免疫アルゴリズムの記憶機構を利用した nTSP の解法に関する考察

當間 愛晃* 遠藤 聡志** 山田 孝治**

Immune algorithms for nTSP

Naruaki TOMA* Satoshi ENDO** Koji YAMADA**

Abstract

In the AI field, adaptive algorithms such as neural networks or genetic algorithms are recognaized as the powerful learning systems or effective search methods. However, about immune algorithms modeled as a human immunity, it's fundamental ability and engineering applicablity is not cleared yet. In this paper, we apply the immune algorithm to the n-th traveling salesman problem (n-TSP) and discuss the effectivity of this algorithm.

Key Words: Immune Algorithm, n-TSP, Genetic Algorithm.

1. はじめに

自律系システム、適応系システム、調和系システムなどのインテリジェントシステムの設計構成において、ニューラルネットワーク、遺伝的アルゴリズムなどに代表される適応アルゴリズムの有効性がさまざまな研究成果により指摘されている。一方、生体系に内在するインテリジェントシステムである免疫系についての工学的応用に関する可能性及びその有効性は十分に議論されているとは言えない。

本稿では, 免疫システムの抗原の認識機構、抗体の記憶機構、抗原の排除機構に着目し, その工学問題に対する有効性を検討するために, マルチエージェントシステムに対する最適化問題の一例である分業巡回セールスマン問題 (n-Traveling Salesman Problem; 以後, nTSP) へ適用し、アルゴリズムの性能、特徴について検討する。また，だまし問題を通して記憶機構の利用に関する考察を行なう．

2. 分業巡回セールスマン問題:nTSP

2.1 TSP と nTSP

巡回セールスマン問題 TSP は NP-hard に分類される問題の複雑さとその応用範囲の広さからさまざまな解法による検討がなされてきた。しかし今日の大規模複雑な工学的諸問題への対応には各エージェント間の協力による効率良く問題を解決を行なうマルチエージェントシステムなどのより強力な問題解決法が求められている。そこで TSP におけるエージェント数を n 人に拡張した nTSP 問題 [4] を設定し、マルチエージェント型アルゴリズムの性能評価の基準問題として位置付ける (図 1)。

nTSP は本質的に各セールスマンに対する都市配分とその最短経路探索という 2 面性を持つ。これらの 2 面性は相互に関連しているため個別に逐次的に解くことが困難であるという特徴を持つ。 免疫システムでは動的な環境に対する適応能力が期待されるためこの種の問題に対し有効な問題解決手法であると考えられる．

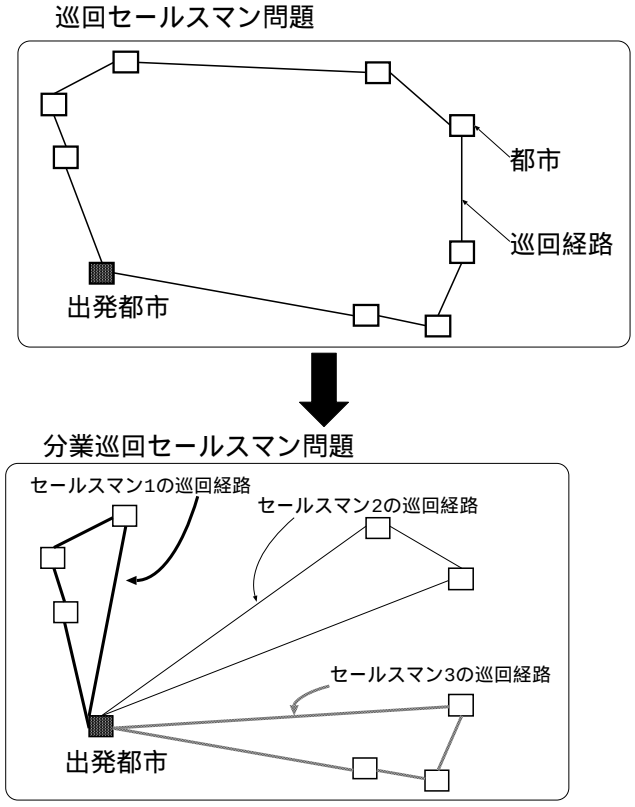

Fig. 1. nTSP の概念図

2.2 nTSP の定式化

Table1 に本稿で使用する記号を示す．

nTSP において目的は各セールスマンのパス (都市を巡

受理：1997 年 12 月 1 日
*工学部情報工学科
(Undergraduate Student, Information Engineering, Fac. of Eng.)
**工学部情報工学科
(Dept. of Information Engineering, Fac. of Eng.)

Table 1. 記号の説明

n	セールスマン数
m	都市数
d_{ij}	都市 i から都市 jへの距離
$(x_i,\ y_i)$	各都市位置の座標
S_i	セールスマン i
P_{S_i}	セールスマン i のパス
$Plan_k$	1 巡回計画

回する順序) の合計 (コスト) を最小化するようなプランを作成することである。付随する制約条件は

- 各セールスマンのパスは出発都市 (=最終都市) を同一とする.
- 出発都市を除く全ての都市は一人のセールスマンにのみ一度だけ訪問される。

[目的関数]

$$\mathbf{min} \quad \left(\mathbf{cost_k} = \sum_{\mathbf{i}} \mathbf{f_d}(\mathbf{P_{S_i}}) \right) \tag{1}$$

式 (1) で, P_{S_i}はセールスマン S_iのパスで, 出発都市と最終都市を除いたパスである. f_dは S_iの距離の総和を求める関数である.

[制約条件]

$$\mathbf{P_{S_i}} \cap \mathbf{P_{S_j}} = \emptyset,\ \mathbf{i} \neq \mathbf{j}\ (\forall \mathbf{i},\mathbf{j}) \tag{2}$$

式 (2) はセールスマン i と jのパス上に都市の重複がないことを意味する制約条件である.

$$Plan_k = \begin{bmatrix} t_s & P_{S_1} & t_s \\ t_s & P_{S_2} & t_s \\ \vdots & \vdots & \vdots \\ t_s & P_{S_n} & t_s \end{bmatrix} \tag{3}$$

式 (3) の各行は各々のセールスマンのパスを表しているプランである. P_{S_1}はセールスマン S_1のパスを意味する.

3. 免疫システム

免疫システムとは, 生体内に侵入する多種多様な未知の抗原に対応するために, 細胞遺伝子の再構築を行なって抗原に対応する抗体を産生し, 抗原を排除する生体監視防衛機構である.

工学的応用の一例に森らによってモデル化された免疫アルゴリズム [1,2] がある。本稿ではそのアルゴリズムの基本構成に従う。

Step 1 抗原の認識
抗原を入力情報として認識する.

Step 2 初期抗体群の生成
記憶細胞から過去に有効であった抗体群を生成する.

Step 3 親和度の計算
抗原と抗体 vの親和度 ax_vと抗体 vと抗体 wの親和度 $ay_{v,w}$を計算する.

Step 4 記憶細胞とサプレッサー細胞への分化
全ての抗体の濃度を計算し, 抗体 vの濃度 c_vが閾値 Tc

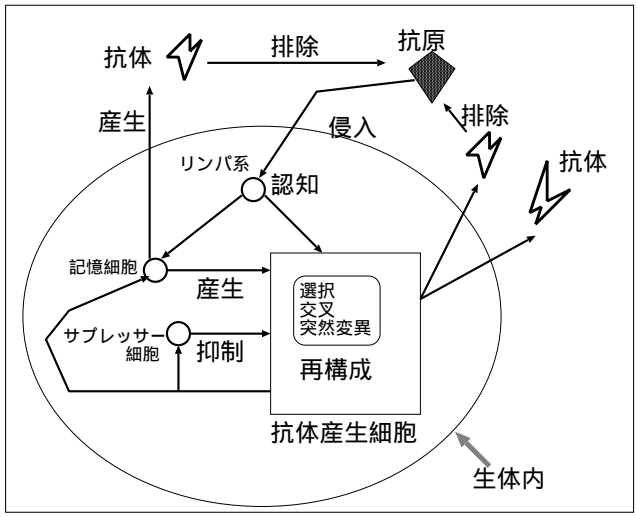

Fig. 2. 免疫システムの概念図

を越えた場合に, 抗体 vを記憶細胞 m に分化させる. また, 新しく分化した記憶細胞と同じ遺伝子を持つサプレッサー細胞 s を分化させ, サプレッサー細胞との親和度が, 類似度の閾値以上の抗体を消滅させる.

Step 5 抗体産生の促進と抑制
次世代に残る抗体 vの期待値 e_vを計算し, 現世代の抗体の親和度の低いもののいくつかの抗体を消滅させる.

Step 6 抗体の産生
4 で消滅させた抗体に変わる新しい抗体を, 乱数を用いてその遺伝子をランダムに決定することによって産生する. 以下, 世代があらかじめ設定した最終世代に達するまで 3 ~6 を繰り返す.

免疫アルゴリズムはその基本構成を (1) 遺伝的アルゴリズムに基づく解の再構成 (2) 記憶細胞による有効解の記憶 (3) サプレッサー細胞による記憶解の再探索の抑制とする。この構成により GA の持つ強力な広域な探索能力に加え、有効解の再利用及び再探索の抑制による探索効率の上昇が期待される。以下に免疫アルゴリズムの処理手順と nTSP との関係を示す (図 3 , 図 4)。

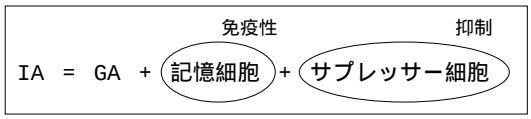

Fig. 3. 免疫アルゴリズムの枠組み

4. nTSP に対する IA の設計

4.1 コーディング

nTSP に IA を適用するためには, 問題の抗体を文字列表現しなければならない. 図 5は, 都市数 $m = 10$, セールスマン数 $n = 3$ に置ける 1 プランの概念図を表している.

1 プランの情報を染色体 (抗体) に記述する必要がある. GA による TSP の一解法として Grefenstette による順序表現 [5] がある。nTSP では抗体情報を n セールスマンに分割しなければならないためそのコーディング法に分割情

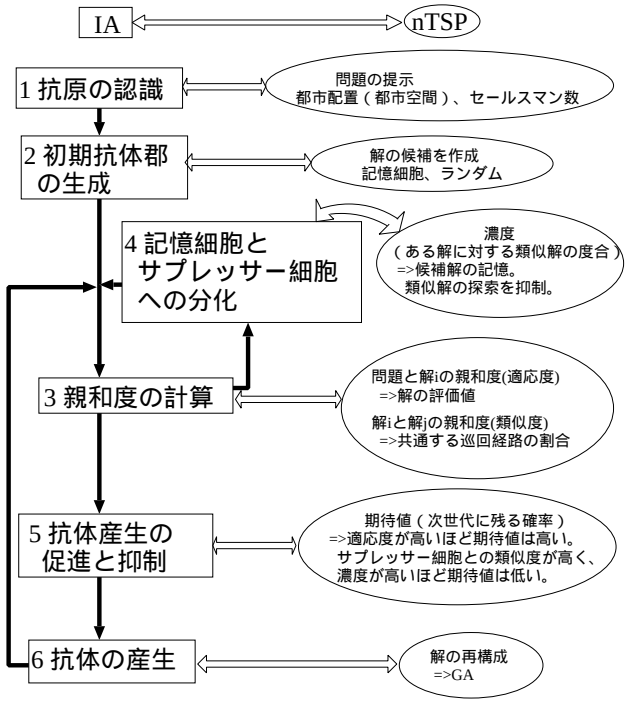

Fig. 4. 免疫アルゴリズムからみた nTSP

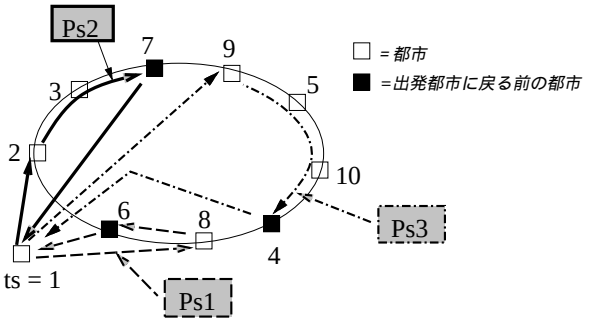

Fig. 5. nTSP の解のイメージ

報を加え拡張した。染色体に分割情報であるセパレータを加えたコーディング例が図 6 である。 図 6では、(1) 各セールスマンのパス (P_{s_n}) と n セールスマンに分けるためのセパレータの位置を文字列表現し、(2) 各セールスマンのパス部分 (巡回経路部) を順序表現することでコーディングを施した例である。

4.2 各評価尺度の計算

免疫アルゴリズムにおいては、適応度、類似度、濃度、期待値の 4 つの計算尺度を求める必要がある。以下に nTSP における評価尺度の計算式を定義する。

適応度

$$fitness_i = max_path - (work + penalty) \quad (4)$$

$max_path = fitness_i$が負にならないような定数

$work = \sum f_d(P_{S_i})$

$penalty =$ 平等に分業するためのペナルティ

類似度

$ay_{v,w} =$ (2 つの抗体に共通する順路の都市数)$/m$(5)

濃度

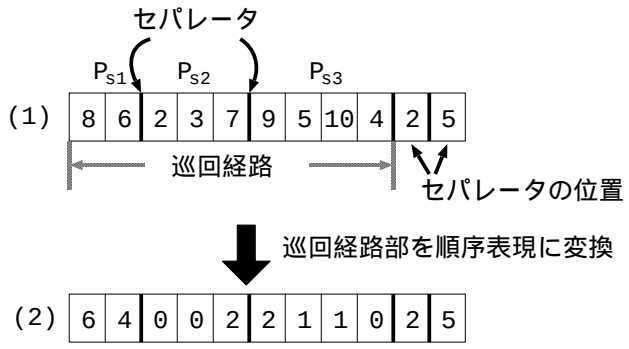

Fig. 6. コーディング例

$c_v =$ (類似度が閾値 $Tac1$ を越えた数)$/pop_size$ (6)

期待値

$$e_v = fitness_i \times (1 - ay_{v,s}) \times (1 - c_v) \quad (7)$$

$ay_{v,s} =$ 抗体 vとサプレッサー細胞との類似度

ここで，penalty は一プランにかかるコスト (work) の平均と各セールスマンの巡回経路コストの差の総和を計算している．また本稿では，ここで挙げた尺度を nTSP と関連させて，適応度を解の評価値，類似度を 2 つの解に共通する巡回経路の割合，濃度をある解が集団を支配する割合，期待値を次世代に残る確率として扱うことにする．

5. 実験

5.1 実験 1

実験 1 では，nTSP に対し IA が有効に機能するかどうかを確認する．設定する問題は都市数 12，セールスマン数 4 とし，都市配置は出発都市を中心に円周上に配置した．IA の各パラメータは Table2 のように設定した．ここ

Table 2. IA のパラメータ

集団数	50
初期集団	ランダム
世代数	10000
交叉確率 1	0.2
交叉確率 2	0.3
突然変異確率 1	0.02
突然変異確率 2	0.1
記憶細胞の総数	100
サプレッサー細胞の総数	5
濃度の閾値	0.43

で，交叉確率 1 及び突然変異確率 1 は巡回経路部における確率であり，交叉確率 2 及び突然変異確率 2 はセパレータ部における確率を意味する．本稿で取り扱うコーディングでは分業に関する情報がセパレータ部のみにあるため，巡回経路部よりもセパレータ部の操作確率を高く設定した．

遺伝子の再構成に使用する GA のオペレータは以下のように設定した．

- 交叉操作: 一点交叉
- 突然変異操作:
  - 巡回経路部: 遺伝子座$_i = random(m - i - 1)$

Table 3. パラメータ

パラメータ	GA	IA
初期集団	一定	記憶細胞の再利用
集団数	50	50
交叉確率	0.2	0.2
突然変異確率	0.02	0.02
記憶細胞の総数	nothing	50
サプレッサー細胞の総数	nothing	5
濃度の閾値	nothing	0.35

– セパレータ部: 遺伝子座$_j = random(m-1)$

図 7は初期集団内のプランの一例であり，図 8は IA によって得られた最適解である． これらの図から，IA が

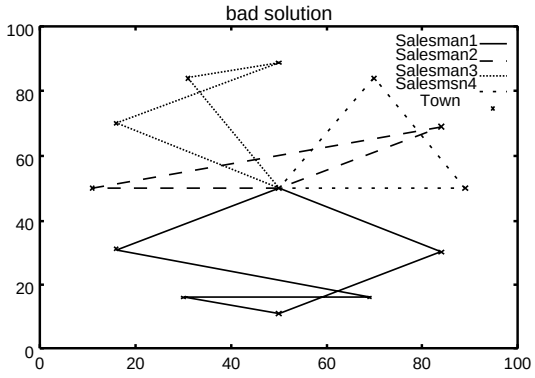

Fig. 7. 初期集団内の最良解

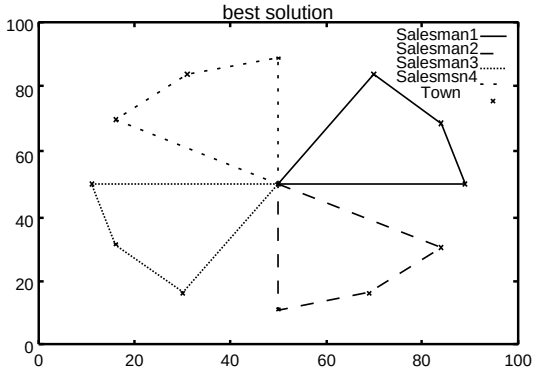

Fig. 8. IA により得られた最適解

適切な nTSP に対するプランを作成可能であると考えられる．

5.2 実験 2

実験 2 では，IA の探索効率を評価するために simpleGA との比較を行なった．比較に用いた問題はセールスマン数 2，都市数 10 とし，ランダムな都市配置を 30 題用意した．また，IA と GA の各パラメータは，Table3 のように設定した．

実験結果を図 9に示す． 図 9は，横軸に問題番号，縦軸に世代数をとり，box が GA，line が IA を示している．グラフから，30 題に対する準最適解探索までの世代数比

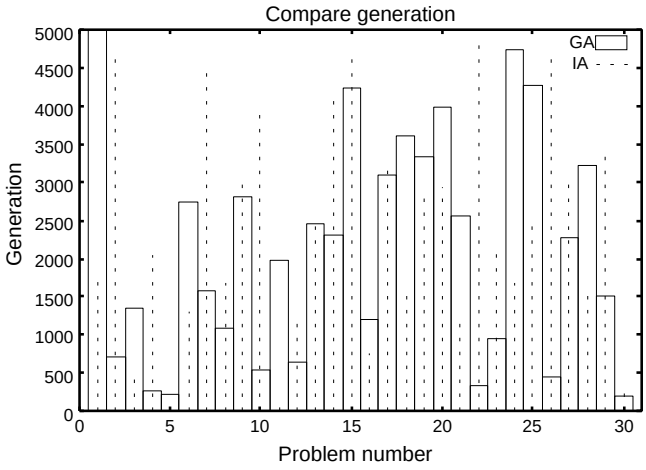

Fig. 9. GA と IA の効率比較

較では，GA:IA=13:17 で IA が優位であることが分かる．また，問題 11 では，IA が GA と比較して極めて早い世代で準最適解を獲得している．これは IA が過去に探索した候補解を記憶し，利用することで，類似性の高い問題に対して，効果的に準最適解を求めることができることによると考えられる．このことは，生体系における実際の免疫システムが一度記憶した抗原に対し適切に反応し，自己防衛を行なう行動様式に一致する．

6. 類似問題

6.1 記憶の利用

探索中の候補解の記憶は，式 (6) で計算した濃度がある閾値を越えた時に行なう (図 10)． 本稿では閾値を越え

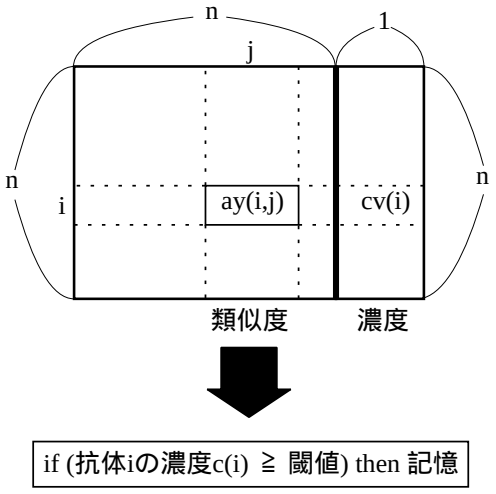

Fig. 10. 記憶方法

た解そのものを記憶細胞とサプレッサー細胞に記憶している．

記憶細胞

記憶細胞の総数 m に達したら，今保存しようとしている解と最も類似度が高い記憶細胞と交換する．

サプレッサー細胞

サプレッサー細胞の総数 s に達したら，最も古いサプ

レッサー細胞と解を交換する．

このように記憶された結果，記憶細胞では探索中における局所解を保存し，サプレッサーでは常に記憶細胞に記憶されたばかりの解を保存するはずである．このため，局所解を最適解に持つような類似問題に対しては記憶細胞の再利用により探索を用意にすると考えられる．また，局所解を記憶したなら局所解に類似する解を再探索することは無駄であると考えられるため，サプレッサー細胞に記憶された解に対する類似解の探索を抑制する．しかし，記憶された解が局所解になっているかどうかは分かり得ないため，サプレッサー細胞の総数を少なくすることにより後々には再探索する可能性があるように実装を行なった．

現在の仕様では，記憶細胞の適応度を新しい問題に対して計算し，適応度の高い記憶細胞で初期集団の半数程度を構成し，不足分をランダムに作成することで記憶細胞の再利用を行なっている．記憶方法及び，記憶の再利用方法にはまだ検討の余地が残っているが，本稿では類似問題について考察することにする．

6.2 だまし問題

だまし問題は 2 重の同心円上に同数の都市が均等に並んだ都市配置を扱い，その内外円の半径比によって 2 通りの大きく異なる最適解，局所解を持つ．図 11はどちらか一方の円を巡回してからもう片方を巡回するプラン (c-type) が最適解となる例で，図 12は外円と内円を交互に巡回するプラン (o-type) が最適解となる例である．

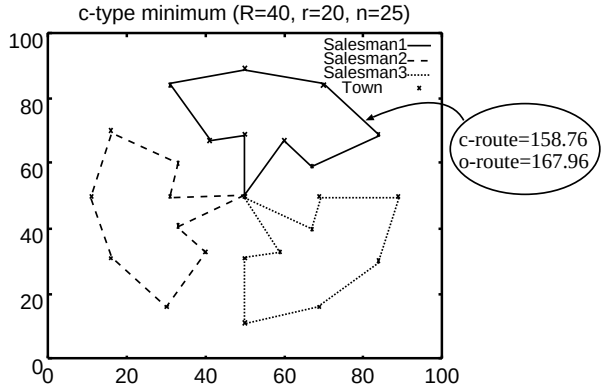

Fig. 11. c-type

このように大きく異なる最適解，局所解が存在するだまし問題においては IA の記憶細胞を再利用することにより探索が容易になると考えられる．

7. 実験 3

7.1 問題設定

実験 3 では，類似問題としてだまし問題を設定し，記憶の再利用により探索が容易になることを確かめる．確認方法は以下のように行なう．

Step1

IA の記憶細胞を構成させるため，記憶細胞なしで c-type が最適になる問題を解く．

Step2

Step1 で得られた記憶細胞を再利用し，異なる順路 o-type が最適解となる問題を解く．

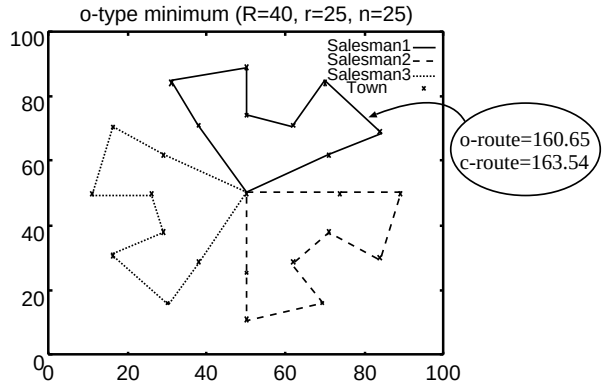

Fig. 12. o-type

Step3

Step1 で得られた記憶細胞を再利用し，同じ順路 c-type が最適解となる (Step1 で解いた問題とは異なる) 問題を解く．

Step1-3 の探索の様子を比較する．この時，Step1 よりも Step2 ，3 の方が探索効率が上昇しているならば探索が容易になっていると考えられる．以下では Step1 を C_{empty}(記憶なしで c-type を解く)，Step2 を O_c(c-type の記憶を利用して o-type を解く)，Step3 を C_c(c-type の記憶を利用して c-type を解く) と表記する．

各 Step で探索する問題は Table4 のように設定した．外

Table 4. 問題設定

パラメータ	C_{empty}	O_c	C_c
都市数	13	13	13
セールスマン数	1	1	1
外円の半径	40	40	40
内円の半径	10	35	15
最適解	c-type	o-type	c-type

円と内円上には 6 都市ずつ，出発都市を円の中心に配置した．

IA のパラメータは Table5 のように設定した．ここで，

Table 5. IA のパラメータ 2

初期集団	記憶細胞の再利用
集団数	100
交叉確率 1	0.2
突然変異確率 1	0.02
記憶細胞の総数	100
サプレッサー細胞の総数	5
類似度の閾値	0.315
濃度の閾値 1	0.315
濃度の閾値 2	0.315

類似度の閾値は類似度の計算，濃度の閾値 1 は濃度の計算，

濃度の閾値 2 は記憶する際の閾値にそれぞれ使用する．

7.2 実験結果

実験 3 の結果を図 13，図 14に示す．

図 13はそれぞれ，C_{empty}(同図上)，O_c(同図左下)，C_c(同図右下) の探索で得られた最良解を図示している．

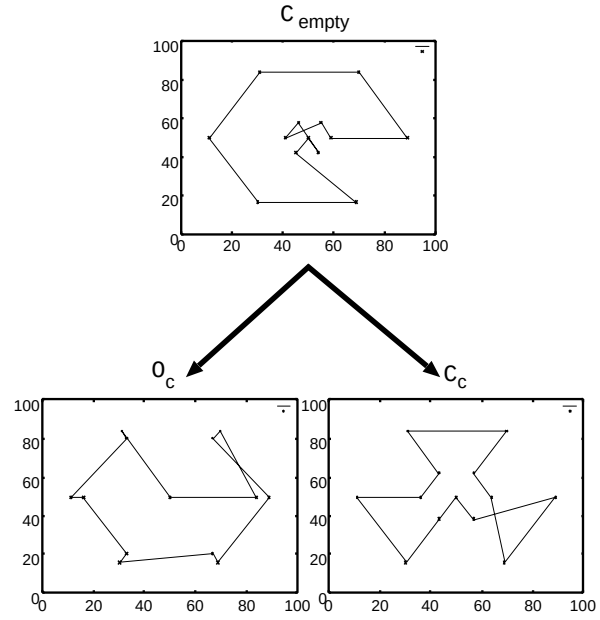

Fig. 13. IA の探索結果

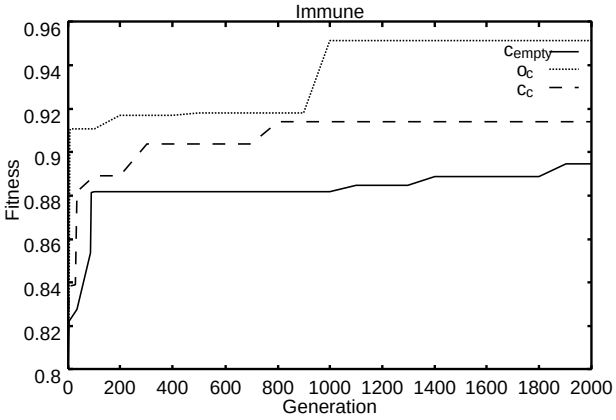

Fig. 14. 探索効率の比較

図 14は実験 3 で扱った 3 問題の探索効率を比較している．縦軸は最適解を 1 としてスケーリングした適応度，横軸は世代数を表している．同図より，100 世代までの適応度の上昇率を比較すると

$$C_{empty} < C_c < O_c$$

の順に良くなっていることが分かる．今回の実験では最適解を探索できていないために断言することは出来ないが，類似問題の探索において過去の記憶を利用することで容易にできると考えられる．

8. 問題点と解決策

8.1 濃度と順序表現

現在の実装では，大きな問題空間に対しては探索がうまくいかなかったため，実験 3 のように問題空間を狭めて実験を行なった．IA にはある解が集団を支配する割合 (濃度) が高くなるとその解を候補解として記憶するが，濃度が高くなるためには前世代の形質を保存する必要がある．しかしながら，本稿で用いた順序表現によるコーディングは，交叉操作における致死遺伝子作成の問題を解決することを主眼に作られた手法であるため，形質が保存されにくい．濃度と順序表現という目的の相反する組合せを取ってしまったため前述した問題点が出てきたと考えられる．

8.2 濃度とパス表現

濃度は表現型における類似度から計算される．そこで表現型を直接コーディングしたパス表現を用いることにする．図 5をパス表現でコーディングした例が図 15である．

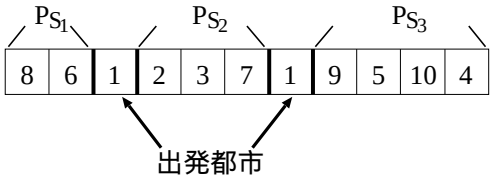

Fig. 15. パス表現によるコーディング例

TSP においては，パス表現された遺伝子に通常の交叉操作を行なうと致死遺伝子を作成しやすくなるため特別な交叉操作を使用することになる．そこで形質の保全にマッチしたサブツアー交換交叉 [6] を採用する方向で考えている．

9. おわりに

適応アルゴリズムの一手法である免疫アルゴリズムを nTSP へ適用し，その工学的応用に対する可能性について検討し，計算機実験によりその有効性について示した．また，類似問題としてだまし問題を取り上げ，探索を容易にできる可能性があることを示した．特に，記憶機構を活用することで，動的環境における類似問題の繰り返し解決を必要とする課題に対して有効に機能すると考えられる．複数の問題解決器を基本とするマルチエージェント系の有効なアルゴリズムとしての IA の利用が期待される．

文 献

[1] 森 一之, 築山 誠, 福田 豊生, 多様性を持つ免疫的アルゴリズムの提案と負荷割り当て問題への応用, T.IEE Japan, Vol.133-C, No.10, 1993.

[2] 森 一之, 築山 誠, 福田 豊生, 免疫アルゴリズムによる多峰性関数最適化, T.IEE Japan, Vol.117-C, No.5, 1997.

[3] 和田 健之介, 和田 佳子, 山登り飛び虫の進化と免疫システム論について, 数理科学, NO.353, NOVEMBER, 1992

[4] 中村 友洋, 角田 達彦, 田中 英彦, 分業セールスマン問題のニューラルネットワークによる解法, 人工知能 94-2, (1994, 6, 20)

[5] J.Grefenstette, R.Gopal, B.J.Rosmaita and D.Van Gucht : Genetic Algorithms for the Traveling Salesman Problem, Proc. of ICGA '85, 16/168 (1985)

[6] 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, Nov. 1992