人 工 知 能 78－2
（1991. 9. 12）

Rule-based Annealing

舘野峰夫　荒木均　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

シミュレーティッド・アニーリング法とルールベース・システムを融合した Rule-based Annealing について述べる。本 Rule-based Annealing は組合せ最適化問題を対象として、(1) ヒューリスティックなアルゴリズムにより効率的に状態を生成する、(2) 確率的なアルゴリズムにより予期できない状態を生成できる、(3) 有効なアルゴリズムを優先的にかつ確率的に適用する機構を持つ、という特徴がある。本手法を論理回路設計に適用し、シミュレーティッド・アニーリング法に比べて、約5倍の高速化を確認した。

Rule-based Annealing

Mineo Tateno　Hitoshi Araki　Hitoshi Kato　Ryuuichi Mato

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

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

In this paper we propose a Rule-based Annealing technique which has the advantage of a Simulated Annealing algorithm and a Rule-based system. This idea is based on the technique of solving combinatorial optimization problems, and has three features; (1)Heuristic algorithms generate an effective configuration, (2)Random algorithms generate an unexpected configuration, (3)Rule-based Annealing is applied efficient algorithms probabilisticly. We used this idea for a digital circuit design, and we obtained improvement in time about 5 times faster than our Simulated Annealing program.

1　はじめに

一般に、設計問題は組合せ最適化問題であり、組合せ最適化問題を高速に解決することが大きな課題になっている。

この組合せ最適化問題の解法の１つに、シミュレーティッド・アニーリング法 [Kirkpatrick　83](以下 SA) がある。 この SA は、コストが増加した状態も確率的に採用することにより局所的最小値からの脱出が図れるという大きな長所を持っているが、膨大な処理時間がかかる。

またヒューリスティックなアルゴリズムによる組合せ最適化問題の解法も多数考えられているが、局所的最小値に陥りやすいのが大きな欠点となっている。

本稿では、SA とルールベース・システムを融合した Rule-Based Annealing(以下 RA) について述べる。この RA では、SA の確率的山登り法とルールベースの知識による最適化の性質を生かすことにより、高速でかつ局所的最小値に陥りにくい枠組の構築を試みた。

我々は、論理回路の機能設計を対象として、RA を適用し、その効果を SA による解法 [Devadas　89] と比較した。

尚、本研究は ICOT が開発したマルチ PSI 上の並列論理型言語 KL1 によりインプリメントした。

２章で SA の問題点とその問題点を解決する RA の枠組、３章で論理回路設計への適用方法と RA の処理手順、４章で RA の実験結果と評価、５章で今後の課題を述べる。

2　RA の枠組

SA は、状態を微小変形しその微小変形を何回も繰り返すことにより、大局的最小値へ近づけていく手法である。 図 2.0-1は SA の収束過程であり、A・C の谷は局所的最小状態、B の一番深い谷は大局的最小状態を示す。

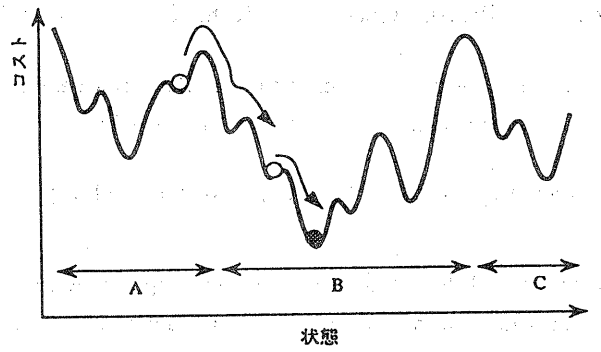


図 2.0-1: SA の収束過程

SA は、コストが増加した状態も確率的に採用する機構によって局所的最小値からの脱出が可能であるが、確率的な状態生成を行なうので無駄な状態遷移が増加し、膨大な処理時間がかかる。

そこで、「効率的な状態生成を増やし、無駄な状態遷移を削減する」ことが優れた解を高速に求める有効な解法であると考え、RA を考案した。 本 RA は、ヒューリスティックな知識によって効率的に状態を生成すると

とができ、かつ、局所的最小値からの脱出を図る機構を備えている。 以下、RA の特徴を述べる。

RA では、ある状態から新状態を生成するための手順を変換アルゴリズムとし、複数の変換アルゴリズムをルールベースとして格納する。 変換アルゴリズムとしては、ヒューリスティックな知識に基づく「ヒューリスティックアルゴリズム」と「ランダムアルゴリズム」の二種類を設ける。 ヒューリスティックアルゴリズムは、目的 (どのコストを減らすか)・規則 (アルゴリズムが発火するための条件) に基づく状態生成を行ない、発火条件に適合すると目的コストを減らすことができる。つまり、人間の持つヒューリスティックな知識による効率的な状態生成が可能になる。 また、ランダムアルゴリズムは、確率的な状態生成を行なう。発火条件がないので、制約条件を満たす限り新状態を生成し、人間が予期できない状態生成も可能である。 尚、各変換アルゴリズムは独立しているので、修正・追加が容易であるという性質をもつ。

また、新しい状態を生成する場合に、複数の変換アルゴリズムのうちどの変換アルゴリズムを適用すれば、最終的に優れた解を求めることができるか判定できない。 したがって、RA では、効率的に変換アルゴリズムを適用するために、各変換アルゴリズムの有効性を判定する基準を設け、有効性の高い変換アルゴリズムを優先的に適用する機構を導入した。 この機構では、有効性の高い変換アルゴリズムの適用比率を高くし、有効性の低い変換アルゴリズムの適用比率を低くして、各変換アルゴリズムをその適用比率に基づいて確率的に適用している。本稿では、受理率の高い変換アルゴリズムを有効性の高い変換アルゴリズムであると考えている。

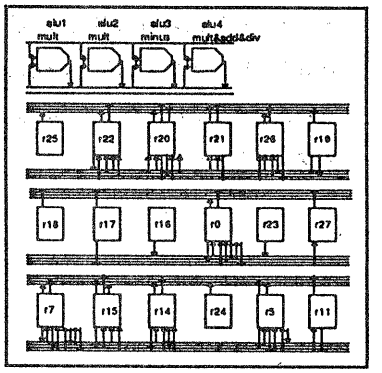
さらに、RA では、SA のクーリング・スケジュール機構を適用して、変換アルゴリズムにより生成された状態を評価して採用するか否かを判定し、無駄な状態遷移の削減と局所的最小状態からの脱出を図っている。

3 処理詳細

本 RA を論理回路設計に適用した。

	ALU1 *	ALU2 *	ALU3 -	ALU4 + * /
time0				t4=v[3]*v[3
time1	t5=t4*3	t2=h[3]*2		
time2	t1=b*t[3]		t6=t5-2	t3=1+t2
time3	t7=g*p	c[3]=t3*t6	w[3]=w[0]-t	
time4	t10=g*c[3]	t14=2*a[1]		t8=t7 /w[3]
time5	t7=g*p		t15=t14-a[2]	t11=t10-v[3

(a)



(b)

図 3.0-2: スケジュール表と設計回路図

図 3.0-2(a) に示すように、動作仕様 (PASCAL 言語) を構文解析してスケジュール表に変換する。スケジュール表の time0-5 は時間ステップ、ALU1-4 は ALU 番号と実行可能な演算、$t7 = g * p$ は演算器 ALU1 で time3 において実行されることを示している。 尚、このスケジュール表は一部であり、全体の時間ステップ数は 38 である。

スケジュール表を 1 つの状態とみなして、その状態に RA を適用してスケジュール表内の式を移動・交換す

ることにより、状態を遷移させて準最適な状態を求める。 図 3.0-2(b) はスケジュール表に対応した設計回路図の一部であり、4 個の ALU(alu1 ～ alu4)、レジスタ (r0 ～ r27) と ALU バス間のリンク情報を示している。

図 3.0-3に、 RA の処理手順を示す。

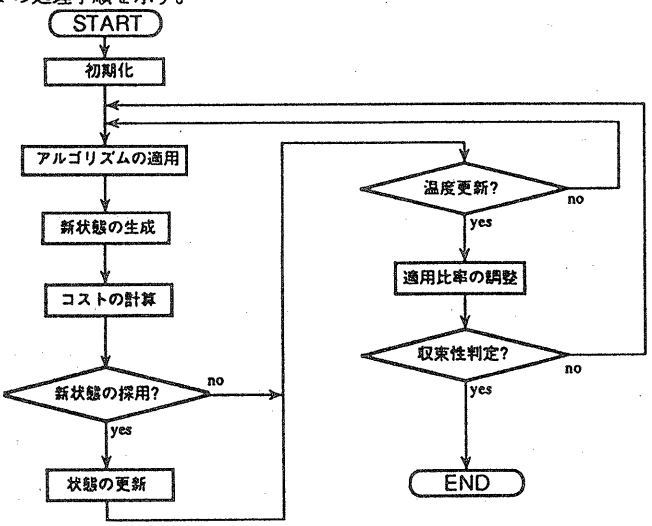


図 3.0-3: Rule-based Annealing の処理手順

- 動作仕様のサイズに基づいて初期スケジュール表を生成し、初期温度、温度毎の適用回数を設定する。

- ランダムアルゴリズムとヒューリスティックアルゴリズムを適用比率に基づいて確率的に適用し、新状態を生成して、コスト計算を行なう。 変換アルゴリズム適用前の状態を S_{old}、そのコストを C_{old}、変換アルゴリズム適用後の新状態を S_{new}、そのコストを C_{new}、 $\Delta C = C_{new} - C_{old}$ とする。式 (1) に示されるように、コストが減少した場合は確率 1 で新状態に遷移し、コストが増加した場合は確率 $\exp \frac{-\Delta C}{Temp}$ で新状態に遷移する (コストが増加した場合は、温度が高い程、コスト増分が小さい程、新状態に遷移する確率が高くなる)。

$$P = \begin{cases} 1 & \text{if } \Delta C < 0 \\ \exp \frac{-\Delta C}{Temp} & \text{otherwise} \end{cases} \tag{1}$$

$$\Delta C = C_{new} - C_{old} \tag{2}$$

- 各温度における変換アルゴリズムの適用回数は一定とし、変換アルゴリズムの適用回数が設定値を越えたら温度 $Temp$ を以下の式により下げる。ここで、 K は温度更新回数である。

$$Temp(K) = 0.9^K \times Temp(0) \tag{3}$$

- 温度を更新したら、各変換アルゴリズムの有効性(受理率が高い変換アルゴリズムを優先的に適用する)に基づいて、次の温度における各変換アルゴリズムの適用比率を決める。
- 上記の処理を繰り返しながら温度を徐々に下げ、各温度における最終コストが3回続けて同じになったら、準最適解が求まったとみなして処理を終了する。

以下、変換アルゴリズム、コスト計算、アルゴリズム適用比率調整について詳細に述べる。

3.1 変換アルゴリズム

変換アルゴリズムは、適用比率に基づいて確率的に適用される。変換アルゴリズムには、ランダムアルゴリズムとヒューリスティックアルゴリズムの2種類がある。現在、10個の変換アルゴリズム(ランダムアルゴリズム3個、ヒューリスティックアルゴリズム7個)をインプリメントした。以下に、主な変換アルゴリズムの例を示す。

3.1.1 ランダムアルゴリズム

確率的に新状態を生成するためのアルゴリズムである。発火条件はないので、順序制約を満たす限り新状態を生成する。スケジュール表内の1式の移動・交換を行なう。

(1) high — スケジュール表内の式をランダムに1つ選び、その式をランダムな場所に移動させる

(2) low2 — スケジュール表内の式をランダムに1つ選び、その式の左右いずれかに移動させる

(3) low4 — スケジュール表内の式をランダムに1つ選び、その式の上下左右いずれかに移動させる

3.1.2 ヒューリスティックアルゴリズム

ヒューリスティックな知識に基づくアルゴリズムであり、ALUコスト、BUSコスト、REGISTERコスト、LINKコスト、時間コストの各コストを減少させる。発火条件に適合しない場合、および順序制約を満たさない場合は、新状態を生成できない。スケジュール表内の複数の式を同時に移動・交換することができる。

(1) alu2 — ALU・BUSコストを減らす

IF (列にある式の数 ≤ 3) *THEN* (式を他の列へ移動させてその列をなくす)

(2) reg — REGISTERコストを減らす

IF (レジスタ密度が最大である行) *THEN* (その行にある式を他の行へ移動させる)

(3) link — LINKコストを減らす

IF (各ALUの減・除算の左引数リストと右引数リストを作り、両リストのうち片リストだけに含まれる引数がある) *THEN* (それらの引数に合わせて、加・乗算の引数を交換する)

(4) time1 — 時間コストを減らす

IF (行にある式の数= 1) *THEN* (その式を他の行に移動させてその行をなくす)

3.1.3 順序制約

動作仕様のデータ依存関係に基づいて式の移動可能性を調べる。順序制約を満たさない場合は、状態を生成できない。

3.2 コスト計算

論理回路設計の場合、面積コスト(ALU・BUS・REGISTER・LINK)、時間コスト(TIME)の5種類のコストを元にコスト計算を行なう。

$$C = (ALU) + (BUS) + (REGISTER) + (LINK) + (TIME) \tag{4}$$

3.3 アルゴリズム適用比率調整

各変換アルゴリズムの適用比率を温度毎に動的に変える機構である。各温度において有効な変換アルゴリズムを優先的に適用し、逆にあまり有効ではない変換アルゴリズムの適用回数を減らすことができる。

ここでは、受理率の高い変換アルゴリズムを有効な変換アルゴリズムと考えている。同一温度において、各変換アルゴリズムの適用回数と採用回数をカウントして各変換アルゴリズムの受理率(採用回数 / 適用回数)を求め、その値の大小に基づいて次の温度における各変換アルゴリズムの適用比率を決定する。

3.3.1 各変換アルゴリズムの適用比率調整値

温度毎の各変換アルゴリズムの受理率(採用回数 / 適用回数)に基づいて、適用比率調整値 $F_i(K)$ を以下のように定める。

$$F_i(K) = \begin{cases} \frac{A_i(K)}{S_i(K)} & \text{if } S_i(K) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

ここで、i は変換ID、K は温度更新回数、$A_i(K)$ は温度毎の各変換アルゴリズムの採用回数、$S_i(K)$ は温度毎の各変換アルゴリズムの適用回数である。$(0.0 \leq F_i(K) \leq 1.0)$

3.3.2 各変換アルゴリズムの適用比率

各変換アルゴリズムの適用比率調整値 $F_i(K)$ から、各変換アルゴリズムの適用比率の増減率 R_i を求める。

$$R_i(K) = W \times (F_i(K) - \frac{\Sigma F_i(K)}{N}) \tag{6}$$

ここで、$F_i(K)$ は各変換アルゴリズムの適用比率調整値、N は変換アルゴリズム数、$\frac{\Sigma F_i(K)}{N}$ は適用比率調整値の平均である。W は増減率幅であり、$(0.0 \leq W \leq 1.0)$ の範囲で設定する。W を大きくすると各変換アルゴリズムの適用比率の増減率は大きくなる。$(-1.0 \leq R_i(K) \leq 1.0)$

各変換アルゴリズムの初期適用比率は同じ値とするので、次の温度における各変換アルゴリズムの適用比率 V_i は以下のようになる。

$$V_i(0) = \frac{1}{N} \tag{7}$$

$$V_i(K+1) = \frac{(1+R_i(K)) \times V_i(K)}{\Sigma((1+R_i(K)) \times V_i(K))} \tag{8}$$

ここで、$R_i(K)$ は各変換アルゴリズムの増減率、N は変換アルゴリズム数である。$(0.0 \leq V_i(K) \leq 1.0)$

4 実験結果

式数 32 の動作仕様 (elliptic) に対して、SA と $RA(W=0)$・$RA(W=0.1)$・$RA(W=0.2)$ の収束コストと処理時間を温度毎の適用回数を変えて測定した。表 4.0–1および図 4.0–4は、初期状態を変えて 15 回の試行を行なった場合の平均値である。 ここで、$RA(W=0)$ は各変換アルゴリズムの適用比率を等しくした RA、$RA(W=0.1)$・$RA(W=0.2)$ は適用比率調整機構を取り入れ増減率幅 W を 0.1・0.2 とした RA である。また、SA はランダムアルゴリズム high だけを適用している。

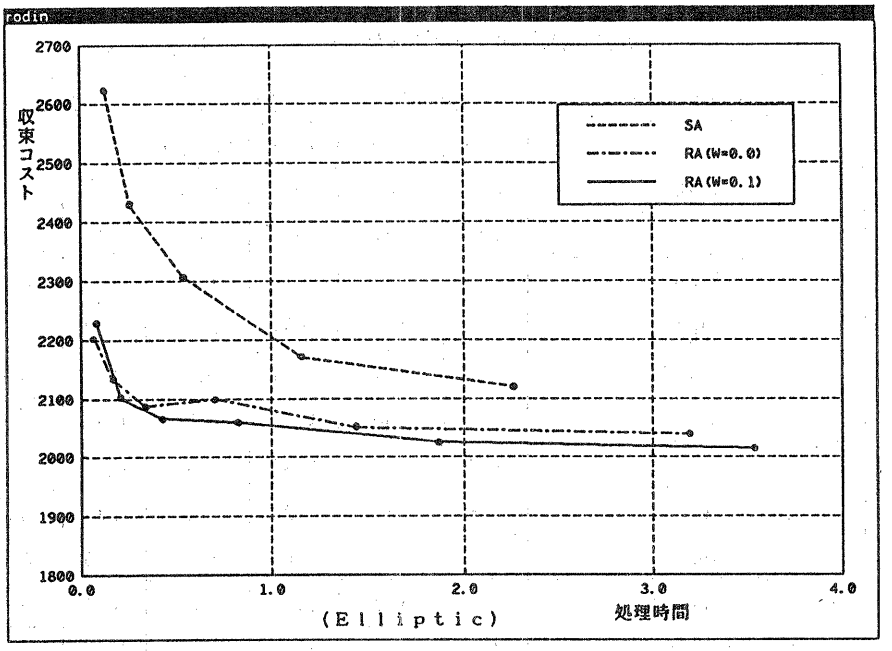


図 4.0–4: SA と RA の平均収束コストと平均処理時間

図 4.0–4から、1.0 時間処理を行なった場合の収束コスト C は、SA は 2205、$RA(W=0.0)$ は 2080、$RA(W=0.1)$ は 2053、$RA(W=0.2)$ は 2041 になる。 最適コストを C_{opt}(=1865) としたエラー率 $e = \frac{C-C_{opt}}{C_{opt}}$ は、SA は 0.182、$RA(W=0.0)$ は 0.115、$RA(W=0.1)$ は 0.101、$RA(W=0.2)$ は 0.094 になる。適用比率調整機構を導入することにより、$RA(W=0)$ に比べて $RA(W=0.1)$ は 12%、$RA(W=0.2)$ は 18% 改善されている。

また、コスト2150($e = 0.153$)に収束するまでの処理時間は、SA は1.56[H]、$RA(W = 0.1)$ は0.16[H]、$RA(W = 0.2)$ は0.32[H]である。

表 4.0–1: SA と RA の平均収束コストと平均処理時間

温度毎の適用回数	収束コスト				処理時間 [H]			
	SA	RA			SA	RA		
		W=0	W=0.1	W=0.2		W=0	W=0.1	W=0.2
32	2620	2204	2230	2256	0.13	0.07	0.08	0.10
64	2429	2136	2100	2169	0.26	0.16	0.21	0.23
128	2307	2086	2066	2117	0.54	0.34	0.43	0.49
256	2169	2099	2057	2037	1.16	0.71	0.86	1.03
512	2119	2050	2025	2001	2.26	1.44	1.88	2.06
1024		2038	2014	1988		3.20	3.54	4.13

表 4.0–2: 適用比率調整機構による各変換アルゴリズムの採用回数の変化

アルゴリズム名	採用回数		適用回数		受理率	
	W=0	W=0.1	W=0	W=0.1	W=0	W=0.1
high	608	748	1245	1656	0.49	0.45
low2	709	942	1236	1943	0.57	0.48
low4	702	1002	1226	2049	0.57	0.49
alu1	288	501	1237	1379	0.23	0.36
alu2	187	221	1221	826	0.15	0.27
alu3	355	560	1273	1207	0.28	0.46
reg	136	113	1260	636	0.11	0.18
link	545	772	1288	1443	0.42	0.53
time1	117	117	1265	613	0.09	0.19
time2	145	193	1293	792	0.11	0.24
ランダムアルゴリズム	2019	2692	3707	5648	0.54	0.48
ヒューリスティックアルゴリズム	1773	2477	8837	6896	0.20	0.36
アルゴリズム合計	3792	5169	12544	12544	0.30	0.41

収束コスト RA(W=0):2060, RA(W=0.1):1925

表 4.0–2 は、初期状態を同じにして、各変換アルゴリズムの適用比率が等しい $RA(W = 0)$ と適用比率調整機構を取り入れた $RA(W = 0.1)$ について、総適用回数(12544 回)を等しくした場合の各変換アルゴリズム

の採用回数・適用回数・受理率(採用回数 / 適用回数)の変化を示している。 ここで、ランダムアルゴリズムは high・low2・low4 の 3 個、ヒューリスティックアルゴリズムは alu1・alu2・alu3・reg・link・time1・time2 の 7 個である。

実験では、受理率(採用回数 / 適用回数)が高い変換アルゴリズムを優先的に適用しているので、各変換アルゴリズムの採用回数が多くなり、適用比率調整機構がない場合に比べて、受理率が 0.30 から 0.41 へ改善されている。 特にヒューリスティックアルゴリズムの受理率が 0.20 から 0.36 へ大きく増加している。そして、収束コストも大きく改善されていることがわかる。

また、$RA(W = 0.1)$ における変換アルゴリズム毎の初期適用比率・最終適用比率の例を、表 4.0-3 に示す。 実験では、各変換アルゴリズムの初期適用比率を等しくした。 alu2・reg・time1・time2 は高中温で有効なアルゴリズムであるが、低温ではあまり有効ではない。 逆に low2・low4・link は、低温で有効な変換アルゴリズムである。

表 4.0-3: 各変換アルゴリズムの適用比率の変化

アルゴリズム名	high	low2	low4	alu1	alu2	alu3	reg	link	time1	time2
初期比率 (553 度)	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
最終比率 (3.5 度)	0.13	0.18	0.20	0.07	0.03	0.12	0.02	0.20	0.02	0.03

一般に、ヒューリスティックなアルゴリズムだけのルールベースではヒューリスティックな知識による効果しか期待できず、局所的最小値に容易に陥ってしまうことが多い。本 RA はランダムアルゴリズムも導入したルールベースなので、ヒューリスティックな知識による記述の限界をランダムアルゴリズムの確率的な試行により補うことができると考える。

5 おわりに

本稿では、シミュレーティッド・アニーリングとルールベース・システムを融合した Rule-based Annealing の枠組を述べ、その有効性の確認を行なった。

Rule-based Annealing の特徴は、(1) 有効なアルゴリズムを優先的にかつ確率的に適用する機構を持つ、(2) 確率的なアルゴリズムにより予期できない状態を生成できる、(3) ヒューリスティックなアルゴリズムにより効率的に状態を生成する、(4)SA のクーリング・スケジュールを適用している、の 4 点である。

上の (1) に関して、現枠組では有効なアルゴリズムの適用比率を増やすという基準を採用しているが、今後は複数のアルゴリズムをどんな順序で適用するのが有効であるかを判定する基準を導入することを検討している。

また (4) に関して、現枠組では SA のクーリング・スケジュールをそのまま適用しているが、もっと効率的な制御機構を検討するつもりである。

本稿では、知識処理による SA の高速化について述べたが、並列処理による SA の高速化 [荒木 91] についての研究も行なっている。

謝辞

本研究は、(財)新世代コンピュータ技術開発機構(ICOT)の委託研究として行なったものである。本研究の機会を与えて下さったICOT第7研究室の新田室長、情報通信東京研究所基礎研究部の山崎部長に深く感謝いたします。

【参考文献】

[Kirkpatrick 83] S. Kirkpatrick, C.D. Gelatt,Jr., and M.P. Vecchi *" Optimization by Simulated Annealing"*, *Science Vo. 220 No. 4598*, 1983, pp. 671–680.

[Devadas 89] S.Devadas and A.R.Newton *"Algorithms for Hardware Allocation in Data Path Synthesis"*, *IEEE Trans. on CAD*, Vol.8, No.7, 1989, pp.768–781.

[荒木 91] 荒木、舘野、加藤、間藤: 疎結合並列計算機上でのシミュレーティッド・アニーリング, 1991年並列／分散／協調処理に関する「大沼」サマー・ワークショップ