

# 次世代数理計画法システム LocalSolver による 大規模組合せ最適化問題への適用

01606110 MSI 株式会社  
MSI 株式会社

\*宮崎 知明 MIYAZAKI Tomoaki  
石村 猛 ISHIMURA Takeshi

## 1. はじめに

IT 技術の進歩は目覚しく、ハードウェア性能 (CPU、メモリ、ネットワーク、記憶媒体)、ソフトウェア (最適化理論、GUI など) の進化により、20 年前と比べて計算能力、表現能力が飛躍的に増大している。また、コンピュータで扱うことができるデータがあらゆる分野で蓄積され、かつ、リアルタイムで獲得できるようになってきている (「ビッグデータ」の世界)。

このため、昔であったら実現できなかったような人間の思考を支援するシステムの構築が可能になってきた。今後、ますます実践的な最適化システムの要求が高まっており、大規模組合せ最適化問題の実用解法が求められている。本稿では、LocalSolver を大規模組合せ最適化問題に試行した結果について述べる。

## 2. 背景

大規模組合せ最適化問題に纏わる課題は様々である。大規模組合せ最適化問題の例としては、以下がある。

- ・ 非線形の割り当て問題：車両の優先順位付け、頻度割り当て等
- ・ パッキング&カバーリング：メディア・プランニング、グラフ分割等
- ・ 設備配置、ロジスティッククラスタリング、通信網の最適化等
- ・ 機械/人員スケジューリング、グループプランニング、看護師スケジューリング等

大規模組合せ最適化問題の特徴は以下：

- ・ 組合せ最適化問題はありとあらゆる場面に存在し、その目的、条件は様々である。
- ・ 大規模組合せ最適化問題は、意志決定変数が膨大になるだけでなく、制約条件が複雑になるケースが通常である。
- ・ 既存の最適化パッケージ (数理計画法システム、制約論理システム等) では、モデルの表現、

求解時間の観点から、大規模問題では現実的に使える汎用パッケージがないのが現状である。

- ・ ほとんどの組合せ最適化問題は、変数に整数制約を含む形で定式化できるが、問題のサイズを大きくすると、整数変数の数が組合せ的に増加するため、全ての解を列挙して解くことは現実的ではない。
- ・ 汎用的な最適化手法として、数理計画法、制約論理プログラミングが使われているが、整数変数が 1、000 を超えると実用的な意味で、解を求めるのが難しいのが現状である。
- ・ 現実的には、問題対応型メタヒューリスティック解法として、モデルを設計し、解法を設計して解を求めていく方法がある。しかしながら、この場合には、現実の問題をどう数学モデルにするか、また、数学モデルをどう解いていくかを考える必要があり、手軽に使うことが非常に難しい世界になっている。

## 3. LocalSolver の概要

LSP 言語を使えば、モデルの定義、制約の定義、目的の定義ができる。また、LP、MIP のように連立不等式として問題を定義する必要はない。

今回、大規模組合せ最適化問題解決のために試行した LocalSolver は、フランスの OR 実務研究者 6 人からなる 5 年間の研究プロジェクトの成果であり、2012 年の夏から市場に投入された製品である (製品名が LocalSolver)。

LocalSolver は百万以上の意志決定変数をもつ大規模組合せ最適化問題でも数日間でモデリングできる汎用的なモデリング言語機能とその言語をインタープリティブで最適化処理 (求解) できる最適化プログラム機能をもつことから、次世代の数理計画法システムと言える。

以下の LocalSolver の概要を示す。

### 3.1 モデリング言語機能：LSP

LSP 言語は、記述量をできるだけ少なくするよう、かつ、制約条件、目的関数をできるだけシンプルに表現できるよう設計された次世代の言語とでも呼べる表現力を持っている。以下にその特徴を、また、図 1 に複数目的をもつナップザック問題の LSP 例を示す。

- 意志決定変数を 0 – 1 整数変数として定義
  - この意志決定変数の値の組合せが一つの解となる。
  - 探索は意志決定変数の値の組合せで行われる。
- 制約式、目的関数の定義
  - 意志決定変数を使って制約式、目的関数を表現する。
  - 式の表現に、算術演算子、論理演算子等が使用でき、非線形表現でも、そのまま記述できる。
- 外部入出力機能、表示機能他
  - 数値データの外部入力、結果の外部出力機能により、モデル本体を簡潔に表現可能。

```
function model()
// 0-1 decisions
x[0..7] <- bool();
weights = {10,60,30,40,30,20,20,2};
values = {1,10,15,40,60,90,100,15};
// weight constraint
knapsackWeight <- 10*x[0]+60*x[1]+30*x[2]+40*x[3]+30*x[4]+20*x[5]+20*x[6]+2*x[7];
constraint knapsackWeight <= 102;

// maximize value
knapsackValue <- 1*x[0]+10*x[1]+15*x[2]+40*x[3]+60*x[4]+90*x[5]+100*x[6]+15*x[7];
maximize knapsackValue;

// secondary objective: minimize product of minimum and maximum values
knapsackMinValue <- min[i in 0..7]{x[i]? values[i] : 1000};
knapsackMaxValue <- max[i in 0..7]{x[i]? values[i] : 0};
knapsackProduct <- knapsackMinValue * knapsackMaxValue;
minimize knapsackProduct;
}
```

図 1. 複数目的をもつナップザック問題の LSP 例

### 3.2 解探索プログラム機能

LocalSolver は局所探索法をベースとしたメタヒューリスティック解法の一つであるが、解探索空間、解探索に関して、使用者が解及び解空間を意識しなくても、自動的に最適化を行うことが出来るようになっている。

LocalSolver の解探索の特徴を以下に示す。

- 解探索を 1 秒間に数万回行う。
  - 分枝限定法のように子問題を解くような探索ではなく、近傍の解の組合せを一つ変えて差分計算をするため、計算コストがはるかに小さい。
  - 解探索は、単調性を保っておこなわれる。
- 探索空間を自動的に変更して探索を進める。
  - シミュレーテッドアニーリングで解空間を変更する。
  - 解空間探索に学習機能を付けることで、自動的に効率の良い解空間探索を行う。
  - 最初に実行可能性を探索し、その後、複数の目的関数に関して、指定された順に探索を行う。

## 4. 事例

大規模裁断計画問題に適用した事例を紹介する。問題の規模は、13,000 以上のパターン候補から複数パターンを選ぶ最小化問題である。既存の数値計画法システムでは、10 分以上実行しても途中解であり、実用的には利用できなかったのに対し、LocalSolver では、10 秒程度で実用的な実行可能解を算出できている。詳細は、当日発表する。

また、人員配置スケジューリングの例も当日示す。

## 5. おわりに

30 年前には殆ど実現出来なかった大規模組合せ最適化問題に対して、実践的なアプローチが実現できる時代になったと考える。「実学に役立つ OR」として、人間と機械の調和を実現して日本の産業界の再生の一助となれば幸いである。

## 参考文献

- 1) T. Benoist, B. Estellon, F. Gardi, R. Megel, K. Nouioua (2011).
  - 「LocalSolver 1.x: a black-box local-search solver for 0-1 programming」、*4OR, A Quarterly Journal of Operations Research* 9(3), pp. 299-316. Springer.
- 2) MSI 株式会社  
「<http://msi-jp.com/localsolver/>」ホームページ