

ソフト制約処理によるインタラクティブコンピューティング

Interactive Computing by Soft Constraints Processing

須藤康裕*1 栗原正仁*2 河村勇一*1
Yasuhiro Sudo Masahito Kurihara Yuichi Kawamura

*1北海道大学工学研究科
Graduate School of Engineering

*2北海道大学情報科学研究科
Graduate School of Information Science and Technology

In this paper, we propose a new kind of graphical information processing software based on Constraint Satisfaction Problems(CSP). CSP has a feature that various problems can be formulized. However, many of previous approach of developing problem solver is not correspondence of formulate various problems. Rest of them are using add-on or plug-in about all. The system named GIPSCSP was developed under those concepts[Kawamura03]. GIPSCSP is able to deal various problems visibly whereas GIPSCSP is ambiguous for a CSP. And so, we have a suggestion for extending GIPSCSP. We apply a type of fuzzy CSP that named HDFCSP and its solving method to GIPSCSP. Fuzzy CSP and HDFCSP are being considered having intermediate satisfaction degrees between complete satisfaction and complete violation, and their constraints are corresponding membership functions. Furthermore, HDFCSP can deal variables that is having discrete domains and continuous domains. A Spread Repair algorithm is a solving method for HDFCSP. We propose introduction of HDFCSP and Spread Repair algorithm to GIPSCSP.

1. はじめに

人工知能のアプローチの一つとして、問題の定式化とその問題解決の構築がある。制約充足問題 (CSP:Constraint Satisfaction Problem) は問題を変数と、変数間に設けられた制約の集合として表現するモデルである。制約には、時間制約、空間制約、論理制約など様々なものが存在し、これらによって多くの問題が CSP として定式化可能である。このような特徴から、現在、プランニング、スケジューリング、各種レイアウト決定など様々な分野で応用されており、今後もさらに多くの分野への適用が予想される。しかしながら CSP はあまりにも単純なモデルのため、問題の定式化が困難である場合が存在し、それに対して様々なタイプの CSP が研究されている。その一つに、ソフト制約充足問題がある。これには、制約に優先度を設けたり充足する制約の数を最大化するようなものが含まれ、厳密な意味での「充足」とは言えない部分もある。これらは制約処理と呼ばれることもあり、ファジィ CSP もその一つである。

計算機を用いて行われる処理は、ユーザインタフェース次第で使用感、理解度が大きく変化する。グラフィカルユーザインタフェース (GUI:Graphical User Interface) はユーザインタフェースの1つであり、マウス操作などを用いてユーザが計算機に情報を入力し、グラフィックスによって計算機がユーザに対して情報を表示する。これは、現在の計算機システムにおいて広く用いられており、計算機の操作を簡易化し、処理の直感的理解をもたらす。

制約処理は、様々な問題を扱うことができるという特徴を有している。1つの問題に特化したソフトウェアや、機能のアドオンにより問題解決を行うソフトウェアはもちろん存在するが、ユーザ自らが自由に様々な問題を定式化し、解決するといったソフトウェアは一般的ではない。河村らによって開発された GIPSCSP(Graphical Information Processing Software based on CSP) は、独自の有向グラフを用いて CSP を表現し、視覚的に問題を構築することができる。

本稿では GIPSCSP に対し、ソフト CSP の一つである HD-

FCSP(Hybrid Domain Fuzzy CSP) およびその解法 Spread Repair アルゴリズムを導入した新しいグラフィカル情報処理ソフトウェアの開発指針を提示する。

2. 制約充足問題

CSP は、変数の集合 $X = \{x_i\}_{i=1}^n$ とその値域となる領域の集合 $D = \{D_i\}_{i=1}^n$ および任意の変数間に設けられる制約の集合 $C = \{C_k\}_{k=1}^r$ の組として定義される。制約 C_k は X の部分集合 $S_k (S_k \subseteq X)$ 上の関係 R_k であり、 S_k に含まれる各変数が同時に取り得る値の組合せを表している。すなわち、 $S_k = \{x_{k_1}, \dots, x_{k_w}\}$ ならば $R_k \subseteq D_{k_1} \times \dots \times D_{k_w}$ である。 S_k を R_k の範囲 (scope) という。また、ここで $w = 2$ であるとき制約は 2 項関係であるといい、 $w = 1$ の場合には単項関係、 $w \geq 3$ の場合を多くに多項関係という。例えば変数 x_1 は常に x_2 よりも大きい値でなければならないという制約 C_k があれば、 C_k の範囲は $S_k = \{x_1, x_2\}$ となり、それぞれの変数は (x_1, x_2) が

$$R_k = \{(x_1, x_2) | x_1 > x_2, x_1 \in D_1, x_2 \in D_2\}$$

の元となるように選ぶ。

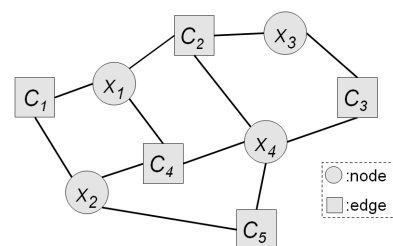


図 1: An example of a hypergraph

CSP は無向グラフによって表現することが可能であり、変数がノード、制約がエッジにそれぞれ対応付けられる。このとき、通常のエッジでは 2 項関係しか表現できないが、ハイパーエッジを使うことにより多項制約を表すことが可能である (図

1). ハイパーエッジとは2個または3個以上のノードを結ぶエッジであり、このようなエッジを持つグラフをハイパーグラフという。このグラフを用いることにより多項制約におけるCSPを表現することができ、さらにグラフ理論における隣接や近傍などの用語をCSPの変数および制約などに対応付けることが可能であり、以下本稿でも使用する。

このような離散系の問題においては効率的で完全な解法は一般に存在せず、もし、しらみつぶしに解を探索すれば、計算時間が問題の規模に応じて指数的に増大することは避けられない。しかしながら様々な方法を駆使し、多くの場合に現実的な時間で近似解を得られるよう研究が進められてきた。CSPの解法として大きく2つ、部分的な変数割当ての逐次拡張に基づく厳密解法と、完全な変数割当ての逐次修復に基づく近似解法とがある。近年ではヒューリスティックな評価関数を用い、変数を逐次変更する手法 [Minton 92] について研究が行われ、これにより100万クイーン問題を数十ステップで解を得るなどの成果が上げられている。本稿で提案するモデルにおいても、この手法を用いて近似解を高速に探索する。

3. 混合領域制約充足問題

混合領域制約充足問題 (HDFCSP) は、1つの変数が連続領域と離散領域を複数持つ問題を扱うためのモデルであり、FCSPの変数領域を拡張して定義される。ここではFCSPおよびHDFCSPの定義を行う。

3.1 ファジィCSP

FCSPは、通常のCSPの制約にファジィ関係を導入したモデルであり、最適化問題の一種である。ファジィ制約 C_k はファジィ関係 R_k に対応付けられ、そのメンバーシップ関数 μR_k は

$$\mu R_k : \prod_{x_i \in S_k} D_i \rightarrow [0, 1] \quad \dots (1)$$

の形式で与えられる。すなわち、制約 C_k の範囲 S_k の割当て v_{S_k} に対するメンバーシップ値 $\mu R_k(v_{S_k})$ は、 $[0, 1]$ の実数値をとり、それが制約の充足度を示す。

制約 C_k と C_l のファジィ論理積 $C_k \wedge C_l$ は、 $S_k \cup S_l$ を範囲とするファジィ関係 $R_k \cap R_l$ であり、そのメンバーシップ関数を

$$\mu R_k \cap R_l(v) = \min(\mu R_k(v[S_k]), \mu R_l(v[S_l])) \quad \dots (2)$$

で定義する。ただし、 $v[S]$ は v の割当てを範囲 S に含まれる変数に制限したものの(射影)である。同様に全ての制約のファジィ論理積すなわちCSP全体の充足度を以下のように定義する。

$$\mu \bigcap_{k=1}^r R_k(v) = \min_{1 \leq k \leq r} (\mu R_k(v[S_k])) \quad \dots (3)$$

つまり、全ての制約の中で、最も制約を満たしていない制約の充足の度合いをCSPの充足度 C_{min} とする。

$$C_{min}(v) = \min_{1 \leq k \leq r} (\mu R_k(v[S_k])) \quad \dots (4)$$

ここで $C_{min}(v) > 0$ のとき、 v をファジィCSPの解という。また、最適解とは最大の充足度を与える解をいう。したがって、FCSPを解くことは、充足度が最低の制約(C^* とする)の充足度を最大化するような変数への値の割り当て v を求める最適化問題となり、以下の式がその最適解の目的関数値となる。

$$\max_v (\min_{1 \leq k \leq r} (\mu R_k(v[S_k]))) \quad \dots (5)$$

ただし、(5)式のみを考慮した場合、 C^* 以外の制約は改善されることがない。現実の応用においては、 C^* 以外の制約違反もできるだけ改善すべきであり、4節でその手法について述べる。

3.2 混合領域ファジィ制約充足問題

通常、CSPの変数領域は有限離散集合であり、その要素を列挙することができるので、全ての組み合わせを探索する木を生成することができる。しかしながら変数領域が、

- i. 要素の多い集合である
- ii. 連続値である
- iii. 連続値と離散値をとる場合がある

というような状況においては、従来のCSPおよびFCSPでは解が得難かった。本稿では上記のような場合でもCSPとして処理可能にするため、変数領域に離散および連続領域を導入した混合領域ファジィCSP (HDFCSP) を提案する。

HDFCSPはFCSPを拡張したモデルであり、領域の定義以外はFCSPと同じである。HDFCSPでは、変数の各領域を閉区間の和集合として、

$$D_i = \bigcup_{j=1}^m [l_j, u_j] \quad \dots (6)$$

であるとする。 $[l_j, u_j]$ は下限を l_j 、上限を u_j とする閉区間、すなわち $\{x | l_j \leq x \leq u_j\}$ という領域を表す。また、メンバーシップ関数は各領域で連続であるとする。ここで $l_j = u_j$ である場合、その区間は単一の実数値を取るため、離散的な領域とみなすことができる。全ての j において、 $l_j = u_j$ となる場合、それは通常のCSPもしくはFCSPの領域と同じであるので、HDFCSPはFCSPを包含している。また、全ての変数領域が唯一の連続領域を持ち、なおかつ制約が連続なメンバーシップ関数によって表現されている場合は、通常非線形計画問題とみなすことができる。

(6)式の定義により、制約充足問題および数理計画問題の両方の性質をもつ問題を扱うことが可能になり、これまでとは異なった、より柔軟な定式化が可能になる。

4. HDFCSPの解法

FCSPに関する研究では、分枝限定法 [Mese 97] やニューラルネットを用いて解が得られることが示されている。HDFCSPにおける変数領域は連続領域を含むので、探索木の生成は不可能であるが、一つの方法として、領域を離散化することによって組み合わせを有限通りにすることは可能である。しかしながら領域が広大である場合に、必要以上の精度で離散化することは探索空間を広げることになり、一方精度が足りなければより良い解を発見できない。また数理計画的な手法は、制約を表現するメンバーシップ関数の形状を限定することによって実用的な解法を得ているが、表現の自由度をなるべく犠牲にしたいくないという人工知能的な価値観にとっては負の側面がある。これに対し、本稿では単純に領域の離散化を行うのではなく、有望な候補のみを高速に採択する手法を提案する。

4.1 Spread Repair アルゴリズム

HDFCSPの解法としてアルゴリズム Spread Repair を示す。反復改善を行う近似解法であるため、得られる解が大域的最適解である保証はない。局所改善的にCSPを解く場合、何らかのヒューリスティックを用いた評価関数を使用する。本手法では(5)式を単純な評価関数として用い、一度に1変数の

値のみ変更するものとする。Spread Repair アルゴリズムは、近傍状態への推移を反復的に行う (図 2)。ここで、 C^* が一度の変更によって改善できない状態を準局所最適ということにする。また、すべての制約が平均的な充足度を下げることなく一度の変更によって改善できない状態を局所最適とし、その時の割当てを局所最適解とする。

アルゴリズムの基本的な動作は、常に C^* の改善を最優先し、準局所最適状態に陥った場合に改善の対象を隣接する変数に移行する。この繰り返しの後、アルゴリズムは局所最適状態で必ず停止する。Spread Repair アルゴリズムでは C^* に対して影響力が強い周辺の変数への割当てが変更されることにより準局所最適状態から抜け出す可能性を生じさせる。また、再帰的に改善の対象となる制約を周辺に広げていくので、 C^* 以外の制約を改善することになり、平均的な制約充足度

$$C_{ave} = \frac{1}{r} \sum_{k=1}^r C_k \quad \dots (7)$$

の値も同時に増大する (ただし山登り的な手法であるため、このことは保証されないが、経験的には増大することがわかっている)。現実問題として平均的な充足度を高めることは重要であり [Rut 94, Mese 97, Kan 95], この利点は Spread Repair アルゴリズムの大きな特徴の一つである。

”Spread Repair” は、改善の対象が周囲に広がっていく様子意味している (図 2)。アルゴリズムは全ての変数に対して改善が行えなくなった時点で停止し、そのときの割当て V を局所最適解として得る。局所最適状態から抜け出す方法については様々な研究がされており、Simulated Annealing (焼きなまし法) や、遺伝的アルゴリズムにおける突然変異がこれに該当する。また、制約の優先度を変化させる様々な手法が研究されている。

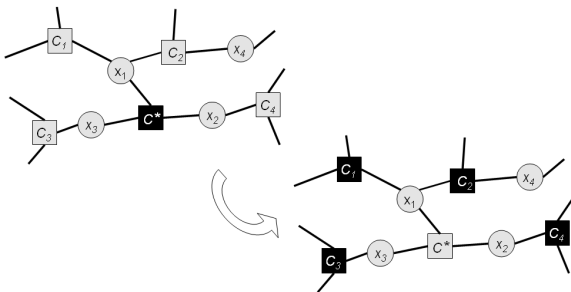


図 2: Spreading target

4.2 変更候補値の算出および選択

C^* を改善する場合、 C^* が w 項制約であるとする、その範囲 S^* は w 個の要素を持ち、その割当てが C^* の充足度に影響をもつ。Spread Repair アルゴリズムでは唯一の変数への割当てを変更するが、このとき最も C^* が改善されるように、変数とその割当てを選択する。

本稿では変更後の割当てについて、単純な離散化を行った後逐一充足度の計算を行うよりも、高速かつ高精度に求める手法を提案する。変更する変数については、範囲 S^* の要素である w 個のそれぞれに対して変更後の充足度を調べればよいので、計算コストは w に対して線形である。しかしながら、その時点での変数への最適な割当てを求めることは、メンバーシップ関数に対して最大値を与えるような割当てを探索することになり、関数の形状によっては多くの計算時間を要し、さらには

最大値を得られない場合すらある。また S^* の要素である変数 $\{x_{ij}\}_{j=1}^w$ も、他の制約の範囲に属する場合があるので、 x_{ij} の割当て v_{ij} を変更するときは複数のメンバーシップ関数を同時に考慮する必要がある (図 4)。ここで x_{ij} を範囲に含む制約の集合を $\{C_{k_h}\}_{h=1}^t (\ni C^*)$ とすると、そのメンバーシップ関数 $\mu R_{k_h}(v_{S_{k_h}})$ は範囲 S_{k_h} への割当て $v_{S_{k_h}}$ の関数であるが、 S_{k_h} の要素のうち唯一 x_{ij} のみが変わることを仮定するので、メンバーシップ値は v_{ij} に対して一意に定まり、

$$\mu R_{k_h}(v_{ij}) = \mu R_{k_h}(v_{S_{k_h}})$$

と略記することにする。

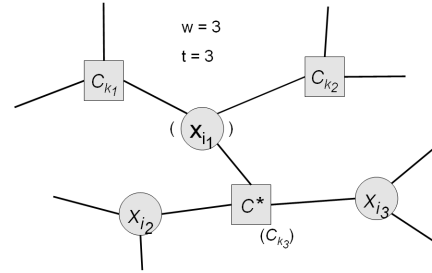


図 3: Changing x_{i1}

任意のメンバーシップ関数同士のファジィ論理積による、新しいメンバーシップ関数の全体の形状を把握するのは困難である。しかし、必要なのは最大値 $\max(\mu R_{k_h}(v_{ij}))$ を与える割当て v_{ij} だけで、関数全体の形状を厳密に求める必要はない。ここで $\max(\mu R_{k_h}(v_{ij}))$ の出現する可能性のある位置を検討する。閉区間 $[l, u]$ において、連続な任意の関数 f の最大値は $f(l)$, $f(u)$, または f の極大値のいずれかである。したがって、それぞれのメンバーシップ関数がどのような形状であったとしても、最大値は l か u , それぞれのメンバーシップ関数の極大値、あるいはある 2 つのメンバーシップ関数の交点に存在する (図 5)。Spread Repair アルゴリズムでは単純に離散化するのではなく、このような候補点から次の割当てを探索する。

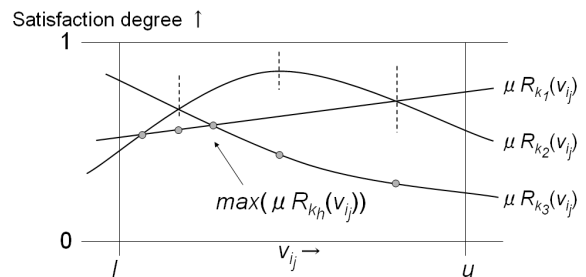


図 4: $\max(\mu R_{k_h}(v_{ij}))$

このような評価値を用いた反復改善では、変更後の充足度が同一となる変更が複数存在する場合がある。本稿ではこれを単に「非決定的に選択する」とし、具体的には指定しないことにする。

4.3 交点と極大値を求める算法

交点を求めることは、1 変数の方程式 (一般には非線形) の数値解を求めることに対応する。4 次以下の単純な代数方程式

であれば公式を用いて瞬時に数値解を計算することが可能であるが、高次代数方程式や指数関数などを含む超越方程式の数値解を求めるには、反復を用いて近似解を求めることになる。導関数が利用できるのであれば、ニュートン法などの高速な方法もあるが、制約を表現するメンバーシップ関数を限定することは、問題表現の柔軟さを欠くこととなる。これらの操作における最新の研究では、Brent のアルゴリズム [Brent 73] が知られている。この手法は収束の速さと確実性の両方を持ち、超 1 次収束する手法を監視しながら用い、超 1 次収束しないようであれば 1 次収束に切り替える。逆 2 次補間を用いたアルゴリズムで、導関数が使用できない場合に非常に有効であり、しかも収束が保証されている。

極大値を求める手法においても、Brent の手法が有効である。呼び方は同じであるが内容は異なり、極大値を求めるアルゴリズムは 2 階導関数が連続のときに放物線補間を用い、そうでないときには黄金分割法を用いる。導関数を利用できない場合でも後者の方法で収束は約束されている。Spread Repair アルゴリズムにもこの手法を用いれば、最大および交点を求める操作に関して複雑なメンバーシップ関数であっても高速に処理することが期待できる。

5. GIPSCSP の拡張

ここでは以上のような柔軟な定式化が可能である HDFCSP を、GIPSCSP に導入する。

5.1 GIPSCSP

GIPSCSP [河村 03] は河村らによって開発された CSP の視覚モデリング環境である (図 5)。変数および制約をノードとし、有効リンクで関連付けることにより CSP を表すことができる。CSP と名乗ってはいるが実際には厳密に解を得るのではなく、制約に対して優先度を設け、その重みを考慮して制約違反の数を減少させる近似解法を採用している。また、変数領域には制限があり、実数型、整数型、boolean 型のいずれかのみが許されている。その点では、通常の CSP の変数領域は有限離散集合であるので、数理計画問題あるいは整数計画問題に近い定式化になる。

論文 [河村 03] では java を用いて実装し、いくつかの問題を定式化している。論理回路の故障診断では、充足することのない回路に対し、制約を 1 つずつ停止させることにより故障箇所を発見することができるとしている。また、地図の色塗り問題 (CSP の例題として有名である) を背景に実際の地形の画像を配置することにより、違反箇所が改善されていくのが視覚的にわかるようにしている。

5.2 HDFCSP および Spread Repair の導入

GIPSCSP は、正確には CSP とは呼べないような実装になっている。また、その曖昧なモデルを厳密に定義しているわけではないので、全体的にどのような問題が定式化でき、どのような解が得られるかがはっきりしていない。そこで本研究では、GIPSCSP に対して前節までで定義した HDFCSP と、Spread Repair アルゴリズムを導入し、柔軟な解と停止性が保証される新しい視覚モデリング環境の開発を提案する。

GIPSCSP では制約の充足をコスト関数で表現し、その値が最大になるように変数の割当てを変更している。提案モデルでは、制約はメンバーシップ関数で表現し、数値計算法のアルゴリズムを用いて充足度が最大となる割当てを探索する。メンバーシップ関数の形状は一般的なものを数種類用意し、またユーザが定義したメンバーシップ関数でも Spread Repair アルゴリズムが動作するようシステムを構築する。

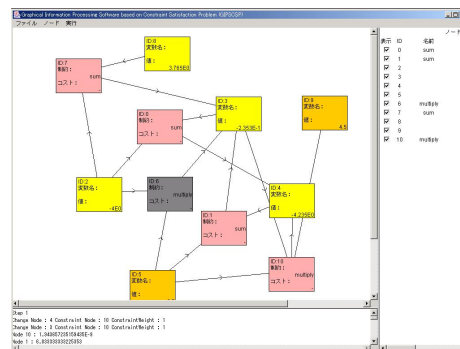


図 5: GIPSCSP

実際に HDFCSP の実装と、Spread Repair アルゴリズムの動作を確認するための実験的なソフトウェアは完成しており、ユーザ定義のメンバーシップ関数を扱うことも可能である。

このようなソフトウェアは、正しい解を得ることはもちろん、解が得られない場合の柔軟な提案が重要な役割の一つである。すなわち、問題の解決に向けて様々な可能性を指し示し、ユーザとの対話的な試行の中で最終的な結果が得られる、エキスパートシステムの一つと呼べるような環境の構築が本研究の目的である。

6. おわりに

本稿では、グラフィカル情報処理ソフトウェア GIPSCSP の拡張を提案した。今後は HDFCSP の実装と、さらなる使いやすさを追求したソフトの開発を行う予定である。

参考文献

- [Rut 94] Ruttkay, Zs. : Fuzzy constraint satisfaction, *Proceedings of 3rd IEEE Intern. Conf. on Fuzzy Systems*, Vol3, 1263-1268(1994)
- [Mese 97] Meseguer, P. and Larrosa, J. : Solving fuzzy constraint satisfaction problems, *Proceedings of 6th IEEE Intern. Conf. on Fuzzy Systems*, Vol3, 1233-1238(1997)
- [Kan 95] Kanada, Y. : Fuzzy Constraint Satisfaction Using CCM - A Local Information Based Computation Model, *Proceedings of 4th IEEE Intern. Conf. on Fuzzy Systems*, Vol4, 2319-2326(1995)
- [Minton 92] Minton, S., Johnston, M.D., Philips, A.B., Laird, P. : Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, *Artificial Intelligence* 58, 161-205(1992)
- [Brent 73] Brent, R.P. : Algorithms for minimization without derivatives. pp.195, Prentice-Hall, Inc, (1973)
- [河村 03] 河村勇一, 栗原正仁 : 制約処理を用いたインタラクティブコンピューティングの構想平成 15 年度電気関係学会北海道支部連合大会講演論文集 pp.261(2003)