

タブーリスト共有による並列探索

作 見 雄 一[†] 大 西 克 実[†]
中 野 秀 男[†] 榎 原 博 之^{††}

タブーサーチは、メタヒューリスティックの戦略の1つとして組合せ最適化問題の近似解を求めるために利用される。近似解の精度を向上させる手法として、探索の並列化が考えられる。本研究ではタブーサーチを並列化するにあたって、プロセス間で枝交換の情報を送信し、タブーリストを共有する仕組みに重点をおく。問題の具体例としては、巡回セールスマン問題を対象とし、並列タブーサーチのプログラムを用意して計算機実験を行い評価する。

An Implementation of Parallel tabu search with shared tabu lists

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

Tabu search is one of the meta heuristic strategy, and it is used for an approximation method for combinatorial optimization problems. Parallelization of the tabu search is also widely investigated in order to improve its performance. In this paper we propose an implementation of parallel tabu search with shared tabu lists. Tabu lists are shared among processors by means of mutual communication about the rearrangement of edges in the neighborhood search. We prepared the computational experiments of this parallel tabu search for the travelling salesman problem, and then examine the results.

1. はじめに

組み合わせ最適化問題の多くは、NP 困難であることが知られており、多くの場合では、現実的な精度の解が求まる近似解法や発見的手法が用いられている。近年のコンピュータの処理速度の向上により、近似解法と様々な戦略を組み合わせるより精度の高い解を求める枠組みであるメタヒューリスティックの研究⁶⁾が盛んに行われている。

メタヒューリスティックの枠組みは、初期解から探索戦略に従って局所探索法を繰り返し、条件に従って改善作業を終了する。代表的な探索戦略には、遺伝アルゴリズム、アニーリング法²⁾、タブーサーチ¹⁾等がある。これらの探索戦略をより高速で精度の高い解を求める方法の1つとして探索の並列化がある。

本研究では、タブーサーチを並列化するにあたって、近傍探索での枝交換に関する情報を各プロセッサ間で

送受信することによりタブーリストを共有する仕組みに重点をおく。タブーリストに探索期間の短期と長期に対応するリストを用意し、長期リストの情報を並列するプロセス間で共有する。

代表的な組合せ最適化問題の1つである巡回セールスマン問題⁵⁾を対象としたプログラムを作成し、計算機実験により、近似解を求め評価を行う。

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章ではタブーリスト共有方法を4章で実験結果と考察について述べる。

2. 準 備

2.1 組合せ最適化問題

組合せ最適化問題とは、与えられた制約条件を基に目的関数の値を評価し、最小もしくは最大になるような組合せを求める問題である。

組合せ最適化問題は、一般的に以下のように表される：

目的関数 $f(x)$

実行可能領域 F

制約条件 $x \in F$

$f(x)$ を最小もしくは最大にする実行可能解を最適

[†] 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

解と呼ぶ。多くの組合せ最適化問題について、計算の複雑さの理論から厳密な最適解を求めることが、NP 困難であることが知られている。

2.2 巡回セールスマン問題

巡回セールスマン問題は、代表的な組合せ最適化問題の1つである。訪問対象となる n 個の都市を1度ずつ訪問して出発した都市に戻る巡回路の距離が最小のものを求める問題である。

n 個の都市の集合 $V = \{1, \dots, n\}$ と、都市 i と都市 j の間の距離を C_{ij} とすると、以下のように定式化できる：

$$f(x) = \sum_{k=1}^{n-1} C_{x(k), x(k+1)} + C_{x(n), x(1)}$$

$x(k) = i$ は、 k 番目におとづれる都市が i であることを表す。

2.3 タブーサーチ

タブーサーチは、メタヒューリスティックの代表的な探索戦略の1つである。

タブーサーチは、初期解から以下のアルゴリズムに従って解を探索する。

- (1) 局所探索を繰り返し、遷移可能な解を求める
- (2) 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
- (3) 局所最適解に至っても、隣り合う解への遷移を許し探索を続ける
- (4) 指定の終了条件になれば探索を終了する

タブーサーチでは、解を遷移するときの情報をタブーリストと呼ばれるリストに保存し、遷移前の解をタブー解とする。解の探索過程において局所最適解に至っても解の遷移を許し、探索を続行する。探索過程においてタブー解への遷移を禁止することで、過去に探索した解に遷移することを防ぎ、未知の解への遷移を可能とする。

タブーサーチでは、タブーリストに登録された解をタブー解として扱う期間を設定する、タブー期間の長さは、タブーサーチの探索の重要なパラメータとして扱われる。

タブーリストの実装には、複数のアイデア⁶⁾があり、単にタブー期間中の解を保存するだけでなく、解の探索過程に役立つ情報の収集も可能であり、実装するにあたって、以下の点について注意が必要である。

- (1) タブー期間の設定 は、探索に直接影響があり長期に設定すると探索可能な解がなくなることや、タブーリストが肥大化し実験環境のメモリを圧迫することが想定される。



図 1 2-opt

- (2) タブーリストの構成 は、短期的なタブー解の保存だけでなく、探索に役立てるための長期的な情報などさまざまなアイデアを盛り込むことが可能である反面、アルゴリズムが複雑化しやすい。結果、プログラムが正しく実装されているか確認しにくくなる。
- (3) サイクリング は、同じ改善改悪を繰り返し実施することである。タブー期間の設定にも関係し、長期的に見て同じことを繰り返した場合、未知の解への遷移がなくなり、より最良の解を発見することが不可能になる。

2.4 近傍探索

タブーサーチは探索戦略であり、局所探索で行う解の遷移の実施には別に近傍探索のアルゴリズムを必要とする。本研究では、巡回セールスマン問題での代表的な 2-opt を採用した。

2-opt では、都市と都市を結ぶ枝を 2 つ選択し、枝の交換を行うことで、巡回路の改善・改悪を行う(図 1)。

3. タブーリスト共有による並列探索

3.1 並列探索

本研究では、ランダムに生成した複数の初期解から始める逐次探索を並列に実施する方法⁴⁾を採用し、探索戦略にタブーサーチを使用する。また、探索戦略の評価をわかりやすくするため、近傍探索にはアルゴリズムが簡易な 2-opt を採用する。

各プロセスでは、探索の過程において 2-opt による枝交換が改善に働いたとき、枝交換の情報を他のプロセスに送信する。受信した枝交換の情報が、自身の解空間に適用可能かつ改善に働く場合は、即時に適用し、解を更新する。

3.2 タブーリストの共有

タブーリストは、短期リストと長期リストを用意する。各リストで扱うデータの単位は枝交換の情報とし、これをタブーで扱う単位とする。

タブーリストの共有は、各プロセス内の近傍探索で枝交換が改善に働いたときにプロセス間で送受信される情報を長期リストに反映し、共通の枝交換情報を各

プロセスで保持することで実現する。

各リストの役割は、以下の通りである。

短期リスト タブーとなった枝交換の情報を管理するリスト。改善・改悪時に発生した情報を保存する。ターン毎に初期化される。

長期リスト タブーの登録状況を解析し蓄積するリスト。プロセス内で発生した枝交換の情報に加えて、他プロセスから受信した枝交換の情報も保存する。解析内容には、それぞれの枝の交換回数と枝の長さにかけてランキングを保存する。長期リストで集計された情報は、2 ターン目以降の初期解の生成時に活用する。

3.3 繰り返し操作

繰り返し操作の扱いについて解説する。

- 一つの探索を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
 - 1 ターン目では、最近傍法で生成した初期解を利用する
 - 2 ターン目からは、長期リストの内容を反映した初期解を生成して探索を行う。初期解生成時は、
 - 長期リストのランキング上位の枝を優先的に巡回路に盛り込む。
 - 採用された枝の情報を短期リストの初期値として登録し、タブーとして扱う。
 - 最終終了条件を満たしたら最小の解を出力して全体の処理を終了する。最終終了条件は、探索全体で行える解の遷移回数とする。
- 1 ターンの終了条件を以下のように定める。
- そのターンの中で最小の解が出てからパラメータで指定した回数の局所探索を行い、最小の解が更新されなかった場合。
 - 探索が、改悪の方向に進み、そのターン内の最小の解より指定の割合を超える解に到達した場合。

4. 実験結果と考察

実験は、下記の性能の計算機を8台用いて行った。

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

実験対象の問題は、TSPLIB のウェブサイトから lin318, pr439, rat575 を入手した。

比較対象として、以下のような並列化を適用しない

単一プロセスのプログラムを用意した。

- タブーサーチを用い、ターンの繰り返しや短期・長期のリストについては、並列タブーサーチと同じ動作をする。
 - 2-opt による局所探索法のみ動作し、長期リストのような解析は行わず、最近傍法で生成した初期解のみを使い、局所最適解に至ったら直ちに次のターンを開始する。
- 各プログラムは、実行するにあたって、以下のパラメータを設定した。
- 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。
 - ターンの終了条件を2つ用意し、どちらかが満たされればターンを終了する。
 - ターン内の最小の解を取得してから解の遷移回数が 300 回になっても最小の解が更新されないとき。
 - ターン内の最小の解から 50% を超える改悪が発生したとき。
 - 最終終了条件をプロセス毎に 10000 回の遷移回数とする。

表 1~3 は、TSPLIB から入手した各問題について、5 回ずつ実験を行った結果をまとめたものである。誤差は、既知の最適解を基に計算した。

“平均” は、各項目について、総処理時間、解、総ターン数のそれぞれの問題例について 5 回の平均値を記した。“最小” は、最小値が取得できた実験の回における総処理時間、最小値、総ターン数を記した。

近似解について、並列タブーサーチはタブーサーチよりも良好な結果が確認できた。並列タブーサーチとタブーサーチの違いは、

- (1) 解交換情報の多様さ
- (2) 他プロセスから受信した枝交換情報が即時に適用される場合がある

の2点が挙げられる

並列タブーサーチでは、改善に働いた枝交換情報を並列する各プロセスに送受信している。各プロセスでは、それぞれで初期解を生成し探索を行っているため、各プロセッサからの多様な枝交換情報が取得できる。そのため、長期リストで集計したランキング情報の精度も高くなると考えられる。対してタブーサーチのみでは、自身が探索中に集計した枝交換の情報から長期リストのランキング情報を生成しているが、並列タブーサーチほどの精度に至っていないと考えられる。長期リストで集計した枝のランキング情報は、ター

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

表 1 lin318(最適解:42029) の結果

lin318		時間(秒)	length	誤差(%)	ターン数
並列有	平均	1178.6	42920.0	2.12	14.5
	最小	1051.3	42423.1	0.94	13.0
並列無	平均	1052.9	43176.2	2.73	16.0
	最小	1036.1	42887.8	2.04	16.0
2-opt	平均	45.7	44003.7	4.70	102.2
	最小	46.1	43749.8	4.09	103.0

表 2 pr439(最適解:107217) の結果

pr439		時間(秒)	length	誤差(%)	ターン数
並列有	平均	2669.2	110729.6	3.28	13.4
	最小	2643.0	109674.1	2.29	11.0
並列無	平均	2221.2	111288.1	3.80	14.5
	最小	2181.0	110472.2	3.04	14.0
2-opt	平均	73.0	113166.0	5.55	92.2
	最小	73.0	110981.5	3.51	92.0

表 3 rat575(最適解:6773) の結果

rat575		時間(秒)	length	誤差(%)	ターン数
並列有	平均	4273.3	7046.7	4.04	11.34
	最小	4176.6	7017.3	3.06	9.0
並列無	平均	3250.5	7102.9	4.87	13.4
	最小	3456.0	7067.3	4.35	14.0
2-opt	平均	94.0	7189.6	6.15	65.8
	最小	92.1	7161.1	5.73	64.0

ンの始めに生成する初期解に反映しているのに加え、ターン開始時の初期のタブーとして短期リストにも反映している。ランキング情報の精度が高くなると、改善対象となる枝の探索に有効に機能し、まだ探索されていない解への遷移が行われやすと考えられる³⁾。

表 4 は、探索全体での枝交換の受信と即時適用の平均回数である。全受信回数に対して適用回数の割合は、0.5~0.6% ぐらいである。それぞれのプロセス内で初期解を独自に生成しているため、ターン内での適用が困難になっていると考えられる。

実行時間について考察すると、並列タブーサーチの枝交換情報の送受信は、枝交換が改善に働くたびにその情報を自プロセス以外のプロセスに送信している。そのため、並列タブーサーチの実行時間が並列無しの場合より長くかかっている。2-opt と比較すると、通信にかかる時間に加えて、初期解の生成と長期リストの更新に多くの時間がかかっていると考えられる。

5. おわりに

本研究では、タブーサーチを並列化するにあたって、各プロセッサ間で枝交換の情報を送受信し、タブーリ

表 4 枝交換の受信と適用の回数

問題	受信回数	適用回数
lin318	34630.9	236.5
pr439	34699.1	185.0
rat575	34752.0	198.7

ストを共有する仕組みに重点をおき、近似解を求めた。長期リストの扱いについて、並列化に有効な方法を検討し実装をおこなった。実行時間が長くなってしまい、通信方法や初期解の生成方法など、実装の見直しをすべき点も見つかった。送受信等も含め実行時間の改善は、今後の課題としたい。

2 ターン目以降について、ターン開始時に設定する短期リストは、通常のタブーと同じにしていたため、探索に対してどのような影響をあたえたか、今回の実験では考察ができていない。ターン開始時には、長期リストのランキング情報を使った初期解の生成に加えて、探索開始時にタブー期間を長めにとることや、タブーに設定する割合を増やすことで探索空間の縮小³⁾をはかることも可能だと考えている。

研究の中心に探索戦略をおいたため、近傍探索には簡単でわかりやすい 2-opt を採用した。実験結果では、既知の最適解には至らなかったこともあり、今後の課題として、精度の高い近傍探索法を採用することも検討したい。

参 考 文 献

- 1) C.-N. Fiechter: A parallel tabu search algorithm for large traveling salesman problems, Discrete Applied Mathematics 51 (1994) 243-267.
- 2) D.Janaki Ram, T.H.Sreenivas, K.Ganapathy Subramaniam: Parallel Simulated Annealing Algorithms, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING 37, 207-212 (1996) ARTICLE NO. 0121
- 3) 沼田一道, 岩倉行信: 解の部分固定により探索空間を縮小するメタ戦略の検討 -巡回セールスマン問題への適用, システム制御情報学会論文誌 Vol. 17, No. 3, pp. 103-112 (2004).
- 4) 半田祐一, 小野廣隆, 定兼邦彦, 山下雅史: 組合せ最適化問題に対する局所探索アルゴリズムの並列化手法の設計, 情報処理学会 火の国情報シンポジウム 2004, (2004).
- 5) 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).
- 6) 柳浦睦憲, 茨木俊秀: 組合せ最適化-メタ戦略を中心として-, 朝倉書店, (2001).