

# All-in-One ソルバに近づく LocalSolver の最新動向

01606110 MSI 株式会社

\*宮崎 知明 MIYAZAKI Tomoaki

## 1. はじめに

ビッグデータ時代を迎え、自動化したいスケジューリングや最適化計算問題のモデリング範囲、制約条件が複雑になっており、大規模な問題を汎用的に扱える実用解法の要求が高まっている

100 万以上の 0-1 整数変数をもつような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) 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は出来なかったのが現状である。

現実世界の多くの問題は大規模組み合わせ最適化問題となる。特に、下記のようなスケジューリング問題に関して、汎用的に解くことができるようになることを、LocalSolver は狙っている。

- ・ 車両の優先順位付け (組立) 問題
- ・ 裁断計画問題 (フィルムなど)
- ・ SCM 問題 (製造-輸送-在庫-販売など)
- ・ 最短経路問題 (カーナビのルート検索など)
- ・ ネットワーク問題 (交通網, 通信網, 電気, ガスなどの設計)
- ・ 配送計画問題 (宅配便, 店舗への商品配送, ゴミ収集など)
- ・ 施設配置問題 (工場, 店舗, 公共施設などの配置など)
- ・ 人員スケジューリング問題 (看護師等の勤務表, 時間割の作成など)
- ・ 機械スケジューリング問題 (工場の運転計画, 装置稼働計画など)

次世代の All-in-One 数理計画法ソルバを目指す LocalSolver は、解探索で組合せ爆発が起こらないよう、局所探索と既存の解法 (事前分析, 途中分析, ヒューリスティック等) を組み合わせたシステムであり、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。

大規模最適化問題を定式化するのに、従来のデータ形式 (MPS 形式データ, 式形式データ) では、開発に多大の工数を必要としていた。LocalSolver のモデル記述言語 (LSP 言語) は、コンピュータの能力を最大限に利用した最新の関数型言語であり、データとロジックを分離したインタープリタ型言語となっている。

本稿では、最新の LocalSolver の概要とモデル記述言語である LSP 言語によるモデリングの考え方を紹介する。LocalSolver の適用範囲を図 1 に示す。

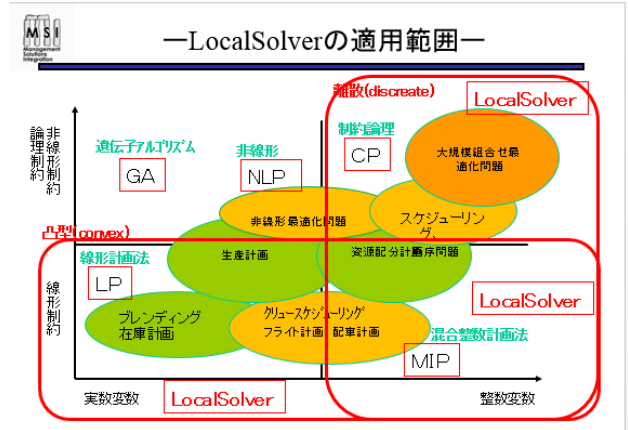


図 1. LocalSolver の適用範囲

## 2. LocalSolver の特徴

LocalSolverは、図 1. LocalSolverの適用範囲で示すように、以下の問題 (手法) を統一的に定義し、最適化することができる。

- LP (線形計画法)
- IP (整数計画法)
- MIP (混合整数計画法)
- NLP (非線形計画法)
- CP (制約論理計画法)

LocalSolverは、近傍探索だけでなく、前処理、制約伝播等、既存の数理計画法システムが開発してきたノウハウを取り入れている。図 2 にLocalSolverが取り入れている手法を示す。

Feasibility search Optimization ↑ ↓ Infeasibility proof Lower bound	Preprocessing	Neighborhood Search	Moves		
	Model rewriting Structure detection Constraint inference Variable elimination Domain reduction	Simulated annealing Restarts Randomization Learning Divide & Conquer	Combinatorial Small Compound Large Propagation	Continuous Small Compound Large Relaxation	Mixed Small Compound Large
	Tree search Interval branching	Discrete propagation Interval propagation	Dual linear relaxation Dual convex relaxation		

図 2. LocalSolver が取り入れている手法

### 3. 最新の LSP 言語機能

従来のLocalSolverは、0-1意思決定変数 (**bool**) で、制約条件、目的関数を定義するのが基本であり、bool変数の数がたとえ1000万変数を超えても実用的な意味で解を求めることができた。

最新のLocalSolverは、上下限を持つ実数意思決定変数 (**float**)、上下限を持つ整数意思決定変数 (**int**) 及びlist関数で指定するセット化された変数を意思決定変数として定義できる。意思決定変数で定義された制約条件、目的関数を計算しながら、超高速に試行することで、最適化を目指している。

また、LocalSolverは、意思決定変数の入れ替えで、目的関数、制約条件の値を差分計算し、目的関数の単調性を保証することで、最適化をおこなっている。このため、目的関数、制約条件を線形式にする必要はなく、自由に非線形関数として表現可能である。

また、非線形関数を定義できない場合には、区分線形関数として定義することも可能である。

#### List関数と関連する関数 :

```
構造化された意思決定関数 : list(n)
• オーダは {0, ..., n-1} からなる値のサブセットである。
• list内では、オーダの値はユニークである。

Listと関連する関数
• count(u) : list内で選択された値の数
• at(u,i) or u[i] : list内の i 番目の値
• indexOf(u,v) : list内の v の値を持つインデックス
• contains(u,v) : もし list内に v の値を持たないときは、-1を設定する
• disjoint(u1, u2, ..., uk) : もし u1, u2, ..., uk が list内でベアなら、ベアを解消する
• partition(u1, u2, ..., uk) : もし、u1, u2, ..., uk が list内にあれば {0, ..., n-1} のパーティションをつくる
```

以下にlistによるTSP、VRPの例をしめす。

#### TSPの例

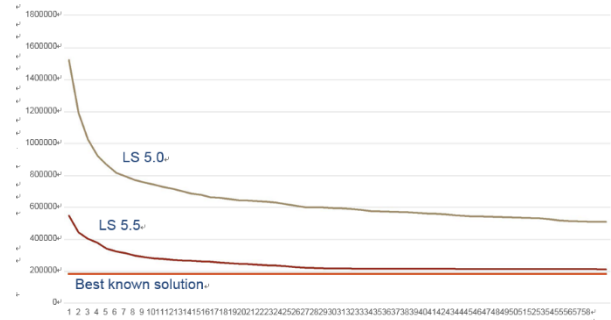
```
function model() {
  x <- list(N); // order n cities {0, ..., n-1} to visit
  constraint count(x) == N; // exactly n cities to visit
  minimize sum[i in 1..N-1]( Dist[ x[i-1] ][ x[i] ] )
  + Dist[ x[N-1] ][ x[0] ]; // minimize sum of traveled distances.
}
```

#### VRPの例

```
function model() {
  x[1..K] <- list(N); // for each truck, order the clients to visit
  constraint partition( x[1..K] ); // each client is visited once
  distances[k in 1..K] <- sum[i in 1..N-1]( dist( x[k][i-1], x[k][i] ) )
  + dist( x[k][N-1], x[k][0] ); // traveled distance for each truck
  minimize sum[k in 1..K]( distances[k] ); // minimize total traveled distance.
}
```

List 関数の効果を以下に示す。

200 箇所以上を廻る TSP の実例である。



List 関数の導入により、大規模な TSP、VRP 問題を実用的に解くことが可能となった。

LSP 言語は、短いステップでモデリングを行うことができ、モデルのチューニングを行うフェーズではリアルタイムで LSP 言語を変更しながら、試行錯誤を行うことができる。

### 4. 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、実践的な汎用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, 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>