GA によるヒューリスティック探索の最適化

仙石浩明 *　　吉原郁夫
(日立製作所)　　(東北大学)

GA based Optimization of Heuristic Search

Hiroaki SENGOKU　　Ikuo YOSHIHARA

Abstract

We present a generic method adding GA (Genetic Algorithm) to a given heuristic search algorithm. In the method, GA is used to optimize the priority of branches in the search tree. Heuristic search algorithms are easy to develop and have been widely used to solve actual problems, which have been conventionally solved manually, but for practical applications, we have to add more detailed knowledge. The proposed method does not require so much knowledge due to using GA.

キーワード: 遺伝的アルゴリズム, ヒューリスティック探索, 最適化, 巡回セールスマン問題
(genetic algorithm, ga, heuristic search, optimization, traveling salesman problem, tsp)

1　はじめに

遺伝的アルゴリズム (GA) は、探索に関する知識が不要なため作成が容易、問題の諸条件が変化しても柔軟に適応できる、探索が大域的である、などの長所を持つが、遺伝子コーディング法、交差方法などを問題毎に考案しなければばならない。

一方、従来から広く用いられてきたヒューリスティック探索法は、人手で解かれていたような問題を解く場合は、容易にアルゴリズムを作ることができる。しかし実問題に適用するには、より詳細な知識を組み込むなどのチューンアップが必須となる。

そこで、我々はヒューリスティック探索に GA を組み込む汎用的手法 [3] を提案した。与えられたヒューリスティック探索を、本手法で定義する標準形に変換できれば、容易に GA を組み込むことができる。

ヒューリスティック探索の部分は問題個別に作成する必要があるが、従来人手で解かれていたような問題に関しては、人手でどのように解いていたかを整理して、ルールあるいは手続きの形で表現することができれば、比較的容易に探索プログラムを開発することができる。そして本手法に基づき GA を組み込むことによって、少ない知識で質の高い解を得ることが可能になる。

ところがヒューリスティック探索の部分がすでに開発済みである場合、これを本手法で定義する標準形へ変換することが難しい場合がある。そこで今回、標準形の要件を緩和した手法を提案する。標準形への変換が容易になる反面、 GA を組み込んだ時の最適化の効率は低下する。

以下、 2 章では、標準形ヒューリスティック探索の定義を行う。そして標準形ヒューリスティック探索に GA を組み込む手法を 3 章で提案する。 4 章で巡回セールスマン問題 (TSP) を例にとり提案法を他手法と比較する。

2　ヒューリスティック探索

定義 1 M 個の互いに異なるアルファベットを $s_1, s_2, \ldots, s_M$ とする。このアルファベットの集合を Σ とおく。

アルファベットの並びを $\sigma = s_{i_1} s_{i_2} \cdots s_{i_m}$ $(m \leq M)$ とし、空列を λ とする。λ および、すべての σ からなる集合を Σ^* とする。 □

定義 2 (半順序) $\sigma = s_{i_1} s_{i_2} \ldots s_{i_m}$, $\sigma' = s_{j_1} s_{j_2} \ldots s_{j_n}$

の間の半順序関係 $\prec$ を次のように定義する。

$$\sigma \prec \sigma' \iff m < n \ \wedge \ \forall k \leq m, s_{i_k} = s_{j_k}$$

$$\sigma \preceq \sigma' \iff \sigma \prec \sigma' \vee \sigma = \sigma'$$

□

$\forall \sigma \neq \lambda,\ \lambda \prec \sigma$ である。

定義 3 (組合せ最適化問題) 組合せ最適化問題を 4 つ組 $(\Sigma, D, \overline{f}, \underline{f})$ で表す。ただし、

- $\Sigma = \{s_1, s_2, \ldots, s_M\}$
- $D \subset \Sigma^*$
- $\overline{f}:\ D \longrightarrow R^+$ (R^+ は 0 および正の実数からなる集合)
- $\underline{f}:\ D \longrightarrow R^+$

であり、 $D, \overline{f}, \underline{f}$ は次の条件を満たすものとする。

- $\forall \sigma, \forall \sigma'$,
  $\sigma \prec \sigma' \wedge \sigma' \in D \Rightarrow \sigma \in D$
- $\forall \sigma \in D, \forall \sigma' \in D$,
  $\sigma \prec \sigma' \Rightarrow \underline{f}(\sigma) \leq \underline{f}(\sigma') \leq \overline{f}(\sigma') \leq \overline{f}(\sigma)$
- $\forall \sigma \in D, \exists \sigma' \in D$,
  $\sigma \preceq \sigma' \wedge \overline{f}(\sigma') = \underline{f}(\sigma')$ ………………………… (1)

$\overline{f}(\sigma) = \underline{f}(\sigma)$ を満たす σ のうち、$\overline{f}(\sigma)$ を最小にする σ を求める問題を、組合せ最適化問題と定義する。 □

D の要素を「部分解」と呼び、$\overline{f}(\sigma) = \underline{f}(\sigma)$ を満たす σ を「解」と呼ぶ。解 σ_t に対し、$\overline{f}(\sigma_t)$ を「評価値」と呼ぶ。評価値を最小にする解を「最適解」と呼ぶ。

たとえば TSP において、都市をアルファベットとし、その並びを巡回経路に対応させると、評価値は巡回経路のコストであり、部分解はサブツアー (sub tour) である。サブツアーの端点に適切な都市を加えていくことにより解を生成できる。この場合、アルファベットの並びと巡回経路が直接一対一に対応しているが、複数のアルファベットの並びに対し 1 つの巡回経路が対応する多対一対応であってもよい。つまり解 σ_t に対し評価値 $\overline{f}(\sigma_t)$ が一意に定まる対応関係であればよい。

$\overline{f}(\sigma)$ は部分解 σ の末尾にアルファベットを連接していくことにより生成可能な解のうち評価値が最小である解の評価値の上界を与え、 $\underline{f}(\sigma)$ は下界を与える。多くの場合、上・下界の評価方法は改善の余地があるが、ここでは一般的に知られている評価方法あるいは容易に考案可能な評価方法について考える。

前述した TSP の例では、サブツアーに残りの都市を加えるすべての順列組合せを調べれば、部分解から生成可能な解の評価値の最小値、すなわち上・下限を求めることが可能であるが、これはしらみつぶし探索そのものであり現実的でない。一般的に知られている下界の評価方法としては、残りの都市それぞれの隣接都市までの距離を合計する方法等がある。

どの程度良い上・下界を求めることができるかは問題に依存する。実用的な計算量で十分に良い上・下界が得られる問題の場合は、ヒューリスティックを用いず、分枝限定法だけで効率良く解を求めることができるので、本報告では扱わない。多くの問題、特に実問題においては解の生成の途中で、良い上・下界を求めることは困難であり、探索空間を限定するためのヒューリスティックが必要となる。

次に、組合せ最適化問題 $(\Sigma, D, \overline{f}, \underline{f})$ を解くヒューリスティック探索アルゴリズムを定義する。この探索では、探索範囲を限定するため、最適解が得られるとは限らない。そこでこの探索範囲における最も良い解を「最良解」と呼ぶ。

定義 4 (ヒューリスティック探索) 組合せ最適化問題 $(\Sigma, D, \overline{f}, \underline{f})$ を解く次のアルゴリズムをヒューリスティック探索と定義する。

```
procedure main {
    x ← ∞
    search(Σ, λ)
}
procedure search(S, σ) { ………………………… (2)
    if f̄(σ) < x then {
        x ← f̄(σ)
        if f̄(σ) = f̲(σ) then {
            σ_t = σ
            return
        }
```

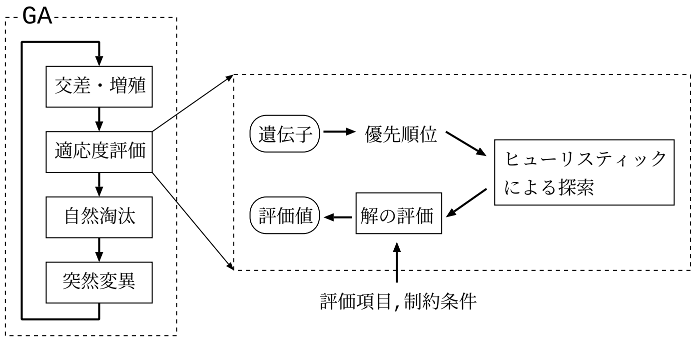

図 1: GA によるヒューリスティック探索の最適化

$\quad\quad\}$

$\quad\quad S \leftarrow \{ s \in S \,|\, \sigma \cdot s \in D$

$\quad\quad\quad \wedge\ \underline{f}(\sigma \cdot s) < x \}$ ………………………… (3)

$\quad\quad \{ T \subset S$ を選択 $\}$ ……………………………… $(*)$

$\quad\quad$ **for all** $s \in T$ **do** $\mathrm{search}(S - \{s\},\ \sigma \cdot s)$

$\quad \}$

ただし、 x, σ_t は大域的変数、'$\cdot$' は連接演算子である。 σ_t に最良解が返される。

$(*)$ で選択した T を、その探索局面における「選択枝集合」と呼ぶ。選択枝集合 T は、ヒューリスティックに基づいて求める。すなわち解の性質に関する経験的知識を用いて、探索すべき枝を限定する。 □

前述した TSP の例では、(3) の S (左辺) は次に訪問する都市の候補 s のうち、サブツアー $\sigma \cdot s$ から生成される解の評価値の下界が x より大きい都市を除いた集合である。一般に S の要素数は多く、そのすべてを探索することは組合せ爆発により現実的ではない。そこで選択枝集合として、次に訪問するコストが一番小さい都市のみからなる集合を用いるヒューリスティックが最近近傍法 (nearest neighbor method) である。しかしこの方法は明らかに選択枝集合が小さすぎ、良い解が得られないことが多い。

TSP の場合、選択した都市を次に訪問するのではなく、サブツアーの途中に挿入することにより効率を向上させることができる [3] が、本報告で扱うような実問題の場合は、適切な選択枝集合を求めるには、問題に関する詳細な知識に基づく複雑なロジックが必要となる。

3 GA によるヒューリスティック探索の最適化

前章で定義したヒューリスティック探索に GA を組み込み、選択枝集合の求め方を最適化する手法を提案する。最適化された探索で最良解を求めることにより、組合せ最適化問題 $(\Sigma, D, \overline{f}, \underline{f})$ の準最適解を求めることができる。

本手法では、ヒューリスティック探索の各探索局面において、各々の選択枝に優先順位を割り当てる。

アルゴリズムの概要を**図 1** に示す。各個体の遺伝子から各選択局面における選択枝の優先順位を求め、ヒューリスティック探索において優先順位を参照しながら最良解を生成する。そしてその最良解の評価値を個体の評価値とする。

3.1 遺伝子表現法

遺伝子列は**図 2** に示す、三つ組 (u_i, s_i, p_i) を要素とする集合である。要素数を $|g|$ で表す。三つ組 (u_i, s_i, p_i) は、探索局面 u_i における選択枝 $s_i \in \Sigma$ の優先順位が $p_i \in \mathrm{R}^+$ であることを示す。探索局面は、 s_i の選択に影響を与える σ の部分列である。探索局面の間に適当な全順序関係を導入するものとし、遺伝子列 $g = \{(u_i, s_i, p_i)\}$ は次の条件を満たす。

- $\forall i < j,\ u_i \leq u_j$
- $\forall i < j,\ u_i = u_j \Rightarrow s_i < s_j$ ………………………… (4)

遺伝子列 g から優先順位 $p(u_j, s_k)$ を求めるアルゴリズムを次に示す。

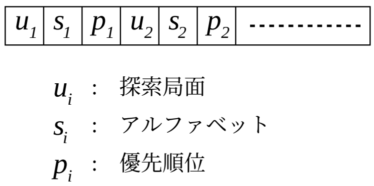

図 2: 遺伝子

for all $(u_i, s_i, p_i) \in g$ **do** {
  if $u_i = u_j \wedge s_i = s_k$ **then** {
    $p(u_j, s_k) \leftarrow p_i$
    return
  }
}
$p \leftarrow \mathbf{rand}(0, 1)$
$g \leftarrow g \cup (u_j, s_k, p)$
$p(u_j, s_k) \leftarrow p$

ただし、'$p \leftarrow \mathbf{rand}(0,1)$' は $0 < p < 1$ を満たす乱数を発生させる。'$g \leftarrow g \cup (u_j, s_k, p)$' は三つ組の集合 (遺伝子列) に三つ組 (u_j, s_k, p) を加える。このように、探索局面とアルファベットの新しい組合せが現れるたびに $|g|$ は大きくなる。

3.2 適応度評価

定義 4 のヒューリスティック探索のアルゴリズムにおいて、最後の部分

  {$T \subset S$ を選択 } ………………………… $(*)$
  for all $s \in T$ **do** search$(S - \{s\}, \sigma \cdot s)$
}

を次のように書き換えることにより、優先順位 $p(u_\sigma, s_i)$ $(i = 1, 2, \ldots, M)$ に基づく探索を行う。

  {$T \subset S$ を選択 } ………………………… $(*)$
  {$\forall s' \in T', \forall s \in (T - T'),\ p(u_\sigma, s') < p(u_\sigma, s)$
    を満たす $T' \subset T$ を選択 } ………… $(**)$
  for all $s \in T'$ **do** search$(S - \{s\}, \sigma \cdot s)$
}

$(**)$ において選択枝集合を縮小するので、$|T|$ は大きくてもよい。すなわち $(*)$ において T を求めるヒューリスティックは、簡単なものでよい。

$|T'|$ が大きいほど、ヒューリスティック探索による局所探索の比重が大きくなるが、組合せ爆発が起きる可能性も高くなる。$|T'|$ が小さいほど GA による最適化の比重が高くなる。$|T'| \equiv 1$ の場合は、探索木の枝分れがない探索となり、T の求め方によらず組合せ爆発が起きることはない。

個体の遺伝子にコーディングされた優先順位に基づく探索によって得られる最良解の評価値をその個体の評価値とし、評価値の逆数を適応度とする。

3.3 自然淘汰

個体集合から R 個 (定数) の個体を取り除く。このとき、個体集合の多様性を維持するために [2]、次のように行う。まず適応度が高い順に個体を並べる。適応度が高い方から順に個体の適応度を調べ、直前の個体の適応度との差が ϵ 以下である個体をすべて取り除く。ただし取り除く個体数は R 個までとする。取り除いた個体数を r とする。$r < R$ の場合は、$R - r$ 個の個体を適応度が低い方から順に取り除く。

3.4 突然変異

最優良個体を除くすべての個体の遺伝子の三つ組 (u_i, s_i, p_i) に対し、p_i の値を確率 P (定数) で、ランダムに選んだ値 p'_i $(0 < p'_i < 1)$ に書き換える。1 つ以上の三つ組を書き換えた個体は、評価値および適応度を再計算する必要がある。

3.5 交差・増殖

交差は一点交差法を用いる。すなわち、親個体の遺伝子列 $g_1 = \{(u_{1_i}, s_{1_i}, p_{1_i})\}$, $g_2 = \{(u_{2_j}, s_{2_j}, p_{2_j})\}$ に対して、$0 \leq r \leq |g_2|$ を満たす乱数 $r \in \mathrm{Z}$ (整数) を生成し、子個体の遺伝子列 $g = (g_1 \backslash h_1) \cup (g_2 \backslash h_2)$ を作る ('\' は差集合演算子)。ここで、

- $h_1 = \{(u_{1_i}, s_{1_i}, p_{1_i}) \mid$
  $\exists j,\ j \geq r \ \wedge\ u_{2_j} = u_{1_i} \ \wedge\ s_{2_j} = s_{1_i}\}$
- $h_2 = \{(u_{2_j}, s_{2_j}, p_{2_j}) \mid$
  $\exists i,\ j < r \ \wedge\ u_{1_i} = u_{2_j} \ \wedge\ s_{1_i} = s_{2_j})\}$

つまり、g_1 と g_2 の和集合から、重複する探索局面・選択肢の組を取り除いたものを子個体とする。r を交差点と呼ぶ。

ランダムに R 組の親個体を選んで交差を行い、個体数を R 個増やす。自然淘汰において個体数が R 個減るので、世代間で個体数は一定である。

3.6 従来法との対比

従来法 [3] では、組み合わせ最適化問題の解 σ は、「異なる」アルファベットの並びであったが、提案法では重複を許すアルファベットの並びである。

従来法では優先順位はアルファベットの関数であるため、探索局面を考慮しない。このため、選択枝であるアルファベットが探索局面に依存する場合、異なる探索局面であっても、同じアルファベットであれば優先順位が同じ、という問題がある。

一方、提案法では優先順位は探索局面とアルファベットの関数であるから、異なる探索局面であれば、それぞれ別の優先順位が GA によって定められる。反面、優先順位を割当てる対象が、探索局面とアルファベットの積集合であるから、アルファベットに優先順位に割当てる従来法に比べて探索空間が広く、最適化の効率が悪い。

なお、提案法において、探索局面が常に同一とおくと、従来法と等価になる。したがって、提案法は従来法を包含する。

4 巡回セールスマン問題への適用

TSP は単純化された問題であり本報告が扱うような実問題ではないが、提案法の効率を他手法と比較評価するため、例題として用いる。

定義 5 (TSP) n 個の都市の集合 $C = \{1, 2, \ldots, n\}$ とそれらの間のコスト $d(i, j)$ $(i, j \in C)$ が与えられたとき、すべての都市を重複なく訪れる経路 $O = c_1c_2\cdots c_n$ $(c_i \in C)$ のうち、コスト $\Sigma_{i=1}^{n} d(c_i, c_{i+1})$ *(ただし* $c_{n+1} \equiv c_1$ *とする)* が最小になるものを求める問題を *TSP* と定義する。 □

TSP の近似解法としてよく知られる手法に近傍法がある。近傍法のアルゴリズムを次に示す。

$O \leftarrow \phi$
$S \leftarrow C$
while $S \neq \phi$ **do** {
  $\{c \in S$ を選択 $\}$ ……………………………… (5)
  $O \leftarrow O \cdot c$
  $S \leftarrow S - \{c\}$
}

近傍法では、(5) の c の選び方が重要であり、よく知られたヒューリスティックとして、O の末尾の都市にもっとも近い都市 $c \in S$ を選択する最近近傍法がある。しかしこの方法は明らかに選択枝集合が小さすぎる。

そこで、まず選択枝集合として、交差が新たに生じない都市の集合 $T \subset S$ を考える。つまり、一般に TSP において、巡回経路に交差が生じる解は最適解とは成り得ないから、選択枝集合から取り除く。ただし、$T = \phi$ となる場合は、$T \leftarrow S$ とする。なお、この実験では簡単化のため部分解から生成可能な評価値最小の解の上・下界の評価は行っていない。

次に優先順位に基づいて選択枝集合を縮小する (3.2 節) ここでは $|T'| \equiv 1$ とした。すなわち、優先順位 $p(u_O, c_i)$ が最も小さい $c_i \in C$ を選び巡回経路に追加することを繰り返して解を構成する。そして優先順位を GA で最適化する。

探索局面 u_O として、O の末尾の都市のみを用いる。なぜなら、次に訪問する都市 c_i の選択にあたって最も影響を与えるのは直前に訪れた都市であり、それ以前に訪れた都市はあまり関係がないからである。

提案法に基づき近傍法に GA を組み込んだ解法 (Opt 近傍) と、従来法で求めた解の評価値 (巡回経路長 [km]) を比較する。従来法としては、挿入法に GA を組み込んだ解法 [3] (Opt 挿入)、Grefenstette の順序表現と一点交差を用いた方法 [1] (Grefenstette)、GA と局所探索を併用した解法 [2] (2optGA) を取りあげる。

ランダムに 30 都市を配置した TSP を各解法で解いたときの収束特性のグラフを**図 3** に示す。各解法とも、個体数 100, 淘汰個体数 $R = 30$ とし、突然変異は行っていない ($P = 0$)。「2optGA」では一世代ごとに 10 個体に対し 2opt による解の逐次改善を 84 回ずつ試みた。

「Opt 挿入」は初期世代において最適解を発見した。「Opt 近傍」は、「Opt 挿入」や「2optGA」に比べれば効率が悪いが、「Grefenstette」よりつねに良い解を高速に求めることができた。

「Grefenstette」は、親個体の形質が子個体の形質に遺伝しないため効率が悪い。近傍法という単純なヒューリスティックであっても、提案法に基づき GA を組み込

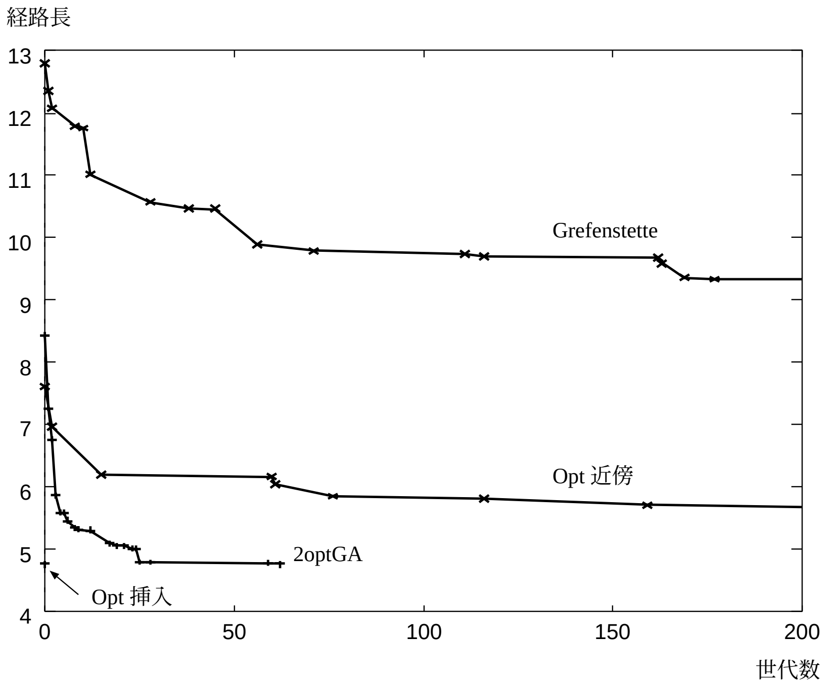

図 3: 従来法との比較 (TSP)

むことにより、形質遺伝性を考慮しない解法より効率が良い解法を構成することができた。

5　おわりに

論文 [3] で提案した、ヒューリスティック探索に GA を組み込む汎用的手法において、標準形ヒューリスティックの要件を緩和した手法を提案した。標準形への変換が容易になることにより、応用範囲が広がることが期待される。

標準形への変換が容易になる反面、最適化の効率は低下するが、形質遺伝性を考慮しない解法よりは効率が良いことを実験で示した。

一般の実問題においては、最適解を求める必要は必ずしもなく、実用上十分な解が求まれば良いことが多い。提案法を実問題に適用することにより、枝刈りの調整を GA で自動化でき、開発工数の短縮や、プログラムが単純になることによる保守性の向上、などが期待される。なお、すでに「GA によるヒューリスティック探索の最適化」手法は、バスダイヤ編成システムに応用され [4]、バス会社数社において実運用に供されている。

スケジューリング、レイアウト問題など、知識工学が応用されている分野は数多い。これらの分野に GA が応用できれば、詳細にわたる知識獲得が不要となる。本報告で提案した手法を用いることにより容易に GA の応用分野を拡大することができるものと期待される。

参考文献

[1] D.E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison Welsley (1989)

[2] 仙石, 吉原: 遺伝的アルゴリズムによる *TSP* の高速解法, 情報処理学会第 46 回全国大会 8D-4 (1993)

[3] 仙石, 吉原: *GA* によるヒューリスティック探索の最適化, 情報処理学会 論文誌 Vol.37 No.10 (1996).

[4] 北野編: 遺伝的アルゴリズム *3,* 産業図書 (1997) pp57-81