

Solver for COnstraint Programming

制約最適化問題（重み付き制約充足問題）を解くための
アルゴリズム



制約最適化ソルバーSCOPと メタヒューリスティクス

野々部宏司（法政大学）

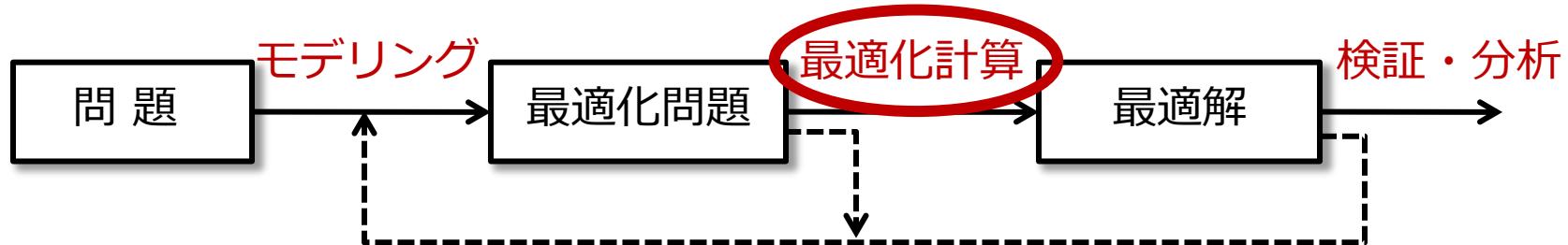
概要

- 背景・目的
- 重み付き制約充足問題（制約最適化問題）
- アルゴリズムの概要
- 適用事例

最適化アプローチ

□ 最適化

- 意思決定・問題解決のためのひとつの手段
- 最適化問題として定式化（モデリング） + 最適解の計算 + 最適解の検証・分析



ここでは**最適化計算**に着目

最適化問題

- 種々の制約のもと，ある与えられた関数を最小化/最大化する問題の総称

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in F \end{aligned}$$

- $f(x)$: 目的関数
- F : 実行可能領域
- $f(x)$ や F の構造として何を仮定するかによって様々な問題が考えられる

組合せ最適化問題

最適化問題

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & x \in F\end{array}$$

- 制約が組合せ的な構造をもつとき、**組合せ最適化問題**と呼ばれる
 - F が離散集合（順列や組合せ・0-1ベクトルの集合など）
- **実社会において広く現れる問題構造**
 - 生産計画
 - スケジューリング
 - ネットワーク設計
 - など

最適化の方法（1）

□ 既存のソルバーを利用

□ 混合整数計画 (Mixed Integer Programming, MIP)

線形等式・不等式制約のもとで、1次関数を最小化/最大化する問題。
変数は実数もしくは整数

- CPLEX, Gurobi, SCIP, GLPK, Excel ソルバー, ...
- モデリング言語が提供されていることが多い、問題記述が容易
- 定式化の工夫が必要になることが多い
- 実用的に解ける問題の規模が限定的であることが多い
 - 現実に解くべき組合せ最適化問題の多くは NP困難
 - 厳密な最適解を効率よく計算することはおそらく不可能： $P \neq NP$ 予想

最適化の方法（2）

- プログラムを自分で作成
 - 方針1：厳密な最適解を計算することを目指す
 - 方針2：厳密性にこだわらず、近似最適解で構わないと考える
 - 一定時間内に実用上許容可能な解が計算できればよい
 - **メタヒューリスティクスは、これを実現するひとつの手法**
- メタヒューリスティクス
 - 計算困難な最適化問題に対する実用的アプローチ
 - アルゴリズムの「枠組み」 ⇒ **設計・実装を適切に行うことが必要**
 - **時間的・予算的制約から、個別のアルゴリズム開発が困難な状況も**
 - 最適化：意思決定・問題解決のためのひとつの「手段」
 - ⇒ アルゴリズム設計の枠組みや指針・テンプレートの提供
 - ⇒ **実装済みのアルゴリズム（ソルバー）の提供**

目的

- いろいろな問題に適用可能な汎用ソルバーを開発し、実行可能なソフトウェアとして提供
 - 対象を**組合せ最適化問題**（静的・確定的）に限定
 - **実務利用を念頭**
 - 良質の解を実用的な計算時間で出力することを目指す
 - 理論的な最適性は保証しない
 - ⇒ メタヒューリスティクス

方法：標準問題によるアプローチ

□ いくつかの標準問題をソルバーとともに用意

1. 解きたい問題のタイプに応じて標準問題を選択
2. 標準問題の形に記述してソルバーを適用

□ 標準問題の例

- **混合整数計画問題** (Mixed Integer Programming, **MIP**) ⇒ **Gurobi** など
- **重み付き制約充足問題**
(Weighted Constraint Satisfaction Problem , **WCSP**) ⇒ **SCOP**
- **資源制約付きプロジェクトスケジューリング問題**
(Resource-Constrained Project Scheduling Problem, **RCPSP**) ⇒ **OptSeq**
- ...

制約充足問題 (CSP)

- n 個の変数 X_i , X_i がとり得る値の集合 D_i ($i = 1, 2, \dots, n$)
- すべての制約を満たすように、各変数 X_i に値 $j \in D_i$ を 1 つずつ割当てることが目的

すべての制約を満たす解（割当て） x が存在するとは限らない
存在するとしても、見つけることは計算困難

重み付きCSP (WCSP)

- それぞれの制約 C_l について,
 - 違反度に応じたペナルティ $p_l(x)$ (制約を満たすとき 0)
 - 制約の重要度に応じた重み $w_l \geq 0$
- ペナルティの重み付き和 $\sum_l w_l p_l(x)$ を最小化
 - 絶対制約を満たしたうえで, 考慮制約の違反度を最小化
- いろいろなタイプの割当て問題を自然な形で定式化可能
 - 時間割問題, シフトスケジューリング, 施設配置, …

例：ナーススケジューリング問題

- #### □ 各ナースの毎日のシフトを決定

□ 制約

- シフト拘束制約
 - 適正人数のナースを確保
 - ナース拘束制約
 - ナースの労働負荷や勤務希望を考慮

□ WCSPへの定式化

- 変数：（ナース，日にち）の各組に対応
 - 領域：シフトの集合

代表的な制約・ペナルティ関数

□ 線形または2次等式・不等式

- 変数 X_i と値 $j \in D_i$ の組それぞれに対して、**0-1変数** x_{ij} を導入

$$x_{ij} = 1 \Leftrightarrow X_i = j$$

- 0-1変数 x_{ij} に関する線形または2次の等式・不等式
- たとえば、「(左辺) \leq (右辺)」の形の不等式に対しては、
ペナルティ := $\max\{0, (\text{左辺}) - (\text{右辺})\}$

□ all-different 制約

- 「与えられた変数集合 V に含まれる変数はすべて異なる値をとらなくてはならない」
- ペナルティ := $|V| - (V$ に含まれる変数がとっている値の種類数)

WCSPソルバー (SCOP) の概要

□ 基本的な枠組みはタブー探索

- 局所探索法の拡張. 近傍内の最良解に移動（改悪でも可）
- 近傍解すべてを探索対象とはしない

□ 基本要素

- **探索空間**：解（割当て）集合全体
- **近傍**：ある 1 つの変数の値を変更することで得られる解の集合
- **タブーリスト**：値を変更した変数をしばらくの間保持，リスト内の変数の値は変更不可
- **初期解**：ランダム割当てを欲張り法で改善

[参考文献] K. Nonobe and T. Ibaraki: An improved tabu search method for the weighted constraint satisfaction problem, *INFOR* 39, pp.131–151 (2001).

WCSPソルバー (SCOP) の概要

□ 効果的・効率的な探索

- 近傍解評価のための補助メモリ利用（高速化）
- 近傍探索領域の限定
- プログラム・パラメータの自動調整
 - タブー期間（変更禁止期間）
 - ペナルティ重み：評価関数の動的調整

近傍解評価のための補助メモリ利用

□ 近傍解の評価

- $X_i, j (\in D_i)$ の組それぞれに対して、 X_i の値を j に変更したときの
ペナルティ変化量 $p(x(X_i \leftarrow j)) - p(x)$ を事前に計算
⇒ 定数時間で近傍解を 1 つ評価することが可能
- 解の移動時には情報の更新が必要

近傍探索領域の限定

- すべての近傍解をチェックすることは非効果的
 - 「多くの計算時間要する」ことだけが問題ではない
 - タブー探索は近傍探索領域内の「最良解への移動」が原則
 - ペナルティにほとんど影響を与えない「軽微な」修正ばかりが繰り返され、「山を越える」改善が起こりにくい

現在の解において満たされていないどの制約についても
ペナルティ値も減少させる効果のない値変更は行わない

パラメータの自動調整 (1)

□ タブー期間 (tabu tenure)

タブーリストにより、変数の値の変更が禁止される期間

□ タブー期間が短い場合

- 繊密な探索（集中化）
- 偏った探索 — 探索履歴から判断し、タブー期間増

□ タブー期間が長い場合

- 解の循環防止、探索の多様化
- 過剰な禁止 — 特別選択規則 (aspiration criteria) により判断し、タブー期間減

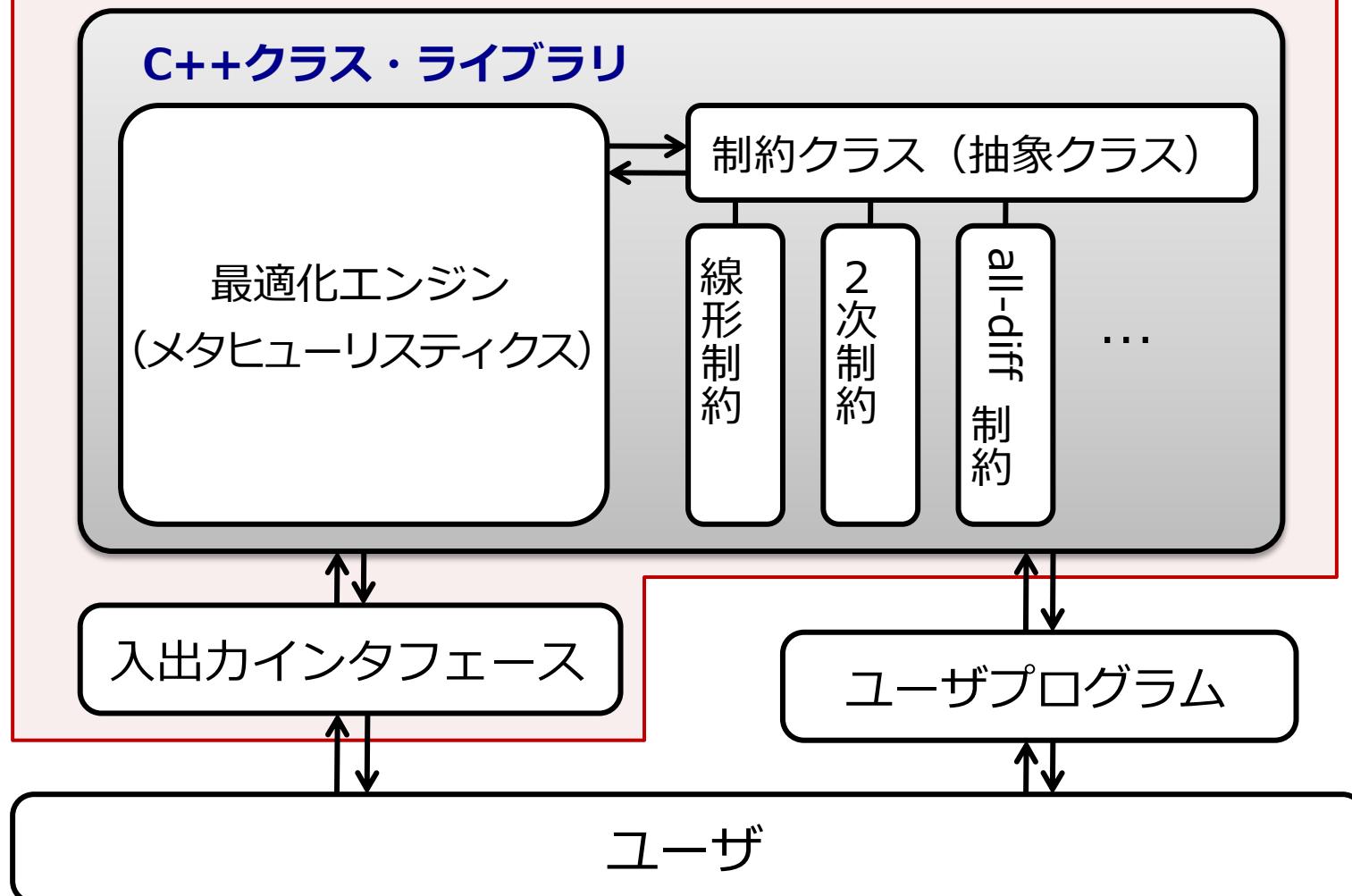
パラメータの自動調整 (2)

□ ペナルティ重み

- ペナルティ関数 $\sum_l w_l p_l(x)$ をそのまま近傍解の評価関数として用いる探索は非効果的
 - 近傍解の良し悪しが重みの大きい制約に強く依存し, その他の制約は軽視されやすいため, 重みの大きな制約を満たす解に探索が制限
- 近傍解の評価には $\sum_l v_l p_l(x)$ を使用
 - $0 \leq v_l \leq w_l$ の範囲で, v_l を自動調整

ソルバーの概要

最適化ソルバー



ユーザ定義制約の追加

□ 「制約を定義」 = 「ペナルティ関数を定義」

実装上は、以下の関数を用意

- (i) 与えられた解 x に対して、ペナルティ $p_l(x)$ を計算し、ペナルティ変化量 $p_l(x(X_i \leftarrow j)) - p_l(x)$ を列挙（非零のみ）
- (ii) 制約に関与している（ペナルティの計算において値を参照する必要のある）0-1変数を列挙

適用事例

- 商用の数理計画パッケージに組み込み
- 国際コンペティション
 - International Timetabling Competition 2007 (ITC2007)
 - International Nurse Rostering Competition 2010 (INRC2010)
- ...

適用例：時間割作成

ITC2007 (International Timetabling Competition)

- トラック1（試験時間割） 3位
 - 試験数：200～1000
 - ピリオド数：20～80, 教室数：1～50
 - 学生数：5000～16000
- トラック2（履修登録に基づいた授業時間割） 2位
 - 授業数：200～400
 - ピリオド数：45, 教室数：5～20
 - 学生数：300～1000
- トラック3（カリキュラムに基づいた授業時間割） 3位
 - 教科数：150～450
 - ピリオド数：25～45, 教室数：5～20

計算時間：約5～10分

[参考文献] 萩木俊秀, 熱田光紀, 野々部宏司：汎用ソルバによる時間割作成—国際コンペティションITC2007に参加して—, スケジューリング・シンポジウム2008講演論文集, pp.173–176 (2008).

適用例：ナーススケジューリング

INRC2010 (International Nurse Rostering Competition)

□ 3つのトラック : **sprint, medium, long**

- ナース数 : 10 (sprint), 30~31 (medium), 49~50 (long)
- 日数 : 28
- シフト数 : 4~5 (sprint), 5~6 (medium), 6 (long)
- 計算時間 : 10秒 (sprint), 10分 (medium), 10時間 (long)
- 制約数 : 約1800~3600 (sprint)
約3300~11,500 (medium)
約7200~21,500 (long)

[参考文献] 野々部宏司 : メタヒューリスティクスを用いた制約最適化ソルバーのナーススケジューリング問題への適用,
電気学会研究会資料, システム研究会, ST-10-6, pp.27–31 (2010).

INRC2010：問題概要

□ シフト拘束制約：絶対制約

- 各日、各シフトについて、指定された人数のナースを過不足なく確保

□ ナース拘束制約：考慮制約（各制約に重み）

制約の有無、上下限値、週末の定義はナースによって異なる

- 各シフトについて、スケジュール期間中の割当て回数に上下限
- 連続勤務日数/連続休暇日数に上下限
- 完全休暇ではない週末の連続数に上限
- 同じ週末に含まれる日には同じシフト
- 夜勤終了後、2日間の休暇日
- 望ましくないシフトの並びを禁止（夜勤の翌日に日勤など）
- いくつかの日に、割当て指定シフト/割当て禁止シフト

□ 考慮制約違反に対するペナルティの加重和を最小化

INRC2010 : コンペティション結果

NURSE ROSTERING
COMPETITION

Log in Register

PATAT 2010

Home Dates Competition Rules Competitor Ranking

Home > Competitor Ranking

Competitor Ranking

For each track, the final ranking of the finalists is presented. The results take the best solutions for each instance (Early, Late and Hidden) into account. For the hidden instances we performed ten runs for the sprint and medium track instances and four runs for the long track instances. A list of competitors can be found [here](#).

Sprint Track

Competitor	Rank
C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, E. Housos	2,08
K.Nonobe	2,45

Organising Partners

CODES

SINTEF
Scheduling & Timetabling Group
DIEGM - University of Udine

Competition Chair
prof. dr. Patrick De Causmaecker
CODES research group

INRC2010 : コンペティション結果

Sprint Track

Competitor	Rank
C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, E. Housos	1,93
E.K. Burke and T. Curtois	2,27
B. Bilgin, P. Demeester, M. Misir, W. Vancroonenburg, G. Vanden Berghe, T. Wauters	2,60
K. Nonobe	3,73
D. Rizzato, A. Constantino, E. Luiz de Melo, D. Landa-Silva, W. Roméo	4,47

Medium Track

K. Nonobe

Zhipeng L. And .

E.K. Burke and T.

B. Bilgin, P. Den

Long Track

Competitor

Rank

Competitor

Rank

C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, E. Housos

1,93

E.K. Burke and T. Curtois

2,27

B. Bilgin, P. Demeester, M. Misir, W. Vancroonenburg, G. Vanden Berghe, T. Wauters

2,60

K. Nonobe

3,73

D. Rizzato, A. Constantino, E. Luiz de Melo, D. Landa-Silva, W. Roméo

4,47

まとめ

- 実務利用を念頭に置いた汎用ソルバーの開発
 - WCSPに対するメタヒューリスティックアルゴリズム
- 今後の方針性
 - 性能向上
 - 適用範囲拡大：標準問題の拡張， その他の標準問題
 - 利用指針：標準問題の選定， モデリング