テクニカルノート

CP-PACSの遺伝的アルゴリズムを用いたレジスタ割付

中家鉄雄[†]　秡川友宏[††,☆]　山下義行[†††]

筑波大学計算物理学研究センターの CP-PACS は，高速演算のためにスライドレジスタという特殊な機構を持ち，そのレジスタ割付は巡回セールスマン問題に類似した性質を持つことが関連研究より分かっている．そこで，本論文では巡回セールスマン問題の有効な解法である遺伝的アルゴリズムを用いてレジスタ割付を行う具体的な方法を提案する．LFK のベンチマークから得られた例題を用いた実験の結果，すべてについて最適なレジスタ割付を得ることができた．

Register Allocation for the CP-PACS by Genetic Algorithm

Tetsuo Nakaie,[†] Tomohiro Haraikawa[††,☆]
and Yoshiyuki Yamashita[†††]

Slide-windowed register architecutre of the CP-PACS computer at the University of Tsukuba provides a new register renaming facility. In this paper the register allocation problem for this architecture is solved by genetic algorithm (GA). Since our problem is translated into the k-person Traveling Salesmen Problem (k-TSP) straightforwardly , the technique solivng the TSP by GA efficiently is applicable to our problem. Experiments show that GA finds optimal solutions for all the examples in Livermore Fortran Kernel.

1. はじめに

筑波大学計算物理学研究センターの超並列計算機 CP-PACS は，高速演算のためスライドレジスタという特殊機構を採用している．このため，従来のアーキテクチャに対するレジスタ割付法を単純に適用することができず，なんらかの解決策が必要となる．CP-PACS のこのレジスタを考慮したヒューリスティック解法[1)] も提案されたが，それらは結果としてしばしば最適解とはかけ離れた割付を提示することがある．そこで，大域的最適化手法である遺伝的アルゴリズム（GA: Genetic Algorithm）を用いることでこれらの問題を解消できると考え，CP-PACS のスライドレジスタ割付の最適化を試みた．このスライドレジスタ割付の研究過程で，この問題が巡回セールスマン問題（TSP: Traveling Salesman Problem）に類似した性質を持つことが分かっており[2)]，TSP において有効な手段とされる GA をこの問題に用いることは効果的であると考えられる．今回，Livermore Fortran Kernel（LFK）のベンチマークから得られた例題を用いた実験の結果，すべてについて最適なレジスタ割付が得られた．

以下では 2 章で，CP-PACS においてのレジスタ割付について概説する．3 章で本研究での遺伝子の構造および世代交代時の選択，交叉，突然変異について述べる．4 章で LFK を用いた実験の結果と，他のヒューリスティック解法との比較を述べる．

2. CP-PACS のレジスタ割付

CP-PACS は，高速演算のためにスライドレジスタ（スライドウィンドウ・アーキテクチャ[3)]）に基づく特殊なレジスタセットを持っている．このアーキテクチャは，レジスタ #0, #1, ··· の内容を，1 命令でレジスタ #(0+K), #(1+K), ··· に移すことができる（K は整数）．このレジスタの表現方法として**Spiral Graph**[3)] が用いられる．Spiral Graph は，横軸に命令ステップ番号をとったらせん状のグラフである（**図 1**）．このらせんを便宜上トラックと呼ぶ．このトラック上にそ

† 筑波大学理工学研究科
Master's Program in Science and Engineering, University of Tsukuba
†† 筑波大学工学研究科
Doctoral Program in Engineering, University of Tsukuba
☆ 現在，静岡大学情報学部
Presently with Faculty of Information, Shizuoka University
††† 筑波大学機能工学系
Institute of Information Sciences and Electronics, University of Tsukuba

```
    do {
0:   load   v1
1:   add    v6, v1 -> v4
2:   mult   v4, v4 -> v5
3:   add    v5, v1 -> v2
4:   mult   v1, v2 -> v3
5:   add    v2, v3 -> v6
6:   store  v2
     slide +1
    }
```

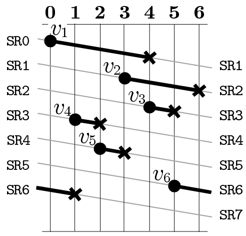

図 1 Spiral Graph
Fig. 1 Spiral Graph.

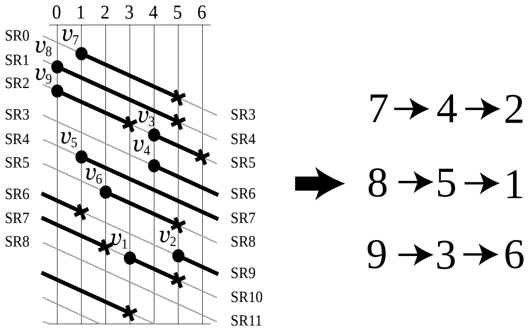

図 2 遺伝子構造
Fig. 2 Structure of gene.

れぞれの変数がどの命令からどの命令までの間レジスタに保持されるかを図 1 のように線分で表現する．たとえば，図 1 の変数 v_1 は，0 番の load 命令から 4 番の add 命令まで使用するのでその区間を線分で示している．この区間のことを**ライブレンジ**という．各ライブレンジをお互いが重なり合わないように配置していく．このときのトラックの総周回数が使用するレジスタの個数となる．しかし，図 1 のように配置するのでは隙間が多くできてしまい，無駄にトラックの総周回数が増えてしまう．このためどのような順番で隙間なく配置するかが問題となる．

図 1 は，スライド数が 1 なので 1 トラックとなっているが，実際には**図 2** 左のように複数スライドするため，複数のトラックを扱うことになる．複数のトラックにおけるレジスタ割付は次の 2 つの部分問題からなる．

(A) 各ライブレンジをいずれかのトラック上に分配する
(B) ライブレンジが分配されたトラックごとに順列を定める

これは，k 人のセールスマンが分担して都市を巡回する巡回セールスマン問題 ***k*-person Traveling Salesmen Problem** (k-TSP)[4] の

(a) 各都市をいずれかのセールスマンに分配する
(b) 各セールスマンごとに都市の訪問順を決める

という部分問題にそれぞれ対応している．よって，これ以降は，スライドレジスタ割付問題を k-TSP と関連づけて扱うこととする．ただし，ここで扱うスライドレジスタ割付問題は以下の 2 点で通常の k-TSP と異なる．1 つは各トラック上に最初に配置するライブレンジはロード命令で生成されるライブレンジでなければならない[1] ため，最初に訪問する都市群が一意に定まる．もう 1 つはライブレンジ間の隙間，すなわち都市間の距離が非対称であることである．これは，図 1 おいて v_1 と v_2 の隙間は 6 であるが，v_2 と v_1 と並べると隙間は 1 となることより明らかである．

3. GA によるスライドレジスタ割付問題の解法

3.1 GA

遺伝的アルゴリズム，GA とは，まず問題の解を導き出すための情報を数列，もしくは文字列として表した**遺伝子配列**とその優劣の度合いを表す**適応度**を持つ**個体**の集まりである**個体群**を形成する．その中で個体に，**選択**，**交叉**，**突然変異**などの操作を繰り返し適用させることで最適解を探索するアルゴリズムである．この GA では問題によって遺伝子配列と適応度を設計し，それに最も合う選択，交叉，突然変異の方法を選ぶことが重要となる．これより，ここで用いる遺伝子配列の構造，適応度と，選択，交叉，突然変異の各操作について説明する．

3.2 遺伝子配列の構造

スライドレジスタ割付問題では，ライブレンジの識別番号，そのライブレンジをどのトラックの何番目に格納するかという情報を遺伝子配列に組み込む必要がある．このため，図 2 右のような遺伝子配列を設計する．図 2 左の Spiral Graph は，3 トラックの場合におけるレジスタ割付例を示している．右の遺伝子配列と左の Spiral Graph とは対応しており，遺伝子配列の数列は Spiral Graph からトラックごとにライブレンジの識別番号を抜き出し，並べたものとなる．この遺伝子配列の情報から Spiral Graph への各ライブレンジの割付を得るが，それは k-TSP における各セールスマンが通る巡回路としてとらえることができる．

3.3 選択と適応度

選択とは，適応度の高い個体を増殖させ，逆に適応度の低い個体を淘汰する操作である．ここではルーレット選択（各個体の適応度に比例した確率で子孫を残す選択）に，最良の個体を保護するためにエリート戦略（集団中で最も適応度の高い個体をそのまま次の世代に残す戦略）を加えたものを使用する．適応度に

は，割付にかかるトラックの総周回数（レジスタ数）の逆数を使用する．

3.4 交　　叉

交叉は，個体群から 2 つの個体をランダムに選び出し，お互いの遺伝子配列を組み替える操作である．GA においては，TSP 用の交叉方法がいくつか研究されている[5)]．この中から TSP により効果的である Edge Recombination Crossover[5)]（以下，EX と略す）に着目する．EX は，2 つ親の遺伝子が表す巡回路からある都市の前後に隣接する都市のリストを作り，そのリストからコストが最小になる都市を次の訪問都市として選択していく交叉である．ここでは，2 章で述べた制約を満たすように EX を k-TSP 用に変更して用いる．

変更した EX のアルゴリズムは次のようになる．

（1） 個体群から 2 つの親となる個体を取り出す．互いの遺伝子配列が表す巡回路から，すべての都市についてその都市の直後（**図 3** の矢印の向き）にくる都市のリストを作成する．ここでは都市間の距離が非対称であるため直前の都市は参照しない．

（2） 生成される子は対応する親（図 3 では子 1 は親 1）から各セールスマンが担当している巡回路の先頭の都市群を受け継ぐ（図 3 の四角部分）．

（3） 対応する親の遺伝子配列上で総距離が最短となる巡回路の先頭の都市を基準として選択する．

（4） 基準となる都市との距離が最短となる都市をリストから選択し次の訪問都市とする．今度はその都市を基準として再び (4) を実行する．リストからの選択肢がなくなれば (5) へ進む．

（5） 対応する親の遺伝子配列上で総距離が次に短い巡回路の先頭の都市を基準として選択して (4) へ戻る．すべての先頭の都市を基準として配置し終えたのであれば (6) へ進む．

（6） 選択されずに残ったものは，各巡回路の最後の都市の中で 1 番距離が近いものの後ろに配置し，すべての都市を配置したところで終了する．

図 3 は，この変更した EX を用いて親 1，親 2 から子 1 を生成した図である．ただし，図 3 では親 1，親 2 は巡回路のコストの多い順に並んでいるとし，点線，線上の数字は都市間の距離の大きさを表している．(1) より隣接する都市のリストを作る．図 3 では，10 の直後にくる都市は親の 2 つの遺伝子配列より {3,6}，9 の場合は {1,4} となる．子 1 は親 1 から先頭の部分を受け継ぎ，(3) より先頭部分の中からまず 9 が基準として選ばれる．9 の次は直後にくる都市のリストから {1,4} の 2 つであるが，より短い 1 を選択する．1 の次は {2} の 1 つだけしか選択肢がないので自動的に 2 に決まる．2 の次は選択肢がないので，次のセールスマンへと移る．このようにして EX を実行していく．

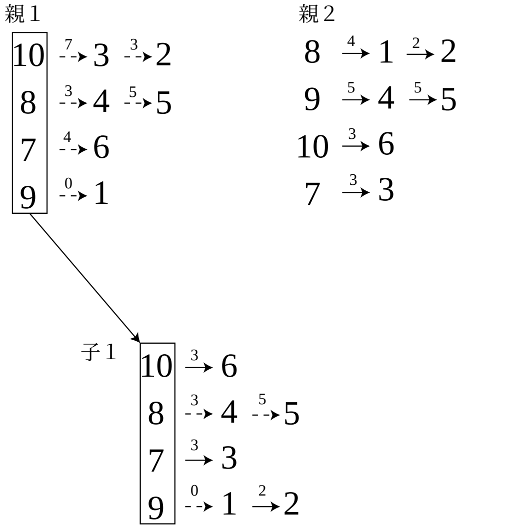

図 3　変更した EX
Fig. 3　Modified EX.

3.5 突 然 変 異

交叉では親に依存した子供しかできないため，ランダムに選び出した遺伝子配列の遺伝子を別の遺伝子に置き換える操作である突然変異を行うことで探索の範囲を広めることができる．ここでは次の 3 種類の方法を用いる．

（1） 2 番目以降の都市からランダムに 2 つ選び出し入れ替える操作

（2） あるセールスマンの 2 番目以降の都市を 1 つ選び，他のセールスマンへと移し替える操作

（3） 先頭の都市群からランダムに 2 つ選び出して入れ替える操作

遺伝子配列の構造が複数の数列となるので 1 つだけの操作では十分に変異させることができないと考えたためと（(1) と (2)），巡回路の先頭に来る都市群が一意に定まるので全体の都市群を 2 つに分類できる（(1), (2) と (3)）からである．適用時にはこの 3 つの操作の中から毎回ランダムに 1 つを選び出して使用する．

各オペレータの確率などの GA のパラメータは，予備実験を行い良い結果を出すものを採用している．以上の結果をもとに GA によるレジスタ割付を行う．

4. 実験と考察

実験には，LFK から得られた 55 の例題を用いて

表 1 LFK を用いた実験結果
Table 1 Experimental result of LFK.

近似アルゴリズム	最適解との差分			
	0	+1	+2	+3
k-NEARINSERT	45	6	3	1
k-NEARNEIGHBOR	50	2	1	2
Slide Coloring Algorithm	51	3	1	
Short Bridge Algorithm	53		1	1
GA	55			

行った．GA のパラメータは，予備実験から，交叉確率を 0.60，突然変異確率を 0.10 とし，十分精度を出すために問題の程度にかかわらず，初期個体数を 200 とする．初期個体の遺伝子配列はすべてランダムに生成する．終了条件には下界☆を用い，下界に到達した場合はそこで探索を打ち切り，到達しなかった場合は世代数 200 をもって打ち切ることとする．

実験環境は，Celeron 400 MHz の FreeBSD 上の gcc にて行った．結果は比較のためにヒューリスティック解法による結果とともに**表 1** に示す．なお k-NEARINSERT および k-NEARNEIGHBOR は k-TSP のヒューリスティック解法[4)]，Slide Coloring Algorithm および Short Brige Algorithm はスライドレジスタ割付問題のヒューリスティック解法[1)] である．LFK の例題を用いた実験の結果より，他のヒューリスティック解法はいくつか最適解でない値を出しているが，GA を用いたレジスタ割付の結果では，すべての例題において最適解を導き出している．

55 の例題の中の例題の 1 つを取り出して考察する．この問題では，ライブレンジ（都市）の数が 34，トラック（セールスマン）の数が 16 である．**図 4** は，この例題においてのレジスタ数の変化の度合いを表すグラフである．ここでは終了条件を変えて 200 世代としているが，20 世代で最適解☆☆へと収束しているのが読み取れる．これは，設計した遺伝子配列に対して交叉，突然変異が効果的に働いていることを表している．

また，この実験における GA によるレジスタ割付の実行時間は 1 ms から 0.1 ms の間であったのに対して，表 1 であげた他の方法はいずれも 0.1 ms もかからなかった．

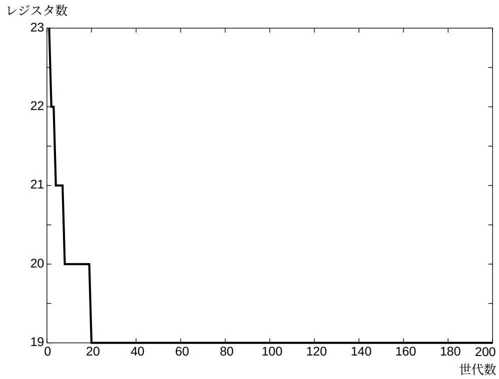

図 4 レジスタ数と世代数
Fig. 4 Numbers of register and generation numbers.

5. まとめ

CP-PACS のスライドレジスタ割付問題において GA を用いて最適解を得る方法を提案し，GA の遺伝子配列の設計を行い，それに選択，交叉，突然変異の各方法を適用した．実験の結果から，LFK の例題すべてに対して最適解を得ることができた．しかし，この GA によるレジスタ割付は時間がかかるため，長時間かかるプログラムやどうしても最適化したいという場合にのみ使用するのが効果的である．これからは，コンパイラの他の最適化問題，たとえばスケジューリング問題などに応用していきたい．

☆ 文献 2) の制約系 3 から式 (3.4) を除いて構成したもの．
☆☆ この問題の最適解は 19 である．

参考文献

1) 籾川友宏，添野元秀，山下義行，中田育男：スライドウィンドウを考慮したレジスタ割付，情報処理学会論文誌，Vol.39, No.9, pp.2684–2694 (1998).
2) 籾川友宏，山下義行，中田育男：スライドレジスタ割付の厳密解法，情報処理学会論文誌，Vol.40, No.9, pp.3524–3536 (1999).
3) 位守弘充，中村　宏，朴　泰佑，中澤喜三郎：スライドウィンドウ方式による疑似ベクトルプロセッサ，情報処理学会論文誌，Vol.34, No.12, pp.2612–2623 (1993).
4) Frederickson, G.N., Hecht, M.S. and Kim, C.E.: Approximation Algorithms for Some Routing Problems, *SIAM Journal on Computing 7*, pp.178–193 (1992).
5) 三宮信夫，喜多　一，玉置　久，岩本貴司：遺伝的アルゴリズムと最適化，朝倉書店 (1998).
6) Goldberg, D.E. and Lingle, R.: Alleles, loci, and the traveling saleman problems, *Proc. International Conference on Genetic Algorithms, Lawrence Erlbaum Associates*, Hillsdale, NJ, pp.154–159 (1985).

(平成 12 年 3 月 24 日受付)
(平成 12 年 9 月 7 日採録)