



LocalSolver6.0 リリースノート

Major novelties released in 6.0 version



日本配給元 MSI株式会社

LocalSolver事業部

Mail: localsolver@msi-jp.com

Web: www.msi-jp.com/localsolver/

Who we are



Bouygues社は、フランスに拠点を置く
330億の収益を持つ大企業

<http://www.bouygues.com>

Innovation24

Bouygues社傘下のオペレーションズ・
リサーチの子会社、20年間の実践と研究
開発の実績を持つ

<http://www.innovation24.fr>

LocalSolver

Innovation 24が開発した
次世代-ハイブリット型・数理計画
ソルバー

<http://www.localsolver.com>



LocalSolver

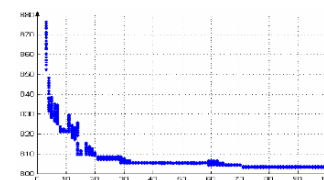
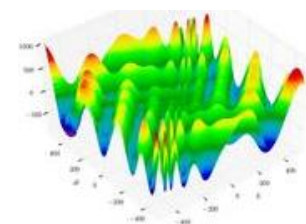
次世代-ハイブリット型・数理計画
ソルバー

組合せ、数值的、
または混合変数 最適化問題を解く

大規模非凸最適化問題に最適

複雑なチューニング作業が不要
最適な解を実用時間で提供(数分間)

LocalSolver
=
LS + CP/SAT + LP/MIP + NLP



LocalSolverお問合せ先
日本配給元MSI LocalSolver事業部
MAIL: localsolver@msi-jp.com

Web: <http://www.msi-jp.com/localsolver/>

Set-based modeling
セットベース・モデリング

経路選択問題 & スケジューリング問題に対する
革新的なモデリング概念



Set-based modeling

構造型意思決定演算子 `list(n)`

- Order a **subset** of values in domain $\{0, \dots, n-1\}$
- Each value is **unique** in the list

“list” で相互作用する基本演算子

- **count**(u): number of values selected in the list
- **at**(u,i) or `u[i]`: value at index i in the list
- **indexOf**(u,v): index of value v in the list
- **contains**(u,v): equivalent to “`indexOf(u,v) != -1`”
- **disjoint**(u1, u2, ..., uk): true if u1, u2, ..., uk are pairwise disjoint
- **partition**(u1, u2, ..., uk): true if u1, u2, ..., uk induce a partition of $\{0, \dots, n-1\}$



Traveling salesman

```
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  
}
```

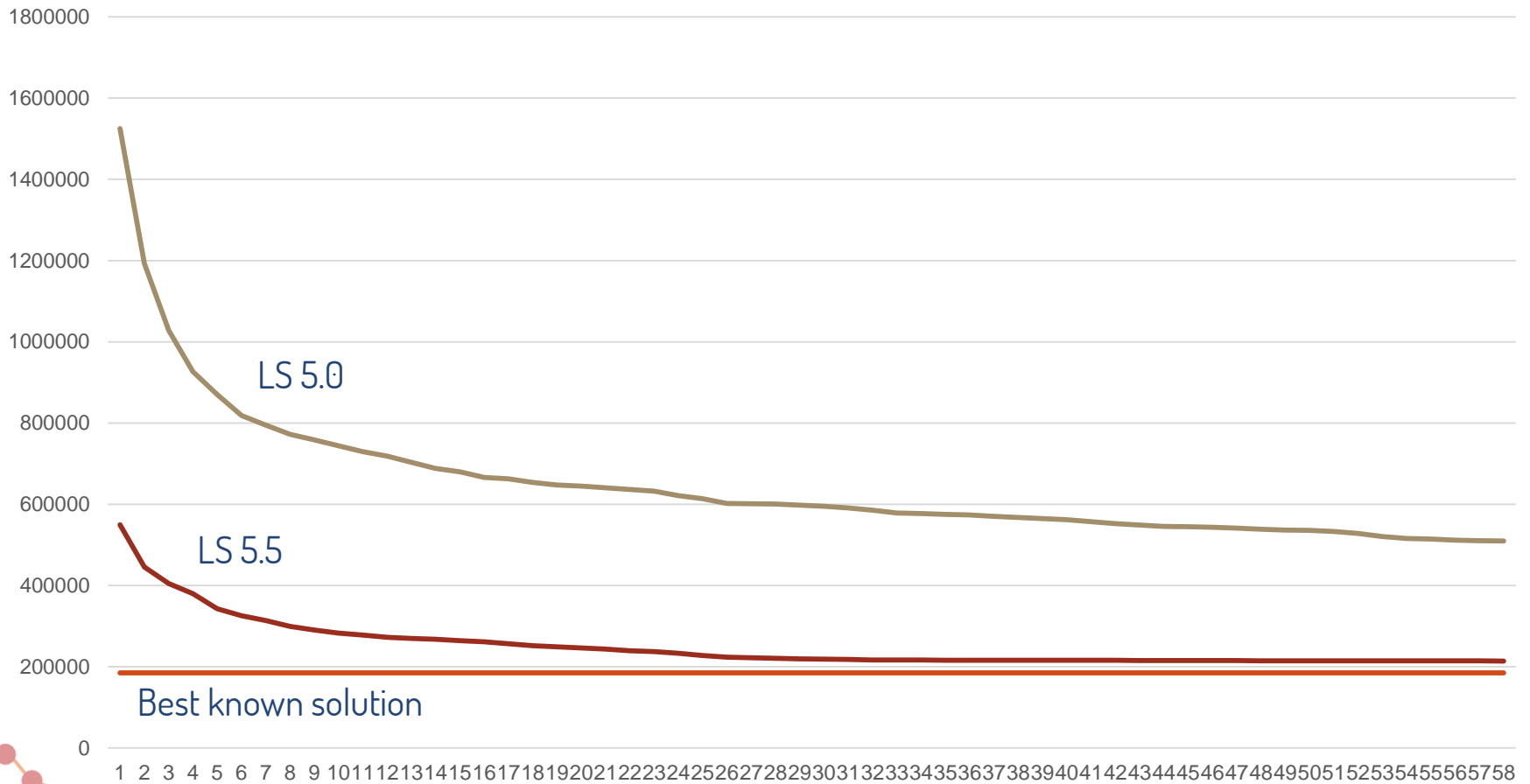
シンプルなモデルとは？

- 自然な形で宣言できるモデル: 直感的に理解できる
- 一般的なset-oriented 指向: 習得しやすい
- 高度なプログラミングスキルが不要
- コンパクト: 入力サイズが線形 → 拡張しやすい(10万ノード)



Traveling salesman

TSP: real-life 200-client instance
LocalSolver 5.0 vs 5.5 (with operator *list*)



Vehicle routing

```
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  
}
```

より一層、容易にするために

- 集合（非順序付け）対 リスト（順序付け）
- 複合集合/リスト:同じ値の複数発生
- 値の代わりにオブジェクト収集
- 反復と 収集に突出した機能（ランダム表現式）



CVRPベンチマーク

CVRP – instances A

- 32-80クライアント、10台のトラック最大化
- 27インスタンス
- 実行時間: 5分
- LS バイナリ: 多くが実行不可能
- **LS list: 1 % avg. opt. gap**

CVRP – instances X100-500

- 100-500 クライアント, 138 台のトラック最大化
- 67インスタンス
- 実行時間: 5分
- LSバイナリ: 多くが実行不可能
- **LS list: 9 % avg. opt. gap**



CVRPTWベンチマーク

CVRPTW – instances Solomon R100

- 101から 112クライアント、19台のトラック最大化
- 13インスタンス
- 実行時間:5分
- LS バイナリ: N/A
- **LS list: 3 % avg. opt. gap**

CVRPTW – instances Solomon R200

- 201から 208 クライアント、4台のトラック最大化
- 8インスタンス
- 実行時間:5分
- LSバイナリ: N/A
- **LS list: 8 % avg. opt. gap**



Black-box optimization

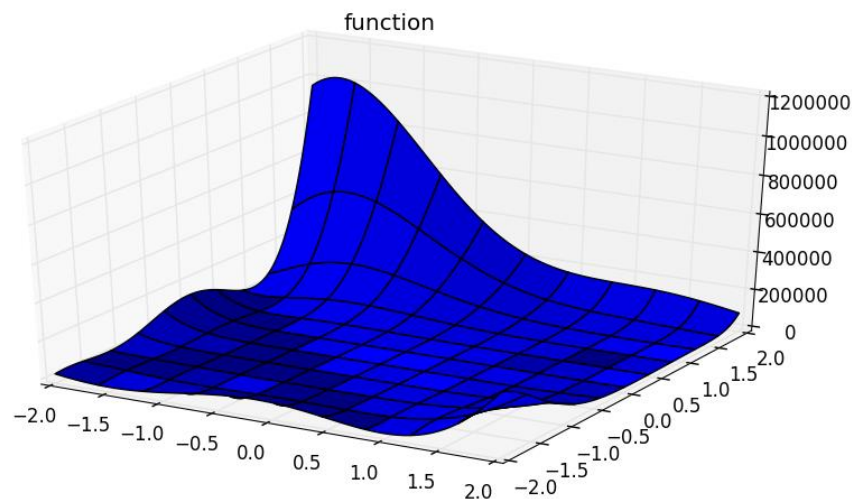
LocalSolverに数值的、
または離散事象シミュレーション



Black-box optimization

Context

- 未知のオブジェクト (oracle)
- 費用関数の評価
- 導関数-無条件
- 連続& 線形意思決定変数
- 意思決定変数の制限



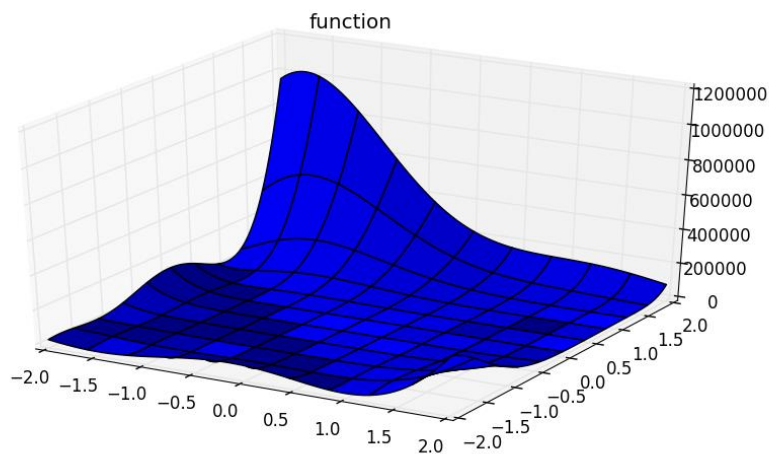
エンジニアリングにおける多数のアプリケーション

- 総合的/パラメトリック最適化
 - シミュレーション最適化 (ノイズ、非決定性)
- 資源のデザイン最適化/システム:メカニズム、電力、ロジスティックス, etc.

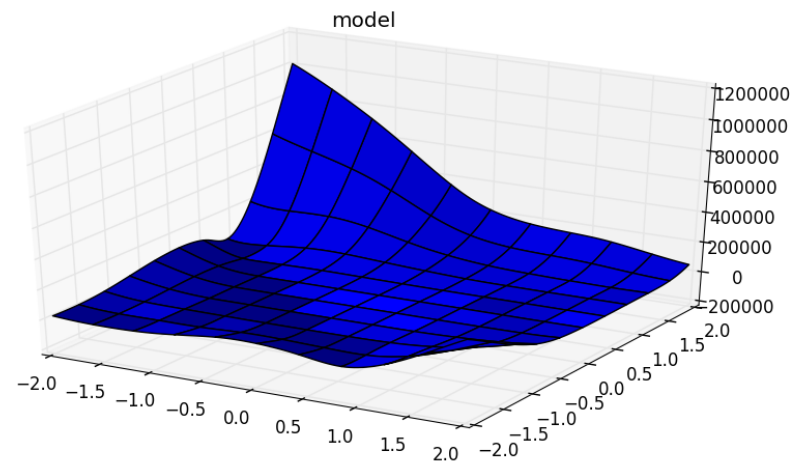


目的関数のランドスケープについて

- 放射基底関数(RBF)でモデル化されたオブジェクト・ランドスケープ
- 複数のモデルは異なる技術/パラメータで構築
- 最適化のために最も期待できるモデルを自動選択



Objective Function



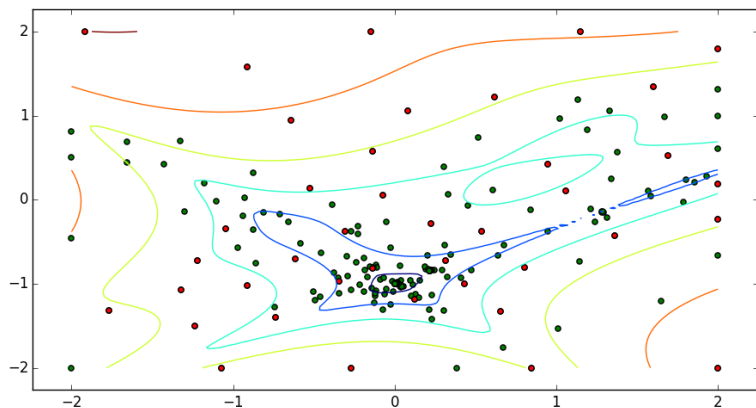
Objective Model



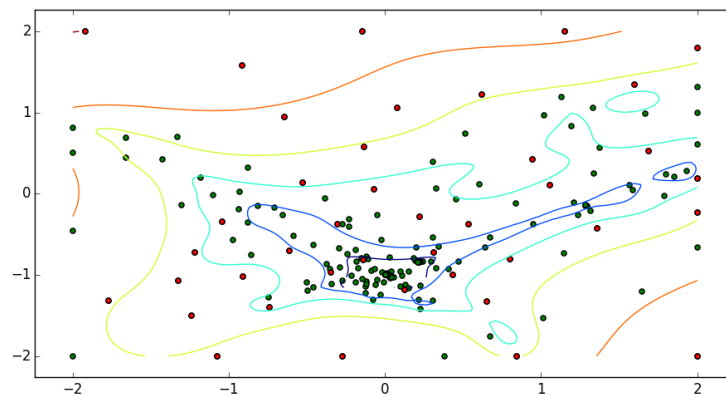
Optimization

開発& 多様化

- 開発: オブジェクトモデル上で最適化
 - 多様化: 新規の期待域の探索
- LocalSolverの技術でNLP部分問題を解く
local & direct 探索、勾配ベースライン探索、etc.



目的関数



オブジェクトモデル



Benchmark

Instances <例題>

- Costa氏 と Nannicini氏による最新論文からの25例題
RBFOpt: 費用関数評価を使ったBlack-Box最適化に関するオープンソース・ライブラリ。最適化オンライン (レビュー下)
- 各例題あたり20回の実行・1回の実行でBlack-boxに150回の maxを呼び出し
- 数値精度: $1e-6$

Preliminary results <中間結果>

- RBFOpt: 345 opt. solutions found, 82 calls avg. per run
- **LocalSolver: 310 opt. solutions found, 94 calls avg. per run**
- NOMAD (GERAD): 170 opt. solutions found



Benchmark

Instance	LocalSolver			RBFOpt			NOMAD	
	#sol	Avg. Eval	Error (%)	#sol	Avg. Eval	Error (%)	#sol	Error (%)
branin	20	23	0,0	20	31	0,0	20	0,0
camel	20	26	0,0	20	34	0,0	19	4,0
ex_4_1_1	20	11	0,0	20	14	0,0	20	0,0
ex_4_1_2	20	51	0,0	20	9	0,0	20	0,0
ex_8_1_1	20	10	0,0	20	7	0,0	19	2,5
ex_8_1_4	20	44	0,0	20	25	0,0	0	341,5
gear	20	34	0,0	20	7	0,0	0	388,0
goldsteinprice	18	122	0,1	20	53	0,0	16	450,0
hartman3	8	130	1,2	20	45	0,0	15	9,4
hartman6	8	121	11,0	17	101	5,1	0	5,7
least	0	150	1308,0	0	150	204,7	0	129,0
nvs04	20	70	0,0	19	64	194,4	4	9997,0
nvs06	16	127	1,0	0	150	13,3	9	8,7
nvs09	20	15	0,0	20	14	0,0	16	1,2
nvs16	8	138	949,0	20	49	0,0	9	885,0
perm0_8	0	150	109,0	0	150	147,2	0	412,0
perm_6	0	150	2424958,0	0	150	44134,7	0	311032,0
rbrock	20	83	0,0	5	136	10,8	0	43,2
schoen_10_1	4	145	66,7	11	139	28,8	0	119,5
schoen_10_2	0	150	96,2	14	133	1,6	0	115,7
schoen_6_1	18	101	100,8	18	101	1,8	0	51,5
schoen_6_2	10	120	28,0	16	102	32,7	0	54,2
shekel10	8	118	29,6	13	107	60,1	0	56,9
shekel5	6	127	51,6	7	126	51,7	1	46,1
shekel7	6	127	28,5	5	137	47,0	2	47,9
	310			345			170	



John N. Hooker (2007)

“Good and Bad Futures for Constraint Programming (and Operations Research)”
Constraint Programming Letters 1, pp. 21–32

“Since modeling is the master and computation the servant, no computational method should presume to have its own solver.

This means there should be no CP solvers, no MIP solvers, and no SAT solvers. All of these techniques should be available in a single system to solve the model at hand.

They should seamlessly combine to exploit problem structure. Exact methods should evolve gracefully into inexact and heuristic methods as the problem scales up.”



LocalSolver

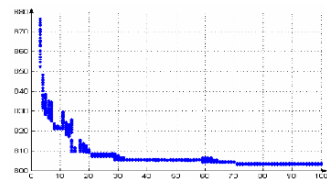
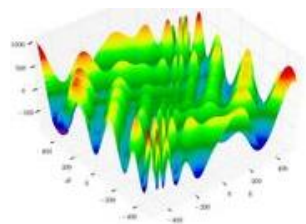
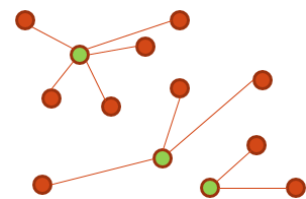
ハイブリット型数理最適化ソルバー

組合せ、数值的、
または混合変数最適化問題を解く

大規模非凸最適化問題に最適

複雑なチューニング作業が不要
実用時間で高品質な解を提供(数秒)

LocalSolver
=
LS + CP/SAT + LP/MIP + NLP



LocalSolverお問合せ先
日本配給元MSI LocalSolver事業部
MAIL: localsolver@msi-jp.com

Web: <http://www.msi-jp.com/localsolver/>