並列制約論理プログラミングシステム GDCC

概要

並列制約論理プログラミング言語 GDCC は問題解決のための高級言語であり、論理プログラミングのパラダイムと制約プログラミングのパラダイムを並列環境下で融合させた、高水準で、柔軟かつ効率的な言語である。本デモンストレーションにおいては、GDCC の様々な問題領域における有効性を示すため、いくつかの試作システムを紹介する。

デモンストレーション

1) GDCC

GDCC の基本的な特徴を、簡単な例を用いて示す。

2) ハンドリング・ロボット設計支援システム

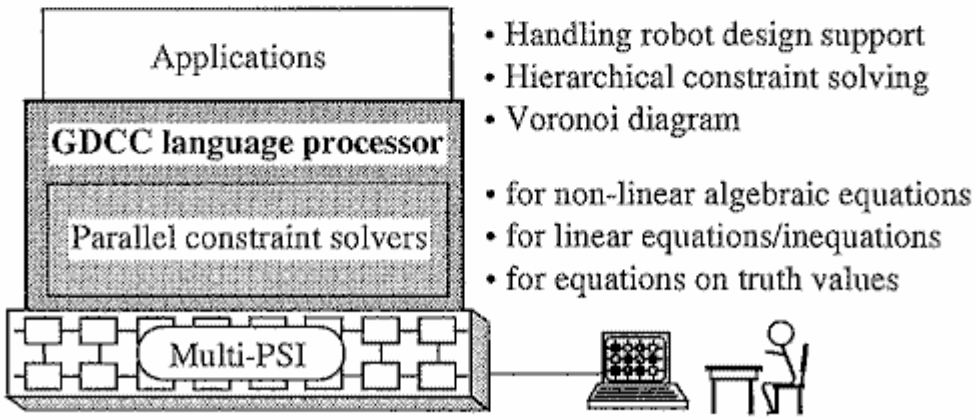
GDCC の強力さと柔軟性を設計の領域を例題として用いて示す。

3) GDCC による階層制約の並列評価

階層制約の並列評価系を GDCC を用いて記述することにより、GDCC の強力さと効率の良さを示す。

4) ボロノイ図

GDCC の高水準さと柔軟性を計算幾何学の領域を例題として用いて示す。



GDCC システム

1. GDCC − 言語と処理系

本システムの目的

1) 制約論理プログラミングのパラダイムに基づいた高水準で宣言的な言語を提供する。
2) 問題解決のための強力で柔軟な枠組を提供する。
3) 並列性を利用した効率の良い問題解決のための枠組を提供する。

制約論理プログラミングによる問題解決

制約とは、問題中で成立する関係である。制約論理プログラミングを用いた問題解決は、通常のプログラミング言語を用いた問題解決とは、次に示すように異なっている。

- 通常のプログラミング言語を用いた問題解決
  1. 問題の解析
  2. 問題を構成する対象間の関係の発見
  3. 解法の発見
  4. 問題を解く手続きとしてプログラムを作成
- 制約論理プログラミングを用いた問題解決
  1. 問題の解析
  2. 問題を構成する対象間の関係の発見
  3. 問題中で成立する関係（制約）の集まりとしてプログラムを作成

どのようにして解くか ⇒ 何を解くか

GDCC の特徴

1) 2つのレベルの並列性
   柔軟で効率の良い制約論理プログラミング言語を実現するため、GDCC においては言語の並列化と制約評価系の並列化という、2つのレベルにおける並列化を利用している。
2) 複数の制約評価系
   1. 代数制約評価系
   2. 有理数／整数線形制約評価系
   3. 真偽値制約評価系
3) ブロック機構

1. 多重環境

   GDCC の代数制約評価系における 1 変数高次制約から実根の近似値を求める機能により、ある変数が複数の値を取り得る状況を扱う必要がある。

2. 失敗の局所化

   コミッテッド・チョイスに基づく並列言語において探索機能を持たせるため、制約評価の矛盾などに起因する失敗を局所化する必要がある。

3. 推論エンジンと制約評価系との同期の問題

   ある制約集合に対して、ある関数の最大値、最小値などを求める際に、その制約集合を与え、その元であるゴールの評価を行わせるような機構が必要となる。たとえば、

   $$\left.\begin{array}{l} 0 \leq X,\ X \leq 1 \\ 0 \leq Y,\ Y \leq 1 \end{array}\right\}$$ という制約下で $max(X+Y)$ を求める

GDCC のプログラム例

ヘロンの公式の導出

以下の GDCC は、三角形に関する 3 つの既知の性質、すなわち三角形の求積公式、直角三角形に関するピタゴラスの定理、そして任意の三角形が 2 つの直角三角形に分割可能であるということから、三角形に関するヘロンの公式として知られる新しい性質を導き出すことが出来る。

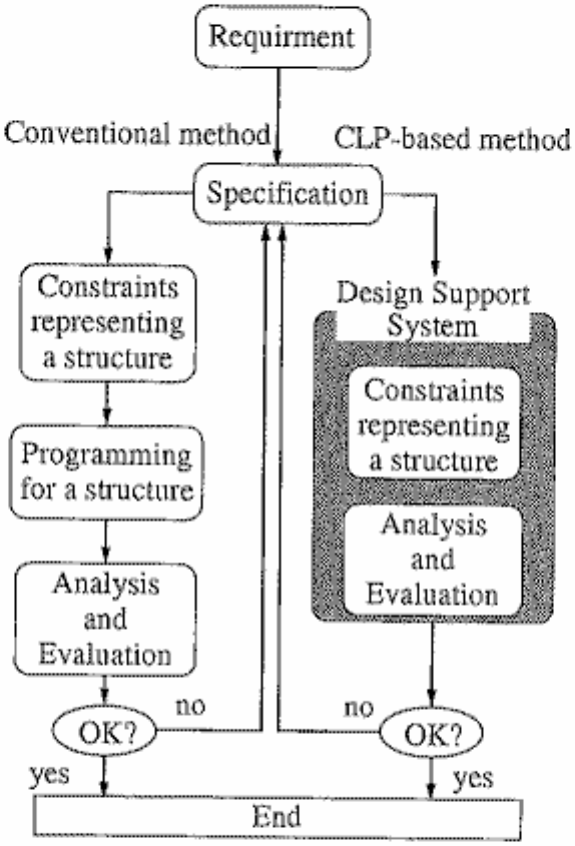
```
:- module heron.
:- public tri/4.

tri(A,B,C,S) :- true |
      alloc(0,CA,CB,H),
        alg#C=CA+CB,
        alg#CA**2+H**2=A**2,
        alg#CB**2+H**2=B**2,
        alg#H*C=S.
```

2. ハンドリング・ロボット設計支援

特徴

本支援システムを用いてGDCCの強力さと柔軟性とを示す。

ハンドリングロボットの設計過程



1) システムへの入力である、ハンドリング・ロボットの仕様を変更するだけで任意構造のロボットの処理が可能である。

2) 個々のハンドリング・ロボットに対する解析プログラムを別々に作成する必要がない。

システムの機能

本支援システムは、以下の機能を持つ。

- 制約の生成
    - 順機構学を解くことによる
- ロボットの解析と評価

- 逆機構学を解くことと
- 各関節のトルク計算と、
- 設計中のロボットの可操作度の計算による

デモンストレーション

本デモンストレーションにおいては、設計対象のハンドリング・ロボットを、関節数3、アーム数3のものとする。

1) 順機構学

手先の位置 (P_x, P_y, P_z) を、各アーム長 (l_1, l_2, l_3) と各関節の回転角度 $(\theta_1, \theta_2, \theta_3)$ を用いて表す。

$$\text{仕様} \Rightarrow \begin{cases} P_x = P_x(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \\ P_y = P_y(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \\ P_z = P_z(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \end{cases}$$

2) 逆機構学

与えられた位置 (P_x, P_y, P_z) に手先を移動するために必要な各関節の回転角度 $(\theta_1, \theta_2, \theta_3)$ を求め、ついである関節の回転角度に制限を加えて解の個数が減少する様子を示す。

$$\text{仕様} \Rightarrow \begin{cases} \theta_1 = \theta_1(P_x, P_y, P_z, l_1, l_2, l_3) \\ \theta_2 = \theta_2(P_x, P_y, P_z, l_1, l_2, l_3) \\ \theta_3 = \theta_3(P_x, P_y, P_z, l_1, l_2, l_3) \end{cases}$$

3) トルクの計算と可操作度の評価

順機構学の結果 (P_x, P_y, P_z) と、手先に作用する力 (F_x, F_y, F_z) とから、各関節にかかるトルク (T_1, T_2, T_3) と可操作度 (D) を求める。ついで手先の動作範囲に制限を加え、トルクと可操作度の式を簡単化する。

$$\left.\begin{array}{l} P_x = P_x(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \\ P_y = P_y(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \\ P_z = P_z(\theta_1, \theta_2, \theta_3, l_1, l_2, l_3) \\ F_x,\ F_y,\ F_z \end{array}\right\} \Rightarrow \begin{cases} T_1 = T_1(\theta_1, \theta_2, \theta_3, F_x, F_y, F_z) \\ T_2 = T_2(\theta_1, \theta_2, \theta_3, F_x, F_y, F_z) \\ T_3 = T_3(\theta_1, \theta_2, \theta_3, F_x, F_y, F_z) \\ D = D(\theta_1, \theta_2, \theta_3) \end{cases}$$

3. 階層制約の並列評価

制約階層

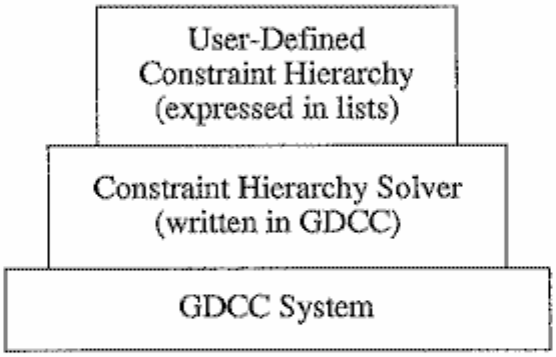
1. 制約に様々な強さを導入
2. 生成型の問題において重要な働きを持つ

特徴

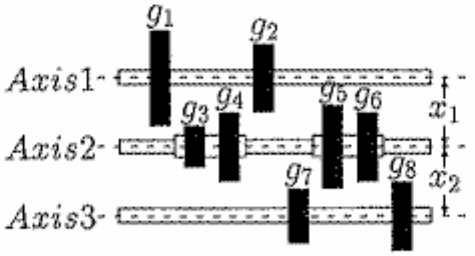
階層制約を、GDCC のブロック機構を用いて並列に評価する。

1. まずブロック機構を用いて、どのようにして階層制約を評価するかを示し、
2. 次いでマルチ PSI 上での並列実行による効率向上を示す。

システム構成



デモンストレーション: 変速機の設計



4 速 3 軸の変速機

まず 2 つの噛み合う歯車の組み合わせによって決定される 3 軸間の間隔の和を与え、変速比を求める。ここで、それぞれの歯車の標準部品は、離散値をとる標準半径を持つものとする。そして、各歯車の半径を、出来るだけ標準部品を利用できるように決定する。

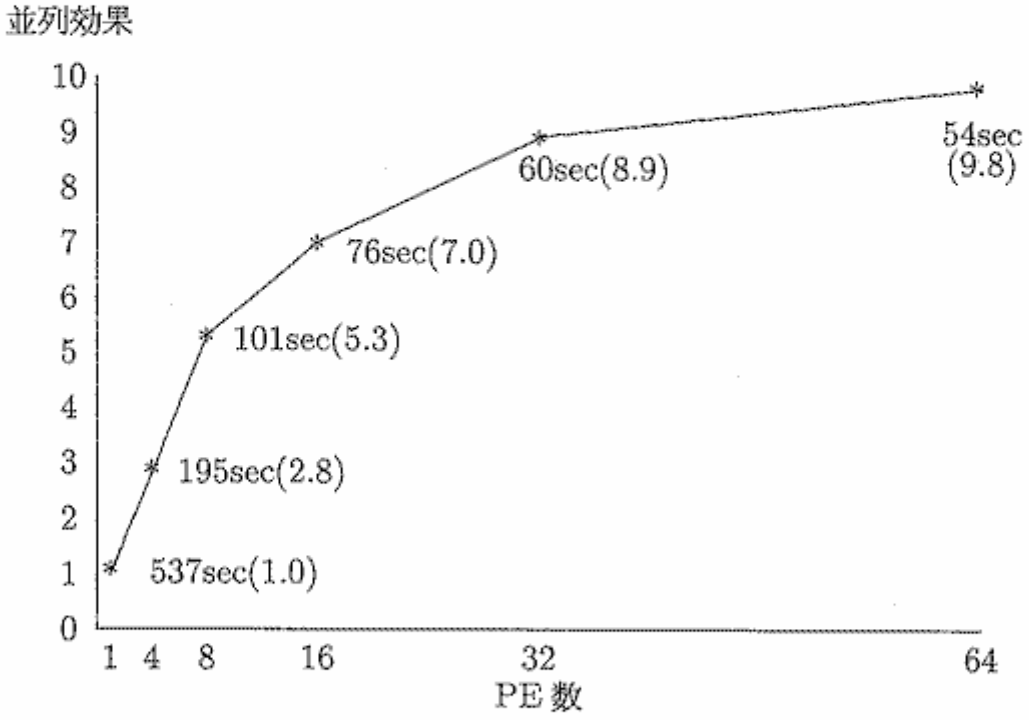
設計過程の例

固い制約 (変速比と軸間距離):

$$ratio(\langle g_1, g_3\rangle, \langle g_5, g_7\rangle) = 1 \quad \text{(a)}$$
$$ratio(\langle g_2, g_4\rangle, \langle g_5, g_7\rangle) = 2 \quad \text{(b)}$$
$$ratio(\langle g_1, g_3\rangle, \langle g_6, g_8\rangle) = 4 \quad \text{(c)}$$
$$ratio(\langle g_2, g_4\rangle, \langle g_6, g_8\rangle) = 8 \quad \text{(d)}$$
$$x_1 + x_2 = 10 \quad \text{(e)}$$

柔らかい制約 (標準部品の利用):
標準部品の歯車の半径を 1、2、3、4 とする。

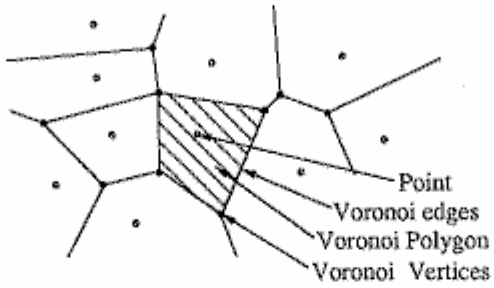
並列効果



4. ボロノイ図

ボロノイ図

平面上の点の有限集合 S のボロノイ図とは、それぞれ S の点をひとつづつ含むようなその平面の分割で、各領域は他のどの S の点よりもその領域内に含まれる S の点の方が近いように決定される。



特徴

このボロノイ図のプログラムは、制約論理プログラミングの高水準さと、柔軟性を示すものである。

1) ボロノイ図の定義を比較的そのまま記述したものである。
2) 計算の複雑さは、点の数を N としたとき、平均 $O(N)$ である。
3) 距離の定義の変更が容易である。

デモンストレーション

1) 与えられた点についてボロノイ図を構築する。
2) ボロノイ図の距離の定義を変更する。

結果