

LocalSolver トレーニングセッションの主な技術トピック

1. 意思決定変数 VS 中間式-- Decision variables VS intermediary expressions

MILP モデリング出身のエンジニアと議論する場合、この点が重要になることが多いです。

MILP ソルバーとは異なり、LocalSolver には「中間変数」のようなものはありません。

意思決定変数と中間式という 2 つの重要な概念を区別しています。式は、意思決定変数から推論することができます。

この点を説明するために、私はよく次のような例を使います。

MILP モデリングでは、Mc Cormick の不等式を使用して、2 つの変数の積を表現します。

ただし、LocalSolver では、単に積を直接 $x * y$ と書けばよいのです。(追加の意思決定を導入する必要はありません)。

より完全な例を挙げると、私はよく施設配置問題

(<https://www.localsolver.com/docs/last/exampletour/facilitylocation.html>) をよく使用します。

この場合、LocalSolver モデルには、開始位置の変数 (opening location variable) だけを必要とします。オープンな位置から推測される割り当てのための変数は必要ありません。

2. 非線形性の利用- Using non linearities

MILP ソルバーとは異なり、LocalSolver は非線形演算子の非常に広く使用することができます。

非線形モデルのほとんどを単純化することができます。2 つの良い例として、

バケット問題の最適化

(<https://www.localsolver.com/docs/last/exampletour/optimalbucket.html>) と

MAX カット問題です。

(<https://www.localsolver.com/docs/last/exampletour/maxcut.html>) 。

3. Set および list 変数

Set と list もまた、非常に差別化された強力な機能があり、モデルを簡潔かつスケーラブルにし、非常に効率的に解けます。

私の好きな例としては、ビン詰め問題

(<https://www.localsolver.com/docs/last/exampletour/binpacking.html>) と

TSP 問題 (<https://www.localsolver.com/docs/last/exampletour / tsp.html>) です。

もし、もっと複雑なモデルを示したい場合は、CVRP

(<https://www.localsolver.com/docs/last/exampletour/vrp.html>) を使用できます。

多くの場合、Set と list は、初めて見たときに同化することが困難です。

通常、私がそれらを完全に紹介するには、多くの時間がかかるので、ご不明な点があればいつでもお問合せください。

4. 外部およびブラックボックス関数-- External and blackbox functions

最後になりましたが、LocalSolver では、最適化プログラム内で任意のカスタムコードを使用することができます。

これは、外部関数（モデルに直接書き込むことが困難な高速計算に使用）またはブラックボックス関数（長時間の難しい計算に使用される、通常はシミュレーター用）として使用することができます。

詳細は

<https://www.localsolver.com/docs/last/modelingfeatures/externalfunctions.html>

をご覧ください。

ブラックボックス関数の詳細については、「サロゲートモデリング」セクションを参照してください。

私はよくこの機能については、

(<https://www.localsolver.com/docs/last/exampletour/hosaki.html>) で説明することができます。

また、ガイドライン (<https://www.localsolver.com/docs/last/guidelines/index.html>) も非常に良い資料です。

その他、必要な情報がありましたらいつでもご連絡ください。

--英語原文--

1. Decision variables VS intermediary expressions

> This point is often crucial when talking to engineers coming from MILP modelling. Contrarily to MILP solvers, there is no such thing as “intermediary variable” in LocalSolver. We make the difference between two key concepts: decision variables and (intermediary) expressions. Expressions can be deduced from decision variables. To illustrate this point, I often use the example of a simple boolean product. In MILP modelling, we use the Mc Cormick inequalities to represent the product of two variables. But in LocalSolver, you simply write the product directly $x*y$ (no need to introduce any additional decision). To give a more complete example, I often use the facility location problem (<https://www.localsolver.com/docs/last/exampletour/facilitylocation.html>). The LocalSolver model only needs the opening location variables. It does not need the assignment variables which are deduced from the opened locations.

2. Using non linearities

Again, contrarily to MILP solvers LocalSolver allows a very wide use of nonlinear operator, simplifying most of nonlinear models. Two good examples are the optimal bucket problem (<https://www.localsolver.com/docs/last/exampletour/optimalbucket.html>) and the max cut problem (<https://www.localsolver.com/docs/last/exampletour/maxcut.html>).

3. Sets and lists variables

Sets and lists are also extremely differentiating and powerful features that allows the model to be concise, scalable and very efficient. My favourite examples to start with are the bin packing problem (<https://www.localsolver.com/docs/last/exampletour/binpacking.html>) and the TSP (<https://www.localsolver.com/docs/last/exampletour/tsp.html>). If you want to show more complex models, you can use the CVRP (<https://www.localsolver.com/docs/last/exampletour/vrp.html>). More often than not, sets and lists are hard to assimilate when seen for the first time. Usually, I take a lot of time to introduce them completely so the attendants have enough time to assimilate these new decision variables.

>

4. External and blackbox functions

Last but not least, LocalSolver allows to use any custom code to be used in the optimization program. It can be external functions (used for quick calculations hard to write directly in the model) or blackbox function (used for long and hard calculations, usually simulators). You can take a look at <https://www.localsolver.com/docs/last/modelingfeatures/externalfunctions.html>. The

blackbox feature is detailed in the “surrogate modelling” section. I often illustrate this feature with the simple Hosaki function (<https://www.localsolver.com/docs/last/example/last/hosaki.html>).

The guidelines (<https://www.localsolver.com/docs/last/guidelines/index.html>) are also very good material to rely on.