

エリートアーカイブを有する遺伝的アルゴリズム

Genetic Algorithm with Elite Archive Mechanism

花田 良子

Yoshiko HANADA

Abstract: Conventional genetic algorithms(GA) have no possibilities of exhaustive search on all search space although they use infinite computations on large scale computer systems. In this paper, we propose GA with an elite archive for large scale computer systems. The proposed method has a database specialized GA which possess informations of space already searched, and the local search mechanism for the remaining space. As this mechanism, it gives quantitative ahold of the rate of searched region during search and ensures exhaustive search on all search space under infinite computations. We applicate this method for 3 deceptive function which is one of trap functions and it showed that this method ensure exhaustive search effectively.

1 はじめに

遺伝的アルゴリズム (Genetic Algorithms: GA) は多くの並列処理を必要とするが, 利用可能な計算資源が膨大でも, 並列処理する探索点の増加において, 有効な探索点を効率的に生成できなければ, 無限に計算を行ったとしても全探索する可能性はない. そこで, 広域計算環境のためのエリートアーカイブを有する GA を提案する. 提案手法は, 既探索領域を格納する GA に特化したデータベースにより, 全探索領域中どの領域を探索したかを記録し, 未探索領域において局所探索を行う. この仕組みにより, 探索中に得られた解がどの程度探索した結果であるか把握することが可能であり, 無限に計算を行った場合には全探索を保证する.

2 エリートアーカイブを有する GA

2.1 提案手法の操作

提案手法は, GA の探索母集団, およびローカルサーチのためのエリート個体群をもつ. 前者を探索アーカイブ, 後者をエリートアーカイブとよぶ. エリートアーカイブはエリート個体群と既探索領域を格納する. 提案手法では, 探索アーカイブにおいて大域探索, エリートアーカイブにおいて局所探索を行う. Fig. 1 に具体的な操作を示す.

まず, 探索アーカイブを初期化する. エリートアーカイブについては, 初期世代において個体数は 0 である. 次に探索アーカイブにおいて, 交叉および選択を行い, 探索アーカイブの上位 1 個体をエリートアーカイブにコピーする. ただし, すでにこの個体がエリートアーカイブに存在するときは格納しない. また, エリートアーカイブの個体数があらかじめ定めた個体数に達していた場合, エリートアーカイブの最悪個体の適合度よりも良ければ, 最悪個体と置き換える. 次にエリートアーカイブ

においてローカルサーチを行い, 適合度の良いものが発見された場合, その個体のコピーを探索アーカイブの最悪個体と置き換える.

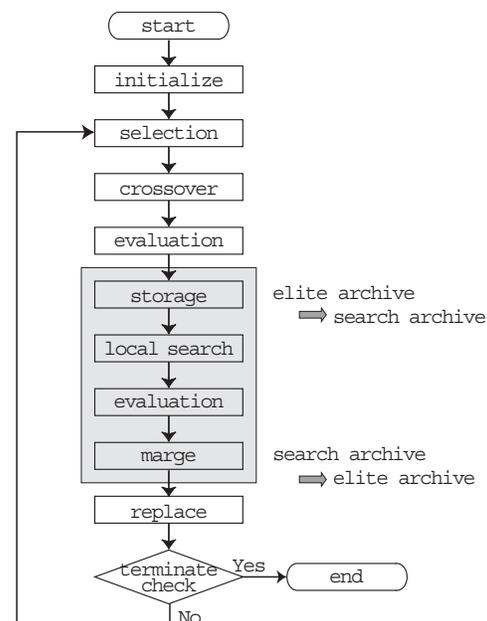


Fig. 1 提案手法の流れ

2.2 エリートアーカイブの個体表現

提案手法では, 個体の遺伝子表現に $\{0, 1\}$ のビットを用いている. エリートアーカイブの個体は, 染色体に加え, 既探索領域を示すマスクを持つ. マスクは染色体長と同じ長さのビット列にする. マスクの "1" は, 染色体においてマスクの "1" の部分に対応する遺伝子座について, すべてのパターンを探索したことを表す. 例えば, Fig. 2 に示すように, マスク "010011" の 6 ビットの個体 "110100" は, "1*01**" (*は任意) の 2^3 通り

をすべて探索し終えた上で最も適合度が高い解が”110100”であることを示す。マスクの”1”の個数をハミング指標と呼ぶ。この個体のハミング指標は3である。なお、探索アーカイブの個体がエリートアーカイブに新しく格納された場合、その個体のマスクはすべて”0”，ハミング指標は0である。

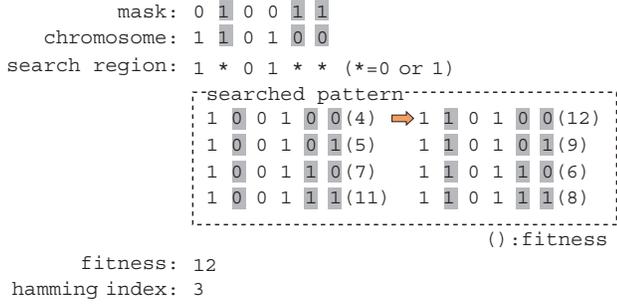


Fig. 2 エリートアーカイブの個体の例

次節に、ローカルサーチの手順を説明する。以降、染色体長を L ，エリートアーカイブに保存されている個体数を N_{elite} ，個体 x の染色体を $(c_{x1}, c_{x2}, \dots, c_{xL})$ ，マスクを $(m_{x1}, m_{x2}, \dots, m_{xL})$ と表現する。 c_{xl}, m_{xl} の $l(1 \leq l \leq L)$ を染色体，マスクそれぞれの遺伝子座とする。また、個体 x とハミング距離 h だけ異なる個体群を近傍 $N_h(x)$ とする。

2.3 エリートアーカイブにおけるローカルサーチ

提案手法では、毎世代、ある1個体にローカルサーチを適用する。対象個体にはエリートアーカイブの保存個体でハミング指標値がもっとも最小の個体(値が同じ時には適合度の最も良い個体)を選ぶ。以下にローカルサーチの手順を示す。

1. 式(1)に示すようにように個体 $x_i(1 \leq i \leq N_{elite})$ の各遺伝子座 $j(1 \leq j \leq L)$ の遺伝子の平均と0.5との差の絶対値 a_j を求める。

$$a_j = \left| \frac{1}{N_{elite}} \sum_{i=1}^{N_{elite}} (c_{x_{ij}}) - 0.5 \right| \quad (1 \leq j \leq L) \quad (1)$$

2. 各遺伝子座 j を $a_j(1 \leq j \leq L)$ の昇順に並び替える。 l_k を k 番目に小さい遺伝子座とする。
3. N_{elite} 個の個体で最もハミング指標値が最小の個体 x を初期探索点とする。
4. <ステップ k > 探索点 x について、 $m_{xl_k} = 0$ の場合、 c_{xl_k} のビットを反転する。 $m_{xl_k} = 1$ の場合、 $k = k + 1$ に更新し、7. に進む。
5. $m_{xj}(1 \leq j \leq L)$ の中で、 $m_{xj} = 0$ の c_{xj} を固定したまま、 $m_{xj} = 1$ の c_{xj} について、近傍 $N_h(x)(1 \leq$

$h \leq h_x)$ を生成する。ただし、 h_x は探索点 x のハミング指標とする。総近傍個体数は $2^{h_x} - 1$ である。

6. x と近傍 $N_h(x)(1 \leq h \leq h_x)$ の中で最も適合度が高い解を探索点 x と置き換える。 $m_{xl_k} = 1, k = k + 1, h_x = h_x + 1$ に更新する。
7. 探索初期点よりも適合度の高い解が得られるか、あるいは k が L に達したら8.へ、そうでなければ4.へ戻る。
8. 探索初期点よりも適合度の高い解が得られた場合、その個体を探索アーカイブの最悪個体に上書きコピーする。 k が L に達したなら、全探索が完了したので、探索アーカイブ、エリートアーカイブにおける探索を終了する。

Fig. 3に6ビットの個体、 $N_{elite}=4$ のローカルサーチの例を示す。Fig. 3では、エリートアーカイブ内の個体でハミング指標値が最小、適合度が最大の個体 gene2 についてローカルサーチの操作を適用する。ローカルサーチを適用する遺伝子座の順は式(1)に示す a_j の昇順で、左から1ビット目 3ビット目 6ビット目 2ビット目 4ビット目 5ビット目(3,6の順および2,4,5の順は任意)である。gene2をも見ると1ビット目はすでに探索済みであるので3ビット目についてローカルサーチを行う。

まず、gene2の染色体”100110”の3ビット目を反転させ、”101110”にする。次に、マスクが”0”の部分に対応する2,3,4,6ビット目の4ビットを固定し、マスクが”1”の部分に対応する1,5ビット目の2ビットについて00,01,10,11の 2^2 通りに変化させた個体をすべてチェックする。これより既にチェックしている”*001*0”の 2^2 通りと新たにチェックした”*011*0”の 2^2 通りで、”*0*1*0”の合計 2^3 通りチェックすることになる。これよりマスク”100010”の3ビット目を1に変更し(”101010”),ハミング指標値を3に更新する。

同様の操作を6ビット目 2ビット目 4ビット目の順に適用する。操作の途中でローカルサーチ前のgene2よりも適合度が高い個体を得られたなら、gene2のコピーを探索アーカイブの最悪の個体と置き換え、ローカルサーチを終了する。Fig. 3の例では、もとのgene2よりも、gene2(new)の方が適合度が高いので、3ビット目における操作が終了した時点でローカルサーチを終了する。

2.4 エリートアーカイブにおける個体のマージ

エリートアーカイブにおいて重複した探索を避けるため、毎世代、ローカルサーチ後に、マージが可能な個体の組合せがあれば適用する。マージが可能な組合せには次のようなものを考えている。

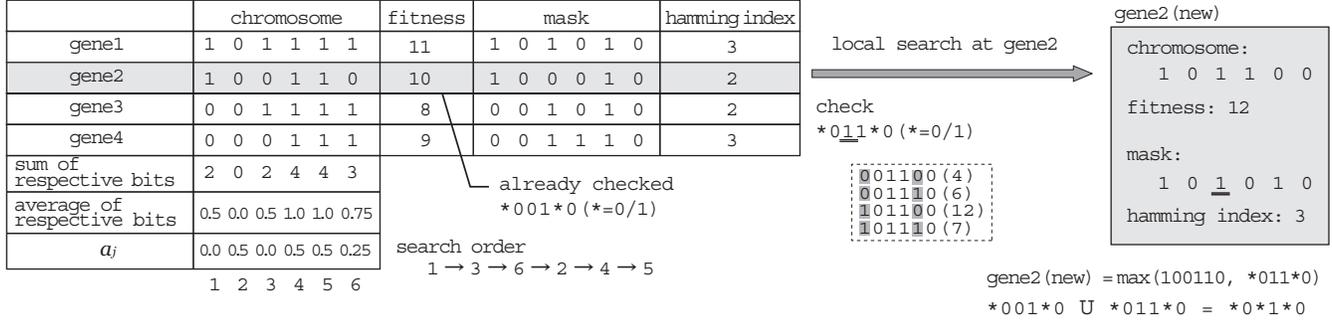


Fig. 3 6ビットの個体, $N_{elite}=4$ の例

<条件 1 >

個体 x_a, x_b のマスクが同一で、マスクの”0”の部分に対応する染色体についての遺伝子座においてハミング距離が1の場合、 $m_{x_a l} = m_{x_b l} = 0$ で、 $c_{x_a l} \neq c_{x_b l}$ となる遺伝子座を l とすると、 x_a, x_b の適合度の高い個体のマスクの遺伝子座 l のビットを1にし、もう一方の個体を削除する。

Fig. 3における gene1 と gene2(new) はマージ可能であり、Fig. 4に示すように gene2(new) のマスクを”1 0 1 0 1 1”とし、エリートアーカイブから gene1 を削除する。

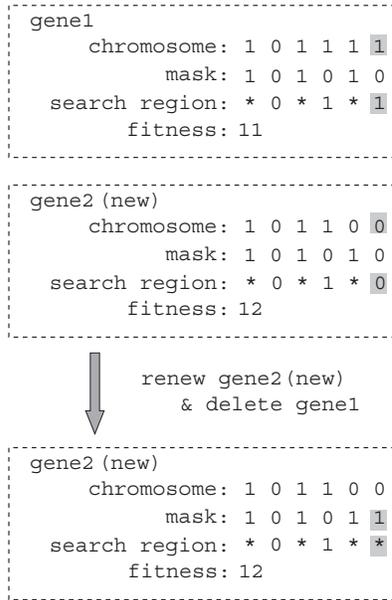


Fig. 4 <条件 1 > のマージの例

<条件 2 >

個体 x_a, x_b について $m_{x_a l} = 1, m_{x_b l} = 0 (1 \leq l \leq L)$ となる遺伝子座が存在せず、 x_b のマスクの”0”の部分に対応する染色体についての遺伝子座においてハミング距離が0となる場合、個体 x_b の既探索

領域は個体 x_a の既探索領域を包含しているので個体 x_a を削除する。

Fig. 3における gene3 と gene4 は Fig. 5 から分かるようにマージ可能であり、エリートアーカイブから gene3 を削除する。

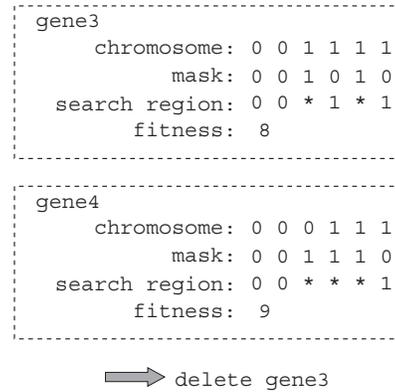


Fig. 5 <条件 2 > のマージの例

また、エリートアーカイブ内で染色体が同一の個体についてもマージ可能になるまで以下のようなローカルサーチを行う。以下の操作は Fig. 6 に示すように、個体 x_1 から x_2 についてローカルサーチを行い、上記のマージの<条件 2 >を適用している。ただし、 x_1 から x_2 へのローカルサーチの適用中に x_1 がより良好な適合度の解に更新されたなら、 x_1 をマージの対象から除外する。

1. 染色体が同一の個体を抽出する。これらの個体の集合を M とする。
2. ハミング指標が1番目に小さい個体 x_1 と2番目に小さい個体 x_2 を探す。
3. x_1, x_2 について、 $m_{x_1 l} = 0, m_{x_2 l} = 1 (1 \leq l \leq L)$ となる遺伝子座 l を1つ探し、 $c_{x_1 l}$ のビットを反転する。見つからなければ6.に進む。

4. $m_{x_{1j}}(1 \leq j \leq L)$ の中で, $m_{x_{1j}} = 0$ の $c_{x_{1j}}$ を固定したまま, $m_{x_{1j}} = 1$ の $c_{x_{1j}}$ について, 近傍 $N_h(x_1)(1 \leq h \leq h_{x_1})$ を生成する. ただし, h_{x_1} は x_1 のハミング指標とする.
5. $m_{x_{1l}} = 1, h_{x_1} = h_{x_1} + 1$ に更新する. $N_h(x_1)(1 \leq h \leq h_{x_1} - 1)$ の中で, x_1 よりも適合度の高い個体が見つければ, x_1 の染色体をその個体の染色体に更新し, Mから x_1 を削除し, 7.に進む. 見つからなければ 3.に戻る.
6. x_2 を Mおよびエリートアーカイブから削除する.
7. Mのサイズが 1の場合, 終了. そうでなければ 2.に戻る.

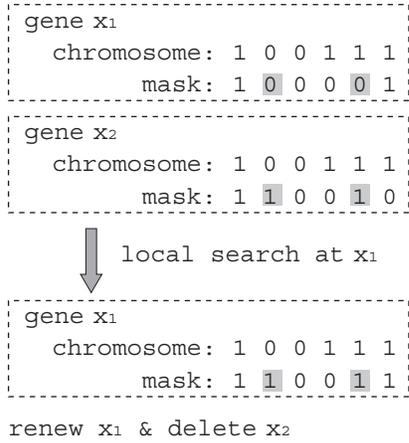


Fig. 6 同一個体のマージの例

3 既探索領域の定量的な把握

提案手法では, 毎世代, 既探索領域の大きさを定量的に示すことができる. 本節では, その求め方を説明する.

3.1 1個体の既探索領域の大きさ

個体 x の既探索領域 X の大きさは $|X|$ で表すことができる. $|X|$ は集合 X の要素数を示す. 例えば, x をマスク"010011"の6ビットの個体"110100"の個体とすると, x の既探索領域 X は"1*01**"であり, $X = \{1*01**\}$ の大きさは要素数 $|X| = 2^3$ 個である. ハミング指標値が h_x の個体の既探索領域のサイズは 2^{h_x} となる.

3.2 複数の個体の既探索領域の大きさ

エリートアーカイブの1つ個体 $x_i(1 \leq i \leq N_{elite})$ の既探索領域を X_i とすると, エリートアーカイブ全体の既探索領域は, エリートアーカイブの個体全体の和集合 $X_1 \cup X_2 \cup \dots \cup X_{N_{elite}}$ である. 添え字 i の集合を $I = \{1, 2, \dots, N_{elite}\}$ とすると, エリートアーカイブ全体の既探索領域の大きさは $|\bigcup_{i \in I} X_i|$ と記述でき, 式(2)で求

めることができる¹⁾. 式(2)では, 偶数個の個体の組合せでできる積集合の要素数は減算し, 奇数個の個体の組合せでできる積集合の要素数は加算している.

$$\left| \bigcup_{i \in I} X_i \right| = \sum_{J \subseteq I, J \neq \emptyset} (-1)^{|J|-1} \left| \bigcap_{j \in J} X_j \right| \quad (2)$$

一般に, 対象集合族の和集合の個数関数は求めにくい場合が多く, 積集合の個数関数は具体的な閉じた式となる場合が多い¹⁾. 個体の既探索領域もこれにあたり, N 個体の既探索領域の和集合の要素数が一定の規則で求められないのに対し, 積集合の要素数は次のような規則が存在する. このことから, 直接, 和集合の要素数をもとめるのではなく式(2)で示したように積集合から求める.

2個体 x_i, x_j についてマスクを考慮した距離 $d(x_i, x_j)$ を式(3)のように定義する.

$$d(x_i, x_j) = \sum_{l=1}^L B |c_{x_{il}} - c_{x_{jl}}| \quad (3)$$

$$B = \begin{cases} 1, & \text{if } m_{x_{il}} = m_{x_{jl}} = 0 \\ 0, & \text{otherwise} \end{cases}$$

N 個体の既探索領域の積集合の要素数は式(4)のように求めることができる. なお, $[R]$ は, R を実数, z を整数としたとき, $z \leq R < z + 1$ ならば, $[R] = z$ を意味する.

$$|X_1 \cap X_2 \cap \dots \cap X_N| = \begin{cases} 2^M, & \text{if } K = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$K = \sum_{i=1}^{N-1} \sum_{j=(i+1)}^N d(x_i, x_j) \quad (4)$$

$$M = \sum_{l=1}^L \left[\frac{1}{N} \sum_{i=1}^N m_{x_{il}} \right]$$

4 提案手法の探索性能の確認

だまし問題の1つである3-deceptive問題²⁾について, 提案手法の解探索性能を確認する. 問題サイズは30ビットとした. 最適解は10である. Fig. 7に最良解の適合度(図中, 左軸, 実線), および全探索領域に対する既探索領域の割合(図中, 右軸, 点線)の推移を示す. 図中, グレーの部分はエリートアーカイブにおけるローカルサーチの結果, 白い部分は探索アーカイブの探索結果である. また, Fig. 8にエリートアーカイブの個体数の推移(左軸, 黒実線), およびハミング指標値の最小値(右軸, グレー実線), 平均値(右軸, 点線)を示す. これは1試行の結果である. なお, 探索アーカイブサイズを10, エリートアーカイブの個体数を3とした. 探索アーカイブにおける世代交代モデルはCCM³⁾, 交叉には一様交叉を用いた.

Fig. 8 から, 4, 5 世代あたりまでハミング指標の最小値が振動していることから, 探索アーカイブで発見された新しい個体がエリートアーカイブを更新していることがわかる. しかし 5, 6 世代にかけてエリートアーカイブの個体数が 1 に変化し, その後も増加していないことから, 探索アーカイブから新たに個体がエリートアーカイブに入っていないことがわかる. これは探索後半において探索アーカイブが収束しているためと考えられる. また, Fig. 7 の既探索領域の結果をみると, 5, 6 世代付近から既探索領域が著しく増加している. これは, 探索アーカイブにおいて有効な探索は行われておらず, エリートアーカイブの 1 個体を中心に集中的にローカルサーチが行われているためである.

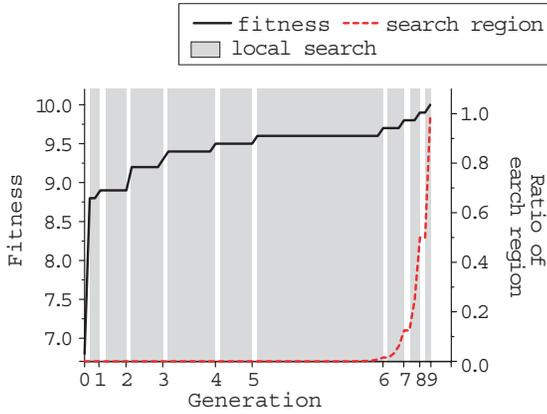


Fig. 7 適合度と既探索領域の割合の推移

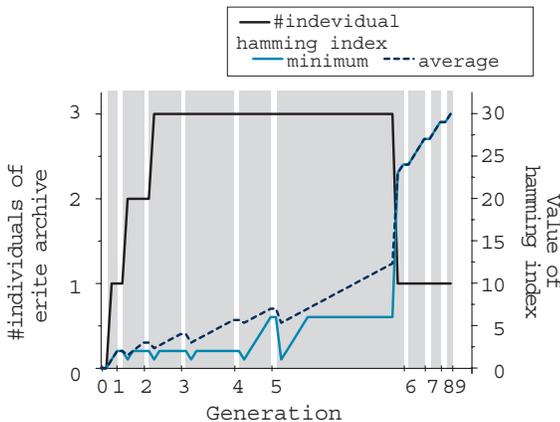


Fig. 8 エリートアーカイブの個体数とハミング指標値の推移

通常の GA で, 個体を増加させる, あるいは複数試行により全探索空間のサイズと同数の 2^{30} 個体の評価計算を行ったとしても最適解を得る保証はない. 一方で, 本手法では, Fig. 7 に示すように, 既探索領域が増加するにしたがって解が更新され, 全探索領域を探索し終えたなら, 得られた解が大域的な最適解である保証がある. また, 点線で示すように, 各世代において探索中の既探

索領域の定量的な把握が可能である. しかしながら, 本手法では既探索領域, すなわち計算量は指数的に増加するため, 世代が進むに従って計算負荷が高くなる. また, 探索が進むに従って探索アーカイブは収束し, 探索後半においては探索アーカイブの解探索能力は期待できない. これについては, 広域計算環境下での実装, および探索アーカイブの未探索領域への再初期化等のメカニズムを取り入れる必要がある.

5 おわりに

本論文では, 広域計算環境のためのエリートアーカイブを有する GA を提案した. 提案手法は既探索領域の定量的な把握により未探索領域を探索し, 無限に探索した場合, 必ず最適解が得られる仕組みを有する. 一方で, ローカルサーチにおける計算負荷が非常に大きい. 今後, 広域計算環境下での実装, および探索アーカイブの未探索領域への再初期化等のメカニズムの付加を行う予定である.

Appendix

3 個体 x_1, x_2 および x_3 の既探索領域の大きさ $|X_1 \cup X_2 \cup X_3|$ を求めた例, および Fig. 9 に $X_1 \cup X_2 \cup X_3$ を図示したものを示す.

< 例 >

$$X_1 = \{ * 0 * 1 * 1 \}, \quad |X_1| = 2^3$$

$$X_2 = \{ * 0 * 1 1 * \}, \quad |X_2| = 2^3$$

$$X_3 = \{ * * 0 1 * 1 \}, \quad |X_3| = 2^3$$

- 2 個体の積集合およびその要素数

$$X_1 \cap X_2 = \{ * 0 * 1 1 1 \}, \quad |X_1 \cap X_2| = 2^2$$

$$X_2 \cap X_3 = \{ * 0 0 1 1 1 \}, \quad |X_2 \cap X_3| = 2^1$$

$$X_3 \cap X_1 = \{ * 0 0 1 * 1 \}, \quad |X_3 \cap X_1| = 2^2$$

- 3 個体の積集合およびその要素数

$$X_1 \cap X_2 \cap X_3 = \{ * 0 0 1 1 1 \}$$

$$|X_1 \cap X_2 \cap X_3| = 2^1$$

- 3 個体の和集合の要素数

$$|X_1 \cup X_2 \cup X_3| = |X_1| + |X_2| + |X_3|$$

$$- |X_1 \cap X_2|$$

$$- |X_2 \cap X_3|$$

$$- |X_3 \cap X_1|$$

$$+ |X_1 \cap X_2 \cap X_3|$$

$$= 24 - 10 + 2 = 16$$

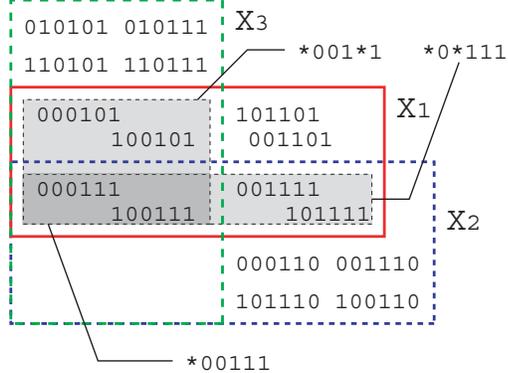


Fig. 9 3個体の既探索領域の和集合の例

参考文献

- 1) 成嶋弘日評数学選書 数え上げ組合せ論入門, 日本評論社, 1996
- 2) Martin Pelikan, David E. Goldberg, Erick Cantu-Paz. BOA:The Bayesian Optimization Algorithm. IlliGAL Report No. 99003 1999.
- 3) Isao Ono . A Genetic Algorithm Taking Account of Characteristics Preservation for Job Shop Scheduling Problems . Intelligent Autonomous System 5 . pp.711-718 , 2000