修士論文

対交換とＳＡアルゴリズムを併用した並列配置の高速化手法

指導教官　　富田 眞治 教授

京都大学工学研究科情報工学専攻

李 国東

平成 8 年 2 月 15 日

対交換とＳＡアルゴリズムを併用した並列配置の高速化手法

李 国東

内容梗概

配置は自動配置配線の第1処理段階として部品の位置を決定する処理である．配置の結果は自動配置配線の全体に大きな影響を与える．自動配線の各段階の問題と同じように，配置は大量の計算時間やメモリを必要とする．

逐次計算機の計算スピードやメモリ容量が大幅に発達しても，VLSIの規模や回路設計上の要求の増加はそれを上回る．この問題は並列計算機を用いることで解決できる．近年，低価格の並列計算機が市販され，ワークステーションも最新並列技術を利用していることから，CAD向け並列アルゴリズムの研究は将来にわたる需要が期待できる．

配置は他の自動配線問題と同じように，配置セル(部品，配置モジュールなど)の接続関係が複雑なため，並列化すると多くの通信回数が必要になる．したがって大量の通信時間がかかり，計算時間を上回ることも頻繁に起こる．この結果多くの場合に並列アルゴリズムは理想なスピードアップを出さない．また，スピードアップのために一部の通信を減らすと，並列化の配置結果が逐次より悪くなること多い．こうした様々な問題は，配置問題の並列化の発展を阻んでおり，高速かつ配置結果の優良な新しい並列配置アルゴリズムが必要である．本研究で逐次配置アルゴリズムの並列化を行った．これらのスピードや結果の優良性の分析から，新しい並列配置アルゴリズムを提案した．

対交換アルゴリズムは伝統的な逐次配置アルゴリズムである．計算が簡単で，セル交換は隣接する場合のみ許すという強い規則性を持つ，配置結果は最新の最適化テクニックを利用したアルゴリズムより劣り，スピードも他のアルゴリズムより優れているとはいえないので，ここ十年間はこのアルゴリズムの新しい並列化手法提案されていない．しかし，アルゴリズム対する観点は逐次と並列で同じではない．アルゴリズムの分析結果から，一定条件下では各プロセッサは自分が担当する配置領域内の計算を独立に行えることが判明した．これを利用し，本研究では通信回数を従来のアルゴリズムより大幅に減少させながら，逐次と同じ結果を得られるアルゴリズムを提案する．このアルゴリズムをベン

チマークデータにより評価した結果，8台プロセッサで6.8倍の高いスピードアップを得た．

配置問題でローカルな改善手段は基本的なものは対交換，自由交換，再配置の三つである．最新のアルゴリズムのほどんどは自由交換，再配置に最適化テクニックを加えたものである．自由交換，再配置の並列化方法の多くはいずれも一回実行する毎に通信が必要であり，通信回数を減らせば，逐次より結果が悪くなる可能性が高い．これに対し，本研究で提案した並列対交換アルゴリズムは，収束するまでの時間が他の多くの並列配置アルゴリズムより短く，スピードも速くなる可能性が大きい．しかし，対交換は並列・逐次共に最終配置結果が他の最新配置アルゴリズムより劣るという欠点もある．本論では最新配置アルゴリズムである，SAアルゴリズムの並列化も行い，8台プロセッサで逐次に対し3.4倍のスピードアップを得た．配置結果は対交換より大幅に優良であることも判明した．しかし，スピードは並列対交換よりやや遅い．

本論では従来の逐次アルゴリズムの並列化だけではなく，新しい並列配置アルゴリズム，並列対交換・SA併用アルゴリズムを提案する．前記の二つのアルゴリズムを処理の段階に応じて使い分けることにより，両者の利点を引き出して，スピードと優良配置結果の両方を目指している．性能評価の結果，アルゴリズムのスピードは並列対交換に近く，配置結果は対交換のものより，大幅に改善され，最適化配置結果へ収束可能といわれるSAアルゴリズムに近く．逐次に対するスピードアップは8台プロセッサで約6倍スピードアップを達成した．

さらに，本研究では並列対交換・SA併用アルゴリズムの補足案として隣接しないセルの交換も許す，並列対交換$^+$アルゴリズムを提案し，このアルゴリズムの性能評価も行った．本アルゴリズムは並列対交換・SA併用アルゴリズムと同程度のスピードで，配置結果は少し劣る．しかし，SAで必要な値決定の難しいパラメータが不要である．

Techniques to Speedup the Prarallel Placement with the Pairwise Interchange and the SA Algorithms

Guodong Li

Abstract

The placement is the first step of the Automated Placement and Routing(APR) in LSI and PCB designs. The result of placement affects all the other steps in the APR. The main problems of the placement are the long computation time and large memory size required.

Even though sequential computers are getting faster and having large memory, the complexity in chip design is growing much faster. Parallel computers can offer a solution for this problem. A large number of medium priced parallel computers are commercially available nowadays. Even workstation are using parallel processing technology currently. Hence, the development of parallel algorithms for CAD becomes a hot of the research.

As well as other CAD problems, the placement cells is very complex. Moreover in the parallel placement we spent a lot of time for communications and sometimes it is longer than the computation time for the placement. Always it is not so easy to achieve we good speedup with a reasonable solution in solving CAD problem, because these exists a trade off between the quality of the solution and the speedup. Developments of the parallel placement algorithm are suffered from these problem. We need a new parallel algorithm which has a high speed and offers good solution.

In this research, at first we have parallelized some sequential placement algorithms. According to the result of the analysis of these parallel algorithms, we proposed two new placement algorithms.

Pairwise Interchange algorithm is a traditional placement algorithm. it is simple and has a strict rule that is allows the interchange only between neighboring cells. But its result is not better than some algorithms which employ an optimization technique, and the speed is not so fast as them, So far only a few researchs have been done. But the key issues of the algorithm designs are different between parallel and sequential algorithm. By analyzing the Pairwise Interchange algorithm, we find that under some condition a big number of cells can calculate and move independently. So in parallel algorithm, processor can calculate a great number of cells only using one communication. And the result is as well as sequential algorithm. In this paper we have gotten 6.8 times of speed up using 8 processor with parallel Pairwise Interchange algorithm.

In local placement roles, always use three methods, there are pairwise interchange, exchange and displacement. A lot of algorithms which employ some optimization technique always use exchange and displacement as the placement role. In exchange and displacement, after one cell(or pair of cells) be moved, always use a communication, the time of communication is much larger than Pairwise Interchange algorithm. Hence the Pairwise Interchange algorithm may become the fastest parallel algorithm. But the Pairwise Interchange algorithm have a drawback that its result is not better than the others. In this reseach we also parallelize the SA algorithm and have obtained 3.4 times of speedup with 8 processors. The result of SA algorithm is quite better than the Pairwise Interchange algorithm, but the speed is much slower than Pairwise Interchange.

We also propose a new algorithm which employs the Pairwise Interchange algorithm and the SA algorithms together. This new algorithm could achieve both the high speed close to the Pairwise Interchange algorithm, and the qualily of solution almost the same as SA. This new algorithm also got about 6 speedup with 8 processor.

At last we propose another new algorithm as a supplement of our new parallel algorithm which employ the Pairwise Interchange that allows the interchange between non-neighbour cells. We also evaluated andanalyize it. Although the speed of this algorithm is same as our new algorithm and the solution also not better than our new algorithm, but it need not some difficult parameter of the SA algorithm.

対交換とＳＡアルゴリズムを併用した 並列配置の高速化手法

目次

1 はじめに 1

1.1 配置問題とは 1

1.2 並列配置アルゴリズム 5

1.3 配置問題の並列化の必要性 7

1.4 並列計算機 AP1000 の紹介 8

2 対交換アルゴリズムの並列化 9

2.1 対交換処理法 10

2.2 従来の並列化手法 12

2.3 対交換並列化における，独立実行可能対の分析 14

2.4 対交換アルゴリズムの並列化手法の提案 19

2.4.1 配置領域の分割 19

2.4.2 メモリの節約 21

2.4.3 並列対交換アルゴリズム 22

2.4.4 並列対交換アルゴリズムの性能評価 22

3 SA アルゴリズムの並列化 24

3.1 自由交換の定義及び並列化 24

3.1.1 スピードアップための改良並列化法 26

3.1.2 エラーについて 29

3.2 SA アルゴリズムの並列化手法提案 30

3.2.1 SA 法における評価関数 31

3.2.2 並列温度 SA アルゴリズム 33

3.2.3 並列温度 SA アルゴリズムの性能評価 34

4 並列対交換と SA アルゴリズムの併用による高速化 35

4.1 並列対交換と並列 SA アルゴリズムの比較分析 35

4.2 並列対交換 ·SA 併用アルゴリズムの提案 38

4.2.1 併用理由と問題点の解決方法 38

4.2.2 並列対交換 ·SA 併用アルゴリズム 41

4.3 並列対交換·SA併用アルゴリズムの性能評価 41

5 並列対交換$^+$アルゴリズムの提案 45

5.1 対交換$^+$アルゴリズム 46

5.2 並列対交換$^+$アルゴリズム問題点及び解決方法 46

5.3 対交換$^+$アルゴリズムの性能評価 48

6 おわりに 50

謝辞 51

参考文献 51

1 はじめに

電子工業は従来多くのモジュールを組み合わせて複雑な機能を実現するアセンブリ工業であった，この場合，工業の付加値のかなりの部分は組立て工程になり，そこでは電子部品の配置や電子部品間を接続する配線の設計と施工，検察が作業の大部分を占めた，実現しうる装置の規模や機能は，使用される部品の数，物理のサイズ，信頼度により制約された，また複雑なシステムや装置の設計・試験における計算機の援用 –CAD–はかなり早い時期から実用に供されてきた．

IC は当初その小形軽量という特長の応用面から注目されたのであったが，多数の部品を相互の結線も含めてシリコン・チップ上に形成する IC 技術は必然的にその LSI 化さらに VLSI 化をうながかし，電子回路の“経済化”とともにその“高性能化”と“高信頼化”の同時に達成を可能とした．このような技術革新を可能ならしめた要因の一つは CAD である．いまや，CAD なしには VLSI の設計を考えることはできない．さらに VLSI 設計量による逐次計算機はスピードとメモリの制限により，解決でかないことも発生した，自動配置配線 CAD の並列化は必要である．

配置は自動配置配線の重要な部分である．本章は配置問題の概要，本研究に配置問題対する定義及び本研究で使用した並列計算機 AP1000 について述べる．

1.1 配置問題とは

(1) 配置問題の重要性

自動配置配線問題は図 1.1で示すように二つの主要な処理段階に分けられる．

VLSI チップなどの配置基板上にモジュール (部品，配置セルなと) の位置を決定することが配置処理である．

モジュールの接続関係を決定するのがルーティング処理である．ルーティング処理はグローバルルーティング (global routing) やディテルルーティング (detailed routing) の二つという問題に分けられる．

配置は，その配置結果をルーティング処理段階へ提供する．グローバルルーティングはこの結果を用いて，一次配線を行う．逆にグローバルルーティングで，配置結果をチェックし，配置処理にフィードバックし，2 回配置を行わせることもできる．ディテルルーティングはグローバルルーティングの結果より，具

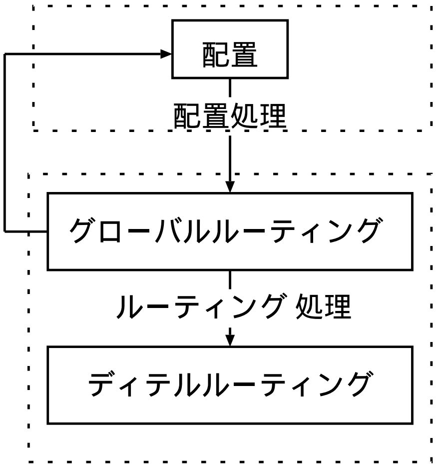


図 1.1: 自動配置配線

体的な配線をする．配置処理からの配置結果の良さが，後のルーティング段階，また自動配置配線問題の全体に多大な影響を与える．

(2) 配置モデルについて

配置配線問題の処理の対象となるモデルにはい多様なものがある．図 1.2に数種のモデルの概要を示す．

スダンダードセルは配置モジュールが一定の規則によって指定配置領域に配置され，モジュール配置領域間をルーティングチャンネルで接続する．ゲート・アレイはサイズが同じ配置領域に基本セル (basic cell) を配置し，配置領域間はスダンダードセルと同じように，チャンネルで接続する．

マクロセルの場合は，配置領域は指定せず，部品を自由に VLSI チップ上に配置する．接続方式はチャンネルではなく，部品を配線で接続する．

配置配線問題モデルに従って，様々な専用配置アルゴリズムがある．本論は図 1.3のようなモデルを用いて，汎用の配置アルゴリズムを論じる．このモデルは部品やモジュールを任意サイズの配置セルで表し，チャンネルなど配置セル間の接続関係は自由配線で表す，配置基板や VLSI チップは二次元格子で表し，配置セルは指定配置領域ではなく, この二次元格子中に自由に配置する．また配置セル位置，二次元格子の座標で表す．

(3) 配置処理の目標

配置処理の目標は配置セルの最適当位置を決定することにより，配置領域と

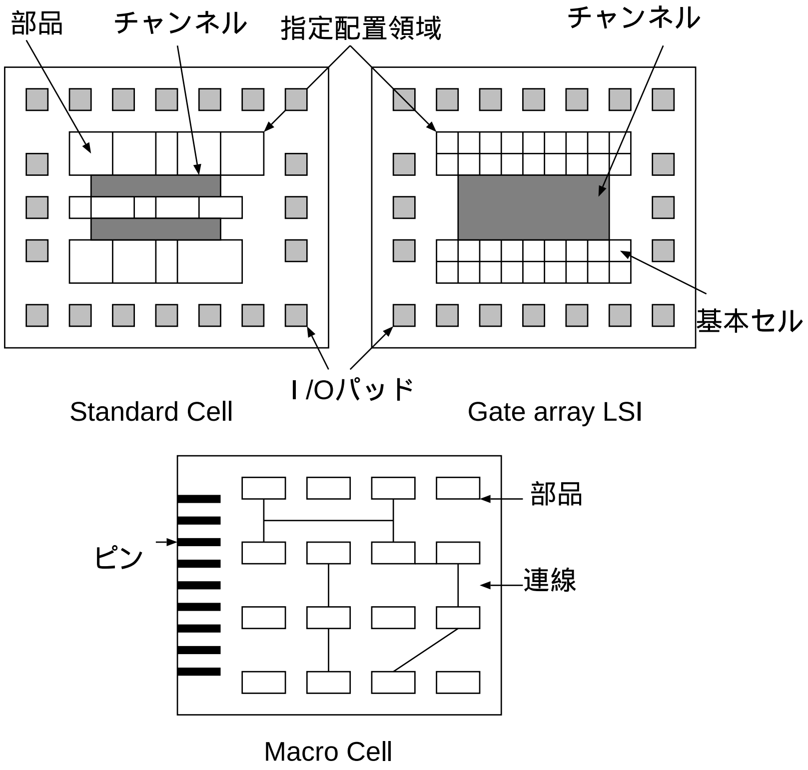


図 1.2: 配置配線問題のモデル

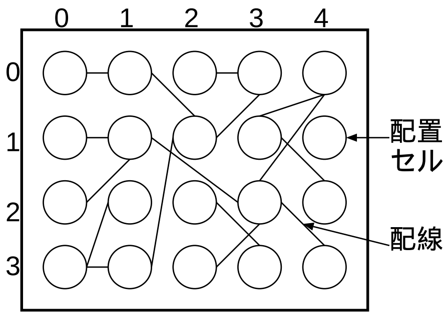


図 1.3: 汎用配置モデル

全配置セル配線ネットの長さを最小化することである．配置領域の最小化を達成すれば，多数の部品を一つのチップ中に配置することができる．近年配置配線問題では，部品の大量化と配線の複雑化が起きている．多くの配線はVLSIの遅延を増加し，性能低下を生じる．配線の長さが減少すれば，配線ネットは簡単になり，部品間の接続関係を密接にし，配置配線に必要な配置領域の最小化を達成することもできる．本研究は，取扱うの配置モデルによって，配線ネットの長さの最小化を第一目標として，並列配置問題の研究を行う．

(4) 配置問題の最小化評価方法

汎用配置モデルで，配置セルは二次元格子の中に配置されており，配置セルの位置や接続関係も，配置状態により固定になる．i，j はそれぞれ接続する配置セルの番号である．配置セルの接続関係は接続関係関数 C_{ij} として表現する．ここで：

$$C_{ij} = \begin{cases} 1 & (\text{i と j 繋がる時}) \\ 0 & (\text{それ以外の時}) \end{cases} \tag{1.1}$$

と定義する．

$C_{ij}=1$ となる任意の配置セル i,j に対し，距離関数 D_{ij} は：

$$Dij = \mid X_j - X_i \mid + \mid Y_j - Y_i \mid \tag{1.2}$$

と定義される．X，Y はセルが二次元格子における配置セルの座標を表す．

以上の定義により，汎用配置モデルの評価関数の一般的な形式は次の方式

$$G = \sum_{ij} C_{ij} D_{ij} \tag{1.3}$$

で表現することができる．

さまざまな配置アルゴリズムでは，配置する過程で上記の評価関数を利用する．配置位置の変更による新しい結果を評価関数による配置結果を評価する．良い評価関数により配置アルゴリズムの配置スピードが大幅に向上し，優良な配置結果を生成することができる．したかって，良い評価関数を用いることは，配置問題において重要である．

1.2 並列配置アルゴリズム

様々な配置アルゴリズムに，最適解への収束方法はグローバル配置状態更新やローカル配置状態更新二つ方法である．グローバルは，全セルに対して，スタート配置状態と異なる状態を生成し，二つの状態を評価し，良い方を保存し，さらに，次の生成状態と評価する．これを最適状態へ収束まで繰り返し実行する．ローカルは何個配置セルに対して，ローカル領域内で評価をする．ローカル配置の結果を改善するとにより，全体の配置の最適解へ収束する．本研究で利用した改善手段は，すべてローカル改善である．ローカル改善方法の基本的な処理手段は対交換 (Pair interchange) と自由交換 (exchange) 及び再配置 (Displacement) である．対交換や自由交換は後の章に詳しく紹介がある

再配置は一つの配置セル A 対し，その最優位置 (重心) を計算し，そこに他のセル B があったらそれを追い出す，次に B の最優位置を計算し，その位置に他のセル C があったらそれを追い出す．全部のセルは一定な位置を配置されるまで，このような繰り返しという手段である．

配置問題に対し代表的な並列アルゴリズムには以下のようなものがある．

- 並列対交換法 (Pairwise Interchange)[1][2]
  2 個のセルを一定のルールによって対に組合せ，対中の二つのセルは評価関数による交換を行われる．違う対の組合せ方式と交換によって，あるセルは基板上の任意の位置へ自由に行くことができる．この方法は計算方式簡単，規則性強いである．しかし，配置結果は最適化テクニックを利用した SA 法より悪い．この方法は 10 年前に特定な並列計算機に並列化された，2 章に詳しく紹介がある．
- 並列重心緩和法 (Force directed cell movement)[3]
  再配置 (Displacement) 処理手法を中心としたアルゴリズムである．配置セルを最初ランダムに配置する．一つプロセッサは担当の領域に一つセル A を選択し，このセルの最優位置を計算し，最優位置へ配置する．最優位置に先についてるのセル B に対し，接続関係による一定量 (全セルの 5%) な重複配置も認める．重複のセルは認める量より多きくなだら，その位置の一つセルの最優位置を計算し，再配置を行う，しかし，一定量のセルを重複配置する後に，通信の必要がある．データの接続複雑性による，重複認める量も違います，文献 [3] に一つ配置位置に 5%の重複量を認める．
- 並列温度 SA 法 (Simulated Annealing)[4]

SA 法は，熱力学の理論から派生した最適化問題に対する有力なテクニックであり，配置問題にも応用できるアルゴリズムである．この方法は評価関数及び温度スケジュールにより自由交換或は再配置を行う．

配置する過程では単調な改善ではなく，改善改悪のいずれにもなりうる配置方法を用いる，最小化評価関数による改善を行うが，改善ができない時は，特有の最適化テクニックにより，改悪の配置状態も採用する，この結果配置セルは最適化位置へ配置することが可能となり．結果は優良である．

最近 SA 法は最有効な配置アルゴリズムの一つと言われており．このアルゴリズムについで，多くの研究論文が発表された，しかし，この方法は最適化テクニックを利用するため，多くの計算時間がかかる，スピードが遅いという欠点もある．この方法の処理手法はグローバル改善とローカル改善両方がある．

文献 [4] はローカル改善の自由交換と再配置を採用した，セルの重複選択ことための通信避けるても Intel iPSC/$2^T M$ Hypercube 計算機上に，8 台プロセッサで 3.2~4.5 スピードアップを得た．

- SIMULATED EVOLUTION 法 [5]

  このアルゴリズムは評価，選択及び配置という三つのステップで組み合わされている．評価ステップはバウンディグボックス [5] 法で評価関数を定義し，配置結果の優良性を評価する．選択ステップは，評価ステップの結果から乱数により，配置ステップへ提供する結果を決定する．配置ステップは単なる配置ではなく，異なる評価関数 (ネットカットなど) による再評価するも行う．

- 遺伝法 (Genetic algorithm)[6]

  最初配置セルをランダムに配置する．この配置状態から，遺伝法による新し配置状態を生成し，評価関数で新し配置状態を評価する．良い配置状態出るまで，新し配置順序を生成する．一般的に評価関数はバウンディグボックスを利用する．このアルゴリズムはスピードが遅いという欠点がある．

並列 SIMULATED EVOLUTION 法 [7] と並列遺伝法 [8] はグローバル配置状態更新を利用する，各プロセッサは通信しない，自分で大量の配置状態状態を生成し，この大量の配置状態を評価する時でけ通信の必要がある，通信回数は少ないので，逐次より高いスピードアップを得た，文献 [8] は並列遺伝法で Network of Workstaions 上に違うデータ 8 台は平均 7 スピードアップを取りました．し

Application	CAD tool	Circuit Size	Runtime
Extraction	HPEX3.0	1,000,000 rectangles	1 hour
Placement	Timberwolf6.0	2,907 cells	1 hour
Logic synthesis	MIS2.2	7,657 gates	2 hours
Test generation	HITEC2.0	17,793 gates	5 hours
Circuit simulation	RELAX2.0	40,000 elements	1 month

表 1.1: $SUN/4^{TM}$で VLSI CAD の実行時間

かし，グローバル配置状態更新手段は大量のメモリをかかり，逐次の場合にもスピードが遅くということもある．

本研究にはすべてローカル配置状態更新手段を利用する．

1.3 配置問題の並列化の必要性

LSI の集積度が指数関数的に増加するのに伴い，LSI の設計を自動化する LSI-CAD の重要性がますます大きくなっている．また，そこで取扱う計算量，必要メモリ量も急激に増大している．これによって，今の逐次計算機の技術の発展では配線配置問題を取扱う計算量の発展より遅くなった．以下，並列自動配置配線問題或の必要性を具体的に説明する．

- 実行スピードの高速性

  配置配線問題を取扱う計算量がますます大きくなっているため 計算機の高速化しなければならない．さらに，将来 自動配置配線ソフトも高速処理及び正確性を要求されている．表 1.1[9]CAD プログラムの SUN/4 $SPARC^{TM}$ 上での実行時間例を示す．

  実際の場合には，VLSI 配置サイズは小さいもので約 1000~10000 大きいもので 100 倍の 100000~1000000 のゲートがある．この膨大なデータは，従来の自動配置配線ソフトで数時間から数カ月の仕事である．最近，VLSI

チップに500000より多いゲートも出てきている，このような大きい計算量とそれら伴い膨大なメモリ必要量は，普通の計算機では満足できない．
将来の複雑な自動配置配線問題は高速な並列方式でしか解決できないない．

- 問題サイズの大型化
  普通の自動配置配線問題ソフトは実行必要メモリやスピード制約のため，実行時間の問題だけではなく，逐次計算機のメモリ不足で実行できないの場合もある．
  並列計算機は，一つ問題を，多くのプロセッサが一緒にやるため，逐次計算機よりスピードが向上する．プロセッサが持つメモリの合計は逐次計算機より多くなる．このため膨大な配置配線問題問題に対し，より適している．
- 性能と価格の合理化
  従来より，並列計算機の高速性について認識されている．価格の問題により，実際応用は少なく，並列配置配線問題の発展は制限されている．最近計算機のハード発展したことにより，高性能，低価格の並列計算機も開発され，応用範囲広がっている．自動配置配線問題の並列化の要求も高くなっている．

以上の理由により，将来の自動配置配線問題は並列化が有望である．

1.4　並列計算機 AP1000 の紹介

本節は並列本研究で利用した計算機AP1000のアーキテクチャと必要なプログラムを紹介する．

本研究は並列計算機AP1000利用して，配置問題の並列化の試みを行っている．AP1000は，富士通研究所で開発され，メッセージ通信を基本とした分散メモリ型の並列計算機である．プロセッシングエレメント(プロセッサ)は，16台から1024台まで接続可能である．プロセッサは自分のメモリを持ち，他のプロセッサとメッセージ通信で情報を交換する．AP1000は，高速通信のため，三種類のネットワーク T-NET,B-NE T, S-NET を持っている．

図1.2にAP1000の全体構成を示す．

AP1000で，アプリケーションは，C言語を用いて作成する．通信や同期などの並列特有の機能は，ライブラリとして用意されている関数を呼び出すだけで実現することができる．特に，並列処理の要である通信には，ネットワーク構成を意識した記述ができるように，種々のライブラリを用意している．

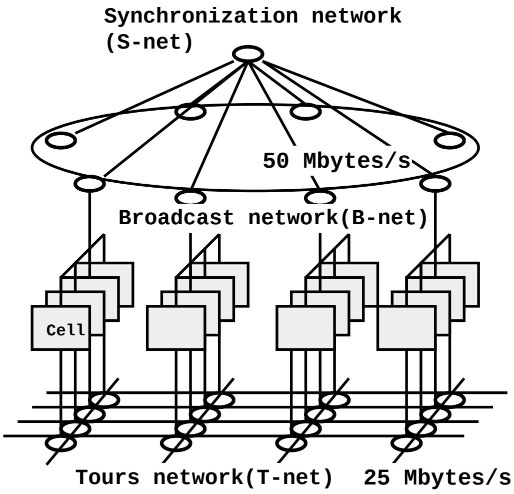


表 1.2: AP1000 アーキテクチャ

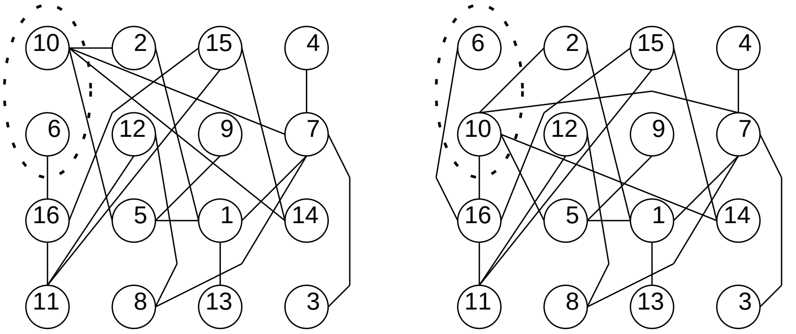
AP1000 では，ホストで実行するプログラムと，プロセッサ (セル) で実行するプログラムの 2 種類のプログラムが必要である．それぞれの役割は

- ホストプログラムセル実行環境設定，セルで実行するタスクの生成，セルへのメッセージ送信，セルからのメッセージの受信，セル状態監視などの処理．
- セルプログラム
  ホストやセル間で通信しながら，セルで並列に実行される処理．

2　対交換アルゴリズムの並列化

逐次対交換アルゴリズムは伝統的な配置アルゴリズムである．計算が簡単であるが，結果は最新の最適化テクニックを利用したアルゴリズムより劣る，スピードも別のアルゴリズムより優勢はないので，近十年内はこのアルゴリズムの並列研究者は少ない．しかし，アルゴリズムの良さは逐次と並列で同じではなく，逐次で良くないアルゴリズムが並列でも悪いとはいえない，

本章は対交換アルゴリズムに対して，このアルゴリズムこのの基本処理手段, 隣接するセル対の組合せ方式を説明する．さらに対の並列実行可能性を分析し，対交換の規則性強く特点より，対独立実行の可能性高いの結論を出した，この結論より，高速な並列対交換並列化方法提案した．



(a)　　(b)

図 2.1: 対交換例

2.1　対交換処理法

対交換法は対となる二つの配置セルを交換すると言う処理手法であり，アルゴリズムによって様々な対の組合せ方式がある．

この交換方法を議論するため，まず図 2.1により簡単の対交換問題を考えよう．

図 2.1a の状態で，もしセル 6 とセル 10 を対として選択すれば，6 と 10 の配線長さはそれぞれ 1 と 13 をである．この配線長さの計算方法は第 1 章の式 1.3 による獲得した計算結果である．

この対 (10，6) で交換すれば，配線長さ増減することを発生する．もし配線長さが減少するなら，交換が実行し，(最適化テクニックを利用しない場合)，図 2.1b のような，新しい配置状態を生じる．この新しい配置状態でセル 6 と 10 の配線長さは 2 や 11 となる，総配線長さは 13 となって前より 1 を減少される．

図 2．1 のセル 10 と 6 は隣接するセルである，この対の生成ルールでは隣接するセルによって対を組合せ法となる，この方法用いて配置アルゴリズムを対交換アルゴリズムと呼ぶ．

対交換の簡単な処理手法は，以下のような実行ステップがある：

1. 現存の配置状態から隣接する二つのセルを選択し，対にする．

2. この対の総配線長さを計算する．

3. セルを交換する，新しい配線長さを計算し，もし新しい長さが交換前の長さより減るなら，交換を実行する，減少なかったら，交換しない．

4. 改善率低い，及び無改善までステップ 1 から 3 を繰り返し実行．

対交換処理法に対の組合せ方式

実際の要求によりさまざまな対の組合せ方式が存在する，本論に対の組合せ

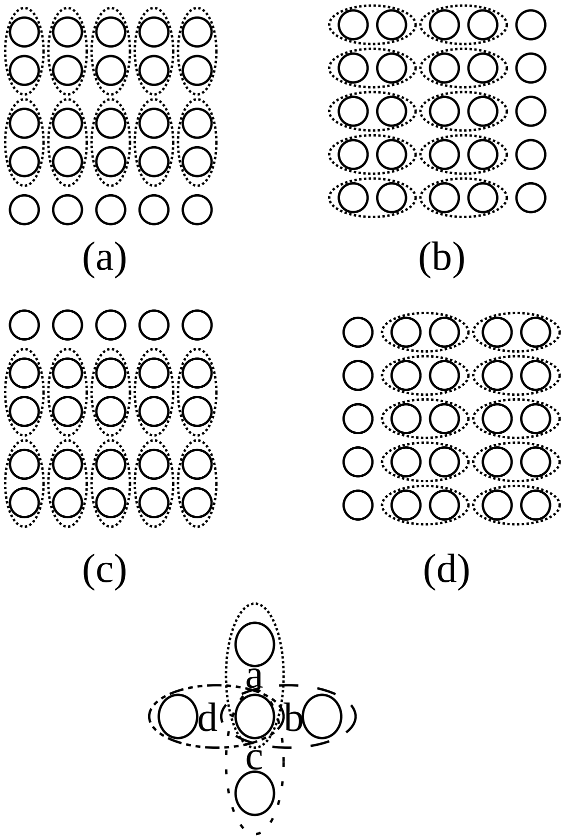


図 2.2: 5× 5 交換対の選択

方法は特に説明しない限り，対交換法は隣接するセルを組合せる．

配置セルは隣接するセル選択方法による，は図 2.2に表現したそれぞれの対組合せ形式があた．

この対組合せの方式は (a)，(b)，(c), (d) 四つの種類がある．

配置領域の行 (列) に対して，上 (左) から，物理位置奇数行 (列) のセルを選択し，選択されたセルは下 (右) に隣接する配置セルと対とすると，(a) あるいは (b) 類の対組合せの方式となる．

配置領域の行 (列) に対して，上 (左) から物理位置偶数行 (列) のセルを選択し，選択されたセルは下 (右) に隣接する配置セルと対になる，配置領域の横縦は N_p偶数の場合は，最下 (右) 行 (列) と最上 (左) 行 (列) が対とすると．(c) あるいは (d) 類の対組合せの方式となる．

対交換を実行の時，全配置領域に (a)，(b)，(c), (d) 方式の一つを実行する．これを繰り返せば，セルは二次元配置領域に，異なる対組合せを選択することによって，任意の方向へも移動できる，(a)，(b)，(c), (d) の一つ方式に対の個

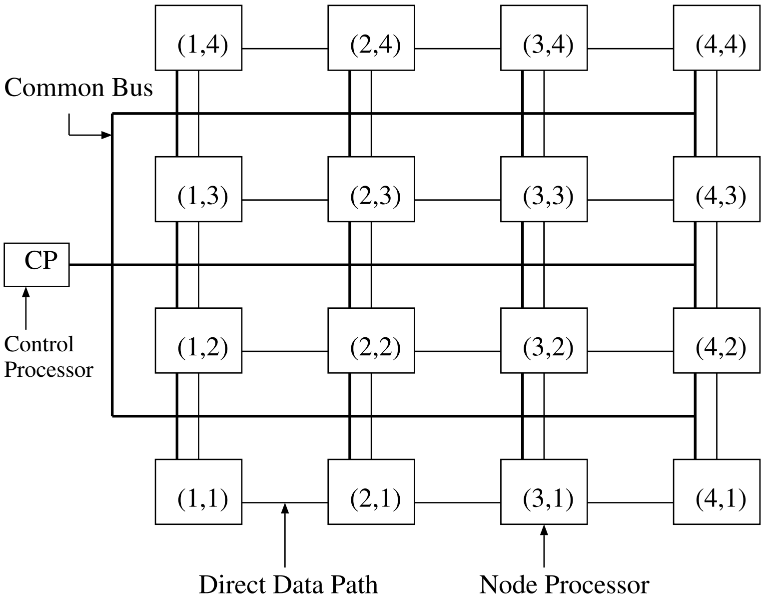


図 2.3: 従来法用並列計算機の構造

数 N_{ea}は

$$N_{ea} = \frac{N_p(N_p - 1)}{2} \tag{2.1}$$

である，

したがって，対交換法は逐次実行時に，配置セル対は一つしか実行できない，並列の場合には，最高は 式 2.1 示す数の対と同時に並列実行ができる．以上の結果は，並列実行の過程で，選択された配置セルを組合せた対は，すべて独立に実行できるという仮定をしている．

2.2　従来の並列化手法

対交換アルゴリズムの並列化方法については，従来の代表的な方法は Ueda[2] や D.J.Chyan は [1] 図 2.3のような MIMD 並列計算機の構造による提案した．

図 2.3 のような MIMD 並列計算機の構造は一つのプロセッサは基本計算プロセッサやメモリを含む，各々のプロセッサは物理位置により，座標を付け，各々方向隣接するプロセッサは Direct Data Path で直接通信する，隣り合わないプロセッサはコントロール・プロセッサで通信する．Direct Data Path で通信する場合はコントロール・プロセッサでの通信より，スピードが速い．

この構造によって，MIMD 計算機の各々のプロセッサは自分のプロセッサ番

号や位置座標を持つ，二次元行列となる，第一章説明の汎用配置モデルにおける配置セルの配列方式も二次元行列で配列している．したがって，汎用配置モデルはプロセッサ構造と簡単にマッピングできる．配置セルの配線の長さ増減や，位置の変換などの評価計算は，各々の担当プロセッサで独立に解決することができる．

例えば，図2.1の配置問題が対交換法で図2.3の4×4のプロセッサ上にマッピングされる場合は，配置セルとプロセッサ間に以下の関係になる，プロセッサ(1，4)はセル10を担当し，二つ組の配置データを含める，一つ組配置セル自分情報，これはは自分を担当したセル番号と位置である，10は配置セルの番号，(1，4)は位置として表現する．もう一つ組は接続関係の情報である．[2，5，7，14]は接続関係があるセルの番号，[(2，4)，(2，2)，(4，3)，(4，2)]はそれぞれの位置である．したっがて，各々のプロセッサは担当するセルの関連する配置データを持っている．

この対応から，各々のプロセッサは自分の持つ情報だけでも，配線の長さの増減，位置の変換などが，独立に計算できることが明らかである．例2.1の通り，もしセル(1，4)と(1，3)を交換すれば，プロセッサ(1，4)は“-2”の長さの減少を独立に評価計算でき，(1，3)は“+1”の長さの増加を計算できる．全配線の長さは(1，4)の“-2”，(1，3)での増加“+1”を合計すると，全配線長さの変化は“-1””となる，したがって，交換する．この二つのセルを交換するための計算は他のプロセッサの情報を必要とせず，担当プロセッサが自分の持つデータだけで独立に解決できる．

この並列方法はプロセッサが必要な情報だけを持つため，メモリを節約できる，通信スピードもDirect Data Path転送方式により高速であるの利点がある．しかし，プロセッサは一回対交換毎にデータ通信が必要である．

対交換処理法の並列化は以下に三つのステップがある．

1. セル交換の判断ステップ

(1) 対の生成および対組合せ方式の選択．

(2) 各セルを対交換の場合に長さの増減量の計算．

(3) 各セルの対交換の結果はによって，交換するかしないか決定する．.

2. セル交換ステップ

もし配置セル交換の判断ステップで二つの配置セルの交換が成立すれば，二つプロセッサ持つの情報をDirect Data Pathで交換を行う，配置位置を更新

する．

3. 情報通信ステップ

一回交換の完了後にその交換セルの新しい位置が放送する．

対交換法の並列実行はこの三つのステップ，評価計算の結果によって無改善になるまで，繰り返しを実行する．

2.3　対交換並列化における，独立実行可能対の分析

対交換法について，従来の並列化方法は一つのプロセッサが一つの交換対を担当し，並列に実行する．しかし，配置セルの接続関係は複雑なため，もし接続関係がある配置セルを同時に移動すれば，プロセッサが適当な時間内通信しない場合は，接続セルの移動情報を知らないために，評価計算でエラーを生じることもある．その交換対に対して，通信しないと，並列実行はできなくなる，以下の分析は，配置対が独立実行できる条件を考察する．

配置対中のセルと他のセルの接続方式は 1. 異なる対行 (列) のセルと接続する，2. 同じ対行 (列) のセルと接続する，2 通りの場合に分けられる．

1. 異なる対行 (列) のセルで接続する場合

この状態を図 2.4に示す．

図 2.4(a) の場合は，対 A 中のセル 1，2 は 対 C，E 中のセル 5，6 とそれぞれ接続している．C，E は対 A と別の対行 (Pair Row) となる．セル 1，2 が仮に交換すされると配線長さは “2” を増加する，したがって，交換しないと判定する．この判定は対 C，E 中のセル 5，6 など移動しても正確ということが明らかである．

一方，対 B と対 D が F を接続するの列では，セル 3，4 は仮想交換によって，“-2” の配線長さ減少量を獲得する．これだけの判断によると，交換が成立する．ただし，この状態は，セル 3，4 と対 D，F 中のセル 7，8 とそれぞれの連絡があり，セル 7，8 も移動する可能性がある，並列実行のときに，セル 7，8 の交換は同時に実行される可能性が高い．

図 2.4(b)，(c) で R2 はセル X と Y の対行となる，R1，R3 はそれ以外の区域である，L は R2 と R2 以外の領域の境界線である．R1(R2 の上) の領域は 2 個配置セルがあた，R3 は 3 個配置セルがある，S は R2 以外の X と接続関係があるのセル，図 2.4(b1) と (b) は S 移動しないの場合のセル X 交換要求の計算，δG=[(5+5)+(5+3+1)]-[(4+4)+(2+4+6)]=-1 図 2.4(c1),(c) は S を移動した

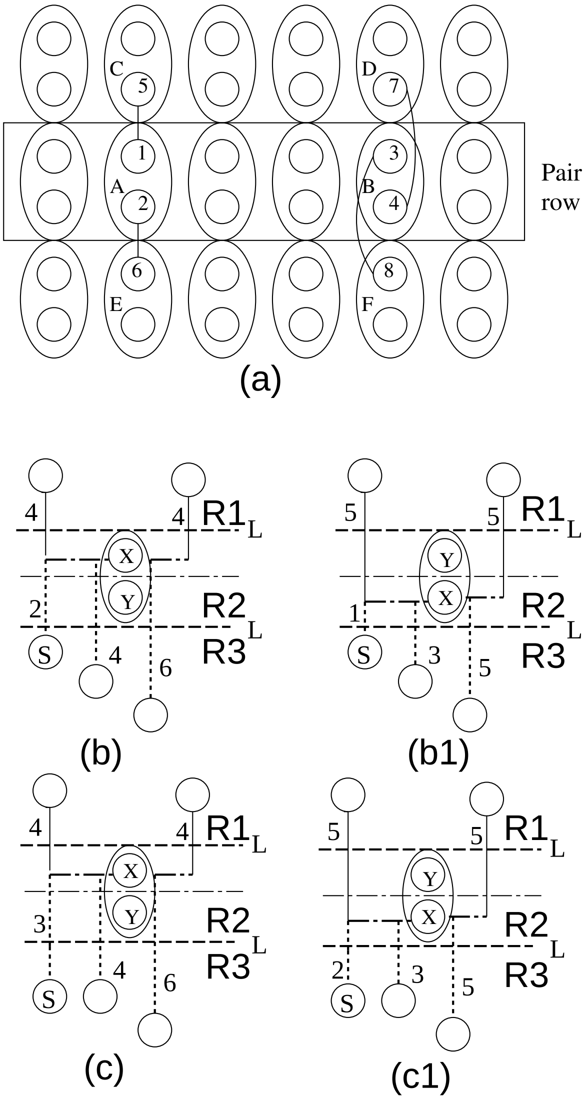


図 2.4: 異なる対行のセルで接続する場合

X 交換要求の計算δG=[(5+5)+(2+3+5)]-[(4 +4)+(2+3+5)]=-1, この-1 は X より上と下のセルの個数差である，更に簡単と言えば，対による交換のときに X 方向あるいは Y 方向の一次元移動となる，評価関数の公式によって，セルは接続セルの枚数が多い方向へ移動の要求をする，

評価関数によりの証明では仮定 D_X，D_Yはセル X，Y に対し各々の配線距離．+,-は上 (右) 及び下 (左) の方向，

$$D_X = \sum_{i=1}^{p} D_{iX}^{+} + \sum_{i=1}^{q} D_{iX}^{-}$$
$$D_Y = \sum_{i=1}^{p} D_{iY}^{+} + \sum_{i=1}^{q} D_{iY}^{-}$$

一つ対 P(XY) の全配線距離 DP(XY) は

$$DP_{XY} = D_X + D_Y$$
$$DP_{XY} = \sum_{i=1}^{m} D_{iXY}^{+} + \sum_{i=1}^{n} D_{iXY}^{-}$$

それによって，対 P(XY) 仮想交換後，新しい状態 DP(X'Y') の 配線距離は

$$\begin{aligned} DP(X'Y') &= D_{X'} + D_{Y'} \qquad (2.2) \\ &= \sum_{i=1}^{m}(D_{iXY}^{+} + 1) + \sum_{i=1}^{n}(D_{iXY}^{-} - 1) \\ &= \sum_{i=1}^{m} D_{iXY}^{+} + \sum_{i=1}^{n} D_{iXY}^{-} + m - n \end{aligned}$$

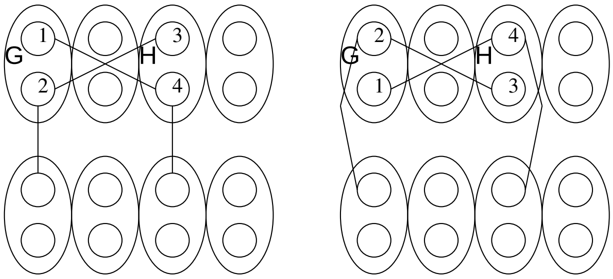
評価関数によって

$$\Delta G = DP(X'Y') - DP_{XY} = m - n \qquad (2.3)$$

m や n は，正，負方向に対 P(XY) と接続する各々セルの個数である．

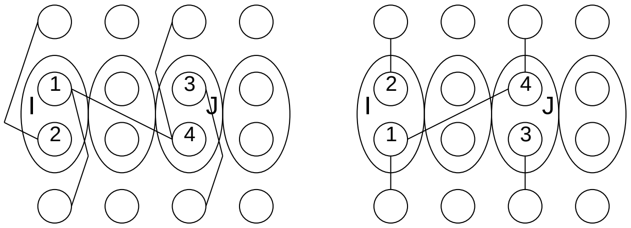
セルが異なる行の対と接続する場合，図 2.4b の S は R1 や R2 の区域に入らないから，S と X は平行になたり，上下関係が逆になることはない，S は同時移動しても X に対して，横縦方向の接続セルの個数は変わらないので，評価計算に対して，影響を与えない．

この時に，実際の配線長さの変化はプロセッサの計算と実際の結果が違う，交換要求の判定は正しい．それによう って，この並列計算は逐次と同じ判断が保証される．



(a) (b)

図 2.5: 並列実行における交換のエラー



(a) (b)

図 2.6: 計算エラーあるでも判定正確

2. 同じ対行 (列) のセルで接続する

この状態の並列処理は難し．各対を並列に場計算した結果は長さが減少するように見えても，実際には増加する可能性がある．

図 2.5a では対 G を仮想交換する配線の長さの変化は “1” 減少する．対 H も同様の判断を下し，判断は交換成立となる．しかし，本当に並列実行をすれば，図 2.6b の状態となり，全配線の長さは減少せずに，増加する．このような位置関係対は独立に実行ができない．

図 2.6では対 I 及び J を仮想交換すると配線長さはいずれも “3” 減少することが計算でわかり，判断は交換成立となる．しかし，全配線長さの減少量は最初の計算結果 “6” と違う，3 となる．この場合は計算エラーが出ても，交換判定は正確になっている．したかってこのような位置関係の対は独立実行ができる．

以上の結果によると，交換対象のセルが同じ対行 (列) のセルと接続する時は，接続関係によって，判断が正しい場合と誤りである場合の 2 通りがある．

図 2.7は同じ行の対と接続する交換対の状態を分類したものである，この類別によって詳しい分析を行う．

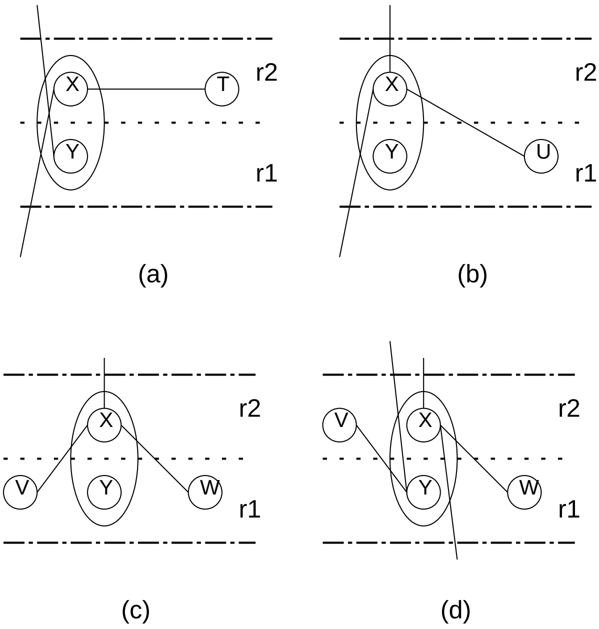


図2.7: 同じ対行(列)のセルで接続するの状態

図2.7aはセルTとXを接続し，また，Tは(X，Y)の対と同じ対行，r2セル行中にある．この時もしXやYを交換すれば，予想結果が出る．Tも同じ時間に対となるセルと交換すれば，予想結果と異なり，それ以上の配線の減少量が得られる．この場合に，判定はいつも正しい．Tと同様な位置関係のセルが２個以上ある場合ても，同じ判断結果が出る．したがって，同じ対行(列)と接続するセルは接続されるセルと同じセル行(列)にあれば，対交換の判定はいつも正しいことがわかる．

次は図2.7bの場合は，セルXと接続するセルUはXより一つ低いセル行r1にある．XやYを交換すれば，“1”減少し，Uも移動すれば，その減少量は0になる．したがってこの交換判定はいずれの場合は誤りでわない．

図2.7a，bから， 同じ行(列)対と接続するセルが一つだけある場合では，交換は独立判定ができる．

図2.7cやdは接続セルが一つ以上であれば，同じ対の行でも，どこの区域でも交換判定のエラーが発生する．独立できないセルは交換する時に，通信しないと行けない．

以上の分析まどめて，以下の結論がある ：

(A) 異なる対行(列)のセルで接続する場合，並列判断いつも正し，独立に判定ができる

(B1) 同じ対行 (列) で接続するセルが **1** 個だけある場合では，交換は独立に判定ができる

(B2) 同じ行 (列) 対で接続するセルが **2** 個以上場合は，交換は独立に判定ができない

2.4　対交換アルゴリズムの並列化手法の提案

本節は前節の分析により，新しい対交換アルゴリズムの並列化方法を提案する．

新しい並列対交換アルゴリズムに一つのプロセッサに対し，配置情報の分割，メモリの節約すること，及びプロセッサ分担を議論する．

2.4.1　配置領域の分割

2.3 節の分析によって，隣接セルの対を組合せる方法を採用すれば，対交換アルゴリズム中で，大部分の対は独立に交換することができ，同じ対行 (列) のセルと接続する場合がある対は，独立に実行できとできない両方のことがある．

対交換アルゴリズム並列化の場合には，図 2.7 の状態 (a)∼(d) 対の組合せ方式によって，並列実行の時にプロセッサはできるだけ同じの対行 (列) が担当する．そうすれば，一つの対行中の対はプロセッサ上で逐次に実行し，2.3 節分析された独立に実行できない状態を避けられる．交換対行毎のプロセッサの分担は配置問題の縦，横の長さ N_P あるいは N_P-1 はプロセッサの台数 P の二倍あるいは 2^N倍の場合は，プロセッサの配置は簡単に実現することができる．図 2.8 は $N_P = 8$，プロセッサ 4 台を利用する時に，配置領域に対しプロセッサの分担を表す．

各プロセッサは各交換対行の評価計算や交換を担当する．図 2.8の例では PE0 は対 a を含むの交換対行を担当し，他のプロセッサと並列に処理を実行する．PE0 が担当した交換対行の中では，対 a と対 b が接続するため，これらは並列実行できない．しかし，対 a と対 b は同じプロセッサ PE0 で担当するので，逐次に実行される．したがって，評価計算する時にエラーを避けることができる．このように並列実行すれば，各々のプロセッサは自分が担当した交換対行 (列) の全べて対の評価計算と交換終るまで，通信する必要はなくなるので，通信回数が十分減少する．もし配置問題の縦，横の長さ N_pあるいは N_p-1 プロセッサ利用台数の 2^N倍なる場合は (a)∼(d) 対組合せ方式で一回交換周期中に，必要通信回数はプロセッサの台数 P となる．

並列計算機の資源を十分利用するため，ユーザは配置問題に並列処理する時

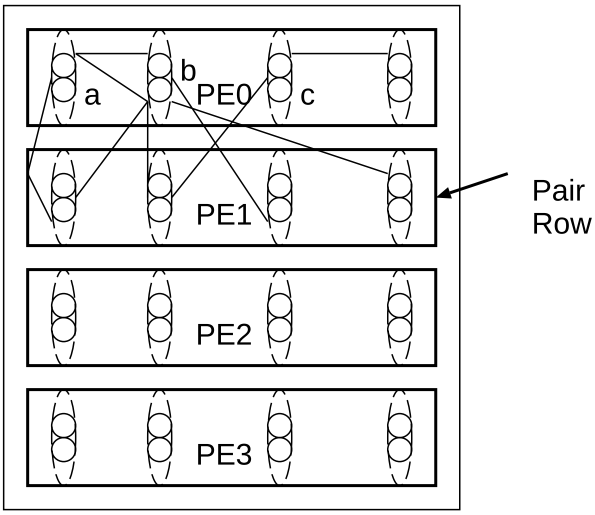


図 2.8: プロセッサの配置処理領域 1

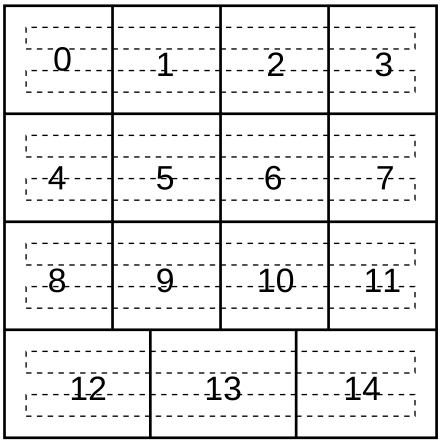


図 2.9: プロセッサの配置処理領域 2

に全部のプロセッサが利用する場合は多い．しかし，N_p，N_p-1 はプロセッ全部台数 P の二倍より小さい，あるいは二倍ならない場合は，一つのプロセッサが一つの交換対行に対し，全部交換対行 (列) を担当することはできない，一行の交換対は何個かのプロセッサで担当させる．独立に実行可能の分析することより，同じ交換対行を担当するのプロセッサが並列実行する時に，通信する必要がある．

図 2.9では $1N_p = 8$ の配置問題に対し，15 台プロセッサを解し場合の対行分担例を示す．

図 2.9には同じ交換対行は数台のプロセッサを担当している．2.3 節で分析されたように，もしあるセルは同じ交換対行の異なるセル行のセルを接続すれば，

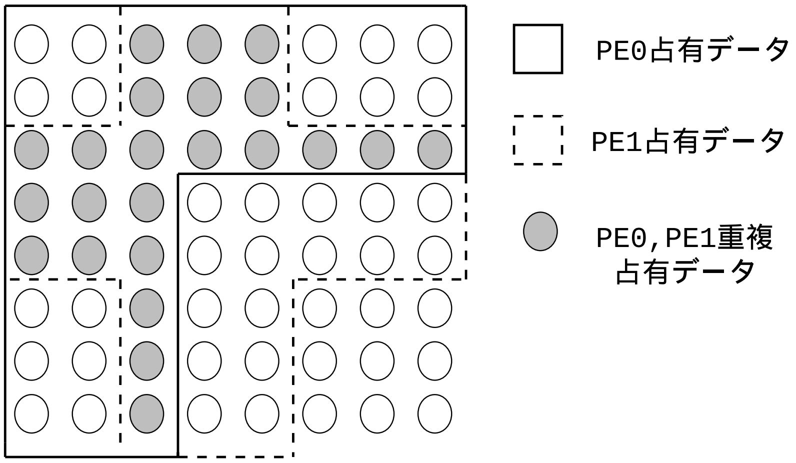


図 2.10: 並列対交換のメモリ利用量

セルは別々のプロセッサを担当している場合は，通信しないと交換エラーを出るの可能性がある．

2.4.2　メモリの節約

一つセルが持つ配置情報は，自身関する情報 (セル番号，座標)，セルと接続する関係あるセル情報 (セル番号，座標) を含む．

プロセッサは一定領域のセルを担当し，評価計算や交換を行う，全部ではなく配置情報の一部が必要である．一つのプロセッサは (a)~(d) の対組合せ方式により，その領域内の情報だけを持てばよい，. もし，一つのプロセッサは一つ交換対行の計算を担当すれば，(a)，(b) の対組合せ方式によって，この交換対行 (列) の配置情報プラス隣接一つのセル行 (列) の配置情報を持ってもよい．(a~(d) の対組合せ方式合わせて，一つプロセッサ必要の配置情報量は配置問題の縦，横は N_pとなる場合は P はプロセッサの台数 :$\beta = 2(N_p/P + 1) * N_p - N_p^2/P^2$ となている．この式により，プロセッサ台数を多く採用する場合は，一つプロセッサは必要の情報量は，全体より多くない．図 2.10分担の例である．

一般な並列自動配置配線問題には情報の持つ量と通信必要回数は大きいな関係がある．持ち情報量は多く，通信回数は相対少ない，逆に多い．この配置情報の分割方式ではプロセッサは 1 個実行段階が終るまで，必要な情報を持っている，通信の必要はない．さらに，通信後に，アップデータため，少しの更新がある．しかし，全体のメモリ利用サイズは変わらない，通信回数が減らしメモリ利用量も少ないの利点がある．

さらに，この分担方法後で提案する並列 SA，対交換と SA 併用アルゴリズム

おいても大いに有効であることが判明した．このことは後の章に詳しい議論がある．

2.4.3　並列対交換アルゴリズム

対交換並列アルゴリズムは以下のようなステップがある：

(1) 配置領域分担

(2) プロセッサは自分の担当範囲内実行 (a)~(d) の一つを交換実行．

(3)(a)~(d) の一つを交換実行し，終ると情報を全プロセッサに放送する．

(4) 受信，必要データを選択．

(5) 各プロセッサはもし自分の担当範囲内に無交換すれば (2) での情報を放送する時に，無交換のこと”0”でけ放送，これに対し，プロセッサの受信情報は全部 0 になれば，実行停止し，ホストデータを送信，実行全部終了．終了判断は自分でできる．ホスト毎一回実行する後に，もし，終了判断はならない，ホストと通信の必要はない，(2) へ戻る，繰り返す実行．

この並列手続きによって，8 台プロセッサで 6.8 倍従来方法より高い処理速度を獲得した，この並列方法によって，対交換アルゴリズムは高速並列化を達成することが明らがである．

2.4.4　並列対交換アルゴリズムの性能評価

図 2.11は二つ配置データ用いて，並列対交換アルゴリズムのスピードアップことを表す．

Ppw1　：配置セル個数 4087 配置領域 64× 64

Ppw2　：配置セル個数 784 配置領域 28×28

図 4.2 による対交換アルゴリズム並列化をする時，配置データによって，スピードアップが異なる．データまた配置領域が大きい場合は，プロセッサは担当の領域が大きい，大きな担当領域内，一つの (a)..(b) 実行周期終るまで，通信の必要はないので，通信回数相対には少ない，データまた配置領域小さい場合は，一つのプロセッサが一つの交換対行を担当する交換対も少ない．さらに，全交換行を担当することならない場合もある，配置セル評価計算するとき，エラーを防止するため，同じ交換対行担当したほかのプロセッサと通信する必要があるので，プロセッサの利用台数 P は配置領域の横，縦の長 $N_p/2$ or $N_p-1/2$ より大きいとき，スピードアップ率は下ろす．

配置問題 Ppw1 では 8 台プロセッサ利用する場合は 6.77 の高スピードアップを獲得した. Ppw2 は 14 台，Sp1 は 32 台から，プロセッサの利用台数 P は配

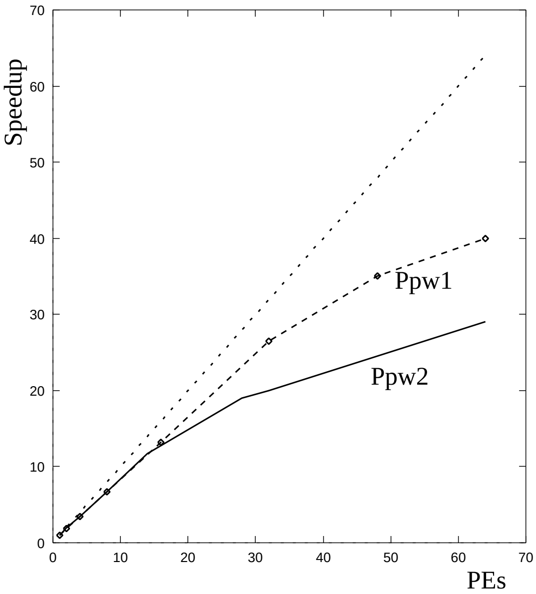


図 2.11: 並列対交換アルゴリズムのスピードアップ

置領域の横，縦の長 $N_p/2$ or $N_p - 1/2$ より大きい, スピードアップ率は減少することが図も表現された．

以上の結果や分析することによって，本研究並列化した対交換アルゴリズムは高速化達成し.

図 2.12に同じ実行周期に逐次や並列実行する対交換アルゴリズムの結果をあらわす．

SPpw1　対交換アルゴリズムで，並列と逐次を実行するΔG > 0 の場合，配置セル交換をする．

SPpw2　対交換アルゴリズムで，並列と逐次を実行するΔG≥ 0 の場合，配置セル交換をする．

プロセッサは担当した領域に，全部の対に対し，評価計算と交換を実行することが終る前に，交換情報を通信しないので，もし接続する配置セルは同じ時間に移動をすれば，全配置問題に対して，配線長さの計算することは，エラーを生じる．そのエラーは交換判断に対し，ホストで，全配置領域の結果よるの長さ計算すれば，そのエラーはなくなる．並列と逐次の実行方式と交換判断は同じので，結果も同じ．並列と逐次対交換は交換判断はすべて同じため，結果の変化も同じとなる．それによる，以上の結果変化を得た．

SPpw1 と SPpw2 の結果より，評価関数ΔG=0 も交換をする場合は，配置セ

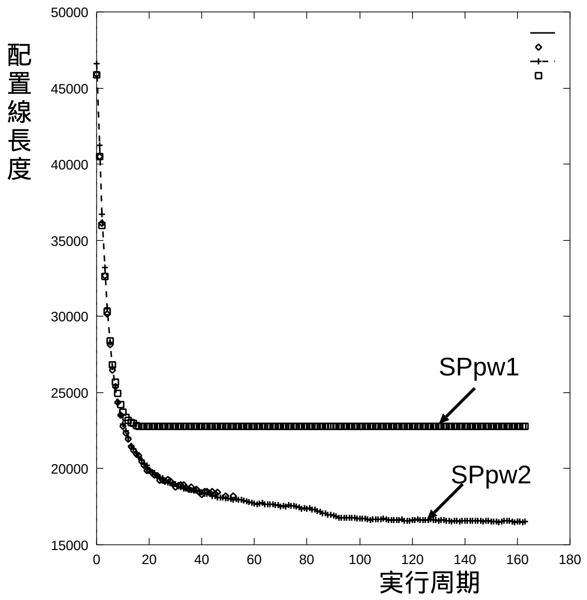


図 2.12: 並列と逐次対交換アルゴリズムの比較

ルの活動性を増加する，良い結果を得た，最小化結果へ収束の可能性も増加することが明らかである．

以上の分析による，SPpw2 法の評価関数は従来と異なる，$\Delta \mathrm{G} \geq 0$ を採用した，従来の対交換アルゴリズム SPpw1 法により，いい結果を出した．さらに対交換法は最後に無改善となる，その理由は新しいアルゴリズム提案のため第 5 章に詳しい分析する．

3　SA アルゴリズムの並列化

本論の SA アルゴリズムが採用された配置状態変化の処理手段は自由交換である．本章はまず自由交換の並列化を行う．次に，SA アルゴリズムを紹介し，自由交換の並列化を SA アルゴリズムへ広げ，SA の並列化を行う．

3.1　自由交換の定義及び並列化

対交換法では全部対の組合せ方式や対になるの二つのセルの決定する. 決定した対方式 (a)...(d) から，組合せ方式を一つ選び，評価計算を実行する．並列実行の場合でも，セルや対などが同時に重複選択されることはない．自由交換での基本的なセルの選択方法では，乱数による任意セルの選択である．並列実

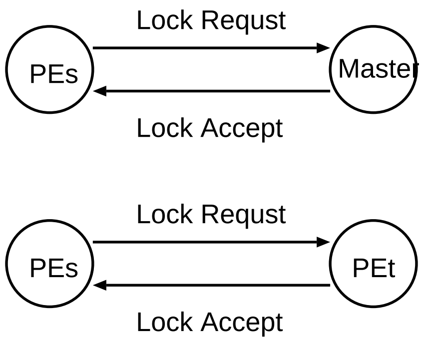


図 3.1: セルをロックための通信

行の時に，異なるプロセッサの間で通信をしない場合，他のプロセッサは選択されたセル番号などの情報は知らないため，2 個以上のプロセッサは同じセルを選ぶ場合は生じる．部分配置情報がなくなるため，その場合生じる誤りは，回復できない誤りと言えます．

この問題の解決には，セルを選択する時に，選択されたセルをロックして，ロック情報を放送し，評価計算後にセルの解放と解放情報の放送が必要です．この大量の放送，通信によって，実行時間が増加し，並列化の効率が低下する．

一般的な自由交換法の並列実行ステップは次のようになる：

(1) セルの選択，ロックの請求送信，受信．

(2) 交換計算．

(3) 新しい配置位置情報を送信，受信．

(4) 停止まで (1) へ戻る．

この実行ステップでも，配置セルのロックすることは通信が必要である．図 3.1にこのロック時の通信することを示す．

並列実行時に，マスターがあれば，各プロセッサはマスターへロック要求を出す．マスターは，その各々のプロセッサからのロック要求によってセルの実行できるかとか判断し，判断結果はロック要求したのプロセッサへ送信する．マスターがない場合は，ロック要求を出したセルは，全プロセッサからのロック要求や要求の返事が来るまで，待つ必要がある．受信の時間，その間にかかる待ち時間が，実際の評価計算や交換の時間より，長くかかる．それに対し，システムの効率が大幅に低下すると予想される．実際に，この方法で 8 台プロセッサを利用した場合，2 倍のスピードアップしか達成できない．システムの効率が大きく低下していることがわかる．

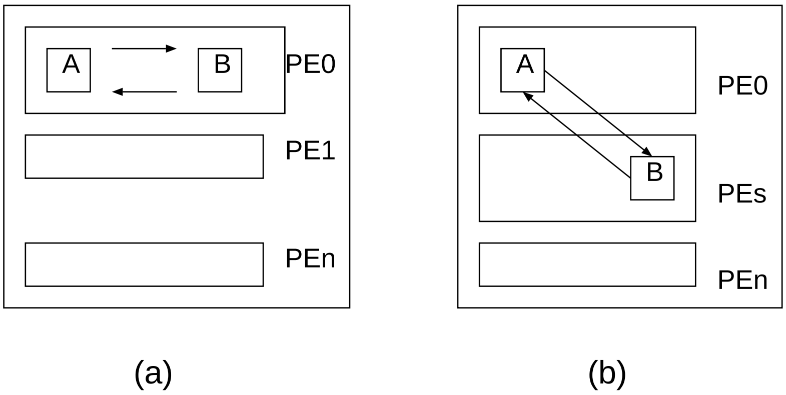


図 3.2: プロセッサに対しの自由交換

3.1.1 スピードアップための改良並列化法

並列計算機 AP1000 は，プロセッサがローカルメモリを持ち，計算データは各プロセッサの担当分配する，一つのプロセッサは，自分の範囲内管理するデータから交換スタートセルを選択する．これに対し，選択された交換スタートセルはプロセッサによって違う．でも交換相手を選択する時に，図 3.2以下の状態がある．

図 3.2a の状態は交換相手は同じプロセッサ中にあり，プロセッサがセル A の交換相手 B がロックされたかどうかの情報を持つ，したがって，通信を行わずに，交換計算できる．

図 3.2b の状態ではセル A の交換相手セル B は他のプロセッサ PEs 上にある，交換要求のあるセル A を持つプロセッサはプロセッサ PEs へロックの請求を送信し，後にその請求の返信を待つ状態に入る．その返信が来る前に，次の運算実行をすることできない．実際には，3.2b 状態発生の確率は高い．プロセッサの個数は P となる，配置データは N_p X N_pである，一つのプロセッサが持つデータ量は $(N_p \times N_p)/\mathrm{P}$，交換相手も違うプロセッサにある確率は :(P-1)/P となる，P は大きい場合，1 と近い確率となる．

改良法 **A**

図 3.2b の状態が生じる時，実際に通信量は大きくないが，通信回数が多く，プロセッサが待ち状態に入ると，何も処理を行わないので，大量にシステム資源を浪費する．

それに対し，次のような並列方法を考える．最初実行する時に，一つのプロセッサは多数の交換セルを選択し，ロック要求を出す．この時に，もしこのセ

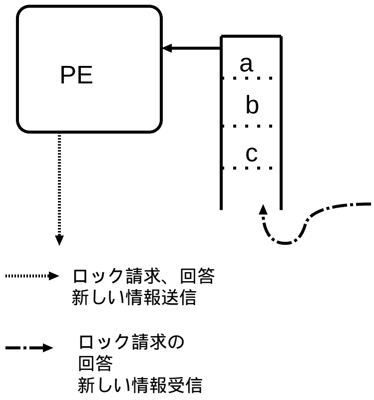


図 3.3: 改良法Ａ :待ち状態の防止

ルの交換相手が同じプロセッサ上に存在すれば，交換計算を行う．その以外場合は，待ち状態に入らなければならない．

ロック要求の返信が来るから，実行状態に入る．一つの交換計算が終ると，この新しい配置状態とを放送する，しかし，待ち状態に入らないため，一定の数量のロック要求存在しなければならないので，この時に，新しロック要求を出すことが必要です．もう一つの問題はそのロック要求の個数である，その個数が多い場合は，重複ロック要求が発生するので，ロック失敗の返信も増加する．プロセッサ数や配置データ数によって，ロック要求個数の定義が必要である．

この並列方法は先に多数のロック要求を出し，また，返信要求は優先になるので，プロセッサに対し，いつも一定数量の交換要求が存在することになり，待ち状態を防止することができる．

この方法 (以下改良法Ａと呼ぶ) 図 3.3で待ち状態の防止を示す．

改良法 **B**

実際の配置問題で，最小化評価関数が複雑でない時，通信時間は最小化評価の時間より長くかかる．また，実際の配置問題におけるサイズや配置セルの個数に対し同じ時間に重複に選択されるセルの個数は多くない．この小さい重複に選択されるセルの個数のため，多くの通信回数をを費すのは無駄である．

改良法Ｂでは交換セルの相手を選択した後に，選択されたセルのロックをしない．それによって，ロックのための通信の必要もなくなり，評価計算によって交換を決定した後に，新しいセル情報だけを通信する．エラーを防止する手

段として，各プロセッサが受信カウンターを設定し，受信元のプロセッサの番号を記録する．プロセッサは，他の全部のプロセッサからメッセージの受信が一回終ると一回受信周期となる．例えばＰＥ０は，ＰＥ１からＰＥ63まで新しい情報全部を一回受信すれば，一回受信周期となる．この一回受信周期の中に，同じセルの位置変化は二度と認められない．

この方法は別の方法と同じ回数の交換を実行すれば，スピードが速くなる．でも評価結果も悪くなる，しかし，同じ評価結果が出るまでの実行時間は改良法Ｂのスピードも速いと言う結果もある．

改良法 C

自由交換では，セルの交換相手は全配置領域中に乱数によって自由選択される．その目的はセルが全配置領域にどこへでも自由な移動することである．以下の方法では，全配置領域を数個の子領域に分割する，各々の配置子領域の交換計算は別々のプロセッサが担当する．一回の交換が終ると，各プロセッサの担当する領域は図3.4(A1)..(A4)のような変わる．プロセッサの担当は横，縦で交換する，セルの自由移動をできた，図3.4にもセルの自由移動可能性を示した，実際は2回の交換でAは自由になった．改良Cに一つプロセッサのデータ必要量あるいは配置領域の分配は対交換アルゴリズムと同じなる．これによって，PEnとPEn+1は1行，1列の情報を重複持ち，同じプロセッサは異なる実行ステップ(A1)..(A4)に担当配置領域のサイズも異なる，重複の1行，1列情報はPEnとPEn+1は切替えを利用する．この利点はセルがさらに自由になる．例図3.4なかの点BはA1の時にPE0を持つ，A3の時にPE1を持ち，この重複の1行，1列はセルの中継点となる移動自由性を増加する．

分割する方法は図3.4で表す

方法	プロセッサ数	実行時間	配線長
普通並列法	8	48 秒	16763
改良法Ａ	8	41 秒	16825
改良法Ｂ	8	36 秒	16663
改良法Ｃ	8	26 秒	16681

以上の結果より，配置結果はあまり変わらない状態に，改良法Ｃのスピードが速い，本論は改良法Ｃの並列化方法による，SAアルゴリズムの並列化を行う．

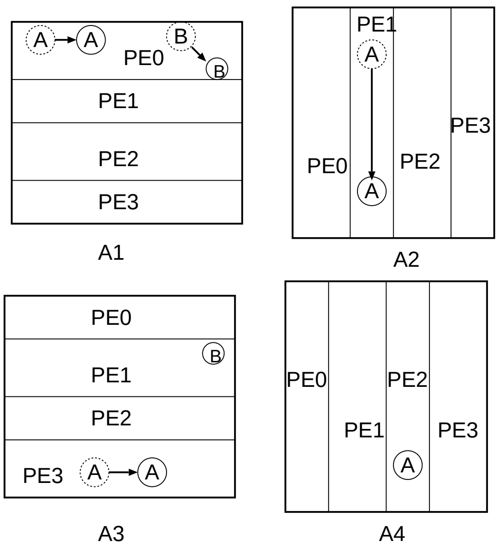


図 3.4: プロセッサの配置処理領域

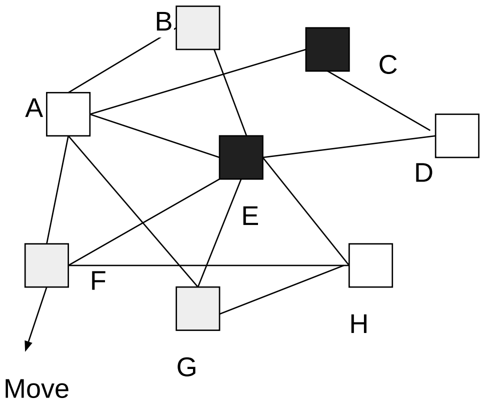


図 3.5: エラーを避ける策

3.1.2 エラーについて

配置問題のを並列実行する時に，接続するセルは同時に移動する可能性がある．プロセッサがその新しい移動位置を知る前に，評価関数をそのセルの元位置で計算してしまい，評価計算のエラーを生じる．並列対による交換法の評価計算エラーの解決法は 2 章に説明した，自由交換や再配置での解決法の一つを図 3.5に示す．

この図の方法は，最初の配置状態から，乱数によって，一つのセルを選び，一つの色を付ける．そのセルと接続する子セルは順番にこのセルと異なる色をつける．色を節約するため，相互接続しない子セルは，同じ色をつければよい．次に，子セルと接続するセルも同じ手続きで色をつける．全部のセルに色をつけ

るまで，この手続きを繰り返し実行する．この結果は，同じ色を持つセルは相互接続しないことを保証される．図3.5では セル A，D，H は同じ白色を持ち，接続しないことがわかるので，同時に移動することが認められる，同様に，C と E 組，B，F，G 組 同じ色を持つ，同時に評価計算や移動できる．しかし，本論に SA アルゴリズムを実行する時に，改悪も認めるので，スピードのため，利用しまかった．

3.2 SA アルゴリズムの並列化手法提案

温度 SA 法は組合せ最適化問題における評価関数値 E を最小にする解を確率的に探索する，近似解法の一つである．その概要を以下に表す．

1. 配置状態はランダムによる設定．
2. 最高温度の設定．
3. 配置状態の変化．
4. 最小化関数ΔG による評価．
5. 改善するならこの改善状態を取る．
6. 改善しない場合は最適化テクニック関数と乱数 [0，1) の比較．
7. もし

$$exp\left(-\frac{\Delta G}{T(k)}\right) \geq random[0,1)$$

なら，配置状態を変化させる．

8. 温度スケジュールより温度を変化．
9. 最低温度まで 3...8 繰り返しを実行する．
10. 最低温度になれば，結果を一回取る，もし，その結果に不満足あれば，温度を上げて，3．へ戻る．満足すれば，終了する．

ここで制御変数 T と評価関数 G はそれぞれ統計物理学とのアナロジーから温度，エネルギーと呼ばれる．SA 法は，保持している解に微小変形を施し，新しい解が以前の解よりも評価関数の数値において結果が改善すれば受理する．そうでなければ，確率 $\exp(-\Delta G/T)$ に従って受理する．一方，温度スケジュールに関しては，式3.1の温度スケジュールを適用することが多い．

$$T_{k+1} = \alpha T_k, 0 < \alpha < 1 \tag{3.1}$$

しかし，温度スケジュールが問題に適したものでなければ，解のクオリテイ

に大きい影響を与える．この評価関数Gや温度スケジュール問題は，後に詳しく議論する．ＳＡ法はこの評価計算温度スケジュール計算に従って，改悪方向への状態移動を確率的に認めるため，理論上は真最適解への収束が保証されている．しかしながら，そのために，非現実的に長い時間が必要となる．

3.2.1 SA法における評価関数

配置問題では，SA法の評価関数Gとして次方式3.2を利用する．

$$G = w_o O + w_s S + w_l L \tag{3.2}$$

ここで，O，S，Lはそれぞれ，重なりペナルティ，部品間のペナルティ，仮想配線長さの和を表す．w_o，w_s，w_lはそれぞれ関数の重みである．SA法に対し，さまざまな研究者がいろいろな評価関数を利用した．バウンディグボックス[9]評価方法とネットカット評価方法を良く利用されている，しかし，この二つの評価関数は計算が複雑であり，エラーが起こりやすく．本研究では，汎用配置モデルを利用するので，式3.2を利用する，簡単化しすると，

$$G = w_l L$$

$$G = \sum_{ij} C_{ij} D_{P(i)P(j)} \tag{3.3}$$

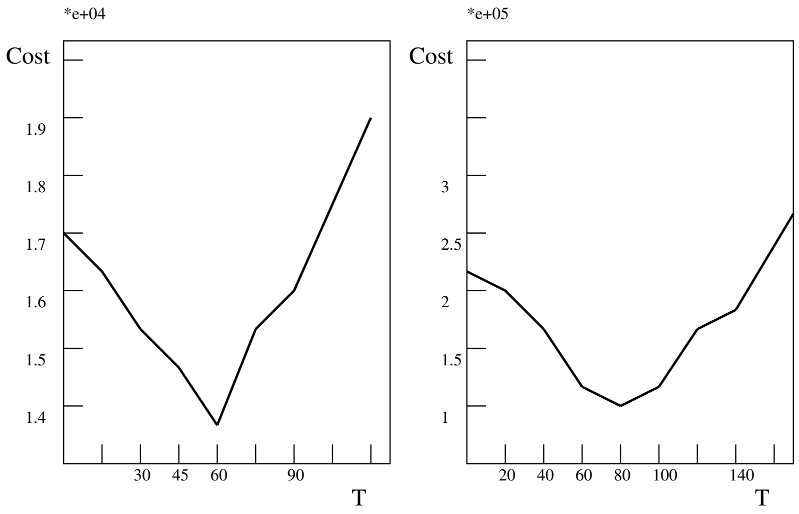
となり，式3.3で表現されたその評価関数は対交換アルゴリズムの評価関数と一致している．

通常のSAアルゴリズムでは，温度Tを温度スケジュール式3.1従って単調減少させる．最終的にセルの移動は，式3.5で定義された関数に従って決定をする．

$$\Delta G = \begin{cases} < 0 & \text{移動をする} \\ \geq 0 \quad exp\left(-\frac{\Delta G}{T(k)}\right) & \text{による決定する．} \end{cases} \tag{3.4}$$

ΔG $\geq$ 0の場合は，評価関数によって，セルを移動すれば，改悪す改悪場合が発生する．しかし，SA法ではそのときでもΔEによって，移動する可能性がある．

$$\Delta E = \begin{cases} > 0 & \text{移動をする} \\ \leq 0 & \text{移動しない} \end{cases} \tag{3.5}$$



Data A　　Data B

図 3.6: 最高温度選択実験

ここで ΔE は

$$\Delta E = exp\left(-\frac{\Delta G}{T(k)}\right) - random[0,1] \tag{3.6}$$

温度スケジュールは式 $T_{k+1} = \alpha T_k$と表現される．温度 SA 法では，時間あるいは実行回数の関数によって，温度スケジュールを実行し，温度を計算する．SA 法スタート時の最高温度，改悪量を制約する最低温度，また，温度スケジュールための重み，α の定義することが必要である．これらのパラメータを，次のような定義する．

最高温度 ：最大の改悪となる状態，移動が 50%[10][11] の確率で受理されるような温度．

最低温度 ：最小の改悪となる状態，移動が一定温度のアニーリング処理における解は一段受理されるような温度．

α ：温度スケジュール用，スケジュールを行うときの重み．

最高温度は移動が 50%の確率で受理されるような温度と定義された，しかし，移動方式や配置データより，評価結果ΔG が異なる，SA 法用いて最高温度の大小も違う，本論研究で扱う SA アルゴリズムでは，移動処理手段は自由交換法を主要移動方法としている．自由交換法は二つの自由配置セルを交換するので，一定範囲内のデータでは，その変化量ΔG も一定範囲内となる．

図 3.6は本論でりようした二つのベンチマークデータによる，自由交換法で実

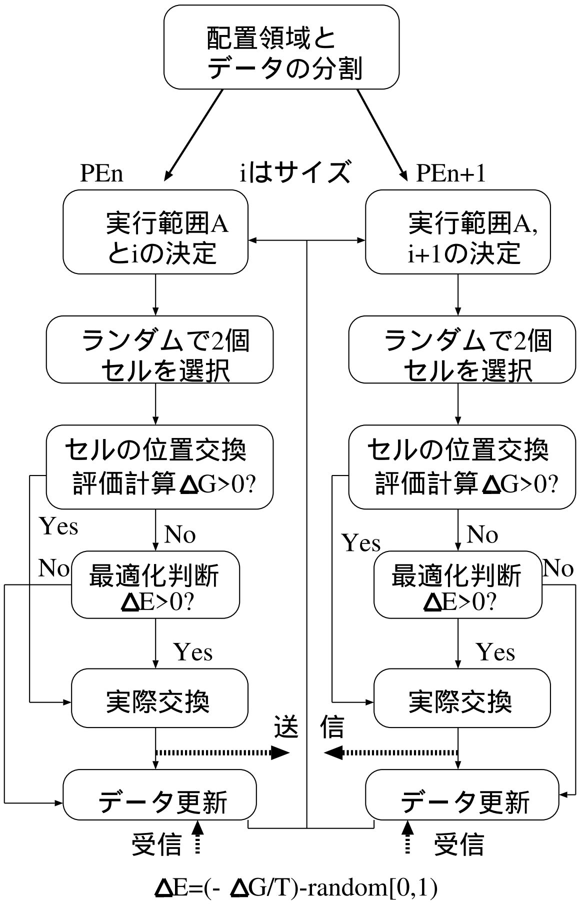


図 3.7: SA のフローチャート

行する場合，60,80 それぞれの温度なる時に，比較良い配置結果が出ることが示す

最低温度も配置データによる，異なる．

αも，様々な値が用いられる．参考文献による，一般に $0.80 \leq \alpha \leq 0.95$[12][10] の範囲内である．

最低温度$\to 0$，温度のアニーリングΔE による交換率$\to 0$ として，満足できる結果が出ない場合には，一定な比例関係で温度上がることも必要である．

3.2.2　並列温度 SA アルゴリズム

本研究に，移動基本処理手段は自由交換あるいは再配置手法を用いて，温度スケジュールのパラメータが様々な大小を決定する．, 自由交換の並列化改良法 C を採用し，図 3.7フローチャートのような並列温度 SA アルゴリズムを生成した．

このアルゴリズムで，並列実行を行い時，優良結果が得られた．

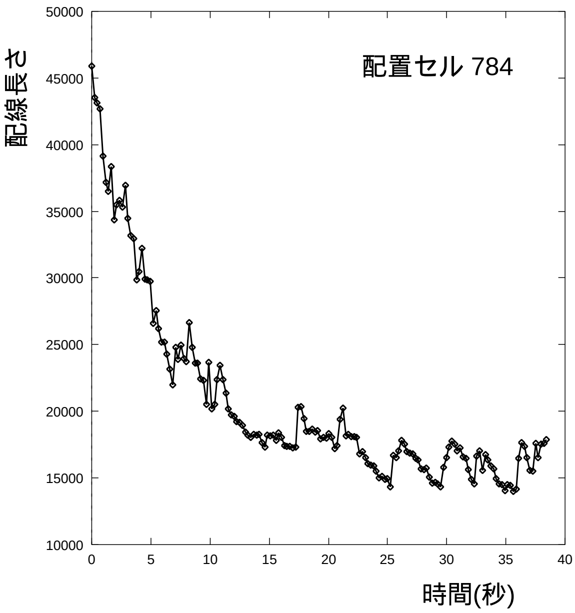


図 3.8: 並列 SA アルゴリズムの配置結果

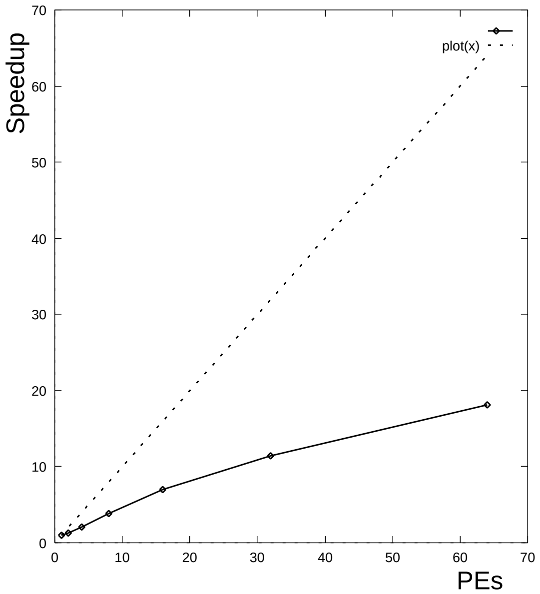


図 3.9: 並列 SA アルゴリズムのスピードアップ

3.2.3 並列温度 SA アルゴリズムの性能評価

図 3.8と図 3.9に示した結果のように，並列温度SA アルゴリズムは温度アニーリングの関数に従って，改悪状態も認める．これによって，配置結果の変化は波形となる．波の山は改悪の状態であり， 波の谷は改善の結果である．全部の趨勢は改善となる．最適化テクニック利用するため，最適化結果へ収束の可能性が大きくなり，理想な結果を出した．

しかし，並列温度 SA アルゴリズムは，自由交換処理手段を利用するため，エラー防止に，多くの通信量や通信回数を必要とする，したかって，スピードアップはあまり高くない．(SA 法のは止まらない，さらに，波形の改善，同じ点から同じ方法ても 2 回で同じ結果出ないので，スピードアップの評価方法は一定範囲内始めての近似結果を出る時の結果として，調査のものである．) 本論に得れるスピードアップは 8 台プロセッサは 3.4 となる．文献 [4] の結果近似，文献 [4] 大量なベンチマークデータ用いて，スピードアップは 3.2∼4.5 平均 4 となる．しかし，配置問題の複雑性による，異なるデータや並列計算機は，正し評価ができない，参考のものである．

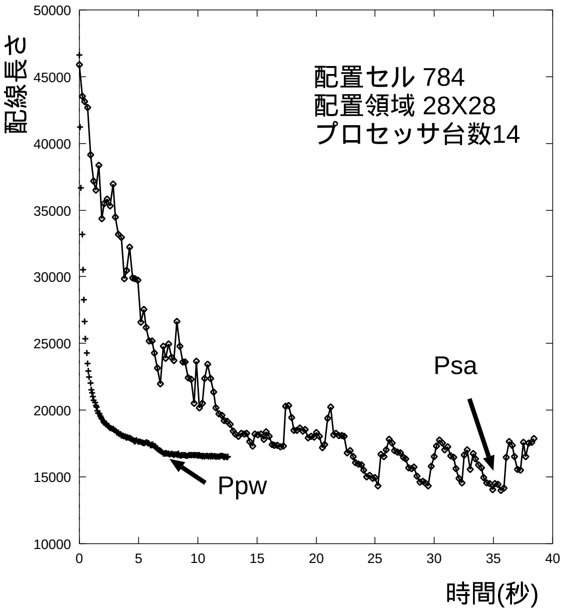


図 4.1: 並列対交換と温度 SA アルゴリズムの比較

4　並列対交換と **SA** アルゴリズムの併用による高速化

本章は二つ並列化されたの配置アルゴリズム，対交換と温度 SA アルゴリズムについて，並列化結果を分析，比較する．また並列対交換と SA アルゴリズムを併用した，新しい並列配置アルゴリズムを提案し．その評価と特性分析も行う．

4.1　並列対交換と並列 **SA** アルゴリズムの比較分析

図 4.1は同じ配置データに，並列 SA，対交換アルゴリズムを適用した場合の時間に対する配置結果の変化である．以下，この結果によって，アルゴリズムの実行スピード，結果の優良性など，様々の方面から，並列対交換と温度 SA アルゴリズムの比較や分析を行う．

図 4.1には

Psa　：並列 SA アルゴリズムを用いての配置結果

Ppw　：並列対交換アルゴリズムを用いての配置結果

(1). 実行スピード

並列対交換アルゴリズム

対交換は，計算が簡単であり，実行スピードが高速である．

対交換の処理手法を採用することにより，対の組合せ方式は一定ルールによるため，セルの選択するための通信は必要ない．また，本研究ではこのアルゴリズム並列化する場合に適当なプロセッサ担当できればプロセッサ間の通信は，自分担当配置領域内対の計算を終るまで，一回だけの通信が必要である．この通信回数は担当した配置領域の大きさと関係ない，並列実行スピードは十分速くなる．さらに，同じベンチマークデータで，, 同じ配置結果を得た時，並列対交換アルゴリズムは並列 SA アルゴリズムより 3.1 倍の高速性も達成した．

並列 SA アルゴリズム

温度スケジュールによる，最適化テクニックを利用するため，計算が複雑となり，長い時間が必要となる [11][13]．本研究ではセルを選択する領域の指定を行ない，自由交換処理手段を利用しても，選択されたセル重複を防止するためのセルをロックするときの通信を回避した．しかし，一つの交換を終ると，通信の必要があり，通信回数は並列対交換アルゴリズムより，大幅に多くなる．したがって，実行スピードが並列対交換アルゴリズムより遅い．

(2)． 結果の良さ

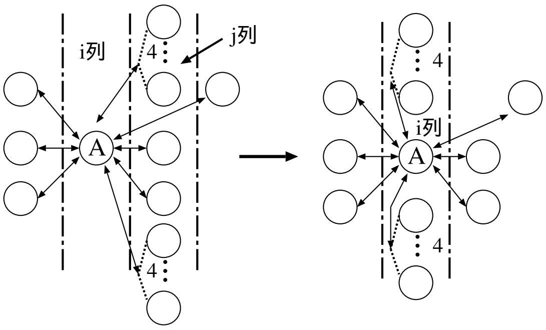
並列対交換アルゴリズム

並列対交換アルゴリズムは交換関数により最適化結果向けて単調に改善を行う，一つの対は，交換を実行する時に，隣接するセルしか交換できないので，特に最適結果へ収束する後半には，移動に一定な制約がある．図 4.2にこの制約の理由の一つを示した．

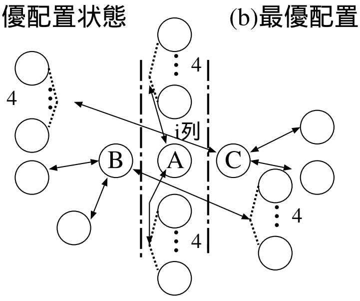
図 4.2はセル A に対し，(b) は A の配置の最優位置，(c) は A と隣接する，セル B，C 交換をできない状態を示した．

配置セル A は図 4.2(a) の状態に非最優配置位置である，A の左は 3 個配置セルがある，右は 11 個となる，A もし一つ右へ移動すれば “8” の配線長さの減少を獲得することができる，A は図 4.2(b) の状態に最優配置位置である，左，右へ移動すればとうちらても，配線長さを増加する．

(c) において A が (b) の状態になっているとすると，B，C は A と隣接する配置セルである，この時に A と B，A と C はという交換対は交換できない，この状態に，B と C 交換できれば，配線長さを減少することができる．しかし，対交換アルゴリズムでは隣接するセルだけ対になるため，B と C の交換は不可能となっている．このような状態は対交換のアルゴリズムにおいて，特に大量の交換を実行する後に多く発生し，最適解へ収束することが不可能となる．



(a)非最優配置状態　　(b)最優配置状態



(c)交換不可能状態

図 4.2: 最優位置と交換不可能状態

以上の考察により対交換は最適化テクニックを利用したSAアルゴリズムより結果が悪いということ明らかである．

並列SAアルゴリズム

シミュレーテッド・アニーリング最適化テクニックを利用し，単調の改善ではなく，改悪方向への移動状態を確率的に認めることによって，理論上は真の最適解への収束が可能となる．したがって，結果は優良である[14]．図4.1のように配線長さによる評価では，本アルゴリズムの結果は対交換アルゴリズムより更に大きいな長さの減少が得られた．

(3). 最適結果への収束

並列対交換アルゴリズム

前半は高速かつ安定な収束率を達成した，しかし後半は，改善率は大幅に下がり，最後は無改善となる．

並列SAアルゴリズム

不安定な収束となる．温度スケジュールによって，高温度の時に収束曲線は変化大きい，低温度の時に安定となる．後半も最適結果へ微小な収束をする．

(4). 終了の判定

並列対交換アルゴリズム

最後は無改善となる，終了判定は簡単となる．

並列 **SA** アルゴリズム

収束曲線は不安定のため，終了の判定は難しくなる．一般的な判定方法は，一定回数の間最低温度を達成すれば，その時点の結果が最終結果なる，最高や最低温度の決定すること，最低温度を達成する回数など，さまざまなパラメータを選択することが難しい．

4.2 並列対交換 ·SA 併用アルゴリズムの提案

前節の比較分析によると，並列対交換とSAアルゴリズムは相補性がある．新しいアルゴリズムの思想は並列対交換法が前半で実行スピードが速く，最適化配置状態への収束率が高いため，対交換方法を基本処理方法として利用する．一方，SA法は優秀な結果を出すため，補足処理方法として利用する．

4.2.1 併用理由と問題点の解決方法

並列対交換アルゴリズムはSAの最適化テクニックを用いた対交換用いて並列SAアルゴリズムとなる．しかし，本論の提案はその簡単な採用ではない．すべての自動配置及び配線CADの目標はスピードと優良な結果である，最適化テクニックを利用したアルゴリズムはスピードを犠牲にして，優良な結果を追求する．結果変化が一般には波形のような最適結果へ収束する．SAもこのアルゴリズムの一つである[9]．セルの交換は評価関数より，改善がある時，交換する．改善がない場合はSA特有の関数によって，改悪も行う，しかし，全体的に，改善の趨勢となる．最適化テクニックを利用しても，前の配置状態を利用し，計算を続ける．

第2章で提案した並列対交換アルゴリズムは，通信回数は少ない，ローカル領域の配置方法は自由交換で一回の交換を実行するとき，最低は一回の通信が必要である，AP1000には一回の通信を利用した時間は100から800ぐらい回数数学計算の時間と同じことである．さらに，プロセッサの通信の待ち時間も入れば，非常に多くな時間である．自由交換，再配置方法を利用した通信回数は，本論の並列対交換より十分多くため，対交換法は自由交換より速い．しかし，対交換は処理スピード速くけど，交換の制約性があるため，終点前にエネルギー(交換要求がない)はなくなる，計算は止まる．

本提案の中心は，対交換のにエネルギーはなくなるまでできるだけ対交換の

実行高速能力を発揮し，他の方法より速くに，比較良い配置状態を達成する，次に他の方法で続く実行する，この時，対交換のエネルギーを蓄積する，エネルギー回復すれば，もう一度実行する，このような繰り返せば，新しいアルゴリズムのスピードも配置結果もよく期待する．これは併用の理由及び発想アイデアのスタート点である．しかし，同じ評価関数用いて，同じ評価計算結果ても，配置状態違うこともある，同じ評価結果に，SA と対交換生成する二つの配置状態の異なることは，本論に調査しなかった．それでも将来の問題である．

以下はこの二つの新しいアルゴリズムについて，両方法を併用するときに，問題点となるとその解決方法を説明した．

(1) 評価関数の採用と温度スケジュール

SA アルゴリズムの評価関数式は対交換アルゴリズムの評価関数と一致になるため，併用アルゴリズムの評価関数も式 (1.3) を採用をする，しかし，SA 法特有な温度スケジュール関数，アニーリング関数などは，利用しないといけない．

特に温度スケジュール対して，温度は実行回数により変化するの必要がある，対交換法を実行する時に，温度も変化しないといけない，しかし，対交換法は隣接するセルしか交換しないので，対交換を獲得される配線の減少量は自由交換と異なる，この実行回数による，温度スケジュールを実行すれば，温度の下ろすスピードと SA 法を実行する時にのスピードは異なる．我々は実際の実験結果による，データ A(セル 784) とデータ B(セル 4087) ランダム実行段階で自由交換と対交換の倍率比較は

実行段階 (回数)	1-100	200-400	400-500	平均
データ A	2.5	1.6	2.3	2.2
データ B	3.1	2.1	2.3	2.4

平均として，対交換 2 回での配線の減少量は自由交換 1 回分の減少量程度である．したがって，計算を簡単するため，対交換 2 回実行毎に，温度スケジュールは自由交換 1 回分温度変化をさせる．しかし，実行回数だけで温度スケジュールをするのは，十分正確ではない，実際には配置データにより，変化量も違うと予想される．したがって，温度スケジュールに対する，新しいのスケジュール方法が必要である，SA 法は万能な温度スケジュールがまだ定義されてないので，これは SA アルゴリズムや本手法の欠点でありは将来の課題である．

(2) プロセッサにおける配置データの分配

配置データの分割及びプロセッサの担当領域は，第 2 章の対交換と同じ分配

方式を用いる，本研究で利用した並列 SA アルゴリズムも同じ分配方式であるので，この二つのアルゴリズムを併用する場合は，利用アルゴリズムを切り替える時にデータをプロセッサ担当に再分配することを回避られる．

(3) 二つ処理方法の実行時点

本アルゴリズムの基本的な思想は，対交換と SA アルゴリズムを配置問題に対して，切り替えて利用する．しかし，簡単なクロックにより二つのアルゴリズムを平均回数だけの切り替えて利用することではない，そうすれば，SA 法のスピードが遅いという特性のため，併用法アルゴリズムは，実行するために大量な時間がかかる．したがって，スピード向上のため，適当な交換節点を決定することも重要である．前節の並列対交換と SA アルゴリズムの分析により，対交換アルゴリズムを実行する場合は，終点の無改善まで，同じ時間内の配置結果の改善率は SA よりよい．この理由により，以下のようにアルゴリズムの切替えを行う：

(1) 併用法アルゴリズムの開始点から，対交換を採用する．
(2) 対交換無改善になると，SA 法による処理を開始する．
(3) 一定回数の SA 法を実行した後に，その新しい配置状態で，対交換法に戻る．
(4) 十分有効な解が得られれば終了，そうでなければ (2) にもどる．

(4) SA 法の実行必要回数

SA 法による処理段階で，SA 法の実行必要回数を定義することが必要である．対交換アルゴリズム結果分析から，最適化結果を出さないの理由は配置セルは実行の最後階段に，エネルギーが足りないので，交換要求が不足する，交換を続けるためには，配置状態が一定量変化を必要ことがわかる．しかし，この変化が大きくなれば，SA の実行回数が増える，対交換の実行の時間もかかる．アルゴリズムのスピードも遅くなる．変化が小さ過ぎてもまた交換要求不足になる．しかし，配置問題あるいは対交換アルゴリズムにも一つのセルを移動すれば，このセルの移動により，多くのセルは連鎖交換する発生率も高い．SA 法の実行回数は適当な多きくないの場合と多数の交換実行回数が対交換のエネルギーに対する，同じ更にもど大きいな影響を与えることができる．本論には利用したデータに数回簡単なテストから 5%の交換を達成する時の実行回数が SA 法の実行必要回数となる．実際による，SA 法を実行する場合，交換と交換しない両方のことがある，このことの計算も難しいである，したがって，全セルの

8%は交換計算の実行回数はSAの実行回数となる，以上の二つのパラメタはいずれも経験数字である，特にSAの実行回数は厳密な実験と理論証明が必要である．さらに，実行回数は温度と結果よりオンラインに決定するアルゴリズムの開発も必要である．本論は時間とデータの制限により，パラメタの厳密定義をしなかったため，この不厳密のパラメタは後の併用アルゴリズム実験結果に対し，一定な影響を与えることも予想された．

(5) 終点の判定

SA法と対交換法を併用することにより，終点の判定は難しくなる．本研究には併用アルゴリズムの終点は経験により，3回両方を繰り返すた結果，改善されない場合は終了し，前の優良結果は最終結果とする．しかし，この停止方法も経験的なものであり，SAの複雑性による，今の停止点のすぐ後で優良な結果出るの可能性は大きい．

4.2.2 並列対交換・SA併用アルゴリズム

図4.3に対交換・SAの併用アルゴリズムを示す．

図4.3のように，並列対交換とSAの併用アルゴリズムは，対交換実行段階とSA実行段階という，二つの段階から構成される，対交換実行段階では2章で提案した並列化手法により対交換を実行し，一定台数のプロセッサが利用すれば，自分が担当するの配置領域内では通信なしで実行できる，スピードが十分速い．改善率が悪くなるかあるいは無改善の場合に，SA実行段階に入る．並列SA法では第3章の方法より，一回の交換実行毎に通信を必要とする，しかし，本手法でSA法実行回数少ないため，SA法の実行時間は，全アルゴリズムの実行時間に対し，大きいな影響を与えない．一定回数のSA法を実行することにより，新しい配置状態を生成できる，この配置状態は，単なる変化ではなく，温度SAテクニックによる，最適化へ収束可能性がある状態である，さらに，この状態は対交換法に対し，エネルギーを入ったことより，新しい移動が可能性が提供する．もう一度対交換法を高速実行する．この繰り返してより，スピードの向上と共に，SA法に近い優良な結果が得られる．次の節は並列対交換・SA併用アルゴリズムの性能評価結果と優良性を実証する．

4.3 並列対交換・SA併用アルゴリズムの性能評価

図4.4に対交換・SA併用法，SA法，対交換法についての配線長の変化を示

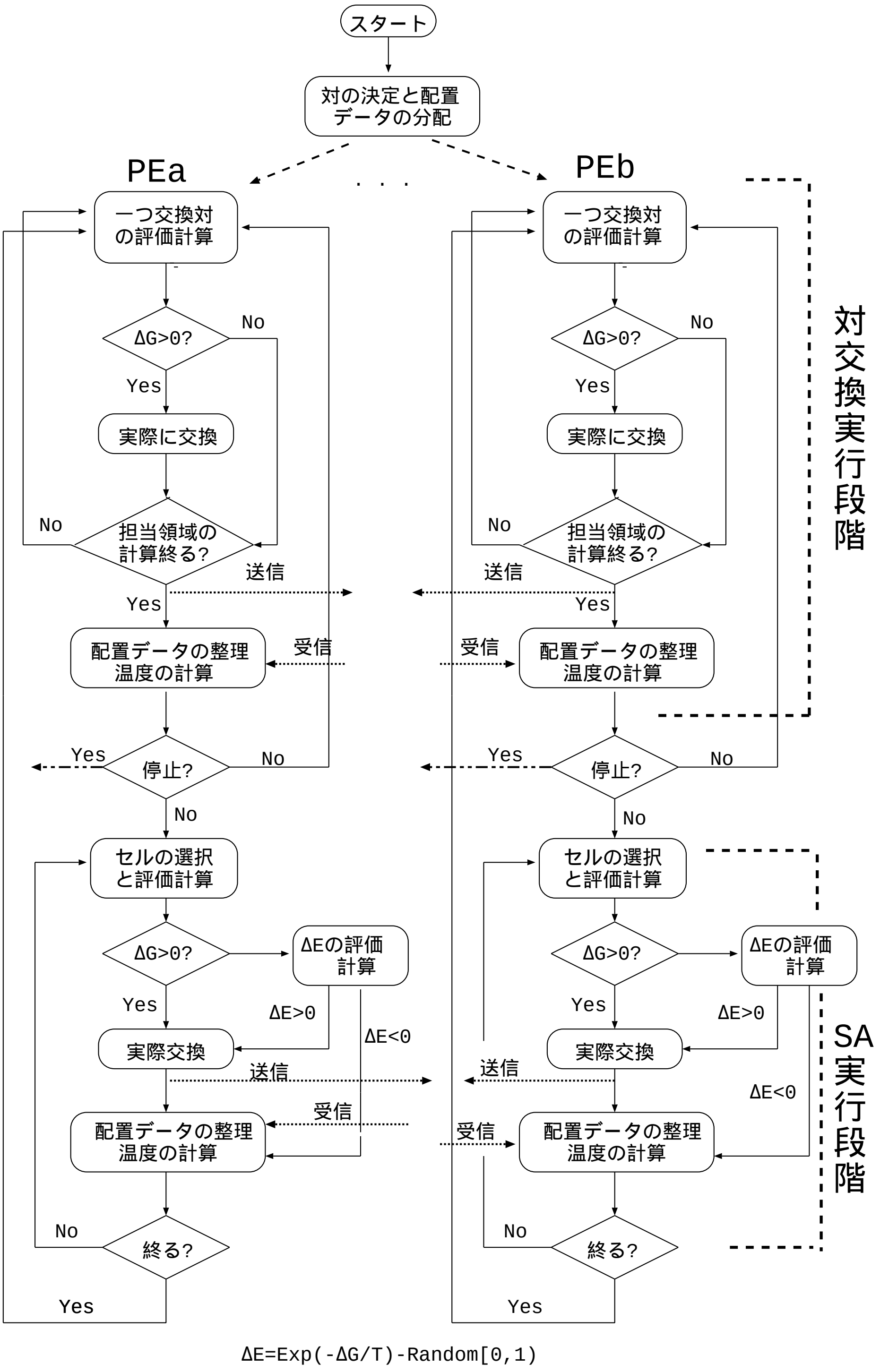


図 4.3: 並列対交換と SA 併用アルゴリズムのフローチャート

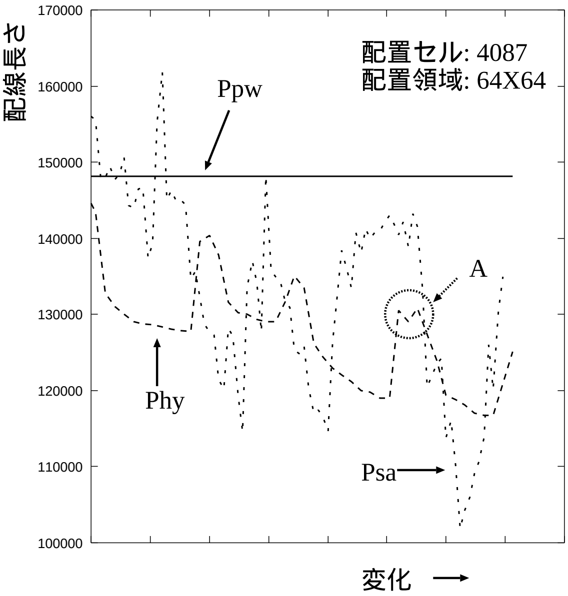


図 4.4: 併用法と SA，対交換法の結果比較

す，併用法と SA 法の結果変化は実行最終段階の拡大したものであるある．

Ppw ：対交換法の最終結果

Phy ：対交換・SA 併用法の終了段階での配線長変化

Psa ：SA 法の終了段階での配線長変化

図 4.4 より，併用法の結果は対交換より，大きく改善されることがわかる．その改善は，二つのアルゴリズムを切替えての利用による，規則的な波型の変化をしている，波の山は SA 法を利用している段階，山から谷へ移る部分は対交換を利用しているに相当する．A 点は SA 法改善と改悪両方を行うの特点を示す．極大点付近と極小点付近はそれぞれの SA 法の実行点及び対交換の実行停止点となっている．併用法では，一定回数の SA 法を実行し，さらに，併用法のスピード向上を目指しため．SA の実行回数が多くないので，一般的に，SA 法実行の段階は，改悪となる，しかし，盲目的な改悪ではなく，最適化テクニックによる改悪であるため，この結果に対し，対交換を実行すれば，前回の極小点よりも良い結果を出すことができる．

併用法の実行回数は$\to \infty$ 併用法中に SA の利用回数も$\to \infty$ 温度とパラメタを正し定義すれば，理論的に SA 法と同じ優良な結果出すことができる．しかし，併用法で最適結果を出すには，実行回数が非常に大きくなるため，, 最適結果の検出することは難しい．

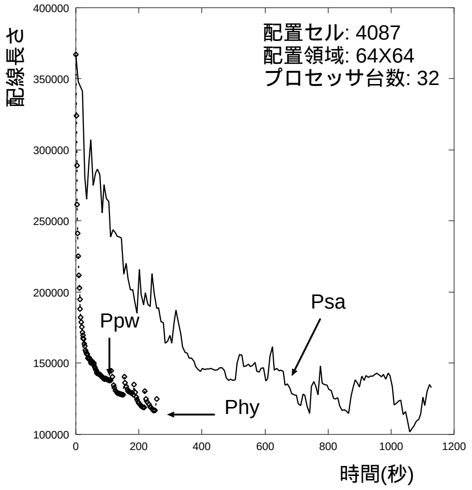


図 4.5: 併用法のスピード調査 1

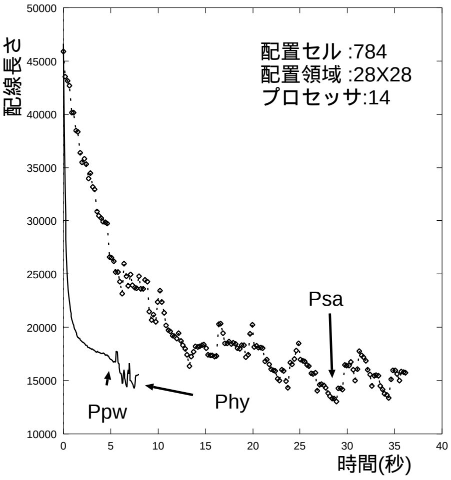


図 4.6: 併用法のスピード調査 2

併用法の改善能力につては，一般的に，対交換より良い，SA より少し劣る，配置結果は対交換と SA の中間で，SA に近い．

以上の結果から，併用アルゴリズムは，優良な配置結果を出すことは明らかである．

図 4.5図 4.6に並列併用法と対交換，SA 法の実行スピードを示す．

Ppw :並列対交換で結果出す時点，あるいは対交換の実行を必要時間である．

Phy: 並列併用アルゴリズムの実行時間である．

Psa: 並列 SA アルゴリズムの実行時間である．

最終配置結果を出すまでの時間は，並列併用アルゴリズムは SA より，十分速いこと (5 倍ぐらい) がわかる．並列対交換と SA 法は同じ結果出すまでの時間は対交換十分速いので，対交換の最終配置結果から併用を実行する良いことはは明らかである，その結果，対交換により，並列併用アルゴリズムのスピードは遅くなる．しかし，この速度低下は小さい，したがって並列併用アルゴリズムは優良な配置結果を出し，スピードも向上する．

図 4.7にことなるデータによる併用法のスピードアップを示した．

図 4.7は並列併用アルゴリズムと対交換，SA など様々アルゴリズムのスピードアップの比較調査である．

Phy1 :併用アルゴリズムで配置セル 4087，配置領域 64×64

Phy2 :併用アルゴリズムで配置セル 784，配置領域 28×28

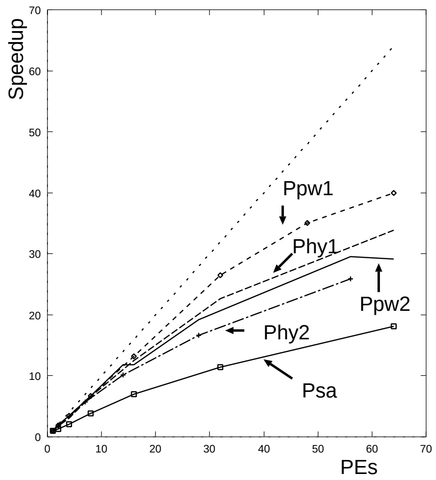


図 4.7: スピードアップ比較

Ppw1 :対交換アルゴリズムで配置セル 4087，配置領域 64×64

Ppw2 :対交換アルゴリズムで配置セル 784，配置領域 28×28

Psa ：SA アルゴリズム (3 章で表現したもの)

図 4.7により，対交換の通信回数は相対的に少ないので，スピードアップは一番良い，併用アルゴリズムは対交換を利用する回数多くので，SA より非常に良いスピードアップが得られた，Ppw1，Ppw2 と Phy1，Phy2 のスピードアップ連線も似ているの形である，しかし併用アルゴリズムは一定回数の SA を利用したため，通信回数を増加し，スピードアップは対交換よりよくないことも明らがである．SA アルゴリズムは一回交換あと通信の必要あるから，スピードアップは通信回数多くので，よくないことも明らがである．

5　並列対交換$^+$アルゴリズムの提案

前章は対交換と SA の併用アルゴリズムを提案した，性能評価により，このアルゴリズムは SA よりスピードが大幅向上し，結果も対交換より大幅な良になる．しかし，このアルゴリズムは SA 法を採用することより，欠点がある，

- SA アルゴリズム利用した様々なパラメターの選択すること．
- 併用のため定義したパラメータの不厳密性．
- 最終結果の判定すること．

本研究では対交換 ·SA 併用アルゴリズムのパラメータの選択は一部が経験数字と実験により数字である (第 3，4 章の説明より)，特に最終結果の判定が難しい

による．本章は対交換法について，対交換とSAを併用アルゴリズムと同じ思想から，対交換アルゴリズムをベースとした，対交換$^{+}$アルゴリズムを提案する．

5.1　対交換$^{+}$アルゴリズム

対交換$^{+}$アルゴリズムの基本思想は対交換とSAを併用アルゴリズムと同じである，最初段階は対交換アルゴリズムを利用する，対交換で無改善となる場合は，対交換の停止する理由の分析により，図4.2 (c)の状態に対し，隣接しない，配置セルを交換する，これを以下に最優位置へ交換と呼ぶ．最優位置の計算により，最優位置への移動範囲を生成をする，移動範囲は直線と方形である．

並列対交換$^{+}$アルゴリズムの実行ステップは以下に示す．

(1) スタートステップ

各々のプロセッサスタート点から対交換を実行する．

(2) 対交換ステップ

対交換を実行する．対交換で無改善時に最優位置への交換ステップに入る．その時に各配置セルは最優位置への移動範囲を計算する．

(3) 最優位置への交換ステップ

一定の配置領域内，一定個数の配置セルを選択し，配置セルの交換相手を決定し，自由交換を行う．この後に，新しい位置の放送と受信，新しい位置による，最優位置への移動範囲の更新，一定回数の優位置への交換を実行するを行い終るともう一度(2)対交換ステップを入る．

(2)..(3)ステップを繰り返し実行し，(3)最優位置への交換ステップでも交換できる配置セルはなくなる場合に，計算を停止する．

5.2　並列対交換$^{+}$アルゴリズム問題点及び解決方法

並列対交換$^{+}$アルゴリズムにも，以下の問題点を存在している．

(1) 最優位置への交換の実行回数

このアルゴリズムは対交換とSA併用アルゴリズムと同じように，最優位置への交換でも1回実行する毎にすぐ新しい配置情報を放送しないといけない，したがって，多くの回数最優位置への交換を実行すれば，アルゴリズムのスピードが下がる，また，回数少ないても対交換に戻る時に，交換要求が不足し改善率低下である．適当な最優位置へ交換を実行回数の決定することが必要である，本アルゴリズムでは経験的な理由と対交換 ·SAを併用アルゴリズムと評価のた

め同じ 8%確率を採用した．

(2) プロセッサの担当領域と必要な配置データその問題に対し，以下の定義がある．

(2.1) 本アルゴリズムでは，プロセッサが持つ配置データは対交換での必要データと同じ．

(2.2) 本アルゴリズムで最優位置への交換の実行する時に，図 2.10 のように，配置データはプロセッサ重複を持ちことがある，実行回数により，図 3.4のような横縦，さらに大きさを変換をする，最優位置への交換相手の選択の領域となる．

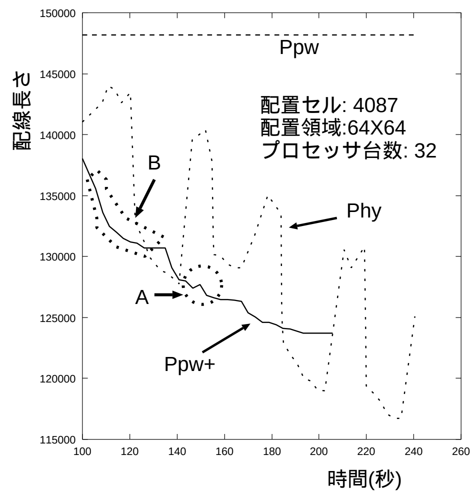
(2.3) 本アルゴリズムでの最優位置へ交換相手の選択方式は，物理位置と自分一番近いの交換が可能のセルは交換相手になる．

(2.4) プロセッサは自分担当したの配置領域しか最優位置への交換をしない．

様々の配置セルは，セルの接続関係により交換相手の選択可能領域も違う．ある配置セルの選択可能領域は，プロセッサの持つ配置データの範囲以外の可能性もある，特にプロセッサの担当配置領域の隣接したセルはそのことがよく発生する．セルはプロセッサ範囲以外だけ交換を可能する場合は，プロセッサがその配置データを持たない時に，交換をできなくなる．(2.2) と (2.3) の定義と本アルゴリズムの配置領域あるいは配置情報を担当方式により，配置セルはプロセッサを持つ配置データの範囲以外交換のことを可能となる．プロセッサ重複を持ちの配置領域は異なるプロセッサ間にの交換の中継点になる可能性がある．横，縦の交換利用により，セルも一定程の自由となる．しかし，セル本当に自由になるため，2 回の移動必要である．

(2.4) の定義による，本アルゴリズムは一定な制約が起こる．最適化結果を追求するため，一回の交換だけで自分プロセッサを担当した以外の領域のセルと交換したい，この時に，プロセッサ間に通信しないといけない，しかし，このことにより，通信回数を大幅に増加する，2 回以上同じ PE の情報を必要時に (特に隣接するプロセッサ間にこのこと多い)，通信があるいは第 1 回通信の時に相手の情報全部持ってくる，通信すればスピードが遅くなる，情報を持ってくれば，とのくらいの情報量を増えることは予想ができない．停止時点も探しにくくなるなど様々な複雑問題が出てくる，特に，スピードはものすごく遅くなるの可能性が大きい，様々の問題による，この問題を避けられる，(2.4) を定義した，停止の時点も探しやすくなる．

最適化位置へ交換は自由交換と同じ，同じ時間に 2 個，さらに 2 個以上接続



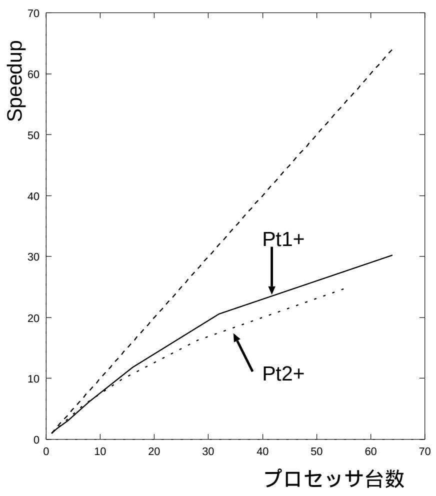


図 5.1: 対交換$^{+}$アルゴリズム結果調査

図 5.2: 併用法のスピードアップ

するセルを選択することが発生の可能性がある．この時に，エラーでるの可能性もあった，しかし，併用の主に目的は対交換のエネルギーを増加することである，さらに一つのエラーは全面に大きいな影響が与えないので，本論に無視された，図 5.1 の A 点はエラーを出た，しかし，全面は改善を行う，大きいな影響が与えなかった．

5.3 対交換$^{+}$アルゴリズムの性能評価

図 5.1にアルゴリズムを切り替えてする段階に，対交換$^{+}$アルゴリズムの結果およびその結果と対交換，対交換とSAを併用アルゴリズムの時間と結果比較．
図 5.2に並列対交換$^{+}$アルゴリズムのスピードアップを示す 図 5.1には ：
Ppw: 対交換法 Phy: 対交換とSAを併用アルゴリズム
Ppw^{+}: 対交換$^{+}$アルゴリズム
図 5.2には ：
$Ppw^{+}1$ 配置セル 4087 配置領域 64×64
$Ppw^{+}2$ 配置セル 784 配置領域 28×28
図 5.1により，対交換$^{+}$アルゴリズムのスピードアップは対交換 ·SAを併用アルゴリズムと変わらない程である．

図 5.1より対交換$^{+}$アルゴリズムの結果は対交換法より優良性がある．対交換・SAを併用アルゴリズムより結果が良くない．対交換$^{+}$アルゴリズムは最優位置

へ交換ステップ後に，同じ配線長さでも対交換の改善率も対交換・SA を併用アルゴリズムより低いことも明らかである．

図 5.1のような対交換$^+$アルゴリズムに対交換を実行段階は，対交換・SA 併用の対交換実行段階の配線改善率より遅い，その理由は：

最優位置への交換により交換したセルは，最優位置あるいは最優位置に近い位置に配置された，それにより，この配置セルの隣接するセルとの交換要求も，最適化テクニックを利用した SA 法の交換による配置するセルより，低下している．全面には Ppw^+中に対交換のエネルギーは Phy より小さい．しかし，対交換$^+$アルゴリズムでは最優位置への交換はまず近い交換を行えるセルから交換をするので，最初のアルゴリズムを交換利用する時に図 5.1B 点のような，相対高速対交換の実行ができた，さらに，この時に最優位置へ交換をできるセルも多い，最優位置へ交換と対交換の交換合わせて，相対高速化の改善を一度達成した．

対交換の交換は連鎖反応の状態は多く，一つの交換を達成すれば，多くの交換更新の可能性が提供する．しかし，(2.4) の制限のため，対交換$^+$ アルゴリズムの実行後半は，最優位値への交換，対交換両方ても改善力がよくない，遅いスピードの改善がある．

しかし，この方法の現時点の利点は

(1) 対交換・SA を併用法欠点の一部が避ける，特に温度スケジュール必要なパラメタ．

(2) 対交換$^+$アルゴリズムは単調改善 (エラーがない時) であるため，停止点は対交換・SA 併用アルゴリズムより判断しやすい．定義された最優位値へ交換の利用回数と利用範囲は停止ための制御関数として利用できる．利用回数が小さいなれば，早く止まる，前節 (2.2) の定義より，最優位値への交換は交換相手が自分の回りから始め，一つのセルに対し，多数の移動可能性が生成した．利用回数と利用範囲の定義より，どの程度の結果が出すことと停止することは判断をできる．

(3) このアルゴリズムも対交換・SA を併用と同じ利点がある，対交換と最優位値への交換はプロセッサのメモリの占有と担当配置領域は変わらない，まだ，重複領域を，中継点として利用ができる．

以上の理由により，対交換$^+$アルゴリズムを提案した，しかし，このアルゴリズムのスピードアップは併用アルゴリズムあまりかわらない図 5.2 結果も併用

アルゴリズムの方がよいので，本論にこのアルゴリズムは対交換とSAを併用アルゴリズムの補足として，提案した．

6　おわりに

従来，多くの研究者は自動配置問題に対し，新しいアルゴリズムを開発する時に，最適化テクニックの利用と新しい配置状態の生成方式を重点として研究してきた．本研究では，現有のアルゴリズムから，様々利点を抽出し，できるだけ欠点を回避でき方複数アルゴリズムを結合したアイデアとして，対交換・SA併用及びその補足である対交換$_+$アルゴリズムを提案した，配置問題は複雑な最適化問題である，それに対する，従来様々のアルゴリズム提案した．しかし異なる配置状態により，アルゴリズムの処理能力も異なる．配置問題に実行段階による，適当にアルゴリズムを利用する．複数アルゴリズムを結合利用ことは，単独一つアルゴリズムを利用場合より，良い結果を得る可能性が高い．

この研究一定の成果が得られたが，次のような課題は残されている．

(1) 並列対交換について

(1) 本研究で提案した並列対交換アルゴリズムでは，プロセッサの利用台数が配置領域の横あるいは縦の$1/2^n$となる時に，自分の計算担当領域の計算を終るまで，通信の必要はない．最高スピードアップを達成する可能性がある．しかし，本論のアルゴリズムでは各プロセッサはほどんど同時に通信，スピードに影響を与える，このこと避けるため，新しい通信方式，計算量の最分配が必要である．

(2) プロセッサの利用台数は配置領域の横および縦) の長さはN_PやN_P-1 プロセッサの利用台数の2^nならない場合，あるいは配置領域は2次元ブロック分割した場合同じ対行を担当したプロセッサの通信が必要である，したがって通信の減少および負荷の均等化が必要である．

(2) 並列**SA**について

(1)SAでは配置状態更新の手法は自由交換を採用したため，自由交換の並列化がSA並列化の中心となる，今回の並列自由交換の提案ではセルの重複選択するを避け高速化するため，一回の実行中に，交換相手は自分担当配置領域中で選択する必要があり，セルの交換は十分自由にはなてない．このような交換と本当の自由交換の優劣は多種類データを用いて，評価することが必要である．されに，本論のSA並列化提案は満足できる提案ではなく，新しい提案が必要

である．

(2) 温度と温度スケジュール方法の更新すること．

(3) 並列対交換と SA を併用することについて

(1) 様々なアルゴリズム切替え時点に，本論の提案より，良い結果得る可能性がある，併用の最適な時点の調査が必要である．

(2) 本論に SA の利用回数は経験数字である，この利用回数の調査及び再定義することが必要である．

(3) 自由交換と対交換で，データによる配置結果の変化は異なる，それによって，温度スケジュールの実行クロックとして，参照可能なもの (回数ではなく，全配置状態の変化量など) を探すことが必要である．

(4) 配置問題において，多く種類のアルゴリズムを併用すること

謝辞

本研究の機会を与えてくださった富田眞治教授に，深甚な謝意を表します．

適切な御指導を賜わった中島浩助教授，森眞一郎助教授および数々の有用な御意見を戴いた大野和彦氏に心から感謝致します．

また, 並列計算機 AP1000 の実行環境を御提供頂きました (株) 富士通研究所に感謝致します．

さらに，共同研究者である Hesham Keshk 氏と日頃御討論頂く富田研究室の諸氏に感謝致します．

参考文献

[1] D.J.Chyan and M.A.Breuer: A Placement Alorithm for Array Processor, *Proc 20th Design Automation Conf.*, pp. 182–188 (1983).

[2] K.Ueda: A Parallel Module Placement Approach for Logic Module Placement, *IEEE Trans. CAD-2(1)*, pp. 39–47 (1983).

[3] E.I.Horvath: A Parallel Force Dircted Standard Cell Placement Algorithm, *Technical report,Dept.Computer Science,Florida Altlantic University,Boca Raton,Fl*, p. 1 (1992).

[4] P.Banerjee, M.: Parallel Simulated Annealing Algorithm for Standard Cell Placement on Hypercube Multiprocessors, *IEEE Trans Parallel Distributed System1(1)*, pp. 91–106 (1990).

[5] R.M.Kling: Placement by Simulated Evolution, *IEEE Trance CAD*, Vol. 8, pp. 245–256 (1989).

[6] J.P.Cohoon: Genetic Placement, *IEEE Trance CAD*, pp. 182–188 (1987).

[7] R.M.Kling and P.Banerjee: ESP: Placement by Simulated Evolution, *IEEE Trans CAD 8(3):*, pp. 254–256 (1989).

[8] S.Mohan and SP.Mazumder: Wolverines: Standard Cell Placement on a Network of Workstations, *IEEE Trans CAD 12(9):*, pp. 1312–1326 (1993).

[9] Banerjee, P.: *Parallel Algorithms For VLSI Computer-Aided Design*, PTR Prentice Hall (1994).

[10] J.Rose, W.: Temperature measurement and equilibrium and dynamics of simulated annealing placement, *IEEE Trance CAD9(10):*, pp. 253–259 (1990).

[11] 龍和男: 並列 LSI-CAD システム技術に関する調査研究報告書, 財団法人新世代コンピュータ技術開発機構 (1995).

[12] K.Shahookar and P.Mazumder: VLSI Placement Techniques, *ACM Computing Surveys,23(2):*, pp. 143–220 (1991).

[13] J.Rose, W.M.Snelgrove, a. Z.: Parallel Standard cell Placement Algorithm with Quality Equivalent to Simulated Annealing, *IEEE Trance CAD*, pp. 378–396 (1988).

[14] S.Kirkpatrick, C.: Optimization by Simulated Annealing, *Science,220:*, pp. 671–680 (1983).