2F3-07

連続領域を導入したファジィ制約充足問題の解法

Solving Fuzzy Constraint Satisfaction Problem with Continuous Domains

須藤 康裕[*1] 栗原 正仁[*1] 三田村 保[*2]
Y.Sudo M.Kurihara M.Kurihara

[*1]北海道大学 Hokkaido University
[*2]北海道工業大学 Hokkaido Institute of Technology

Fuzzy CSP(FCSP) is an extension of CSP(Constraint Satisfaction Problem), a powerful tool for solving various problems based on constraints among variables. In traditional CSP and FCSP, values for the variables are chosen from the discrete domains.However, this is often inconvenient when one wants to express real world problems.

In this paper, we present a new technique that allows variables to have a mixture of discrete and continuous domains.We show that this model, called HDFCSP(Hybrid Domain FCSP), can be solved by a new algorithm SpreadRepair, an extension of the well-known heuristic repair algorithm.

1. はじめに

機械（計算機）を使用して問題解決を行う場合，それをどのように記述するかということをまず考えねばならない．より良い記述を与える優れた表現は，有効な解決手段を導くこととなる．制約充足問題 (Constraint Satisfaction Problem, 以下 CSP) という問題表現の方法は，現実世界における様々な問題を表現可能であり，これまで多くの研究が行われてきた．

CSP は，与えられた制約をすべて満たす変数の割り当てを求める問題である．しかしながら，制約を満たしている，いないの 2 値だけで表そうとするよりも，その制約の充足度を考慮し，出来るだけ制約を満たそうとする方がより現実的である．そこで CSP にファジィ関係を導入し，制約に曖昧さを持たせたファジィCSP [Ruttkay 1994, Meseguer 1997, Bistarelli 2002] がある．

一般に，現実の問題を CSP として定式化する上でいくつかの制限がある．CSP は通常，変数の領域は有限で離散的な集合であり，制約に曖昧さを設けたファジィCSP でもそれは例外ではない．しかしながら，一部の問題を定式化しようとしたとき，変数の領域が莫大に広域であったり，また，そもそも領域に含まれるかどうかが曖昧であったりするような場合，解を探索する上で不都合が生じる．例えば，Job-shop スケジューリング問題を CSP として定式化しようとした場合 [Prosser 1989]，時刻が変数領域に対応付けられるが，最適解を求めるには領域を細かく区切らねばならない．

本稿では，このような制限を緩和するために，ファジィCSP の変数領域に連続化した部分を持つことを許した混合領域ファジィCSP[Sudo 2002] と，反復改善法に基づくその解法を提案する．また，制約を記述するメンバーシップ関数の形状と，アルゴリズムについての関係を調べ，双方の性質から関数の効果的な線形近似法を述べる．

2. 制約充足問題

CSP は通常，有限で離散的な領域から値を選ぶ複数の変数に対して，全ての制約を満たすような値の割り当てを探しだす組み合わせ問題と定義され，そのときの変数への割り当てを解とする．クリスプな制約（完全に満たしているか，完全に違反しているか）C_k に含まれる変数の組は，C_k を満たすような D_i の直積集合から値を取る．

ここで制約とは，数学的には変数間の関係として表現される．例えば，$X_1 > X_2$ という制約は，$X_1 > X_2$ という関係を満たす対 (X_1, X_2) の集合，$\{(X_1, X_2)|X_1 > X_2, X_1 \in D_1, X_2 \in D_2\}$ である，それぞれの領域同士の直積集合から値を選ぶわけである．制約の表現の仕方はこのように述語的に表しても良いし，満たされる値の組を列挙しても良い．本稿で提唱する新しいモデルは，すべての組み合わせを列挙することは不可能であるので，以下制約を表すときは述語的に表現する．

関係（制約）は，２項関係であるとは限らず，$X_1 > 0$ 等という場合は単項関係，変数が３以上の場合にはとくに，多項関係，多項制約という．ただし，話を単純にするため，以下本稿ではとくに断りがない限り，制約は２項制約のことを指すものとする．

連絡先: 須藤康裕，北海道大学工学研究科，札幌市北区，sudoy@main.eng.hokudai.ac.jp

3. ファジィCSP の拡張

3.1 ファジィCSP

Fuzzy 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 $C_k(v)$ は，ファジィ関係 $R_k(v)$ に対応付けられ，そのメンバーシップ値 $\mu R_k(v)$ は

$$\mu R_k(v) : Di \times Dj \times \cdots \rightarrow [0, 1]$$

の形式で与えられる．すなわち，領域 $D_i, D_j, \ldots$ から取られた値の組み合わせ v に対するメンバーシップ値 $\mu R_k(v)$ は，$[0, 1]$ の実数値をとり，それが制約充足の度合いを示す．また，制約 C_k と，別の制約 C_j との２つの制約の間に新しいファジィ関係が生まれ，それを以下のように定義する．

$$\mu R_k \cap R_j(v) = min(\mu R_k(v), \mu R_j(v))$$

これは，ファジィ制約の論理積 $C_k \wedge C_j$ をメンバーシップ値の最小値として定義することを意味する．例えば“金持ちである”かつ“背が高い”というファジィ集合を考えたとき，前者が 0.8，後者が 0.5 であったとすると，“金持ちである”かつ“背が高い”程度は $min(0.8, 0.5) = 0.5$ となる．同様に，連

鎖的に全ての制約を考慮し，CSP 全体の充足の度合いは以下のように定義される．

$$\mu \cap R_{k=1\ldots r}(v) = min_k(\mu R_k(v))$$

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 の充足度としている [Meseguer 1997]. また，解は全ての制約について，少なくとも部分的に満たしている，完全な割り当てである．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 を解こうとすることは，

最も制約を満たしていない制約の充足度を最大化するように変数の値を割り当てることになり，以下の式がその目的関数となる．

$$max_v(min_k(\mu R_k(v))) \quad \cdots (1)$$

ただし，(1) 式についてのみ考慮した場合，あまりにも雑な解が得られてしまうので，最悪の制約違反以外の制約違反も出来るだけ改善すべきであり，それについていろいろ議論されている [Ruttkay 1994, Kanada 1595].

3.2 HDCSP の定義

領域を連続で表現することは，実に有用である．例えば，時刻を表現するとき,「明日，７時，７時１分，７時２分, (中略)，１２時は暇です」等と列挙する人はおそらく存在しない．ほとんどの人は「７時から１２時までは暇です」と言うであろう．そういう意味で，連続領域を問題の定義として使用できるのは，実に柔軟で自然である．本稿では変数領域に連続領域を導入した混合領域ファジィCSP（Hybrid Domain Fuzzy CSP, 以下 HDFCSP）を導入する．HDFCSP は基本的に，FCSP を拡張したモデルであり，領域の定義以外は FCSP と同じであるので，前節を参照していただきたい．ここでは領域の定義のみ行う．

通常，CSP の変数領域は，有限で離散的とされる．すなわち有限な集合であり，大抵の場合それは列挙することによって表現された.「100 以下の整数」などというようにも表現可能であるが，これにさらに「3 の倍数を除く」とか「ただし 5 の倍数は除かない」などという複雑な条件が加わってくると，結局列挙した方が都合の良い場合もあるので，ここでは集合を列挙するとして扱う．領域は例えば，{1, 2, 5, 7} というように表現されていた．HDFCSP では，変数の各領域を閉区間の和集合として，

$$D_i = \bigcup_{i=1}^{m} [l_j, u_j]$$

であるとし，$[l_j, u_j]$ は下限を l_j, 上限を u_j とする連続区間，すなわち $\{x | l_j \leq x \leq u_j\}$ という領域を表す．また，$l_j = u_j$ の場合，その区間は単一の実数値を取るので，離散的な領域と同じとなり，全ての j において，$l_j = u_j$ となるような場合，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 を包含していることになる．

4. HDFCSP の解法

4.1 既存の手法

制約充足問題の解法として大きく 2 つ，BackTrack や ForwardCheck 等の厳密解法と，Breakout 等の反復改善法がある．前者はある程度全探索になるし，後者は最適解を得られる保証がないという特徴がある．前章で定義した HDFCSP は，変数領域に連続領域が含まれているので，どちらの手法もそのままでは利用することが出来ない．そこでまず始めに考えられるのが領域の離散化である．これによって既存のアルゴリズムをほとんどそのまま利用することが出来る (図 1)．しかし冒頭で述べたように，領域が広大であった場合に離散化して領域を増やすことは探索空間を広げてしまうだけでなく，サンプリング周波数によってはより良い解を見逃してしまうかも知れず，結果的に HDFCSP を提案する意味がなくなる．

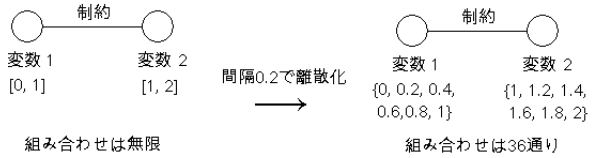

図 1: 領域の離散化

4.2 SpreadRepair アルゴリズム

HDFCSP を解くために，新しいアルゴリズムを提案する．基本的には局所的な改善を反復的に繰り返して最適解を探索するが，そこで必要なのは与えられた状態から近傍の状態への移行である．ここで，最悪の制約違反が，一度の変更によって改善出来ないような状況を準局所最適ということにする．また，すべての制約が一度の変更によって改善出来ないような状況を局所最適といい，その時の割り当てを局所最適解とする．

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関数と呼ばれる評価関数を使用する．式 (1) を単純なヒューリスティック関数として用いようとすると (すなわち最急降下法)，すくなくとも近傍の状態の中で最も充足度の高い割り当てに移行する必要がある．我々はこの採択を工夫し，非常に少ない候補に絞り込むことによって高速な収束が期待できるアルゴリズムを開発した．以降，そのアルゴリズムについて説明する．

ここで，必ずしも目的関数＝ヒューリスティック関数の関係が成り立つとは限らないことに注意して欲しい．本論文で提唱する Spread Repair アルゴリズムも，評価関数を式 (1) と少し変化させながら探索を進める．ただし，評価関数の形を何も考えずに変更し，目的関数とあまりにもかけ離れてしまうと解の質に影響を及ぼすであろうことは容易に想像できる．HDFCSP では最悪の制約違反を改善することが目的関数となっているが，Spread Repair アルゴリズムはもしそれができないような場合に評価関数を少し変更し，改善の対象をその周辺の変数に移す．この繰り返しにより，準局所最適の状態となっても，近傍を広げることによって準局所最適から抜け出す可能性を得，さらに目的関数値の局所最適性を保ったままそれ以外の制約違反もできるだけ改善することができる．”Spread Repair”は，改善の対象が周囲に広がっていく様子を意味している (図 2)．

このように局所改善していくと，いずれどの変数の値を変更してもどの制約も違反度が下がらない状態に陥る．この状態から抜け出すことについてはまた別の研究がされているので，ここでは言及しないことにする．

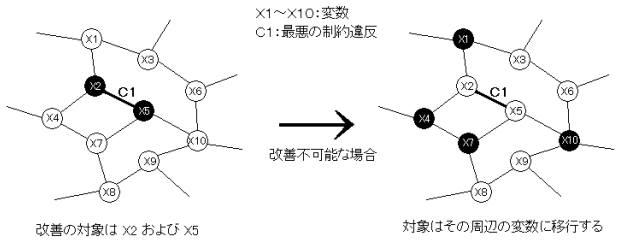

図 2: 改善の対象の移行

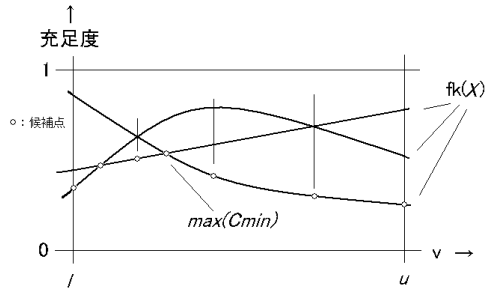

図 3: $max(C_{k_{min}})$ の候補点

4.3 変更候補値の算出および選択

[Jussien 2002] では，”candidate neighbor”という言葉を使用しているが，変更の対象になる変数のことを変更候補変数，またその変数の変更する予定の値を変更候補値として，以降本稿で使用する．またとくにことわりが無い限り，変更候補値は最も充足度が高くなるような変更候補変数への割り当てとする．

では，まず単純な 2 項制約を例にとって考えてみる．ある変数 X との制約が存在する変数が $X_1, \ldots, X_n$ まであったとする．また，v は X への割り当てであるとする．ここで，X と他の変数 X_k の間の制約が関数 $\mu R(X, X_k)$ によって表すことが出来るとすると，変数 X の割り当てを変更する場合，X_k は固定されるので定数とすることが出来，$\mu R(X, X_k)$ を X の関数 $C_k(X)$ と表すことにする．

ここで，全ての C_k について，ファジィ論理積を求めることが可能であれば，変数 X の割り当てを変更した場合の改善値が求まる．したがって，全ての変数について改善値を求め，その最大となるような変数 X を局所変更していくことによって最適解に近づくことが出来る．

ファジィ論理積は 3 章で述べたように $minimum$ であり，全ての制約の積によって生じるメンバーシップ関数を C_{min} とし，次のように定義する．

$$C_{min}(v) = \prod_{k=1}^{n} C_k(v) \qquad \cdots (2)$$

X の領域に含まれる 1 つの閉区間に対して 1 つの最大値 $(max(C_{min}))$ を求め，全ての閉区間に対する $max(C_{min})$ の中から改善値が最大のものを選べば，変数 X の新しい割り当てが求まる．ところが，C_{min} が任意のメンバーシップ関数についてはこのままでは最大値を見つけるのは困難である．しかし，C_{min} のうち必要なのは最大値 $(max(C_{min}))$ だけで，関数全体の値は必要ない．

筆者は $max(C_{min})$ の出現する可能性のある位置を考えてみた．1 つの閉区間について，その始点と終点は l と u で表されているが，1 つのメンバーシップ関数について l から u の間で，最大値と最小値は絶対に l か u もしくは極に存在することは，明らかである．同様に，全てのメンバーシップ関数のファジィ論理積による C_{min} は，それぞれのメンバーシップ関数がどのような形状をしていても，最大値は l か u，またはある 1 つのメンバーシップ関数の極，あるいはある 2 つのメンバーシップ関数の交点に存在し，それ以外の位置に出現する可能性は全くない（図 3）．

したがって，ここで問題となるのはメンバーシップ関数同士の交点を求める操作と，極を求める操作である．一般には数値計算法のアルゴリズムを利用して求めることになると思われるが，1 次や 2 次の単純な関数ならば代数的に求めることも可能である．また，多くの場合メンバーシップ関数をそのような単純な関数で近似，もしくはそれらの合成によって構成しているようであるので，極や交点を解析的あるいは代数的に求めることは現実的である．全ての区間について同じ操作を行っても良いが，実際には全ての区間について求めるというよりは，候補点を全て求めた後に区間外のものを除去することによって求めることが出来る．また，同程度の変更候補値が複数存在したり，他の変更候補変数の変更候補値を用いた改善量 r が同じ値であったりした場合，どれを選択し，変更するかが問題となる．本論文ではそれを「単に非決定的に選択する」と述べるのみにして，具体的には指定しない．この事項について実際には，順に試したりランダムで採択するような戦略が存在するが，本研究で実験的に構築したプログラムでは，最初に発見されたものを使用している．

5. HDFCSP における線形近似手法

一般に，メンバーシップ関数を線形近似しようとした場合，ある一定以上のゆるやかな曲線は 1 つの一次式によって表され，しかもその傾きが 0 に近くなれば 0 として使用されることが多い．これはすなわち，メンバーシップ関数の戻り値がある値 q(多くの場合 0 もしくは 1) にある程度近づいたときに，そこから先は q として扱うことを意味している．図 4 (a) は有名な sigmoid 関数の単純な例であるが，大抵の場合処理を単純化するために図 4 (b) のように近似される．つまりファジィとはいえ，上限と下限ではクリスプな関係になっているということである．

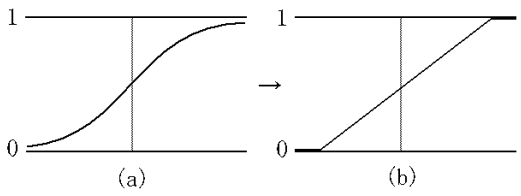

図 4: 線形近似の例

ところが実は，HDFCSP のモデルにおいて，この水平な部

分が左右どちらへ行っても充足度が加減しない，いわゆる”高原”の発生に関与していることが，実験によって明らかになった．図５はその一例である．目的関数および評価関数にC_{min}を使用する場合特に，この現象が起こる可能性が高く，もうすこし柔軟な評価関数を使用した場合でも全く起こらないとは言い切れない．そこで，我々は新しいメンバーシップ関数の設定により，この現象を回避する方法を提案する．

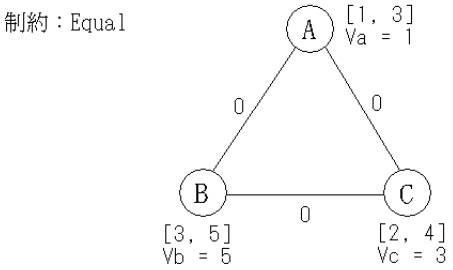

図 5: ”高原”の例 (何をやっても改善しない)

我々は線形近似されたメンバーシップ関数の水平部分に，出力された解に実際にはほとんど影響ない程度の傾斜を設ける手法を提案する (図６)．見た目がほぼ水平な直線であっても，計算機はその精度の範囲において充足度の大小を計算する．これが，実際には”高原”脱出の手掛かりとなってくれる (図７)．メンバーシップ関数の形状をほんの少しだけ変えてやるだけで劇的な効果が得られるこの手法は，是非導入すべきである．

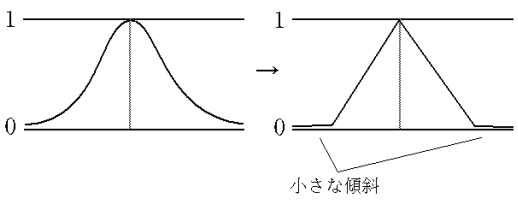

図 6: 新しい線形近似手法

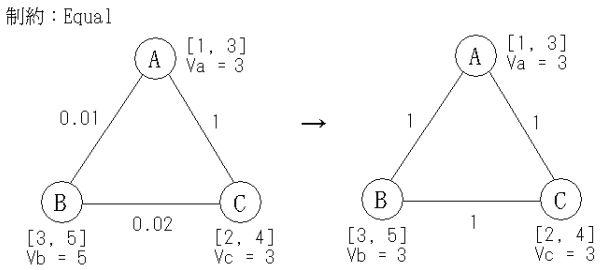

図 7: 高原の脱出

6. おわりに

本稿ではファジィCSP の変数領域を連続化した HDFCSP と，その解法について述べ，さらにそのモデルについての効率的なメンバーシップ関数の線形近似の手法について提案した．実際，このアルゴリズムを使用して，”曖昧なスケジューリング問題”の解を得ることが出来ている．今後，最適解との精度の差や，アルゴリズムの性質に関して，様々な実験を行う予定である．また，メンバーシップ関数の線形近似手法についても，その他の情報処理にもこの手法が応用可能かどうか，現在調査中である．

参考文献

[Ruttkay 1994] Ruttkay, Zs.: Fuzzy constraint satisfaction, Proceedngs of 3rd IEEE Intern. Conf. on Fuzzy Systems, 1263- 1268(1994).

[Meseguer 1997] Meseguer, P. and Larrosa, J.: Solving fuzzy constraint satisfaction problems, Proceedings of 6th IEEE Intern. Conf. on Fuzzy Systems, 1233-1238(1997).

[Bistarelli 2002] Bistarelli, S., Codognet, P. and Rossi, F.: Abstracting soft constraints: frameworks, properties, examples, Artificial Intelligence 139, 175-211(2002).

[Prosser 1989] Prosser, P.:A Reactive Scheduling Agent, Proc.IJCAI-89, 1004-1009(1989).

[Sudo 2002] 須藤康裕:ファジィ制約充足問題への連続領域の導入, FIT2002 情報科学技術フォーラム一般講演論文集第一分冊, 47-48(2002).

[Kanada 1995] Kanada, Y.:Fuzzy Constraint satisfaction using CCM-a local information based computation model, Proceedings of 4th Intern. Conf. on Fuzzy Systems, 2319-2326(1995).

[Jussien 2002] Jussien, N., Lhomme, O.:Local search with constraint propagation and conflict-based heuristics, Artificial Intelligence 139, 21-45(2002).