2K－11

遺伝的アルゴリズムを用いた超並列計算機CP-PACSのレジスタ割付

中家鉄雄　秡川友宏　山下義行

筑波大学

1．はじめに

遺伝的アルゴリズム（Genetic Algorithms：GA，以下GAと略す）は、これまで多くの分野において応用されているが、コンパイラに関連した分野では応用された例が少ない。本研究では、第1段階としてまずレジスタ割付にGAを応用した。

著者らは、筑波大学計算物理学研究センターの超並列計算機CP-PACSを対象として、そのレジスタ割付の最適化について研究している。CP-PACSは、高速演算のために後に説明するスライドレジスタという特殊な機構をもつ。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[2]も提案されたが、それらはいずれも局所最適手の積み上げに基づく手法であるため、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GA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。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ので報告する。

2．基本概念

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[1]）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。このレジスタの表現方法としてSpiral Graph[2]が用いられる。Spiral Graphは，横軸に命令ステップ番号を取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。

Register Allocation for the CP-PACS computer by Genetic Algorithms
Tetsuo NAKAIE , Tomohiro HARAIKAWA , Yoshiyuki YAMASHITA
University of Tsukuba

このトラック上に、そ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。例えば、図1の変数v_1は、0番目のload命令から4番目の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わけであるが、図1のようにそのまま順に配置するのでは隙間が多くできてしまい、効率が悪い。このためどのような順番で配置するかが問題となる。

図1は、1トラックの場合を示しているが、実際には複数のトラックを扱う。複数のトラックにおいてレジスタ割付を行うには、まず各ライブレンジをいずれかのトラック上に分配し、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問題となる。

```
   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
   }
```

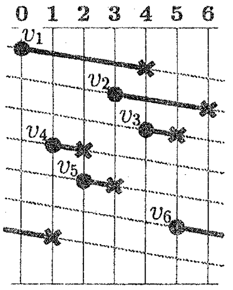

図1　Spiral Graphの考え方

3．GAによるスライドレジスタ割付問題の解法

このレジスタ割付問題では、先ほど示したように遺伝子に組み込む情報として、まず、ライブレンジの識別番号、そして、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この3つを組み込む必要がある。このため、図2のように二重構造の遺伝子配列を用いて、これを表現する。まず、ライブレンジの数だけ配列の長さを取り、1段目の配列にライブレンジの識別番号、2段目の配列にはトラック番号を代入する。トラックに格納する順番については、配列の先頭に入っている方から順番を決める。評価については、この遺伝子の情報からSpiral Graph上に割付し、レジスタの本数を算出する。適応度にはこのレジスタの本数の逆数を用いる。

GAの各オペレーターについて説明する。

◇ 選択　ルーレット選択に最良の個体を保護するためにエリート戦略を加えたものを使用する。

◇ 交叉　ライブレンジを格納している遺伝子の１段目の配列が重複を許さない順列表現になっているので、Edge Recombination Crossover[3] (以下EXと略す) を用いる。EXは、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未訪問都市を次の訪問都市として選択していく交叉であるが、これをスライドレジスタを考慮して改良した。まず、このレジスタ割付問題ではライブレンジとライブレンジの距離が非対称[1] なので後続のものだけを隣接リストに格納するようにしている。また、隣接の定義はトラックの番号を格納している遺伝子の２段目の配列を考慮する。例えば図２では、遺伝子上ではライブレンジ"４"とライブレンジ"６"は隣り合っているがトラック番号が違うので隣接してはいない。この場合ライブレンジ"４"と隣接しているのはそれに一番近い同じトラック番号のライブレンジ"５"となる。各トラックの先頭のものを対応する親から受け継ぎ、それを基準にしてEXを実行する。

◇ 突然変異　遺伝子が二重構造になっているので、１段目と2段目の両方の遺伝子配列をまんべんなく変異させたほうが良いと考えた。そこで、１段目の配列の2つ遺伝子を交換する操作、そして2段目の配列の遺伝子を別の遺伝子に置き換える操作の２つの操作の中から毎回ランダムに1つを選択して使用する。

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の操作によりレジスタ割付を行う。

ライブレンジ番号	1	2	3	4	6	9	7	8	5
トラック番号	2	3	1	4	1	3	2	2	4

図２　遺伝子構造

４．実験と評価

実験には、LFKから得られた55の例題を用いて行った。十分精度を出すために問題の程度に関わらず、世代数を100、個体数を200に設定した。実験環境は、PowerPC G3-266MHz上にて行った。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共に表１に示す。

なお、NEARINSERTおよびNEARNEIGHBORはTSPのヒューリスティック解法[4]、SHORTBRIDGE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[2] 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。

表１　GAと他の近似解法の比較

近似アルゴリズム	最適解との差分			
	0	＋1	＋2	＋3
GA	55	0	0	0
NEARINSERT	45	6	3	1
NEARNEIGHBOR	50	2	1	2
SHORTBRIDGE	53	0	1	1

５．まとめ

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、LFKの例題は比較的簡単な問題であるので、これよりも難しい問題に対して最適解を導き出せるかどうかが問題となる。また、実用のためにはGAの収束判定のための条件をどう決めるかということも問題である。今後は、これらの問題を解決すると共に、コンパイラの他の最適化問題、例えばスケジューリング問題などに応用していきたい。

参考文献

1）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)

2）秾川友宏，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)

3）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

4） G.N.Frederickson, M.S.Hecht & C.E.Kim : "Approximation Algorithms for Some Routing Problems" , SIAM Journal on Computing 7 , 178-193 (1992)