

琉球大学学術リポジトリ

解のパッケージ化競合共進化アルゴリズムの詰将棋への適用

メタデータ	言語: 出版者: 琉球大学工学部 公開日: 2010-01-14 キーワード (Ja): キーワード (En): Competitive co-evolution algorithm, Genetic algorithms, Complemental solutions, life-time, Tsume-shogi game 作成者: 根路銘, もえ子, 遠藤, 聡志, 山田, 孝治, 宮城, 隼夫, Nerome, Moeko, Endo, Satoshi, Yamada, Koji, Miyagi, Hayao メールアドレス: 所属:
URL	http://hdl.handle.net/20.500.12000/14769

解のパッケージ化競合共進化アルゴリズムの詰将棋への適用

根路 銘 も え 子 * 遠 藤 聡 志 ** 山 田 孝 治 ** 宮 城 隼 夫 **

Application of Competitive Co-evolution Algorithm with Packaging Solutions to Tsume-shogi Game

Moeko NEROME* Satoshi ENDO** Koji YAMADA** and Hayao MIYAGI**

Abstract

In this paper, we introduce a competitive co-evolution algorithm with a packaging solutions to solve the problem. This algorithm was proposed by us to a problem which doesn't have the optimal solution. In the case to solve such the problem, it needs to decide a set of effective solutions as the best solution. Our algorithm has two characteristics. The one is minimization of the number of individuals in the set by extraction of the complementary solutions. The other is evaluating solutions in some continued generations by setting a *life-time* to an individual. We apply the proposal method to the Tsume-shogi game in order to investigate its effectiveness. Furthermore, we analyze the process of the set formation. In the simulation results, our method can acquire the complementary strategies and shows a better performance than a conventional method.

Key Words: Competitive co-evolution algorithm, Genetic algorithms, Complementary solutions, *life-time*, Tsume-shogi game.

1. まえがき

競合共進化アルゴリズムは、生物の生態メカニズムである生物種間の闘争による進化プロセスから着想を得た適応型探索アルゴリズムである。このアルゴリズムは、相反する目的を持つ競合集団の進化を促進する特長を有する [1]。そこで、「競合相手に対する適応」を「問題に対する最適化」と捉えることにより、最適化問題へ適用する研究が行われている [2], [3]。特に、fitness landscape が明示的に決定できない種々の問題において、集団中の個体評価を集団間の競合結果として与えることによって、有効な解を獲得することが示されている [2], [3]。しかしながら、競合相手に応じて最適解が異なる場合には、競合共進化アルゴリズムは、全ての競合相手に対する最適解を決定することができない。このとき、1つの解を最適解と定めるのではなく、異なる競合相手に対する複数の有効解の集合が最適解として獲得されなければならない [4], [5]。筆者らは、これまでの研究において個体集合を獲得する必要性を論じてきた [6]。

本稿では、個体集合をパッケージと定義し、問題の最適化に対して必要最小限の個体で構成されるパッケージを獲得する手法として筆者らが提案した、解のパッケージ化

競合共進化アルゴリズム [7] について説明した。問題に対する有効なパッケージは、競合結果を補完し合う個体同士（相補的な個体集合）で構成されていなければならない。有効なパッケージは、全個体間の競合結果を求めることにより獲得可能であるが、全探索には膨大な計算コストを要する。本アルゴリズムでは、各世代の競合結果から相補的な個体集合を生成し、相補関係を維持しつつ個体の追加・削除を逐次的に行うことによってパッケージの再構成を行う。このとき、1世代のみの競合結果に基づく個体の淘汰は、有効なパッケージに必要とされる個体までも淘汰する恐れがある。そこで、個体の継続評価を行うために、各個体に対して、存続可能世代数を示すパラメータ：*life-time* [8] を設定する。したがって、本手法は、局所情報に基づき相補的な個体集合を生成し、*life-time* の設定による個体継続評価、個体淘汰の緩和を行うことによって、競合集団の変化に頑健なパッケージの獲得が期待できる。

本稿ではまず、競合共進化アルゴリズムの問題点を述べ、解決手法として、部分有効解集合をパッケージとして獲得する“解のパッケージ化競合共進化アルゴリズム”について説明する。さらに、適用例として競合結果が明白であるゲーム問題を扱う。詰将棋を対象問題として取り上げ、従来の競合共進化アルゴリズムとの適用結果の比較により、解のパッケージ生成の有効性を示す。また、獲得戦略の推移を解析することによって有効なパッケージの生成過程について議論する。これらの計算機実験を通して、最適解が複数の有効解の集合により構成される問題に対して本手法が有効に機能することを示す。

受理：2000年12月25日

*大学院理工学研究科 総合知能工学専攻

(Graduate Student, Doctoral Course in Complex Intelligent Systems Engineering, Graduate School of Science and Engineering)

**工学部情報工学科

(Department of Information Engineering, Faculty of Engineering)

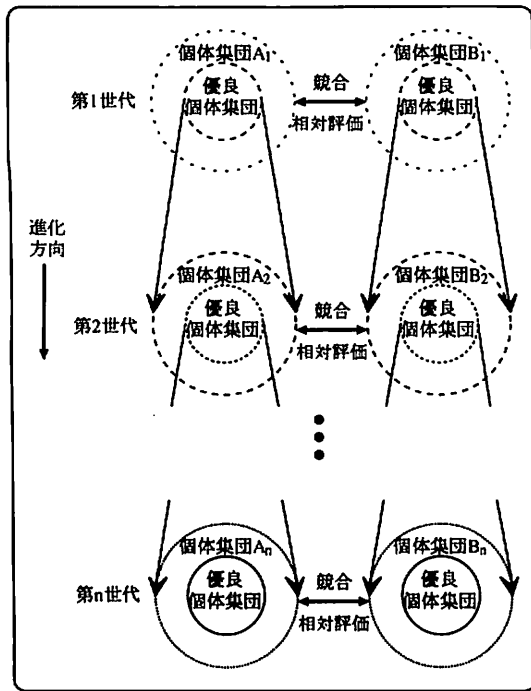


Fig. 1. 競合共進化の概念図

2. 競合共進化アルゴリズム

2.1 競合共進化の概念

自然界において、他の繁殖集団の生物と関わり、その影響で集団の遺伝子頻度が変化したり個体の性質が進化していく過程を共進化という [9], [10]。特に、競合関係にある生物同士が他の生物に対して優位に立とうとする結果、相互作用に関連している性質が互いに進化する現象を競合共進化と呼ぶ。本稿では、個体間における競合事象において、優勢である個体を競合相手に対する優位個体と定義する。競合共進化の概念図を Fig.1に示す。

競合には、同種内の個体間で起こる競合と異種の個体間もしくは集団間で生じる競合がある。Fig.1は2つの異種集団間の競合共進化を示している。異種間競合共進化の場合、両集団の目的は相反しており、競合相手個体に対して目的を達成した個体は優位個体と判断される。各異種個体間における同様の優劣比較により、一方の集団は目的をどれだけ達成したかの相対評価がなされ、もう一方の集団はその目的の達成を如何に妨げるかが相対的に評価される。

結果的に、両集団とも各世代の競合相手に対する優良個体集団が次世代へと存続することとなる。

2.2 競合共進化アルゴリズムの問題点

生態系の競合共進化の計算モデルである競合共進化アルゴリズムは、個体の評価が集団間の競合結果として与えられる。そのため、fitness landscape が明示的に決定できない問題、例えば対戦相手に応じて戦略の評価が異なるゲーム等において有効解の獲得が可能であり [3]、競合集団が互いに進化を促進するという特長を持つ [1]。その一方で、以下の問題点が指摘されている [4], [5]。

[問題点]

競合相手に応じて最適解が異なる問題に対しては、局所的な競合相手に対する最適解を獲得するため、最終的に得られた解であっても有効に機能しない場合が生じる。

全ての競合相手に対する最適解を決定できない問題では、最適解を複数の有効解の集合として定めることにより問題解決が望める。したがって、競合結果を補完し合う解を集合として獲得する枠組みが必要である。

本稿では、解集合をパッケージとして獲得する手法である、解のパッケージ化競合共進化アルゴリズム [7] について以下で説明する。

3. 解のパッケージ化競合共進化アルゴリズム

3.1 諸定義

本節では、提案アルゴリズムを説明するにあたり、必要な諸定義を行う。ここでは、最も単純な集団間の相互作用として、2種のみを進化を扱う。

[集団]

- P_i : 進化個体集団 ($i = 1, 2$)
- \bar{P}_i : P_i の競合集団 ($i = 1, 2$)
すなわち, $\bar{P}_1 = P_2, \bar{P}_2 = P_1$
- \bar{P}_i^* : P_i を評価するために \bar{P}_i からサンプリング (復元抽出) したサンプリング集合

[個体]

- p_j^i : P_i の j 番目の要素である個体 ($p_j^i \in P_i$)
- \bar{p}_j^i : \bar{P}_i^* の j 番目の要素である個体 ($\bar{p}_j^i \in \bar{P}_i^*$)
- p_k^{new} : 生成された k 番目の新個体

[パッケージ]

- \mathcal{P}_k^i : P_i の k 番目のパッケージ
($P_i = \{\mathcal{P}_1^i, \mathcal{P}_2^i, \dots, \mathcal{P}_k^i\}$)
- $\hat{\mathcal{P}}_k^i$: パッケージ \mathcal{P}_k^i における相補的な個体集合 ($\hat{\mathcal{P}}_k^i \subseteq \mathcal{P}_k^i$)
- $\tilde{\mathcal{P}}_k^i$: パッケージ \mathcal{P}_k^i において、 p_k^{new} を含めて生成した相補的な個体集合 ($\tilde{\mathcal{P}}_k^i \subseteq \mathcal{P}_k^i$)

[個体の優位相手集合]

- $\bar{P}_{i,p_j^i}^*$: 個体 p_j^i が優位である \bar{P}_i^* の部分個体集合 ($\bar{P}_{i,p_j^i}^* \subseteq \bar{P}_i^*$)

[パッケージの優位相手集合]

- $\bar{P}_{i,\mathcal{P}_k^i}^*$: \mathcal{P}_k^i の各個体 p_j^i が優位である $\bar{P}_{i,p_j^i}^*$ の積集合
($\bar{P}_{i,\mathcal{P}_k^i}^* \subseteq \bar{P}_i^*$) $\bar{P}_{i,\mathcal{P}_k^i}^* = \bigcap_{p_j^i \in \mathcal{P}_k^i} \bar{P}_{i,p_j^i}^*$
- $\bar{P}_{i,\hat{\mathcal{P}}_k^i}^*$: $\hat{\mathcal{P}}_k^i$ の各個体 p_j^i が優位である $\bar{P}_{i,p_j^i}^*$ の和集合
($\bar{P}_{i,\hat{\mathcal{P}}_k^i}^* \subseteq \bar{P}_i^*$) $\bar{P}_{i,\hat{\mathcal{P}}_k^i}^* = \bigcup_{p_j^i \in \hat{\mathcal{P}}_k^i} \bar{P}_{i,p_j^i}^*$
- $\bar{P}_{i,\tilde{\mathcal{P}}_k^i}^*$: $\tilde{\mathcal{P}}_k^i$ の各個体 p_j^i が優位である $\bar{P}_{i,p_j^i}^*$ の和集合
($\bar{P}_{i,\tilde{\mathcal{P}}_k^i}^* \subseteq \bar{P}_i^*$) $\bar{P}_{i,\tilde{\mathcal{P}}_k^i}^* = \bigcup_{p_j^i \in \tilde{\mathcal{P}}_k^i} \bar{P}_{i,p_j^i}^*$

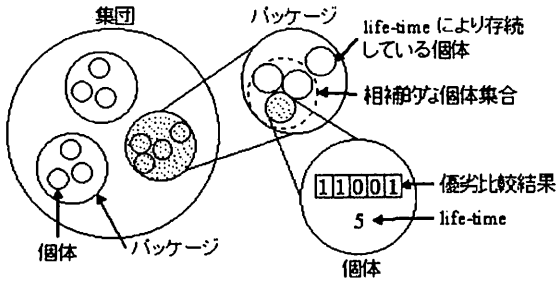


Fig. 2. パッケージの概念図

3.2 相補関係

個体 p_j^i と個体 p_i^j ($p_j^i, p_i^j \in P_i$) について, 以下の条件を満たす場合における p_j^i と p_i^j の関係を指す。

$$\begin{aligned} \text{条件} \quad & (\overline{P}_{i,p_j^i}^* - (\overline{P}_{i,p_j^i}^* \cap \overline{P}_{i,p_i^j}^*)) \neq \phi, \\ & (\overline{P}_{i,p_i^j}^* - (\overline{P}_{i,p_i^j}^* \cap \overline{P}_{i,p_j^i}^*)) \neq \phi \end{aligned}$$

3.3 パッケージ

パッケージ P_k^i は, 相補性を保つ個体集合 \hat{P}_k^i と *life-time* によって存続している個体とで構成される個体集合である。個体は, 各世代におけるサンプリング集合 \overline{P}_i^* との優劣比較の結果 (優位である場合を 1, 劣る場合を 0 で表したビット列) と存続可能世代数を示す *life-time* を属性として持つ。*life-time* については, 次節で説明する。パッケージの概念図を Fig.2 に示す。

3.4 life-time

本手法では, 1 世代の競合結果から相補的な個体集合 \hat{P}_k^i の生成を行う。しかしながら, 1 世代のみの評価では, 有効なパッケージにおける必要性の判断が難しい。そこで, 数世代にわたり個体を評価するために, 存続可能世代数 (寿命) を示す *life-time* を各個体に設定する。*life-time* は, 集団中の個体の多様性を維持する事を目的とし, Michalewicz らによって GA に導入された [8]。Michalewicz らの手法 (GAVaPS) において, 各個体は年齢を示す *age* を持ち, 世代毎に *age* は増加する。*age* が *life-time* に達した個体は淘汰される。本手法では, 多様性の維持と個体集合の継続評価に加え, 相補的な個体集合を維持するために *life-time* を用いる。また, 進化過程において *life-time* の増加, すなわち “延命” を行うように拡張する。これにより, 相補的な個体の淘汰の緩和を計る。*life-time* の初期値は一意に定められ, 各世代における個体 p_j^i の *life-time* T_j^i は式 (1) により更新される。

$$T_j^i(t+1) = \begin{cases} T_j^i(t) + \alpha & (p_j^i \in \hat{P}_k^i) \\ T_j^i(t) - \beta & (p_j^i \notin \hat{P}_k^i) \end{cases} \quad (1)$$

ただし,

- t : 競合共進化世代数
- α : *life-time* の増加定数
- β : *life-time* の減少定数

3.5 提案アルゴリズム

競合共進化アルゴリズムを実装する進化形態として, 同時的進化と交互進化の 2 通りが考えられる。本稿では, 一方の集団を一定期間進化させた後にもう一方の集団を進化させる交互進化の形態を採用する。その理由は, 同時的進化では局所的な競合集団に対して絶対的な優位性を示せないまま次世代へと移行する可能性が高いからである。それに対し交互進化では, 完全に優位性を示した後に次世代へと移行する。交互進化を実装するには, 競合共進化を行う世代と GA オペレータを適用する世代の 2 種の世代概念がある。本手法では, 競合共進化世代を t , GA オペレータ適用世代を $ga.t$ と表記する。提案アルゴリズムを Fig.3 に示す。アルゴリズムの手順は以下の通りである。

step1: 初期個体集団の生成

初期個体集団 P_1 および P_2 において, 各パッケージ内の最大初期個体数 ini_num 以下の個体数をランダムに決定し, N 個のパッケージをランダムに生成する。(step2 ~ step7 において, P_1 が進化集団であるため, P_2 を \overline{P}_1 と表記する。)

step2: 評価基準の個体集合 \overline{P}_1^* のサンプリング

\overline{P}_1 から M 個体 (\overline{P}_1 内の個体数以下) をランダムにサンプリングし, \overline{P}_1^* を生成する。

step3: 個体の優劣比較

P_1 の各個体は \overline{P}_1^* の全個体と優劣を比較する。その結果から, 個体 p_j^i が優位となる \overline{P}_1^* の個体集合 $\overline{P}_{i,p_j^i}^*$ を求める。

step4: 相補的な個体集合 \hat{P}_k^i の生成

[$t = 0$ の場合]

$\overline{P}_{i,p_j^i}^*$ を計算する。
 個体 \overline{p}_j^i に関して,
 $\overline{p}_j^i \notin \overline{P}_{i,p_j^i}^*, \overline{p}_j^i \in \overline{P}_{i,p_j^i}^*$ の時,
 個体 p_j^i を \hat{P}_k^i の要素とする。

[$t \neq 0$ の場合]

$\overline{P}_{i,p_j^i}^*$ と $\overline{P}_{i,p_k^i}^*$ を計算する。
 個体 \overline{p}_j^i に関して,
 $\overline{p}_j^i \notin \overline{P}_{i,p_j^i}^*, \overline{p}_j^i \notin \overline{P}_{i,p_k^i}^*, \overline{p}_j^i \in \overline{P}_{i,p_j^i}^*$ の時,
 個体 p_j^i を \hat{P}_k^i の要素とする。

step5: 新個体 p_k^{new} の生成

P_1 の個体からランダムに 2 個体を選択し, 任意の GA オペレータ [11] を適用することにより, p_k^{new} を生成する。新個体数が N に達するまでこの処理を行う。

step6: p_k^{new} の追加判断

後述の追加判断処理を実行する。 $ga.t = ga.t + 1$ とした後に, P_1 の進化過程の終了判断を行い, 終了条件を満たしていれば step7 へ。そうでなければ, step5 へ戻る。なお, 終了条件は, 以下の 2 つである。

- 全パッケージが他集団中の全個体に優位となる場合
- $ga.t$ が設定した上限に達した場合

step7: 個体の淘汰 (*life-time* の更新)

P_1 の各個体の *life-time* を式 (1) に基づき更新する。 $lifetime \leq 0$ である個体を淘汰する。

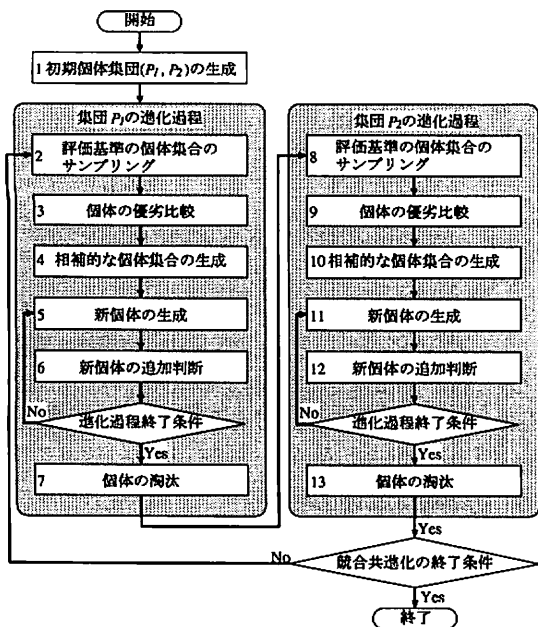


Fig. 3. 解のパッケージ化法を導入した競合共進化アルゴリズム

step8 ~ step13: P₂ の進化過程

P₁ と P₁(= P₂) の役割りを入れ替え、P₂ を進化集団として、step2 ~ step7と同様の処理を行う。その後、競合共進化アルゴリズムの終了条件を満たしていなければ、t = t + 1とした後に step2 へ。満たしていれば終了。

本アルゴリズムにおいて、step2 ~ step13 までを競合共進化1世代とする。なお、step6 (12) の p_k^{new} 追加判断における内部手続きについて以下で説明する。

[p_k^{new} の追加判断アルゴリズム (step6, step12)]

- step1: p_k^{new} を P_k¹ に含め、P₁p_k¹ を計算する。
 個体 p_j¹ に関して、p_j¹ ∉ P₁p_k¹, p_j¹ ∈ P₁p_k¹ の時、個体 p_j¹ を P₁p_k¹ の要素とする。
- step2: p_k^{new} ∈ P₁p_k¹ の時、step3 を実行する。そうでなければ、p_k^{new} を淘汰する。
- step3: |P₁p_k¹| と |P₁p_k¹| を比較し、p_k^{new} の追加判断を行う。(ただし、| | 記号は、各集合の個体数を表す。)
- Case1: |P₁p_k¹| < |P₁p_k¹|
 p_k^{new} を P_k¹ に追加し、P₁p_k¹(ga.t + 1) = P₁p_k¹(ga.t) とする。
- Case2: |P₁p_k¹| = |P₁p_k¹|
 P₁p_k¹, P₁p_k¹ の両集合内において、各々最大個体数を示す |P₁p_j¹| (p_j¹ ∈ P₁p_k¹) と |P₁p_k¹| (p_k¹ ∈ P₁p_k¹) を比較し、|P₁p_j¹| > |P₁p_k¹| ならば、p_k^{new} を P_k¹ に追加し、P₁p_k¹(ga.t + 1) = P₁p_k¹(ga.t) とする。そうでなければ、p_k^{new} を淘汰する。
- Case3: |P₁p_k¹| > |P₁p_k¹|
 |P₁p_k¹| と |P₁p_k¹| を比較し、|P₁p_k¹| > |P₁p_k¹| ならば、p_k^{new} を P_k¹ に追加し、P₁p_k¹(ga.t + 1) = P₁p_k¹(ga.t) とする。そうでなければ、p_k^{new} を淘汰する。

4. 詰将棋への適用実験

本節では、最適戦略の決定が難しい問題として詰将棋を取り上げる。詰将棋は、各局面の最善手によって最適戦略が構成される上、正解手以外の有効な手も多く存在するため、最適戦略の獲得が難しい問題であるといえる。獲得解の推移を解析することによって有効なパッケージの生成過程について議論し、最適解が複数の有効解の集合により構成される問題に対する有効性の検証を目的とする。

4.1 詰将棋

詰将棋は、将棋の盤と駒を使って先手が王手の連続で後手の玉を捕獲する (詰める) 1 人用パズルである [13]。先手は最短、後手は最長となるように指し続け、最終的に詰むことができた手順が正解手順とされる。プレーヤは、この正解手順の獲得を目指す。詰将棋の規則と用語を以下に示す。

- 完全作：正解手順が1つしか存在しない問題
- 余詰：先手が正解手順以外の手を指しても、後手玉を詰める事ができること
- 不完全作：余詰のある作品 (発表された詰将棋問題のうち、約 1 割に余詰が存在)
- 変化手順：後手の選択による詰手順の変化
- 合駒：駒の効きを妨害する手
- 無駄合：手数をのばす以外に意味をもたない合のこと (無駄合は詰手順の中から除外する)

詰将棋問題は完全作でなければならないが、問題の作者の意図に関わらず、変化手順や余詰を含む場合がある。特に、正解手順と判定されても良いような変化手順が含まれる作品は多く存在する。しかしながら、余詰、変化手順の有無はプレーヤには知らされておらず、プレーヤが問題を解くためには、正解手順の手だけではなく、変化手順に対応する手も獲得する必要がある。人工知能の分野では、詰将棋を組み合わせ最適化問題として捉え、詰将棋をコンピュータで解く試みが行われてきた [12], [13]。以下では、正解手順の手順だけでなく、変化手順に対応する手順をパッケージとして獲得することにより本手法の有効性を検証する。

4.2 詰将棋問題に対するモデル設計

- 個体：戦略を個体とする。
- 戦略：手の系列を戦略とする。
- 集団：P₁ を先手戦略集団、P₂ を後手戦略集団とする。
- 先手戦略の相補性：後手戦略の変化手順がある場合、正解手順に対する手順と変化手順に対処し得る手順が相補関係となり、パッケージが生成される。
- 後手戦略の相補性：変化手順がある場合には、変化手順と正解手順によりパッケージが生成される。
- 個体のコーディング：1 戦略は、実行手の成り情報、駒の動作パターンの優先順位を 1 手とした手の系列 (動作パターン総数: 438), 持ち駒を置く位置の優先順位 (位置総数: 81) により構成される。Fig.4に戦略のコーディング例を示す。Fig.4の先手戦略において、第 0 遺伝子座は駒の成り情報を示している。遺

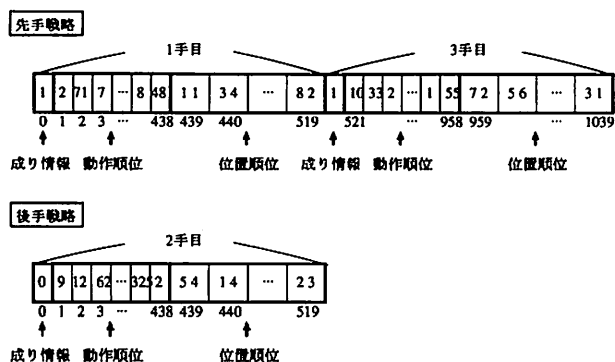


Fig. 4. 個体のコーディング

伝子は、0：成らない、1：成る を示す。第1～438 遺伝子座は1手目における動作の優先順位を示しており、遺伝子は各優先順位における動作パターンのインデックスを示している。また、第439～519 遺伝子座は1手目において持ち駒の指し位置の優先順位を示しており、遺伝子は各優先順位における駒の指し位置を示している。

対戦時は、優先順位順に動作を選択する。先手ならば「王手している手」を実行手とし、後手ならば「王手から逃げる手」を実行手と決定する。このコーディングをにより、ゲームが常に実行可能となる。

4.3 提案手法の拡張

提案アルゴリズムを詰将棋に適用するにあたり、以下の拡張を行う。

p_k^{new} の追加判断アルゴリズム (step6, step12) : 詰将棋では、対戦相手に対する詰み結果が優劣比較結果に相当する。しかしながら、詰み結果だけでなく詰み上がりの手数も戦略の質を決定する要素の一つである。したがって、詰将棋において相補的な個体集合を決定する場合には、

1. 戦略の詰み結果
2. 戦略の適応度

の比較によって、相補的な集合を生成する。具体的には、「Case2:」を以下のように変更する。

Case2: $|\hat{P}_k^1| = |\hat{P}_k^2|$
 \hat{P}_k^1, \hat{P}_k^2 の両集合内において、各々最大適応度を示す $F_{p_j^1} (p_j^1 \in \hat{P}_k^1)$ と $F_{p_k^2} (p_k^2 \in \hat{P}_k^2)$ を比較し、 $F_{p_j^1} > F_{p_k^2}$ ならば、 p_k^{new} を \hat{P}_k^1 に追加し、 $\hat{P}_k^1(ga.t+1) = \hat{P}_k^1(ga.t)$ とする。そうでなければ、 p_k^{new} を淘汰する。

適応度

アルゴリズムの拡張に伴い、詰み上がりの手数を含めた適応度を設定する必要がある。その適応度を以下に示す。

1対戦における先手戦略の適応度 $F_{p_j^1}$ は、式(2)により求められる。先手戦略において、詰み上がりの持ち駒の数も解に影響するため、持ち駒数も考慮する。

$$F_{p_j^1} = \frac{result \times GAME}{tsumi.num \times (1 + pieces)} \quad (2)$$

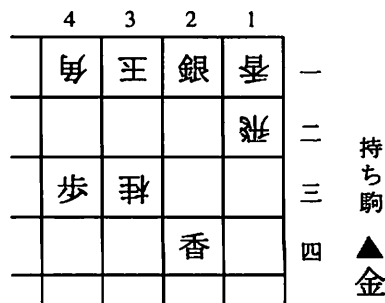


Fig. 5. 3手詰問題例

ここで、

GAME : 問題終了の設定手数
 tsumi.num : 実際に詰んだ手数
 pieces : 持ち駒の数

後手戦略の適応度 $F_{p_k^2}$ は、式(3)により求められる。

$$F_{p_k^2} = \frac{result \times tsumi.num}{GAME} \quad (3)$$

4.4 GA オペレータ

GA オペレータは、個体のコーディングに応じたオペレータを採用する。

- 一点交叉：1手毎の遺伝子を交叉させる。
- 突然変異1：成り情報を0→1, 1→0にする。
- 突然変異2：動作順位内の1遺伝子をランダムに選び、同手内の他の遺伝子と交換する。
- 突然変異3：位置順位内の1遺伝子をランダムに選び、同手内の他の遺伝子と交換する。
- ランク操作1：実行した動作の順位を上げる。
- ランク操作2：実行した位置の順位を上げる。

4.5 問題設定

詰将棋問題集 [14] の中から、変化手順が存在する3手詰め問題を採用する。本実験で用いる問題を Fig.5に示す。Fig.5において、正解手順および変化手順は以下の通りである。

正解手順：▲3二銀成 △同飛 ▲2一金
 変化手順：▲3二銀成 △同角 ▲4二金
 ▲3二銀成 △同玉 ▲4二金

したがって、先手戦略 (▲) は「1手目：▲3二銀成、3手目：▲2一金」と「1手目：▲3二銀成、3手目：▲4二金」の2戦略を解集合として獲得しなければならない。

なお、本実験で用いたパラメータを Table1に示す。従来の競合共進化アルゴリズムの結果と比較するため、競合共進化アルゴリズムの適用実験におけるパラメータも Table1に示す。

4.6 実験結果と考察

獲得したパッケージを評価するために、先手戦略を後手の解となり得る3戦略と対戦させ、後手戦略を先手の解となり得る2戦略と対戦させる。各戦略は以下の通りである。

Table 1. パラメータ

	提案手法	従来手法
サンプリング数 (M)	全個体	全個体
パッケージ数 (N)	4	-
パッケージ内最大初期個体数	5	-
パッケージ内最大個体数	10	-
集団内個体数	-	20
個体の life-time 初期値 (T)	10	-
life-time の増加定数 (α)	1	-
life-time の減少定数 (β)	2	-
交叉率	0.2	0.2
突然変異率 1	0.25	0.25
突然変異率 2	1.0	1.0
突然変異率 3	1.0	1.0
GA オペレータの最大適用回数	10	10
競合共進化世代数	20	20

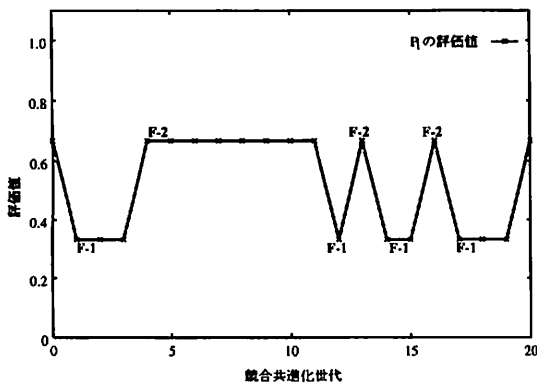
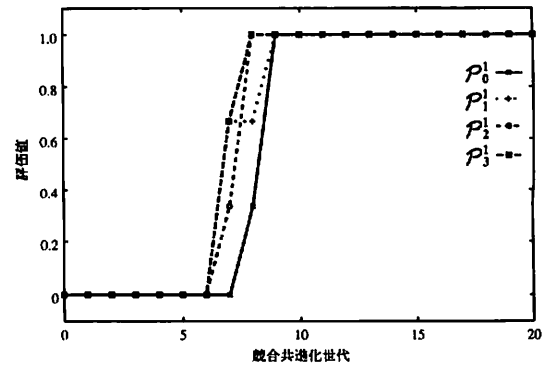


Fig. 6. 従来手法による獲得先手戦略の評価値の推移

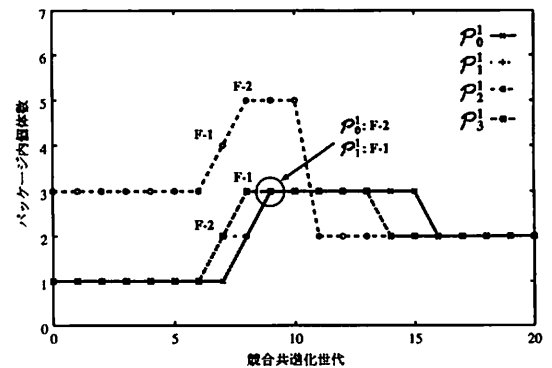
- 先手戦略 F-1: ▲ 3 二銀成 ▲ 2 一金
- 先手戦略 F-2: ▲ 3 二銀成 ▲ 4 二金
- 後手戦略 S-1: △ 3 二飛
- 後手戦略 S-2: △ 3 二角
- 後手戦略 S-3: △ 3 二玉

従来手法による獲得先手戦略に関して、評価戦略との対戦における詰み結果を評価値とした場合の最良個体の評価値の推移を Fig.6 に示す。また、提案手法による獲得パッケージに関して、評価戦略との対戦における詰み結果を評価値とした場合の各パッケージの評価値の推移を Fig.7 および Fig.8 の上図に示す。Fig.6 および Fig.7, 8 の (a) において、横軸は競合共進化世代であり、縦軸は評価値を示す。Fig.6 において 1.0 の値は、獲得戦略が正解手順および変化手順に対して詰むことを意味する。Fig.7, 8 の (a) において 1.0 の値は、獲得パッケージが最良パッケージであることを示している。

Fig.6 と Fig.7 の (a) を比較すると、Fig.6 の評価値が振動しているのに対して、提案手法は、徐々に評価値を上げ、最終的に、正解手順および変化手順の両方を詰むことが可能なパッケージを獲得していることがわかる。採用した問題は 1 つの戦略で両手順を詰むことができないため、従来手法は、各世代における後手戦略集団に応じて獲得する最良戦略が異なる。Fig.6 に示されているように、先手の 2 戦略である F-1 と F-2 が交互に獲得されており、その結果、Fig.6 の評価値の振動現象が起こっている。この結果から、複数の部分有効解によって最適化が実現される問題



(a) 先手戦略集団における各パッケージの評価値の推移



(b) 先手戦略集団における各パッケージ内個体数の推移

Fig. 7. 先手戦略集団における各パッケージの評価値およびパッケージ内個体数の推移

において、解集合生成機構導入している提案アルゴリズムが従来法よりも有効であることが示されたといえる。

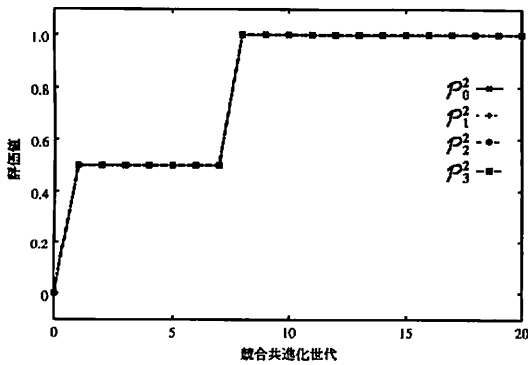
次に、提案手法が有効なパッケージを獲得するまでの集団内部変化を Fig.7, 8 を用いて以下で説明する。なお、Fig.7, 8 の (b) は、各戦略集団における各パッケージ内の個体数の推移を示している。これらの図において縦軸はパッケージ内の個体数を示す。両図と併せて、相補的な個体集合を獲得するまでの集団内部変化を以下で説明する。

先手 1 ~ 6 世代： 後手戦略に対して詰み得る戦略を探索。

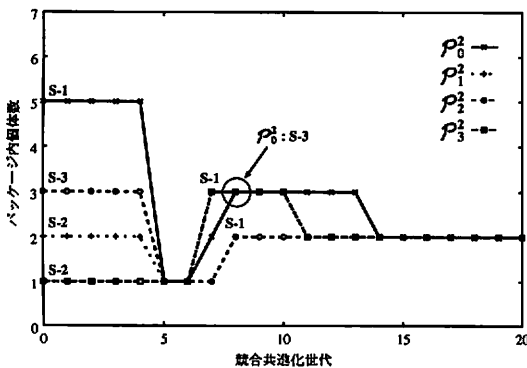
後手 1 ~ 6 世代： パッケージ P_0^2 は戦略 S-1, P_1^2, P_3^2 は戦略 S-2, P_2^2 は戦略 S-3 を各々保持しており、先手戦略に詰まれない手として次世代へ存続する。また、不要戦略の life-time が 0 になる、5 世代目からは各パッケージとも 1 戦略のみを保持することになる。

先手 7 世代： P_1^1, P_3^1 が S-2, S-3 に対して詰むことができる戦略 F-2 を戦略の追加により獲得。また、 P_2^1 が S-1 に対して詰み得る戦略 F-1 を獲得。

後手 7 世代： 戦略 S-2, S-3 を詰む戦略 F-2 の出現により、 P_1^2, P_3^2 が F-2 に詰まれない戦略 S-1 を追加している (Fig.8 における 7 世代目の個体数増加より)。しかしながら、追加された戦略は、F-2 との対戦で効力を発揮していない (Fig.8 における 7 世代目の評価値が 0.5 であることより)。その理由として、F-2 に対処するための動作順位がまだ低いため、評価集団と対戦において実行手として選択されていない可能性があると考えられる。



(a) 後手戦略集団における各パッケージの評価値の推移



(b) 後手戦略集団における各パッケージ内個体数の推移

Fig. 8. 後手戦略集団における各パッケージの評価値およびパッケージ内個体数の推移

先手 8 世代: P_3^1 が戦略 S-1 を読む戦略 F-1 を戦略の追加により獲得。また、 P_2^1 が戦略 S-2, S-3 を読む戦略 F-2 を獲得。その結果、 P_2^1, P_3^1 は相補的な解集合として F-1, F-2 の組み合わせを獲得。

後手 8 世代: P_1^2, P_3^2 に 7 世代目で追加された戦略の動作順位が上がったため、両パッケージは相補的な解集合として S-1, S-2 の組み合わせを獲得。また、戦略 F-1 の出現に伴い、 P_0^2 が F-1 に詰まれない戦略 S-3 を獲得。さらに、 P_2^2 は、F-2 に詰まれない戦略 S-1 を戦略の追加によって獲得。したがって、この世代で全てのパッケージが相補的な解集合を獲得。

先手 9 世代目以降: P_0^1 が戦略 F-1 を P_1^1 が戦略 F-2 を獲得。以降、不要個体が徐々に淘汰される。

後手 9 世代目以降: 後手戦略も先手戦略と同様、不要個体が淘汰され、個体数が収束する。

この結果は、部分的な結果から段階的に相補的な個体集合を獲得していることを示している。また、後手戦略の 7 世代目で追加された戦略も、動作順位を上げる遺伝操作によって有効な動作の順位を上げ、最終的には必要な戦略を生成している。したがって、順位上げの操作も解獲得に有効に機能しているといえる。

獲得戦略集合の構成に関して、後手戦略は正解手順および変化手順で 3 戦略による構成が考えられる。変化手順 2 種類に対して、先手戦略は同じ手で読むことができるため、後手戦略における変化手順 2 種類の評価は同じである。したがって、後手戦略に関しては、S-1 と S-2, S-1 と S-3 のいずれかの組み合わせが獲得されれば良い。例とし

てあげている結果では、双方を別パッケージとして獲得する結果を得た。他の実験においては、いずれか一つの組み合わせを獲得する場合もあった。したがって、提案手法は、最小個体数で最大の利得を得る組み合わせとして、「双方を別パッケージとして獲得する」もしくは、「いずれかを獲得する」結果を得た。先手戦略に関しては、全ての実験結果において F-1 と F-2 の組み合わせを獲得した。したがって、本手法が、詰将棋における変化手順の発見および対応手の獲得に対し有効な結果を示した。

本実験を通して、詰将棋で部分最適解の集合を獲得できたことは、同様の問題にも対応できると考えられる。

5. おわりに

本稿では、まず、筆者らの提案手法“解のパッケージ化を導入した競合共進化アルゴリズム”について述べた。提案手法は、(1) 局所的な競合結果のみから解の相補性を導出し、さらに解集合の継続評価および相補的な解集合を維持するための (2) *life-time* を設定することにより、有効なパッケージの獲得を可能にしている。本稿では、本アルゴリズムを詰将棋へ適用し、従来手法との比較により解のパッケージ化法の導入による提案手法の有効性を示した。また、相補的な解集合が段階的に獲得される様子を示し、結果として、問題を解くために必要最小限の個体で生成されるパッケージの獲得を示した。以上のことから、最適解が複数の有効解の集合により構成される問題に対して、本アルゴリズムが有効に機能することを確認した。

文献

- [1] 井庭 斉志: 進化論的計算の方法, 東京大学出版会 (1999).
- [2] W.D. Hillis: *Co-evolution parasites improve simulated evolution as an optimization procedure*, *Artificial Life II*, Addison-Wesley, pp.313-323(1991).
- [3] M. Nerome, K. Yamada, S. Endo and H. Miyagi: *Competitive Co-evolution Model on the Acquisition of Game Strategy*, *Lecture Notes in Artificial Intelligence*, Springer, pp224-231(1997).
- [4] Dario Floreano and Stefano Nolfi: *God Save the Red Queen! Competition in Co-Evolutionary Robotics*, In *Proceedings of the second International Conference on Genetic Programming*, pp.398-406(1997).
- [5] 根路銘 もえ子, 山田 孝治, 遠藤 聡志, 宮城 隼夫: ゲーム戦略の獲得における競合共進化モデル, 琉球大学工学部紀要, 第 54 号, pp.109-116(1997).
- [6] M. Nerome, K. Yamada, S. Endo, H. Miyagi: *Competitive Co-evolution Based Game-Strategy Acquisition with the Packaging*, *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems (KES'98)*, Adelaide, South Australia, pp.184-189(April 21-23, 1998).
- [7] 根路銘 もえ子, 遠藤 聡志, 山田 孝治, 宮城 隼夫: 解のパッケージ化法を導入した競合共進化アルゴリズムの提案, 電気学会論文誌, Vol.121-C, No.3, 2000 (掲載決定).
- [8] Zbigniew Michalewicz: *Genetic Algorithms + Data Structures = Evolution Programs*, Third, Revised and Extended Edition, Springer(1996).
- [9] Futuyma, D.J. and D. Jablonski: *Coevolution*, Sinauer(1983).
- [10] 河田 雅圭: 進化論の見方, 紀伊國屋書店 (1989).
- [11] D.E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley(1989).
- [12] 松原 仁: 将棋とコンピュータ, 共立出版 (1994).
- [13] 伊藤 琢己, 河野 泰人, 脊尾 昌宏, 野下 浩平: 詰将棋, ゲームプログラミング, pp.130-138, 共立出版 (1998).
- [14] 飯野 健二: 実践に勝つ! 詰め将棋, 池田書店 (1998).