3A1-01

TPSAにおける重要温度領域探索と並列化効率

Important temperature domain search and parallel efficiency in TPSA

三木 光範[*1] 廣安 知之[*1] 吉田 武史[*2] ○伏見 俊彦[*2]
Mitsunori Miki Tomoyuki Hiryasu Takeshi Yoshida Toshihiko Fushimi

[*1]同志社大学工学部 Dept. of Knowledge Engineering , Doshisha University
[*2]同志社大学大学院 Graduate Student , Doshisha University

Simulated Annealing is an effect approximate method for solving combinational optimazation problems. But , this method requires huge amount of calculation. Therefore , researches on Parallel Simulated Annealing (PSA) have been performed. Especialy Temperature Parallel Simulated Annealing (TPSA) shows good performance. So I propose Adaptive TPSA (ATPSA) , which automatically control temperature range. The experimental results of ATPSA shows better performance than TPSA. And I search parallel efficiency of ATPSA.

1. はじめに

シミュレーテッドアニーリング (Simulated Annealing : SA) は「焼きなまし」と呼ばれる物理現象をモデル化した汎用的最適化アルゴリズムである．近年では多くの分野に利用されているが，膨大な計算時間が必要である事や温度パラメータの推移が解精度が影響するなどの問題点は依然として存在する．

計算時間短縮を目的として，SA を並列化する研究が多く考えられており，なかでも温度並列 SA(Temperature Parallel SA : TPSA)[1] は温度スケジュールが原理的に不要であるという優れた特徴をもつ．

一方，近年の研究において従来の SA で行う温度冷却ではなく，特定の温度領域のみの解探索で良好な精度を示す事が報告されている [2]．しかし，この重要な温度領域は対象問題に依存しており，明確な決定方法は明らかになっていない．

そこで本研究では，TPSA に重要温度領域を自律的に探索するメカニズムを組んだ適応的 TPSA(Adaptive TPSA : ATPSA) を提案し，解探索性能と並列化効率について検証する．

2. 温度並列 SA

温度並列 SA(TPSA)[1] は並列処理との高い親和性を持ち，温度スケジュールが原理的に不要である．TPSA で は，複数のプロセッサに異なる温度を与え，各プロセッサは一定の温度でアニーリングを行い，一定の間隔で隣接する温度のプロセッサ間で解交換を行なう． 図 1 に逐次 SA と TPSA の温度スケジュールを示す．

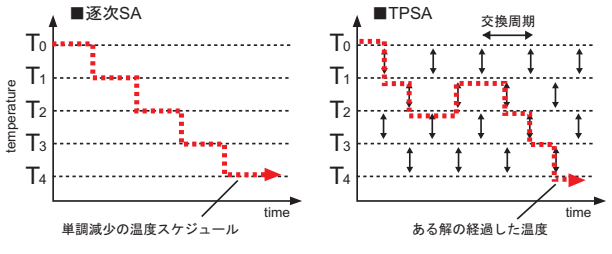

図 1: 逐次 SA と TPSA の温度スケジュール

連絡先: 三木 光範，同志社大学工学部，京都府京田辺市多々羅都谷 1-3，0774-65-6930，mmiki@mikilab.doshisha.ac.jp

しかし，TPSA においても解探索開始時の温度範囲をチューニングする必要があり，また高温プロセスのアニーリングと解品質の関係などの検証が必要な点が残る．

3. 重要温度領域の探索

一般的な SA では，十分に高い温度から十分に低い温度まで緩慢に冷却する事で良好な結果を示すと考えられている．この温度推移は温度スケジュールとよばれ，これを決定するための膨大な実験が必要である．一方，特定の固定温度 (以後，重要温度領域とする) のみの解探索で良好な結果を示す報告もあるが [2]，この温度を決定するためにも多くの実験が必要である．

TPSA では，各プロセスが最高温度と最低温度の間で固有の温度を持ち，解探索途中で解交換を行うことで，自動的に温度スケジュールが生成される．しかし各プロセスの解推移を検証した結果，重要温度領域以外の温度プロセスでは効率的な解探索が行われていない事が分かった．そこで本研究では，自律的に各プロセスの温度を重要温度領域内に設定する手法を考案する．

4. 適応的 TPSA

適応的 TPSA(ATPSA) は，TPSA の解探索途中で自律的に重要温度領域を探索するアルゴリズムである．ATPSA では解の値とは別に評価値という値を用い，この値をもとに重要温度領域を探索する．図 2 に ATPSA のアルゴリズムを示す．

図 2 に示す解生成，受理判定，状態遷移，解交換については TPSA と同じ操作を行う．ここでは評価値計算と，その値を元にした温度範囲設定，温度割り当てについて示す．

- **評価値計算**
  ATPSA では全プロセスに同一の基準値を設定する．各プロセスはその基準値を下回る解遷移が生じた場合，解と基準値の差を加算する．その結果，解交換周期ごとに各プロセスが異なる評価値をもつ．
- **温度範囲設定**
  解交換周期ごとに全プロセスが同期をとり，評価値の比較を行う．ATPSA の最高温度と最低温度の間に重要温度領域が含まれる場合，重要温度領域の評価値が高く，他の温度の評価値は低い，または加算されない結果となる．そこで評価値が加算されている温度範囲のみを次の解交換周期の解探索温度範囲と設定する．

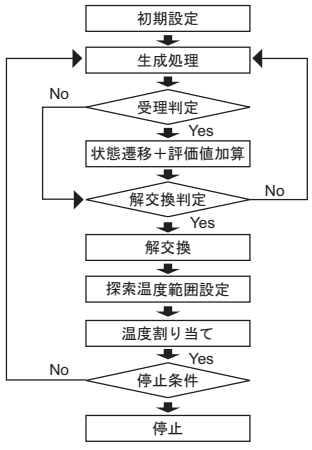

図 2: 適応的 TPSA のアルゴリズム

- **温度割り当て**
  前操作で決定した温度範囲内で，各プロセスの温度を決定する．ここでは等比間隔で各プロセスに温度を割り当てた．

基準値は，解交換周期ごとの全プロセスの解平均を用いた．

5. 数値実験

ATPSA を用いて数値実験を行い性能を検証する．対象問題として 4 つの巡回セールスマン問題 (Traveling Salesman Problem : TSP)[3] を用い，ATPSA，逐次 SA と TPSA を同一パラメータで比較した．実験結果を図 3 に示す．図 3 の横軸は TSP の問題名，縦軸は最適解との誤差を示す下界からの距離 ((解-最適解) / 最適解) を示す．なお 20 試行の平均値である．

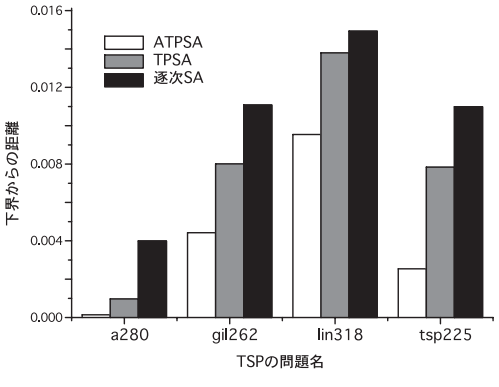

図 3: 実験結果

上記の結果よりわかるように，ATPSA の解探索能力が他の手法に比べ良好である．これは温度探索メカニズムにより，各プロセスの温度が重要温度領域に収束しているためと考えられる．

6. 並列化効率

ATPSA の並列化効率についての実験を行い，検証する．対象問題は TSP を用い，ATPSA と TPSA を同一パラメータで比較した．実験に用いたコンピュータは，Pentium 800MHz, 32CPU のクラスタシステムを用い，TSP の問題ごとで 1，2，4，8，16，32 のプロセッサ数での実験を行なった．実験結果を図 4 に示す．図 4 の横軸はプロセッサ数，縦軸は並列化効率 (1-cpu time/n-cpu time) を示す．なお 10 試行の平均値である．

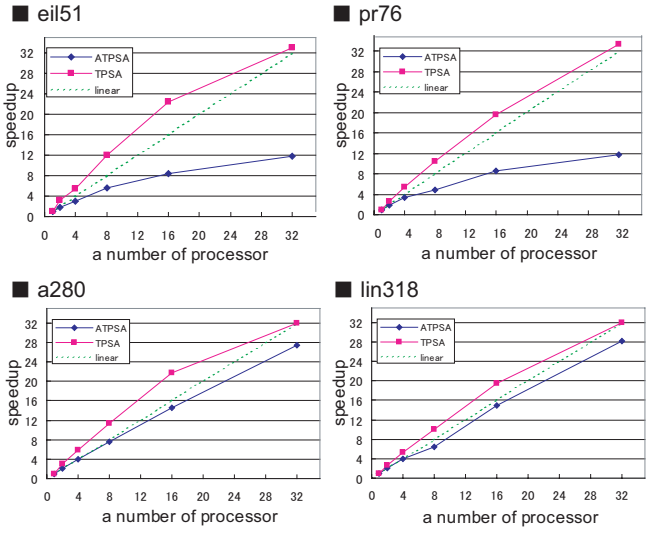

図 4: 並列化効率

図 4 の結果からわかることは，eil51 や pr76 などの比較的簡単な問題においては ATPSA の並列化効率が 32CPU を用いたときで 12 倍程度であった．それに対して TPSA においては 33 倍という高い並列化効率という結果になった．この理由としては，ATPSA はアルゴリズムにおいて，評価値計算，探索温度範囲設定と温度割り当てを行なうため，TPSA より通信量が多くなるためだと考えられる．また，ATPSA は 1 つのプロセスに評価値を集めて探索温度範囲を決定するため，同期を取る必要があることも原因の 1 つと考えられる．

一方，a280 や lin318 などの都市数の多い複雑な問題になると ATPSA の並列化効率も改善されている．これは問題が複雑になるにつれて解探索に要する計算時間が大きくなり，全計算時間に対して通信時間が占める割合が小さくなるためであると考えられる．

7. まとめ

本研究において重要温度領域を自律的に探索する TPSA(ATPSA) を提案した．TSP を用いた数値実験により，従来の TPSA に比べ解探索能力が向上した事を確認した．また，今回提案した ATPSA の並列化効率を検証した．実験より，簡単な問題においては並列化効率は認められなかったが，複雑な問題においては十分に並列化効果があることを確認した．今後は他の問題に適用し，有効性を検証する．

参考文献

[1] 小西健三，瀧和男．温度並列シミュレーテッド・アニーリング法の評価．情報処理学会論文誌，1995

[2] MARK FIELDING. SIMULATED ANNEALING WITH AN OPTIMAL FIXED TEMPERATURE. Society for Industrial and Applied Mathematics,2000

[3] TSPLIB,http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/