

National TSP 向け PZ-crossover 法の性能評価

関 謙人^{1,a)} 山森 一人^{2,b)} 相川 勝^{3,c)} 吉原 郁夫⁴

概要：本稿では、ヒューリスティックと組み合わせたハイブリッド GA (HGA) において、初期の解の改善に優れる可変分割領域交叉法と、解の探索空間の広い部分領域交叉法を組み合わせた PZ-crossover 法を提案する。PZ-crossover 法では初期世代を可変分割領域交叉法で改善し、解の改善が進んだ後期世代で部分領域交叉法に切り替える。National TSP を用いて PZ-crossover 法の性能を評価する。

キーワード：巡回セールスマン問題, 遺伝的アルゴリズム, 組合せ最適化問題

Performance evaluation of PZ-cross for natinal TSP

KENTO SEKI^{1,a)} KUNIHITO YAMAMORI^{2,b)} MASARU AIKAWA^{3,c)} IKUO YOSHIWARA⁴

Abstract: In this paper, we propose a new crossover suitable for solving National TSPs which are benchmarks of Traveling Salesman Problem. We name the proposed crossover as PZ-crossover (PZ-cross) which combines Zoning crossover (Z-cross) and Partitioning crossover (P-cross). P-cross can improve solution quality at early generations. Z-cross can search a wide range solution space. We evaluate performance of PZ-cross by National TSPs from the view of tour quality and CPU time. We compare Z-cross, P-cross and PZ-cross by three benchmarks. As a result, PZ-cross succeeds to find a better solution than that of the Z-cross and P-cross in the all benchmarks. In addition, PZ-cross was about four times faster than Z-cross for all benchmarks.

Keywords: Traveling Salesman Problem, Genetic Algorithm, Combinatorial optimization

1. はじめに

巡回セールスマン問題 (TSP: Traveling Salesman Problem) は、あるセールスマンが与えられた都市を一度ずつ訪問して出発点に戻るとき、総移動距離が最少となる巡回経路 (以後、最適解と呼ぶ) を求める問題である [1]。TSP は 19 世紀以来、多くの研究者達によって研究され続け、様々

な解法が提案されてきた。

TSP の解法を大別すると厳密解法と近似解法の 2 種類がある。厳密解法は必ず最適解が得られる解法のことであり、TSP のもっとも簡単な厳密解法は、すべての解を列挙し最適解を見つける列挙法である。しかし、TSP の解候補は都市数が n のとき $\frac{(n-1)!}{2}$ 通り存在し、都市数が増加するにしたがって爆発的に解候補が増えるため、大規模な TSP を列挙法で解くことは事実上不可能である。TSP の有力な厳密解法として、分枝カット法 [2] がある。分枝カット法は分枝限定法 [3] に切除平面法 [4] を組み込むことで無駄な探索を省き、高速化を図る解法である。分枝カット法で最適解を求めた例として、スウェーデンの 24,978 都市を巡る問題 [5] があるが、これは 96 台の PC クラスタを使って約 8 年の計算時間を要した。

近似解法は厳密解法より少ない計算量で最適解、あるいは最適解に近い解を求める解法であり、大規模な TSP と

¹ 宮崎大学 工学研究科
Graduate Engineering, University of Miyazaki, Japan
² 宮崎大学 工学教育研究部
Faculty of Engineering, University of Miyazaki, Japan
³ 宮崎大学 工学部 教育研究支援技術センター
Technical Center, Faculty of Engineering, University of Miyazaki, Japan
⁴ 宮崎大学 名誉教授
Professor emeritus, University of Miyazaki, Japan
a) seki@taurus.cs.miyazaki-u.ac.jp
b) yamamori@taurus.cs.miyazaki-u.ac.jp
c) aikawa@taurus.cs.miyazaki-u.ac.jp

の相性が良い。近年では近似解法を使った大規模な TSP の解法の研究が盛んである [6–10]。TSP の近似解法として、強力なヒューリスティックである Lin-Kernighan 法 (LK 法) [11] があり、近年における大規模 TSP のベンチマークの最適解の多くは、LK 法を改良した Helsgaun's LK (LKH) [6] 法によって発見されている。他の有効な近似解法として、生物進化の過程を模した遺伝的アルゴリズム (GA:Genetic Algorithm) [12] がある。また、局所的探索能力に優れたヒューリスティックと、大域的探索に優れる GA を組み合わせたハイブリッド GA (HGA:Hybrid GA) も、大規模な TSP に対して有効であることが報告されている [8]。

大規模な TSP に対して有効な HGA における交叉法の一つに、黒田らの提案した部分領域交叉法 (Z-cross 法: Zoning-Crossover 法) [10] がある。Z-cross 法は親個体のある一部の領域を切り取り、その領域の中にある巡回経路を交換する交叉法である。黒田らは Z-cross 法と LK 法を組み合わせた HGA を用いて、TSP のベンチマークである VLSI TSP [13] の一つ、rbz43748 [14] について当時の世界記録を更新した。しかしながら、Z-cross 法は National TSP [15] のような、都市の分布に疎密がある問題には適していない。Z-cross 法の交叉領域は、問題の南北の長さや東西の長さを基準にランダムに作成されるため、National TSP のような問題では、Z-cross 法は都市を非常に多く含む、あるいはほとんど含まない交叉領域を作成する場合がある。このような交叉領域での交叉では、子個体は親個体から長い部分経路を受け継ぐことになる。LK 法を組み合わせた HGA では、親個体の部分経路は既に LK 法によって改善されているため、長い部分経路を子個体に受け継がせても改善が行われることは少ない。このため、都市の分布に疎密がある問題では、交叉領域に含まれる都市の数を適切に調整することが重要となる。そこで著者らは、Z-cross 法をベースとし、交叉領域の作成にカーブの分割アルゴリズム (KPA:Karp's Partitioning Algorithm) [16] を用いることで領域の中に含まれる都市の数を調整する可変分割領域交叉 (P-cross:Partitioning crossover) 法 [17] を提案した。P-cross 法は、改良した KPA を交叉に用いることで交叉領域に含まれる都市の数を調整することができ、Z-cross 法よりも速い実行時間と、高い初期解の改善率を示した。しかしながら、P-cross 法は交叉領域に含まれる都市の数を調整する関係上、改良した KPA を用いたとしても、作成できる交叉領域のパターンに限界があるという問題があった。

本研究の目的は、National TSP のような都市の分布に疎密がある問題に対して有効な交叉法を開発することである。本稿では、初期の世代では P-cross 法を用い、解の改善が進んだ世代で Z-cross 法に切り替える PZ-crossover (PZ-cross) 法を提案する。提案する PZ-cross 法を、LK 法

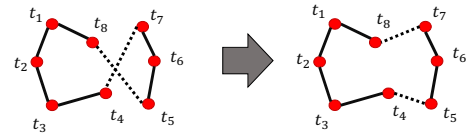


図 1 2-opt 法の例。

と組み合わせた HGA として用い、計算機実験によって得られる解の良さと実行時間を、Z-cross 法および P-cross 法と比較して、その有効性を示す。

本稿の構成は以下通りである。第 2 章では LK 法や Z-cross 法、KPA、P-cross 法などの TSP の近似解法を説明する。第 3 章では提案手法である PZ-cross 法を説明し、National TSP を用いて計算機実験を行って、提案手法と Z-cross 法、P-cross 法の性能を比較して評価する。第 4 章は本研究のまとめと今後の課題である。

2. TSP の近似解法

2.1 2-opt 法

2-opt 法は、図 1 に示すように任意の 2 本の枝を選んで繋ぎ変え、元の経路よりも短い経路となればその経路を採用する手法である。以下に手順を説明する。なお、 i, j を都市としたとき、 i, j を直接結んでいる経路を枝 (i, j) とし、枝 (i, j) の長さを $c(i, j)$ で表す。

STEP1 任意の枝 (t_{i_1}, t_{i_2}) と (t_{i_3}, t_{i_4}) を選択する。

STEP2 式 (1) を満たすとき、枝 (t_{i_1}, t_{i_2}) と (t_{i_3}, t_{i_4}) を切断し、枝 (t_{i_1}, t_{i_4}) と (t_{i_2}, t_{i_3}) を繋ぐ。

$$c(t_{i_1}, t_{i_2}) - c(t_{i_2}, t_{i_3}) + c(t_{i_3}, t_{i_4}) - c(t_{i_4}, t_{i_1}) > 0. (1)$$

STEP3 STEP1~2 をあらかじめ指定した回数繰り返す。

2.2 LK 法

2-opt 法では繋ぎかえる枝の数を固定していたが、LK 法では繋ぎかえる枝の数を固定せず、短い経路が存在する可能性がある限り繋ぎかえを続ける手法である。図 2 は LK 法による繋ぎかえの例である。

STEP1 任意の都市 t_1 を選ぶ。

STEP2 巡回経路上で都市 t_1 の次にある都市を t_2 とし、 $i = 2$ 、 $G = c(t_1, t_2)$ 、 $G^* = 0$ とする。

STEP3 式 (2) を満たす都市 t_{i+1} を探す。なければ **STEP7** へ。

$$G - c(t_i, t_{i+1}) > 0. (2)$$

STEP4 巡回経路上で t_{i+1} の前にある都市を t_{i+2} とし、 $G = G - c(t_i, t_{i+1}) + c(t_{i+1}, t_{i+2})$ とする。

STEP5 枝 (t_1, t_i) と枝 (t_{i+1}, t_{i+2}) を切断し、枝 (t_i, t_{i+1}) を繋ぐ。この時、式 (3) を満たすならば、 $G^* = G$ とし、この経路を T' とする。

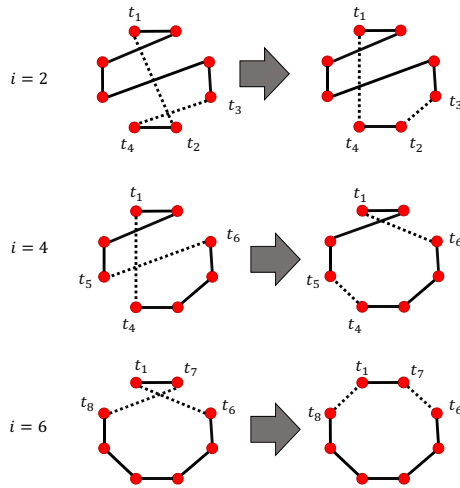


図 2 LK 法の例。

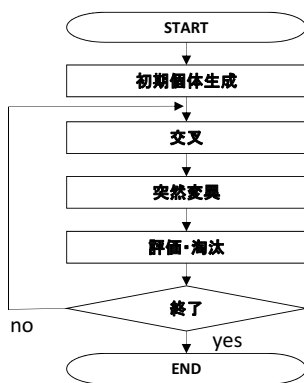


図 3 一般的なGAの流れ。

$$G - c(t_{i+1}, t_{i+2}) > G^*. \quad (3)$$

STEP6 $i = i + 2$ として、**STEP3**に戻る。

STEP7 $G^* > 0$ ならば $T = T'$ とし、巡回経路を更新する。

STEP8 **STEP1**~**7** を、あらかじめ指定した回数繰り返す。

2.3 遺伝的アルゴリズム

遺伝的アルゴリズム (GA: Genetic Algorithm) は、生物進化の原理に着想を得たアルゴリズムである。GA では、図 3 のように交叉、突然変異、評価・淘汰といった遺伝的操作を繰り返し行い、より良い個体を生成する。また、大域的な探索が得意な GA と、局所的な探索が得意なヒューリスティックを合わせたハイブリッド GA (HGA: Hybrid GA) は、大規模 TSP に対して有効である。

2.3.1 巡回経路の遺伝子表現

TSP に対して GA を用いるためには、巡回経路を遺伝子として表現する必要がある。本稿では、起点となる都市から巡回する順番に都市番号を列挙した数列を遺伝子とするパス表現 (path representation) を用いる。図 4 のように、 $a \sim m$ までの都市番号が振られた都市群を、 a から順番

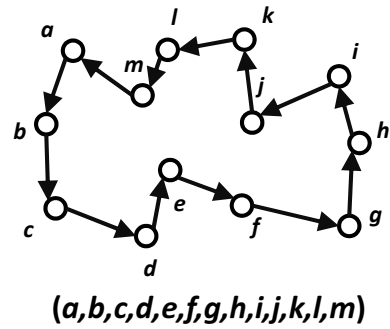


図 4 巡回経路のパス表現。

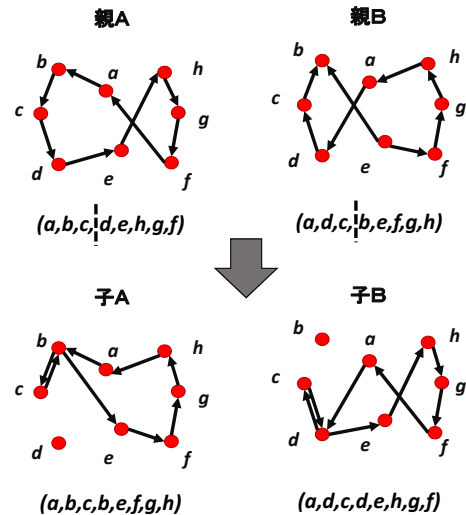


図 5 一点交叉法と致死性遺伝子を持つ個体の例。

に巡る巡回経路は $(a, b, c, d, e, f, g, h, i, j, k, l, m)$ と表す。

2.3.2 交叉

交叉とは、複数の遺伝子の一部を交換し、新たな個体を作り出す操作のことである。もっとも簡単な交叉法として 1 点交叉法がある。1 点交叉法では、親個体の遺伝子を切断し、切断した箇所から後ろの部分交換する。TSP を解く場合、1 点交叉法では生み出された個体が巡回経路とならない場合がある。例えば図 5 のように、2 つの親個体に対して、3 番目と 4 番目の都市の間を切断箇所として 1 点交叉を行った場合を考える。生成された子 A と子 B には、重複している都市と欠落している都市があるため、「すべての都市を 1 度ずつ訪問する」という TSP の制約を満たしておらず、解として不適切である。このような個体を「致死性遺伝子を持つ個体」という。TSP を GA で解くうえで、致死性遺伝子を持つ個体の生成を抑制する交叉法を用いることが重要である。

致死性遺伝子を持つ個体の生成を抑制する交叉法として、黒田らの提案した部分領域交叉法 (Z-cross 法: Zoning-Crossover 法) がある [10]。Z-cross 法とは、親個体の一部の領域を切り取り、その領域の中にある巡回経路を交換する交叉法である。Z-cross 法の手順について、以下で説明する。

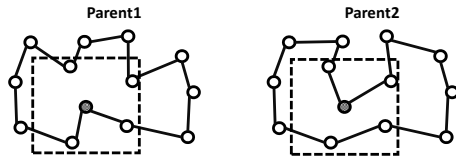


図 6 Z-cross 法手順 1 : 交叉領域の作成。

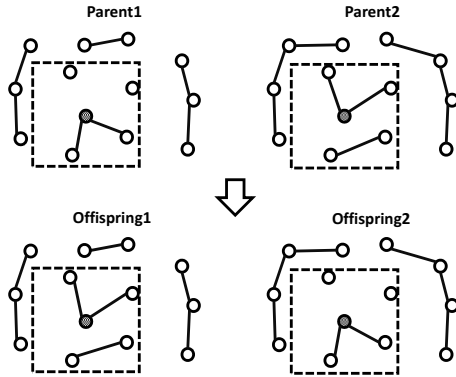


図 7 Z-cross 法手順 2 : 枝の切除と交叉。

STEP1 図 6 のように、任意の都市を一つ選択し、その都市を中心とした交叉領域を決定する。交叉領域の幅を L_x 、交叉領域の高さを L_y とすると、 L_x, L_y は式 (4) によって算出される。

$$\begin{aligned} L_x &= Xwidth \times \alpha, \\ L_y &= Ywidth \times \alpha. \end{aligned} \quad (4)$$

ここで、 $Xwidth$ は都市の最大 x 座標と最小 x 座標の差であり、 $Ywidth$ は都市の最大 y 座標と最小 y 座標の差である。 α は Z-cross 法のパラメータとして与え、黒田らは $0.1 \leq \alpha \leq 0.3$ の範囲で一様乱数により決定している。

STEP2 図 7 のように、交叉領域の中に含まれている経路を交換し、子個体にコピーする。このとき、交叉領域をまたぐ経路は切断する。

STEP3 図 8 のように、巡回経路になるように部分経路の端点同士を繋いで子個体を得る。このとき、なるべく巡回経路が短くなるように、近くにある部分経路の端点同士を繋ぐ。

Z-cross 法の交叉領域は、問題の南北の長さと東西の長さから乱数によって決定される。そのため、National TSP のような都市の配置に疎密があるような問題においては、図 9 のように含まれる都市が非常に多い、あるいは非常に少ない交叉領域を作成してしまう恐れがある。このような交叉領域は、子個体が一方の親個体の持つ経路を多く引き継ぐことを意味する。LK 法を組み合わせた HGA では、親個体の持つ経路は前の世代で LK 法による改善が行われているため、改善の余地が少ないことが多いことから、Z-cross 法は National TSP のような問題には適していない。そこ

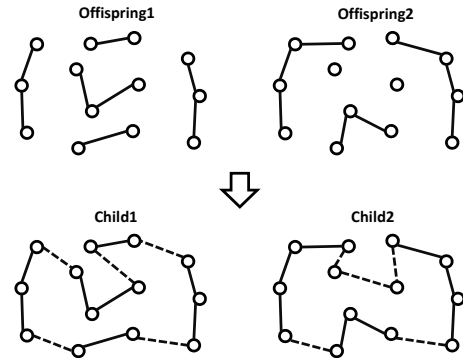


図 8 Z-cross 法手順 3 : 経路の再接続。

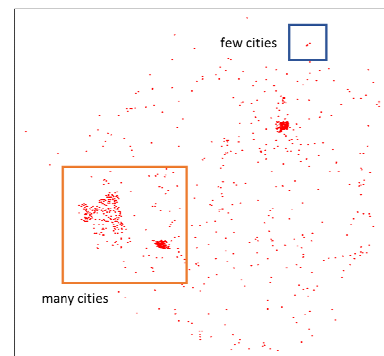


図 9 都市の疎密の著しい問題における交叉領域の例。

で、交叉領域の作成に、2.4 節で述べるカーブの分割アルゴリズム (KPA:Karp's Partitioning Algorithm) を用いることで、交叉領域内に含まれる都市の数を調整する P-cross 法が提案されている。P-cross 法の詳細は 2.5 節で述べる。

2.3.3 突然変異

突然変異は、遺伝子を一定の確率で変化させる操作のことである。遺伝子を実作無作為に変更することにより、交叉だけでは見つけることのできない経路を発見することを可能にする。HGA においては、ヒューリスティックを突然変異として扱うこともある。

2.3.4 評価・淘汰

個体の評価は巡回経路長に基づいて行い、巡回経路長が短いほど評価が高くなる。個体を評価した後、次の世代の親個体となる個体を選択する。もっとも単純な選択は、巡回経路長の短い順にソートし、上位の個体を次の世代の親個体として選択するランキング選択である。しかし、Z-cross 法は小領域の経路を交換する交叉法のため、親 A と子 A、親 B と子 B はそれぞれ似た巡回経路を持つ。そのため、単純なランキング選択では似た個体が増えてしまい、解の多様性を保つことができない。そこで、親 A と子 A、親 B と子 B の共存を禁止し、評価値の高いどちらか一方だけを次の世代の親個体を選ぶ。

2.4 カーブの分割アルゴリズム

KPA は問題の都市群を複数の領域に分割するためのア

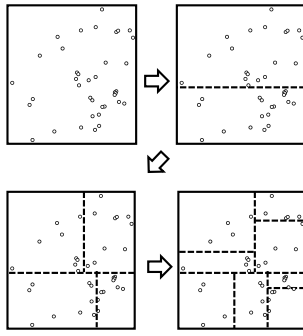


図 10 KPA による分割の例。

ルゴリズムである。この際、各領域に含まれる都市の数がほぼ等しくなるように分割する。

KPA の手順について説明する。図 10 は KPA による分割の例である。

STEP1 問題の領域を 2 つの領域に分割する。このとき、それぞれの領域に含まれる都市の数が等しくなるように分割する。

STEP2 分割されたそれぞれの領域を、**STEP1** と同じように 2 つの領域に分割する。

STEP3 領域の数があらかじめ決められた数に達するまで **STEP2** を繰り返す。

2.5 P-cross 法

P-cross 法では、Z-cross 法の交叉領域の作成に改良 KPA を用いることで交叉領域に含まれる都市の数を調整し、都市を過剰に含む、あるいはほとんど含まない交叉領域の作成を抑制する。

図 11 に P-cross 法の流れを示す。「初期個体生成」は初期解を生成するステップである。本稿では乱数によってランダムに個体を生成する。「領域分割」と「交叉」については、2.6 節および 2.7 節でそれぞれ詳しく説明する。「LK 法の適用」では交叉によって生成された個体に対して LK 法を適用する。「評価・淘汰」では、Z-cross 法と同様に親 A と子 A、親 B と子 B の共存を禁止し、評価値の高いどちらか一方だけを次の世代の親個体として選択する。「再分割」では問題領域の分割パターンを変更するかどうかを判断する。

2.6 改良 KPA

P-cross 法では KPA に対していくつかの変更を加えている。これは、元の KPA は各領域に含まれる都市の数が等しくなるように問題を分割するために、作成できる交叉領域のパターンが少ないためである。ヒューリスティックと組み合わせた HGA では、同じ交叉領域で交叉を続けた場合、早い段階で解が改善されつくしてしまい、解の改善が進まなくなる恐れがある。より良い解を発見するためには、交叉領域のパターンを増やす必要がある。

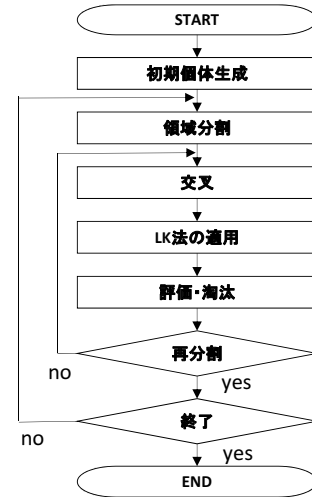


図 11 P-cross 法の流れ。

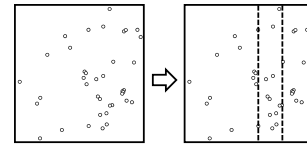


図 12 改良 KPA Step1：垂直方向の分割。

問題を $n \times n$ の領域に分割する場合を考える。P-cross 法で用いている改良 KPA の手順は以下の通りである。

STEP1 図 12 のように、問題の領域を $n - 1$ 本の垂直線で分割する。 i 番目の領域に含まれる都市の数 N_i は、問題の都市数を N とするとき、式 (5) によって定める。

$$N_i = \frac{N}{n} \times (1 + \beta). \quad (5)$$

ここで、 β は P-cross 法のパラメーターとして与え、本稿では $-0.1 \leq \beta \leq 0.1$ の範囲で一様乱数により決定している。

STEP2 図 13 のように、分割されたそれぞれの領域を $n - 1$ 本の水平線で分割する。

j 番目の領域に含まれる都市の数を N_{ij} で表す。ここで、 i は **STEP1** において作成された領域を表す番号である。このとき、各領域に含まれる都市の数は式 (6) によって定める。

$$N_{ij} = \frac{N_i}{n} \times (1 + \gamma). \quad (6)$$

ここで、 γ は P-cross 法のパラメーターとして与え、本稿では $-0.1 \leq \gamma \leq 0.1$ の範囲で一様乱数により決定している。

改良 KPA では β と γ により、 n の値が同じ場合でも異なる交叉領域を作成することができる。P-cross 法では、この改良 KPA によって作られた領域を交叉に用いる。

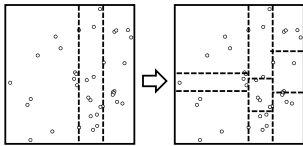


図 13 改良 KPA Step2: 水平方向の分割。

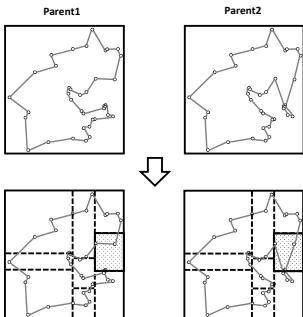


図 14 P-cross 法 Step1: 親個体と交叉領域の選択。

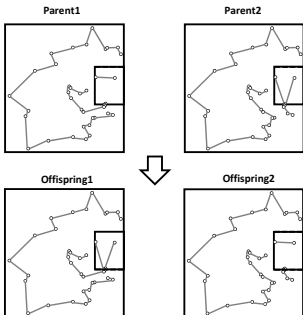


図 15 P-cross 法 Step2: エッジの切除と交叉領域の交叉。

2.7 交叉

P-cross 法での交叉手順について説明する。

STEP1 図 14 のように、2 つの親個体と、改良 KPA によって作られた領域から、交叉させる領域を 1 つ選択する。

STEP2 図 15 のように、交叉領域の中に含まれている経路を交換し、子個体にコピーする。このとき、交叉領域をまたぐ経路は切断する。

STEP3 図 16 のように、巡回経路になるように部分経路の端点同士を繋いで子個体を得る。このとき、なるべく巡回経路が短くなるように、近くにある部分経路の端点同士を繋ぐ。

3. 提案手法と評価

3.1 PZ-crossover 法

P-cross 法は、交叉領域に含まれる都市の数を調整する関係上、改良 KPA を用いたとしても、作成できる交叉領域のパターンに限界がある。そこで、初期の世代では P-cross 法を用い、解の改善が進んだ世代で Z-cross 法に切り替える PZ-crossover 法 (PZ-cross 法) を提案する。

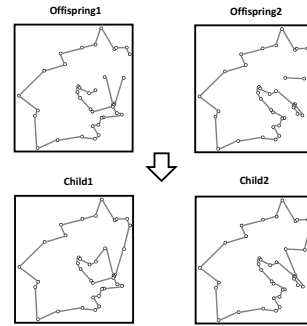


図 16 P-cross 法 Step3: 経路の再接続。

表 1 実行環境。

CPU	2.4GHz Xeon E5530
Core	4
Memory	16GB
OS	Linux 2.6
Language	C
Compiler	gcc ver.4.1.4 (-O2 optimize)

表 2 実験条件。

問題名	qa194	zi929	mu1979
都市数	194	929	1,979
1 世代あたりの個体数	50 個体		
最大世代数	10,000 世代		
再分割	100 世代ごと		
β	$-0.1 \leq \beta \leq 0.1$		
γ	$-0.1 \leq \gamma \leq 0.1$		
分割線の最小本数	2×2	2×2	3×3
分割線の最大本数	6×6	11×11	12×12
P-cross と Z-cross の切り替え	5,000 世代		

3.2 評価方法

提案手法の効果を調べるために、Z-cross 法、P-cross 法、PZ-cross 法を LK 法と組み合わせた HGA として用い、各手法の求解精度と実行時間を比較する。求解精度は式 (7) で定義する *Quality* で評価する。

$$Quality = \frac{L - L_{OPT}}{L_{OPT}} \times 100(\%). \quad (7)$$

ここで、 L は得られた解の巡回経路長であり、 L_{OPT} は最適解の、あるいは現在得られている準最適解の巡回経路長である。*Quality* の値が 0 に近いほど最適解に近い良い解であることを意味する。

実験に用いた計算機環境は表 1 の通りである。TSP のベンチマークには、National TSP [15] で公開されている 3 問を用いる。P-cross 法と PZ-cross 法のパラメータは表 2 の通りであり、これは事前に予備実験を行って求めた。P-cross 法の分割線の本数は、最小本数から 100 世代ごとに垂直、水平方向ともに 1 本ずつ増やしていく。分割線の本数が最大値に達した場合、次は最小本数に戻る。

実験はそれぞれの手法で各問題に対して 5 回ずつ行い、

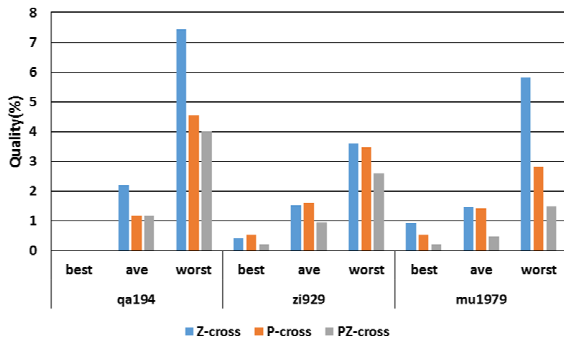


図 17 解の質の比較。

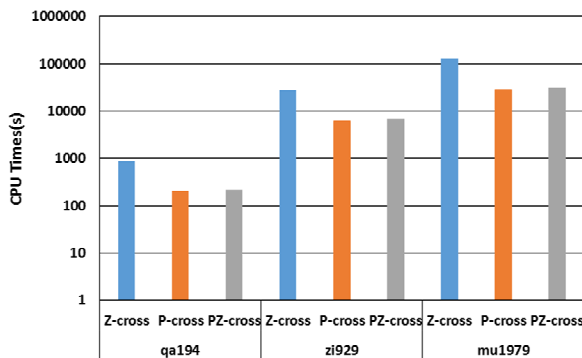


図 18 実行時間の比較。

その平均 *Quality*、最良 *Quality*、最悪 *Quality* の 3 つの *Quality* と、平均実行時間を計測する。

3.3 実験結果

実験によって得られた *Quality* を図 17 に示す。図 17 より、qa194 においては 3 つの手法すべてで最適解を求めることができたことがわかる。また、平均 *Quality* と最悪 *Quality* では、P-cross 法と PZ-cross 法は共に Z-cross 法よりも良い *Quality* を得ることができた。zi929 においては最良 *Quality*、平均 *Quality* 共に P-cross 法は Z-cross 法にわずかに劣っていたが、PZ-cross 法はすべての *Quality* において Z-cross 法よりも優っている。mu1979 について、すべての *Quality* において PZ-cross 法が最も良い解を得ることができ、P-cross 法も Z-cross 法より良い解を得ることができた。

実行時間を図 18 に示す。図 18 より、すべての問題において P-cross 法は Z-cross 法よりも約 4.5 倍速く、PZ-cross 法は P-cross 法よりわずかに遅いことがわかる。

3.4 考察

比較実験を通じ、P-cross 法は 2 つの問題で Z-cross 法よりも良い解を得ることができた。これは、初期の解の改善率の差による。図 19 は、mu1979 における各手法の平均 *Quality* の世代毎の推移である。図 19 を見ると、P-cross 法と PZ-cross 法は Z-cross 法に比べて初期の解の改善率が

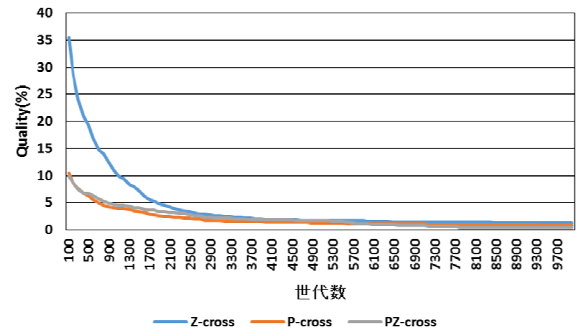


図 19 mu1979 における *Quality* の推移。

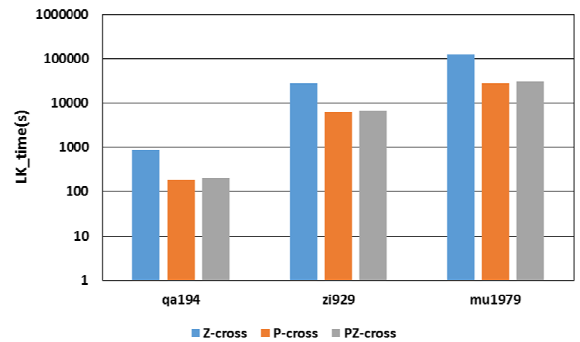


図 20 各問題における LK 法に費やした時間の比較。

高いことがわかる。特に最初の 100 世代で大きな差が出ているが、これはこの世代間での P-cross 法の交叉領域が大きく多くの都市を含み、かつ交叉領域の形状が 100 世代の間変化せず、LK 法による解の改善が効果的に行われるためと考えられる。多くの都市を含む交叉領域の交換は、経路の短い優れた個体から、多くの経路を受け取ることを可能にする。P-cross 法は、初期の交叉領域が多くの都市を含むことで、優れた個体の経路を個体群に行きわたらせ、高い初期の解の改善率を得たと考えられる。また、PZ-cross 法はすべての問題において P-cross 法や Z-cross 法よりも良い解を得ることができた。これは、早期に良い解を生成する P-cross 法と、解の探索空間の広い Z-cross 法を組み合わせることにより、P-cross 法や Z-cross 法単体よりも良い解を得ることができたと考えられる。

実行時間において、すべての問題で P-cross 法は Z-cross 法の約 4.5 倍の速さを示した。これは、LK 法に要した時間の差に因る。図 20 に、実行時間のうち LK 法に要した時間を示す。図 20 を見ると、P-cross 法が LK 法に要した時間は Z-cross 法の 25% 以下となっている。P-cross 法は初期の解の改善率が高く、早い世代で良い解を生成することができる。これらの解は LK 法による解の改善をさほど必要とせず、結果として LK 法にかかる時間が Z-cross 法よりも短くなる。また、PZ-cross 法が P-cross 法よりわずかに遅い原因は、PZ-cross 法の方が P-cross 法よりも解の探索空間が広く、LK 法による解の改善に要する時間が P-cross 法よりも長いためである。

4. 終わりに

TSP は 19 世紀以来多くの研究者達によって研究され続け、様々な解法が提案されてきた。近似解法は厳密解法より少ない計算量で最適解、あるいは最適解に近い解を求める解法であり、近年では近似解法を使った大規模な TSP の解法の研究が盛んである。

黒田らは Z-cross 法と LK 法を組み合わせた HGA を用いて、TSP のベンチマークである VLSI TSP の rbz43748 の問題の当時の世界記録を更新した。しかしながら、Z-cross 法は National TSP のような、都市の分布に疎密がある問題には適していない。これは Z-cross 法の交叉領域が、問題の南北の長さや東西の長さを基準にランダムに作成され、交叉領域に都市を非常に多く含んだり、あるいはほとんど含まない場合があるためである。それに対して著者らは、Z-cross 法をベースとし、改良した KPA を交叉領域の作成に用いた P-cross 法を提案した。P-cross 法は Z-cross 法よりも高い初期の解の改善率と、Z-cross 法よりも速い実行時間を示したが、都市の数を調整するために、作成できる交叉領域のパターンに限界があった。

本稿では、都市の分布に疎密がある National TSP のような問題に適した交叉法として、PZ-cross 法を提案した。PZ-cross 法は初期の解の改善率の高い P-cross 法と、解の探索空間の広い Z-cross 法を組み合わせた手法である。

計算機実験によって、得られる解の良さと実行時間の観点から比較したところ、PZ-cross 法はすべての問題において、P-cross 法や Z-cross 法単体よりも良い解を得ることができた。また、PZ-cross 法は P-cross 法よりもわずかに遅いものの、Z-cross 法より約 4.2 倍速いという結果となった。P-cross 法は Z-cross 法よりも初期に良い解を生成するため、LK 法による解の改善に要する時間が短いためであることが分かった。

今後の課題としては、より大規模な問題での評価が挙げられる。

参考文献

- [1] 山本芳嗣 and 久保幹夫:巡回セールスマン問題への招待, 朝倉書店 (1997).
- [2] Manfred, P. and Giovanni, R. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, Vol. 33, No. 1, pp.60–100, 1991.
- [3] Land, A. H. and Doig, A. G.: An automatic method of solving discrete programming problems. *Econometrica*, Vol. 28, pp.497–520, 1960.
- [4] Gomory, R. E.: Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, Vol. 64, pp.275–278, 1958.
- [5] Sweden.TSP, ”入手先” (<http://www.tsp.gatech.edusweden/index.html>)” (参照 2016-1-30).

- [6] Helsgaun, K.: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *EJOR*, Vol. 126, No. 1, pp.106–130, 2000.
- [7] Applegate, D. and Cook, W. and Rohe, A.: Chained Lin-Kernighan for Large Traveling Salesman Problems. *INFORMS Journal on Computing*, Vol. 15, No. 1, pp.82–92, 2003.
- [8] Nguyen, H. D. and Yoshihara, I. and Yamamori, K. and Yasunaga, M.: Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 37, No. 1, pp.92–99, 2007.
- [9] Dong, C. and Jger, G. and Richter, D. and Molitor, P.: Effective Tour Searching for TSP by Contraction of Pseudo Backbone Edges. *University Halle-Wittenberg Institute of Computer Science*, 2008.
- [10] Kuroda, M. and Yamamori, K. and Munetomo, M. and Yasunaga, M. and Yoshihara, I.: A proposal for Zoning Crossover of Hybrid Genetic Algorithms for large-scale traveling salesman problems. *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp.1–6, IEEE, 2010.
- [11] LIN, S. and Kernighan, B. W.: An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research*, Vol. 21, No. 2, pp.498–516, 1973.
- [12] 樋口哲也 and 北野宏明: 遺伝的アルゴリズムとその応用. 情報処理, Vol. 34, No. 7, 1993.
- [13] VLSI-TSP, ”入手先” (<http://www.math.uwaterloo.ca/tsp/vlsi/index.html>)” (参照 2016-1-30).
- [14] VLSI-TSP_rbz43748, ”入手先” (<http://www.math.uwaterloo.ca/tsp/vlsi/rbz43748.log.html>)” (参照 2016-1-30).
- [15] National.TSP, ”入手先” (<http://www.math.uwaterloo.ca/tsp/world/countries.html>)” (参照 2016-1-30).
- [16] Karp, R. M.: PROBABILISTIC ANALYSIS OF PARTITIONING ALGORITHMS FOR THE TRAVELING-SALESMAN PROBLEM IN THE PLANE. *Mathematics of Operations Research*, Vol. 2, No. 3, pp.209–224, 1977.
- [17] Kento, S. and Yamamori, K. and Aikawa, M. and Yoshihara, I.: Performance evaluation of partitioning crossover for national TSPs. *THE TWENTY-FIRST INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS*, pp.311–316, 2016.