社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－MPS－46 (2)
2003／9／18

連続領域におけるファジィ制約充足問題の反復改善アルゴリズムによる解法

須藤 康裕[*1] 栗原 正仁[*1]

FCSP(ファジィ制約充足問題)はCSPを拡張した新しいモデルの1つであり，変数間に設けられた制約の充足に度合いが存在する最適化問題の1つである．CSPおよびFCSPでは通常，変数は有限で離散的な集合から値をとり，より現実的な問題を定式化する場合にそれが障害になる場合がある．

本稿では，離散値と連続値の両方を変数領域として持つ，混合領域FCSP(Hybrid Domain FCSP)を提案し，反復改善法を用いたSpreadRepairアルゴリズムによる解法を述べる．また，HDFCSPによって曖昧なスケジューリング問題をモデル化し，簡単な例を用いて実際に解が得られることを示す．

Solving by heuristic repair algorithm of the Fuzzy Constraint Satisfaction Problem with Continuous Domains

YASUHIRO SUDO[*1] and MASAHITO KURIHARA[*1]

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. Finally, we show modeling of Fuzzy scheduling problem by HDFCSP, and it is shown that a solution is actually obtained using easy examples.

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がまず問題となり，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，より厳密な最適解を求めるためには領域を細かく区切り，分単位，秒単位での莫大な組み合わせを考慮しなければならない．

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，Job-shopスケジューリング問題をHDFCSPとしてモデル化し，実際に解が得られることを示す．

[*1]北海道大学 Hokkaido University
sudoy@main.eng.hokudai.ac.jp

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，$\{(X_1, X_2) \mid X_1>X_2,\ X_1\in D_1,\ X_2\in D_2\}$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の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．以下，話を単純にするため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．ただし，本稿で提案する手法自体は多項制約においても解を探索することが可能である．

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

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

の形式で与えられる．すなわち，領域D_i, D_j…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．

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

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．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．

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

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．

$$\max_v(\min_k(\mu R_k(v))) \qquad \cdots (1\text{-}1)$$

ただし，(1-1)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 HDFCSPの定義

領域を連続で表現することは，実に有用である．また，連続領域を複数持つような場合も当然考えられ，それらの考えを全て取り入れた変数領域が最も自然な表現方法であると言える．そこで，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．ただし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$\{x \mid l_j \le x \le u_j\}$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に基づく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（近似解法）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．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．

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(1-1)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(1-1)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．すなわち，このときの評価関数は

$$\max_r(\max_k(1-\mu R_{k\,neighbor}(v))) \quad \cdots(1\text{-}2)$$

となる．ここで$k_{neighbor}$は最悪の制約違反kに隣接する制約であり，rは

$$r = \mu R_k(v_{future}) - \mu R_k(v_{now})$$

と定義され，rが最も大きくなるように変更する変数を選択する．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1）．

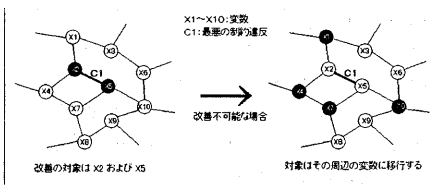

図1　改善の対象の移行

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．このアルゴリズムによって得られる最適解はここまでで，最急降下的な手法であるために局所解である場合もあり，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．

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$X_1, \cdots, X_n$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．

ここで$max(C_{min})$の出現する可能性のある位置を考えてみる．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2）．

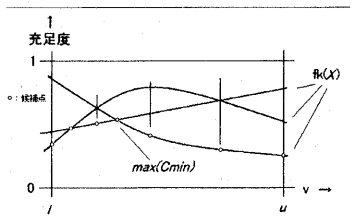

図2　$max(C_{min})$の候補点

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が同じ値であるような場合，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．Job-shopスケジューリング問題の定式化

5.1 CSPとしてのモデル化

Job-shopスケジューリング問題をCSPとして定式化する試みがなされている[Prosser 1989]．Job-shopスケジューリング問題とは，複数のJobが同一の資源を使用する場合にコンフリクトしないよう各Jobの実行時刻を決定する問題である．CSPとして定式化する場合，Jobが変数，Jobの実行可能時刻が変数領域に対応付けられる．また，同一の資源を使用する可能性があるJob間に制約が設けられる．ここで，各Jobの実行時間が単位時間であったとすると，制約は次のように表される．

$$\{(X_i, X_j) \mid d \lhd |X_i - X_j|, X_i \in D_i, X_j \in D_j\}$$

通常のCSPの場合，変数領域は有限離散集合であるので，問題は図3(a)のように表現される．また，CSPは単純な無向グラフとして表現可能であるので，その例が(b)である．この例では，全てのJobが同一の資源を共有するとしているが，予め交錯する可能性の無いJob間の制約は取り払っている．

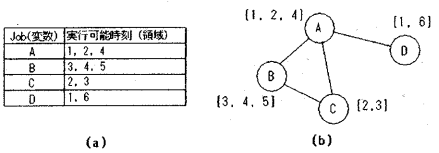

図3　スケジューリング問題の例

5.2 HDFCSPとしてのモデル化

同様に，HDFCSPとしてのモデル化を考えて見る．先ず領域は連続値をとることが出来，より細かい時刻を指定することが出来る．また，ファジィ制約になるので，”なるべく交錯しない”という曖昧な制約になる．これにより，実際には必ずどこかで制約違反するようなスケジューリングでも，なるべく制約違反を少なくするような問題として解を求めることが出来る(図4)．この場合，メンバーシップ関数は有名なLorentz-peak関数を逆さにしたものを線形近似して使用することにした．このようにシステムではいくつか基本的な関数を用意してあり，問題によってパラメータを変更して使用することが可能である．また，ユーザが関数を構築することも可能である．

Job	実行可能時刻	資源
A	[0.6,1],[2.4,2.8]	①，③
B	[1,1.2],[2.2.5]	①
C	[1,2]	①，②
D	[0.0.5],2	②，③
E	[1,2],[4,5.5]	②
F	[0,1],[2,2.3]	②

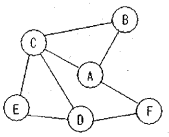

図4　曖昧なスケジューリング問題

この例題は図3の例をやや拡張し，複数の資源を使用する場合がある．この例題を実際にSpreadRepairアルゴリズムを用いて最適化を行うと，充足度0.8の割り当てを得ることが出来る．この例では0.8が最適解であり，単峰性の問題であれば必ず最適解に収束する．

このように，本来であれば解が存在しない問題も，柔軟な制約を導入し，領域を自由に表現することによって最善な解を探索することが可能になる．

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．

参考文献

[Ruttkay 1994] Ruttkay, Zs.: Fuzzy constraint satisfaction, Proceednge 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 COM-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).