

# 新しいソリューションを可能とする LocalSolver (新数理計画法システム)

## —最新かつ汎用的なモデリング言語 (LSP) —

01606110 MSI 株式会社

\*宮崎 知明 MIYAZAKI Tomoaki

### 1. はじめに

大規模な数理最適化問題は、既存の数理計画法システム (MIP) や制約論理システム (CP)、非線形最適化法システム (NLP) では、解探索で組合せ爆発が起こり、実用時間内に解くことが出来なかったのが現状である。

LocalSolver は、局所探索をベースとして、大規模組み合わせ最適化問題を実用時間内で効率よく求めることを目的とした新しい数理計画法システムである。

LocalSolver はフランスの6人の若手OR実践者が5年の歳月をかけて開発したものであり、大規模最適化問題に対して、メタヒューリスティクス (ローカルサーチ) の解法をベースとした次世代の数理計画法システムである。LocalSolver は、1000万以上の0-1整数変数及び実数変数を扱うことができる。

LocalSolver の適用範囲 (問題) を図1に示す。

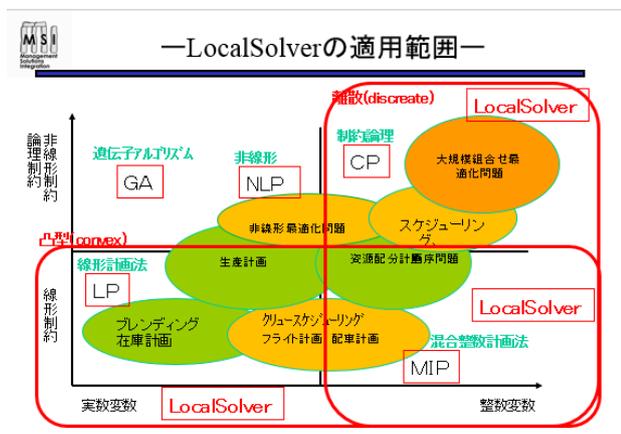


図1. LocalSolver の適用範囲 (問題)

また、LocalSolver は大規模最適化問題を汎用的かつコンパクトに定式化できるモデル記述言語 (LSP 言語) を備えている。LSP は、LP、MIP、CP、NLP の問題をひとつの言語体系で定義することが可能である。

本稿では、最新の技術である関数型のプログラミング言語の思想を取り入れた LocalSolver のモデル記述言語 (LSP 言語) とモデリングの考え方を紹介する。

### 2. LocalSolver による定式化の考え方

LocalSolver は、まず意思決定変数を定義し、意思決定変数を使って目的関数と制約条件を定義することで問題を定式化する。意思決定変数には、0-1 整数型意思決定変数 (bool) と上下限を持つ実数型意思決定変数 (float) がある。

最適化計算は、この意思決定変数の組合せで決定される目的関数値と制約条件値を超高速度に試行することで行われる。目的関数値、制約条件値を計算するだけで、探索をすすめるため、目的関数および制約関数の表現には、多次元での算術演算、論理演算、条件演算が可能である。LP、MIP、CP、NLP の問題を同じスキームで表現できる。

LSP 言語によるモデリングは以下の手順で行う。

- 1) 意思決定変数 (bool 変数及び float 変数) を定義する。
- 2) bool 変数及び float 変数を使い、制約条件、目的関数を定義する。

重要なことは、選択された意思決定変数により、制約条件値および目的関数値が決まらなければならない。

以下に大規模組み合わせ最適化問題を表現する典型的な bool 変数の定義イメージを示す。

- ナップサック問題 :  $X_p$  ( $p$  : 品物)
- ルート選択問題 :  $X_r$  ( $r$  : ルート)
- 裁断計画問題 :  $X_{p,q}$  ( $p$  : 裁断パターン、 $q$  : パターンの使用回数)
- 人員配置問題 :  $X_{p,t,j}$  ( $p$  : 人員、 $t$  : 時間、 $j$  : ジョブ)
- 車両投入計画 :  $X_{c,p}$  ( $c$  : 車両、 $p$  : ポジション (順番))
- SCM :  $X_{t,i,j,k,p}$  ( $t$  : 期、 $i$  : 工場、 $j$  : ライン、 $k$  : 倉庫、 $p$  : 製品)
- スケジューリング :  $X_{t,i,j,p}$  ( $t$  : 時間等、 $i$  : 工場、 $j$  : ライン、 $p$  : 製品等)

これにより、大規模な組み合わせ問題を効率的に定式化することができる。

### 3. LSP 言語

LSP 言語は、最適化問題をモデル化し、モデルの検証及び解の検証を行うフェーズでの試行錯誤を行うのに最適な環境を提供することを目的として開発されている。

LSP 言語は、最新の関数型プログラミング言語の考え方を導入した、関数型プログラミング言語であり、型推論を備えた言語であるため、コンパイラが自動的にデータの種別を推定するため、データの種別(型)をプログラマが指定する必要がない。

LSP 言語の特徴は以下：

- 迅速に開発できる (開生産性が良い。従来に比べて、1/5から1/2の開発量)
- バグを抑えやすい (コンパイラが型の間違い等を自動的にチェックする)
- アプリケーションの性能を向上させやすい
- 簡潔かつシンプルなモデリング言語 (できるだけ省略できるよう設計)  
※大規模問題でも制約条件及びデータがそろっていれば、1日でモデリングと実行が可能である。
- 作成(修正) ↔ 実行が同時にできる (一つはエディタ、もう一つは DOS コマンドプロンプトの二つのウィンドウを操作しながら開発が可能である。
- 目的計画法のように目的を段階的に設定することができるため、モデルの開発及び解の検証を段階的に行うことができる。

ナップサック問題を例に、LSP による定式化を示す。

品物が、8品あり、それぞれの重さと価値以下に定義する。

重さ：10、60、30、40、30、20、20、2 kg

価値：1、10、15、40、60、90、100、15 円

ナップサックには最大 102kg まで品物を入れることができ、価値が最大になる

よう、どの品物を選べばよいか、その時の価値はいくらになるかが問題である。

```
/****** toy.lsp *****/
```

```
function model()
{
  // 0-1 decisions
  x_0 <- bool 0; x_1 <- bool 0; x_2 <- bool 0; x_3 <- bool 0;
  x_4 <- bool 0; x_5 <- bool 0; x_6 <- bool 0; x_7 <- bool 0;
  // 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;
}
```

### 4. 事例

LocalSolver による事例を紹介する。

#### 1) 要員配置問題

##### 1. 意志決定変数の定義(Bool変数)

$X(\text{人員、日付、時刻、業務}) = \{0, 1\}$

ここまでは装置のスケジューリングとあまり変わらないが...

##### 2. 人のスケジューリング特有の制約

- ・最大勤務時間
- ・最小勤務時間←あまり短いとアルバイトは出てこない
- ・1日1シフト←勤務の途中に何もしない時間を入れないを導入。

実行結果 (業務モデル)

人員：83名、1週間、15分毎の54タイムスロット、5業務  
(意志決定変数：  $83 \times 7 \times 54 \times 5 = 156870$ )

LocalSolver3.1：準最適解まで約210秒  
既存 MIP：600秒以上

#### 5. おわりに

30年前には殆ど実現出来なかった大規模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
- 3) 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.
- 4) MSI 株式会社  
「<http://msi-jp.com/localsolver/>」ホームページ
- 5) T. Benoist, J. Darlay, B. Estellon, F. Gardi, R. Megel (2014).  
・ 「LocalSolver 4.0 Hybrid Math Programming」、Presentation slides at INFORMS 2014,  
・ <http://www.localsolver.com/home.html>