MULTIOBJECTIVE OPTIMIZATION VIA WEB-BASED SERVICES

Ignacy Kaliszewski, Olga Karelkina

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

December 14, 2020

Table of contents

Motivation

- commercial software availability to the academic community
  - to broaden the scope of the research to more complex problems
  - to deeper the analyses by solving instances of problems with sizes previously beyond computational tractability
- for the multiobjective optimization the offer is very limited
  - (Fourer 2019) among 47 optimization software brands listed, only 13 support methods to handle multiple objectives

Motivation

- commercial software availability to the academic community
  - to broaden the scope of the research to more complex problems
  - to deeper the analyses by solving instances of problems with sizes previously beyond computational tractability
- for the multiobjective optimization the offer is very limited
  - (Fourer 2019) among 47 optimization software brands listed, only 13 support methods to handle multiple objectives

Large-scale real-world problems

- Build financial portfolios by balancing risks and rewards
- Long-term plans for typical supply chain processes
- Hospital resources and capacity management

Optimization software

- commercial
  - IBM CPLEX Optimizer
  - Gurobi Optimization Solver
- web-based services
  - NEOS Server
  - WWW-NIMBUS
  - ONLINEMOCO
- callable libraries (free, open-source)
  - CBC
  - GLPK
  - LP_Solve

Multiobjective optimization problem

Let x denote a solution
X a space of solutions, $X \subseteq \mathbb{R}^n$
X_0 a set of feasible solutions, $X_0 \subseteq X$
The general multiobjective optimization problem is defined as:

$$\begin{gathered} vmax f(x) \\ \text{s.t.} \\ x \in X_0 \,, \end{gathered} \tag{1}$$

where $f : X \to \mathbb{R}^k$, $f = (f_1, \dots, f_k)$,
$f_l : X \to \mathbb{R}$, $l = 1, \dots, k$, $k \geq 2$, are objective functions, and *vmax* denotes the operator of deriving all Pareto optimal solutions in X_0.
$\mathbb{R}^k$ is called the objective space.

Optimal solutions

Pareto optimal

Solution $\bar{x}$ is **Pareto optimal (or: efficient)** if $f_l(x) \geq f_l(\bar{x})$, $l = 1, \ldots, k$, implies $f(x) = f(\bar{x})$. If $f_l(x) \geq f_l(\bar{x})$, $l = 1, \ldots, k$, and $f(x) \neq f(\bar{x})$, then we say that x dominates $\bar{x}$ and we write $\bar{x} \prec x$.

We shall denote the set of Pareto optimal solutions to (1) by N (the efficient set). Set $f(N)$ is called the Pareto front (PF).

Chebyshev scalarization

Solution x is **Pareto optimal** if and only if it solves the Chebyshev weighted optimization problem

$$\min_{x\in X_0} \max_{l} [\lambda_l (y_l^* - f_l(x)) + \rho e^k (y^* - f(x))] , \tag{2}$$

$\lambda_l > 0, \; l = 1, \ldots, k, \;\; e^k = (1, 1, \ldots, 1)$
$y_l^* > \max_{x\in X_0} f_l(x)$ if the maximum exists and $y_l^* \geq \sup_{x\in X_0} f_l(x)$ if the maximum does not exist
ρ is a positive "sufficiently small" number
An equivalent formulation to (2) is

$$\begin{gathered} \min_{x\in X_0} s \\ \text{s.t.} \\ s \geq \lambda_l (y_l^* - f_l(x)) + \rho e^k (y^* - f(x)), \quad l = 1, \ldots, k. \end{gathered} \tag{3}$$

The workflow

To derive a solution to a multiobjective optimization problem by a web-based service or a stand-alone application, the following generic workflow applies:

- upload problem data input file,
- upload a number of weight sets,
- establish y^*,
- for each weight set solve either (2) or (3) problem formulation,
- download output files.

General web application architecture

Structural web application components

- client side
  - a user-friendly representation of a web app's functionality that a user interacts with
  - written in HTML, JavaScript and CSS
  - exists within the user's web browser and doesn't need any specific OS/device-related adjustments
- server side
  - web server with app logic (the main control center)
  - database (storage of all persistent data)
  - written in PHP, Java, .NET, Python, Ruby on Rails or Node.js

IDOL

Idea-Demonstrating multiobjective Optimization pLatform

Development stack of the prototype

Development stack of the stand-alone Idol version includes:

- Python 3.7
- Tkinter Package
- PyInstaller 3.1
- Gurobi 9.0.1 solver
- Windows 10

Idol workflow

First, to solve the problem with Idol, one should construct an .lp file with the problem in multi-objective format according to Gurobi specification.

Then the user should follow the steps:

1. Load file.
2. Generate reference point.
3. Load weights.
4. Generate Chebyshev scalarization.
5. Optimize Chebyshev scalarization.

Idol GUI: start window

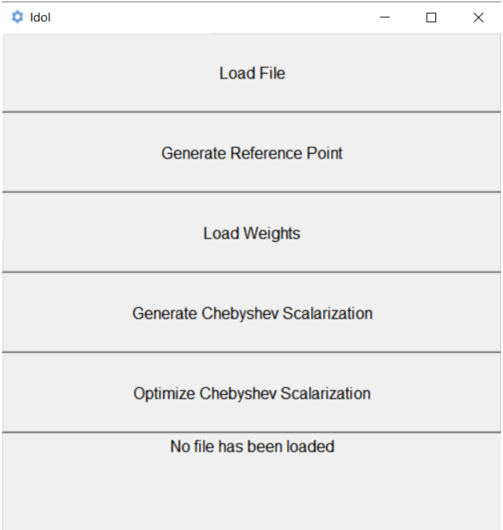

Figure: Idol interface upon application start

The MIP model

The MIP model (LP model, as the special case) consists of three primary parts:

1. The set of decision variables. Variables can be continuous and/or integer (in particular, binary).
2. The set of constraints.
3. The objective function (singleobjective case) or the multiple objective functions (multiobjective case).

According to Gurobi requirements all expressions in multiobjective case have to be linear.

Multidimensional knapsack problem

Problem instance from OR-library

$n = 100$ is the number of variables
$k = 3$ is the number of criteria (objectives)
$m = 10$ is the number of constraints

$$\max \sum_{j=1}^{n} c_j^k x_j,$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \;\; i = \overline{1, m}$$

$$x_j \in \{0, 1\}$$

Problem parameters a_{ij}, b_i and c_j^k are integers drawn at random but uniformly from $(0, 1000)$

Gurobi multiobjective lp format

```
\ LP format - for model browsing. Use MPS format to capture full model detail.
Maximize multi-objectives
  Obj1: Priority=1 Weight=1 AbsTol=0 RelTol=0
   803 x1 + 728 x2 + ... + 600 x100
  Obj2: Priority=1 Weight=1 AbsTol=0 RelTol=0
   630 x1 + 674 x2 + ... + 993 x100
  Obj3: Priority=1 Weight=1 AbsTol=0 RelTol=0
   909 x1 + 917 x2 + ... + 694 x100
Subject To
 constr1: 300 x1 + 446 x2 + ... + 334 x100 <= 13640
 constr2: 78 x1 + 955 x2 + ... + 441 x100 <= 12048
 constr3: 275 x1 + 886 x2 + ... + 489 x100 <= 12574
 constr4: 724 x1 + 871 x2 + ... + 374 x100 <= 13915
 constr5: 674 x1 + 557 x2 + ... + 695 x100 <= 13232
 constr6: 227 x1 + 978 x2 + ... + 929 x100 <= 12190
 constr7: 84 x1 + 336 x2 + ... + 411 x100 <= 12242
 constr8: 587 x1 + 442 x2 + ... + 438 x100 <= 12394
 constr9: 768 x1 + 71 x2 + ... + 416 x100 <= 12258
 constr10: 51 x1 + 222 x2 + ... + 805 x100 <= 10960
Bounds
Binaries
 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19 x20 x21
 x22 x23 x24 x25 x26 x27 x28 x29 x30 x31 x32 x33 x34 x35 x36 x37 x38 x39
 x40 x41 x42 x43 x44 x45 x46 x47 x48 x49 x50 x51 x52 x53 x54 x55 x56 x57
 x58 x59 x60 x61 x62 x63 x64 x65 x66 x67 x68 x69 x70 x71 x72 x73 x74 x75
 x76 x77 x78 x79 x80 x81 x82 x83 x84 x85 x86 x87 x88 x89 x90 x91 x92 x93
 x94 x95 x96 x97 x98 x99 x100
End
```

Figure: Gurobi multiobjective lp format structure

Idol GUI: load model

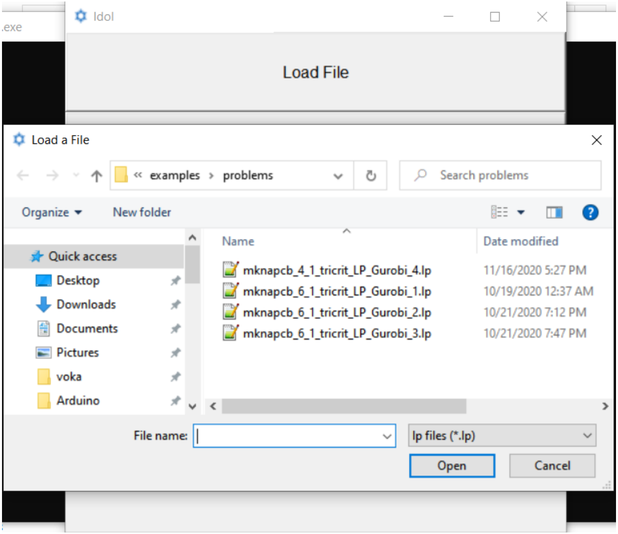

Figure: Load File dialog

Idol GUI: reference point generated

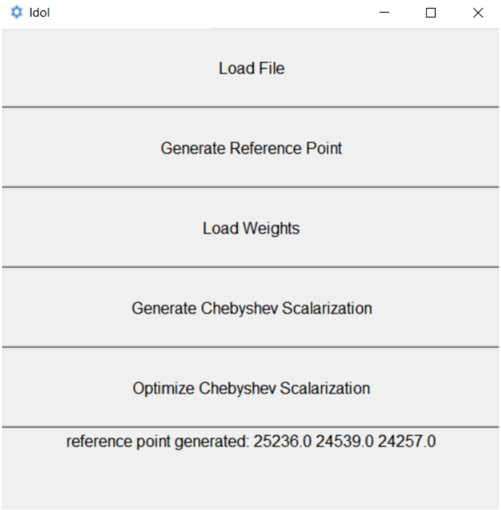

Figure: Idol interface after reference point has been generated mknapcb_4_1_tricrit_LP_Gurobi_4.lp.

Idol GUI: error message

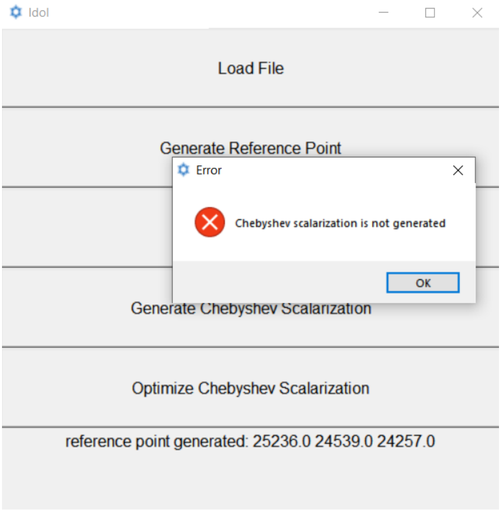

Figure: Idol interface with error message.

Idol GUI: console

C:\MOLP GUI\gui.exe

```
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  128.84413    0    15 15720.3520 128.84413  99.2%     -    0s
H    0     0                     2334.8570000 128.84413  94.5%     -    0s
H    0     0                      745.1700000 128.84413  82.7%     -    0s
H    0     0                      551.3460000 128.84413  76.6%     -    0s
H    0     0                      543.4260000 128.84413  76.3%     -    0s
     0     0  132.39169    0    15  543.42600 132.39169  75.6%     -    0s
     0     0  133.86606    0    15  543.42600 133.86606  75.4%     -    0s
     0     0  140.35892    0    17  543.42600 140.35892  74.2%     -    0s
H    0     0                      489.8010000 140.35892  71.3%     -    0s
     0     0  140.91588    0    17  489.80100 140.91588  71.2%     -    0s
     0     0  140.91588    0    17  489.80100 140.91588  71.2%     -    0s
     0     0  141.18451    0    17  489.80100 141.18451  71.2%     -    0s
     0     0  141.25377    0    19  489.80100 141.25377  71.2%     -    0s
     0     0  149.39125    0    20  489.80100 149.39125  69.5%     -    0s
H    0     0                      394.3130000 149.39125  62.1%     -    0s
     0     0  161.70682    0    20  394.31300 161.70682  59.0%     -    0s
H    0     0                      264.6730000 161.70682  38.9%     -    0s
     0     2  161.70682    0    20  264.67300 161.70682  38.9%     -    0s

Explored 43710 nodes (210153 simplex iterations) in 2.18 seconds
Thread count was 8 (of 8 available processors)

Solution count 8: 264.673 394.313 489.801 ... 15720.4

Optimal solution found (tolerance 1.00e-05)
Best objective 2.646730000000e+02, best bound 2.646730000000e+02, gap 0.0000%
Optimization is done!
```

Figure: Idol console window.

Discussion and Conclusions

- Developing problem specific web-based service
- Making the service internally available in the Institute, for testing purposes
- Making the optimization paradigm more accessible for the general audience
- Solving large-scale problems under a limited resource budget using new methodologies
- Enhancing web-service with problem generator based on the known model structure