分散制約推論：マルチエージェントシステムの基盤技術

横尾　真
九州大学大学院
システム情報科学研究院
知能システム学部門

1

経歴／研究分野

経歴：

- 1986年 東京大学工学系研究科 電気工学専攻 修士課程修了，同年 NTT入社
  - 1988年にICOT研修 (3ヶ月間)
- 1995年：博士（工学），東京大学工学系研究科電子情報工学専攻，渕研究室
- 2004年より現職

研究分野：

- 計算機科学
  - 人工知能
    - »マルチエージェントシステム
      - 分散制約充足
      - オークション理論／メカニズムデザイン

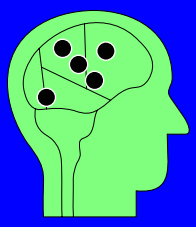
2

アウトライン

- 分散制約充足問題
  - –問題の定義
  - –例題・適用領域
  - –アルゴリズム

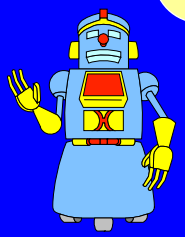
3

エージェントとは?



心の社会?

移動プログラム?



知的ロボット?

4

エージェントとは?

辞書的な定義
- The producer of an effect
- An active substance
- A person or thing that performs an action
- A representative …

要するに,
- 何らかの行動をする主体

マルチエージェントシステムとは,行動をする複数の主体 (エージェント) から構成されるシステムの,エージェントの相互作用に着目した研究分野

5

マルチエージェントシステムの基礎理論

- 論理
- ゲーム理論／経済学
  - 利己的なエージェントの行動の予測
  - メカニズムデザイン: 集団意思決定のためのプロトコルを設計する
- 探索/制約充足
  - 協調的なエージェント間の行動の調整
  - 分散制約充足問題: 各エージェントは変数を持ち, これらの変数に制約を満たす値を割当てる

6

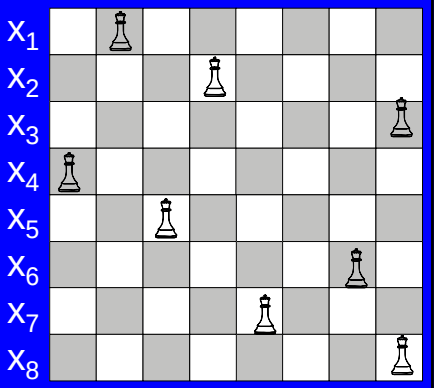
制約充足問題の定義

定義:

- 有限で離散的な領域 $D_1, D_2, \ldots, D_n$から値を取る
  変数の集合 $x_1, x_2, \ldots, x_n$
- 制約の集合，制約は変数を引数とする述語として記述される (e.g., $p_{ij}(x_i, x_j)$)

目標:

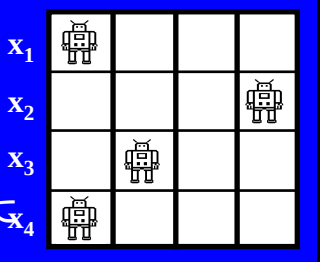
- すべての制約を満足する変数の値の組合せを求める (NP完全)



分散制約充足問題

定義:

- エージェント 1,2,…,n
- 各エージェントは一つないし複数個の変数を持つ.
- エージェント内，エージェント間に制約が存在する



前提:

- エージェント間通信はメッセージ通信によって実現される.
- 各エージェントは問題に関する部分的な知識のみを持つ

なぜ分散制約充足問題を考えるか?

- 同じ環境内で動作する複数のエージェントの行動間には，なんらかの制約条件が存在するのが通例
- 制約充足技術はエージェント間で整合の取れた行動を実現するための一般的な方法を与える
  - »協調のためのインフラストラクチャ
- 様々なマルチエージェントシステムの応用問題が分散制約充足問題として表現可能

9

アウトライン

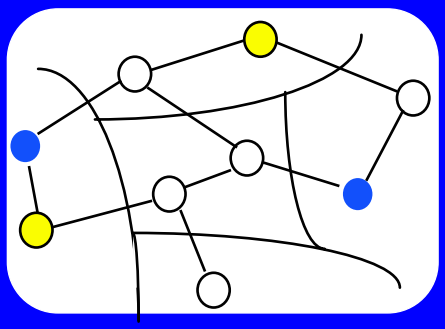
- 分散制約充足問題
  - –問題の定義
  - –例題・適用領域
  - –アルゴリズム

10

応用問題: 分散資源割当て

(Conry, *et al.* 91)

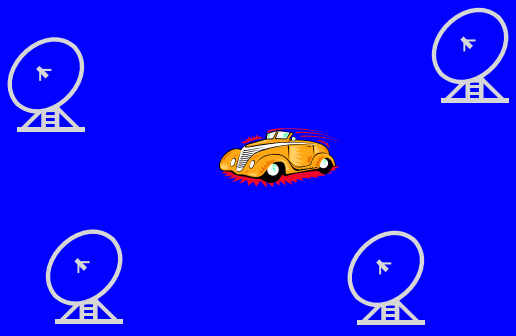
- 分散型の通信ネットワーク
- 接続要求を満足するプランの組合せを求める
- 分散制約充足問題として定式化可能
  - 各エージェントは各要求に対応する変数を持つ
  - 変数の領域は可能なプラン



11

応用問題: センサネットワーク

- 移動体をセンサがトラック
- 複数のセンサが同時にトラックしないと正確な位置が分からない



12

人工知能アドバンスドコース: 分散制約充足アルゴリズム

何が難しいか?

通常の制約充足問題の難しさに加えて,

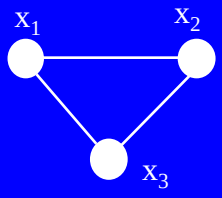
- 各エージェントは問題に関する部分的な知識しか持っていない.
- なるべく問題に関する知識を集中させずに解を求めたい --- 知識を移動させるコスト, プライバシー・セキュリティ
- なるべくグローバルなコントロールなしで解を求めたい --- 制御のボトルネック, 処理の並列性

エージェントが非同期・並行に, ローカルな知識に基づいて, グローバルな制御なしに動作して解に到達できるか?

13

分散制約充足問題を解くアルゴリズム

- 同期バックトラッキング
- 非同期バックトラッキング
- 非同期弱コミットメント探索
- 分散breakout
- 分散制約最適化



簡単化のための仮定

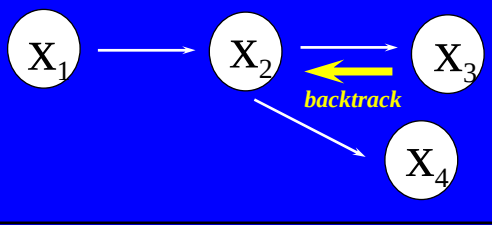
- –各エージェントは唯一の変数を持つ
- –すべての制約はバイナリ(二変数間)

14

同期バックトラッキング

- メッセージ通信により集中型のバックトラッキングをシミュレートする
- 制約ネットワークが木であれば並列処理が可能 (Collin, *et al.*, 91)

グローバルな知識が必要（順序があらかじめ決められている必要がある）



15

非同期バックトラッキング (Yokoo, *et al. ICDCS*-92)

特徴：
- 各エージェントはグローバルな制御なしに，並行，非同期に動作
  - » 各エージェントは暫定的な値の割当てを通信し，制約条件違反が生じていれば調整を行う

利点：
- 通信／制御のボトルネックがない，セキュリティ／プライバシー

研究課題：
- アルゴリズムの完全性を保証する
  - » 処理の無限ループを避ける
  - » 行き詰まり (dead-end) からの脱出

16

無限ループの回避

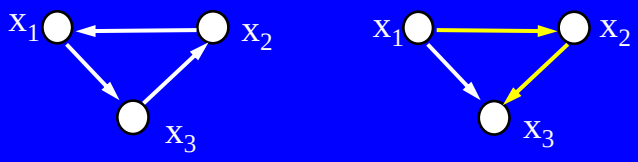
処理の無限ループの原因:

– 値の通信／変更のループ= ***制約ネットワークにおけるサイクル***

対応策:

– エージェントの優先順位を用いる (e.g., 識別子のアルファベット順)

» 分散システムにおけるデッドロックの回避方法に類似



17

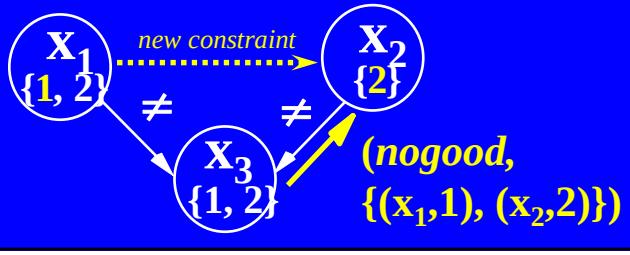
デッドエンドからの脱出

制約を満足する値が存在しない場合:
逐次的なバックトラッキング:直前の決定を変更

– 非同期な変化がある場合，単純な制御は不適切

非同期バックトラッキング:新しい制約条件(nogood)を導出／通信

– 他のエージェントが新しい制約条件を満足するように値を変更;制約条件の送り手はデッドエンドから脱出
– 非同期，並行的に実行可能



18

人工知能アドバンスドコース: 分散制約充足アルゴリズム

分散制約充足問題を解くアルゴリズム

- 同期バックトラッキング
- 非同期バックトラッキング
- 非同期弱コミットメント探索
- 分散breakout
- 分散制約最適化

19

非同期弱コミットメント探索 (Yokoo, CP-95)

非同期バックトラッキングが非効率な原因:

- 優先順位が高いエージェントの決定が悪い場合に，その決定を変更するために網羅的な探索が必要で解への収束に時間がかかる

改善方法:

- より良い決定を行なう:制約違反最少化ヒューリスティックの導入
  - »他の(優先順位の低い)エージェントに対する思いやり
- 網羅的な探索なしに悪い決定を変更する: 優先順位の動的な変更

20

優先順位の動的な変更

- 変数／エージェントの優先順位を表わす非負の整数値（優先度）を定義
  - 優先度の大きい変数／エージェントの方が優先順位が高い
- 優先度が同じ場合は識別子のアルファベット順で決定
- 優先度の初期値は0
- デッドエンドになったエージェントは自分の優先度をm+1に変更（mは関連するエージェントの優先度の最大値）

21

分散制約充足問題を解くアルゴリズム

- 同期バックトラッキング
- 非同期バックトラッキング
- 非同期弱コミットメント探索
- 分散breakout
- nogood学習／分散制約最適化

22

分散breakout
(Yokoo & Hirayama, ICMAS-96)

基本的なアイデア

- 近傍での相互排除
  - 近傍のエージェントが同時に値を変更すると，制約違反の個数が減少しない可能性がある
  - 一度に一つのエージェントのみが動作するように強い制御を行うと並列性が生かせない
- 準局所最適での重みの変更
  - エージェント全体として局所最適に陥ったことを検出するにはグローバルな通信が必要

近傍での相互排除

- 近傍のエージェント間で，可能な改善の度合い (制約条件違反をいくつ減少させられるか) を交換
- 近傍間で，最大の改善が可能なエージェントのみに値を変更する権利を与える (同点の場合はエージェントのidで決定)

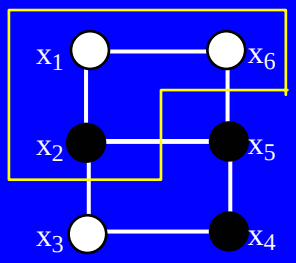
近傍でなければ同時に動作可能

準局所最適

真の局所最適より弱いが，近傍の情報のみを用いて検出可能な条件

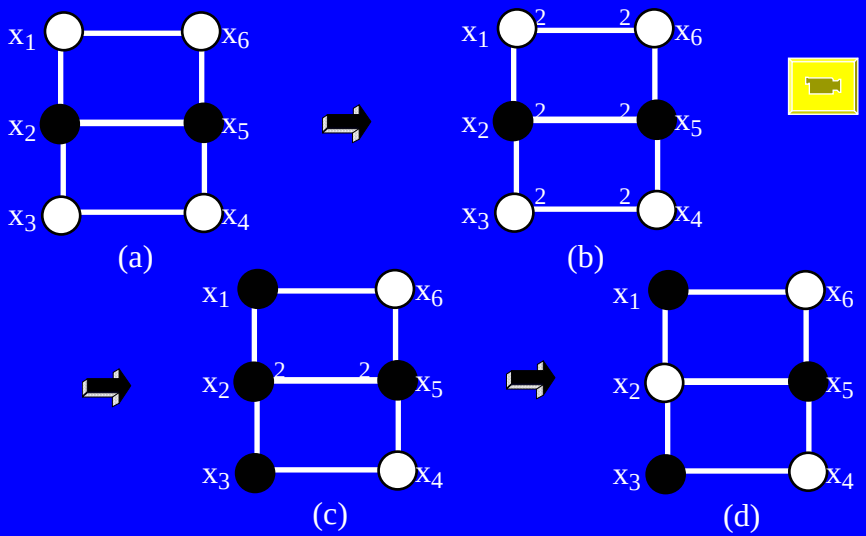
定義: x_i の視点から見て，現在の状態が局所最適であるとは，以下の条件が成立すること

- x_i がいずれかの制約を満足せず，x_i および他の近傍のエージェントの可能な改善量が0



25

アルゴリズムの実行例



26

人工知能アドバンスドコース: 分散制約充足アルゴリズム

分散制約充足問題を解くアルゴリズム

- 同期バックトラッキング
- 非同期バックトラッキング
- 非同期弱コミットメント探索
- 分散breakout
- 分散制約最適化

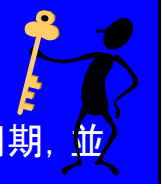
27

分散制約最適化問題

- 通常のCSPでは各制約は完全に満足されるか，されないかの二値.
- 制約に対してコストを割当てることにより拡張:
  - e.g., 制約 x1≠x5 を違反するとコスト10, 制約 x1≠x3 を違反するとコスト 5.
- 目的は違反する制約のコストを最小化する解を求めること.
- 通常の (分散) CSP は，コストが0か無限大の特別な場合に対応.

28

ADOPT (Asynchronous Distributed OPTimization) アルゴリズム



性質:

- 完全に非同期である; 各エージェントは非同期, 並行して動作する.
- 解の最適性が保障される.
- 必要なメモリ量は変数の数に対して多項式のオーダで抑えられる.

これらの性質をすべて満たす最初のアルゴリズム

キーアイデア:

- nogood を最適化問題が扱えるように拡張する.
- 最良優先探索を行う.

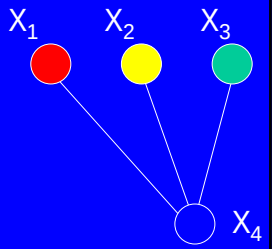
29

Nogoodの拡張

Nogoodに対して閾値を付加する, 例えば:
[{(x1, r),(x5, r)}, 10],
{(x1, r),(x5, r)} は, コスト10未満の解が必要なら nogood

以下のように新しいnogoodを生成:

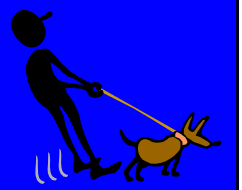
- for red: [{(x1, r),(x4, r)}, 10]
- for yellow: [{(x2, y),(x4, y)}, 7]
- for green: [{(x3, g),(x4, g)}, 8]
- then, [{(x1,r), (x2,y), (x3,g)}, 7], (7 は 10, 7, & 8の最小値)



Nogoodおよび閾値は単調に増加!

30

ADOPTにおける最良優先探索

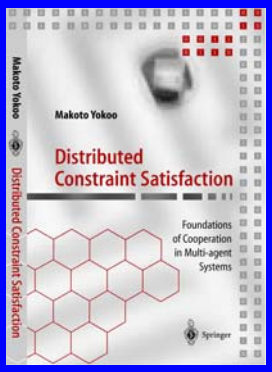


- 各エージェントは，現時点で分かっている情報に基づいて，コストを最小化する値を選択．
- コストに関する情報は，nogoodを介して統合／通信される．
- エージェントはいずれは最適解に到達．
- 情報を統合した後のnogoodは捨てても良い．これによりメモリの必要量は多項式オーダで抑えられる．

31

分散制約充足研究の経緯

- 1987年頃に研究をスタート
  - 当初は，マルチエージェント，制約プログラミングの両方のコミュニティであまり評判が良くなかった
- 次第に両方のコミュニティから評価を得る
- 両方の研究コミュニティの拡大：国際会議，論文誌
- 分散制約充足ワークショップ
  - 2000年から毎年開催
- 2001年 Springerから単行本を出版



32

参考文献

- 教科書: M. Yokoo and T. Ishida, `Search Algorithms for Agents', in Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Gerhard Weiss ed., MIT Press, 1999.
- 解説 :横尾 真, 平山 勝敏, `解説: CSPの新しい展開:分散／動的／不完全CSP', 人工知能学会誌, Vol.12, No.3, 1997.
- 専門書 : M. Yokoo, `Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems', Springer, 2001.