

巡回セールスマン問題に対するメタ・ヒューリスティックスの並列実行

杉元 貴洋、中条 恭也 (沼田 一道 助教授)

1. はじめに 問題

巡回セールスマン問題とは、組合せ最適化問題に属し、中でも特に難しい問題の一つとされている。組合せ最適化問題は実行可能解の数が有限個であり、全解列挙することにより厳密な最適解を求めることができるが、問題の規模が大きくなるにつれて列挙数が飛躍的に増加するので、実用的な計算時間で厳密な最適解を求めるのは非常に困難である。そのため様々な発見的解法(Heuristics)が提案されてきている。最近では、アニーリング法、遺伝的アルゴリズム、RSB法のようなメタ・ヒューリスティックス(Meta Heuristics)と呼ばれる解探索の一般的な枠組みを巡回セールスマン問題に適用する試みも多い。メタ・ヒューリスティックスは手軽に構成できる反面、同じ近似解法であるヒューリスティックスと比べるとかなり実行時間を要する。

そこで本研究では、複数台のコンピュータを用い、メタ・ヒューリスティックスを並列計算させることによって実行時間の短縮を図る。メタ・ヒューリスティックスとして、頑健で高精度であるがやや時間のかかるRSB法を取り上げる。RSB法は独立な局所最適解を多数生成するので並列計算に適していると考えられる。これらを用いて数値実験を行い、実行時間の短縮を確認する。また、RSB法で使用する逐次改善法の違いによる実行時間と精度の比較を行う。

2. 巡回セールスマン問題

巡回セールスマン問題 (Traveling Salesman Problem : TSP) は、与えられた都市をすべて訪問するとき、距離や時間等に基づく総移動費用が最小となる巡回路を求める問題である。

本研究では、都市 i から都市 j に移動する時と都市 j から都市 i に移動する時の費用が等しい対象型 TSP を取り扱う。以下に使用する記号を表記する。

n : 都市数

S_n : $\{1, \dots, n\}$ の順列全体の集合

$\sigma(i)$: 順列 σ において、 i 番目に訪問する都市 $(1, \dots, n)$
ここで、 $\sigma(n+1) = \sigma(1)$ とする

C_{ij} : 都市 i, j 間の移動費用 ($C_{ij} = C_{ji}, i, j = 1, \dots, n, i \neq j$)

Z : 巡回路の総移動費用

以上より巡回セールスマン問題 (TSP) は、次のように定式化される。

$$\text{Minimize } Z = \sum_{i=1}^n C_{\sigma(i)\sigma(i+1)}, \quad \text{Subject to } \sigma \in S_n$$

3. RSB 法

RSB法(Reduction by Selection Bias)とは、多数発生させた局所最適解の上位何個かを選び、その共通部分を「最適解の構成要素」とみなして固定し、問題のサイズを縮小する。問題のサイズが充分小さくなるまで、これを繰り返す方法である。RSB法の動作は以下のように書ける。

- step 1: 問題のサイズを $s:=n$ とする (n :都市数) *
- step 2: ランダムに選んだ初期解を逐次改善して局所最適解を求めることを L 回行う
- step 3: L 個の局所最適解から目的関数値の良いものを k 個選び出す
共通部分が無ければ、ステップ 2 に戻る
- step 4: しかるべき長さ以上の共通部分を最適解の構成要素とみなして固定し、問題のサイズを縮小する (s を更新する)
- step 5: s が充分小さくなったらステップ 6 に進む、そうでなければ、縮小された問題を新たな問題としてステップ 2 に戻る
- step 6: 元の n 都市の問題に戻し、今までに得られた最良の解を出力する

3. 1 RSB 法の巡回セールスマン問題への適用

前節の step2 で用いる逐次改善法としては、2-opt 法などの比較的短時間で局所最適解を出力するものが望ましい。RSB 法の進行に伴い都市は (1) 両隣の都市が共に固定されていない都市、(2) 片隣の都市固定された都市、(3) 両隣の都市が固定された都市の 3 種類に分類される。最初の Step2 では、すべての都市が (1) に属する。問題が縮小されると、step2 では、本来の都市 (1) と、固定された都市列の両端の都市 (2) が対象となる。

4. 2-opt 法、3-opt 法

2-opt 法とは、ある巡回路においてある 2 本の枝を取り除き、別の 2 本の枝を付け加えて再び巡回路を作ることを考える。この時、巡回路の長さが短くなる場合にはこの入れ替えを行い、改善されない場合には行わない。この操作を巡回路の長さを減少させる入れ替えがなくなるまで繰り返す方法である。例えば、図 1 の左図のような巡回路において、道 ab, cd を除き、代わりに道 ac, bd を加えたとき、巡回路の長さが減少すれば新しい道に交換する。

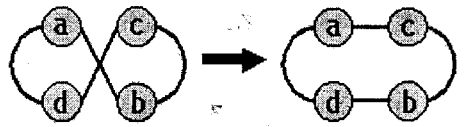


図 1 2-opt 法

同様に、3 本の枝の入れ替えを考えるのが 3-opt 法である。

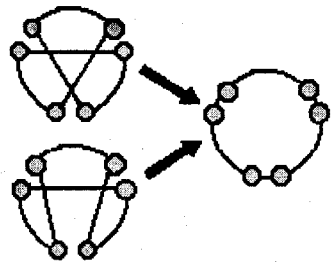


図 2 3-opt 法

5. RSB法の並列計算

RSB法を実行する場合、step2(ランダムに選んだ初期解を逐次改善して局所最適解を求めることを L 回行う)が計算時間の大半を占める。この計算は、それぞれ独立に行うことができるので、複数台のコンピュータで並列に実行させれば、計算時間の大幅な短縮が期待できる。

L は各コンピュータに均等に配分する。

5. 1 動作環境概要

複数のプログラムが協調して仕事を行うには、

- ① ネットワークで結ばれたコンピュータ間でのデータ送受
- ② 各コンピュータ内のプログラムと転送制御部との間でのデータ送受
- ③ プログラム間でのデータ交換上の取り決め

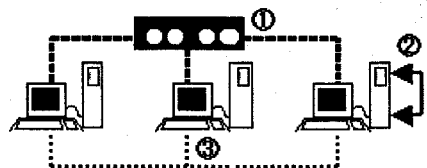


図 3 ネットワーク

が必要となる。①については、OSI 参照モデルとネットワークに遍在する伝送制御機構（プロトコル）を利用する。②については、プログラムの記述に用いた言語（Inprise Delphi3.1）が提供するシステム呼び出し機構を利用する。③については、クライアント/サーバ型のシステムを基本とし、プログラムの開始、問題の送信、問題の縮小、出力、終了等を行うクライアントプログラムと、必要個数分の局所最適解の生成、計算結果の送信等のサーバプログラムで構成する。

5. 2 プログラムの概要

S . 8

以下に作成したサーバ/クライアントプログラムの状態の遷移を示す。

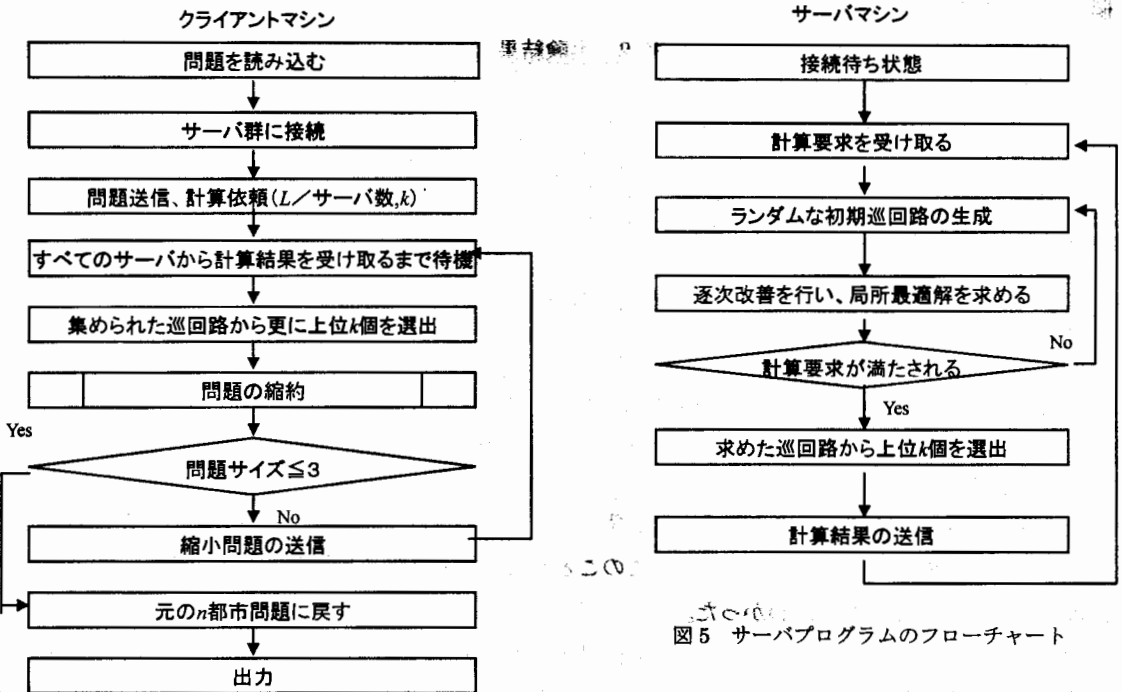


図5 サーバプログラムのフローチャート

図4 クライアントプログラムのフローチャート

5. 2. 1 サーバ/クライアント間の送信内容

- ・ クライアント⇒サーバ
 - 1回目 【問題都市数、都市座標、逐次改善法、巡回路生成回数、巡回路取出個数、サーバ番号】
 - 2回目以降 【縮約問題都市数、縮約問題都市番号、都市連結情報】
- ・ サーバ⇒クライアント
 - 【(巡回路、巡回路長) × k 個、サーバ番号】

6. 数値実験

6. 1 実験方法

数値実験では、TSPのベンチマーク問題集であるTSPLIBのatt48とkroA100を使用した。RSB法における逐次改善法としては、2-opt法と3-opt法、2-opt法が収束してから3-opt法を適用する3種類を使用した。使用したコンピュータは以下の通りである。

表 1 使用コンピューター一覧

CPU、メモリ	使用台数	1台	3台	4台
AMD K-6 300MHz, 62.0MB		○	サーバ	サーバ
PenIII 450MHz, 127.0MB		—	クライアント	クライアント
PenII 300MHz, 64.0MB		—	サーバ	サーバ
PenII 450MHz, 255.0MB		—	—	サーバ

6. 2 実験結果

計算機による実験結果を以下の表に示す。準最適値、実行時間は10回の試行の平均による。準最適値欄のカッコ内は最適解が得られた回数を示す。

表 2 実験結果 (1)

問題名 (最適値)	台数 (サーバ数)	逐次改善法 (L, k)	準最適値	実行時間[s]	短縮時間比率
att48 (10628)	1	2-opt (300,7)	10631 (7)	3.85	—
att48 (10628)	3 (2)	2-opt (300,7)	10631.5 (8)	2.12	0.55
att48 (10628)	4 (3)	2-opt (300,7)	10634.5 (6)	1.39	0.36
kroA100(21282)	1	2-opt (300,7)	21282 (10)	41.15	—
kroA100(21282)	3 (2)	2-opt (300,7)	21282 (10)	20.44	0.49

表 3 実験結果 (2)

問題名 (最適値)	台数 (サーバ数)	逐次改善法 (L, k)	準最適値	準最適解/最適解	実行時間[s]
att48 (10628)	3 (2)	2-opt (100,3)	10643 (2)	1.0014	0.81
att48 (10628)	3 (2)	3-opt (100,3)	10628 (10)	1	18.79
att48 (10628)	3 (2)	2⇒3-opt(100,3)	10628 (10)	1	11.78

6. 3 考察

表 2 から、並列計算させることによって、実行時間が短縮されることが確認された。また、短縮時間比率は約 $(1 / \text{サーバ数})$ になっている。このことから、並列計算させる際に生ずる通信時間等のロス はほとんど影響しないことがわかった。

表 3 では、3-opt法と2⇒3-opt法とも10回とも最適解を示しており、精度の高さが伺える。二つの実行時間をみると、2⇒3-opt法の方が3-opt法より約40%早くなっている。このことから、3-opt法を単体で適用するより、2-opt法で一度収束させ3-opt法を適用した方が効率が良いことがわかる。また、2-opt法では、他の二つからみれば精度は充分なものではないが、実行時間は極めて早く実用的である。

7. まとめ

本研究では、複数のコンピュータで並列計算させることにより、RSB法の実行時間短縮を試みた。この方法により、かなりの時間軽減がなされ、並列計算の有効性が確認できた。また、逐次改善法の工夫である、2-opt法で収束させ3-opt法を用いる方法でも、予想以上の結果が得られた。今回の実験では、4台 (サーバ3台) までしか使用できなかったが、台数制限はないのでサーバ台数が増えるほど時間の短縮につながることが予測できる。しかし、今回作成したプログラムでは、一番遅いコンピュータに全てのコンピュータが合わせてしまうため、速いコンピュータは十分に性能を発揮できていない。クロック数の違いによる繰り返し回数の配分、逐次改善法の高速度等は今後の課題である。

【参考文献】

- [1]鈴木 宏史、三塚 信也：「ネットワークを介して対戦可能なダイヤモンドゲームの試作」 卒業論文 1998年
- [2]児玉 淳：「巡回セールスマン問題に対する発見的解法の提案」 卒業論文 1996年
- [3]山本 芳詞、久保 幹雄：「巡回セールスマン問題への招待」 朝倉書店 1997年