

LocalSolver

Large-scale food supply chain optimization at Pasco Shikishima

We developed the Pasco optimization model using LocalSolver within a few days of work. LocalSolver provides outstanding solutions, as considered by Pasco planners, in just a few minutes of running times, while the number of variables is gigantic (tens of millions). We could not imagine that such a feat was possible at first, since state-of-the-art MIP solvers like CPLEX, Xpress and Gurobi were unable to tackle the problem in hours. Now we know that using LocalSolver, it's possible!

Mr. Shinichi Kuroda,
Project director

All-terrain & all-in-one optimization solver

LocalSolver は、自然な形の数理的制約条件式を使用して最適化モデルを宣言するだけで、数秒から数分で高品質な解を導きます。独自の革新的なハイブリッド型技術をベースとし、スタンダード・コンピュータ上で実行する場合でも、最大で数百万変数まで変数をスケールアップすることが可能です。LocalSolver は迅速なプロトタイプ向けに高水準数理モデリング言語や様々なプラットフォームでの開発を容易にする軽量オブジェクト指向型 APIs を提供しています。

LocalSolver hybrid techniques

Real-life
Large-Scale
Optimization
Problem

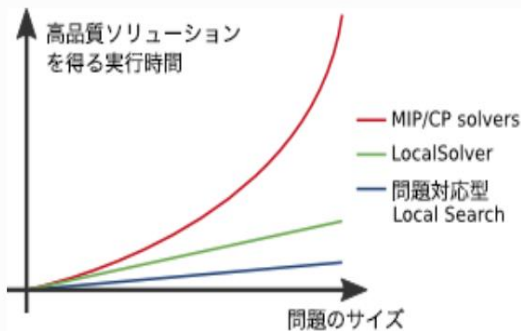


High-quality
solutions
in short
running time

Innovative modeling language (LSP)

- 離散素、ニューメリック、ブラックボックス問題に適用
- Python、C++、Java、C#で利用しやすい APIs
- 数百万変数を持つ問題を迅速に解く
- 手軽にできるライセンス取得/設定
- 高水準数理モデリング言語
- 専門家による迅速なサポート
- 革新的なハイブリッド型技術

経路問題 & スケジュール問題に適した LocalSolver 独自の Set-based モデリング機能



```
// Capacited Vehicle Routing Problem (CVRP)
function model() {
  // for each vehicle, the list of visited clients
  routes[1..nbTrucks] <- list(nbCustomers);
  constraint partition [k in 1..nbTrucks](routes[k]);

  for[k in 1..nbTrucks] {
    route <- routes[k];
    c <- count(route);
    // truck capacity constraint
    constraint sum(0..n-1, i => demands[route[i]]) <= capacity;
    // sum of distances between each pair of clients and depot
    distances[k] <- sum(0..n-2, i => distance(route[i], route[i+1]))
      + distance(depot, route[0]) + distance(route[n-1], depot);
  }
  minimize sum[k in 1..nbTrucks](distance[k]);
}
```

LocalSolver に関するお問合せ

MSI 株式会社(日本配給元)

〒261-7102

千葉市美浜区中瀬 2-6 WBG2 階

Tel:043-297-8841 fax:043-297-8836

Email:localsolver@msi-jp.com

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

Unit commitment
Supply chain Network optimization Car sequencing
Nurse rostering Parametric optimization
Facility location Group planning
Logistic clustering Workforce scheduling
Machine scheduling Media planning
Optimization Portfolio optimization
Frequency assignment

Why LocalSolver?

LocalSolver は様々な最適化テクニックを組み込んでいます

革新的ハイブリッド型技術により数百万変数まで問題をスケールアップ可能！ 実用時間内に高品質な解を提供！

All-terrain & all-in-one

LocalSolver は線形制約、非線形制約および代数的、論理、集合を含む目的関数や連続変数、離散変数を持つブラック・ボックスな表現式などのモデルを解くことが可能です。LocalSolver は何百万変数を持つ問題に対し、数秒から数分で高品質な解を提供します。解はダイレクトに得ることができ、解法に関して複雑なチューニングを行う必要はありません。またライセンス取得やライセンス設定も従来製品より（当社製品比）簡単かつ迅速に行えるようになりました。

Innovative math modeling language

LocalSolver は強力なモデリングおよびスクリプト言語を提供します。この機能により迅速にプロトタイプ最適化アプリケーションの開発を行うことが可能です。

LocalSolver の独自のプログラミング言語である LSP 言語は効率的なプログラミング形式を提供します：

ダイナミックかつ暗黙的な変数宣言、コンパクトなシンタックスのループ宣言等。最新の関数型言語方式を取り込んでいます。

LocalSolver チームが目指しているのは、可能な限りプログラミングにかかる時間や労力を削減し、プロトタイプ開発の枠組みを提供できる最適化ソルバーの実現です。LSP モデルを使用したモデルは既存のモデリング言語を使用して記述したモデルと比較し、モデル全体が短縮され直観的かつ理解しやすいモデル構造です。

Easy object-oriented APIs

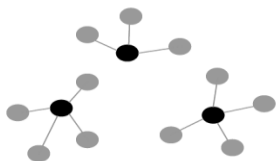
ユーザのビジネス・アプリケーションシステムに LocalSolver を組み込むために、Python、Java、C#、および C++ 向けに可能な限りクラス数を少なくしたオブジェクト指向型プログラミングインターフェース (APIs) を提供しています。LocalSolver が提供する APIs を使用することで LSP を簡単にオブジェクト指向言語に移行することが可能です。ユーザは数理最適化モデルの作成に専念するだけでよく、問題を分解する必要も、ソルバーのチューニング変更も必要ありません。

Reliable, robust and fully portable

信頼性、ロバスト性がなければ効率性は意味を持ちません。ドラスティックかつ一貫した統合メソッドロジーによりクライアントへ最高品質の製品を提供するだけでなく、最適化の専門家による迅速かつ親切、丁寧なサポートも保証しています。LocalSolver は完全ポータブル環境で提供しています。

Windows、Linux、MacOS、Solaris (各 32 ビット/64 ビット) でご利用頂けます。

Some examples



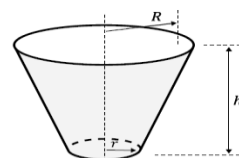
Facility location

```
// Select a subset P among N points minimizing the sum of distances
// to each point from N to the nearest point in P.
function model() {
  // decisions: point i is in P iff x[i] = 1
  x[1..N] <- bool();

  // Select P points among N
  constraint sum[i in 1..N](x[i]) == P;
  minDist[i in 1..N] <- min[j in 1..N](x[j] ? Dist[i][j] : +inf);
  minimize sum[i in 1..N](minDist[i]);
}
```

Parametric optimization

$$V = \frac{\pi h}{3}(R^2 + Rr + r^2)$$
$$S = \pi r^2 + \pi(R+r)\sqrt{(R-r)^2 + h^2}$$



```
// Maximize the volume of a bucket with a given surface of metal
function model() {
  R <- float(0,1);
  r <- float(0,1);
  h <- float(0,1);
  V <- PI * h/3.0 * (R * R + R * r + r * r);
  S <- PI * r * r + PI * (R + r) * sqrt(pow(R-r,2) + h * h);
  constraint S <= SMAX;
  maximize V;
}
```

Some clients

