特　集　「制約充足問題の基礎と応用」

制約充足問題の近似解法

Approximate Algorithms for Constraint Satisfaction Problems

狩野　均*
Hitoshi Kanoh

*　筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba.

1997年3月3日　受理

Keywords: search, solution, hillclimbing, neural network, genetic algorithm.

1. まえがき

制約充足問題(以下CSPと称す)は，離散値をとるいくつかの変数に割り当てられる値の組合せのうち，与えられた制約をすべて満たす組合せを発見する探索問題である．制約を充足するという視点は，プラニング，スケジューリング，デザイン，ビジョン，言語など多くの分野で用いられている．代表的な例題としては，設備・部品などのレイアウト問題，時間割・会議室割当てなどの編成問題，論理式の充足可能性問題，地図の色塗り問題などがあげられる．

CSPの解法としては，他のAI分野と同様，厳密解法と近似解法が競合している．前者は，縦型探索に代表されるように，バックトラックにより，必要ならすべての可能性を探索するもので，アルゴリズムの完全性を保証している．ここで，アルゴリズムの完全性とは，解があれば必ず発見できること，また解がなければ解がないことを判定できることをいう．また，後者は，山登り法に代表されるように，これを犠牲にして，任意の一つの解を高速に求めようとするものである．

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CSPの解法は大きく分けて次の三つの方向から研究されている．第一は，汎用の探索アルゴリズムで効率良く解ける問題の部分クラスを見いだすことや，並列処理による高速化を目的とするものである．第二は，個別の応用問題が与えられたとき，問題領域の知識を利用して発見的な解法を開発するというものである．第三は，近似解法によって，実用的な時間内になるべく高い確率で解を求めようとするものである．本解説は，第三の方向からの研究に関するものである．

組合せ最適化問題(以下COPと称す)が目的関数を最適化することを目標とするのに対して，CSPは，任意の実行可能解を一つ発見することを目標とする場合が多い．これは，現実世界における人間の行動や判断が，何かを最適化するというより，いくつかの制約条件を満たす範囲で，ランダムに行われていると考える立場に立っているともいえる．CSPは，制約の充足度を目的関数と考えると，COPとみなすことができるが，次の特徴を有している．

(1)　目的関数が明示的に表現できない．
(2)　すべての制約を満たす完全解を求めることを目標とする(準最適解ではない)．
(3)　解が得られたことを判定できる．

このため現状では，CSPの近似解法として，比較的ロバストな探索手法が研究されている．本解説では，山登り法，ニューラルネットワーク，遺伝的アルゴリズムに分類して，代表的な近似解法を紹介する．これらは，制約の充足度をそれぞれ，ヒューリスティック関数，エネルギー関数，適応度関数として，ランダムに生成した解候補の制約違反数が減少する方向に探索を進めるものである．

なお，本解説ではこのように広い分野の解法を説明するため，記号や表記法を著者の判断で統一させていただいた．したがって，原論文のものとは異なっているので注意されたい．また，ページ数の関係で，読者は，相互結合型ニューラルネットワークと遺伝的アルゴリズムの基礎的なことは知っているものと仮定した．これらについては[Goldberg 89，北野 93，中野

89]を参照されたい．

2．制約充足問題と解法の戦略

本章では，制約充足問題(CSP)の定義と，CSPの近似解法でよく用いられる戦略について述べる．探索における値の選択戦略として制約違反最少化戦略，ならびに局所最適解からの脱出方法のメタ戦略として，リスタート戦略とブレークアウト戦略について述べる．

(1) CSPの定義

本解説では，CSPを次のように定義する．まず，n個の変数X_i $(i=1, \cdots, n)$とそれら変数のとる離散的な値の領域D_iを考える．次にこれらの間に，m個の二項制約$C_j(X_p, X_q)$が与えられており，$C_j(X_p, X_q)$は，変数X_pとX_qが同時にとり得る値の組を明示的に表すものとする．CSPの例を次に示す．

変数：X_1, X_2, X_3, X_4
領域：$D_1=D_2=D_3=D_4=\{a, b, c\}$
制約：$C_1(X_1, X_2)=\{(a, b), (b, c)\}$,
$C_2(X_2, X_3)=\{(c, b), (b, a), (b, b)\}$,
$C_3(X_1, X_4)=\{(a, c)\}$
解：$(X_1, X_2, X_3, X_4)=(a, b, a, c)$, (a, b, b, c)

CSPの一般的な定義については，本特集の他の解説[西原 97]に詳しく述べられている．

以下では，n個の変数のとる値の組を解候補と呼び，$X=(c\ b\ a\ b)$のように書く．また，各変数X_iの制約違反数をCV_iと書く．上記のCSPでは，$X=(c\ b\ a\ b)$に対して，X_1は制約C_1とC_3に違反しているので$CV_1=2$となる．また，一般に集合Sの要素数を$|S|$と書く．

本解説では，変数の数に対する制約の数(m/n)を制約密度と定義する．一般に制約密度の低いCSPは探索の手掛りとなる制約の数が少ないため，局所最適解が多く存在し，解くことが難しいとされている．図1は探索空間における目的関数の形状(以下コストサーフェスと称す)を表している[Morris 93](横軸は状態すなわち解候補を示す)．図1の上の図は制約密度が高い場合，下の図は低い場合に相当している．

(2) 制約違反最少化戦略

変数の集合と二項制約の集合が与えられており，各変数には，ある特定の値が代入されているものとする．ここで，制約に違反している変数を一つ選択し，その変数に制約違反数が最少となるような値を代入することを制約違反最少化戦略という．また，違反数が同じになる値が複数あった場合は，ランダムに選択するものとする．

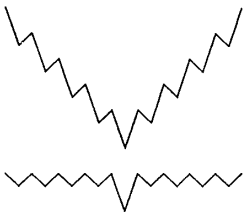

図1 探索空間におけるコストサーフェスの例
(横軸：状態，縦軸：制約違反数)[Morris 93]

(3) リスタート戦略

与えられた初期値を局所探索法で反復改善して解を求めるとき，局所最適解に陥った場合は，現在の解候補を放棄して，別の初期値から再スタートする戦略である．初期値の生成方法としては，ランダムに生成する方法とグリーディ法で生成する方法が一般的である．コストサーフェスが図1の下の図のような場合は，この戦略が適していると思われる．

(4) ブレークアウト戦略

各制約に重みを設定し，重みの合計を目的関数とする．そして，局所最適解に捕捉された場合には，現在違反している制約の重みを変更することによって，目的関数の形状を変化させて局所最適解自体をなくしてしまうという戦略である．ただし，これによって別の局所最適解が発生することもある．コストサーフェスが図1の上の図のような場合はこの戦略が適していることが指摘されている[Morris 93]．

3．山登り法を用いた解法

3・1 概　　要

[Minton 90, Minton 92]は，制約違反最少化戦略を山登り法に適用することにより，大規模なnクイーン問題とグラフ彩色問題を高速に解くことができることを示した．この方法は，MCHC(Min Conflict Hill Climbing)と呼ばれ多くの文献で引用されているが，制約密度の低いCSPに対しては，探索途中で局所最適解に捕捉される危険性が高いことが知られている．このため，SelmanらはMCHCにリスタート戦略を適用した充足可能性問題の解法(GSAT)を提案した[Frank 96, Kask 95, Kask 96, Selman 92, Selman 93]．また，[Morris 90, Morris 93]は，MCHCにブレークアウト戦略を適用したブレークアウト法(またはIDM)を提案し，充足可能性問題とグラフ彩色問題に対して，GSATやMCHCよりも優れていることを示している．以下では，[Ackley 87]より，HC(MCHC)，IHC(Iterated HC：GSAT)，SHC(Stochastic HC)ならびに，[Morris 93]よりブレークアウト法を紹介する．

3・2 アルゴリズム

(1) HCのアルゴリズム

[**step 1：初期値の生成**] 各変数 X_i に各領域 D_i の値をランダムに代入し，これを解候補とする．

[**step 2：終了判定**] 各変数の制約違反数 CV_i を計算し，制約違反がまったくなければ「成功」として探索を終了する．

[**step 3：変数の選択**] 制約に違反している変数を一つ選択する．

[**step 4：値の選択**] その変数に制約違反数が最少となるような値を代入する(同数のときはランダム)．

[**step 5：山登り**] step 2へ戻る．ただし，指定された回数繰り返しても制約違反数が減少しない場合は，「失敗」として探索を終了する．

(2) IHCのアルゴリズム

HCのstep 5において，「失敗」した場合に，次のstep 6へ進む．

[**step 6：リスタート**] step 1へ戻る．ただし，指定回数繰り返しても「成功」しない場合は，「失敗」として探索を終了する．

(3) SHCのアルゴリズム

HCのstep 4を次のように変更する．

[**step 4.1：値の選択**] その変数に代入する値の候補をランダムに選択し，この値を代入したときの制約違反数を計算する．

[**step 4.2：遷移確率の計算**] 次の確率 P によって，その値を採用するかどうかを決定する．

$$P=1/(1+\exp(\Delta/T))$$

ここで，Δ はstep 4.1で計算した制約違反数の合計からstep 2で計算した制約違反数の合計を引いた数，T は定数である．

(4) ブレークアウト法のアルゴリズム

HCが解候補の制約違反数をコストとしているのに対して，ブレークアウト法は，各制約に重みを設定し，重みの合計をコストとする．ここで，重みの初期値は1とする．この方法では，まず，局所最適解に陥るまでHCを行い，局所最適解に陥ったときは，そこから抜け出すまで，違反している制約の重みを1ずつ増加させる．そして，局所最適解から抜け出したら，またHCを行う．以下にアルゴリズムの例を示す．

[**step 1：初期値の生成**] 各変数に各領域内の値をランダムに代入し，これを解候補とする．

[**step 2：終了判定**] 制約違反数している変数に対して，重みの合計を計算し，この合計が0ならば「成功」として探索を終了する．

[**step 3：局所最適判定**] どの変数の値を変更しても重みの合計が減少しないときは，違反している全制約の重みを1だけ増加させる．また，減少するときは，制約違反最少化戦略によって値を更新する．

[**step 4：山登り**] step 2へ戻る．ただし，指定回数繰り返しても「成功」しない場合は，「失敗」として探索を終了する．

4. ニューラルネットワークを用いた解法

4・1 概　　要

Hopfield型ニューラルネットワーク(以下NNと称す)は，組合せ最適化問題(以下COPと称す)の準最適解を求める方法として広く研究されている[中野89]．[Tagliarini 87]はこれをCSPに適用し，例題として，n クイーン問題とグラフ彩色問題を解いた．この方法は，与えられた制約の充足度をエネルギー関数として，これに，各変数は一つの値をとるという制約の違反数をペナルティとして付加したものである．この方法は，COPでは常套手段とされているものであるが，一般に局所最適解に陥りやすいことが知られている．この解はCOPでは準最適解となる可能性はあるものの，CSPでは解とみなすことはできない．

[Adrof 90]は，Hopfieldネットワークのほかに，ガードネットワークと呼ばれるニューロンを付加することによって，発火するニューロン数が変数の数以下となる局所最適解を回避する方法を提案している．また，値の選択規則として，制約違反最少化戦略を導入した．このNNは，GDSネットワークと呼ばれているが，ネットワークが非対称なので収束性は保証されていない．このため，リスタート戦略を適用している．

[Davenport 94]はGDSに対して，局所最適解からの脱出方法としてブレークアウト戦略を適用したGENETと呼ばれるNNを提案している．また，ネットワークの構造を拡張して，多項制約・否定制約を扱えるようにした．以下では，[Adrof 90, Davenport 94]をもとにGDSとGENETを紹介する．

4・2 GDSネットワークによるCSPの解法

Hopfieldモデルでは，解候補 $X=(a\ b\ a\ c)$ を変数番号を行，値を列とする図2のような行列で表現する．図2で，白丸は発火しているニューロン，黒丸は発火していないニューロンを表すものとする．ここで，ニューロンの内部状態を x_{ij} とすると，次のように書ける．

$$[x_{ij}]=\begin{pmatrix}1&0&0\\0&1&0\\1&0&0\\0&0&1\end{pmatrix}$$

ここに，i：変数，j：値

また，ニューロン(i,j)と(m,n)の結合係数$w_{ij,mn}$を次のように決める．

$$w_{ij,mn}=\begin{cases}0 & (\text{制約充足})\\ -\omega & (\text{制約違反})\\ -\eta & (i=m, j\neq n)\end{cases}$$

ただし$\omega,\eta>0$，$w_{ij,mn}=w_{mn,ij}$，$w_{ij,ij}=0$とする．**2章**で示したCSPの例に対しては，次のようになる(下記以外のものは$w_{ij,mn}=-\omega$)．

制約 C_1：$w_{1a,2b}=w_{1b,2c}=0$

制約 C_2：$w_{2c,3b}=w_{2b,3a}=w_{2b,3b}=0$

制約 C_3：$w_{1a,4c}=0$

変数 X_1：$w_{1a,1b}=w_{1b,1c}=w_{1c,1a}=-\eta$

変数 X_2：$w_{2a,2b}=w_{2b,2c}=w_{2c,2a}=-\eta$

変数 X_3：$w_{3a,3b}=w_{3b,3c}=w_{3c,3a}=-\eta$

GDSネットワークは，Hopfieldネットワークに，ガードネットワークと呼ばれるニューロンを付加したものである．図3において，各行のガードニューロンは，その行の全メインニューロンと結合している．ガードニューロンの状態をx_i^gとして，ガードニューロンおよびメインニューロンへの入力の総和y_i^gおよびy_{ij}を次式で計算する．

$$y_i^g=-\theta\sum_j x_{ij}+\gamma \qquad (\theta>\gamma>0) \tag{1}$$

$$y_{ij}=\sum_{mn} w_{ij,mn}x_{mn}+\beta+\phi x_i^g \tag{2}$$

GDSは以下の手順で動作する．

[**step 1**] すべてのメインニューロンをoffにする．

[**step 2**] 変数(行)をランダムに選択する(k行が選ばれたとする)．

[**step 3**] x_k^gを次式で計算する．すなわち，ガードニューロンを先に更新する．

$$x_i^g=\begin{cases}1 & (y_i^g\geq 0)\\ 0 & (y_i^g<0)\end{cases}$$

[**step 4**] y_{kj} $(j=1,2,\cdots)$を計算する．

[**step5**] k行のメインニューロンのなかで，出力と入力の差d_{kj}が最も大きいニューロンの状態を変化させる．すなわち，値の選択方法として制約違反最少化戦略を適用する．

$$d_{kj}=\begin{cases}y_{kj} & (x_{kj}=0, y_{kj}\geq 0)\\ |y_{kj}| & (x_{kj}=1, y_{kj}<0)\end{cases}$$

[**step 6**] step 2へ戻る．

以上の手順において，Step 3で，$x_i^g=1$となるのは，式(1)よりすべてのjに対して$x_{ij}=0$のときである．したがって，式(2)のϕを十分大きくとっておくと，step 5でx_{kj}のどれかはoff→onとなる．GDSではこのようにして，1行のメインニューロンがすべてoffになるという局所最適解を回避している．GDSネットワークは非対称なので，収束性は保証されていない．そこで，一定回数試行しても収束しないときは最初からやり直す戦略(リスタート戦略)をとっている．

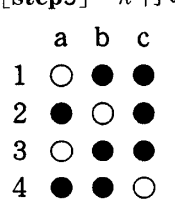

図 2 Hopfield ネットワーク

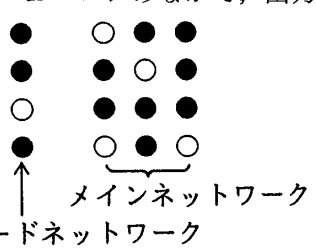

図 3 GDS ネットワーク

4・3 GENETによるCSPの解法

(1) アルゴリズム

GENETでは，GDSと同様のメカニズムにより，変数に値が完全に(唯一)割り当てられていることが仮定されている．各ニューロンへの入力の総和は，次式で定義されている．

$$y_{ij}=\sum_{mn} w_{ij,mn}x_{mn}$$

GENETは以下のように動作する．

[**step 1**] 各行からランダムに1ニューロンを選んでonとする．すなわち変数をランダムに初期化する．

[**step 2**] 1行のなかで最も入力の大きいニューロンをonとする．すなわち値の選択として制約違反最少化戦略を適用する(同点のときはランダム)．

[**step 3**] すべての行(変数)に対してstep 2を行う．ただし，値が更新されなかった変数に対しては，次の①または②を実行する．

① onニューロンへの入力の総和がすべて0なら「成功」として探索を終了．

② その他の場合は次に示す「学習」を実行．

[**step 4**] step 3を繰り返す．

(2) 学 習

ここでの学習とは，ブレークアウト戦略により局所最適解から脱出することを意味している．すなわち，制約違反しているニューロン間の結合係数を次式で調整する．

$$w'_{ij,mn}=w_{ij,mn}-x_{ij}x_{mn}$$

ここで，$w'_{ij,mn}$は調整後の結合係数である．この学習は，制約違反のコストを増加させることによって，局

所最適解近傍のコストの穴を埋める効果を持つが，学習の結果，新たな局所最適解が生成される可能性があることが指摘されている．

5．遺伝的アルゴリズムを用いた解法

5・1 概　　要

遺伝的アルゴリズム(GA)は生物の進化過程をモデル化した確率探索アルゴリズムであり，大域的探索能力が高いため，多くの分野においてCOPの近似解法として利用されている．しかしGAは，局所探索能力が低いため，完全解を必要とするCSPに適用する場合は，HCなどの局所探索法とハイブリッド化する必要があることが指摘されている．[Dozier 94]は，MGA(マイクロGA：集団サイズの小さいGA[Karr 91])の遺伝的操作にHCを取り入れ，GAの各世代ごとにHCを1ステップずつ実行し，局所最適解に捕捉された場合は，ブレークアウト戦略を適用する方法を提案している．また，[Bowen 95]はこのMGAに，CSPの厳密解法でよく用いられる，辺整合のチェック操作を取り入れることにより，探索空間の縮小と，「CSPが解けないことの判定」を達成している．[松本 95]と[Kanoh 95]は，GAで得られたエリート個体に対してHCを適用する方法を提案した．この方法は，IHCにおける初期値の生成をGAで行うというものであるが，IHCよりも高速に収束することが実験的に示されている．さらに[Kanoh 96a，狩野 96b]は，CSPの部分解をウイルスと見立て，GAの遺伝的操作としてウイルス感染を導入することにより，制約密度の低いCSPを高速に解く方法を提案している．

また，GAの拡張として，進化論的計算モデルによるCSPの解法も提案されている[Bowen 96，Hao 95，平岡 96]．

5・2 MGAによるCSPの解法

以下では，GAの習慣に習って解候補を個体と称する．各個体は適応度のほかにpivotと呼ばれる属性を持つ．また，個体中の各変数は，値と制約違反数のほかにh値と呼ばれる属性を持つ．MGAは，このpivotとh値を用いて遺伝的操作のなかにMCHCを取り込んでいる．

MGAのアルゴリズムを図4に示す．以下，図4に沿って説明する．

(1) 初期集団の生成

[**step 1**] 個体を一つランダムに生成．

[**step 2**] この個体の変数をランダムに選択し，これにランダムな値を代入して新しい個体とする．

[**step 3**] 個体数$=N$まで[step 2]を繰り返す．すなわち，step 1で生成した個体から$(N-1)$個の個体を生成する．

(2) 適応度の評価

次式で適応度を計算する(nは個体のサイズ)．

$$f=\sum_{i=1}^{n}CV_i+\sum_{i=1}^{n}\sum_{j=1}^{n}B_{ij}(b)$$

$$B_{ij}(b)=\begin{cases}b & (X_i \text{ と } X_j \text{ が制約違反})\\ 0 & (\text{その他})\end{cases}$$

N_1をエリート個体より適応度が悪く生成された子個体の総数とする．また，Rをエリート個体のなかで制約違反のある変数の集合として，N_2を次式で定義する．

$$N_2=\sum_{i\in R}|D_i|$$

ここで，$N_1>N_2$が成立したときbを1ずつ増加させる．ただし，bの初期値は0とする．

(3) 個体の選択

線形ランク戦略で2個体を選択する．選択された個体は，(5)の突然変異を施された後次世代に残され，代わりにランクの低い2個体が削除される．また，ランク1～$(N-2)$までの個体はそのまま次世代に残される．

(4) 変数の選択

選択された2個体に対して次のs値を計算する．

$$s_i=CV_i+h_i \qquad (i=1,\cdots,n)$$

次に，s値が最大の変数をpivotとして選択する．ここではp番目の変数が選択されたとする．また，h値の初期値は0とする．

(5) 突然変異

pivotに対してランダムな値$v_p(\in D_p)$を代入する．

(6) h値の更新

突然変異による制約違反数の変化量から次式でh値を更新する．ただし，h'_pは更新後のh値とする．

$$h'_p=\begin{cases}h_p-1 & (X_p \text{ の制約違反数が増加})\\ h_p & (X_p \text{ の制約違反数が減少})\end{cases}$$

以上のアルゴリズムにおいて，h値の初期値は0なので，初めはCV_iが最大の変数が選択される．しかし，制約違反している変数のh値が減少すると，いずれは，制約違反していない変数も選択されることになる．また，pivotを継承する個体を追跡すると，最

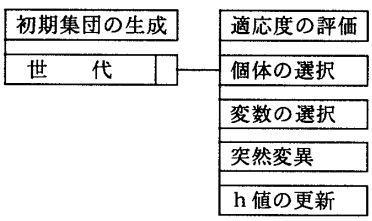

図 4 MGAのアルゴリズム

表 1　FT の内容

familyを定義しているorder=n-1のスキーマ
pivotの位置(=pとする)
family のＩＤ番号
D_pの中でまだ割り当てられていない値の集合
集団中のfamilyメンバーの数
不適合ベクトル

後にはその変数に対して，CV_i を最少にする値が選択されることになる．

5・3　AGSA による CSP の解法

AGSA の個体のデータ構造は MGA と同じだが，属性として，適応度と pivot のほかに family の ID 番号を持つ．また，遺伝的操作として MHIC(Multiple-point Heuristic Interval Crossover)と呼ばれる交叉を導入した．

family とは，同じ変数を pivot とし，pivot 以外の変数は同じ値をとる個体の集団として定義されている．familiy は表 1 の FT(Family Table)で管理されている．表 1 で，不適合ベクトル z_i $(i=1, \cdots, n;\ i \neq p)$ とは，pivot 以外の変数に対応する成分からなり，初期値は，全成分 0 とする．pivot に値が割り当てられたとき，これと衝突する(制約違反となる)変数に対応する成分に 1 を加算するものとする．family を導入した目的を次に示す．

- 集団内の個体の重複を避け，多様性を維持．
- 突然変異を改良：pivot の値を FT の 4 行目からランダムに選択．
- Arc-Revision により探索空間を縮小：$z_i=|D_i|$ となったときの X_i の値を，D_i から削除．
- CSP が解けないことを判定：D_i が空となったときに，X_i と X_p は辺不整合であると判定．

交叉(MHIC)は，選択された親 1 と親 2 から 1 個の子を生成する．まず親 1 の制約違反していない変数の値を子にコピーする．違反している変数は，親 1 と親 2 に対して，$P_{cross}=0.5(1+1/|R|)$ の確率で次の IC (Interval Crossover)により値を決める．

[IC のアルゴリズム]

ある変数に対して，子の値を親 1 と親 2 の間の値からランダムに選択する．すなわち，f_i を D_i から $\{1, \cdots, |D_i|\}$ への上への 1 対 1 写像，f_i' を f_i の逆写像，また，v_i(個体 1)を個体 1 の X_i の値としたとき，

$$v_i(\text{子})=f_i'(k)$$

$$f_i(v_i(\text{親 1}))<k<f_i(v_i(\text{親 2}))$$

とするものである．

6. あ と が き

1995 年の IJCAI で“Systematic vs. Stochastic Constraint Statisfaction”というタイトルのパネル討論が行われた[Freuder 95]．これは，CSP の厳密解法と近似解法の特徴を整理しようという試みであった．厳密解法は大規模な問題には適さないことや，近似解法はパラメータの調整が難しいことが一般に指摘されている．厳密解法サイドの主張は，近似解法はアルゴリズムの完全性を満たしていないので，制約が矛盾を含んでいることもある現実の問題には対処できないというものである．これに対して，近似解法サイドは，アルゴリズムの完全性は，探索空間のサイズが 10^{30} 以上になると事実上達成できないことを指摘している．また，制約が矛盾している問題こそ，近似解法で，なるべく多くの制約を満たす準最適解を求めることが現実的であると主張している．定理の証明は厳密解法，モデルの発見は近似解法という結論は興味深い．

本解説では，制約充足問題の代表的な近似解法を山登り法，ニューラルネットワーク，遺伝的アルゴリズムに分類して紹介した．著者の浅学のため不適切な記述があったかもしれない．この点は，賢明な読者の方々よりご意見をいただければ幸いである．

最後に，貴重なご意見をいただいた，本学電子情報工学系西原清一教授，構造工学系星野　力教授，ならびに文献調査にご協力いただいた，本学理工学研究科松本美幸，水野一徳の両君に感謝の意を表します．

◇ 参　考　文　献 ◇

1．制約充足問題

[Freuder 95]　Freuder, C. E., *et al.*: Systematic Versus Stochastic Constraint Satisfaction, *IJCAI-95*, pp. 2027-2032(1995).

[Mitchell 92]　Mitchell, D., Selman, B. and Levesque, H.: Hard and Easy Distributions of SAT Problem, *Proc. Nat'l* Conf. on Artif. Intel., pp. 459-465(1992).

[西原 97]　西原清一：制約充足問題の基礎と展望，人工知能学会誌, Vol. 12, No. 3(1997).

[Tsang 93]　Tsang, E. P. K.: *Foundations of Constraint Satisfaction*, Academic Press(1993).

2．山登り法

[Ackley 87]　Ackley, H. D.: *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers (1987).

[Frank 96]　Frank, J.: Weighting for Godot: Learning

Heuristics for GSAT, *AAAI-96*, pp. 338-343(1996).
[Kask 95] Kask, K. and Dechter, R.: GSAT and Local Consistency, *IJCAI-95*, pp. 616-622(1995).
[Kask 96] Kask, K. and Dechter, R.: A Graph-Based Method for Improving GSAT, *AAAI-96*, pp. 350-355(1996).
[Minton 90] Minton, S., Johnston, M. D., Philips, A. B. and Laird, P.: Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method, *AAAI-90*, pp. 17-24(1990).
[Minton 92] Minton, S., *et al.*: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, *Artif. Intell.*, Vol. 58, pp. 161-205(1992).
[Morris 90] Morris, P.: Solutions Without Exhaustive Search: An Iterative Descent Method for Binary Constraint Satisfaction Problems, *AAAI-90* (1990).
[Morris 93] Morris, P.: The Breakout Method for Escaping from Local Minima, *AAAI-93*, pp. 40-45(1993).
[Selman 92] Selman, B., Levesque, H. and Mitchell, D.: A New Method for Solving Hard Satisfiability Problems, *AAAI-92*, pp. 440-446(1992).
[Selman 93] Selman, B. and Kautz, H.: Domain-Independent Extension to GSAT: Solving Large Structured Satisfiability Problems, *IJCAI-93*, pp. 290-295(1993).
[横尾 94] 横尾 真: 弱コミットメント戦略を用いた制約充足問題の解法, 情処学論, Vol. 35, No. 8, pp. 1540-1548(1994).
[Yugami 94] Yugami, H. Ohta, Y. and Hara, H.: Improving Repair-based Constraint Satisfaction Methods, *AAAI-94*, pp. 344-349(1994).

3. ニューラルネットワーク

[Adorf 90] Adorf, H. M. and Johnston, M. D.,: A Discrete Stochastic Neural Network Algorithm for Constraint Satisfaction Problems, *IJCNN-90*, pp. 917-924(1990).
[Dahl 87] Dahl, E. D.: Neural Network Algorithm for an NP-Complete Problem: Map and Graph Coloring, *ICNN-87*, Vol. III, pp. 113-120(1987).
[Davenport 94] 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(1994).
[喜多 92] 喜多 一: Hopfield 型ニューラルネットワークとシミュレーテッドアニーリング, 人工知能学会誌, Vol. 7, No. 6, pp. 970-979(1992).
[中野 89] 中野 馨 監修, 飯沼一元 編集, ニューロンネットグループ+桐谷 滋 著: ニューロコンピュータ, 技術評論社(1989).
[Tagliarini 87] Tagliarini, C. A. and Page, E. W.: Solving Constraint Satisfaction Problems with Neural Networks, *ICNN-87*, Vol. III, pp. 741-747(1987).

4. 遺伝的アルゴリズム

[Bowen 95] Bowen, J. and Dozier, G.: Solving Constraint Satisfaction Problems Using A Genetic/Systematic Search Hybrid That Realizes when to Quit, *ICGA-95*, pp. 122-129(1995).
[Bowen 96] Bowen, J. and Dozier, G.: Constraint Satisfaction Using A Hybrid Evolutionary Hill-climbing Algorithm That Performs Opportunistics Arc and Path Revision, *AAAI-96*, pp. 326-331(1996).
[Clearwater 94] Clearwater, S. H. and Hogg, T.: Exploiting Problem Structure in Genetic Algorithms, *AAAI-94*, pp. 1310-1315(1994).
[Dozier 94] Dozier, G., Bowen, J. and Bahler, D.: Solving Small and Large Scale Constraint Satisfaction Problems Using a Heuristic-Based Microgenetic Algorithm, *IEEE Proc. ICEC'94*, pp. 306-311(1994).
[Eiben 94] Eiben, A. E., *et al.*: Solving Constraint Satisfaction Problems Using Genetic Algorithms, *IEEE Proc. ICEC'94*, pp. 542-547(1994).
[Goldberg 89] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley(1989).
[Hao 95] Hao, J.-K. and Dorne, R.: An Empirical Comparison of Two Evolutionary Methods for Satisfiability Problems, *IEEE Proc. ICEC'95*, pp. 450-455(1995).
[Holland 75] Holland, J. H.: Adaption in Natural and Artificial System, The University of Michigan, 1975, and MIT Press(1992).
[平岡 96] 平岡敏洋, 堀内 匡, 片井 修ほか: 制約充足問題に対する進化論的計算アプローチ, 計測自動制御学会第23回知能システムシンポジウム, pp. 69-74(1996).
[Kanoh 95] Kanoh, H., Matsumoto, M. and Nishihara, S.: Genetic Algorithms for Constraint Satisfaction Problems, *IEEE Proc. SMC'95*, pp. 626-631(1995).
[Kanoh 96a] Kanoh, H., Hasegawa, K., Matsumoto, M., Kato, N. and Nishihara, S.: Solving Constraint Satisfaction Problems by a Genetic Algorithm Adopting Viral Infection, *IEEE Proc. INBS'96*, pp. 67-73(1996).
[狩野 96b] 狩野 均, 長谷川和代, 松本美幸, 西原清一: ウイルス進化論に基づく制約充足問題の解法, 計測自動制御学会第23回知能システムシンポジウム, pp. 75-80(1996).
[Karr 91] Karr, C. L.: Air-Injected Hydrocyclone Optimization via Genetic Algorithm, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold(1991).
[北野 93] 北野宏明 編: 遺伝的アルゴリズム, 産業図書(1993).
[松本 95] 松本美幸, 狩野 均, 西原清一: 遺伝的アルゴリズムによる制約充足問題の解法, 情処研報, Vol. 95, No. 86, pp. 33-40(1995).
[松本 97] 松本美幸, 狩野 均, 西原清一: 制約違反最小化戦略に基づくハイブリッドGAによる制的充足問題の解法, 情処学論, Vol. 38, No. 5, (1997).
[Warwick 94] Warwick, T. and Tsang, E. P. K.: Using a genetic algorithm to tackle the processors configuration problem, *ACM Proc. SAC'94*, pp. 217-221(1994).

著 者 紹 介

狩野 均(正会員)

1978年筑波大学第一学群自然学類卒業. 1980年同大学院理工学研究科修士課程物理学専攻修了. 同年, 日立電線(株)入社. 同社オプトロシステム研究所において人工知能・神経回路の応用に関する研究に従事. 1993年より筑波大学電子・情報工学系. 現在, 同助教授. 制約に基づく知識表現, 遺伝的アルゴリズムの研究に従事. 工学博士. 1992年電気学会論文賞受賞. 電気学会, 情報処理学会, 計測自動制御学会等各会員.