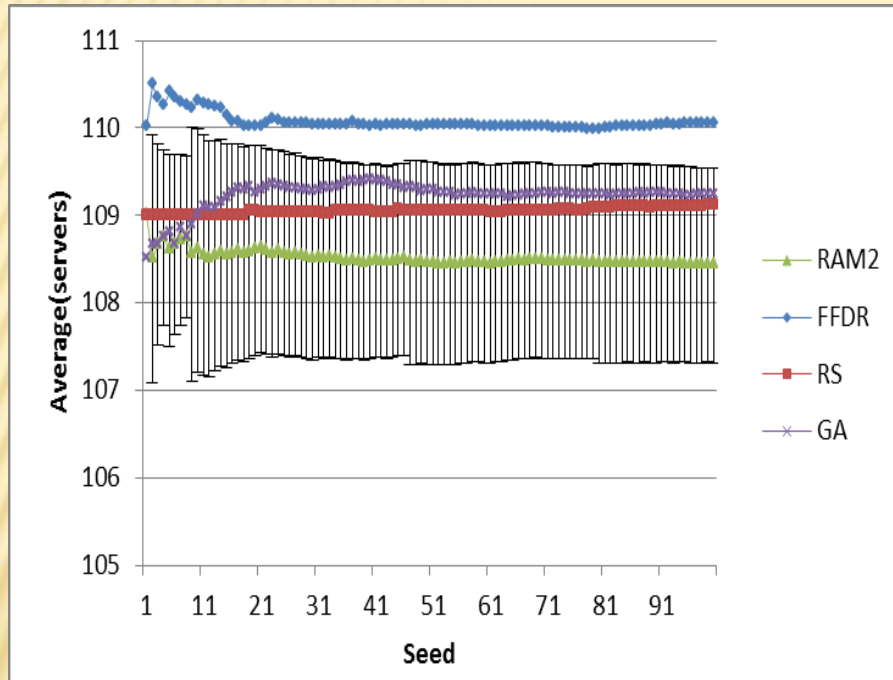
試行平均の推移



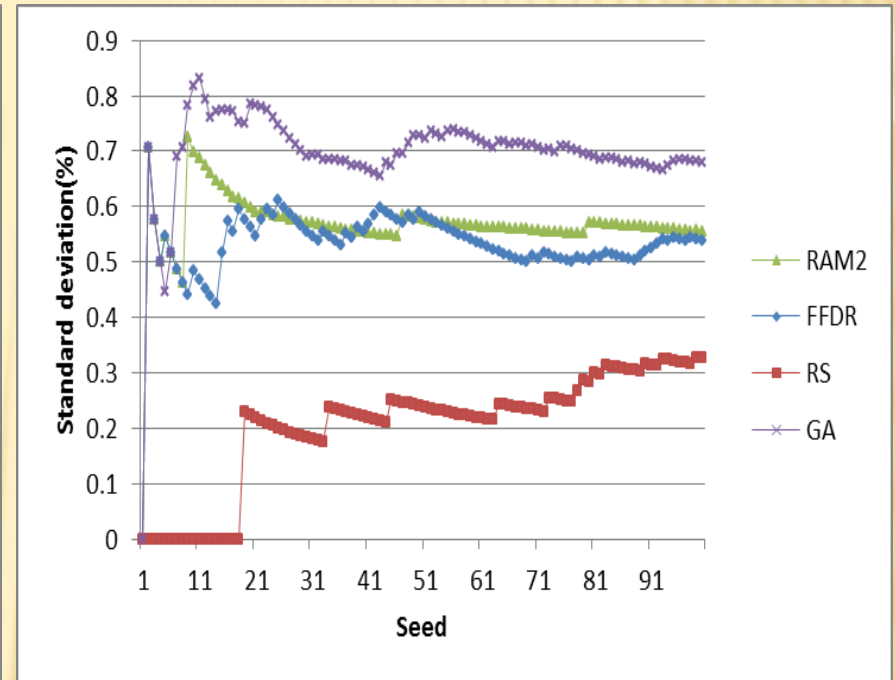
試行標準偏差の推移

- ✕ データが正規分布に従う場合、 $\text{平均値} \pm 2\sigma$ 以内に95.4%の解が存在することになる

比較検証結果（問題サイズ500）



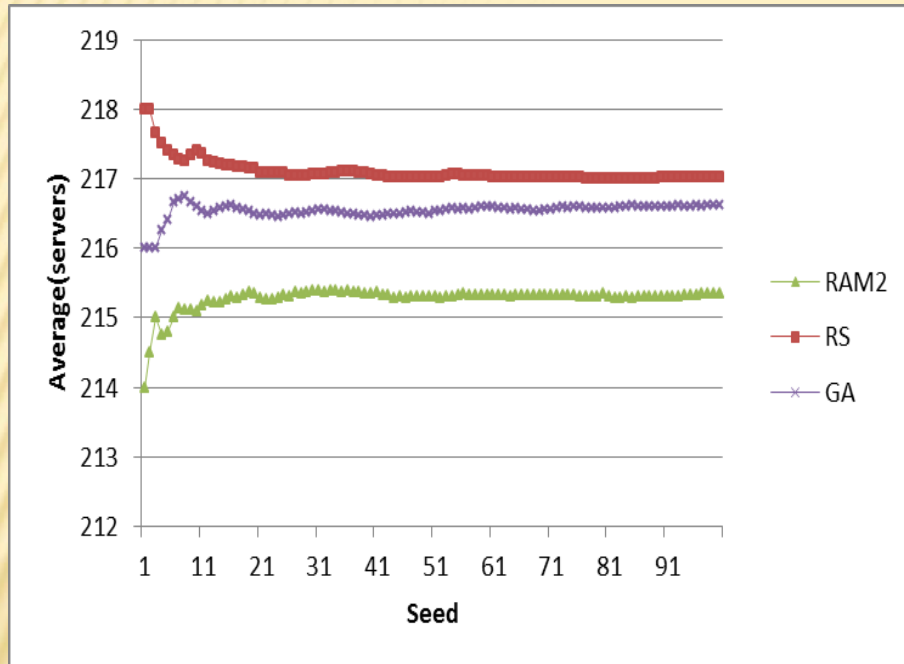
試行平均の推移



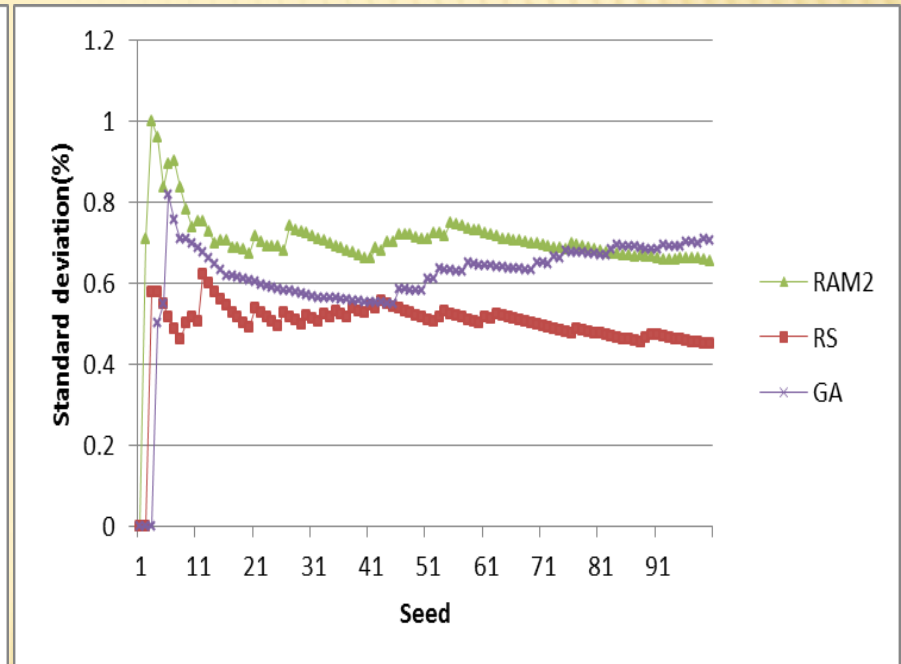
試行標準偏差の推移

- ✖ データが正規分布に従う場合、平均値 $\pm 2\sigma$ 以内に95.4%の解が存在することになる
- ✖ RAM2が他のアルゴリズムに比べて完全に良いとは言えない

比較検証結果（問題サイズ1000）



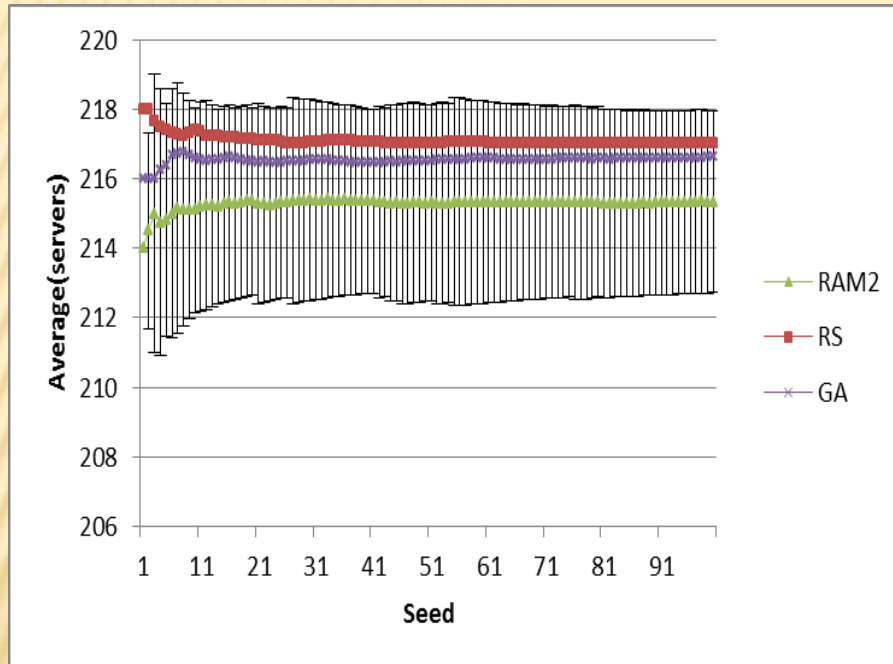
試行平均の推移



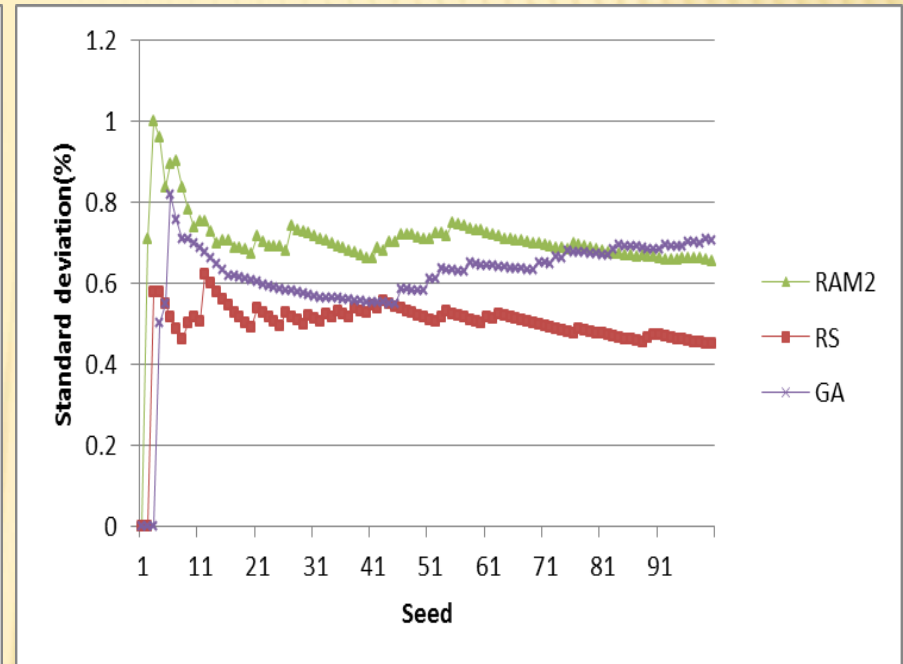
試行標準偏差の推移

- × F F D Rは1回の実行に3500秒前後かかり、300秒以内に有効な解を出力できなかったためRAM2、RS、GAのみ表示

比較検証結果（問題サイズ1000）



試行平均の推移



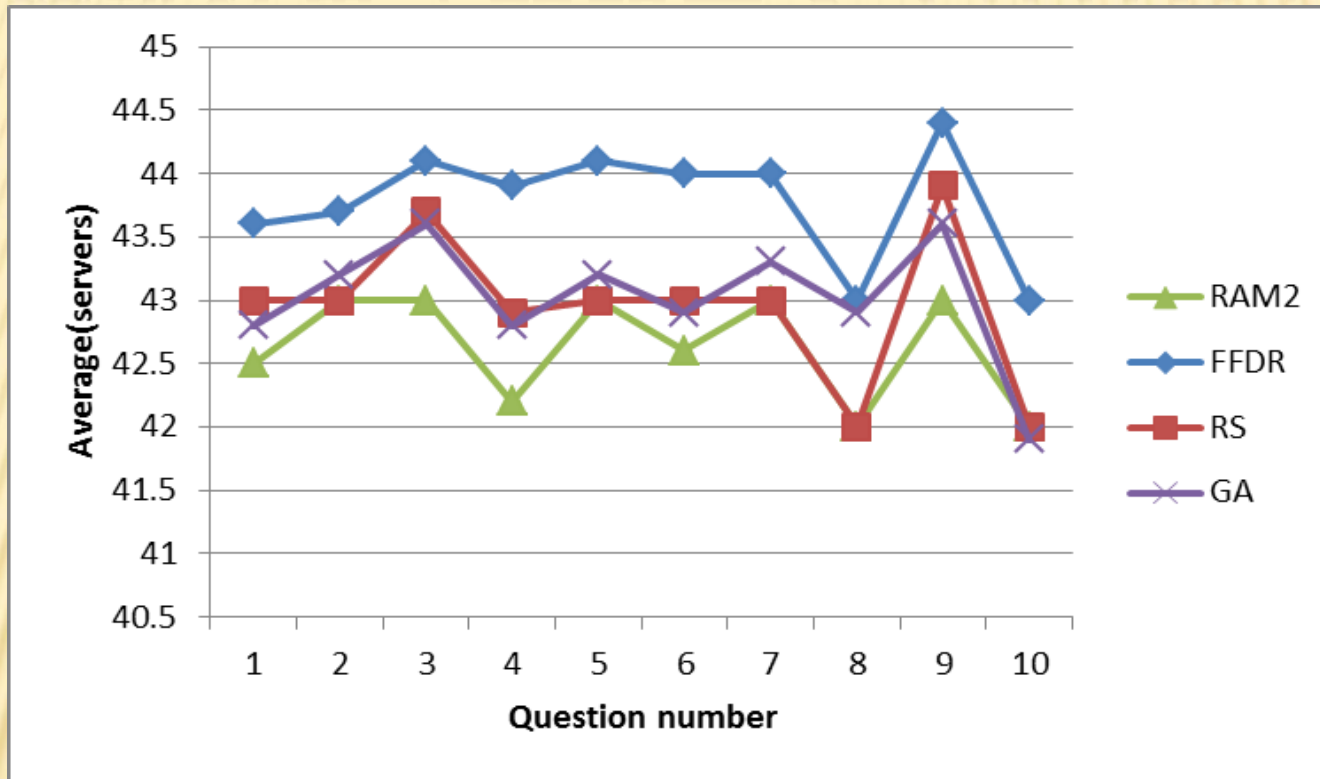
試行標準偏差の推移

- ✖ FFD Rは1回の実行に3500秒前後かかり、300秒以内に有効な解を出力できなかったためRAM2、RS、GAのみ表示
- ✖ やはり問題サイズ500と同じくはっきりとした差は示せず、RAM2が他のアルゴリズムに比べて完全に良いとは言えない

比較・検証（２）

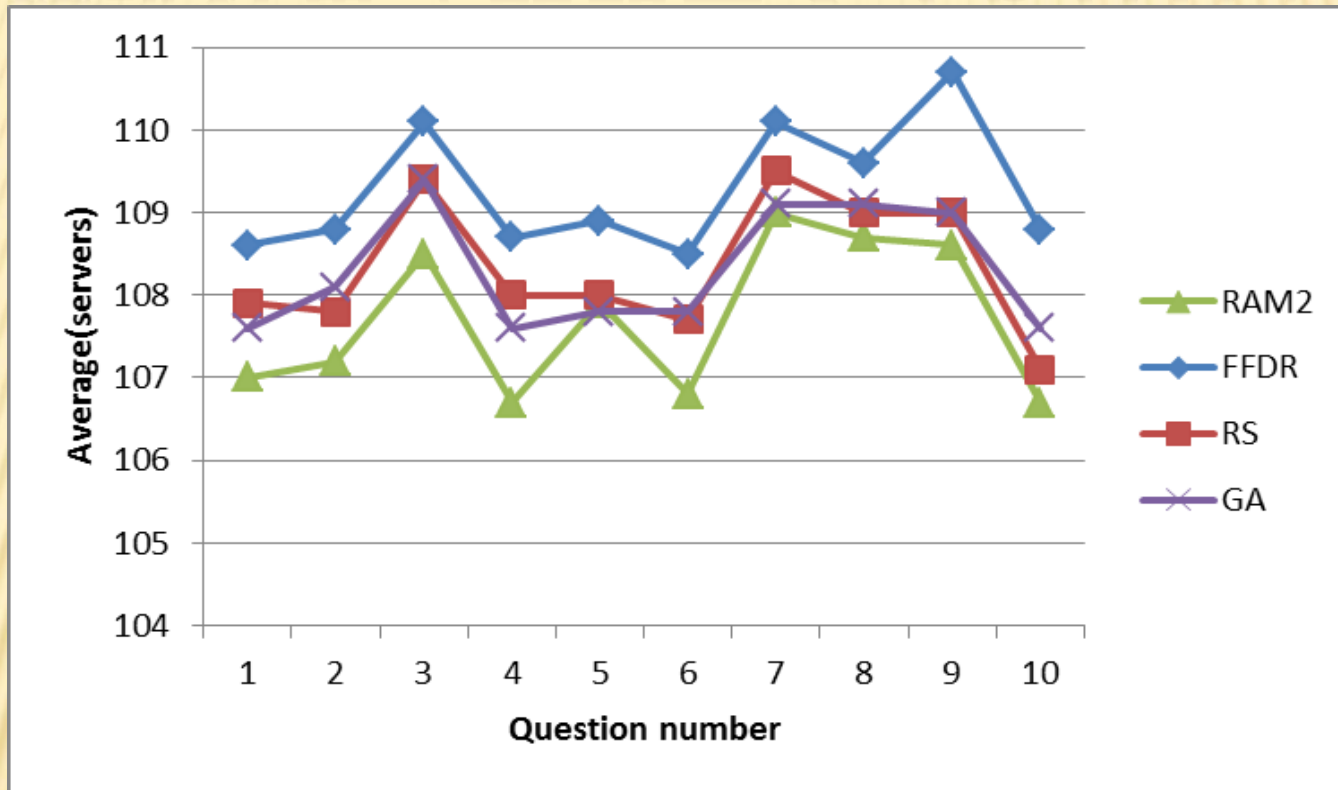
- × 統合対象サーバ数を 200、500、1000 の問題に以下の解法を適用して平均値の比較を行った
 - + 改良型ランダムサンプリング（RAM2）
 - + 改良型FFD（FFDR）
 - + ランダム探索（RS）
 - + 遺伝的アルゴリズム（GA）
- × 計算資源を以下の5つとする
 - + CPU性能使用量
 - + メモリ使用量
 - + ディスク性能使用量
 - + ディスク容量使用量
 - + ネットワーク性能使用量
- × 閾値はすべて80%
- × 乱数の種の数と実行時間、問題数
 - + 乱数の種：10種類
 - + 実行時間：300秒間
 - + 問題数：10問

比較検証結果（問題サイズ200）



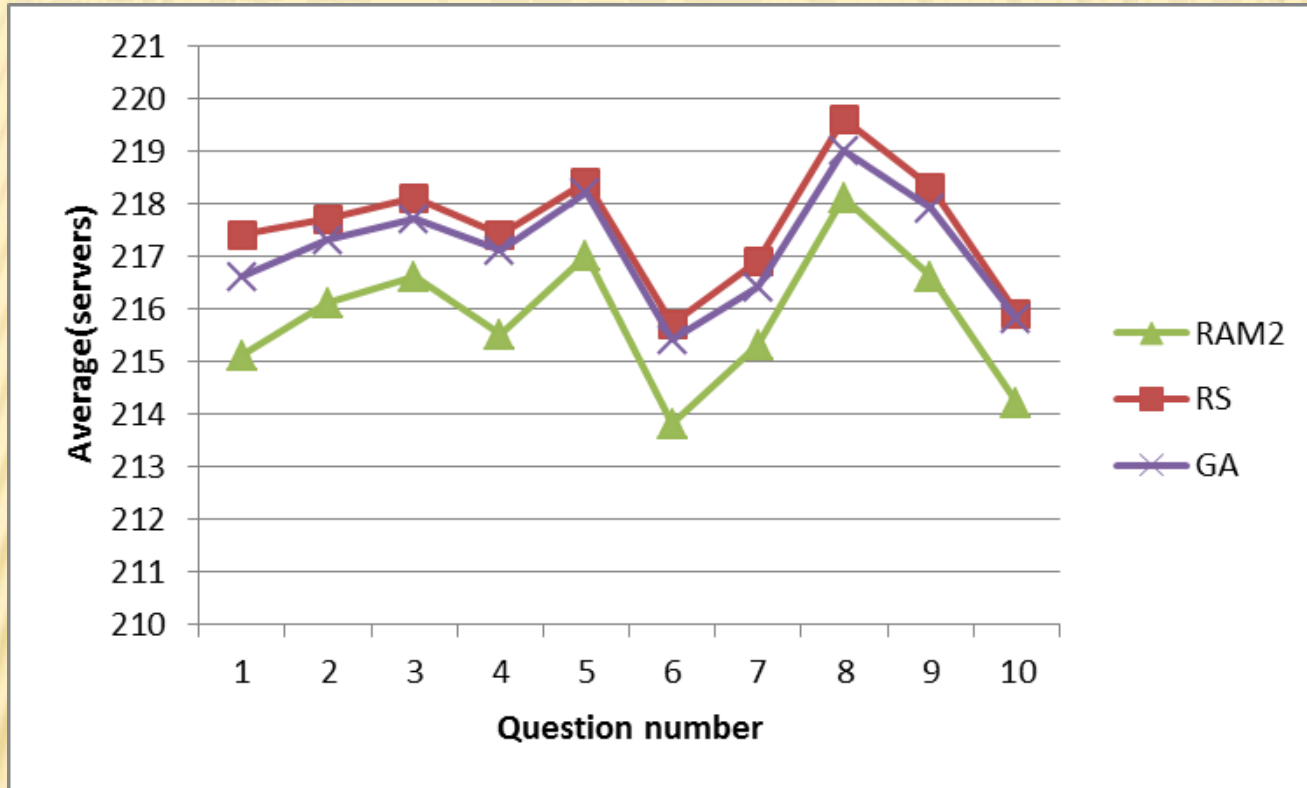
- ✖ RAM2、RS、GAでは出力される解に差はほとんどないが、平均値で見ればRAM2はほぼすべての問題で最も良い平均値を示した

比較検証結果（問題サイズ500）



- × 問題サイズ200に比べて平均値の差は見られるようになった
- × やはり平均値ではRAM2が最も良い結果となった

比較検証結果（問題サイズ1000）



- × 問題サイズ500に比べて平均値の差は見られるようになった
- × やはり平均値ではRAM2が最も良い結果となった

まとめ

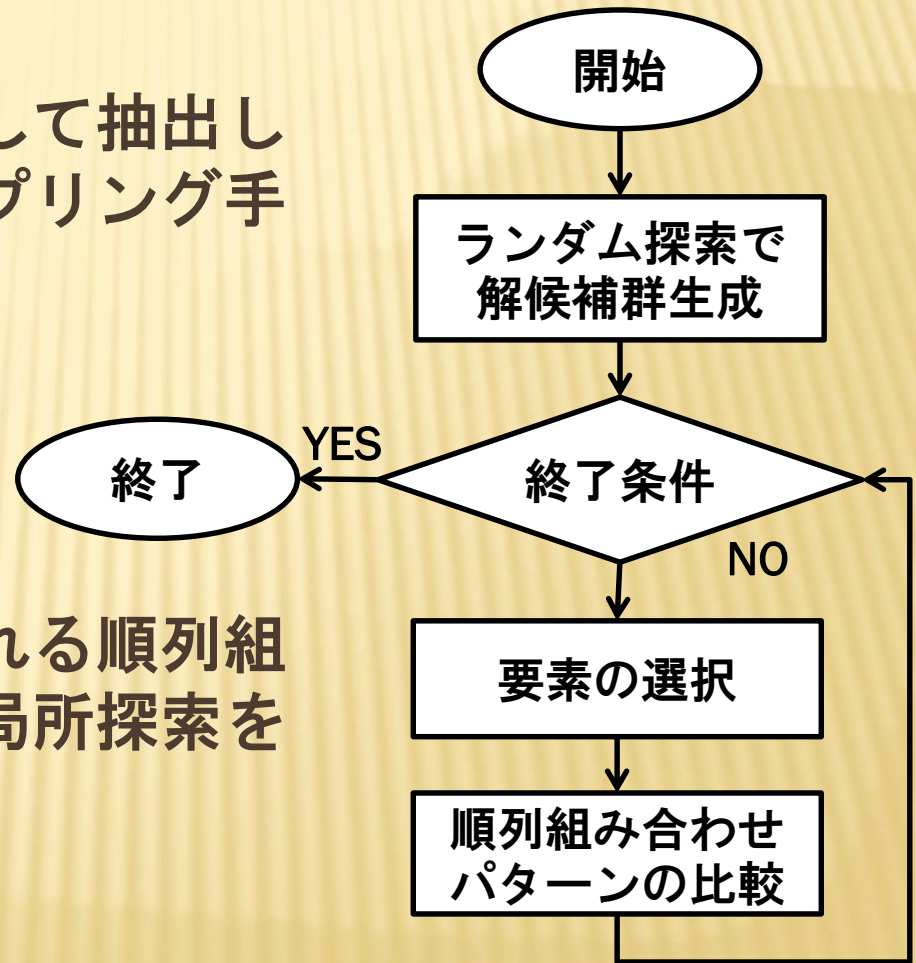
- × 統合対象サーバ数を200、500、1000の問題に4つの解法を適用して比較を行った結果
 - + 今回実装した4つの解法にははっきりとした差は見られなかった
 - + 改良型ランダムサンプリングが最も良い平均値を出力する可能性があることが分かった
- × 今後の課題
 - + 時系列を考慮したさらに次元数の多い問題や問題サイズの問題に適用する必要があると考えられる

✕ ご清聴ありがとうございました。

改良型ランダムサンプリング（RAM2）

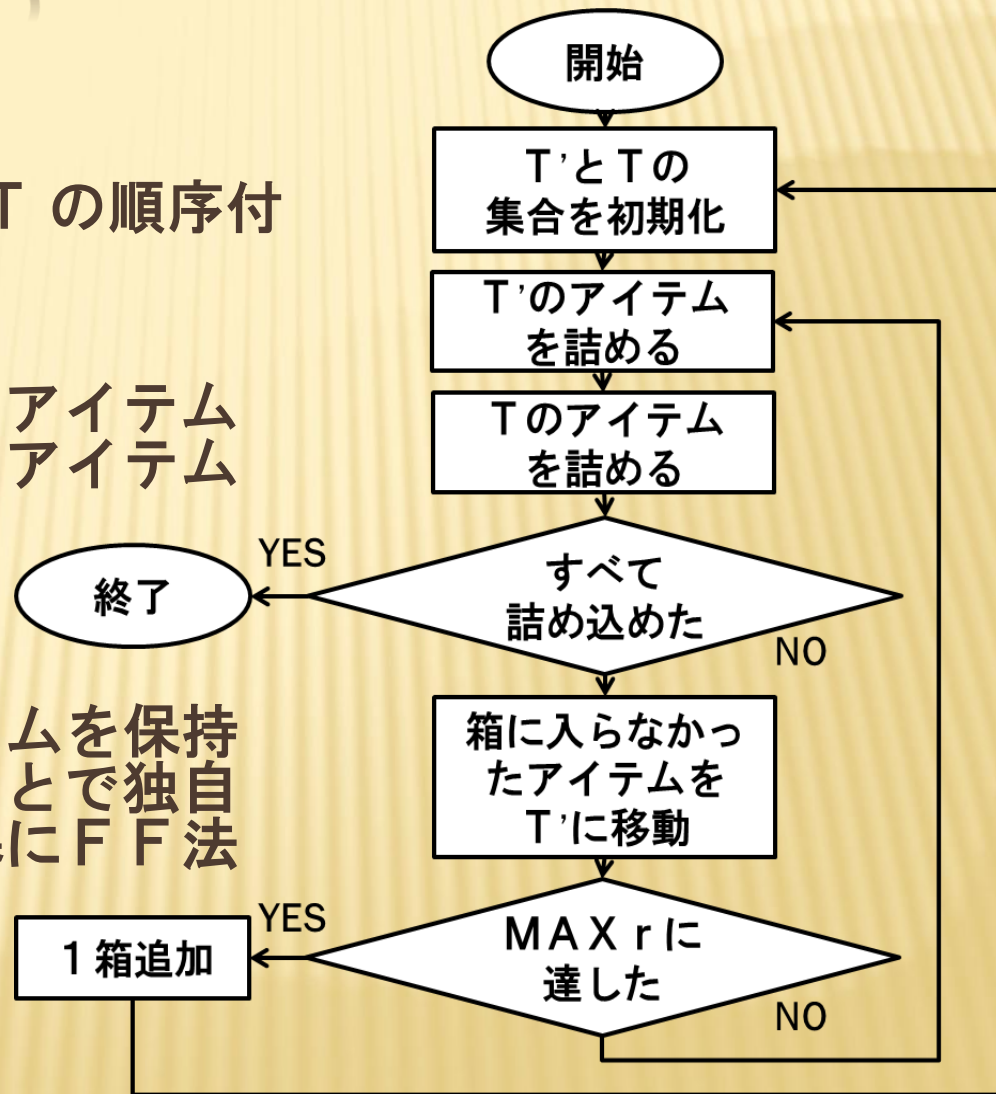
- × ランダム探索で解候補群として抽出した複数箇所にランダムサンプリング手法を適用

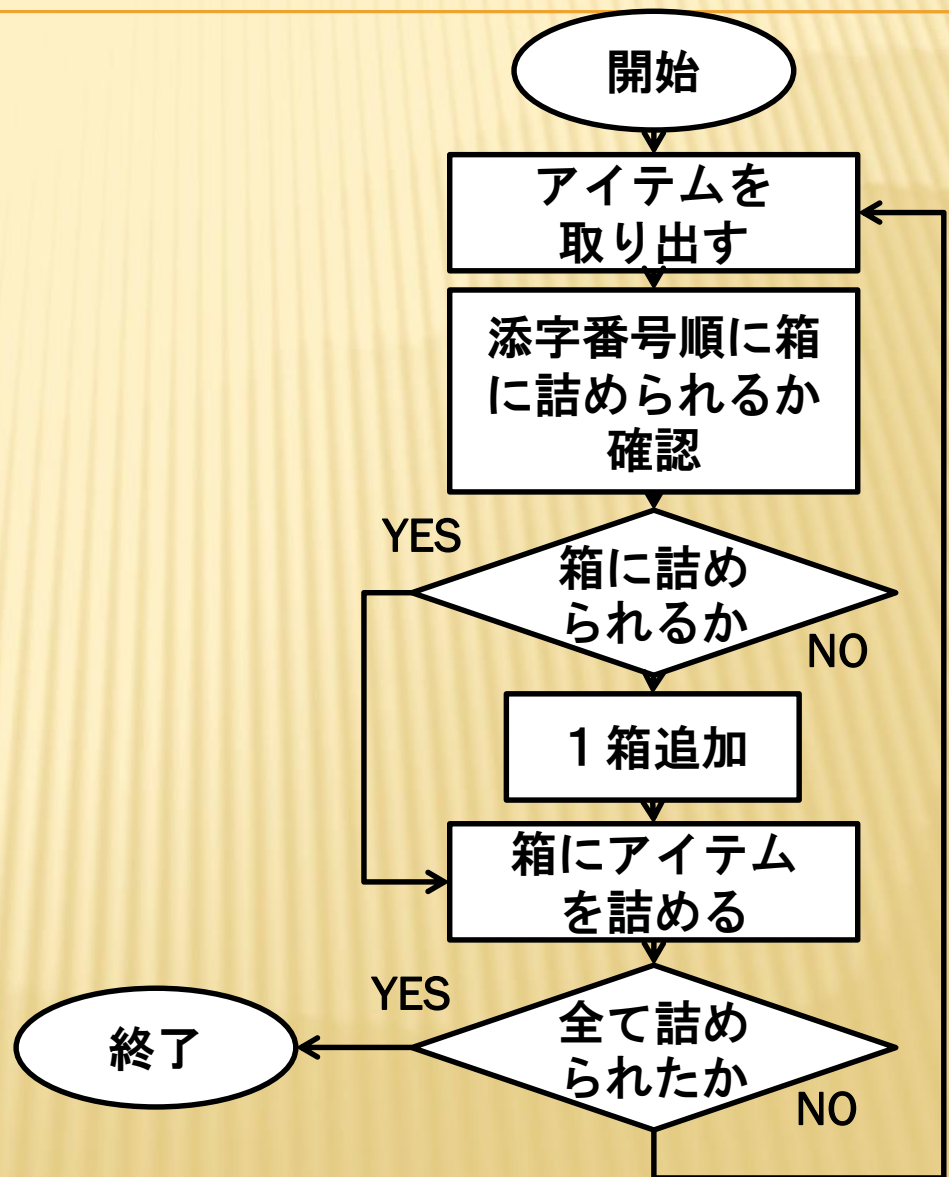
- × 選択された要素から生成される順列組み合わせパターンを用いて局所探索を行う

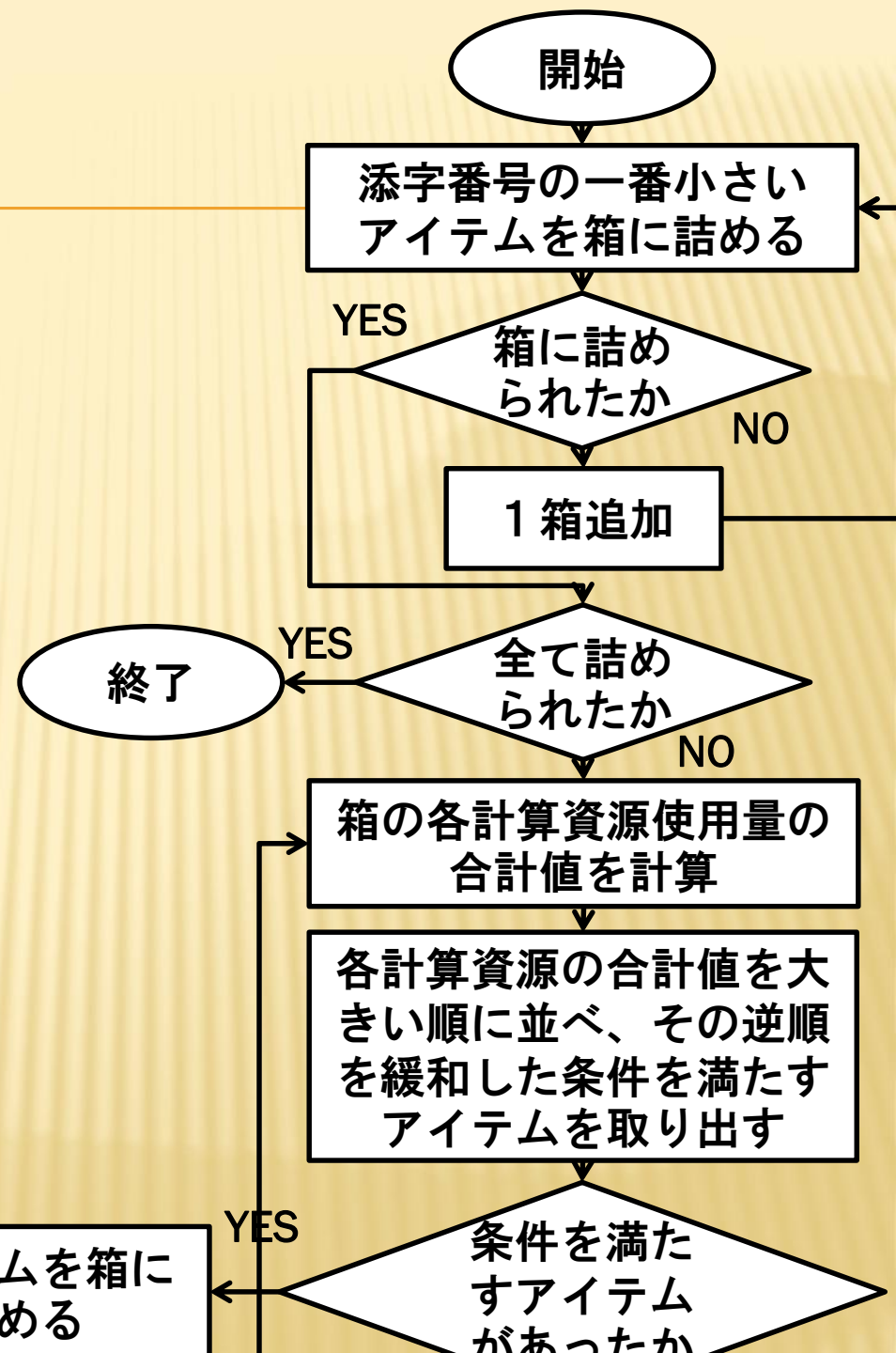


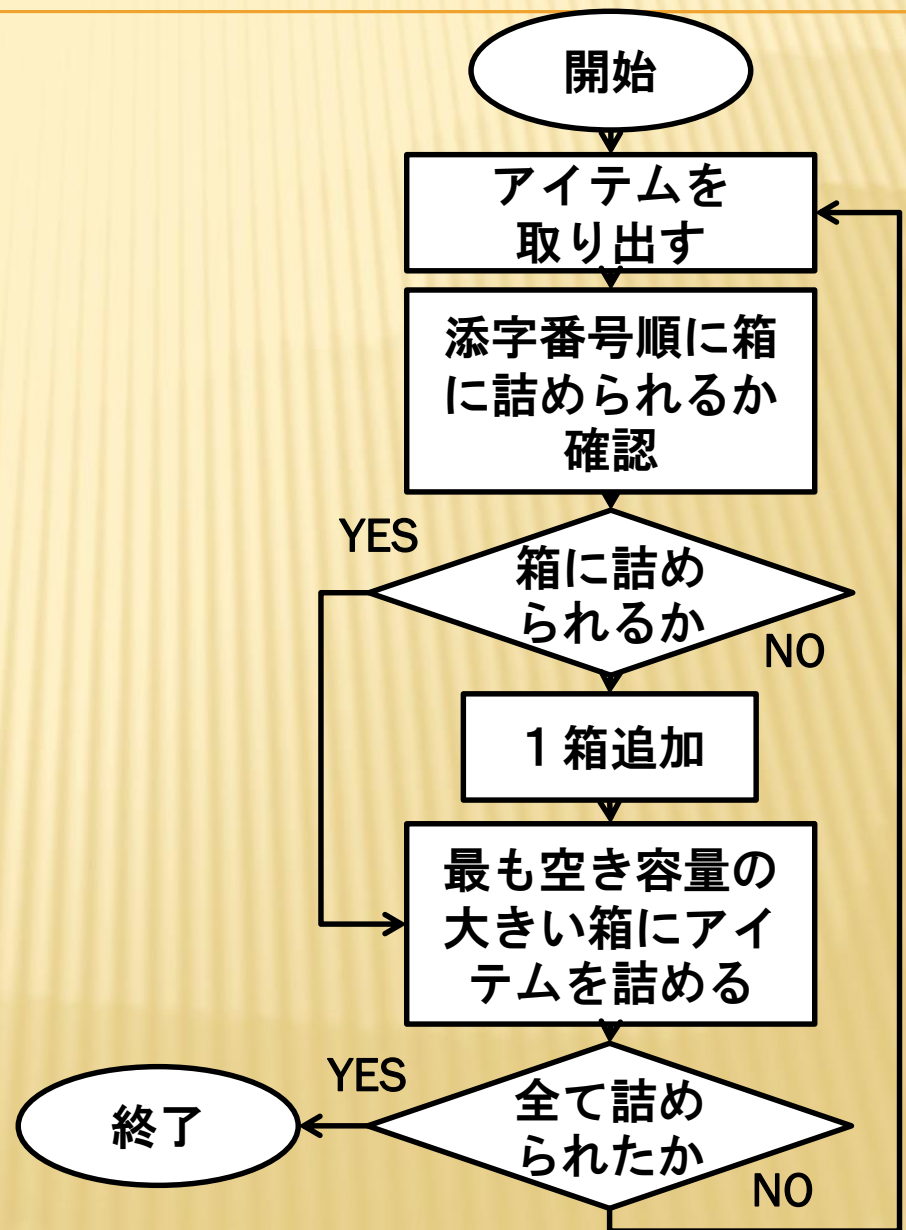
改良型FFD (FFDR)

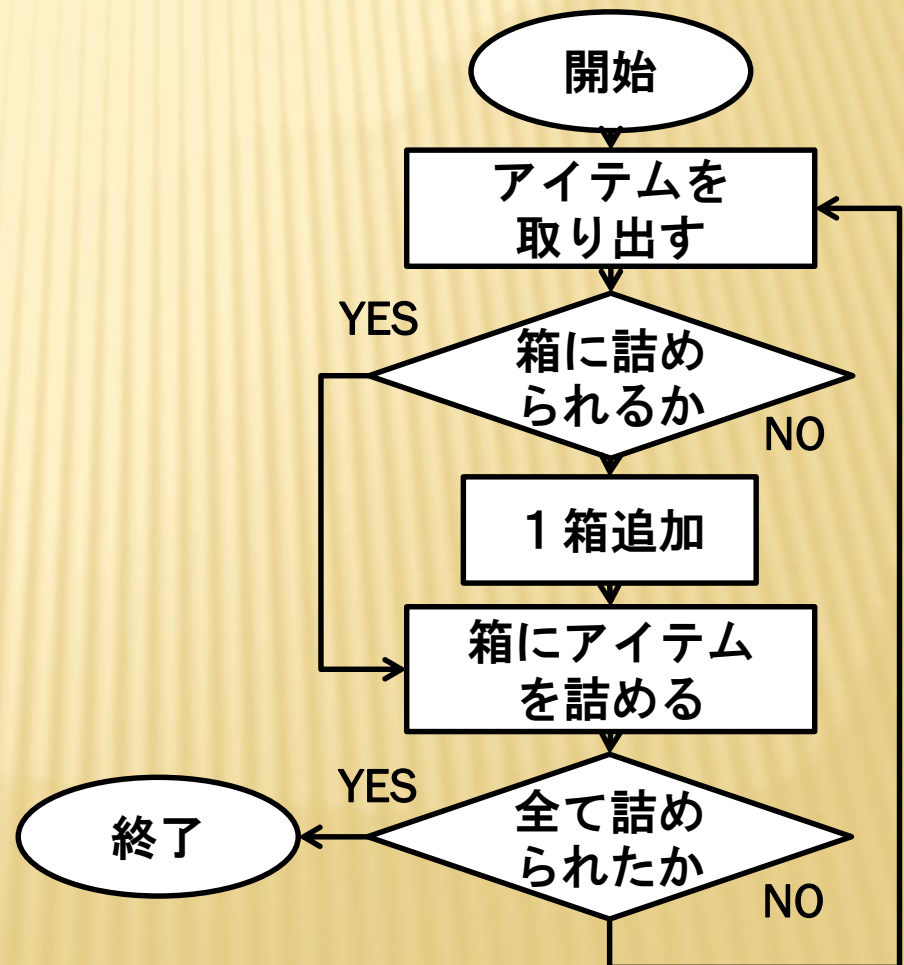
- ✕ アイテムを格納する T' と T の順序付き集合を2つ用意
- ✕ 初期状態では T にすべてのアイテムがあり、箱に入らなかったアイテムがあれば T から T' に移動
- ✕ 箱の増加要因となるアイテムを保持し、優先的に箱に詰めることで独自のソートを行い、それを基にFF法で詰める







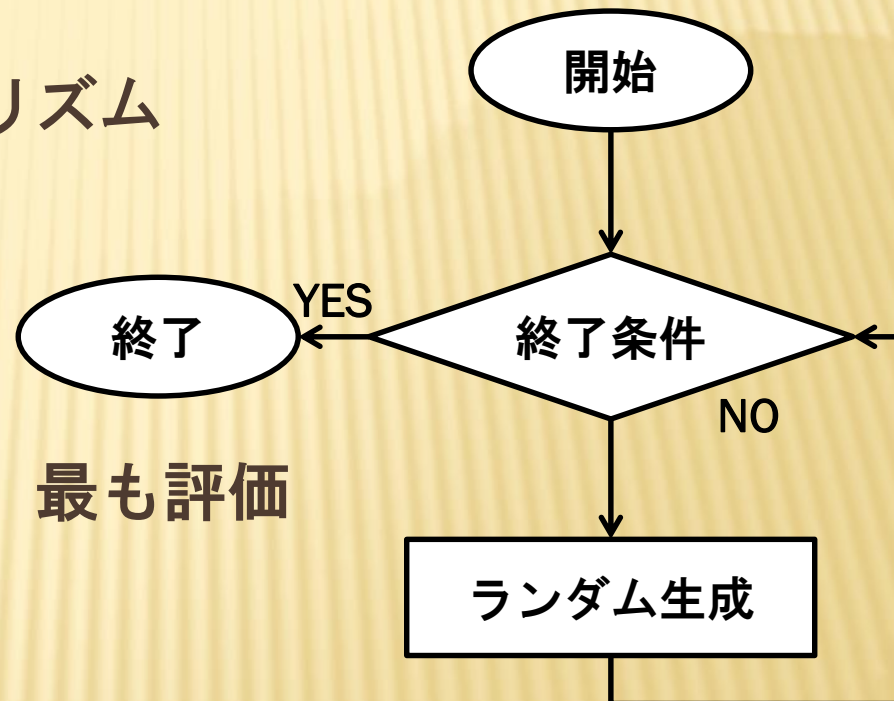




ランダム探索 (RS)

✕ 最も単純な大域探索アルゴリズム

✕ ランダムに解個体を生成し、最も評価の
良い個体を残す

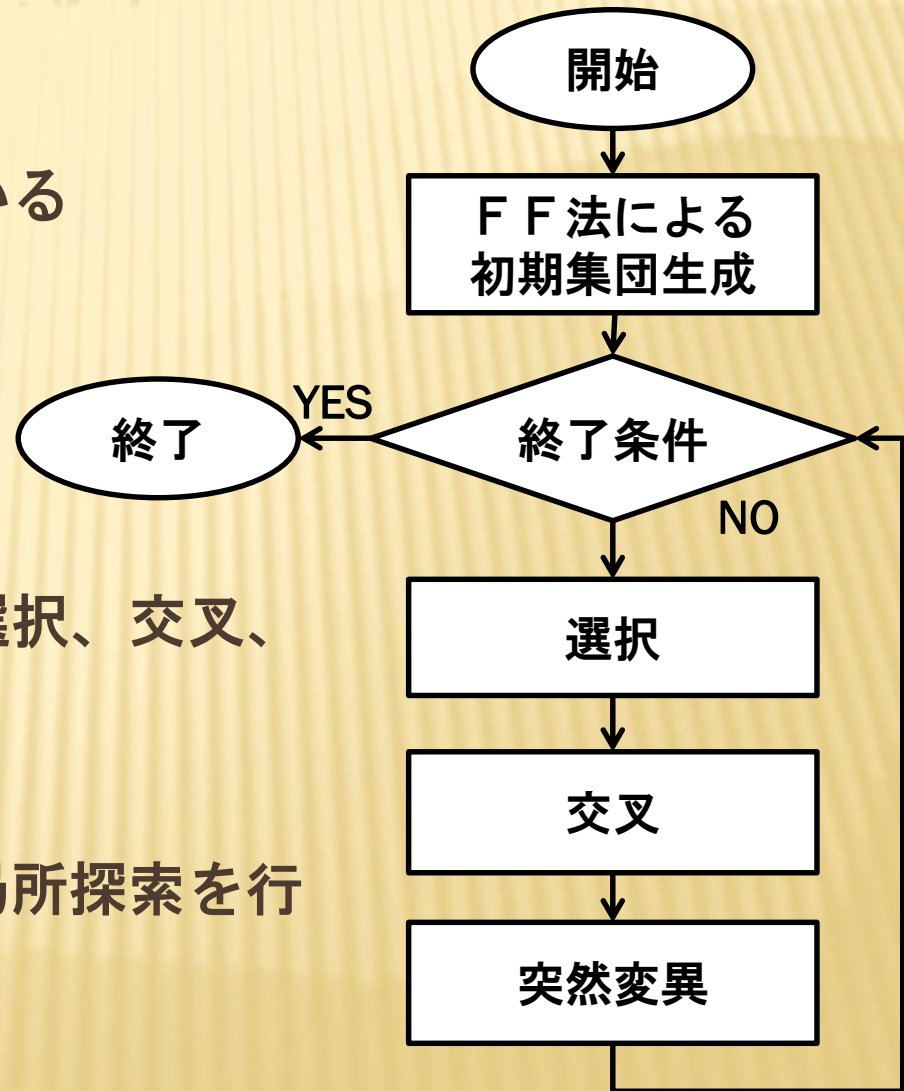


遺伝アルゴリズム (GA)

- × 初期集団の生成に F F 法を用いる

- × 生成された個体集団に対して選択、交叉、突然変異を行う

- × 交叉により一度に複数箇所の局所探索を行うことができる



各アルゴリズムの世代数

	問題サイズ500	問題サイズ1000
RAM2	約1000世代	約80世代
FFDR	1世代	—
RS	約6500世代	約1500世代分
GA	約60世代	約13世代