

2024年

# DXを進化させる Hexaly 最新版

宮崎 知明 (MSI株式会社)

[miyazaki@msi-jp.com](mailto:miyazaki@msi-jp.com)

- 1 はじめに
- 2 Hexalyが実現したDXの事例
- 3 これからの最適化
- 4 Hexalyの特徴機能
- 5 Hexaly最新版の新機能
- 6 おわりに

1

はじめに

Hexaly 最新版は、従来の最適化システムを包含した **All-In-One Solver** である。LP、MIP、NLP、CP等を包含し、同一形式の柔軟な定式化による最適化機能を備え、大規模な問題を高速に解くことが可能である Hexalyが実現しているDXの事例から Hexaly の特徴を示す。

- 第4世代のモデリング言語による **自然なモデリングを実現**  
(MIP、NLP、CP等の問題を同一形式で自由に定義可能)
- **高速な最適化計算**  
(従来解けなかった大規模な問題を高速に解くことが可能)

# 今までのOR手法（最適化）ソフトの使われ方

今までは、ハードウェア、ソフトウェアの進化にそって理論を進化させ、実用的に使えるソフトを開発してきた。

適用分野は以下であり、それぞれで進化

- プランニング（計画立案）⇒ LP、MIP、NLP等
- スケジューリング（実行計画）⇒ 組合せ最適化等
- オペレーション（実行処理）⇒ 人間、AI

# 最適化システムの歴史

- **従来型の数理計画法システム**
  - ✓ 理論的に最適解を求めるアプローチ
  - ✓ 大規模問題になると現実的な時間では解けない
  - ✓ 非線形問題に弱い
- **ヒューリスティックプログラムの出現**
  - ✓ 人間の行動を計算機で実現
  - ✓ AIとして発展（一種の自動化、最適化を実現）
- **メタヒューリスティック解法の出現**
  - ✓ ヒューリスティックを汎用的な解法に
  - ✓ 問題対応型で様々な解法が出現（Greedy algorithm、Local search、GA・・・）
  - ✓ 実用的な最適解を試行

# 今なぜ、Hexaly か

- 従来手法では解くことができなかった大規模な最適化問題、大規模なスケジューリング最適化問題を実用時間で解くことができる
- 世界中のDX推進に利用されており、従来では実現できなかった問題に活用されている

特に、スケジューリングを含む現実的な最適化処理ができる時代になっている

# DX推進に求められるOR手法とは

- 全社計画に必要な広範囲な問題の最適化  
→ 大規模（変数、制約条件、複数目的）
- 実行可能なスケジューリング（複数工程等）  
→ 大規模組合せ、複雑な条件等
- オペレーション（実行指示、運用）  
→ 人間中心の運用管理システム（AI等）

将来的には、上記3つの融合が理想的である

# 次世代の最適化を実現した Hexaly

- グローバル探索を実現した大規模問題用の**汎用数理計画法システム (All-In-One Solver)**
- 様々な解法（メタ解法技術、既存解法技術）**を融合**
- 従来の解法依存型での定式化は必要なく、非線形問題をも含む**汎用問題として定式化が可能**  
（例として：罰金関数を使用する必要はない）
- 誰でも容易に問題定義ができ、使えることを実現

2

# Hexaly が実現したDXの事例

## 事例1 パスコ敷島のグローバルサプライチェーン最適化問題 (1/3)

パスコのサプライチェーンには、日本国内に13の工場があり、それぞれに複数の生産ライン、100以上の供給拠点、26の配送ハブがある。

パスコの工場では毎日90万件の注文を生産しなければならず、注文ごとに、パスコはいつどこで何を生産するかを決める必要があるとともに、パスコが原材料を調達する場所と、配送センターを配送するルートを決定する必要がある。

目標は、生産と流通の能力を尊重しながら、数日間にわたった生産と流通コストを最小限に抑える計画を立案することにある。

## 事例1 パスコ敷島のグローバルサプライチェーン最適化問題 (2/3)

### ・入力データ

- ✓1,000カタログ製品
- ✓1日あたり900,000件の生産注文
- ✓100の供給サイト
- ✓13の生産拠点
- ✓26の配送サイト

### ・決定事項

- ✓各注文をいつどこで生産するかを決める
- ✓原材料の調達先を決める
- ✓配送センターと配送センターから配送するルートを決める

## 事例1 パスコ敷島のグローバルサプライチェーン最適化問題 (3/3)

### • 制約条件

- ✓生産能力
- ✓生産時間
- ✓配電容量配布時間

### • 目標

- ✓生産コストと流通コストの最小化
- ✓原材料の使用効率の最大化
- ✓仕掛品（WIP）の最小化

### 定式化

- 3,200万個の変数
- 100万の制約
- 800万のバイナリ変数
- 16の目標

## 事例1 顧客の声

このようなモデルを、入出力処理時間を含め、基本サーバー上で3分の実行時間で解決している。

従来のMIPソルバーでは、元の問題を20のサブ問題に分解して別々に解く必要があり、その場合には、元のグローバル問題に対して低品質の解がもたらされた。

Hexalyは分解することなく、直接一つの問題としてそれに取り組むことができた。

## 事例2 ルノーのワークスペース割当て問題 (1/3)

建物内のスタッフの組織を最適化するために、ルノーチームは2つの段階を経ている。

最初のステップは**マクロ・ゾーニング**である。  
これは、Renaultのさまざまな部門にワークスペースを割り当てることで構成されている。

次ステップは、デスクを人々に割り当てることであり、これは**マイクロゾーニング**と呼ばれている。

## 事例2 ルノーのワークスペース割当て問題 (2/3)

ルノーグループが解決する最適化モデルの概要は以下の通りである。

### • 入力データ

- ✓5 部門
- ✓27 ワークスペース
- ✓1,109名

### • 決定事項

- ✓部門へのワークスペースの割り当て
- ✓デスクをユーザに割り当てる

## 事例2 ルノーのワークスペース割当て問題 (3/3)

### ・制約

- ✓モジュールの容量を尊重する
- ✓すべてのスタッフメンバーを割り当てる
- ✓部門用に一部のモジュールを予約（特定のモジュール）
- ✓高度な機密性を必要とする部門は、モジュールを他の部門と共有しない

### ・目標

- ✓異なるモジュールの充填率の平滑化
  - 同じモジュール・または隣接モジュール内の同じ部門のメンバーをグループ化する
  - 密接に連携する2つの部門からの同じモジュール（または近隣のモジュール）サービスにグループ化する
  - モジュールにできるだけ多くの部門を割り当てる

### なぜ Hexaly なのか？

ワークスペース割り当ての問題は、マッチング制約とパッキング制約が結合し、最適化する複数の非線形目標があるため、解決が困難であった。

たとえば、異なるモジュールの充填率を平滑化すると、2次式が誘導されるが、その非線形性があるにもかかわらず、Hexaly は1分未満のランタイムでほぼ最適なソリューションを提供した。

3

# これからの最適化

# これからの最適化

コンピュータの飛躍的な性能向上により、リアルタイムで意思決定、判断をすることが可能になり、空前のAIブームを迎えようとしている。

AI、IoT、Big Dataのキーワードのもと、最適化機能を追加することで、必要な情報を必要なタイミングで得ることができ、あらゆる分野でさらなる情報活用が可能となりつつある。

AIによる自動化は右のように大別でき、最適化計算を入れることでさらなる発展が期待できる。

- **センサと連動した自動運転制御**  
(自動車の安全装置、プラント自動制御など)
- **大量の実績データを統計処理し、統計確率にもとづいたオペレーションの自動化**  
(事務処理の代替など)
- **人間の思考パターンの高速シミュレーション**  
(ゲーム、ディープラーニング等)

# 最適化手法と融合した次世代AI

現状のAIは統計予測をベースとした統計理論による確率論的なものや、センサ等による情報に対する対処方法がきまっていることが基本であり、意思決定理論が確定していることが重要である。

これに対して、**新しい事象でも対応できるようにする（最適解を導く）**ことが次世代AIである。

一言で言えば、**意思決定モデルを作成し、最適化手法を組み込むことにより、実績にない事象に対しても、さらなる効率化、自動化を可能することである。**

# 高速な最適化計算

すべてのモデルで、より優れたソリューションの高速化とより優れた境界計算（上界値、下界値）の高速化を実現。  
解変数の交換による掃き出し計算。

特に、**パッキング、ルーティング、およびスケジューリング問題**などの**大規模組み合わせ最適化問題**、および非線形連続問題のパフォーマンスが大幅に向上。

例としては、時間枠付き容量制限車両ルーティング問題（CVRPTW）および時間枠付き集配問題（PDPTW）問題。

# MIPLIBモデルのベンチマーク結果

| instances       | Status | Variables | Hexaly3.1     | Gurobi 5.5  | Cplex 12.4  | Optimum |
|-----------------|--------|-----------|---------------|-------------|-------------|---------|
| opm2-z10-s2     | hard   | 6,250     | * -25,719     | -19,601     | -18,539     | -33,826 |
| opm2-z11-s8     | hard   | 8,019     | * -33,028     | -21,661     | -18,883     | -43,485 |
| opm2-z12-s14    | hard   | 10,800    | * -46,957     | -11,994     | -36,469     | -64,291 |
| opm2-z12-s7     | hard   | 10,800    | * -46,034     | -12,375     | -30,887     | -65,514 |
| pb6             | hard   | 462       | -62           | -62         | -62         | -63     |
| queens-30       | hard   | 900       | -38           | -36         | -39         | -40     |
| dc11            | open   | 37,297    | 11,100,000    | 21,300,000  | 1,840,402   | unknown |
| do-big          | open   | 6,020     | 9,844         | 62,520      | 5,256       | unknown |
| ex1010-pi       | open   | 25,200    | 249           | 251         | 247         | unknown |
| ivu06-big       | open   | 1,812,044 | * 479         | 9,416       | 678         | unknown |
| ivu52           | open   | 1,423,438 | 4,907         | 16,880      | 3,285       | unknown |
| mining          | open   | 753,404   | * -65,720,600 | 902,969,000 | no solution | unknown |
| pb-simp-nonunif | open   | 23,848    | * 90          | 140         | 94          | unknown |
| ramos3          | open   | 2,187     | * 223         | 274         | 267         | unknown |
| rmine14         | open   | 32,205    | * -3469       | -170        | -968        | unknown |
| rmine21         | open   | 162,547   | * -3657       | -184        | no solution | unknown |
| rmine25         | open   | 326,599   | * -3052       | -161        | no solution | unknown |
| sienal          | open   | 13,741    | 256,620,000   | 315,186,152 | 54,820,419  | unknown |
| sts405          | open   | 405       | 342           | 342         | 354         | unknown |
| sts729          | open   | 709       | 648           | 648         | 665         | unknown |

- 最少化問題（Hexaly はMIIP変数を減らす定式化をおこなっている）
- 実行CPU時間：5分
- PC：Intel Core i7-820QM（4cores, 1.73GHz, 6GB RAM, 8MB cache）

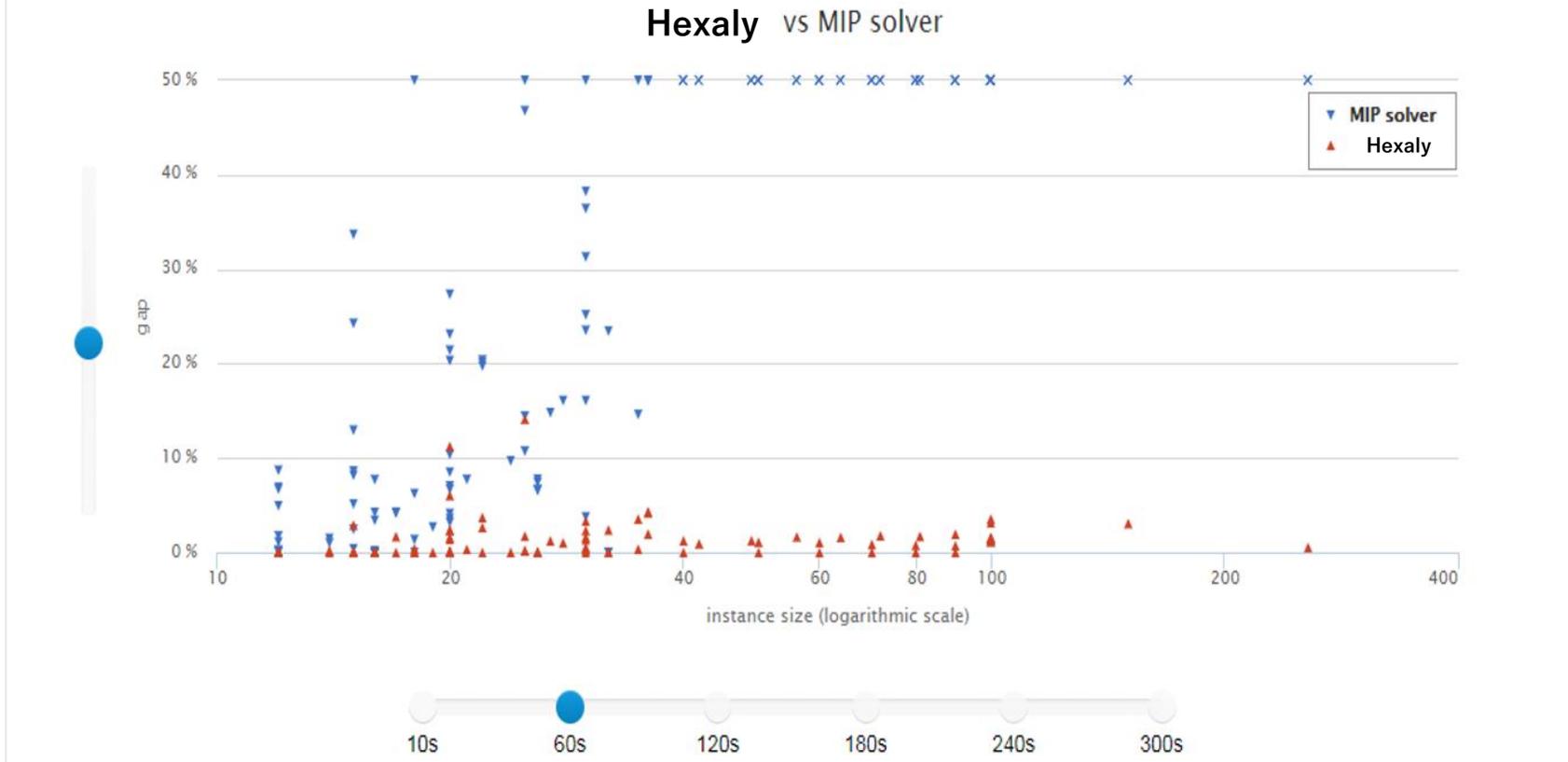
# MIPモデルの変換

- **意志決定変数にの再定義**  
意思決定変数で目的関数、すべての制約関数の値を決めることが肝要
  - ※MIPモデルで整数変数セットが独立してはN.G.
  - ※探索は意志決定変数の組合せで行われる
- **制約式、目的関数の直接定義**  
意志決定変数を使って制約式、目的関数を自由に表現する。
  - ※式の表現に、算術演算子、論理演算子等が使用できる
  - ※罰金関数等で表現する必要はない
- **既存のMIP生成プログラムからの移行**  
意思決定変数を $X_{i,j,k,l,\dots}$ と再定義することで、可能。
  - ※いつ、どこで、なにを、どうするよな5W1Hの変数でMIPを再定義するだけでよい

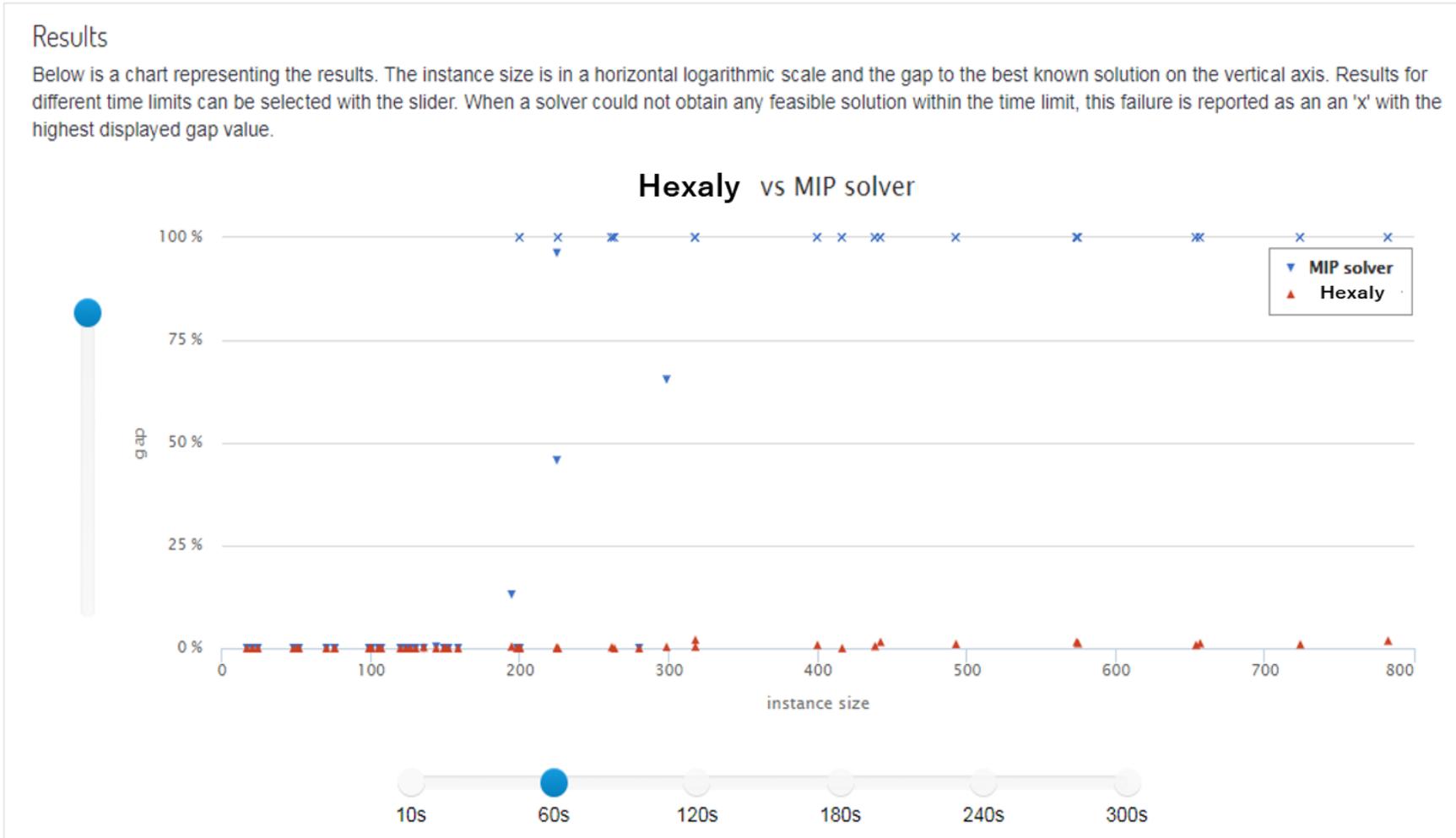
# ベンチマーク結果 (QAP問題)

## Results

The chart below reports the results with the instance size on a horizontal logarithmic scale and the gap to the known optimal solution on the vertical axis. When a solver could not reach a feasible solution within the time limit, it is plotted as a 'X'. You can use the slider to select different time limits and update the chart accordingly.



# ベンチマーク結果 (巡回セールスマン問題)



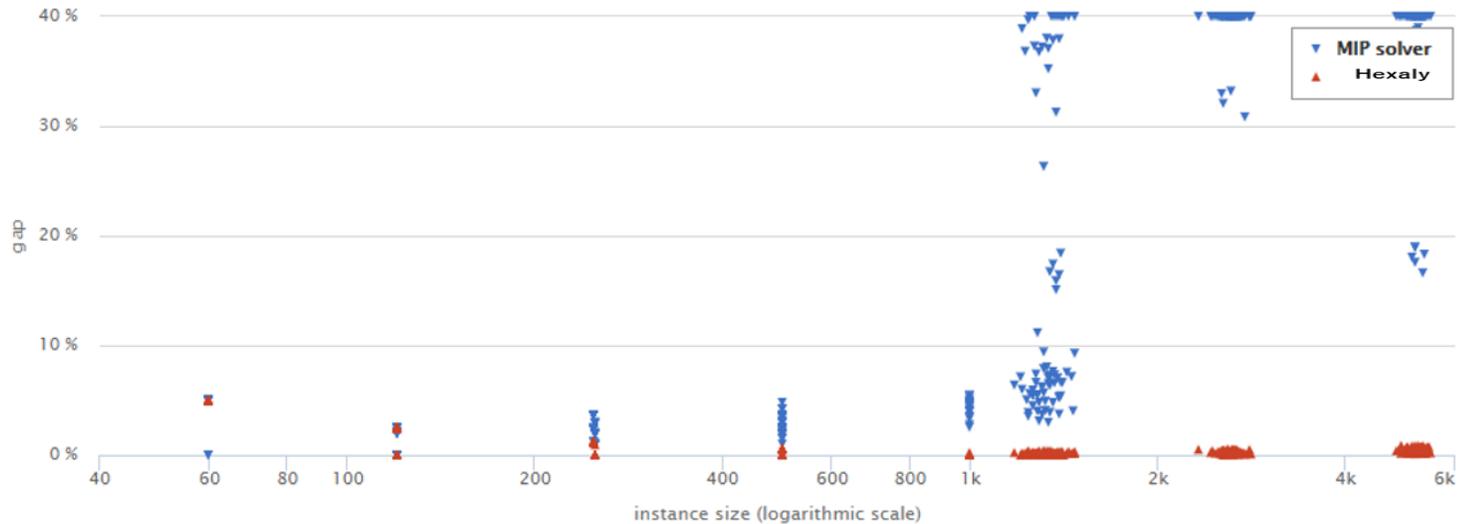
# ベンチマーク結果（ビンパッキング問題）

## Performance

We have measured the gap value regularly until 300 seconds. The comparison is made between LocalSolver 8.5 and a commercial MIP solver, both without any specific configuration or tuning. The BPP Library has more than 6000 bin packing instances, with 50 to more than 5000 items. We used the Falkenauer and GI datasets (160 and 240 instances respectively) which cover the whole range of instance sizes.

Below is a chart representing the results. The instance size is on the horizontal axis and the gap to the best known solution on the vertical axis. Results for different time limits can be selected with the slider.

### Hexaly vs MIP solver



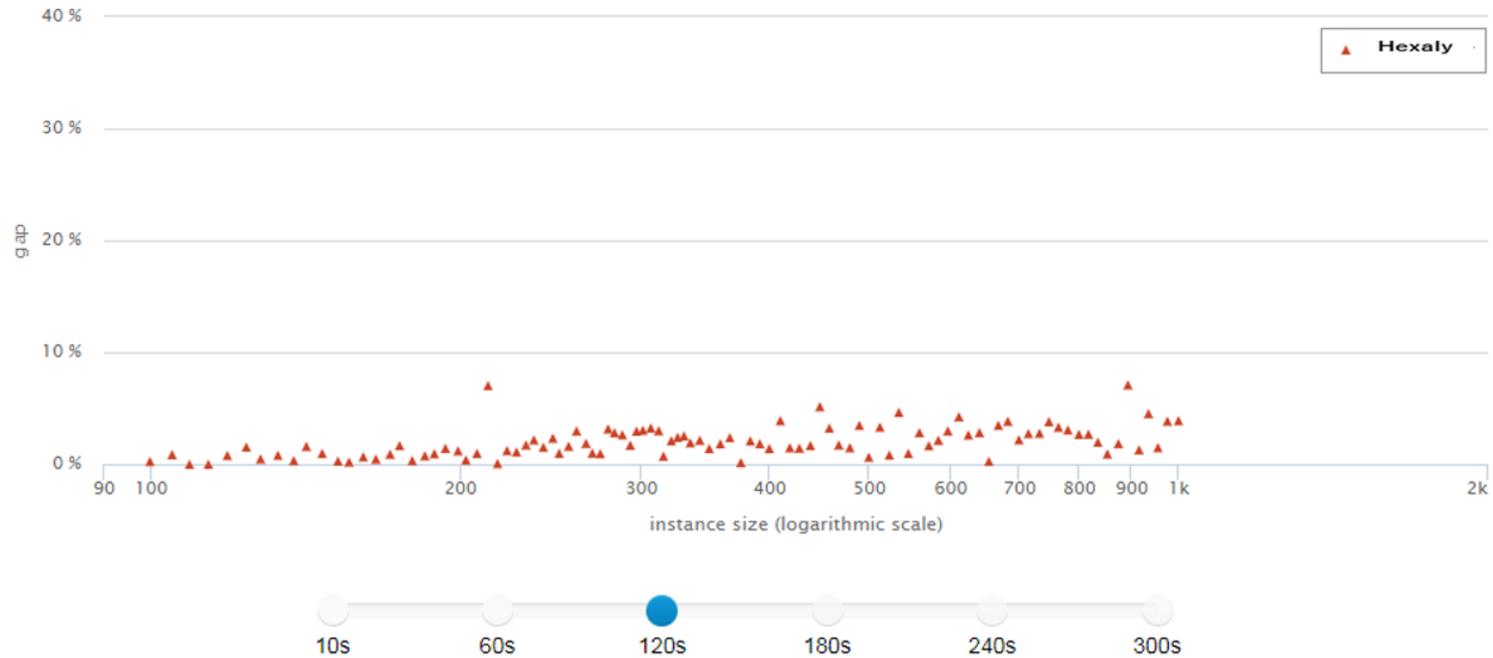
# ベンチマーク結果（車両ルーティング(CVRP)）

## Performance

The chart below presents the results obtained with this LocalSolver model with different running times (from 10 seconds to 5 minutes) on all instances. In summary, we observe that good solutions are obtained even with short running times, and given 5 minutes of computation time, LocalSolver reaches a average gap smaller than 2%.

## Curve

### Hexaly results



4

# Hexaly の特徴機能

# 1) 自然なモデリング方法論

- 従来のMILPソルバーとは異なり、Hexalyには「中間変数」というものは存在せず、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おり、意思決定変数で目的関数とすべての制約式を定義すればよい。
- 式は決定変数から推論することができ、線形不等式（従来のLP、MIP形式データ）にとらわれることなく自然な定式化が可能である。
- 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するが、LocalSolverでは、単に積を**直接 $x*y$** と書けばよく、追加の決定式を導入する必要はない。

# 例題のステップ数比較

|                          | lsp steps Python |            | cpp steps  |       |         |         |       |
|--------------------------|------------------|------------|------------|-------|---------|---------|-------|
|                          | steps            | Python/lsp | lsp/Python | cpp   | cpp/lsp | lsp/cpp |       |
| assem bly_line_balancing | 82               | 128        | 1.561      | 0.641 | 173     | 2.110   | 0.474 |
| car_sequencing           | 75               | 112        | 1.493      | 0.670 | 194     | 2.587   | 0.387 |
| cvrptw                   | 182              | 246        | 1.352      | 0.740 | 292     | 1.604   | 0.623 |
| jobshop                  | 94               | 116        | 1.234      | 0.810 | 199     | 2.117   | 0.472 |
| m ovie_shoot_schedu ling | 136              | 158        | 1.162      | 0.861 | 267     | 1.963   | 0.509 |
| pdptw                    | 195              | 263        | 1.349      | 0.741 | 306     | 1.569   | 0.637 |
| social_golfer            | 83               | 103        | 1.241      | 0.806 | 178     | 2.145   | 0.466 |
| stee l_m ill_slab_design | 124              | 148        | 1.194      | 0.838 | 234     | 1.887   | 0.530 |
|                          |                  |            | 1.323      | 0.763 |         | 1.998   | 0.512 |

モデル定義が大幅に削減

## 2) 大規模組み合わせ最適化問題の最適化

- 問題に特化した機能 (**List**、**Set**) を使ったモデリングにより、順番を決める変数をセットで定義し、セット内で順番を変更しながら最適化をすすめることができ、パフォーマンスを大幅に向上することができるようになった。
- たとえば、Hexaly11.5は、何千ものクライアントにサービスを提供するCVRPTWおよびPDPTWの問題で世界最強の最適化ソリューションを提供することができる。

### 3) 従来と同等以上の上界値を実現

Hexaly 最新版は、従来の凸型解領域の問題に対して、bool、int、またはfloat等の意思決定変数で表現される非線形、非凸型のモデルの上界値を計算する。

結果として、Hexalyは最適性のギャップを計算することができ、中小規模の問題に対しては最適性を証明することができるようになった。

このことは、Hexalyは汎用的な従来のCplex、Xpress、Gurobiソルバと完全に競合できるだけでなく、(MI) QPまたは(MI) NLPに関するQPソルバーとも競合することができるといえる。

5

# Hexaly 最新版の新機能

# 1) Hexaly Cloud

Hexaly 10.5から、新しいサービスとしてCloudのリリースを欧米に限定して開始した。クラウドコンピューティングの力を活用し、最適化問題に対して最短時間で最良の結果を得ることがこれまで以上に簡単になっている。

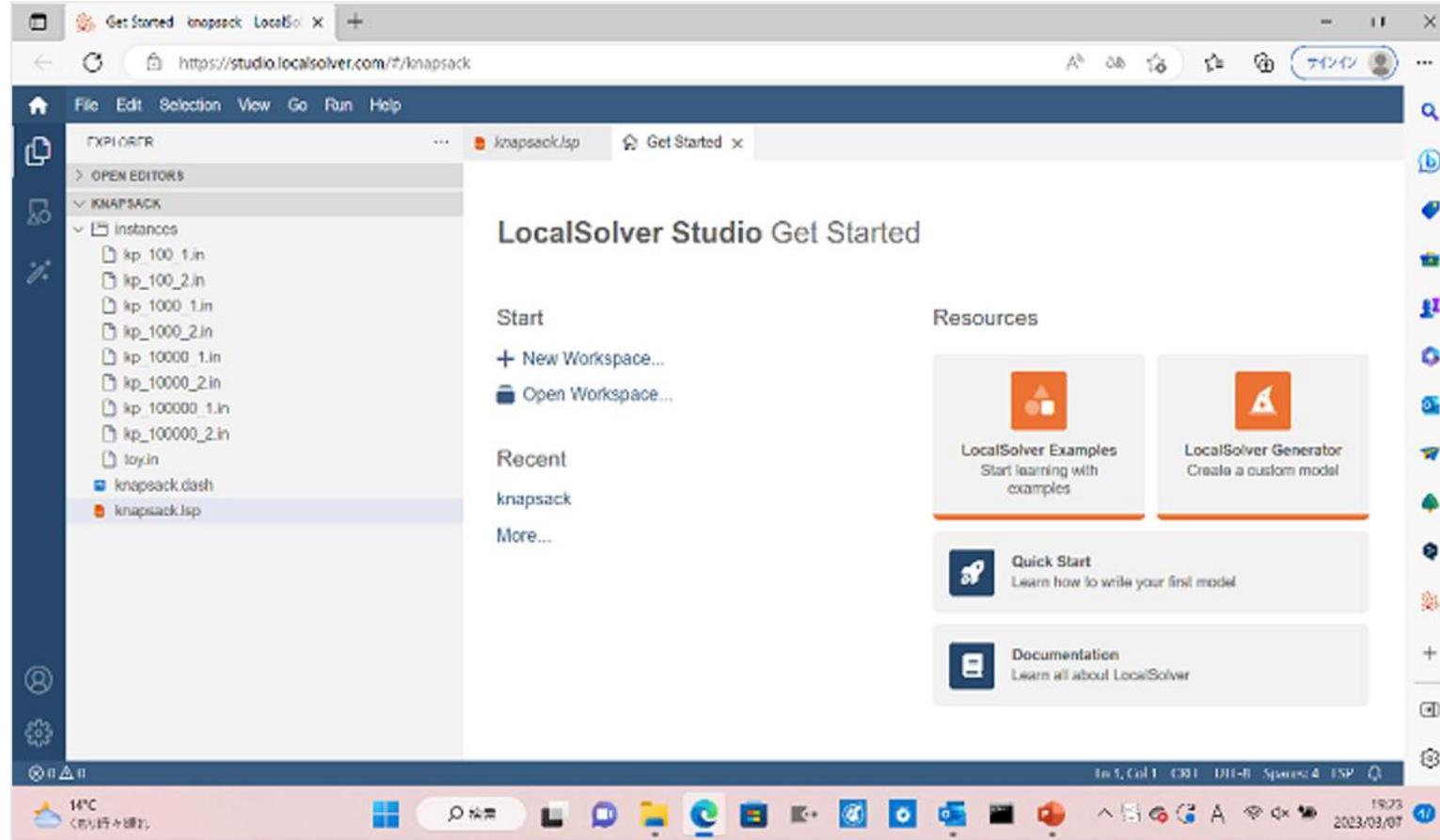
クラウド環境は、Amazon EC2でホストされ、Hexalyチームによって完全に管理される安全なリモート仮想マシンに委任されることとなり、ニーズに最も適した強力なハードウェアにアクセスする。

## 2) Hexaly Studio

Hexaly Studioは、ブラウザからどこでもHexalyを使った最適化アプリケーションを開発・テストすることを可能とした開発、テスト環境である。現在評価中であるが、既存ユーザにはテスト可能である。

DX環境での開発には、最適なソフトである。

# Hexaly Studio 画面



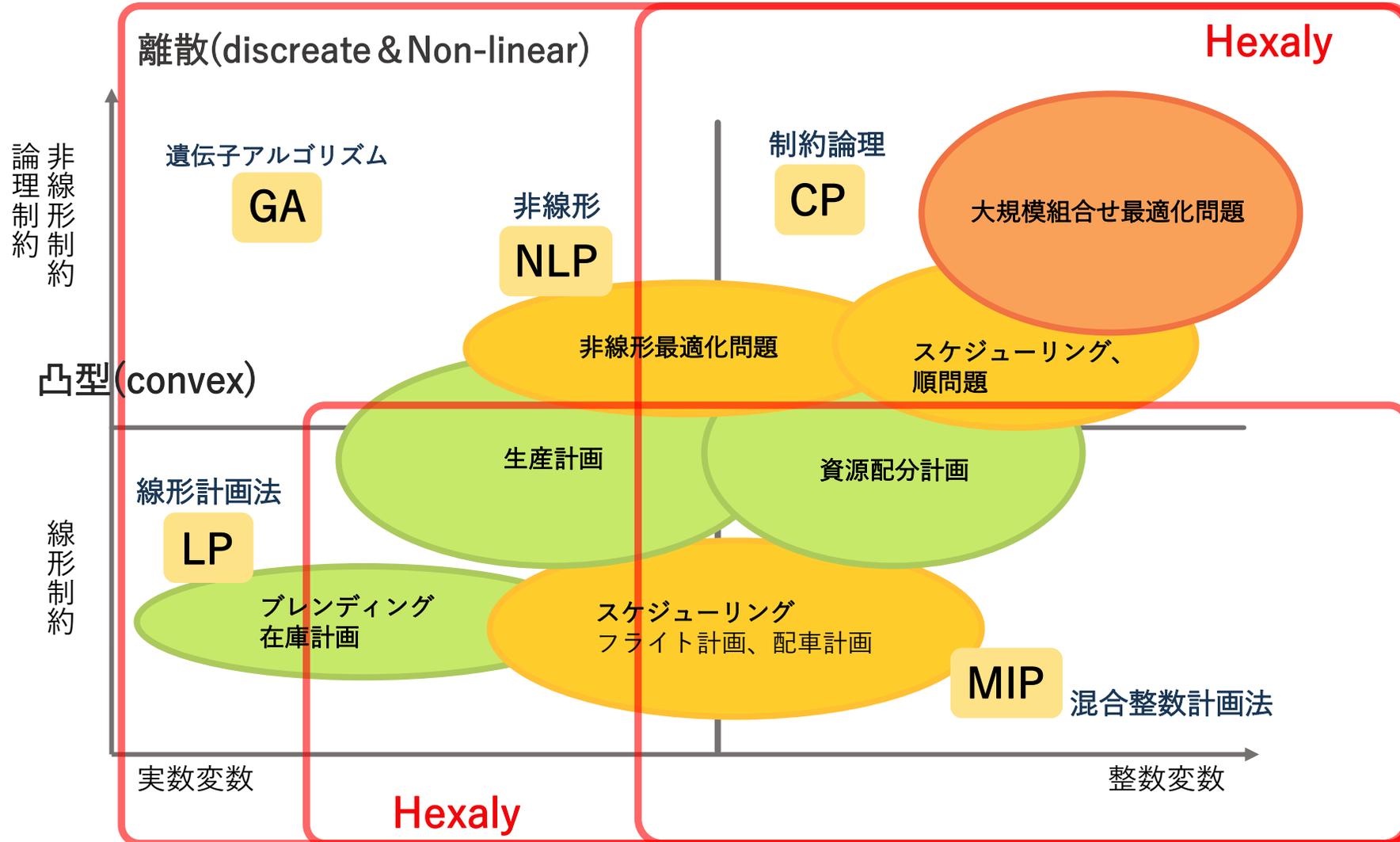
# Hexaly Studio の機能

- 一つのインターフェースで、次のような利点を実現
  - ✓データをインポートして共有するためのオンライン・ディスク・スペース
  - ✓Hexaly Cloudのパワーを活用し、Hexaly 最適化モデルを 作成、起動、デバッグするためのエディター機能
  - ✓KPI、地図、ガントチャート、その他様々なGUIウィジェットで、Hexaly 最適化モデルに命を吹き込むインタラクティブ・ダッシュボード

# 200社、1,000ユーザ、25ヶ国



# Hexaly の適用範囲



# 数式を記述して実現するモデリング

第4世代のモデリング言語による柔軟な定式化が可能（LP、MIP、NLP、CP等の問題を自由に定義可能）。

特に、従来の数理計画法の定式化（線形不等式による定式化）に限定することなくMIP問題を非常に**コンパクトに定義可能**。

従来の定式化と異なる点は、**意思決定変数**を定義し、定義した意思決定変数を使って、目的関数、制約条件を**自由形式（非線形、論理関数、集合関数等を使って）**で定義するだけでよい。

# 最適化手法と融合した次世代AI

現状のAIは統計予測をベースとした統計理論による確率論的なものや、センサ等による情報に対する対処方法がきまっていることが基本であり、意思決定理論が確定していることが重要である。

これに対して、**新しい事象でも対応できるようにする（最適解を導く）**ことが次世代AIである。

一言で言えば、**意思決定モデルを作成し、最適化手法を組み込むことにより、実績にない事象に対しても、さらなる効率化、自動化を可能することである。**

# お客様の声

“

膨大な数の変数にもかかわらず、わずか数分の実行時間で優れたソリューションを得られた



Pasco

“

私たちが最も高く評価していることは、Hexalyの数学的形式で提供された比類ないモデリングの容易さです



GROUPE  
RENAULT

“

グローバル最適化の夢が叶う



AIRBUS

“

技術的に優れた製品かつ優れたサポートレベル



PosAm

“

高度な組み合わせ問題と非線形問題を解決するための世界最高のパフォーマンスを実現する最適化エンジン



LAC

# LSPによる自然なモデリング

- **最初に、意志決定変数の定義**

- ✓例えば0-1整数変数の型を持つ意志決定変数を定義
- ✓この意志決定変数の組合せが一つの解となる
- ✓探索は意志決定変数の組合せで行われる

- **次に、制約式、目的関数の定義**

- ✓意志決定変数を使って制約式、目的関数を表現する
- ✓式の表現に、算術演算子、論理演算子等が使用でき、非線形表現で記述可能である

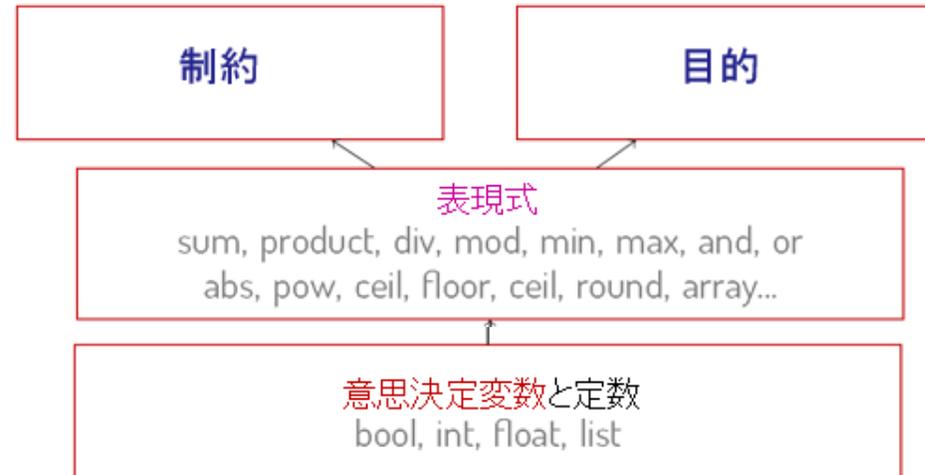
- **外部入出力及び、表示**

- ✓数値データの外部入力、結果の外部出力機能により、データと問題定義を完全に分離できる。このため、どんなに大きなモデルでも、モデル本体を簡潔に表現可能である

# 最適化問題（組合せ最適化）の例

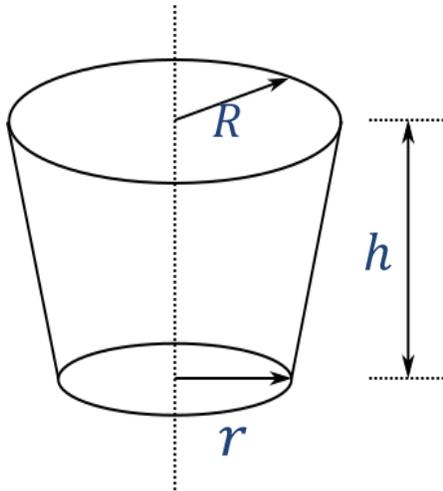
問題を自然な形で定義し実行するだけ

```
function model() {
  x[i in 0..nbItems-1] <- bool();
  knapsackWeight <- sum[i in 0..nbItems-1](weights[i] * x[i]);
  constraint knapsackWeight <= knapsackBound;
  knapsackValue <- sum[i in 0..nbItems-1](prices[i] * x[i]);
  maximize knapsackValue;
}
```



# 非線形最適化問題の例

## 体積が最大になるような形状の設計



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

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

```
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 <= 1;
```

```
  maximize V;
```

```
}
```

# 目的関数、制約条件を表現するオペレータの例

| 意思決定変数 | 算術型   |        |       | 論理型       | 大小比較 | Set関係     |
|--------|-------|--------|-------|-----------|------|-----------|
| bool   | sum   | sub    | prod  | not       | eq   | count     |
| float  | min   | max    | abs   | and       | neq  | indexof   |
| int    | div   | mod    | sqrt  | or        | geq  | partition |
| list   | log   | exp    | pow   | xor       | leq  | disjoint  |
|        | cos   | sin    | tan   | iif       | gt   |           |
|        | floor | ceil   | round | array+at  | lt   |           |
|        | dist  | scalar |       | piecewise |      |           |

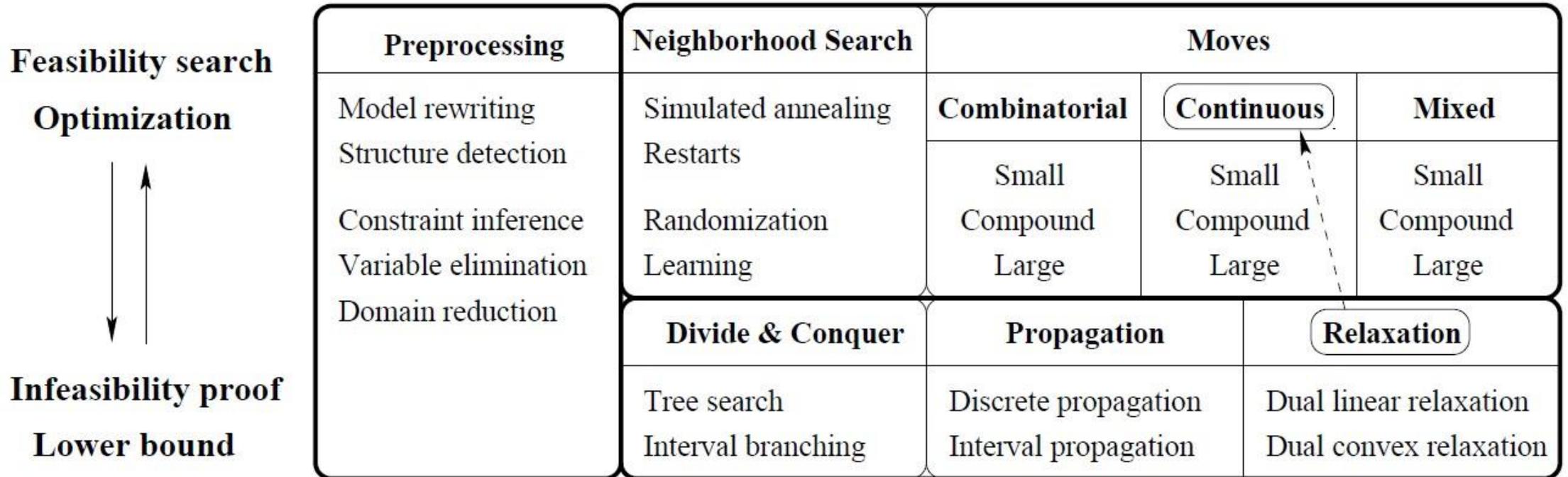
意思決定変数を定義し、上記オペレータを使って、  
目的関数、制約条件を表現するだけでモデリングが完成。  
後は実行させるだけ。

# Hexaly の最適化の特徴

- 当初、Hexaly の最適化計算方法はローカルサーチとメタヒューリスティクスだけだったが、徐々に**LP機能、MIP機能、MILP機能、MINLP機能等**の要素を包含し、各種問題に対する最適化をブラックボックスとして実行可能とした。
- 一回の解探索（イタレーション）は、意思決定変数の値を一組だけ変更し、目的関数値、制約関数値を差分計算で評価し、行います。このため、従来の最適化プログラム（Simplex法、B&C）とは異なり、最適化計算の時間は圧倒的に速く、問題規模に比例するだけで、最適化が可能となっている。

# Hexaly (技術：ノウハウの集大成)

大規模混合整数変数、非凸最適化問題に対しても、既存の最適化技術 (LS, LP/MIP, CP/SAT, NLP, ...) を統合した **All-In-One** の数理計画法システムを実現。



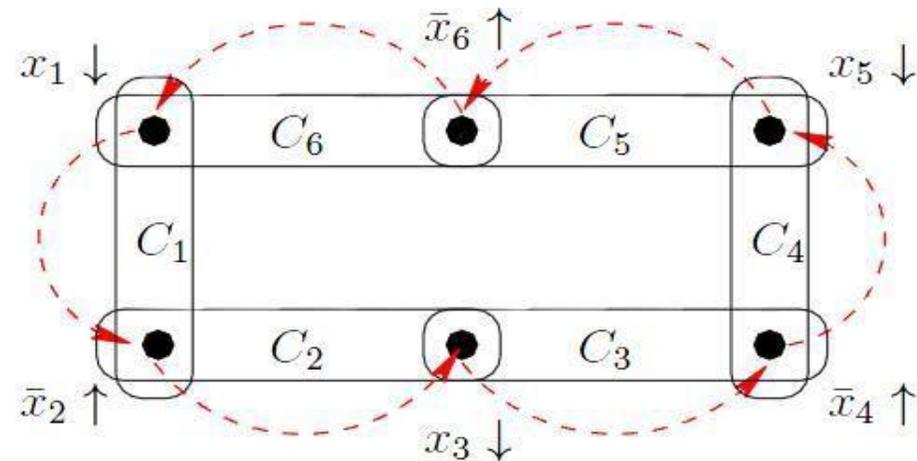
# Hexaly (探索イメージの例)

## 1) 実行可能性を保つ”Moves”

- ejection chainsの作成

※意志決定変数と制約条件からなるハイパーグラフを作成

※下記チェーンを順番にたどる (k-Chains and k-Cycles) ことは、パッキングまたはカバーリング問題でのk-Moves and k-Exchanges に該当する

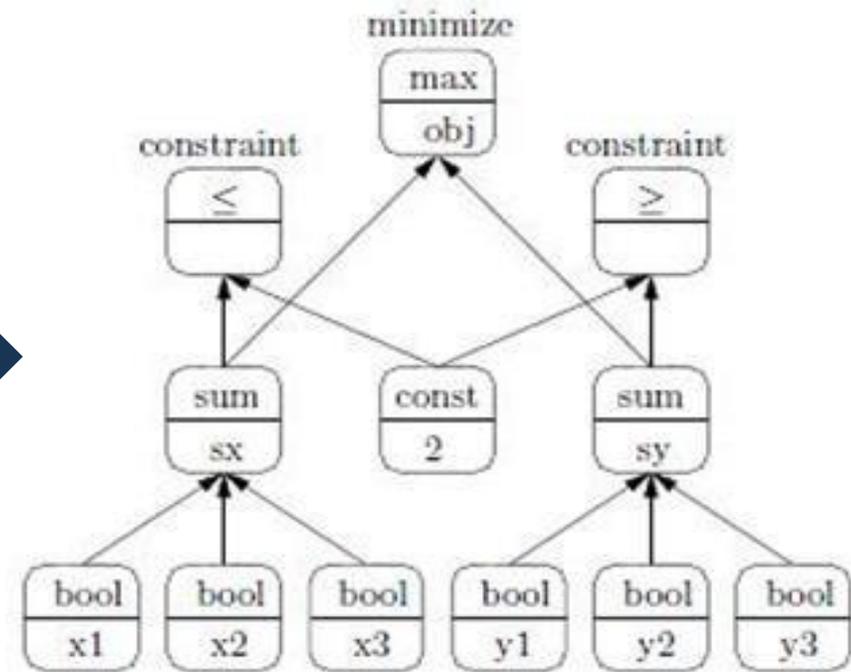


## 2) DAG:Directed acyclic graphでのモデル表現

```

x1 <- bool(); x2 <- bool(); x3 <- bool();
y1 <- bool(); y2 <- bool(); y3 <- bool();
sx <- sum(x1, x2, x3);
sy <- sum(y1, y2, y3);
constraint sx <= 2;
constraint sy >= 2;
obj <- max(sx, sy);
minimize obj;

```



# Hexaly (改良イメージ)

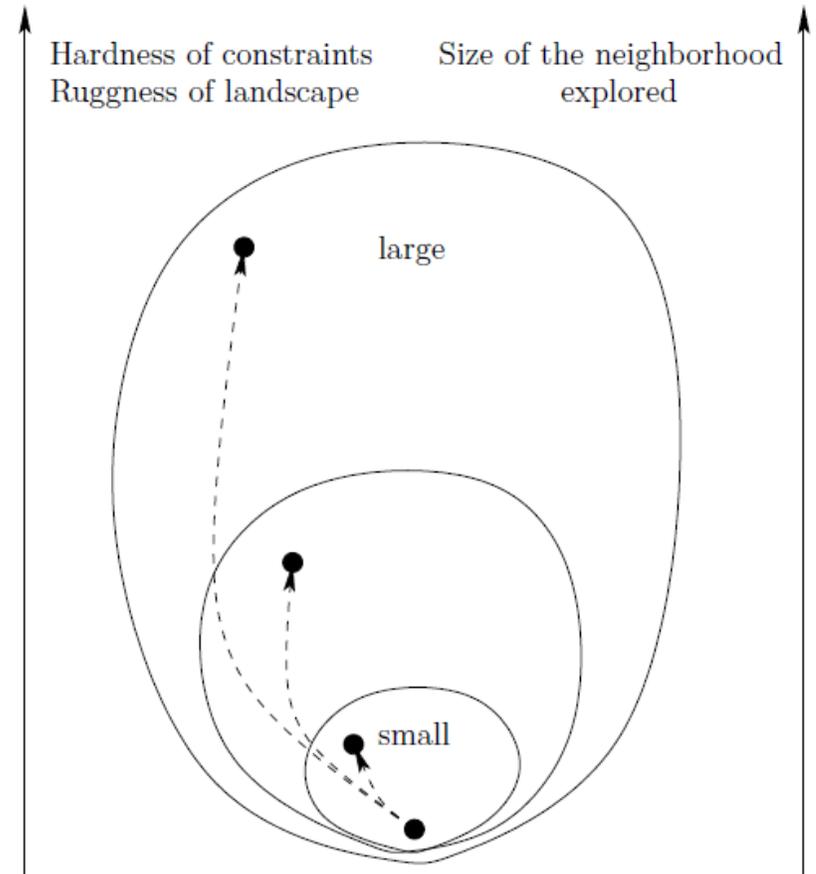
## 探索を効率を良くしていく評価方法 (DAG:Directed acyclic graph)

- **DAG内のムーブの評価 (Lazy propagation)**  
DAGの各ノードは、子ノードの値が修正された場合にのみ、1回だけ参照される。  
例： $x \leftarrow a < b$  が真のとき、ムーブで  $a$  が小さくなったり、 $b$  が増加する場合には、 $x$  を評価しない
- **算術演算子による値が変更されない場合**  
例： $z \leftarrow \text{or}(a_1, \dots, a_k)$  リスト  $T$  の  $a_i$  が 1 であり、リスト  $M$  の  $a_i$  がムーブで修正された場合、 $|T| \neq |M|$  ならば  $z = 1$  となる。この場合、 $O(1)$  回、ショートカットする

# Hexaly (グローバル探索イメージ)

## グローバル探索としての近傍探索

- 小さい近傍から探索を広げていく
- 近傍探索からダイナミックに探索領域を調整：縮小、拡大、特化
- 大規模領域探索にツリー探索(MIP、CP)を導入
- 完全な近傍と正確な探索では最適解に到達



# Hexaly のAPI:

- 独自の関数型言語LSPによる開発、実行環境
- C++、C#、JAVA、のAPIを装備
- 他に、Python 2.7、3.2、3.4、3.6、3.8、3.9用のAPIを装備

Pythonからの利用が可能

# 大規模最適化の現状

| 課題レベル  | 課題  |
|--------|---|
| 出来ないこと | <p>大規模MIP最適化問題が解けない</p> <p>大規模組合せ最適化問題が解けない</p> |
| 適用の課題  | 時間がかかる、制約が表現できない                                |
|        | 大規模なシステム資源が必要（CPU、メモリ等）                         |
| 運用の課題  | モデルチューニングがタイムリーに出来ない                            |
|        | モデルの変更が簡単にできない                                  |

# 何故今 Hexaly なのか

- 今までのOR手法（最適化）の使い方が広がり、DX推進に求められるOR手法（最適化）が必要
- 従来の解法依存型での定式化は必要なく、非線形問題をも含む**汎用問題として定式化が可能**  
（例として：罰金関数等を使用する必要はない）
- 誰でも容易に問題定義ができ、使えることを実現

# Hexaly で実現した大規模な事例

以下の事例は、仏 Hexaly 社のホームページからの抜粋である。

いずれも、DXを推進した事例であり、複数の事業部門、複数の工程、複数の意思決定を必要とする計画業務をDXの推進とともに実現した例である。

# 自然なモデリングの実現

第4世代のモデリング言語による柔軟な定式化が可能（LP、MIP、NLP、CP等の問題を自由に定義可能）

従来の定式化と違い、まず、意思決定変数を定義し、定義した意思決定変数を使って、目的関数、制約条件を数式（非線形、論理関数、集合関数等を使って）で定義するだけでよい。

特に、従来の数理計画法の定式化（線形不等式による定式化）に限定することなく最適化問題を非常にコンパクトに定義可能。

6

おわりに

# 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。

メタバース（仮想空間）、DXの発展により、だれでも、最適化を考えることが可能となりつつある。

「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。

## From Bouygues e-lab

フランスの企業グループ。中核企業の建設会社「ブイグ建設」は1952年設立。参加に建設会社のほか放送局「TF1」や電気通信事業会社「ブイグ・テレコム」を置く。現在、Hexaly社として別会社になっている。

- **Thierry Benoist** : heading the Optimization department at Bouygues e-lab.
  - ✓ OR projects leader to operational softwares in the Bouygues Group.
  - ✓ Graduated from École Polytechnique (1998), he holds a PhD in computer science (2004).
  - ✓ 2005 Doctoral Prize for Innovative Applications by French Information Technology Society
  - ✓ 2006 Robert Faure 3rd Prize awarded every 3 years by French Operations Research Society (ROADEF) to the best young OR researchers
- **Frédéric Gardi** : deputy head of Optimization department at Bouygues e-lab  
He conducts OR projects since 10 years in various business areas, from banking to heavy industry. He holds a PhD in computer science from Aix-Marseille University (2005).  
Expert in the design and engineering of high-performance local-search algorithms, he won with B. Estellon and K. Nouioua several international OR competitions (1st Junior and Senior Prize of ROADEF 2005 Challenge, 2nd Senior Prize of ROADEF 2007 Challenge, ROADEF 2011 1st Prize for Industrial Applications).

# LocalSolver 開発プロジェクト

- 2007: Beginning of **Hexaly** project
  - ✓ Long-term objectives:
    1. Defining a simple, generic declarative formalism suited for LS (**modeling**)
    2. Developing an effective LS-based **solver** with fundamental principle:
- 2009: First software concretization: **Hexaly 1.0**
  - ✓ Allows to tackle large-scale **0-1 programs**
  - ✓ Binaries **freely** distributed at [www.localsolver.com](http://www.localsolver.com)
- 2011: **Hexaly 1.1** (multithreading, enriched moves, annealing)
- 2012: **Hexaly 3.0** (floating, finding feasible solution)
- 2021: **Hexaly 10.0** (All-In-One Solver)

現在も発展中

## 宮崎 知明 MSI株式会社 技術顧問。数理モデリング研究所所長。

|   |  |  |  |
|---|--|--|--|
| <p><b>1. 得意分野</b></p> <p>(1) SCM: サプライチェーンマネジメント</p> <ul style="list-style-type: none"> <li>・戦略レベルの最適化計画の策定およびビジネスモデルの提案</li> <li>・戦術レベルの最適化計画の策定およびビジネスモデルの提案</li> <li>・オペレーションレベルの各種スケジューリングの策定</li> <li>・生産計画、在庫計画、物流計画、販売計画の策定</li> </ul>   |  | <p>(2) OR: オペレーションズリサーチ</p> <ul style="list-style-type: none"> <li>・モデリング</li> <li>・解法設計、開発</li> <li>・最適化(数理計画法ほか)</li> <li>・スケジューリング、計画支援</li> </ul>  |  |
| <p><b>2. 業務履歴</b></p> <p>(1) 1974年3月: 東京大学工学部応用物理学物理工学コース卒業</p> <p>(1) 1974年4月: 富士通(株)に入社</p> <p>(2) 1974年-1983年: 数理計画法の研究、製品開発及び顧客適用業務に従事</p> <p>(3) 1983年-1991年: OR手法全般の製品化戦略及び顧客適用業務に従事</p> <p>(4) 1992年-1997年: OR手法全般、DWH、OLAP、DM製品戦略及び顧客適用業務に従事</p> <p>(5) 1998年-2001年: SCMパッケージ適用コンサルティング、ソリューション開発</p> <p>(6) 2002年4月-2010年6月(株)富士通総研に出向。数理解析技術による各種最適化の適用コンサルティング</p> <p>(7) 2010年7月-2012年11月 株式会社メルキュールシステムに入社。数理モデリング研究所を社内で設立。最適化コンサルティング。</p> <p>(8) 2012年12月; MSI株式会社 技術顧問。数理モデリング研究所所長。最適化コンサルティング。</p>   |  |  |  |
| <p><u>&lt;顧客への主なコンサルティングプロジェクト&gt;</u></p> <p>①石油精製A社: 生産計画(予算計画)モデルの構築及びシステム導入支援</p> <p>②石油元売B社: 原油購入計画モデルの構築およびシステム開発</p> <p>③製鉄業C社: 生産計画モデルの構築及びシステム導入支援</p> <p>④衣料製造業D社: 人員配置及び生産計画システム策定及び解法設計</p> <p>⑤自動車部品製造業E社: 合金生産計画システム策定及び解法設計</p> <p>⑥自動車製造業F社: エンジン制御シミュレーションシステムコンサルティング</p> <p>⑦鋳鉄製造業G社: 月次生産計画コンサルティング及び解法設計</p> <p>⑧製紙業H社: 生産計画コンサルティング及び解法設計</p> <p>⑨製紙業I社: OLAP活用コンサルティング</p> <p>⑩石油化学J社: 月次工場生産計画コンサルティング及び解法設計</p> <p>⑪石油化学K社: 全社生産計画コンサルティング及び解法設計</p> <p>⑫総合化学L社: 配車配送計画コンサルティング及び解法設計</p> <p>⑬食品製造業M社: 企業合併に伴う生産物流戦略コンサルティング</p> <p>⑭石油元売N社: タンクローリ配車スケジューリングシステム解法設計</p> <p>⑮自動車製造業O社: 完成車配送スケジューリングシステム策定及び解法設計</p> <p>⑯電機製造業P社: SCMパッケージ導入コンサル及びプロジェクトマネジメント</p> <p>⑰食品製造業Q社: 生産、物流(SCM)計画導入コンサルティング</p> <p>⑱エネルギー製造業R社: 保守要員計画導入コンサルティング その他</p> |  | <p><u>&lt;コンサルティング技法、ツール開発、製品企画など&gt;</u></p> <ul style="list-style-type: none"> <li>・数理計画法システムの製品企画及びシステムの開発</li> <li>・OLAPツール製品企画</li> <li>・DM(データマイニング)製品企画</li> <li>・シミュレーションシステム製品企画</li> <li>・物流最適化システム企画、開発</li> <li>・プロセス最適化システム企画、開発</li> <li>・フィールドイノベーション活動支援(顧客業務改善、改革)</li> </ul> <p><u>&lt;対外発表・その他活動&gt;</u></p> <ul style="list-style-type: none"> <li>・OR学会編集委員1996-2000</li> <li>・OR学会、情報処理学会発表1995-</li> <li>・PSLXコンソーシアム評議委員</li> <li>・SCC日本支部ボードメンバー1999</li> <li>・OR普及賞受賞(富士通ソフトウェア事業部として)</li> <li>・スケジューリング学会理事2007</li> <li>・日本オペレーションズ・リサーチ学会フェロー2008</li> <li>・筑波大学講師(2012年上期)、中央大学講師(2014年下期)</li> </ul> |  |