

遺伝的プログラミングを用いた 積み付けアルゴリズムの対話的生成

蛭田 悠介* 西原 慧（横浜国立大学）
小熊 祐司（株式会社 IHI） 藤井 正和（株式会社 IHI・横浜国立大学）
中田 雅也（横浜国立大学）

Interactive construction of loading algorithm with Genetic Programming

Yusuke Hiruta*, Kei Nishihara, (Yokohama National University)

Yuji Koguma, (IHI Corporation), Masakazu Fujii, (IHI Corporation, Yokohama National University)

Masaya Nakata, (Yokohama National University)

This paper integrates an interactive framework into Automatic Construction technique of Loading Algorithms based on Cartesian Genetic Programming (ACLA-CGP). While ACLA-CGP has a capacity to derive a loading algorithm optimizing a given objective function, its validity still remains unclear under human evaluations. Our interactive ACLA-CGP, i.e., i-ACLA-CGP, is designed to reduce human evaluations as possible and thus evaluates' burdens. Experimental results show that i-ACLA-CGP successfully derives loading algorithms with 50 human evaluations while satisfying their preferences.

キーワード：遺伝的プログラミング，対話型進化計算，積み付け最適化
(genetic programming, interactive evolutionary computation, loading optimization problem)

1. 序 論

近年，物流業界とりわけ現場の定型作業に要する人材不足が指摘されている⁽¹⁾。特に積み付けパターン（段ボール箱の積み方）の設計作業は，作業者の経験則を頼りにするため人的かつ時間的な負担が大きい。例えば，積み付けパターンは輸送コストの抑制や段ボール箱の検品性向上など，複数の目的に配慮して設計される必要がある。さらに目的の優先順位は物流拠点の規模に依存するため，これを熟知した作業者の選好を積み付けパターンに反映することが求められる。現状では，段ボール箱の種類や個数に応じて熟練の作業者が積み付けパターンを毎回設計する必要があり，これに伴う負担を軽減するためには作業者の選好を反映した積み付けパターン設計の自動化が重要となる。

我々は，作業者の選好を反映した積み付けアルゴリズムを自動生成可能な手法構築を目標とし，ハイパーヒューリスティクス[†]と対話型進化計算⁽²⁾を融合した方法論を検討している。この方法論では，ユーザの解評価負担を軽減するために，対話的最適化におけるユーザの解評価回数を可能な限り削減することが重要となる。そこで我々の既存研

究では，Cartesian Genetic Programming (CGP)⁽³⁾を用いた積み付けアルゴリズムの自動生成技術（Automatic Construction technique of Loading Algorithms based on CGP: ACLA-CGP）を構築した⁽⁴⁾。ACLA-CGPは，段ボール箱の積み付け方法や位置を決定するヒューリスティック（選択ルール）を複数定義し，選択ルールの実行順をCGPを用いて最適化する。この結果ACLA-CGPは，アルゴリズムの設計空間の膨大化を抑制する効果を生み，200回（4解個体×50世代）の解評価で所与の目的関数を満たす積み付けアルゴリズムが導出できることを示した。

一方でACLA-CGPは，段ボール箱の検品性などの単一かつ所与の目的関数を前提としており，対話型進化計算と融合した際の性能確認はできていない。したがって，ユーザ評価の不確定性を伴う対話的最適化において，ACLA-CGPの有効性は明らかでない。具体的には，最適化過程でのユーザ評価の変化や複数の目的を同時に考慮する場合において，ACLA-CGPがユーザの選好を反映させた積み付けアルゴリズムを自動生成可能か検証する必要がある。また，200回の解評価では依然としてユーザの解評価負担が大きいことも懸念される。

そこで本稿では，対話型進化計算を組み込んだACLA-CGP（interactive-ACLA-CGP: i-ACLA-CGP）を構築し，被験者

[†] 進化計算を用いた解法アルゴリズムの自動生成技術のこと

実験を通してその有効性を検証する．具体的には，ACLA-CGPにおける(1+4)-ESの解評価をユーザ評価に置き換えることでi-ACLA-CGPを構築する．ユーザ評価の方法には，点数方式⁽⁵⁾やランキング方式⁽⁶⁾などが存在するが，本稿ではユーザが単一の解のみを選択する単一選択方式を採用する．この方式は，CGPが用いる(1+4)-ESとの親和性が高く，掲示された全ての解を定量的に評価する必要がなくなる．したがって，ユーザの解評価負担は主に世代数に依存し，ユーザの解評価負担を軽減できる．例えば，4つの解個体を50世代だけ最適化させると，実質的に50回の解評価作業で済み，解個体の比較も容易である．

本稿の構成は次の通りである．2章でACLA-CGPについて述べ，3章でi-ACLA-CGPについて説明する．4章では，被験者実験の方法と結果について述べる．5章でi-ACLA-CGPが導出した積み付けアルゴリズムの分析を行い，最後に6章で本稿の結論を述べる．

2. ACLA-CGP

本章では，ACLA-CGPで扱う計算上の諸概念を述べた後，具体的なメカニズムについて述べる．なお，本稿では異なる種類の段ボール箱を複数個のパレットに積み付ける混載積み付け問題を扱う．

〈2・1〉 計算上の諸概念の定義

- **積み付け空間と座標系** ACLA-CGPで用いる積み付け空間と座標系を定義する．積み付け空間は，段ボール箱を積み付け可能な3次元の空間であり，パレットの幅(x 軸)，奥行(y 軸)，最大積み付け高さ(z 軸)で規定される．パレットの幅と奥行はパレットの典型的な平面寸法(T11型)⁽⁷⁾である1100 mm，1100 mmにそれぞれ設定し，最大積み付け高さを1800 mmに設定する(図1(a))．積み付け空間の座標系は，正面からみてパレットの左，奥，下の隅を原点とし，原点から右方向，手前方向，上方向をそれぞれ x, y, z 軸方向とする． z 軸の原点はパレットの表平面に設定し，パレットの厚みは積み付け空間に含まない．

- **アイテムとブロック** 段ボール箱を計算上の概念としてアイテムと呼ぶ．アイテムは複数種類存在し，形状(幅，奥行，高さ)が既知の直方体形状である．ブロックは同一種類のアイテムをまとめた直方体形状であり，図1のように棒積みあるいはピンホール積みで構成される．

- **〈2・2〉 メカニズム** ACLA-CGPで導出する積み付けアルゴリズムはブロック単位で積み付けを行い，積み付け空間を逸脱しない積み付けパターンを出力する．また，ブロックの種類と積み付け位置に関するヒューリスティック(選択ルール)を定義し，同ルールの実行順を最適化することで積み付けアルゴリズムを自動生成する．具体的にCGPは選択ルールの実行順を最適化した関数モデル(CGPモデル)

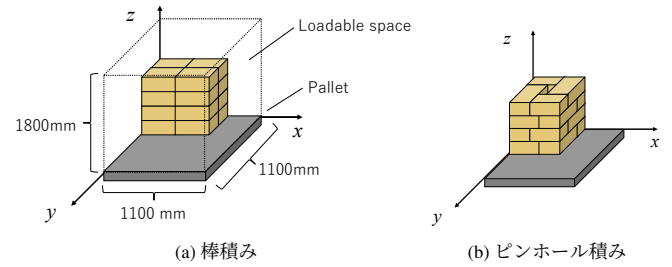


Fig. 1.: 積み付け空間とブロックの構築例

Loadable space and an example of block construction.

ル)を導出し，これを積み付けアルゴリズムの雛形に組み込むことで，目的関数を満たす積み付けアルゴリズムを導出する．以下では，積み付けアルゴリズムの実行手順と，選択ルールを用いたCGPによる最適化手順について述べる．

- **積み付けアルゴリズムの実行手順** CGPモデルを組み込んだ積み付けアルゴリズムは，以下の手順で積み付けパターンを決定する．

入力 アイテムの種類ごとの個数と形状，使用可能なパレット数，積み付け空間の規定値

出力 積み付けパターンもしくは積み付け失敗ステータス

Step 1 (初期化) 未積み付けアイテムリストに全てのアイテムを登録する．

Step 2 (終了判定) 未積み付けアイテムリストが空であれば，現時点での積み付けパターンを出力する．

Step 3 (積み付け可能なブロック群の生成) 現在の積み付け状態において積み付け可能位置を全て求め，それらの位置に積み付け可能なブロックを生成しブロック群に追加する．ブロックには積み付け位置の情報も含まれる．

Step 4 (積み付けブロックの選択) CGPモデルが出力した実行順で選択ルールを適用し，積み付け可能なブロック群からブロックを1つ選択する．

Step 5 (積み付け実行) Step 4で選択されたブロックを積み付け位置に積み付ける．Step 2に戻る．

- **選択ルールを用いたCGPによる最適化手順** CGPは選択ルールの最適な実行順を求める関数同定問題を扱う．具体的に，CGPモデル(解個体)は現在の積み付け状態から計算可能な特徴量を入力とし，選択ルール毎に実行重要度を算出する関数を示す．そして，重要度の降順を同ルールの実行順とする．本稿では，文献(4)で用いた45個の特徴量にパレットごとの構成アイテム種類数の最小値・平均値・最大値を加える．既存の45個の特徴量については文献(4)を参照されたい．また，関数表現には四則演算，剰余，絶対値，総和，定数(0.0, 0.1, ..., 1.0, 10.0, -1.0)を用い，選択ルールは表1に示す8個を用いる．

CGPでは(1+4)-ESを用いてCGPモデルを生成する．

Table 1.: 選択ルール
Selection rules.

ID	Definition
$rule_1$	Select the block loaded by minimum pallet index. パレットのインデックスが最小となるブロックを選択する
$rule_2$	Select the block contains with the largest number of items. 構成するアイテム数が最大となるブロックを選択する
$rule_3$	Select the block with the shortest z coordinate when loaded. 積み付けた際に頭頂部 z 座標が最小となるブロックを選択する
$rule_4$	Select the block with the smallest height. ブロック単体の高さが最小となるブロックを選択する
$rule_5$	Select the block with the largest top area. ブロック単体の上面積が最大となるブロックを選択する
$rule_6$	Select the block with the largest volume. ブロック単体の体積が最大となるブロックを選択する
$rule_7$	Select the block with block pattern. 棒積みのブロックを選択する
$rule_8$	Select the block with pinwheel pattern. ピンホール積みのブロックを選択する

まず、初期解として5つのCGPモデルを生成し適合度を求める。ACLA-CGPにおける適合度は、CGPモデルを組み込んだ積み付けアルゴリズムから積み付けパターンを出力し、同パターンの目的関数値に設定する。そして、最良の適合度を持つ親CGPモデルを一つ選択し、同モデルから4つの子CGPモデルを生成する。この処理単位を1世代とし、最大世代数まで繰り返す。なお、2世代目以降では、親CGPモデルを含めた5つのモデルから最良の適合度を持つモデルを選択する。

3. i-ACLA-CGP

本章では、ACLA-CGPに対話型進化計算を融合したi-ACLA-CGPのメカニズムについて述べる。

〈3・1〉設計方針 i-ACLA-CGPは、ACLA-CGPにおけるCGPの適合度計算をユーザ評価に代替することで実現できる。具体的には、(1+4)-ESにおいて1つのCGPモデルを選択する過程をユーザが行うように拡張する。したがって、本稿ではユーザ評価の方法として単一選択方式を採用する。一般に対話型進化計算では、ユーザ評価の方法やそれを活用した解生成法について改良されることが多い⁽⁸⁾。加えて単一選択方式は最も基本的な評価方法の1つであり、ユーザから得られる定量的情報が少ない欠点がある。しかし次の2つの理由から、i-ACLA-CGPでは単一選択方式を用いてACLA-CGPと同じフレームワークを再現するように設計する。

- 単一選択方式によって実質的に(1+4)-ESと同等の最適化手順を行うことができ、1世代あたり5つのCGPモデルの比較で済む。したがって、ユーザによる積み付けパターンの比較評価が容易となり、ユーザの解評価作業の負担は主に世代数に依存することになる。
- ACLA-CGPの修正を最小限に抑えることで、ACLA-CGPの高い性能を維持したi-ACLA-CGPの構築が期待できる。つまり、少ない世代数で所望の積み付けア

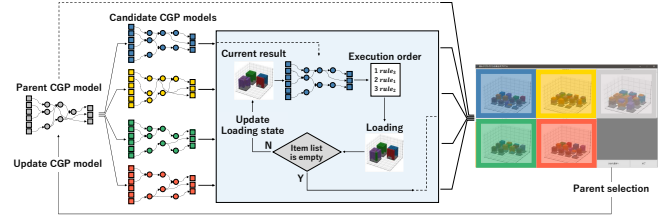


Fig. 2.: i-ACLA-CGP のフレームワーク
The framework of i-ACLA-CGP.

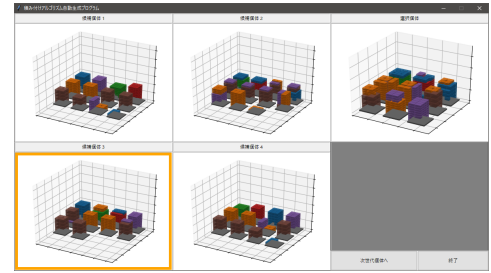


Fig. 3.: 設計した GUI
Designed GUI.

ルゴリズムを生成できることが期待される。

〈3・2〉メカニズム 図2にi-ACLA-CGPのフレームワークを示す。図に示すように、単一選択方式を組み込んだ(1+4)-ESを用いてCGPモデルを最適化する。具体的には、親CGPモデルと4つの子CGPモデルをそれぞれ積み付けアルゴリズムに組み込み、積み付けパターンを求める。これらの積み付けパターンを図3に示すGUIを通してユーザに提示する。次に、ユーザは1つの出力パターンを選択し、選択された出力パターンを導出したCGPモデルを次世代の親モデルとする。ここで、ユーザは5つの出力パターンから必ず1つ選択しなければならないが、前回選択した(親CGPモデルの)出力パターンを選択しても良い。ACLA-CGPと同様にこの処理単位を1世代とし、最大世代数 G まで繰り返す。

4. 実験

〈4・1〉実験設定 i-ACLA-CGPが導出した積み付けアルゴリズムがユーザの選好を適切に反映していることを評価するために、ACLA-CGPが導出した同アルゴリズムと比較可能な実験手順を設計する。まず、混載積み付け問題の設定を表2に示す。この問題設定に対し、ACLA-CGPとi-ACLA-CGPが構築する積み付けアルゴリズムの性能を比較する。次に、定式化可能な3つの目的関数 f_1, f_2, f_3 を以下の通り定義する。

f_1 : 使用パレット数の最小化

f_2 : パレットごとに積み付けるアイテム種類数の最小化

f_3 : ピンホール積みのブロック数の最小化

ここで、 f_1, f_3 は輸送コストの抑制、 f_2 は段ボール箱の検品

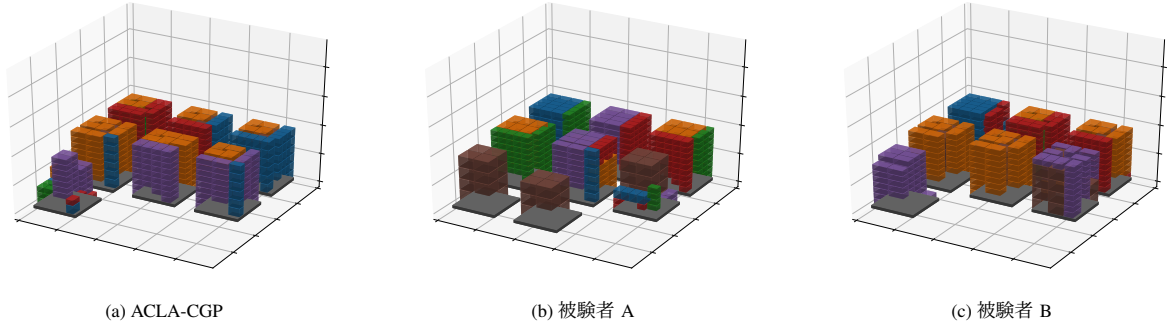


Fig. 4.: f_1 を主目的とした場合の積み付けパターン (50 世代目)

Loading patterns with main objective function f_1 at 50-th generation.

Table 2.: 積み付け問題の設定
Problem settings.

ID	Width [mm]	Depth [mm]	Height [mm]	Quantity
$item_1$	300	200	300	100
$item_2$	300	400	200	120
$item_3$	250	200	200	140
$item_4$	200	300	200	110
$item_5$	400	250	200	100
$item_6$	400	400	400	60
Total.				890

性の向上にそれぞれ配慮した積み付けパターンの設計を目的とする。なお、ピンホール積みは、積み付けた段ボール箱の安定性の向上に寄与するが、パレットごとの積載率が低下する。したがって、ピンホール積みのブロック数を削減することで積載率の向上が達成できる。

ACLA-CGP では上述の目的関数一つを設定することで、導出した積み付けアルゴリズムの性能を定量的に評価できる。一方で、i-ACLA-CGP の性能評価は被験者実験を通して行う。具体的に、被験者に定義した目的関数一つを与え、これを主目的（優先順 1 位）として積み付けパターンを選択するように指示する。加えて、他の目的関数や独自指標とそれらの優先順位を設定し、同パターンを選択可能であることを申し添える。例えば、主目的として f_1 を与えられた被験者は、 f_2, f_3 の観点も踏まえて評価しても良い。最大世代数に達した後、被験者が追加的に考慮した目的関数と独自指標の優先順位についてアンケート調査を行う。

主目的に設定した目的関数 f_* に対して、ACLA-CGP と、ならびに被験者実験を通して i-ACLA-CGP が導出した積み付けアルゴリズムの f_* の目的関数値を比較することで、主目的に対する性能評価を行う。独自指標については、定量的評価を行わず次章で定性的な分析を行う。加えて、ACLA-CGP が考慮しない他の目的関数に対しても同様に比較することで、i-ACLA-CGP がアンケートに従う積み付けアルゴリズム導出可能かを評価する。

被験者は 6 名 (A~F) とし、各目的関数を主目的とする計 3 ケースに 2 名ずつ割り当てる。また、最大世代数を 50 とし、CGP で用いるノードの行数 $n_r = 1$ 、列数 $n_c = 800$ 、(1 + 4)-ES で用いる突然変異確率を $\mu = 0.08$ とする。

〈4・2〉 実験結果 アンケート結果を表 3 に示す。同表では、主目的の目的関数を 1st、ユーザが追加的に考慮した目的関数や独自指標 (Remarks) を優先順位にしたがって 2nd, 3rd, 4th でそれぞれ示す。なお、被験者 C は 30 世代付近で f_1 と独自指標の優先順位が変化している。図 5 は、 f_1, f_2, f_3 をそれぞれ主目的とした場合の各目的関数の推移を示す。このアンケート結果と各目的関数の推移を比較し、導出した積み付けアルゴリズムの妥当性を議論する。また、積み付けパターンの例として、図 4 に ACLA-CGP ならびに被験者 A と B の調整によって i-ACLA-CGP が最終世代で導出した同パターンを示す。

図 5(a) より、 f_1 を主目的とする実験ケースについて、ACLA-CGP ならびに被験者 A, B を通して i-ACLA-CGP が導出した積み付けアルゴリズムは、 f_1 を削減する積み付けパターンを導出できていることがわかる。被験者 B の積み付けアルゴリズムは、 f_2, f_3 の目的関数値が ACLA-CGP よりも低い値を示していることから、被験者 B の選好が適切に反映されている。図 4(a), (c) より、被験者 B の積み付けパターンは、ACLA-CGP と同数のパレットを用いながらも同じ種類のアイテムを各パレットに集約していることがわかる。一方で、被験者 A の積み付けアルゴリズムは、優先順 2 位である f_2 が最適化されておらず、選好を十分に捉えられていない。これは、被験者 A の独自指標（視覚的に整理された積み付けを選択）やユーザ評価の不確定性が原因であると考えられる。被験者 A の積み付けパターン (図 4(b)) は、ACLA-CGP (図 4(a)) と比較して、異なる種類のアイテムの積み付け場所を把握できる意味で視認性が高く、独自指標を満たした積み付けパターンを生成できている。

図 5(b) より、 f_2 を主目的とする被験者 C と D のケースでも、図 5(a) と同様に f_2 を削減するように積み付けアルゴリズムが調整されている。被験者 D のケースでは優先順 3 位である f_3 についても同時に削減できており、ACLA-CGP では最終的に f_3 の目的関数値が大きく上昇している。したがって、i-ACLA-CGP は被験者 D の選好を反映した積み付けアルゴリズムが導出できていることがわかる。なお、被験者 C については 30 世代付近で選好の優先度に変更した

Table 3.: アンケート結果
Result of questionnaire.

Subject ID.	f_1	f_2	f_3	Remarks
A	1st	2nd	–	Select loading patterns well-organized visually (3rd). 視覚的に見やすく整理された積み付けパターンを選択
B	1st	2nd	3rd	Minimize the number of pallets having the largest number of types of items loaded (4th). 積み付けるアイテムの種類数が最大値をとるパレットの数の最小化
$C_{<30}$	3rd	1st	–	Minimize the number of pallets having the largest number of types of items loaded (2nd). 積み付けるアイテムの種類数が最大値をとるパレットの数の最小化
$C_{>30}$	2nd	1st	–	Minimize the number of pallets having the largest number of types of items loaded (3rd). 積み付けるアイテムの種類数が最大値をとるパレットの数の最小化
D	–	1st	3rd	Minimize the number of pallets having the largest number of types of items loaded (2nd). 積み付けるアイテムの種類数が最大値をとるパレットの数の最小化
E	2nd	3rd	1st	Minimize the height of loaded items (4th). 積み付けるアイテム数の高さの最小化
F	2nd	3rd	1st	–

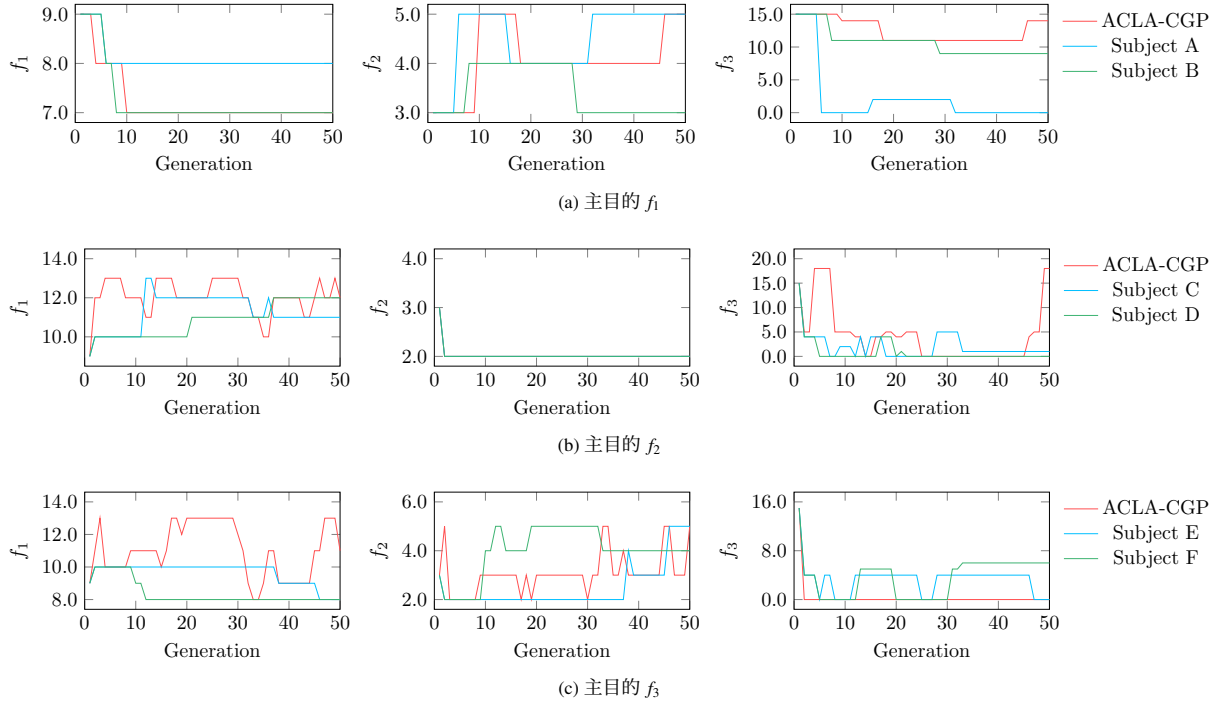


Fig. 5.: 各目的関数を主目的とした場合の目的関数値 f_1, f_2, f_3 の推移
Objective values of f_1, f_2, f_3 for each main objective function.

ことから、 f_1 の目的関数値が次第に増加している。この被験者 C の選好変化に対する積み付けアルゴリズムの調整結果については〈5・3〉節で分析する。

最後に、図 5(c) より、 f_3 を主目的とする被験者 E と F の積み付けアルゴリズムは、 f_3 の目的関数値が変動し収束していないが、優先順 2 位のルールである f_1 は ACLA-CGP と比べて適切に削減されている。特に、i-ACLA-CGP における f_3 の劣収束性に対しては、GUI における棒積みとピンホール積みのブロックの視認性の低さが原因と考えている。例えば、表 2 に示す $item_6$ は立方体であり、棒積みとピンホール積みの区別が困難となる。実際、被験者 F はピンホール積みのブロックを棒積みとして誤認したことを聞き取り調査から確認している。

以上より、i-ACLA-CGP はユーザの選好を満たす積み付けアルゴリズムを 50 世代の評価で対話的に生成できることがわかる。加えて、GUI の設計を再検討し、積み付けパターンの視認性を改善すれば、i-ACLA-CGP の有効性がさらに向上する可能性がある。

5. 積み付けアルゴリズムの挙動解析

最終世代で導出した積み付けアルゴリズムについて CGP モデルが導出した選択ルールの実行順に注目する。そして、この実行順とユーザの選好との関係を定性的に分析する。前章で導出した積み付けアルゴリズムが求めた選択ルールの実行順を表 4 に示す。〈2・2〉節で述べた通り、i-ACLA-CGP は積み付け状態ごとに選択順を算出しアイテムリス

Table 4.: 獲得した積み付けアルゴリズムにおける選択ルールの実行順

The order of selection rules calculated by obtained loading algorithms in Section 4.

ID.	Iteration	1st	2nd	3rd	4th	5th	6th	7th	8th
A	1 ~ 23	rule ₇	rule ₁	rule ₆	rule ₂	rule ₈	rule ₃	rule ₄	rule ₅
B	1	rule ₁	rule ₆	rule ₈	rule ₇	rule ₂	rule ₃	rule ₄	rule ₅
	2 ~ 39	rule ₁	rule ₈	rule ₇	rule ₂	rule ₃	rule ₄	rule ₅	rule ₆
C ₃₀	1 ~ 14	rule ₂	rule ₈	rule ₇	rule ₄	rule ₃	rule ₅	rule ₁	rule ₆
C ₅₀	1	rule ₆	rule ₁	rule ₄	rule ₇	rule ₂	rule ₃	rule ₅	rule ₈
	2 ~ 14	rule ₆	rule ₅	rule ₁	rule ₄	rule ₇	rule ₈	rule ₂	rule ₃
D	1 ~ 5	rule ₇	rule ₃	rule ₆	rule ₁	rule ₅	rule ₈	rule ₄	rule ₂
	6 ~ 14	rule ₇	rule ₃	rule ₆	rule ₁	rule ₅	rule ₈	rule ₂	rule ₄
E	1 ~ 4	rule ₁	rule ₆	rule ₃	rule ₅	rule ₄	rule ₇	rule ₂	rule ₈
	5 ~ 23	rule ₁	rule ₆	rule ₃	rule ₅	rule ₄	rule ₂	rule ₇	rule ₈
F	1	rule ₆	rule ₃	rule ₁	rule ₂	rule ₄	rule ₅	rule ₈	rule ₇
	2 ~ 30	rule ₁	rule ₄	rule ₆	rule ₃	rule ₅	rule ₂	rule ₈	rule ₇
	31 ~ 34	rule ₁	rule ₄	rule ₆	rule ₅	rule ₃	rule ₂	rule ₈	rule ₇

トが空になるまで繰り返す。同表では、この繰り返し回数 (Iteration) ごとの選択ルールの実行順を示している。以下では、ユーザの選好が特に正しく反映された被験者 D、ユーザの選好と乖離が見られた被験者 F、選好の変化が生じた被験者 C について述べる。

〈5・1〉 被験者 D：選好が正しく反映されたケース 被験者 D の選好順位は、1) パレットごとの構成アイテム種類数の最小化 (f_2)、2) 最大種類数で構成されているパレットの個数が少ない候補を選択、3) ピンホール積みのブロック数の最小化である。なお、被験者 D の主目的である f_2 の最適化は、同じ種類のアイテムを可能な限り集約し、かつ、異なるパレットに分散して積み付けることで達成される。表 4 より、被験者 D の積み付けアルゴリズムは $rule_7$ 、 $rule_3$ 、 $rule_6$ の実行順位が高い。 $rule_7$ で棒積みブロックを選択し、 $rule_3$ でパレットごとの積み付け高さを最小化することで、ピンホール積みのブロックを削減しつつ分散して積み付けるようなブロックの選択圧を実現している。さらに、体積が大きいブロックを選択する $rule_6$ は、同種類のアイテムを可能な限り集約するようなブロックの選択圧を生む。以上より、CGP によって選択ルールの実行順を最適化することで、ユーザの選好が適切に反映された積み付けアルゴリズムが構築できていることがわかる。

〈5・2〉 被験者 F：選好と乖離があるケース 被験者 F の積み付けアルゴリズムは、主目的である f_3 (ピンホール積みブロック数の最小化) よりも f_1 (使用パレット数の削減) の改善効果が高い点でユーザの選好と乖離がある。なお、 f_3 の改善には $rule_7$ (棒積みブロックの選択) が重要であり、 $rule_8$ (ピンホール積みブロックの選択) の実行順位は最も低くする必要がある。しかしながら、表 4 に示す通り、 $rule_7$ の実行順位は $rule_8$ よりも低い。また、繰り返し数に応じて $rule_6$ と $rule_1$ が実行順 1 位となる。 $rule_6$ は積載率の向上、 $rule_1$ は可能な限り同一のパレットに集約する

選択圧を生むことから、積み付けアルゴリズムは f_1 の改善を重視するように調整されていることがわかる。選好が特に正しく反映された被験者 E のケースでは、 $rule_7$ の実行順位が $rule_8$ よりも高いことから、被験者 F の場合は CGP によって最適化が適切に成されなかった可能性がある。

〈5・3〉 被験者 C：選好が変化するケース 被験者 C は 30 世代付近で f_1 と独自指標 (最大種類数で構成されているパレットの個数が少ない候補を選択) の優先順位が変化する点に特徴があった。この独自指標は、複数のパレットに分散して積み付けることが肝要であることから、 f_1 の目的とトレードオフが生じる。両目的に肝要な $rule_1$ と $rule_3$ に注目すると、30 世代目と 50 世代目で実行順位が変わる (表 4)。先述した通り、 $rule_3$ はブロックを分散して積み付けるが、 $rule_1$ は可能な限り同一のパレットに集約するように積み付ける。30 世代における積み付けアルゴリズムは、 $rule_3$ が先に適用されることで、被験者 C の独自指標を優先して反映している。50 世代では $rule_1$ を先に適用することで f_1 を優先的に最適化している。したがって、被験者 C のケースでは、i-ACLA-CGP がユーザの選好変化を受けて適切に積み付けアルゴリズムを調整していることがわかる。

6. 結 論

本稿では、対話型進化計算を組み込んだ積み付けアルゴリズムの自動生成技術を導入した。被験者実験では、ユーザによる解評価作業を 50 回に制限した設定で、提案法がユーザの選好を適切に反映した積み付けアルゴリズムを導出可能であることを示した。今後の予定として、生成した積み付けアルゴリズムを段ボール箱の種類や個数が異なる問題に転用することで、同アルゴリズムの汎用性を検証する。また、積み付けパターンの視認性を考慮した GUI の再設計を行い、ユーザ評価の不確定性を軽減する。

文 献

- (1) 厚生労働省. 令和元年版労働経済の分析—人手不足の下での「働き方」をめぐる課題について—, 2019.
- (2) Hideyuki Takagi. Interactive evolutionary computation. In *Proc. Int. Conf. Soft Comput. Inf./Intell. Syst.*, pages 41–50, 1998.
- (3) Julian F. Miller. *Cartesian Genetic Programming*, pages 17–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- (4) 蛭田悠介, 西原慧, 小熊祐司, 藤井正和, 和中田雅也. Cartesian genetic programming を用いた転移利用可能な積み付けアルゴリズムの自動生成. *研究報告数理モデル化と問題解決 (MPS)*, 2020(10):1–6, 2020.
- (5) Ying Zhang et al. Reduced human fatigue interactive evolutionary computation for micromachine design. In *2005 International Conference on Machine Learning and Cybernetics*, volume 9, pages 5666–5671. IEEE, 2005.
- (6) Chris Wijns, Fabio Boschetti, and Louis Moresi. Inverse modelling in geology by interactive evolutionary computation. *Journal of Structural Geology*, 25(10):1615–1621, 2003.
- (7) 日本工業規格. Jis z 0105 : 2015 包装貨物—包装モジュール寸法.
- (8) Hiroshi TAKENOUCI and Masataka TOKUMARU. Interactive evolutionary computation system with user gaze information. *International Journal of Affective Engineering*, 18(3):109–116, 2019.