

区間制約ソルバにおけるパラメータ化制約の導入

野村 亮太[†]
福井大学[†]

石井 大輔[‡]
北陸先端科学技術大学院大学[‡]

1 はじめに

区間制約ソルバ (2 節) は, 実数領域の数値制約充足問題を区間計算に基づいて求解するソフトウェアであり, ロボットアームや動的システム等の応用事例をモデル化し, 解析するのに役立つ.

動的システムを制約充足問題として記述する際に, 複数ステップで同様の記述を強いられたり, 記述順が悪いため求解時間が大きくなる問題がある.

本研究では, 上記問題を解決するため, 制約を効率良く記述し, 求解する手法を開発する. 同形の一連の制約を表すパラメータ化制約 (3 節) をサポートした区間制約ソルバを実現する. また, 削減率に基づく制約選択機構 (4 節) を実装する.

2 区間制約ソルバ

区間制約ソルバは, 数値制約充足問題を区間計算に基づいて求解するソフトウェアである.

制約充足問題は, 変数の集合, 矩形 (区間ベクトル), 変数間の制約の集合で定義され, 実数のある領域内から制約を満たす解を求める問題である. 制約充足問題の解は, 制約を満たす矩形内から取り出した実数ベクトルである.

区間制約ソルバでは, 探索アルゴリズムとして branch-and-prune [1] が使われている. branch-and-prune では制約充足問題と精度を入力とし, 解を近似する矩形集合を出力とする. 求解時には, 制約を満たさない矩形領域を削減し, 矩形が精度以

上であるとき, 分割するという処理を繰り返す. 出力として得られる矩形集合内の各矩形は, 解を包含しており, 矩形中の区間幅が精度より小さい矩形である.

branch-and-prune の実装として ICPy^{*1}がある.

3 パラメータ化制約

制約充足問題の例として図 1 がある. 2 行目では変数 `input[0]~input[9]`, 3 行目では変数 `fb[0]~fb[9]` を宣言している. 2 目と 3 行目は区間 $[-1, 1]$ と区間 $[-1e+3, 1e+3]$ を持つ要素数 20 個の矩形を宣言している. 5 行目~9 行目で, 9 個の等式と 1 個の不等式で制約を宣言している. 図 1 の制約充足問題は,

$$fb_0 = input_0$$

$$fb_n = input_n + 0.9 * input_{n-1} \quad (n = 1, \dots)$$

で表される動的システムをモデル化している. このときの制約充足問題の解は, 10 ステップ目で値が 6 以上となるときの振る舞いを表している.

動的システムを制約充足問題として記述する際, 複数ステップについて同様の記述を強いられる場合があり, ステップ数が多くなるほど, 記述時にかかる時間や記述誤りのリスクが大きくなる. そこで, 複数ステップについて同様の記述となる制約を一纏めの制約で表すためにパラメータ化制約を導入する. パラメータ化制約とは, 同形の一連の制約を表した, パラメータ変数を持つ制約である.

図 1 の制約充足問題は, 本研究で導入したパラメータ化制約で表しており, 6 行目~8 行目で, 複数の制約を一纏めに記述している.

Introducing parameterized constraints in an interval constraint solver

[†] Ryota Nomura, University of Fukui

[‡] Daisuke Ishii, Japan Advanced Institute of Science and Technology

^{*1} <https://github.com/dsksh/ICPy>

```

1 Variables
2   input[10] in [-1, 1];
3   fb[10] in [-1e+3, 1e+3];
4 Constraints
5   fb[0] = input[0];
6   for i in range(8):
7     fb[i+1]=input[i+1]+0.9*fb[i];
8   rof;
9   fb[9] >= 6;
10 end

```

図1 制約充足問題の例

4 削減率に基づく制約選択機構

ICPy で求解する際、制約の記述順が悪いと求解時間が大きくなる場合がある。branch-and-prune 内で用いられる削減アルゴリズム [2] では、制約を1つずつ選択し、制約を満たさない矩形中の変数区間を削減する。ある制約で大きく削減できた場合は、共通変数を持つ別の制約で削減を実施すれば、前記の削減を別の削減へと波及させることができる (制約伝播)。しかし、動的システムモデル等では、制約が多くなるため、効果的な制約伝播を実施するのが難しくなる。

効率的に制約伝播を行うため、削減率に基づく制約選択機構を提案する。この制約選択機構では、まず、1つの制約を用いて矩形の削減を行う。その削減により、削減前と削減後の矩形を比較し、削減率が最大となるような変数を求める。この変数が含まれている制約を次の削減に用いる制約として選択する。この制約選択機構により、制約の記述順によらず、効率的な求解ができると考える。

5 実装と実験

区間制約ソルバでパラメータ化制約を扱うために、区間制約ソルバ ICPy のパーサと削減器の拡張を行った。まず、パーサ生成の文法を書き換えることで、パラメータ化制約、if 文、for 文、range を読み込むことができるようにした。パーサで読み込んだ後に中間表現を生成し、解釈処理を実行している。次に、パラメータ化制約とパラメータの値を持

つデータ構造を用意した。このデータ構造を用いることでパラメータ化していない同形の一連の制約と同様に扱うことができる。

削減率に基づく制約選択機構を ICPy に実装するために優先度付きキューを用意した。優先度付きキュー内に同一優先度で制約を格納し、まず、先頭から1つ制約を取り出す。その制約を用いて矩形を削減した後、削減率が最大となる変数を求める。その変数が使われている制約を優先度付きキューの先頭に追加し、元々格納されている制約をスキップする仕組みにより制約選択の優先度を向上する仕組みを実装した。

実験として、変数と制約がそれぞれ 10, 20, 30, 40, 50, 100 個で、制約の順番を入れ替えたものを3種類、合計 18 個の制約充足問題を用意して実験を行った。その結果、人工的な例 (100 変数) において、制約選択機構がなしの場合に 2.0 秒だったのが、0.1 秒に求解時間が短縮できた。ただし、明確に効果が確認できたのは 6 個のみであった。その原因としては、削減率の計算に時間がかかっていることが挙げられる。

6 まとめ

本稿では、制約充足問題を効率良く記述し、求解するために、パラメータ化制約と削減率に基づく制約選択機構の提案を行った。

参考文献

- [1] F. Benhamou, L. Granvilliers, Handbook of Constraint Programming, Chapter 16, pp.571-603, 2006.
- [2] F. Benhamou, F. Goualard, L. Granvilliers, J. F. Puget, Revising hull and box consistency, Procs. ICLP, pp.230-244, 1999.