

仮想サーバ配置問題のための最適解近似アルゴリズム

1175085 播田 直紀 【 福本研究室 】

Optimal solution approximation algorithm for virtual server placement problem

1175085 HARITA, Naoki 【 Fukumoto Lab. 】

1 はじめに

仮想サーバ配置問題とは、統合先サーバに収容されるサーバの各計算資源使用量の合計値が閾値を超えないという制約の下で、目的関数である統合先サーバ台数が最小になるようなサーバの組合せを求める問題である。統合対象のサーバによる資源の必要量が統合先サーバの備える資源容量を超えると、重大な性能劣化を引き起こすことになる。

仮想サーバ配置問題を定式化した N 次元パッキング問題は NP 完全な問題に分類されており、文献 [2] で 2 次元ビンパッキング問題に対してアイテム数が 100 以下の場合には数十分で厳密解法を見つける手法が提案されているが、それ以上のアイテム数の問題となると現実的な時間内では厳密解を得られない。

そのため、アイテム数の多い問題では現実的な時間内に解を必要とする場合、短時間で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必要がある。

本稿では、より適したアルゴリズムを検討するため、仮想サーバ配置問題にヒューリスティックなアルゴリズムを適用し、改良したランダムサンプリング手法、及び他の手法との比較・検証結果を報告する。

2 サーバ統合における 5 次元パッキング問題の定式化

サーバ統合における統合対象サーバをアイテム、統合先サーバをビンと考えた場合、与えられたアイテムをビンに詰める際に使用するビンの数や詰めたアイテムの高さを最小化する組合せ最適化問題はビンパッキング問題と呼ばれ、計算資源が複数ある場合、計算資源の種類数が N となり、 N 次元パッキング問題として定式化されている [1]。

本稿では、仮想サーバの計算資源を CPU 性能、メモリ使用量、ディスク性能、ディスク容量、ネットワーク性能の 5 つとし、5 次元パッキング問題として定式化される。これらの計算資源の値はすべて統合先サーバの最大使用量を 100% とした場合の割合で与えられる。

統合対象のサーバ数を n 、統合対象のサーバを $s_i (1 \leq i \leq n)$ とする。 s_i には計算資源としてそれぞれ

CPU 性能使用量 pc_i 、メモリ使用量 pm_i 、ディスク性能使用量 $pd1_i$ 、ディスク容量使用量 $pd2_i$ 、ネットワーク性能使用量 pn_i が与えられる。また、 m 台の統合先サーバ $s'_j (1 \leq j \leq m)$ に統合される統合対象サーバの集合を X_j とする。統合先サーバはすべて同じ容量を持つものであることを前提とし、与えられる閾値はそれぞれ CPU 性能使用量 Rc 、メモリ使用量 Rm 、ディスク性能使用量 $Rd1$ 、ディスク容量使用量 $Rd2$ 、ネットワーク性能使用量 Rn とする。

このとき、サーバ統合の最適化問題は以下のように定式化される。

目的関数: $m \rightarrow$ 最小化

$$\begin{aligned} \text{制約条件: } & \sum_{s_i \in X_j} pc_i \leq Rc, \\ & \sum_{s_i \in X_j} pm_i \leq Rm, \\ & \sum_{s_i \in X_j} pd1_i \leq Rd1, \\ & \sum_{s_i \in X_j} pd2_i \leq Rd2, \\ & \sum_{s_i \in X_j} pn_i \leq Rn (i = 1, \dots, n, j = 1, \dots, m) \end{aligned}$$

3 コーディング方法

統合対象サーバの台数分の長さを持つリスト $[s_1, s_2, \dots, s_i, \dots, s_n] (1 \leq s_i \leq n)$ を順列で表現し、 s_1 から順に統合先サーバに FF 法を用いて詰めていく。

3.1 FF 法 (First-Fit algorithm)

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、詰めるのが可能な箱に詰める動作を繰り返し、詰められなければ箱を増やしそれに詰める。

4 改良型ランダムサンプリング

4.1 ランダムサンプリング

ランダムサンプリングはヒューリスティックなアルゴリズムの 1 つである。まず、ランダムに作成した解個体を 1 つ生成する。次に解個体からランダムにユーザが指定した数だけ要素を選択する。その選択された部分要素からなるすべての順列組み合わせパターンを入れ替えた解個体を評価し、最も評価の高い解個体を残し、この過程を繰り返すことで解の改善を行う。

4.2 改良型ランダムサンプリング (RSM2)

ランダムサンプリングは 1 つの解から局所探索のみを行うため、問題サイズの大きい問題では選択する要素

を増やすか時間をかけないとあまり良い結果を示さない。そこで、大域探索後に局所探索を行えるように改良を行った。まず、初回のみ任意の回数だけランダム探索を行い、その時点で得られた同じ最良解を持つ解個体集団を用意し、解個体それぞれにランダムサンプリングを適用する。最良解が更新されればまたその時点で得られた同じ最良解を持つ解個体集団を用意し、それぞれにランダムサンプリングを適用する。これを繰り返すことで解の改善を行う。

5 比較結果

統合対象サーバ 500 台の問題に対して乱数の種を 100 種類用い、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) の 4 つのアルゴリズムでそれぞれの試行平均と試行分散の推移を比較した。RAM2 のランダム探索回数は 1000、選択する要素数は 3、FFDR の MAXr は統合先サーバ台数と同数、GA の集団サイズは 100、交叉方法は PMX、交叉確率 90%、突然変異確率 10% として 300 秒間の試行を繰り返した。このとき、統合先サーバの閾値はすべて 80% とする。

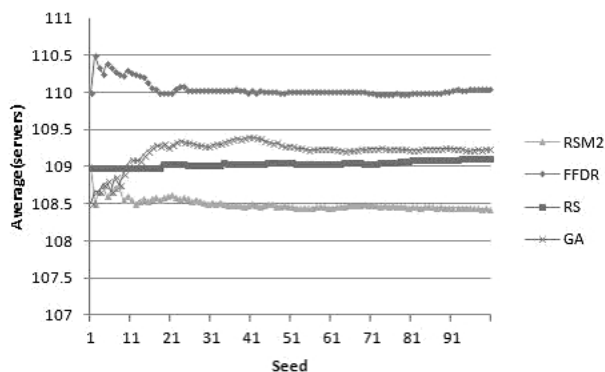


図 1 問題サイズ 500-試行平均の推移

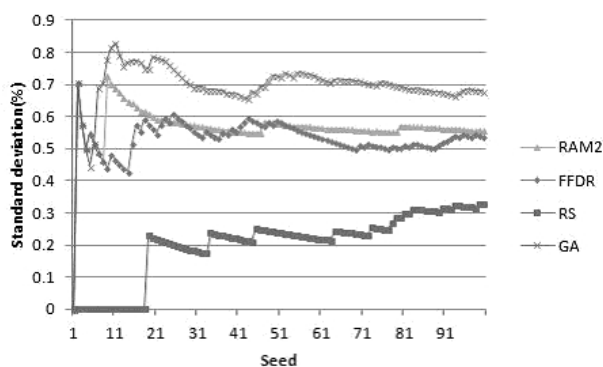


図 2 問題サイズ 500-試行標準偏差の推移

標準偏差を σ とした場合、図 1、図 2 から試行平均の最も良い RAM2 でも $\pm 2\sigma$ 以内では他のアルゴリズムの $\pm 2\sigma$ 以内と範囲が被ってしまうことが分かった。データが正規分布に従う場合、平均値 $\pm 2\sigma$ 以内に約 95.4% の出力される解が存在することになるため、この問題サイズでははっきりとした差は見られなかった。

この問題サイズでは差が確認できなかったため、さら

に統合対象サーバ 1000 台の問題に乱数の種を 10 種類用いて 300 秒間の試行を繰り返した。その試行平均と試行標準偏差の推移を図 3、図 4 に示す。

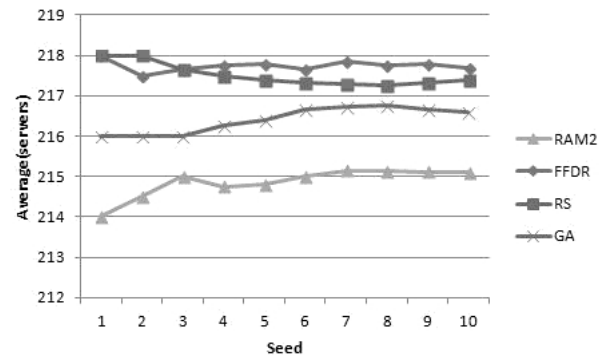


図 3 問題サイズ 1000-試行平均の推移

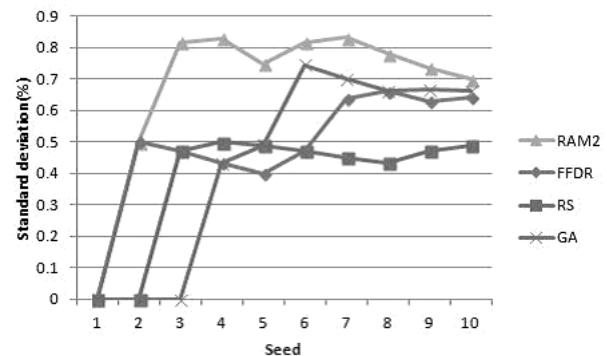


図 4 問題サイズ 1000-試行標準偏差の推移

図 3 を見ると試行平均では、問題サイズ 500 のときと同じく RSM2 が最も良い結果を示した。しかし、試行平均では問題サイズ 500 のときよりも差は生まれたが、この問題サイズでも図 4 を見る限り $\pm 2\sigma$ 以内では他のアルゴリズムと範囲が被っており、出力される解の範囲にはっきりとした差はなかった。

6 まとめ

計算資源の数を 5 とする仮想サーバ配置問題では問題サイズが 500 や 1000 の問題において試行平均の観点から見れば改良型ランダムサンプリングが最も有効であると考えられる。しかし、他のアルゴリズムと $\pm 2\sigma$ 以内の範囲で出力される解が被っているため、完全に良いとは言えない結果となった。今後の課題として時系列を考慮したさらに次元数の多い問題やさらに問題サイズの大きい問題に適用する必要があると考えられる。

参考文献

- [1] 網代育大, 田中淳裕, NEC システムプラットホーム研究所, サーバ統合のための組み合わせ最適化アルゴリズムの提案と評価, 2007.
- [2] J.Carlier,F.Clautiaux,and A.Moukrim. New reduction procedures and lower bounds for the two dimensional bin packing problem with fixed orientation. Computers and operations reserch, Vol.34,No.8,pp.2223-2250, 2007.