

疎結合並列計算機上でのシミュレーテッド・アニーリング

荒木均 館野峰夫 加藤等 間藤隆一

松下電器産業株式会社 情報通信東京研究所

シミュレーテッド・アニーリング法を並列化した新しい手法を提案する。本手法は、状態を複数のプロセッサに一つずつ割り当て並列にアニーリングを行なうもので、次の二つの特徴を持つ。一つは複数プロセッサから得られるコストにより平衡状態を判断することであり、もう一つは、アニーリングの過程で極小解に陥ってしまったプロセッサだけが他のプロセッサが持つ状態を受信し、その状態からアニーリングを開始することである。前者は高温での変換回数を削減し、後者はより高い確率でコストが小さい状態に収束することができる。本手法を 16 プロセッサの疎結合並列計算機マルチ PSI 上で実験を行ない、約 10 倍の処理速度が得られることを確認した。

Simulated Annealing on a Loosely-Connected Parallel Computer

Hitoshi Araki Mineo Tateno Hitoshi Kato Ryuichi Mato

Tokyo Information and Communications Research Laboratory
Matsushita Electric Industrial CO.,LTD.

3-10-1, Higashimita, Tama-ku, Kawasaki, 214 Japan

This paper proposes a parallel formulation of Simulated Annealing Algorithm. This algorithm first allocates a state to each processor and the processors begin the annealing in parallel. This algorithm has two features. One is the detection of equilibrium by examining the cost values of the processors. The other features is that the processor trapped in a local minimum in the annealing process receives a state from another processor, and starts the annealing again. This algorithm is implemented on a loosely-connected parallel computer consisting of 16 processors, Multi-PSI. We have observed a factor of 10 speedup in real time.

1 はじめに

シミュレーティッド・アニーリング (SA) 法を並列化した新しい手法を提案する。SA 法は、各種組合せ最適化問題に適用可能な手法であり、最適解を見つけることが可能であるが、膨大な計算時間を要することが問題であった。

本手法は、状態を複数のプロセッサに一つずつ割り当て並列にアニーリングを行なうもので、短時間でより最適な状態を見つける二つの特徴を持つ。一つは複数プロセッサから得られるコストにより平衡状態を判断することであり、もう一つは、アニーリングの過程で極小解に陥ってしまったプロセッサだけが他のプロセッサが持つ状態を受信し、その状態からアニーリングを開始することである。前者は高温での交換回数を削減し、後者はより高い確率でコストが小さい状態に収束することができる。

本手法を論理アーキテクチャ設計問題を例に採り、(財)新世代コンピュータ技術開発機構 (ICOT) の疎結合型並列計算機マルチ PSI 上で通常の SA 法と比較実験を行なった。その結果、16 プロセッサのマルチ PSI では、約 1/10 の処理時間で同程度の解を得られることを確認した。

2 SA 法

SA 法 [Kirkpatrick 83] は、物理現象である焼きなましを計算機上で模擬することにより、エネルギー (コスト) 最小の状態を求める手法であり、各種組合せ最適化問題に適用可能な最適化アルゴリズムの一つである。そのアルゴリズムを以下に示す [木村 90]。

SA 法は適当な初期状態 s_0 から出発し、状態の系列 $\{s_n\}_{n=0,1,2,\dots}$ を次のように順に生成し、コスト最小の状態 (最適解) に近づけていく。まず、 s_n に微小な変換を施した s'_n を生成し、コストの差 $\Delta c = c(s'_n) - c(s_n)$ を計算する。 $\Delta c \leq 0$ ならば、 $s_{n+1} = s'_n$ とし、 $\Delta c > 0$ ならば、 $p = \exp(-\Delta c/T)$ として、確率 p で $s_{n+1} = s'_n$ 、確率 $1-p$ で $s_{n+1} = s_n$ とする。ここで T は温度パラメータと呼ばれ、交換回数 n に依存して、高温から低温に徐々に変化させる。これをクーリング・スケジュールと呼び、温度一定で一定回数変換を繰り返し、 T を段階的に下げていくのが一般的である。これは最も単純

なクーリング・スケジュールであるが、比較的良い結果をもたらすことが知られている [Kirkpatrick 83]。

温度一定の時、SA はエネルギー平衡の状態に向かう統計熱力学的な系の挙動としてとらえることができ、無限回変換を繰り返す極限では、どのような状態分布から出発しても、状態 i となる確率が $\pi_i(T)$

$$\pi_i(T) = \frac{1}{Z(T)} \exp\left(-\frac{c_i}{T}\right), \quad (1)$$
$$Z(T) = \sum_{i \in S} \exp\left(-\frac{c_i}{T}\right)$$

である平衡状態の分布 $\pi(T)$ に収束する (ただし、 c_i は状態 i のコスト)。 $\pi(T)$ は、 $T \equiv \infty$ のとき一様分布であり、 T を小さくしていくと、コストが小さい状態になる確率は大きくなり、 $T \rightarrow 0$ では、コスト最小である確率が 1 である分布となる。したがって、各温度で平衡状態に十分近づくことができる程度に変換を繰り返し、温度を 0 に近づけていけば、コスト最小である状態に収束することができる (図 1 参照)。以下では、この収束過程を大きく三段階にわけて説明する。ただし、各段階は明確な境界を持つものではなく、段階の移行は徐々に行なわれる [山下 89]。

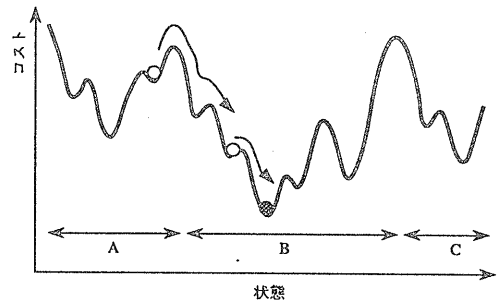


図 1: SA の収束過程

○ 高温部

十分大きな温度では、SA はランダムウォークのようにふるまう。そのため、図 1 の大きなコスト障壁も簡単に越えることができ、自由に遷移することができる。すなわち、高温での平衡状態の分布 $\pi(T)$ は一様分布に近く、すぐに平衡状態に達

することができるので、変換回数を多くしてもあまり意味はない。

● 中温部

アニーリングが進み、温度が徐々に下がってくると、やがて、大きなコスト障壁を越えることは困難になってきて、図1の大きな谷(例えば、A,B,C)のどこかに落ち込んでいく。これは、式(1)で示される状態 i になる確率 $\pi_i(T)$ に差が生じ、コストの高い状態になる確率が小さくなるためである。この $\pi_i(T)$ に差が生じてくる中温部で平衡状態に達することができるかが最終的な収束状態を大きく左右する。したがって、この中温部に多くの変換を施し、慎重に収束先の状態を決定することが重要である。

● 低温部

低温部では、多くの変換がリジェクトされてしまい、状態の大きな変化は望めなくなる。すなわち、図1の大きな谷から、飛び出すことはもはや不可能になってきて、大域的最小値へと収束していく。逆に、中温部で十分な回数変換が行なわれていないと、局所的最小値に収束してしまう。

上記のような収束過程から、SA法は、各温度で平衡状態に十分近づくことができる程度に変換を行えば、最適解に近い状態を見つけることができる。しかし、一般に、どの程度変換を行えば、平衡状態に近づくことができるかは解明されていない。そのため、通常は、一定回数の変換を行なうことが多いが、その回数は経験的な値である [Laarhoven 87]。また、一定回数の変換以外にも、受理回数(新しい状態が採用された回数)がある値を越えるまで変換を行なう方法 [Kirkpatrick 83] や、アニーリングの過程で得られたコストの最小値と最大値から計算することも提案されている [Romeo 85]。しかし、それらの手法が最終的な解にどの程度影響を及ぼすかは十分検討されていない。

SA法は、他の最適化アルゴリズムと比較して、より最適な解を見つけることができるが、多大な計算時間を要することが問題であった。そこで、SA法を並列化して、高速化しようという研究も行なわれている。

SA法の並列化として最もよく研究されているものは、状態を表現しているデータを分割して各プロセッサに割り当て、各プロセッサで並列に各部の変換を行なう方法である [Rose 86][Casotto 87]。この方法は、各プロセッサで正確なコストを知ることができないため、分割された各部の依存関係が大きい場合、得られる解が極小解に陥ってしまうことが問題である。

また、各温度での変換の開始時に、各プロセッサに同じ状態を一つずつ割り当て並列に変換を繰り返し、次の温度では、すべてのプロセッサの中でコストが最も小さい状態を各プロセッサに割り当て変換を繰り返す方法も提案されている [Aarts 89]。しかし、この方法で得られる解は、プロセッサ間通信を行わずに各プロセッサで独立にSAを行ない、最終的に得られる解の中で最もコストが小さいものとほとんど差がないと言われている [Aarts 89]。

これらの方法以外にも、各プロセッサに同じ状態を割り当て並列に変換を行ない、最初に新しい状態を受理することができたプロセッサが他のプロセッサにその状態を送信し、再び複数のプロセッサが同じ状態に対して変換を繰り返す方法もある [Aarts 86]。この方法は、高温では状態の受理率が高いため、実行効率が低下してしまう。そこで、同じ状態に対して変換を行なうプロセッサ数を、温度の低下とともに徐々に増加していくことにより、実行効率をあげている。しかし、一般に疎結合型の並列計算機では通信コストが大きいので、この方法を適用することは処理速度の低下になると考えられる。

これらの並列方法に対して、本手法は、状態を複数のプロセッサの一つずつ割り当て並列にアニーリングを行なうもので、次の二つの特徴を持つ。一つは、複数のプロセッサから得られるコスト分布により各温度での平衡状態を判断することであり、もう一つは、アニーリングの過程で極小解に陥ってしまったと判断されるプロセッサが、他のプロセッサが持つ状態を受信し、その状態から再びアニーリングを開始することである。この二つの処理により、高温での変換回数を削減し、より高い確率で最適解に近い状態に収束することができる。

3 並列 SA 法

上で述べた SA の定性的な収束過程から、高温部では交換回数を多くする必要はないが、中温部で交換回数が少ないと、図 1 の大域的最低値には収束せず、局所的最低値に落ち込んでしまう可能性がある。この可能性は交換回数が少なければ少ないほど大きくなる。そこで、高温から中温にかけては、複数プロセッサのコスト分布から平衡状態を判断することにより交換回数を削減し、中温から低温にかけては、アニーリングの過程で見つかったコストが小さい状態の近傍を複数のプロセッサで探索する以下のような手法を考える。

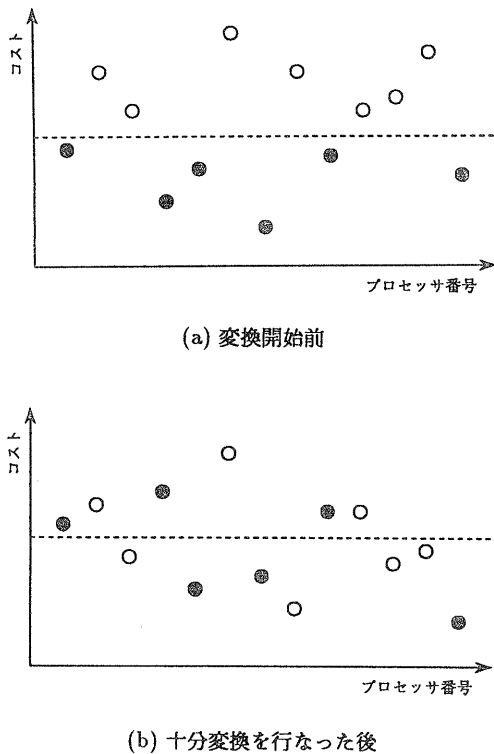


図 2: プロセッサのコストの分布

まず、各プロセッサに状態を一つずつ割り当て、同じ温度から並列にアニーリングを開始する。高温では、アニーリングを開始する前に、図 2(a) のように、コストの大小により大きいコストを持つグループと小さいコストを持つグループにプロセッサを二分する。そして、一定温度でアニーリングを行なうことにより、各プロセ

ッサに割り当てられた状態は変化し、図 2(b) のように二つのグループのコスト分布に違いがなくなってくる。この時、その温度での平衡状態に達したと判断して、全プロセッサの温度を下げ、再び大きいコストと小さいコストのグループにプロセッサを分けて、同様な処理を繰り返す。逆に、十分交換を施しても同様なコスト分布にならなければ、低温まで温度が下がってきたと考える。低温では、図 3 のように現時点 I において、全プロセッサの中で最も小さいコストを持つプロセッサ (図中、P8) の N 回前からのコストの最大値を基準として、その値よりも、 N 回前からのコストの最小値が大きいプロセッサ (図中、P1) は極小解に陥ってしまったとみなす。そのようなプロセッサは、他のプロセッサのコストが小さい状態を受信し、それ以後は受信した状態についてアニーリングを行なう。

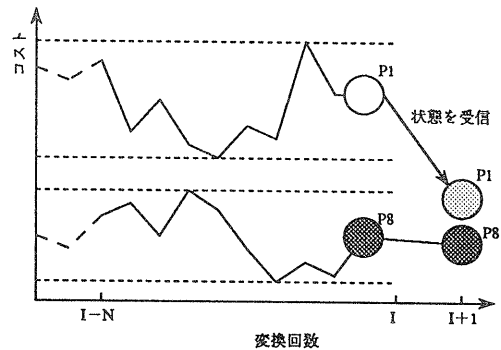


図 3: 極小解の判断

以上のような並列 SA 法は、高温部では二つのグループのコスト分布の違いがすぐになくなるので、交換回数を削減することができる。また、低温部ではコストが小さい状態に対して複数のプロセッサが SA を行なうことが多くなり、より高い確率でコスト最小の状態を見つけることが期待できる。更に、プロセッサ間通信について考えると、負荷が小さいコストの通信がほとんどで、負荷が大きい状態の通信は、極小解に陥ってしまったと判断される時にだけ生じる。そのため、プロセッサ間の通信に時間がかかる疎結合型並列計算機でもその性能を十分に発揮することができる。

以下では、コストの分布に違いがあるかどうかを判断するために用いた順位検定について述べ、その後、本

並列 SA 法の処理について詳しく述べる。

3.1 順位検定

順位検定 [竹内 75] とは、二つの母集団のサンプルから、二つの母集団の確率分布が等しいかどうかを判定する検定方法の一つである。例えば、二つの母集団 G_x , G_y から得られたサンプルが

$$(X_1, \dots, X_7) = (12, 27, 50, 61, 79, 91, 110)$$

$$(Y_1, \dots, Y_7) = (10, 18, 25, 30, 39, 85, 120)$$

とする。これを大きさの順に並べると、

$$yxyxyxyxxyxyxy$$

となる。このとき、図4のようなグラフを考える。原点 $(0, 0)$ から出発して、まず最初の値が x ならば右へ一目盛進み、 y ならば上へ一目盛進む。次に第2の値を見て、 x ならば右へ、 y ならば上へ進む。このようにして、点 $(7, 7)$ まで進む道を考えることができる。そして、図4の影の部分、すなわち 45° とこのグラフで囲まれる部分の面積を検定統計量にしたものが順位検定である。

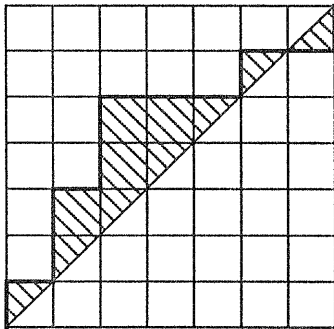


図 4: 順位検定

順位検定を用いた理由は、温度一定の SA では、変換を無限に繰り返せば、平衡状態に達した時に、初期状態に関係なく唯一つの平衡状態の分布 $\pi(T)$ に収束するためである。すなわち、コストの大小によりプロセッサをコストが小さいグループ G_x と大きいグループ G_y に分け、二つのグループのコストの分布に違いがなくなれば、平衡状態に達したと考えるわけである。

3.2 処理詳細

本手法では、温度の管理、順位検定、状態送受信の決定などを行なう一つの管理プロセッサ (以下では MP と呼ぶ) と、実際にアニーリングを行なう複数の状態生成プロセッサ (以下では GP と呼ぶ) が存在し、MP に一つのプロセッサが、GP にその他のプロセッサが割り当てられる。MP の処理の流れを図5に示す。

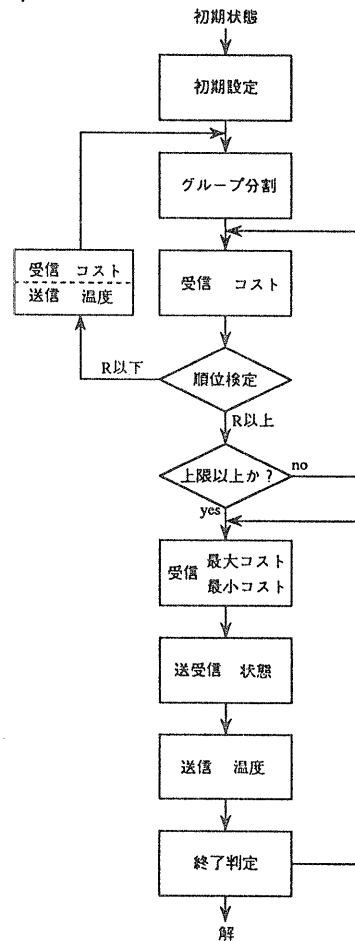


図 5: 管理プロセッサ (MP) の処理の流れ

初期設定では、MP は一つの初期状態を各 GP に割り振る。GP は温度無限大で一定回数変換を繰り返し、

異なる状態を生成する。そして、MP にそのコストを送信すると同時に、初期温度を受信し、再び変換を開始する。このとき、MP は送信されたコストにより、コストが小さいグループ G_x とコストが大きいグループ G_y の二つに分ける。

次に、MP は各 GP からコストを受信し、 G_x のコストと G_y のコストで順位検定を行なう。このとき、検定統計量がしきい値 R 以上ならば、まだ平衡状態に達していないと判断し、再びコストの受信を行なう。そうでなければ、平衡状態に達したと判断し、新しい温度を送信すると同時に、コストを受信する。その後、グループ分割部で、受信したコストにより再び二つのグループに分け、GP からのコストの受信を再開する。

ただし、変換回数が上限値 K 以上になっても、平衡状態に達することができなかった場合、コストが小さくなりそうもない GP だけが現在持っている状態を捨てて、コストが小さい状態を受信する。すなわち、各温度度での変換終了時に、GP は N 回前から現在までのコストの最大値と最小値を MP に送信し、最も小さいコストを持つ GP の N 回前からの最大値よりも、最小値が大きい GP だけが、他の GP が持つ状態をコストの小さいものから順に受信する。例えば、極小解に陥ってしまったと判断された GP が 3 個あるとすれば、その他の GP からコストが小さい順に 3 個選び、それらの状態を送信する。

終了判定は、各 GP から送信されるコストの最小値が、ある回数連続した温度で同じならば、各 GP にアニメーリングを終了するようにメッセージを送信し、最小のコストを持つ状態を解とする。

4 結果

上記のような並列 SA 法を(財)新世代コンピュータ技術開発機構(ICOT)のマルチPSI上に並列論理型言語KL1を用いてインプリメントした。本実験で用いたマルチPSIは16個のプロセッサを持ち、各プロセッサは格子上に接続されている。

例題として、論理アーキテクチャ設計問題を用いた。ここで、論理アーキテクチャ設計問題とは、PASCALなどの高級言語で書かれた動作仕様からレジスタ・トランスファー・レベルの回路を生成するものである。この問題にSA法を適用するためには、動作仕様の構文解析

を行ない、その結果得られる二項演算を図6のようなスケジュール表にマッピングする必要がある[Devadas 89]。スケジュール表の各行は各タイムステップで同時に実行される演算を、各列は各ALUで実行する演算を表す。このとき、SA法における変換は、演算の移動および交換に対応し、そのコスト c は、ALU、レジスタ、バスのチップ面積、処理時間を考慮した

$$c = (alu) + (register) + (bus) + (time) \quad (2)$$

で表される。

	ALU0 *	ALU1 +, *	ALU2 -	ALU3 /
time0	bd=ad*cc	k=k+1		
time1				cc=n/k
time2		sc=sc+bd		
time3		kk=2*k		
time4			a4=ad-c	
time5				sc=ad/kk

図6: スケジュール表

最終的に得られたコストを評価するために、最小コストに対する、最終コストと最小コストの差として、コスト誤差 e

$$e = \frac{c_{fin} - c_{opt}}{c_{opt}} \quad (3)$$

を定義した。ここで、 c_{fin} は最終コスト、 c_{opt} は最小コストである。一般に、 c_{opt} は分からないが、これまでの実験で得られた解の最小値とした。

また、通常のSA法との比較のため、次のようにパラメータを設定した。

初期温度 T_0 553°
 温度減少規則 $T_{k+1} = 0.9 \cdot T_k$
 終了条件 最小コストが3回連続同じ

ただし、 T_0 は受率率 $\chi_0 = 0.9$ として、次式により求めたのものである[Laarhoven 87]。

$$T_0 = \frac{\Delta C^{(+)}}{\ln(\chi_0^{-1})} \quad (4)$$

ここで、 $\overline{\Delta C}^{(+)}$ はランダムウォークの平均コスト増加値である。

上記の初期温度、温度減少規則、終了条件は、通常の SA の最も一般的なパラメータ値の一つである。これらの設定以外に並列 SA 法のパラメータとして、

平衡状態判定のしきい値 R $0.4 \cdot r_0$
 極小解判定の履歴数 N $5 \cdot K$

とした。ここで、 r_0 は各温度での開始時の順位検定値、 K は変換回数の上限である。

K を変化させて実験を行ない、各実験で乱数の種を変えて 5 回試行した。その平均のコスト誤差と計算時間を表 1 に示す。表の変換回数 K は、並列 SA 法では各温度での変換回数の上限を、通常の SA 法では各温度で K 回の交換を行なったことを意味する。

表 1: 並列 SA 法と通常の SA 法の比較

変換回数 K	コスト誤差		計算時間 (sec)	
	並列 SA	SA	並列 SA	SA
32	0.218	0.405	474	471
64	0.141	0.302	776	920
128	0.080	0.237	1359	1936
256	0.099	0.163	2494	4159
512	0.046	0.137	5012	8154

表 1 から、通常の SA 法で平均エラーが 0.15 を下回るには、各温度での 512 回の交換回数が必要であるが、並列 SA 法では 64 回で十分であることが分かる。そして、SA 法で交換回数が 512 回の場合は 8154(sec) であり、並列 SA 法で交換回数が 64 回の場合は 776(sec) である。このことから、本並列 SA 法ではプロセッサ数が 16 個の場合の台数効果は約 10.5 (= 8154/776) であると言える。

5 まとめ

本稿では、SA 法の新しい並列処理方式について述べた。本手法は、複数プロセッサのコストの分布から平衡状態を判断することと、コストの変動範囲から極小解に陥ってしまったと判断されたプロセッサが状態の受信を行ない、その状態から再びアニーリングを開始するという二つの特徴を持つ。論理アーキテクチャ設計問題に

ついて本手法の有効性を確認したが、他の並列 SA 法およびクーリング・スケジューリング法との定量的な比較が不十分である。今後の課題としては、これらの比較に加えて、大規模並列計算機では、管理プロセッサの通信負荷がかなり大きくなることが予想されるので、平衡状態の判定、状態の送受信を局所的な範囲内のプロセッサ間で行なう処理方式にする予定である。

謝辞

本研究は、(財) 新世代コンピュータ技術開発機構 (ICOT) どの再委託契約 (発仕 S5203) に基づき行なわれたものである。また、木村宏一氏、新田克己室長を始めとする ICOT 第 7 研究室の諸氏にご助言して頂いたこと、特に、シミュレーティッド・アニーリングの関連研究をご教授頂いたことは本研究の推進に大変有用でありました。ここに記して感謝します。

参考文献

- [木村 90] 木村、瀧: 時間的に一様な並列アニーリングアルゴリズム, 信学技法, NC90-1, pp.1-8, (1990).
- [竹内 75] 竹内: 確率分布と統計解析, 日本規格協会, (1975).
- [山下 89] 山下、秋山、安西: エントロピーを用いた改良シミュレーティッドアニーリングに関する研究, 信学技法, NC89-67, pp.37-42, (1989).
- [Aarts 86] E.H.L.Aarts, F.M.J.Bont, J.H.A.Habers and P.J.M. van Laarhoven, "Parallel implementations of the statistical cooling Algorithm", *Integration*, journal 4, pp.209-238, (1986).
- [Aarts 89] E.H.L.Aarts, J.H.M.Korst, "Simulated annealing and boltzmann machines", John Wiley & Sons, (1989).
- [Casotto 87] A.Casotto and A.Sangiovanni-Vincentelli, "Placement of standard cells using simulated annealing", *Proc. IEEE Int. Conf.*

on *Computer-Aided Design*, Santa Clara,
pp.350-353, (1987).

- [Devadas 89] S.Devadas and A.R.Newton, "*Algorithms for hardware allocation in data path synthesis*", *IEEE Trans. on CAD*, vol.8, no.7, pp.768-781, (1989).
- [Kirkpatrick 83] S.Kirkpatrick, C.D.Gelatt and M.P. Vecci, "*Optimization by simulated annealing*", *Science*, vol.220, no.4598, pp.671-683, (1983).
- [Laarhoven 87] P.J.M. van Laarhoven and E.H.L.Aarts, "*Simulated Annealing : theory and applications*", Kluwer Academic Publishers, (1987).
- [Romeo 85] F.Romeo and A.Sangiovanni-Vincentelli, "*Probabilistic hill climbing algorithms : properties and Applications*", *Proc. Chapel Hill Conference on VLSI*, pp.393-417, (1985).
- [Rose 86] J.S.Rose, D.R.Blythe, W.M.Snelgrove and Z.G.Vranesic, "*Fast, high quality VLSI placement on an MIMD multiprocessor*", *Proc. IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, pp.42-45, (1986).