

# リン・カーニハン法による TSP 求解の高速化

吉原郁夫<sup>1)</sup> 大西亜実<sup>2)</sup> 黒田正文<sup>3)</sup> 山森一人<sup>4)</sup> 相川勝<sup>5)</sup>

## A Rapid Solution of Lin-Kernighan Algorithm for TSP

Ikuro YOSHIHARA Ami ONISHI Masafumi KURODA Kunihito YAMAMORI  
Masaru AIKAWA

### Abstract

Lin-Kernighan (LK) algorithm has been widely used to solve Traveling Salesman Problems (TSP). The LK method searches for the optimal solution by changing cities in a tour in the surrounding regions one after another. The search region is limited by the number of candidate cities to be changed, e.g. 10 or 20. Our basic idea is to limit the number of candidate cities to reduce CPU time according to pre-executions. Here, pre-executions are to solve small size TSPs whose cities are partly chosen from the original TSP and to collect the edges appeared in the solutions of the small problems.

Key Words:

Traveling Salesman Problem, Lin-Kernighan Algorithm, Improvement

### 1 はじめに

巡回セールスマン問題は、与えられた都市すべてを1回ずつ通って元の都市に戻ってくる巡回路のうち、移動距離が最短になる経路を求める問題のことである<sup>1)</sup>。

巡回セールスマン問題はNP困難の問題であり、準最適解を現実的な時間で求めるための近似算法、またはヒューリスティック手法と呼ばれる手法が多く提案されている<sup>2)</sup>。その中でも有名な解法がLin-Kernighan(LK)法である<sup>3)</sup>。

LK法は、改善法の1つであり、何らかの方法で巡回路を得た後、より距離の短い巡回路に改良していく。2都市間の経路のことをエッジと呼ぶことにする。巡回路の何本かのエッジ、これを $k$ 本としたとき、 $k$ 本のエッジを組み替え、巡回路の距離が短くなるようにする。この操作を距離の短い巡回路ができるまで繰り返し行うことで、最適巡回路を求めていく。この $k$ を変化させる方法

を提案したのがLK法である。

LK法では、ある都市 $t$ からエッジをつなぎ替える都市を探索する場合、都市 $t$ から距離の近い都市を何都市か選び、距離の近い順に探索する。選んだ都市の集まりのことを近傍都市と呼び、その近傍都市の数は、TSPを解く前に定めておく。巡回路が短くなるように探索した都市にエッジをつなぎ替えたなら、都市 $t$ を変えて同じように探索していく。

近傍都市を使って都市を探索する場合、つなぎ替えるエッジが見つからない場合、都市 $t$ は定めた近傍都市の数だけエッジの長さを比較しなければならない。離れた場所に都市がある場合、接続するような都市は限られてくる。しかし、近傍都市を使って探索する際、接続する可能性の低いような都市まで探索する分時間がかかってしまう。そこで、それぞれの都市に接続する可能性のある都市を調べ、可能性のある都市だけを探索する。一部の都市の集まりを選び、選んだ都市だけで巡回路をつくり、接続する都市を調べる。また別の部分の都市の集まりを選び、巡回路をつくり、接続する都市を調べる。この操作をくり返し何度も行うことで、どの都市と接続可能性が高いのかが

<sup>1)</sup>情報システム工学科教授

<sup>2)</sup>情報システム工学科学生

<sup>3)</sup>工学研究科修士1年

<sup>4)</sup>情報システム工学科准教授

<sup>5)</sup>工学部教育研究支援技術センター技術職員

わかる。接続する可能性の高い都市だけを探索することで、探索する都市の数を減らし、LK法の処理時間を短縮する。この方法を使って、ハイブリッド遺伝的アルゴリズム (HGA) に組み込むことで、処理時間の短縮をすることを意図している4)5)6)7)8)9)。

## 2 解法

### 2.1 LK法

$D(t_1, t_2)$  は都市  $t_1$  から、都市  $t_2$  までの距離を表す。

1度、ランダムな巡回路をつくる。その巡回路を1方向に、都市を探索していく。ある都市を都市  $a$  としたとき、その隣にある都市を都市  $b$  とする。都市  $b$  から1番近い都市を都市  $c$  とし、都市  $c$  の隣で、つなぎ替えたときに巡回路となるような都市を都市  $d$  とする。このとき、 $D(a, d) + D(b, c)$  と  $D(a, b) + D(c, d)$  を比較し、 $D(a, d) + D(b, c) < D(a, b) + D(c, d)$  ならエッジ  $ab$  とエッジ  $cd$  からエッジ  $ad$  とエッジ  $bc$  に入れ替え、より短い巡回路を作る。もし、エッジの入れ替えが行われなかった場合は、都市  $b$  から2番目に近い都市を都市  $c$  とする。エッジの入れ替えをした場合は、次の都市  $a$  を選び、すべての都市を探索し、 $D(a, d) + D(b, c) < D(a, b) + D(c, d)$  とならなくなるまで都市  $a$  を変えていき、都市の探索とエッジの入れ替えを繰り返す。

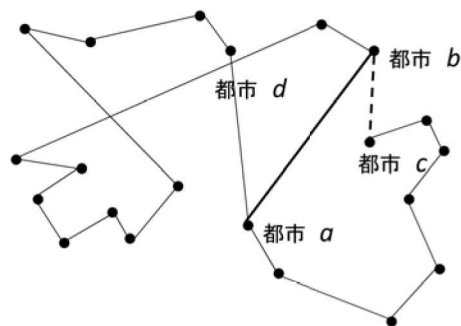


図1: エッジの比較

### 2.2 処理時間を短縮するための方法

2.1節の都市  $c$  を近傍都市から選ぶのではなく、都市  $c$  となる都市の候補をつくり、その候補から選ぶようにする。

都市数  $N$  の問題を解くとする。都市数  $N$  の問題を繰り返し解くことで、どの都市とどの都市が接続する可能性があるのかを調べ、都市  $c$  の候補をつくる。都市数  $N$  の問題を何度も解くと時間

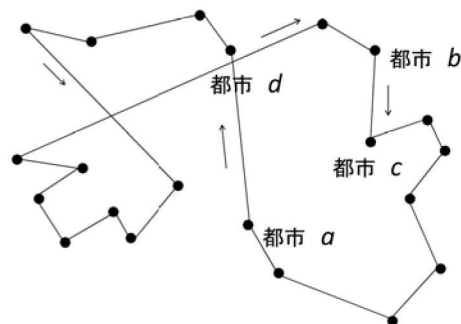


図2: エッジをつなぎ替えた巡回路

がかかりすぎるため、ある1都市をランダムに選び、その周辺の都市を使って小規模な巡回路をつくる。その時の小規模な巡回路の都市数を  $n$  とする。このときの  $n$  は  $N > n$  となるようにする。

都市数  $n$  をLK法で解き、巡回路を得た後、改めてランダムに1都市を選び、小規模な巡回路をつくる。繰り返し都市数  $n$  をLK法で解いて小規模な巡回路をつくる。できた複数の巡回路の内、それぞれの都市が接続したことのある都市を都市  $c$  の候補とし、接続した回数の多かった都市から順に探索を行うようにする。例えば、図3のように小規模な巡回路  $abcdea$  と巡回路  $efghdc$  の2つを重ねたとき、都市  $c$  に注目すると、都市  $b$  は1回、都市  $d$  は2回、都市  $f$  は1回、都市  $c$  と接続し、巡回路をつくったことになる。これを回数の多い順に並べ替えると、 $d, b, c$  の順になり、これを都市  $c$  の候補とする図4。候補をつくることで、接続しないような都市まで探索することがなくなるので、LK法の処理時間の短縮となる。

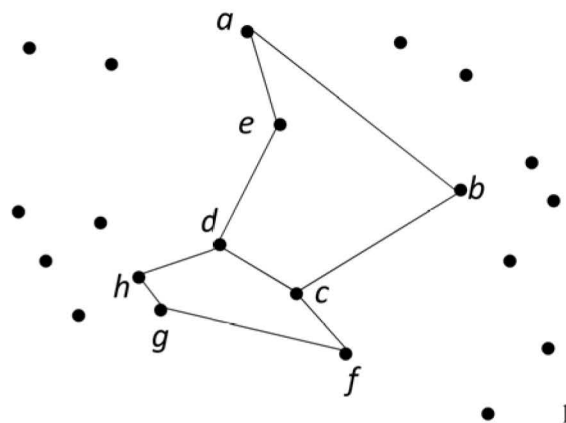


図3: 小規模な巡回路を生成

### 2.3 小規模な巡回路の都市数と生成回数

実験では小規模な巡回路を何度も生成しているが、その時の都市数を  $n$ 、生成した回数を  $r$  としたとき、その都市数  $n$  と生成した回数  $r$  はその

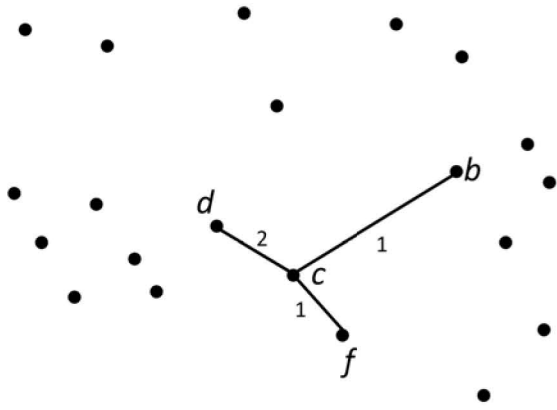


図 4: 都市  $c$  に注目したときのエッジの出現回数

都市問題の都市数によって変える必要がある。小規模な巡回路として出現している都市の回数は平均したときに、LK 法で解く際に定めた近傍都市の都市数より多くなるようにしている。何度も小規模な巡回路を生成し、いくつかの都市と接続したか、実験を繰り返してみたところ、都市が小規模な巡回路として出現する回数の平均が、近傍都市の都市数より多くしなければ、接続した都市数に極端な偏りができてしまう。都市の出現回数の平均が多ければ多いほど少しずつ解は良くなるが、時間がかかりすぎてしまうので、都市数で変えるようにしている。

### 3 実験

#### 3.1 ベンチマーク問題

実験では対象問題として、表 1 に示した TSPLIB<sup>10)</sup> と National TSPs<sup>11)</sup> のサイトの 6 つの問題を用いた。表の値は開始からそれぞれの工程終了時までの処理時間で単位は [s] である。

近傍都市の都市数は LK 法を解く際、一般に 10 から 20 とするため、実験では 20 近傍とした。

表 1: ベンチマーク

サイト	問題	最適解
TSPLIB	d493	35002
	nrw1379	56638
	fl3795	28772
	rl5915	565530
	brd14051	469385
National TSPs	vm22775	569228

#### 3.2 実験の手法

都市数  $N$  の問題を解く。小規模な巡回路をつくる時、 $\frac{N}{10}$  都市や  $\frac{N}{100}$  都市の規模にする。 $\frac{N}{10}$  都市の小規模な巡回路を  $r$  回生成するとき、

市の小規模な巡回路をつくるために、まず、ランダムに 1 つの都市を選ぶ。その選んだ都市と、選んだ都市の近傍都市で  $\frac{N}{10}$  都市の巡回路をつくり、LK 法で解く。 $\frac{N}{10}$  都市の小規模な巡回路を  $r$  回生成し終わったら、各都市が現れるように、接続した都市をその回数が多かった順に並べる。その並べたものを利用して都市数  $N$  の問題を LK 法で解く。この接続した回数が多い都市順に並べたものをリストと呼ぶことにする。しかし、そのままでは、距離の近い都市が含まれていないまま解いた場合がある。距離の近い都市を加えずに解くと解が良くなる場合がある。これを防ぐために、近傍都市をリストに重複しないように加えたものを使って、もう 1 度 LK 法で解く。このとき、LK 法で解く際に必要な初期の巡回路は、リストを使った LK 法で解いた巡回路を使う。

2.3 節にあるように、それぞれの都市数によって小規模な巡回路の都市数、生成回数、小規模な巡回路の出現回数の平均を表 2 のようにした。

表 2: 実験の条件

問題	小規模な巡回路の都市数	小規模な巡回路の生成回数	小規模な巡回路となる都市の平均回数
d493	50	493	50
nrw1379	140	500	50
fl3795	380	500	50
rl5915	600	500	50
brd14051	300	1600	34
vm22775	500	1500	32

### 4 実験結果

以下は d493 の問題を都市数 50 の小規模な巡回路を 493 回生成したときの表である。

表 3: リストを使った LK 法の処理時間

実験回数	(1)リストのLK法の処理時間[s]	(2)リスト+近傍都市のLK法の処理時間[s]	(1)の処理時間+(2)の処理時間[s]
case-1	0.62	0.23	0.85
case-2	0.85	0.35	1.20
case-3	0.60	0.10	0.70
case-4	0.92	0.14	1.06
case-5	0.74	0.11	0.85
case-6	0.50	0.10	0.60
case-7	0.52	0.27	0.79
case-8	0.35	0.11	0.46
case-9	0.81	0.13	0.94
case-10	0.37	0.17	0.54
平均	0.63	0.17	0.80
合計	6.28	1.71	7.99(55%)

表 3 は、提案法の LK 法にかかった処理時間であり、単位は [s] である。(1) はリストを使った LK 法、(2) はリストに近傍都市を加えたものを使った LK 法の処理時間、その (1) と (2) の処理時間を足したものをあらわしている。合計の括弧内にあるパーセントの数字は、表 4 の平均の処理時間と

表 4: 近傍都市を使った LK 法の処理時間

実験回数	近傍都市の LK法[s]
case-1	1.99
case-2	1.15
case-3	1.31
case-4	1.28
case-5	1.39
case-6	1.12
case-7	1.28
case-8	1.55
case-9	1.60
case-10	1.67
平均	1.43
合計	14.34

比較したときの処理時間の結果である。ここで表 4 は、従来の LK 法にかかった処理時間である。

LK 法の処理時間を比較すると、リストの LK 法の処理時間は近傍都市を使った LK 法の処理時間よりも速いことがわかる。(1) リストのみの LK 法の処理時間に (2) リストに近傍都市を加えて解いた LK 法の処理時間を足したものと近傍都市を使った LK 法の処理時間を比較しても、リストを使った LK 法の処理時間の方が速くなっている。

表 5 は提案法の LK 法にかかった処理時間である。それぞれの項目の処理時間の単位は [s]、距離の単位は [km] である。近傍都市の生成は、従来法ではあらかじめ決められた近傍都市の数だけでよいが、提案法では、都市数  $n$  の小規模な巡回路の生成のために、近傍都市は  $n$  だけ必要となるので、従来法と比較すると処理時間がかかる。

表 6 は近傍都市を使って解いた場合、従来の LK 法の処理時間を単位 [s] である。

表 5、表 6 は次頁に表示している。

表 7 は、従来法と提案法の LK 法の処理時間の平均で単位は [s] である。近傍とリストの都市数はそれぞれ 20 としている。解いた 6 つの問題すべて近傍都市を使った LK 法の処理時間より、リストと近傍都市を使った LK 法の処理時間が速くなっていることがわかった。

近傍都市を使った LK 法の処理時間と、リストを使った LK 法の処理時間を比較すると、6 つの問題すべてリストを使った LK 法の処理時間が速くなっている。

#### 4.1 距離の比較

処理時間を比較してきたが、リストを使った LK 法で解き、解いた巡回路を初期の状態として、近傍都市を使った LK 法で解いているので解はよくなっているはずである。表 8 にそれぞれの問題の

表 7: LK 処理時間の比較

都市数	従来法 [s]	提案法 [s]
d493	1.43	0.8 (55%)
nrv1379	11.07	3.57 (32%)
fl3793	73.74	33.13 (44%)
rl5915	176.26	39.63 (22%)
brd14051	1833.25	1288.48 (70%)
vm22775	1272.80	1021.82 (80%)

解となる巡回路の距離を平均したものを表示している。最適解は National TSPs に表示されていたものを使っている。

表 8 は次項に表示している。

## 5 おわりに

LK 法の処理時間を短縮するために LK 法の都市の探索部分を工夫した。493 都市の問題から、22775 都市の問題まで、都市数の違う 6 つの問題を解き、従来の方法と比較評価した。

処理時間の比較実験の結果、提案した都市探索を使った LK 法は、従来法よりも LK 法の処理時間が短縮されることがわかった。巡回経路長を比較すると、提案法は従来法より短くなっている。

提案法は、LK 法の前に近傍都市の計算のほか、小規模な巡回路を繰り返す時間が必要である。また、提案法は小規模な巡回路をつくるため、あまり都市数が少ない問題には適さないが、都市数が多い問題を解く場合に適していると考えられる。

今後、GA と組み合わせ、大規模 TSP を解くのに利用する予定である。

表 5: 提案法の LK 法にかかった処理時間

実験回数	リスト生成のための処理時間[s]			LK法の処理時間[s]					距離[km]	
	近傍の生成	リストの生成開始	リストの生成	リストを使ったLKの開始	リストを使ったLKの終了	リスト+近傍都市のLKの開始	リスト+近傍都市のLKの終了	全体の終了	リストを使ったLK法	リスト+近傍都市を使ったLK法
case-1	0.14	10.69	10.69	10.71	11.33	11.33	11.56	11.56	36297	36230
case-2	0.13	10.65	10.65	10.65	11.50	11.51	11.86	11.86	35965	35916
case-3	0.13	10.90	10.90	10.90	11.50	11.50	11.60	11.60	35733	35733
case-4	0.14	10.96	10.96	10.96	11.88	11.88	12.02	12.02	35617	35598
case-5	0.13	10.80	10.80	10.81	11.55	11.55	11.66	11.66	35921	35921
case-6	0.14	10.59	10.59	10.59	11.09	11.09	11.19	11.19	36022	36022
case-7	0.14	10.63	10.63	10.64	11.16	11.17	11.44	11.44	35687	35665
case-8	0.11	6.94	6.94	6.94	7.29	7.29	7.40	7.40	35774	35692
case-9	0.12	10.56	10.56	10.56	11.37	11.38	11.51	11.51	35985	35978
case-10	0.10	7.16	7.17	7.17	7.54	7.54	7.71	7.72	36116	35993
平均	0.13	9.99	9.99	9.99	10.62	10.62	10.80	10.80	35912	35875
合計	1.28	99.88	99.89	99.93	106.21	106.24	107.95	107.96	359117	358748

表 6: 従来の LK 法にかかった処理時間

実験回数	単位[s]				単位[km]
	近傍の生成	LKの開始	LKの終了	全体の終了	巡回路の距離
case-1	0.07	0.07	2.06	2.07	35804
case-2	0.07	0.07	1.22	1.23	35594
case-3	0.08	0.08	1.39	1.40	36241
case-4	0.08	0.08	1.36	1.37	36061
case-5	0.08	0.08	1.47	1.48	35745
case-6	0.08	0.08	1.20	1.21	36098
case-7	0.08	0.08	1.36	1.37	35829
case-8	0.09	0.09	1.64	1.65	35878
case-9	0.13	0.13	1.73	1.74	35876
case-10	0.13	0.13	1.80	1.81	36230
平均	0.09	0.09	1.52	1.53	35935.6
合計	0.89	0.89	15.23	15.33	359356

表 8: 解いた問題の経路長

問題	リストを使ったLK法の巡回路の距離	リストと近傍を使ったLKの巡回路の距離	近傍を使ったLK法の巡回路の距離	最適解
d493	35911.7 (12.5%)	35874.8 (2.4%)	35935.6 (12.6%)	35002
nrv1379	58188.0 (12.7%)	58123.5 (2.6%)	58678.1 (13.6%)	56638
fl3793	32708.3 (13.6%)	31383.7 (9.0%)	33097.2 (15.3%)	28772
rl5915	591113.7 (14.5%)	588519.5 (4.0%)	589200.4 (14.1%)	565530
brd14051	494602.4 (15.3%)	492950.4 (5.0%)	494078.4 (15.2%)	469385
vm22775	636068.2 (11.7%)	617812.6 (8.5%)	633256.0 (11.2%)	569288

## 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, 株式会社 朝倉書店, 1997.
- [2] K. Helsgaun, “An effective implementation of the lin-kernighan traveling salesman heuristic”, *European Journal of Operations Research*, Vol. 126, No. 1, pp. 106–130, (2000).
- [3] S.Lin and B.W.Kernighan, “An effective heuristic algorithm for the traveling-salesman program”, *Operations Research*, Vol. 21, No. 2, pp. 498–516, (1973).
- [4] 黒田正文, 岩切淳一, 山森一人, 吉原郁夫, “摂動法を併用した TSP の準最適解生成法”, 電子情報通信学会 2008 総合大会講演論文集, D-1-2(CD-ROM) (2008).
- [5] 片山謙吾, 平林永行, 成久洋之, “遺伝的アルゴリズムの交叉法に対する性能評価”, 電子情報通信学会論文誌, Vol. J81-D-I, No. 6, pp. 639–650, (1998).
- [6] 永田裕一, 小林重信, “巡回セールスマン問題に対する交叉：枝組み立て交叉の提案と評価”, 人工知能学会誌, Vol. 14, No. 5, pp. 848–859, (1998).
- [7] 吉澤大樹, 橋本周司, “巡回セールスマン問題における地形構造の解析”, 人工知能学会誌, Vol. 16, No. 3, pp. 309–315, (2001).
- [8] 城戸隆, “GA を用いたハイブリッドアルゴリズムの実験”, 情報処理学会全国大会講演論文集, pp. 95–96, (1992).
- [9] 山村雅幸, 小野貴久, 小林重信, “形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法”, 人工知能学会誌, Vol. 7, No. 6, pp. 1049–1059, (1992).
- [10] “<http://www.iwr.uni-heidelberg.de/groups/comopt/software/tsplib95/>”.
- [11] “<http://www.tsp.gatech.edu/world>”.