GUI構築のための数値的な制約階層解消系の機能的改良

1ZA－8

細部 博史　(文部省 学術情報センター 研究開発部)

1　はじめに

GUIの構築で，制約の**優先度**を実現する**制約階層**を容易に利用可能にするため，我々はこれまでに，線形計算を用いたインクリメンタルな制約解消系 HiRise [5] を提案している．本研究では，HiRiseの機能的な改良を目的として，(1) 制約の連言 (AND) を可能にする手法，および (2) 優先度が内部的に全順序であることに起因する制限を緩和する手法を提案する．

2　HiRise制約解消系の概要

HiRise制約解消系 [5] が扱う制約には，1次等式，stay (変数の値を一定にする)，edit (変数の値を繰り返し更新する) の3種類がある．そして，各制約には，有限個 (デフォルトで16) の段階からなる，**強さ**と呼ばれる優先度が与えられる．

HiRiseは，内部的に，制約階層を**階層線形系** [4] に変換して処理を行う．制約階層では，優先度は有限段階であるが，階層線形系では，優先度は個々の制約ごとに異なり，全順序になっている．

HiRiseの主要データ構造は階層線形系のLU分解である．これは，**階層独立**である1次等式制約 (より優先度の高い制約に対して係数ベクトルが線形独立である制約) の係数ベクトルを集めて，修正しやすい形にLU分解したものである．以下では，LU分解の対象となる制約を**有効**な制約，そうでないものを**無効**な制約と呼ぶ．

3　提案する機能的改良法

本節では，HiRise制約解消系のための2種類の機能的改良法を提案する．

3.1　制約の連言の導入

制約を用いて2次元のGUIアプリケーションを作成する際，x 座標と y 座標に関する2個の制約が組になっていることが多い．このような状況で，それらを連言 (AND) で結合し，同時に満たすことが望ましい場合がある．しかし，従来のHiRiseでは，制約の連言[1]の機能を提供していなかったため，アプリケーションによっては好ましくない振舞いが得られる可能性があった．

一般には，連言を導入すると，組合せ問題的な性質が入り込むため，インクリメンタルな制約解消が難しくなる．しかし，HiRiseは2次元GUIをその主な応用対象としているため，2つの制約の連言をサポートしていれば，実用上十分であることが多い．本研究では，このように問題の形を制限することで，バックトラックなどを導入せずに，制約の連言を実現することを試みた．

3.1.1　手法の概略

特に問題になったのは，いったん解かれた階層線形系に新しい制約を追加したり，既存の制約を削除したりするという，インクリメンタルな制約解消の部分である．本研究では，この部分について，新たに有効化された制約のために，それまで有効だった，より弱い制約の連言を無効化する必要がある場合，その直後に，さらに弱い無効な制約の中から，最終的に有効になる可能性がある制約を選び出すという処理を挿入することで対処した．

3.1.2　アルゴリズム

制約の連言が存在する状況で，制約が追加，削除された場合の処理の流れは以下のようになる．

1. 削除された制約を内部データにおいて無効化する．
2. 追加された制約を強いものから1個ずつ必要なら有効化する．ただし，連言によって組になっている制約については，両方とも充足できる場合に限る．また，制約の有効化のために，より弱い矛盾する制約を無効化しなければならない場合，その無効化すべき制約が連言によって組になっていれば，もう一方の制約も無効化する．この場合は，さらに弱い無効な制約の中から充足できるものを選び出し，有効化する (この制約が連言によって組にされている場合，この処理はさらに連鎖される)．
3. 自由な変数が残っていれば，その値を一定に保つ，仮想的な弱い制約を有効化する．

3.2　優先度に関する制限の緩和

従来のHiRise制約解消系では，同じ強さの制約が矛盾し合うときに，好ましくない解き方をすることがあった．これは，その基盤である階層線形系において，制約の優先度が全順序になっているために，外部的に (制約階層で) は強さが同じであっても，内部的に (階層線形系で) は優先度が異なるために生じる制限である．これは，実際のアプリケーションにおいて，水平または垂直に近い直線上に制約されたオブジェクトのドラッグが困難になるといった問題を引き起こすことがある．

このような問題を解消するために，本来の制約階層理論 [1] では，例えば，同じ強さの制約の間で最小2乗法を行うような解の決め方を定めていた．しかしながら，

Functional Enhancement of a Numerical Constraint Hierarchy Solver for Constructing Graphical User Interfaces

Hiroshi Hosobe

R&D Department, National Center for Science Information Systems

[1]制約階層の理論上は，2つの1次等式の連言が，制約階層における1つの制約に対応する．

現実的には，効率を低下せずに，HiRiseに最小2乗法のような大域的処理を導入することは困難である．

しかし，HiRiseが主に2次元GUIを対象とすることを考えると，組になっている2つの制約だけを実質的に同じ強さで指定できれば，実用上十分なことが多い．本研究では，このような観点から，優先度に関する制限の緩和を効率的に行うことを試みた．

3.2.1 手法の概略

本研究では，実質的に同じ強さとして組にされた2つの制約について，動的にそれらの間の優先度を適切に決定することで，この問題に効率的に対処した．この手法では，最小2乗法のような処理を用いないが，2つの制約の誤差の2乗和は，最小2乗法の場合に比べて，高々2倍程度で抑えられる．このため，実用上，ユーザーが不便を感じることはほとんどなくなると言ってよい．

3.2.2 アルゴリズム

この手法を適用すべき状況は，組になっている制約の一方が有効であり，他方が無効である場合である．以下では，有効な制約をc^+，無効な制約をc^-と呼ぶ．

1. 仮想的に，c^+の定数項を1，それ以外の有効な制約の定数項を0とし，LU分解を用いて，各変数の値を計算する．
2. c^-の1次の項の各変数に，上で求めた値を代入し，それらの和δを得る．
3. δについて調べる．
   (a) $-1 \leq \delta \leq 1$の場合，そのままでよい[2]．
   (b) $\delta < -1$または$1 < \delta$の場合，c^+を無効化して，その代わりにc^-を有効化する．

この処理は，c^+とc^-に関係する制約の集合に変化があるごとに行う必要がある．しかし，有効な制約と無効な制約を切り換える必要がない限り，この処理は高速に行うことができるため，効率上の問題は小さい．

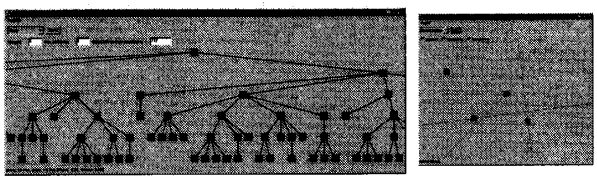
4 実装

改良を施したHiRise制約解消系と，そのサンプルアプリケーションをJavaで実装した．アプリケーションの1つは木を編集するもので，オブジェクトのドラッグのためのx座標とy座標に関する2つの制約を実質的に同じ強さで指定するようにしている(図1(a))．このほかにも，本研究で提案した2つの手法を用いた場合に，オブジェクトの振舞いがどのように変化するかを実演するアプリケーションを作成した(図1(b))．

5 関連研究および議論

局所伝播法による制約解消系SkyBlue [7]で提供されている多出力のメソッドは，制約の連言として代用できる．この方式には，連言で組にされた2つの制約についてデータフローの方向をそろえることができ，便利な一面がある．ただし，HiRiseでも，適切に制約の強さを指定すれば，データフローの方向を制御できる．

[2] $\delta = 0$ならば，c^+とは無関係に，c^-は充足できない．



(a) (b)

図1: (a) 木を編集するアプリケーションと (b) 提案した機能を実演するアプリケーション

本研究で扱った制約の優先度に関する制限は，従来のHiRiseに限らず，DeltaBlue [3]やSkyBlueなどの局所比較子 [1] に基づく制約解消系においても生じる．これを解決するには，一般的には，*DETAIL* [6]やQOCA [2]のように最小2乗法などを行う大域比較子 [1] を用いる必要がある．このような一般的な方式は，表現力の面で有利であるが，本研究で提案する手法で扱える範囲であれば，HiRiseの方が効率的である．

6 おわりに

本研究では，HiRise制約解消系の機能的改良として，制約の連言を実現する手法と，制約を実質的に同じ強さとして指定できるようにする手法を提案した．

これにより，HiRiseの機能的拡張という観点では，現時点で可能な機能をほぼ一通り実現したと考えている．今後は，HiRiseの改良だけでなく，異なったアプローチを模索していくことも検討している．

参考文献

[1] Borning, A., Freeman-Benson, B., and Wilson, M.: Constraint Hierarchies, *Lisp and Symbolic Computation*, Vol. 5, No. 3 (1992), pp. 223–270.

[2] Borning, A., Marriott, K., Stuckey, P., and Xiao, Y.: Solving Linear Arithmetic Constraints for User Interface Applications, *ACM UIST*, 1997, pp. 87–96.

[3] Freeman-Benson, B. N., Maloney, J., and Borning, A.: An Incremental Constraint Solver, *Comm. ACM*, Vol. 33, No. 1 (1990), pp. 54–63.

[4] 細部博史, 松岡 聡, 米澤明憲: 階層線形系を用いた効率的な制約階層解消法, インタラクティブシステムとソフトウェアV (WISS'97), 近代科学社, 1997年, pp. 129–134.

[5] 細部博史, 松岡聡, 米澤明憲: HiRise: GUI構築のためのインクリメンタルな制約解消系, コンピュータソフトウェア, 日本ソフトウェア科学会 (採録決定).

[6] Hosobe, H., Miyashita, K., Takahashi, S., Matsuoka, S., and Yonezawa, A.: Locally Simultaneous Constraint Satisfaction, *Principles and Practice of Constraint Programming—PPCP'94*, LNCS, Vol. 874, Springer-Verlag, 1994, pp. 51–62.

[7] Sannella, M.: SkyBlue: A Multi-Way Local Propagation Constraint Solver for User Interface Construction, *ACM UIST*, 1994, pp. 137–146.