

要旨

本問題はある与えられた出口を持つ入れ物（コンテナ）に入れられたいくつかの形状物体（ブロック）を効率的に出口から出す問題である。

ここで、ブロックは自律的に動きを決定し、より多くのブロックが出口から出ることができるようふるまう。そのためにここでは、GAを利用したアプローチを試みる。

1. はじめに

テーキングブロックアウト問題とは、ある与えられた入れ物としての空間内に収められたいくつかの形状物体（ブロック）を入れ物に設定された出口から効率的に取り出す問題である。入れ物および出口の属性、ブロックの大きさ、形状そのものおよび個数等により様々な問題の設定が考えられる。ここで、すべてのブロックが出れるとは限らないため、できるだけ多くのブロックが短時間出ること（以後脱出と呼ぶ）ができるように、最適化を行う必要がでてくる。よって、本問題は空間の認識を伴う組み合わせ最適化問題で、解の探索が困難な問題を含んでいる。たとえば、ブロックの脱出の可否、脱出順序の決定、脱出経路の決定などの問題を内部に持っている。

本報告では、2次元のテーキングブロックアウト問題を扱い、ブロックが脱出できない場合の詰め直し作業に重点をおいて、これに対してGA的アプローチを適用するための方法論を示す。なお、ここでは時間軸は考慮しない。

2. 問題の設定

ここで、対象とする問題の設定を行う。入れ物は1つとし、出口は任意の場所に1ヵ所存在するものとする。また、内部に収まるブロックは長方形とし、座標軸に直交するものとする。（図1）。

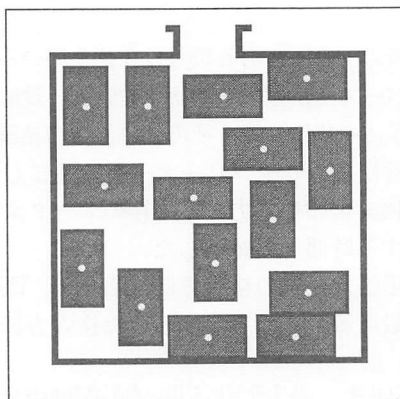


図1 入れ物とブロック

入れ物（以後コンテナと呼ぶ）、出口及びブロックはすべてその寸法と位置情報のみを持ち、脱出の優先順位等の他の属性は持たないとする。コンテナとブロックの幾何情報 DC 、 DB は次式により記述される。

$$DC = (W, H, Exit) \quad (1)$$

$$Exit = (E, w) \quad (2)$$

$$DB = \{ db_j; j=1, NB \} \quad (3)$$

$$db_j = (G_j, w_j, h_j) \quad (4)$$

ここで、コンテナは幅と高さ W 、 H からなる長方形状で、中心位置 E と幅 w によって示される出口 $Exit$ 付きとする。各ブロックはその重心位置 G_j と幅、高さ w_j, h_j によって示されテーキング処理により G_j が刻々と変わることになる。また、ブロック全体の個数は NB で与えられる。

本問題では脱出ブロックの個数を最大にするためのアプローチを行う。

3. ブロック脱出のアプローチ

出口に近いブロックが最も先に脱出する可能性が高いことは人間が行うときのヒューリスティックな知識により、脱出順序を決定する関数に利用できる。すなわち、

$$b^* = \min_{i \in IB} [BD_i] \quad (5)$$

$$BD_i = (E_x - G_{xj})^2 + (E_y - G_{yj})^2 \quad (6)$$

ここで、 $(E_x - G_{xj})$ 、 $(E_y - G_{yj})$ は出口中心とブロック重心の距離である。また、 IB は各ブロック番号の集合である。ここで、決定されたブロックが脱出できない場合、以下のアプローチにより脱出可能なブロックを見つけるための詰め直し作業を行う。すなわち、コンテナ出口にブロックの回転も含めた十分な移動空間を作り出すことにより脱出を容易にしようとするものである。

- ① 脱出ブロックが存在するかをチェック
- ② 存在すればそれを脱出させ、存在しなくなれば詰め直しを行う。
- ③ 上記作業を繰り返し、詰め直し後も脱出ブロックが存在しなければ、作業を終了する。

3.1 詰め直し作業のGA的アプローチ

コンテナの出口付近に空間を作り出すためにブロックの移動プランをGAにより生成する。ブロックを出口から遠くしかもむだな空間が少なくなるように詰め直しを行うため、回転を含んだ移動情報をストリング中に構築する。

目的関数を次式で与える。

$$f(S_j) = \sum_i d(E, P(S_{ij}, T_{ij})) \quad (7)$$

ここで、 E は出口中心位置で $P(S_{ij}, T_{ij})$ は移動プラン後のブロックの位置情報である。すなわち、ブロックの詰め直し具合を出口中心とブロックの重心位置の距離の総和によって表現する。ここでは、移動は必ずX、Y軸に直行するような姿勢で行われるとする。

3.2 GAのインプリメント

ここでは標準的な再生、交叉、突然変異の3種のオペレータを利用した単純GAをインプリメントする。移動プランを獲得するため1本のストリング bs_i を次のように与える。

$$bs_i = \{(BlkNo, Rot, Xmov, Ymov)_j; j=1, NB\} \quad (8)$$

ここで、 $BlkNo$ はブロックにあてられた番号、 Rot は回転、 $Xmov$ はX方向の移動、 $Ymov$ はY方向の移動を表わすビット列で、必要なビット数が確保される。これがブロック数 NB 分集まって1本のストリングが構成される。このストリングのランダムな集まりを初期集団とし、世代をおうごとに集団が進化し、より良い集団をプランとして獲得する。

3.2.1 遺伝子型と表現型の連携

前述のいわゆる遺伝子のストリングはあくまでどのブロックが回転し、X、Y方向に移動するかという情報のみ表現しているため、実際の問題空間において回転が可能かどうか、どの程度移動すればよいか等を表現し、遺伝子の評価を行うための表現型を記述する必要がある。

ブロック番号を翻訳する際、ビット情報そのものを扱うといわゆる致死遺伝子が発生し、同じブロックばかりが移動プランに発生してくる可能性がある。そこで、ブロック番号の遺伝子デコードにおいてTSPでよく用いられている基準順序とストリングから得られた順序からブロック番号を導く方法¹⁾を採用する。

ブロックの回転、移動は次のステップで行う。

Step-1) 初期状態を設定。

Step-2) S_{ij} に従って回転、移動

- ① 移動ステップ幅を設定
- ② X方向移動ビットがたっているとき
X方向にぶつかるとまで（他のブロックあるいはコンテナの壁に）1ステップ移動
- ③ Y方向移動ビットがたっているとき

Y方向にぶつかるとまで（他のブロックあるいはコンテナの壁に）1ステップ移動

④ 回転ビットがたっているとき

回転するための空間があれば回転する

Step-3) 移動後の状態を設定

Step-4) すべてのブロックが移動した後

各ブロックの重心とコンテナ出口の中心との距離を算出

ここで、Step-2の②③④のすべての条件が成り立つときは②③④の順で行うが、回転は移動の各ステップにおいて回転可能になった時点で回転する。これらの様相を図2に示す。

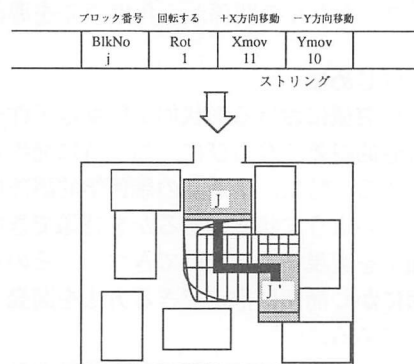


図2 遺伝子型と表現型

4. 計算機実験

上述の問題設定、方法論に従ってシミュレーションシステムを作成し、ここでは特に詰め直しの計算機実験を行った。詰め直し後のコンテナ内のブロックの状態を図3に示す。ブロック数は15、集団数50、再生、交叉、突然変異の各率は0.6、0.8、0.02で行った。コンテナ形状は正方形、ブロック形状は長方形に限定し、出口はコンテナ上部1カ所とした。

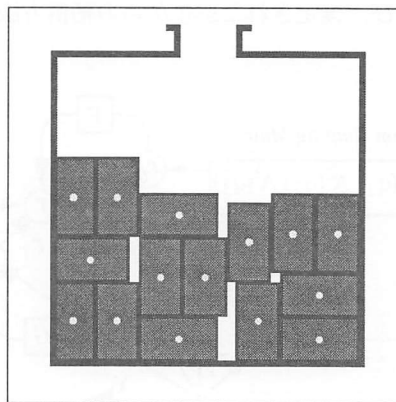


図3 詰め直しの結果

5. おわりに

本報では、脱出そのものの方法論については詳しく触れていないが、脱出をより可能にするための詰め直しの方法論を示した。しかし、ランダムな形状のブロック、任意の出口およびコンテナ形状についての方法論は今後の課題である。

参考文献

- 1) 遺伝的アルゴリズム、北野宏明編著、産業図書