

巡回セールスマン問題を対象とした 並列タブーサーチにおけるプロセス間通信の効果について

大 植 裕 之[†] 大 西 克 実[†]
中 野 秀 男[†] 榎 原 博 之^{††}

巡回セールスマン問題は組み合わせ最適化問題の中でも特に代表的な問題例の一つで、これまで数多くの研究がなされている。本研究では、この問題を対象にメタヒューリスティクス手法の一つとして知られるタブーサーチ法を並列化する。特に、並列化する際にプロセス間で枝交換に関する情報を共有するために必要となる通信に注目し、より効率よく情報共有を行うことができる手法を検討する。計算機実験としては並列タブーサーチのプログラムを複数用意し、得られた結果から評価を行う。

About the interprocess communication with parallel tabu search algorithms for the traveling salesperson problem

HIROYUKI OHUE,[†] KATSUMI ONISI,[†] HIDEO NAKANO[†]
and HIROYUKI EBARA^{††}

The traveling salesperson problem is one of the typical examples in the combinational optimization problems and many researches have been conducted so far. In this paper, the tabu search method which is one of the meta-heuristics method for this problem is parallelized. Especially we focus on the communication needed to share the information regarding to "the edge exchange" between processes, and examine a method which enable to share the informations more efficiently. The computing examination has been done by several evaluations from the result of running the parallel tabu search programs.

1. はじめに

巡回セールスマン問題 (Traveling Salesperson Problem)¹⁾ はあるセールスマンが幾つかの都市を一度ずつ訪問して出発点に戻ってくるときに、移動距離が最短になる経路を求める問題である。この問題は組合せ最適化問題²⁾ の代表的な問題として知られ NP 困難として知られている。

本研究では、この問題に対してメタヒューリスティクス手法の一つであるタブーサーチ法³⁾ を用いて近似解を求める。メタヒューリスティクスは近傍探索によって得られた局所最適解をさらに改善するための枠組であり、他にアニーリング法や遺伝的アルゴリズムなどがある。また、これらの手法は並列化することにより、より精度の高い解を求めることができる⁴⁾⁵⁾。

そこで本研究では、タブーサーチを並列化することにより、より短い時間で高い精度を持った解を求めることを目的とする。特に、並列処理において解の交換情報を各プロセス間で共有するための通信部分に注目し、より効率よく情報の共有を実現する手法を検討する。そして、計算機実験により並列と逐次での場合における解の比較を行い評価する。

以下、2章で対象とする問題とタブーサーチについて述べる。3章では実装した並列タブーサーチと通信部分の改良について述べ、4章で実験結果を示し考察を行う。

2. タブーサーチ法

2.1 巡回セールスマン問題

巡回セールスマン問題は組合せ最適化問題の中でも特に代表的な問題の一つで、巡回する都市数が N のときの巡回経路は $(N-1)!/2$ 通りになり、 N の増加により巡回経路数が指数的に増加する。

したがって、大きい N に関して、全ての組合せの中から最短巡回経路を見つけることは非常に困難な問

[†] 大阪市立大学大学院 創造都市研究科
Graduate School for Creative Cities, Osaka City University

^{††} 関西大学 工学部

Faculty of Engineering, Kansai University

題である。

2.2 タブーサーチ

タブーサーチ法は近似解法と様々な戦略の組合せによってできるメタヒューリスティクス手法の中の一つである。タブーサーチでは、解の遷移過程において局所最適解に至り、それ以上探索する候補が無くなっても、引き続き未知の解への遷移ができるように、様々な工夫がなされている。

その主な特徴は、タブーリストと呼ばれるリストに解を遷移する時の情報を保存しておき、遷移元への解の遷移を禁止することが挙げられる。これにより局所最適解に至っても、脱出することが可能となり未知の解への探索を継続する事が可能となる。

2.2.1 アルゴリズム

単純なタブー探索を実現するアルゴリズムを以下に簡単に記す。

- (1) 初期解を生成する。
- (2) 近傍探索を繰り返し遷移可能な解を求める。
- (3) 解の遷移と同時に、遷移の記録をタブーリストに保存する。
- (4) 局所最適化に至っても、一時的な改悪を許しタブーでない解へ遷移することで探索を続ける。
- (5) 指定の終了条件を満たせば、探索を終了する。

2.2.2 タブーリスト

タブーリストの実装としては、“短期タブーリスト”と“長期タブーリスト”を使用する。これにより、単にタブーとされる解を保存するだけでなく、解の探索過程で役立つ情報の収集も可能となるようにする。

短期タブーリストは解のサイクリングを避けるために設けられたリストである。サイクリングとは遷移過程で改善と改悪が繰り返し行われることであり、未知の解への遷移ができなくなることである。これを防ぐために最近探索した解をリストに登録し、タブーとなった解を選択せずに別の解への遷移を行うことで元の解への移動を妨げる。これにより、局所最適解からの脱出が実現でき、短い周期のサイクリングを防ぐことができる。

長期タブーリストでは、タブーとなった回数を保存したり、同時に交換回数の多さや枝の短さに分けて解析を行い、その情報を蓄積する。このように長期タブーリストでは探索に伴う解の特徴を記憶しておく。これを用いると、未探索の領域への探索を組み入れるなど、様々な工夫が可能となる⁶⁾。

3. 並列タブーサーチ

本研究では、タブーサーチ法を用いた探索アルゴリ

ズムを並列化し、その有効性を評価する。以下、本研究で実装した並列タブーサーチについて詳しく述べる。

3.1 プログラムの実装

並列に探索する方法として、各プロセスがランダムな初期解を生成しそれぞれタブーサーチを行う方法を用いる。各プロセスは解の遷移を行いながら自身の持つ短期タブーリストにタブーとされる枝の交換情報を登録していく。

また、解の遷移後、その解が改善に働いているときのみ、枝交換の情報を他のプロセスに送信し、長期リストの更新に利用する。また、受信した枝交換の情報が、自身の探索に適用可能かつ改善に働いた場合、即時に適用する。

近傍探索: 2-opt 法¹⁾を用いる。2-opt 法は巡回路から2本の枝を切り離した後、再び巡回路になるように枝をつなぎ変える近似算法である。この手法は簡易な実装ですみ探索にあまり負荷がかからないので採用した。

短期タブーリスト: タブーとなった解を保存しておくもので、枝交換の情報を保存しておく。短期リストは繰り返し操作のターン毎に保持している枝交換の情報を削除する。

長期タブーリスト: タブーリスト共有の対象であり、解を改善方向へ遷移させた枝交換の情報は、他プロセスにも送信する。受信側では送られてきた枝の構成や枝の交換情報を長期リストに反映する。長期リストは枝がタブーとなった回数を保存し、更に交換回数や枝の長さにかけてランキング情報を生成し保存する。長期リストは繰り返し操作の2ターン目以降から初期解の生成や、探索開始時のタブーの設定に用いられる。

ターンの設定: 初期解の生成から始まる一回の探索を1ターンとして扱い、ターン終了条件を満たせば別の初期解を新たに生成し再度探索を続ける。各ターンの設定を以下に述べる。

- 1ターンの終了条件はターンの中で最小の解が発見されてからパラメータで指定した回数の近傍探索を行い、最小の解が更新されなかった場合か、探索が改悪の方向に進み、そのターン内の最小解より指定の割合を越える解に到達した場合に終了する。
- 1ターン目は初期解の生成を最近近傍法で生成する。このとき、並列する各プロセスでは、1つ目の都市の選択をランダムで行う。ランダムに選択することで、各プロセスで異なる初期解を生成し、多様な枝の構成と交換の情

報が取得できることを期待する。

- 2 ターン目以降は長期リストの内容を反映して初期解の生成を行う。具体的には、長期リストの枝交換の多い順、もしくは枝の長さ順に初期巡回路を構成する。更に、長期リストのランキング上位の枝を短期リストに登録し、初期のタブーとする。

3.2 通信手法の提案

3.1 で述べたように従来⁷⁾ は解の遷移後、その解が改善に働いた時に他のプロセスへ枝の交換情報を送受信していた。これにより受信した枝の交換情報を長期リストに反映させることと、受信した枝交換の情報を自身の探索に適用することを実現する。しかし、この実装方法では受信回数に対する枝交換の適用回数が極端に少なく通信部分の効率が悪いことが考えられる。

表 1 は各問題に対する受信回数と受信した枝交換の情報が、即時に自身の探索に適用可能でかつ改善に働いた回数を示したものである。表から受信回数に対する適用回数の割合は 1%前後とほとんど適用されていないことがわかる。これは各プロセスの探索する解空間がそれぞれ違うため、受信した枝交換の情報を即時に適用することができないためであると考えられる。

表 1 枝交換の受信と適用回数

問題	受信回数	適用回数
lin318	57290.5	398.1
pr439	25873.1	215.9
rat575	17810.3	135.9

このように受信回数が多い事が原因となり探索に余分な影響を与え、効率よく探索ができないと考えられ、より効率の良い実装方法を検討する。そこで、まず送受信された枝交換情報がどれだけの効果を解の探索にもたらすのかを調べるために以下の実験を行うこととする。図 1 は並列タブー探索の流れ図を示している。図より、解の遷移後、改善に働いた枝交換情報の送受信は点線内の部分で実行される。本研究ではこの通信部分(点線内)を取り除いて比較実験を行った。これにより、各プロセスがタブーリストを共有せず自身の情報だけから長期リストを構成し探索を行うことになる。

4. 実験と考察

4.1 計算機実験の方法

実験方法: 並列タブーサーチ, 逐次タブーサーチ, そしてタブーサーチを用いずに局所探索である 2-opt 法のみで探索を行うの 3 つの場合について比較実験を行う。また、並列タブーサーチでは、枝

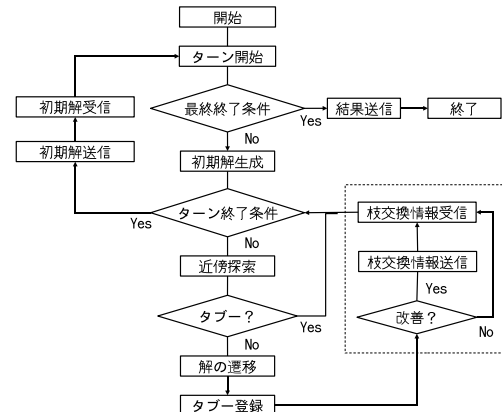


図 1 並列タブーサーチの流れ図

情報の送受信を行わなかった場合との比較も行う。2-opt 法のみでの探索に関しては、局所最適解に至ると直ちに次のターンを開始する。

実験対象とする問題は最適解が既知である、ウェブサイト上の TSPLIB から lin318, pr439, rat575 を使用し、各問題に対して 5 回づつ実験を行う。

動作環境: 以下の計算機を 8 台用いて実験を行う。

CPU: PentiumIII 1GHz
Memory: 512MB
OS: Linux
分散環境: SCore (MPI)

パラメータ設定: 以下のパラメータ設定で実験を行う。

- 2 ターン目以降の初期解作成時に、巡回路を構成する枝の中から 10% を長期リストのランキング順にタブーとして短期リストに登録する。
- ターンの終了条件はターンの中で最小の解が出てから 300 回を越えても解が改善されない場合か、探索が改善の方向に進み、そのターン内の最小の解より 50%を越える解に到達した場合終了する。
- 最終終了条件は探索時間が 240 分 (並列の場合は 1 プロセス 30 分 (30 分 × 8 台) として実行する) になると終了する。

4.2 結果

表 2~4 は各問題において計算機実験を行った結果を示している。表では 5 回行った実験の平均と求めた最小解を示しており、誤差は既知の最適解を基に計算した結果である。

<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
SCore Cluster System Software: <http://www.pcluster.org/>

表 2 lin318(最適解:42029)の結果

lin318		length	誤差(%)	ターン数
並列	平均	42773.0	1.77	16.2
	最小	42533.7	1.20	19.0
並列 (通信無)	平均	42823.9	1.89	19.2
	最小	42597.6	1.35	19.0
逐次	平均	42870.4	2.00	150.2
	最小	42764.0	1.42	157.0
2-opt	平均	43395.8	3.25	33258.8
	最小	43395.8	3.25	33263.0

表 3 pr439(最適解:107217)の結果

pr439		length	誤差(%)	ターン数
並列	平均	109272.1	1.92	5.8
	最小	108737.1	1.42	5.0
並列 (通信無)	平均	109838.5	2.45	8.6
	最小	109274.0	1.92	6.0
逐次	平均	109987.9	2.58	62.0
	最小	109550.3	2.18	67.0
2-opt	平均	110460.1	3.02	18302.0
	最小	110196.7	2.78	18305.0

表 4 rat575(最適解:6773)の結果

rat575		length	誤差(%)	ターン数
並列	平均	7046.6	4.04	3.8
	最小	7023.9	3.70	3.0
並列 (通信無)	平均	7068.0	4.36	5.4
	最小	7051.2	4.11	5.0
逐次	平均	7070.3	4.39	40.8
	最小	7015.0	3.57	43.0
2-opt	平均	7103.8	4.88	10428.8
	最小	7084.4	4.60	10433.0

表の結果から、全ての問題において並列タブーサーチの実行が最も良好な結果を得た。またタブーサーチを用いなかった 2-opt のみの手法では、逐次タブーサーチの結果より誤差が大きくなった。

並列探索の通信部分を取り除いた場合では送受信を行った場合より誤差が大きい結果となったがターン数は増えた。3種類のタブーサーチ(表の上から3つ)の平均を比べると全ての都市数において並列、並列で通信を取り除いた場合、逐次の順で最適解に近い解を得た。

4.3 考察

結果より並列タブーサーチの探索手法が有効に働いていることがわかった。これは並列で探索を行うことにより各プロセスが別の解空間を探索し、その時得られた情報を共有しており、より効果的な探索ができているからであると考えられる。

通信を行わなかった並列タブーサーチ手法はターン数及び遷移回数が増えていることがわかった。これは送受信が行われなくなったことにより、解の遷移に費

やされる時間が増し、より多く探索できていると考えられる。しかし、枝交換の情報が送受信されず長期リストに反映できないため、解は送受信を行った場合より劣る結果となった。

通信部分を取り除いた結果が逐次タブーサーチの結果より良い結果を得たのは、ターン終了後他のプロセスに初期解の送受信を行い、その初期解についての情報が探索に役立った可能性がある。

5. おわりに

本研究では、巡回セールスマン問題を例とし並列タブーサーチを用いて解を求める手法を対象にして、並列化での各プロセスが行う通信手法について注目し実験を行い近似解を求めた。その結果、解の遷移が改善に働くたびに行われる送受信が探索に影響を与えるが、即時適用回数は少なく無駄が多いことがわかった。しかし、同時に枝交換情報によって得られる長期リストの情報が探索に多様さを与え探索効率を上げている事が計算機実験の比較結果から考察できた。

そこで今後の課題として、この送受信を遷移後改善に働いた時に即時に行うのではなく、ある程度枝交換情報を自身で保存しておき、定期的にその情報を交換することにより、よりよい探索を実現する事が挙げられる。

参考文献

- 1) 山本芳嗣, 久保幹雄: 巡回セールスマン問題への招待, 朝倉書店, (1997).
- 2) Sadiq M.Sait, Habib Youssef 著, 白石洋一訳: 組合せ最適化アルゴリズムの最新手法-基礎から工学応用まで- 丸善株式会社, (2002)
- 3) Colin R. Reeves: モダンヒューリスティクス-組合せ最適化の先端手法-, 日刊工業新聞社, (1997).
- 4) Miroslaw Malek, Mohan Guruswamy, Mihir Pandya: Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem, Annals of Operation Research, 21(1989) 59-84
- 5) E.G. Talbi, Z. Hafidi, J-M. Geib: A parallel adaptive tabu search approach, (1998)
- 6) 杉原厚吉, 茨木俊秀, 浅野孝夫, 山下雅史: アルゴリズム工学-計算困難問題への挑戦-, 共立出版, (2001).
- 7) 作見雄一, 大西克実, 中野秀男, 榎原 博之: タブーリスト共有による並列探索, 情報処理学会 研究報告(MPS), 2004-MPS-52, Vol.2004, No.10, pp.37-40(2004)