人 工 知 能 101－6
（1995. 9. 12）

遺伝的アルゴリズムによる制約充足問題の解法

松本 美幸　内野寛治　狩野 均　西原 清一
筑波大学　電子・情報工学系

近年、山登り法やニューラルネットワークによって、大規模な制約充足問題を実用的な時間内に解くための研究が盛んに行われている。しかしこれらの研究は、局所最適解からの脱出方法について検討したものであり、積極的に大域的探索を行う方法はあまり見あたらない。そこで本論文では、大域的探索戦略を有する遺伝的アルゴリズム (GA) と制約違反最少化戦略による山登り法 (MCHC) をハイブリッド化した探索法を提案する。本手法は、GA の集団中で制約違反数が最少の個体を初期値として MCHC により局所的に探索するものである。本手法が制約密度の小さい CSP に対して従来手法より探索成功率が高いことを詳細な実験で示す。

Genetic Algorithms for Constraint Satisfaction Problems

Miyuki Matsumoto　Kanji Uchino　Hitoshi Kanoh　Seiichi Nishihara

Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba, Ibaraki 305, Japan

e-mail:miyuki@algor.is.tsukuba.ac.jp

Several approximation algorithms using hill-climbing techniques and neural networks have recently been proposed to solve large constraint satisfaction problems (CSPs) in a practical time. In these proposals, many methods of escaping from local optima are discussed;however, there are very few methods actively perform global search. Among this paper we propose a hybrid search method that combines the genetic algorithm with the min-conflicts hill-climbing (MCHC). In our method, the individual that has the fewest conflicts in the population is used as the initial value of MCHC to search locally. The detailed experimental simulation is also performed to prove that the proposed method generally gives better efficiency than the naive random restarting MCHC when CSPs are sparsely-connected.

1 はじめに

制約充足問題 (CSP : Constraint Satisfaction Problem) は、与えられた制約を満たす解を発見する組合せ探索問題である [1][2]。CSP が解を持つかどうかという決定問題は NP 完全であり、効率的な一般解法は存在しない。このため、反復改良法による近似解法によって、なるべく高い確率で実用的な時間内に解を求めようとする研究が盛んに行われている。これは、バックトラックによる厳密解法がアルゴリズムの完全性 (停止性) を保証しているのに対して、これを犠牲にして高速化や高度の並列化を実現しようというものである。

近年、制約違反数を最少化する方向に山登りする方法 (MCHC)[3] や相互結合型のニューラルネットに補助ネットを付加した方法 (GDS)[4] が比較的良い結果が得られたと報告されている。しかし、これらの方法には大域的探索戦略が含まれていないため、探索途中で局所最適解に陥ってしまう危険性があり、適用範囲が限られていた [5]。このため、局所最適解からの脱出方法も提案されているが (GSAT[6] [7], breakout 法 [8],GENET[9], EFLOP[10])、積極的に大域的探索を行う方法はあまり見あたらない。

一方、遺伝的アルゴリズム (GA : Genetic Algorithms) は、生物の進化過程をモデル化した確率アルゴリズムであり、大域的探索戦略を有するため、多くの分野において最適化問題の解法として利用されている [11][12]。しかし、GA を CSP に適用した例はあまり見あたらない。この理由は、最適化問題が、準最適解を見つければ良いのに対して、CSP ではすべての制約を満たす完全な解を見つける必要があり、局所探索戦略をもたない GA はそのままでは適用できないためと考えられる。

そこで本論文では、GA と MCHC をハイブリッド化した探索法を提案する [13]。本手法はまず、GA で大域的に探索し、次に集団中で制約違反数が最少の個体 (エリート個体) を初期値として MCHC により局所的に探索する。その際、GA では多様性の維持を最優先させるために、エリート個体とのハミング距離が近い個体は、その適応度を減少させる。また、局所最適解に陥ったエリート個体は死滅させ、GA による別の初期値から再スタートするものである。

以下では、まず CSP を変数の数と制約の数の関係から 4 種類に分類する。次に、GA による CSP のコード化の方法と提案するハイブリッド化の方法について述べる。最後に、Haralick の定義による部分解列挙型 CSP を用いたシミュレーションにより、GA、ランダムな初期値から MCHC を繰り返す方法、ならびに本手法の探索速度と探索成功率を比較検討する。

2 CSP の分類と本手法の基本戦略

2.1 CSP とその分類方法

CSP は、論理式の充足可能性問題やグラフ色塗り問題を例に理論的に検討されてきた。ここでは、より一般的な Haralick の定義を用いる [1]。CSP は、4 つ組 (U, L, T, R) で定義される。$U = \{1, \cdots, n\}$ は変数の集合で、各要素は対象とする問題の構成要素にあたる。L は値の集合で、各要素は変数に割り当てられる候補を表す。$T = \{t_1, \cdots, t_m\}$ は制約関係 t_j の集合で、$t_j = (p, q)$ は、p 番目の変数と q 番目の変数の間に制約が存在することを示す。$R = \{R_1, \cdots, R_m\}$ は制約関係 t_j にある変数が満たす部分解 R_j の集合である。CSP の例を図 1 に示す。ここでは、制約は全て 2 項制約とする。以上述べた部分解列挙型の CSP に対しては、ヒューリスティクスを用いた高速解法が提案されている [1]。本

U = {1,.....,4},
L = {a,b,c,d,e},
T = {t1,.....,t4},
t1 = (1,2), t2 = (1,3),
t3 = (3,4), t4 = (1,4),
R = {R1,.....,R4},
R1 = {(a,b),(e,c)},
R2 = {(a,c),(a,e)},
R3 = {(c,e),(a,c)},
R4 = {(a,d),(a,e),(b,c)}
変数 : (1,2,3,4)
解 : (a,b,c,e)

図 1: CSP の例

論文では一般的な CSP の解法を対象としているので、このような問題固有のヒューリスティクスは用いずに解候補の制約違反数のみで議論することにする。

CSP は NP 完全な組合せ探索問題であり、その意味で効率の良い汎用の探索アルゴリズムは存在しない。そこで本論文では、変数の個数 n と制約の個数 m の関係に着目して CSP を次の 4 種類に分類し、種類別に有効な解法を考察するという立場をとる。

$$\begin{array}{llll} Type1: & m = & n-1 & \\ Type2: & n-1 & < m \ll & n(n-1)/2 \\ Type3: & n-1 & \ll m < & n(n-1)/2 \\ Type4: & m = & n(n-1)/2 & \end{array}$$

図 2 は、これらの代表例を制約グラフで表したものである。グラフの頂点は変数、辺は制約関係に対応している。*Type*1 は、制約グラフが木構造をしている場合で、このとき、変数の数に対する制約密度 $d = m/n$ は最小となる。これに対して *Type*4 は完全グラフの場合で、d は最大となる。*Type*2, 3 は、これらの中間の場合である。一般に、制約密度の小さい問題は、局所最適解に陥り易く難問とされているが、これは Type1 に相当する。探索アルゴリズムのベンチマークとしてよく用いられる n クィーン問題は、*Type*4 に属する。また、CSP の近似解法としてしばしば引用される文献 [5] のグラフ色塗り問題で、粗結合グラフはほぼ *Type*1、密結合グラフは *Type*3 に相当する。この分類に基づいて CSP の解法を比較検討することにより、各手法の適用範囲を明確にすることができると思われる。

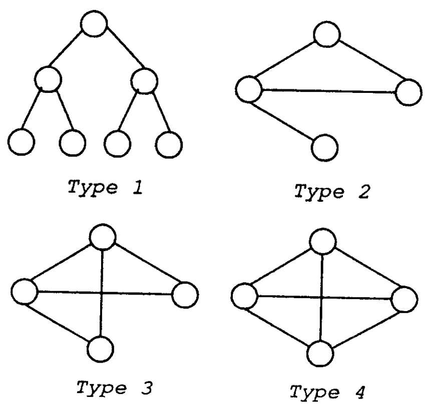


図 2: CSP の分類

2.2 CSP の近似解法

制約違反を最少化する方向に山登りする方法 (以下,MCHC:mini conflict hillcl imbing と称す)[5] を用いた大規模 CSP の解法が多数提案されている [6][8][10]。これは値の選択規則として、次の経験則用いている。

[制約違反最少化戦略]
制約違反している変数に対して制約違反数が最少になる値を代入する
(値が同じときはランダム)。

MCHC は、GDS とほぼ同等の性能が得られたと報告されている [5] が、GDS と同様に、制約密度が低い問題に対しては、探索途中で局所最適解に陥ってしまう危険性が高いとされている。この欠点を補うため、制約の重みをダイナミックに変化させて局所最適解から抜け出す方法 [8]、他の初期値から探索をやり直す方法 (GSAT)[6] [7]、また、GDS の改良である GENET[9] などが提案されている。

本論文では、GSAT と同じ方法を GA と比較検討する。すなわち、1 つの初期値について MCHC により一定回数探索を進めても制約違反数が減少しない場合は、局所最適解に陥ったとみなして探索を中止し、別の初期値から探索をやり直す方法である (以下、IHC:iterated hillclimbing と称す)。これは、状態空間におけるコスト関数の形状に関する次の仮説に基づいている。

[Restart **仮説**]
制約密度の小さい CSP では、最適解は局所最適解と離れたところにある。

2.3 遺伝的アルゴリズム

GA は、探索空間上の状態を個体と見なし、その個体の集団に対して、選択, 交叉, 突然変異という遺伝的操作をおこなうことにより、制約違反のない状態を発見しようとするものである。

GA は、次の BB 仮説に基づいて大域的探索を行うもので、山登り法やニューラルネットワークとは異なった特性が得られる可能性があると考えられる。

[Building Block (BB) 仮説]
制約違反の少ない個体は、有効な部分解を多く含んでいる。これらは選択と交叉により破壊されるよりも積み上げられる確率の方が高い。

3 提案する手法

3.1 基本戦略

本論文では GA と MCHC をハイブリッド化する方法を提案する。以下では本手法を GAHC と略記する。その基本戦略は、次の３項目である。

(1) まず、GA で大域的に探索し、次に集団中で制約違反数が最少の個体 (エリート個体) を初期値として MCHC により局所的に探索する。(エリート個体が複数のときはランダムに選択する)

(2) GA では、集団における多様性の維持を最優先させことにより、大域的探索能力を強調する。このため、エリート個体とのハミング距離が近い個体は、その適応度を減少させる。

(3) MCHC で局所最適解に陥ったエリート個体は死滅させ、GA による別の初期値から再スタートする。すなわち、再出発仮説を適用する。

3.2 CSP のコード化と適応度の計算

本モデルでは、CSP を GA 上で表 1 のように表現する。 GA の k 番目の個体の染色体を G_k、G_k

表 1: CSP と GA の対応

CSP	GA	例
変数	遺伝子座 i	$1,2,3,4$
値	遺伝子 $G_k(i)$	a,b,c
解候補	染色体 G_k	(a,b,a,c)

の i 番目の遺伝子座にある遺伝子を $G_k(i)$ とかく。また、表 1 の第 3 列は、図 1 の CSP の例である。このようなコード化に対して、個体 k の適応度 F_k を以下のように計算する。

$$F_k = 1 - \sum_{j=1}^{m} CONF_k(j)/m \qquad (1)$$

ただし、$t_j = (p,q), R_j$ に対して,

$$CONF_k(j) = \begin{cases} 1 & ((G_k(p), G_k(q)) \in R_j) \\ 0 & (otherwise) \end{cases}$$

m : 制約数

3.3 選択確率の計算

2.3 節で述べたとおり GA においては多様性の維持を最優先させるため、エリート個体との類似性が高い個体は選択確率を通常より減少させる。エリート個体の染色体を G_e とすると、G_e と G_k のハミング距離 H_k を次のように定義する。

$$H_k = n - \sum_{i=1}^{n} \delta(G_k(i), G_e(i)) \qquad (2)$$

ただし、

$$\delta(x,y) = \begin{cases} 1 & (x = y) \\ 0 & (otherwise) \end{cases}$$

n : 遺伝子座の数 = 変数の数

このような H_k を用いて、選択確率 PS_k は次の (3) 式で計算する。

$$PS_k = F_k \times H_k/n \qquad (3)$$

3.4 本手法のアルゴリズム

本手法のアルゴリズムを図3 に示す。図 3 で、N_p は集団サイズ (個体数),N_g は世代数の上限とする。

まず、N_p 個の個体 (染色体 G_k, $k = 1,$ をランダムに発生させる。次に以下の (a),(b) を解が発見されるまで (または N_g 回) 繰り返す。

(a) 局所的探索: 全個体の適応度 F_k を (1) 式で計算し、F_k が最大の個体をエリート個体とする。エリート個体の適応度 F_e がしきい値 F_t をこえた場合、エリート個体の染色体を初期値として MCHC で局所的探索を実施する。

(b) 遺伝的操作: まず、(3) 式の選択確率によるルーレット戦略とエリート保存戦略を併用し、集団から N_d 個の個体を選択する。この個体から、一様交叉により N_p 個の個体を次世代として生成する。次に、ランダムに選んだ個体の遺伝子座に対して、ランダムな遺伝子を予め設定された確率 (突然変異率) で割り当てる。

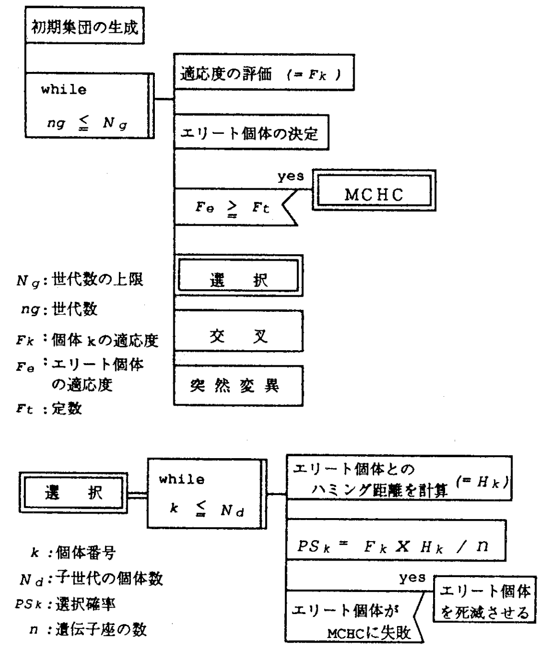


図 3: 本手法のアルゴリズム

4　性能評価

4.1　実験方法

IHC,GA,GAHC の性能を比較検討するため、$Type1$〜4 の CSP をランダムに発生させ、解探索を試みた。以下では、値の数 (対立遺伝子数) は 4、変数の数は 50 とした。したがって、探索空間の大きさは約 10 の 30 乗である。表 2 は、制約数と制約密度をタイプ別に示したものである。また手法の評価は、次の 2 項目についておこなった。CSP の近似解法においては、一般にこれらはトレードオフの関係にあり、どちらを優先させるかはアプリケーションによって異なる。

表 2: 制約数と制約密度

$Type$	1	2	3	4
m	$n-1$	$9(n-1)$	$17(n-1)$	$n(n-1)/2$
d	0.98	8.82	16.7	24.5

(a) 探索成功率 (%): 探索した CSP のうち解を発見できたものの割合。

(b) 平均解法時間 (min): 解を発見できた CSP についての平均探索時間。

なお、計算機は HP715/33,LUNA88K_2 を使用し、プログラムはすべて C 言語で記述した。

4.2　パラメータの設定

HC と GA では探索方法が全く異なるので、性能を比較するときには注意が必要である。特に、解が得られない場合に探索をどこで打ち切るかによって、成功率と探索時間の統計値が変化することがある。本論文では、IHC における反復回数の上限 N_i と GA における世代数の上限 N_g を、探索に要する計算コストの理論的な最大値が各手法でほぼ等しくなるように設定した。これらのコストはそれぞれ以下のように求めることができる。

$$cost(IHC) = C_{HC} \times N_h \times N_i \quad (4)$$

$$cost(GA) = C_{GA} \times N_p \times N_g \quad (5)$$

$$cost(GAHC) = C_{GA} \times N_p \times N_g + C_{HC} \times N_h \times N_g \quad (6)$$

ただし、$C_{HC} = C1 \times m \times n$,
$C_{GA} = C2 \times m + C3 \times n$
$C1, C2, C3$: 定数

C_{HC} は、山登りの 1 ステップに要するコストで、変数ごとの制約違反数を計算するコストにほぼ比例する。C_{GA} は、GA の 1 世代 1 個体当たりの計算コストである。これは次の 2 つの項の和となる。第 1 項は、個体の制約違反数を計算するコストで (1) 式のように m に比例する。第 2 項は、選択確率の計算コストと遺伝的操作の計算コストで n に比例する。

表 3 に C_{HC}/C_{GA} の実測値を示す。また、N_h は実際に山登りの行われた回数の平均値である。われわれが最も興味のあるのは制約密度の低い $Type1$ の CSP に対して、GAHC の性能を IHC および GA と比較することである。

表 3: C_{HC} と C_{GA} の実測値

$Type$	1	2	3	4
C_{HC}/C_{GA}	2.98	9.37	14.5	15.8

したがって、以下の実験では、表 3 の $Type1$ の値を用いて、計算コストの理論的な最大値を推定し、次の (7) 式の条件を満たす範囲で各パラメータを設定するようにした。

$$cost(GAHC) \leq cost(HC), cost(GA) \qquad (7)$$

(4) ～ (7) 式は解が得られない場合に探索をどこで打ち切るかを判断するとき重要となる。

なお、GA および GAHC では全実験を通して、生存率 $N_d/N_p = 0.5$、1 遺伝子座当たりの突然変異率 =1% とした。

4.3 通常の GA による CSP の解法

GA では、(1) 式の適応度 F_k の 2 乗を選択確率としたルーレット戦略、交叉方法は一様交叉を用いた。また、世代数の上限 N_g=200 とした。比較のためエリート保存戦略を併用した GA(以下、GAE と称す。) でも実験を行った。表 4 ，表 5 にこの結果を示す。各値は $Type1$～4 の平均値である。表 4 では、集団サイズが大きくなるとともに成功率が増加している。

表 4: GA の実験結果

N_p	500	1000	2000	3000
Succ	71	72	77	81
Time	4.8	9.0	8.9	12

Succ: 探索成功率 (%)
Time: 平均探索時間 (min)

一方、表 5 では、成功率は 71 ～ 74% の範囲と、ほぼ一定であるにもかかわらず、探索時間は集団サイズの変化量に比例して増加している。一般的に最適化問題では、エリート保存戦略を併用することにより、より最適解に近い準最適解が得られるといわれている。しかし、本実験のような CSP では解が発見できた否かを問題にしているためこの効果が現れていないものと思われる。

表 4 ，表 5 を通じて、エリート保存戦略を併用しない GA で $N_p = 2000$ のときが比較的良い結果が得られた。

表 5: GAE の実験結果

N_p	500	1000	2000	3000
Succ	71	74	74	73
Time	3.6	7.1	11	20

4.4 GAHC による CSP の解法

表 6 は表 4 , 表 5 と同様の実験を GAHC で行った結果である。世代数の上限は N_g=100,F_t=0.5 とした。表 6 では, N_p=1000 のとき最も良い結果が得られた。

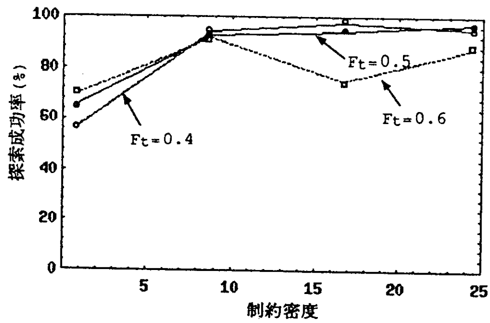
表 6: GAHC の実験結果

N_p	500	1000	2000	3000
Succ	76	87	88	86
Time	2.9	3.5	7.2	11

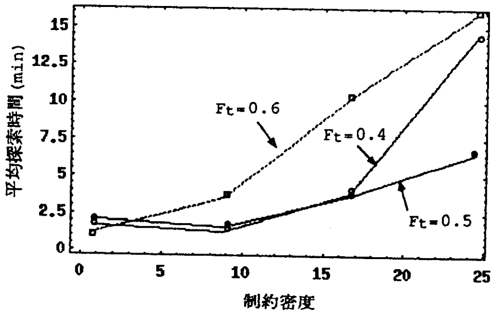
図 4 は、$N_p = 1000, N_g = 100, F_t = 0.4, 0.5, 0.6$ としたときの実験結果である。横軸は制約密度 $d = n/m$ で、表 2 に示した $Type1$～4 の値である。図 4 中の各データは、40 個の CSP の平均値である。また、F_t は、大域的探索から局所的探索へ移行するときの適応度のしきい値で、GAHC においては最も重要なパラメータである。GAHC の動作は、$F_t = 0$ では IHC と一致し、$F_t = 1$ では GA と一致する。図 4 (a) では、$F_t = 0.6$ のケースは $d = 17$ 付近で成功率が劣化している。また、図 4 (b) では、$F_t = 0.4$ のケースは $d = 25$ 付近で探索時間が大きく増加している。従って、$F_t = 0.5$ のときが最も良い結果が得られたといえる。

4.5 GAHC と他の手法の比較

実験結果を図 5 に示す。各データは 400 個の CSP の平均値である。IHC では $N_i = 500$, とした。GA と GAHC のパラメータは上記の最も良い値とした。本実験中に測定した N_h は、平均で IHC では 200、GAHC では 100 であった。これらのパラメータは、(7) 式の条件を満たしている。図 5 から以下のこと

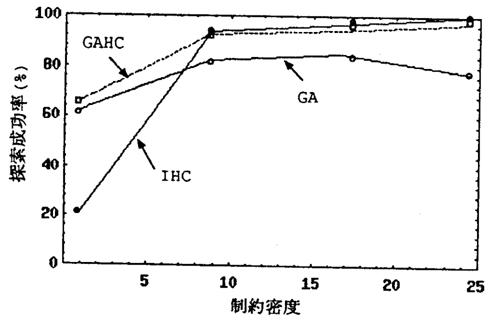


(a) 探索したCSPのうち解を発見できたものの割合

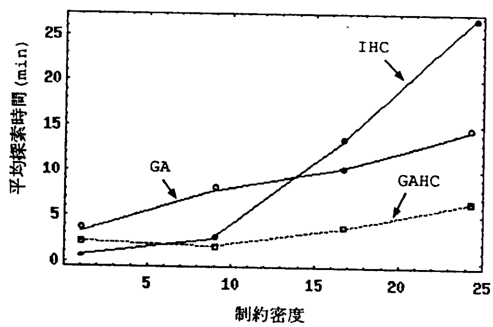


(b) 解を発見できたCSPについての平均探索時間

図 4: F_t に関する実験結果



(a) 探索したCSPのうち解を発見できたものの割合



(b) 解を発見できたCSPについての平均探索時間

図 5: IHC,GA,GAHC に関する実験結果

がわかる。IHC と GAHC を比べると $d = 8 \sim 25$ では成功率 (図 5 (a)) がほぼ一致しているが $d = 0.98$ では前者が約 20% であるのに対し、後者では約 70% と優れている。また平均探索時間 (図 5 (b)) は、$d < 9$ では両者ともほぼ一致しているが、$d = 16 \sim 25$ では GAHC の方が IHC の 4 倍の速さで解の探索に成功している。また、$d = 8 \sim 25$ のときに前者の平均探索時間が d に比例して急激に増加しているのに対し、後者ではゆっくり増加している。また、GA と GAHC を比べると成功率、探索時間とも全領域で本手法が優れている。

5 おわりに

本論文では、GA と HC をハイブリッド化した CSP の解法を提案した。また、Haralick の CSP を用いて評価実験を行い、通常の GA や反復山登り法より優れていることを示した。

今後は、他の確率アルゴリズムとの比較や具体的なアプリケーションへの適用が課題であると考えている。

参考文献

[1] Haralick R.M. and Shapiro L.G. :The Consistent Labeling Problem, Part I, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PARM-1, No. 2, pp. 173-184, 1979.

[2] 西原清一：整合ラベリング問題と応用, 情報処理学会, pp.500–pp.507, 1990.

[3] Minton S., Johnston M.D., Philips A.B. and Laird P. :Solving Large-Scale Constraint Satisfaction and Sceduling Problems Using a Heuristic Repair Method, *AAAI'90*, pp. 17-24.

[4] H.M. Adorf and M.D. Johnston:A Discrete Stochastic Neural Network Algorithm For Constraint Satisfaction Problems, *IJCNN'90 III* , pp. 917–924.

[5] Minton S. et al.:Minimizing conflicts : A Heuristic Repair Method For Constraint Satisfaction and Scheduling Problems, *Artificial Intelligence*, 58, pp. 161-205, 1992.

[6] Selman B., Levesque H., Mitchell D.:A New Method for Solving Hard Satisfiability Ploblems, *AAAI'92*, pp. 440-446.

[7] Selman B. and Kautz H.:Domain-Independent Extension to GSAT : Solving Large Structured Satisfiability Proglems, *AAAI'93*, pp. 290-295.

[8] Morris P.:The Breakout Method For Escaping From Local Minima , *AAAI'93*, pp. 40-45.

[9] Davenport A., Tsang E., Wang C.J. and Zhu K.:GENET: A Connectionist Architecture for Solving Constraint Satisfaction Problems by Iterative Improvement, *AAAI'94*, pp. 325-330.

[10] Yugami H., Ohta Y. and Hara H.:Improving Repair-based Constraint Satisfaction Methods, *AAAI'94*, pp. 344-349.

[11] Goldberg D.E.:Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.

[12] 北野宏明: 遺伝的アルゴリズム, 産業図書, 1993.

[13] 松本, 内野, 狩野, 西原: 遺伝的アルゴリズムによる制約充足問題の解法, 情報処理学会第 *50* 回全国大会 *5Q-1*, 1995.