QUICK START GUIDE

Решение задач опримизации с помощью Optimization Toolbox™

Используйте естественный синтаксис для определения и решения линейных в том числе смешанных целочисленных, квадратичных, линейных задач наименьших квадратов и нелинейных задач оптимизации.

1. Определение задачи

Следуя основанному *рабочему процессу*, сначала создайте задачу оптимизации с помощью **optimproblem**, чтобы передать туда целевую функцию, ограничения и связанные переменные.

Примеры:

```
assignmentProb = optimproblem
responseProb = optimproblem
```

2. Определение переменных

Создайте переменные оптимизации с помощью **optimvar**. Задайте отображаемое имя и дополнительные размеры, границы и тип. Индекс с целыми числами или символьными строками.

Примеры:

```
x = optimvar("x");
y = optimvar("y");

employees = ["a","b","c"];
tasks = ["t1","t2","t3"];
assign = optimvar("assign",employees,tasks,"LowerBound",0,"UpperBound",1,"Type","integer")
```

3. Определение выражений для использования в задачах и ограничениях

Непосредственно укажите **OptimizationExpression**, представляющее собой отношение многочленов.

Примеры:

```
response = -3*(y - x.^3 - x).^2 - (x - 4/3).^2;
totalCost = sum(sum(cost.*assign));
sumByEmployee = sum(assign,2);
sumByTask = sum(assign,1);
```

Укажите другие выражения в качестве функций MATLAB и преобразуйте их в оптимизационные выражения с помощью **fcn2optimexpr**.

Примеры:

```
a = 4;
xyfcn = @(x,y,a)exp(y)*a*x.^2;
xyexpr = fcn2optimexpr(xyfcn,x,y,a);
```

4. Определение целевой функции

Установите *смысл* оптимизации. Задайте *целевую функцию* со скалярным **OptimizationExpression**.

Примеры:

```
responseProb.ObjectiveSense = "maximize";
responseProb.Objective = response;

assignmentProb.ObjectiveSense = "minimize";
assignmentProb.Objective = totalCost;
```

5. Определение ограничений

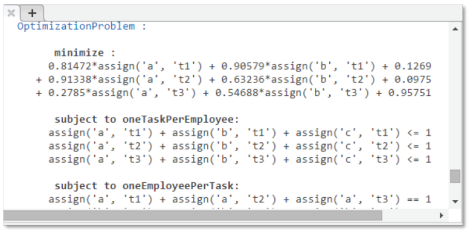
Объедините выражения оптимизации с реляционным оператором, чтобы задать **`OptimizationConstraint`**. Назначьте задачу.

Примеры:

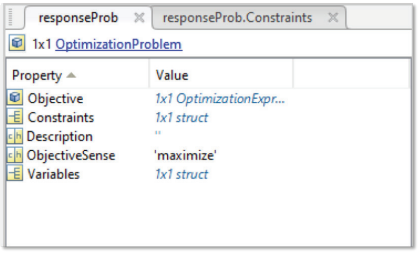
```
responseProb.Constraints.ellipse = x.^2/2 + y.^2/4 <= 1;
responseProb.Constraints.xyconstr = xyexpr >= 1;

assignmentProb.Constraints.oneTaskPerEmployee = sumByTask <= 1;
assignmentProb.Constraints.oneEmployeePerTask = sumByEmployee == 1;
```

6. Проверка

Вывод в командную строку с помощью: **`showexpr`**, **`showconstr`**, **`showbounds`**, и **`showproblem`**.



Просмотр в воркспейсе



7. Решение и анализ

Решите проблему, возвращая значения решения, значение целевой функции и причину, по которой решатель остановился. Укажите *начальную точку* для нелинейных задач.

Примеры:

```
x0.x = 0;
x0.y = 0;
[sol,fval,exitflag] = solve(responseProb,x0)
```

```
sol = struct with fields:
    x: 0.8883
    y: 1.5563
fval = -0.2013

exitflag =
    OptimalSolution
```

Решение на основе *опций оптимизации*

Примеры:

```
o = optimoptions(assignProb,"MaxTime",10);
sol = solve(assignmentProb,"Options",o)
```

Дополнительно:

- Используйте **`evaluate`** и **`infeasibility`** для анализа результата
- *Интерпретируйте* и *улучшайте* результат
- Преобразуйте к подходящему виду с помощью **`prob2struct`**
- *Добавьте производные*

Узнать больше: *docs.exponenta.ru/optim/index.html*