A-020

遺伝的プログラミングによる巡回セールスマン問題の解法について
On the Solution of Traveling Salesman Problems Using Genetic Programming

原　裕一†　　山内　仁‡　　金川 明弘‡　　高橋 浩光‡

Yuichi Hara　Hitoshi Yamauchi　Akihiro Kanagawa　Hiromitsu Takahashi

1　はじめに

遺伝的プログラミング (GP) は遺伝的アルゴリズム (GA) の遺伝子型を木構造に拡張することにより，GA より幅広い分野への適応を可能にした手法である．

GA をはじめとするメタ解法の適用が期待できる分野の一つに NP 困難な組合せ最適化問題がある．しかしながら GP に関しては組合せ最適化問題への適用例は少ない．その理由は，巡回セールスマン問題 (TSP) のような解の形態が順列形で表現されるような問題に対して，遺伝的プログラミングで扱う遺伝子型 (染色体) のリスト構造が適合していないことがあげられる．

本研究では組合せ最適化問題のうち，代表的な問題である巡回セールスマン問題に対して，GP を用いて求解する方法を提案する．具体的には，解そのものを遺伝させるのではなく，現存する TSP 近似解法に対して，経路を構築する手段をリスト構造により適応的に選択させることにより，最適なヒューリスティック解法の手順を構築することで実現する．またベンチマーク問題を通して GA との解の比較を行う．

2　遺伝的プログラミング

遺伝的プログラミング (GP) は 1990 年代初頭に Koza[1] により提案された探索手法であり，遺伝的アルゴリズム (GA) に対して，構造的表現を直接遺伝子コードとして扱えるように拡張した概念である．このことにより従来の GA の適用範囲にとどまらず，ブール関数の合成や，法則の発見，人工生命のシミュレートなど，GA と比較してさらに発展的な内容を扱えるようになった [2]．

具体的には遺伝子型を GA では一次元の配列で扱っていたのに対し GP では遺伝子型を木構造 (遺伝木) で表現することにより問題の値そのものではなく最適な構造パターンを探索することで実現している．

GP の特徴としては木の葉の部分を終端記号，枝の部分を非終端記号と呼び，それぞれが別々の意味を持っている．また一般的な交叉や突然変異の操作で木の構成要素の数が増減する可変長型遺伝子による探索であり，それに応じた問題の設計が必要となる．

3　巡回セールスマン問題

巡回セールスマン問題（TSP）は与えられた N ヶ所の都市を一人のセールスマンが一つの都市から出発して全ての都市を唯一度だけ通ってもとの都市に戻ってくるとき，最短経路を与える都市の訪問順序を求める問題である．

この問題が非常に有名であるのは，多項式時間では求解できない NP 困難な問題であり，搬送問題や LSI の設計問題等の実際の応用例がある上に，ベンチマーク問題も整備され，研究例が豊富であることから，新しい解法の試金石として格好の題材といえるからである．

TSP の解法に関しては，分枝限定法や切除平面法のように厳密な最適解算出を志向する方法と，大きな都市数の問題に対しても比較的短時間で準最適解を求める近似解法の立場がある．

以下に TSP の近似解法の中で発見的解法 (ヒューリスティック解法) として知られているもののうち代表的なものをあげる [3, 4]．ここで巡回路のコストとはその巡回路の総経路長のことを表す．

3.1　最近近傍法 (Nearest Neighbor Method)

このアルゴリズムは，まずただ一つの都市からなる部分巡回路 T から初めて，T に一番最後に追加された都市を i として i に最も近い都市 k を i の隣に追加する処理を全ての都市を含むまで繰り返す方法である．

† 岡山県立大学 大学院 情報系工学研究科
‡ 岡山県立大学 情報工学部

3.2 最近追加法 (Nearest Additional Method)

最近傍法とほぼ同様であるが，最近傍法と違い T に含まれる全ての都市と T に含まれていない全ての都市の組合せの中から距離が最短になる都市のペアを選ぶ $(i,\ k$ とする). その後 i の両隣の内，T のコストの増加が少ない方に k を追加する方法である.

3.3 最近挿入法 (Nearest Insersion Method)

最近追加法と同様の方法で i, k のペアを選ぶが k は i の隣りではなく全ての T に k が挿入できる場所を調べて，k が挿入されたとき最も T のコストの増加が少ない部分に挿入する方法である.

3.4 最安挿入法 (Cheapest Insersion Method)

T 内の隣接する都市 i と j の間に T に含まれていない都市 k を挿入する全ての組み合わせの中で最も T のコストの増加が少ない組合せ $\{i, k, j\}$ を選びその場所に k を挿入することを繰り返す方法である.

3.5 最遠挿入法 (Furthest Insersion Method)

T に含まれていない都市の中で T から最も遠い距離にある都市 (T との最短距離が最も長い都市)k を T のコストの増加が最小になる位置に挿入することを繰り返す方法である.

これらの解法は与えられた問題によって出力される解の精度が大きく違ったり，精度が悪くても最適解と多くの枝を共有しているため解の改善が容易であったりと一概にどれが優れているかとは言えない.

4 構築法を終端記号とするGPによるTSP解法の提案

前述したTSPのヒューリスティック解法は全てただ一つの都市からなる部分巡回路 T から始めて，T にそれぞれの条件と挿入場所で都市を1つずつ追加する点で一致した方法であり，なおかつこれらの方法にはランダムな要素を持っていないので最初に T に含まれる都市の選び方が一定なら出力は常に一定である.

このことに着目して前述のヒューリスティック解法の組み合わせることによって各問題に対して最適な巡回路の構築手順はどんなものであるかをGPで探索する方法を提案する.

すなわち，各ヒューリスティック解法を用いて都市を一つ追加するという行動を終端記号に用いる. また非終端記号としては T 内の都市の数 n_T として n_T の値や T のコストによって実行する枝を変えるような記号を用いることにする.

本発表では終端記号，非終端記号は表1，2のものを用いる. ただし N は問題の全都市数，$COST(T)$ は T の現在のコスト，$average$ は全ての都市間の平均の距離を表す.

表1: 終端記号の一覧

ラベル名	意味
(NNM)	最近近傍法で T に都市を1つ追加する.
(NAM)	最近追加法で T に都市を1つ追加する.
(NIM)	最近挿入法で T に都市を1つ追加する.
(CIM)	最安挿入法で T に都市を1つ追加する.
(FIM)	最遠挿入法で T に都市を1つ追加する.

表2: 非終端記号の一覧

ラベル名	引数の数	意味
(PROGN ...)	2~5	引数を順に実行する
(PRIME ...)	2	n_T が素数なら引数1 違うなら引数2を実行する.
(TURN3 ...)	3	$n_T < 1/3 \times N$ なら引数1 $n_T < 2/3 \times N$ なら引数2 それ以外は引数3を実行する.
(TURN2 ...)	2	$n_T < 1/2 \times N$ なら引数1 それ以外は引数2を実行する.
(MOD3 ...)	3	$n_T/3$ の余りが0なら引数1 余りが1なら引数2 それ以外は引数3を実行する.
(MOD2 ...)	2	n_T が偶数なら 引数1，奇数なら 引数2を実行する.
(AVE ...)	2	$COST(T)/n_T < average$ なら引数1 それ以外は引数2を実行する.

GPの遺伝木の生成アルゴリズムは以下の通り

1. まず木の根として非終端記号をランダムに一つ選

択する．

2. 非終端記号の引数の数 (PROGN は2〜5の中からランダムで選択) だけ以下の操作を行なう
   (a) 現在の深さを計る (根を0とする)
      - あらかじめ指定した深さ (初期個体の最大深さ) に達した場合は終端記号からランダムに一つ選択する．
      - そうでない場合は非終端記号と終端記号の中からランダムに一つ選択する．
   (b) 終端記号が選択された場合はその記号を返す．そうでない場合は現在の深さに1加えた (a) 以下の操作を引数の数だけ再帰的に行ない出力された木を子供にした木を返す．
3. 出力された木を根の子供にする．

遺伝木を持ったGPの個体は次のような操作で巡回路を生成する．まずただ一つの都市からなる部分巡回路 T を与えられる，ここで T の初期状態で含まれている唯一の都市は常に同じ都市にする．

T を与えられた個体は遺伝木の根の部分を実行する．各非終端記号は表2の通りに実行を行ない終端記号が実行された場合は対応する構築法を用いて T に都市を1つ追加するこのとき T は

1 - 2 - 3 - 4

のような1次元のリストの形で表され (NNM) はリストの末尾 (この場合は4) に最も近い都市を末尾に追加し，それ以外はリストの末尾が先頭に繋がっている ($1-2-3-4-1$ となっている) と解釈してそれぞれの構築法手順で都市を追加する．遺伝木の実行が終わっても T が全ての都市を含んでいない場合は T が全ての都市を含むまで実行を繰り返す．こうして完成した巡回路をこの個体の表現型とする．

5 比較実験

提案手法の有効性を確認するため，いくつかのベンチマーク問題集を解き，その解の性能をGAと比較する．ただし同じ進化的手法とはいえ，両者は解の表現形態が全く異なる上，提案手法は既存のヒューリスティック解法を用いているなど，求解のシステムが全く異っている．よって優劣の比較と言うより参考程度の意味しか持たないことに注意する．本研究で使用したパラメータは表3のとおりとする．

表 3: GP のパラメータ

個体数	200
最大世代数	50
適合度ペナルティ	要素数 $\times 10^{-4}$
初期個体の最大深さ	4
生成個体の最大深さ	8
交叉率	0.8
突然変異率	0.1
選択方式	ルーレット選択
エリート保存	1個体

一方，比較するGAのパラメータは表4のとおりとした．交叉，突然変異は主に文献[5]から，なるべく対等の条件での探索となるように個体数，選択方式，交叉率をGPと同じにして実験を行なった．

表 4: GA のパラメータ

個体数	200
最大世代数	5000
交叉率	0.8
交叉方法	枝交換交叉 (EXX)
突然変異率	0.01
突然変異方法	2点間の都市を逆順にする
選択方式	ルーレット選択
エリート保存	1個体

世代 t での個体 i の適合度 $\mathrm{fitness}_i(t)$ はGP，GAの両者ともに次式で与えられることとする．

$$\mathrm{fitness}_i(t) = \mathrm{worst}(t) - \mathrm{cost}_i(t) \tag{1}$$

このとき $\mathrm{cost}_i(t)$ は世代 t での個体 i の生成した巡回路のコスト，$\mathrm{worst}(t)$ は世代 t で生成された個体の中で最も長い巡回路を生成した個体のコストを表している．

対象とする問題として TSP のベンチマーク問題集 TSPLIB95[6] に収録されている問題の内，eil51，berlin52，st70，pr76，kroA100，lin105 の6種類の問題を使用した．

各問題に対して20回ずつ試行して，平均のコスト，実行時間，全試行中で最小のコストを表5に示す．

この結果GPでの探索はいずれもGAの実行結果より良質な巡回路を出力している．

またberlin52とpr76問題に関しては最適解を出力することに成功した．50都市以上の問題になると，ヒューリスティック解法の単独での使用では，真の最適解を得ることはなかなか期待し難い．しかしながら種々の解法を適応的に組み合わせることでいくつかの問題に対しては最適解を出力することが可能であったことは非常に興味深い成果である．

表 5: ベンチマーク問題での比較

(1) eil51 問題での適用結果

eil51(最適コスト 426)			
	平均コスト	処理時間 (s)	最小コスト
GP	429.55	24.3	427
GA	465.85	40.6	438

(2) berlin52 問題での適用結果

berlin52(最適コスト 7542)			
	平均コスト	処理時間 (s)	最小コスト
GP	7578.40	28.6	7542
GA	8391.70	41.7	7752

(3) st70 問題での適用結果

st70(最適コスト 675)			
	平均コスト	処理時間 (s)	最小コスト
GP	681.75	53.8	679
GA	875.55	56.1	769

(4) pr76 問題での適用結果

pr76(最適コスト 108159)			
	平均コスト	処理時間 (s)	最小コスト
GP	108533.15	68.9	108159
GA	148836.00	61.0	130627

(5) kroA100 問題での適用結果

kroA100(最適コスト 21282)			
	平均コスト	処理時間 (s)	最小コスト
GP	21443.40	158.6	21317
GA	34461.60	80.3	27680

(6) lin105 問題での適用結果

lin105(最適コスト 14379)			
	平均コスト	処理時間 (s)	最小コスト
GP	14586.15	157.6	14456
GA	18392.20	82.1	16538

しかしながら平均の処理時間を見ると都市数が増えるにつれ処理時間の増加が急激であることがわかる．これは終端記号に用いた構築法の多くが $O(n^3)$ であるので，新しい巡回路を生成する時，最悪の場合で $O(n^3)$ の時間を必要とするためである．

6 おわりに

本研究では現存する TSP のヒューリスティック解法の経路生成の方法を終端記号に用いる TSP の GP による解法を提案した．

また GA の TSP 解法としてよく知られた方法と，ベンチマーク問題を解くことにより，提案の GP による解法との比較を行なった．これらの比較では厳密な優劣を断じることはできないが，概して提案法の方が優れた解を得ることができた．

またいくつかの問題では最適解を出力することも出来た．提案の解法では構築法のみを用い改善法 (2-opt 法など) は用いていない．よって改善法も適用することにより，さらに良質の解を得ることが期待できる．

今後の課題としては，並列化など探索時間短縮に対する検討，より有効な非終端記号と終端記号の検討，TSP 以外の組合せ最適化問題に対する同様の方法の考案などが挙げられる．

参考文献

[1] J.R.Koza : Genetic Programming, The MIT Press, 1992.

[2] 伊庭斉志 : “遺伝的プログラミング入門”, 東京大学出版会, 2001.

[3] E.L.Lawler,J.K.Lenstra,A.H.G.Rinnooy Kan and D.B.Shmoys edited : “The Traveling Salesman Problem A Guided Tour of Combinational Optimization”, JOHN WILEY & SONS, 1985.

[4] 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.

[5] 三宮信夫, 喜多 一, 玉置 久, 岩本貴司 : “システム制御情報ライブラリー 17 遺伝アルゴリズムと最適化”, 朝倉書店, 1998.

[6] Gerhard Reinelt : TSPLIB95, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/