IDOL – User Guide

Ignacy Kaliszewski[*][†] Olga Karelkina[‡][§]

The Purpose

IDOL (Interactive-Demonstrating multiobjective Optimization pLatform) constructs files with the Chebyshev scalarizations of multiobjective mixed-integer problems (MIP) in formulation (3).

Contents

[*] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.
[†] Warsaw School of Information Technology, Warsaw, Poland.
[‡] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.
[§] Corresponding author, Olga.Karelkina@ibspan.waw.pl

1 Installation and Usage

IDOL source code is publicly available and can be downloaded from GitHub via the following link "https://github.com/rhombicosi/Idol". The detailed instructions on how to run the web service locally and technical dependencies are described in README file.

The web service, is currently hosted by the Systems Research Institute server and available at http://213.135.34.49 IP address via VPN only.

Current version of IDOL supports Chrome and FireFox web browsers.

2 Multiobjective Optimization

Let $x \in \mathbb{R}^n$ and $X_0 \subseteq \mathbb{R}^n$. We call x a *solution*, and if $x \in X_0$, we call it a *feasible solution*. Accordingly, X_0 is called the *set of feasible solutions*.

Feasible solution $\bar{x},\ \bar{x} \in X_0$, 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 $\bar{x} \prec x$. Feasible solution $\bar{x},\ \bar{x} \in X_0$ is *weakly Pareto optimal* (or: weakly-efficient) if there is no such x that $f_l(x) > f_l(\bar{x}),\ l = 1, \ldots, k$. Below, 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).

Then the general multiobjective optimization problem is defined as:

$$\begin{aligned} &\max f(x) \\ \text{s.t.}\quad &x \in X_0\,, \end{aligned} \tag{1}$$

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

It is a well known that solution x is (properly) Pareto optimal, if and only if it solves the *Chebyshev weighted optimization problem*

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

where:

- parameters (*weights*) $\lambda_l > 0, \ l = 1, \ldots, k$,

- $e^k = (1, 1, \ldots, 1)$,

- $y_l^* = \sup_{x\in X_0} f_l(x) + \varepsilon, \ \varepsilon > 0, \ l = 1, \ldots, k$,

- ρ is a positive "sufficiently small" number.

By solving problem (2) with $\rho = 0$, weakly Pareto optimal solutions are derived.

By the "only if" part of this result, no Pareto optimal solution is a priori excluded from being derived by solving an instance of optimization problem (2). In contrast to that, maximization of a weighted sum of objective functions over X_0 does not possess, in general (and especially in the case of problems with discrete variables), this property.

The only assumption which has to be made for problem (2) to be well-defined is $\sup_{x\in X_0} f_l(x) < +\infty, \ l = 1, \ldots, k$, which is rather an obvious condition when it comes to numerical optimization calculations.

By formulation (2), the multiobjective optimization problem, which in principle consists in having all Pareto optimal solutions derivable, reduces to the existence of an appropriate solver capable to solve (2) for any set of weights $\lambda_l, \ l = 1, \ldots, k$.

An equivalent formulation to (2) is

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

This formulation is suitable for mixed integer programming (MIP) solvers, which in general are not able to cope with the nonlinear *max* function.

3 IDOL Interface

IDOL accepts data prepared for multiobjective optimization as required by MIP solver Gurobi. Namely, data have to be structured in the LP standard format, with two exceptions (the Gurobi genre). First, the number of objective functions can be more than one. Second, variable coefficients have to be separated from variable names by whitespace character(s).

Once the necessary input data is assembled as an *.lp file, Gurobi genre, you can upload this file to IDOL.

Together with the *.lp file you can, but you do not need, to upload the *Weights.txt* file (name can be any) containing one or several sets of weights. A single set of weights occupies a single row in the file, each successive weight separated by whitespace character(s). If *Weights.txt* file is absent, IDOL uses default weights all equal 1.

Likewise, it is possible, but not necessary, to upload the *Y.txt* file (name can be any) containing one or several sets of y^* elements. A single y^* element occupies a single row in the file, each successive component separated by whitespace characters space. If *Y.txt* file is absent, IDOL establishes y^* maximizing each objective function separately.

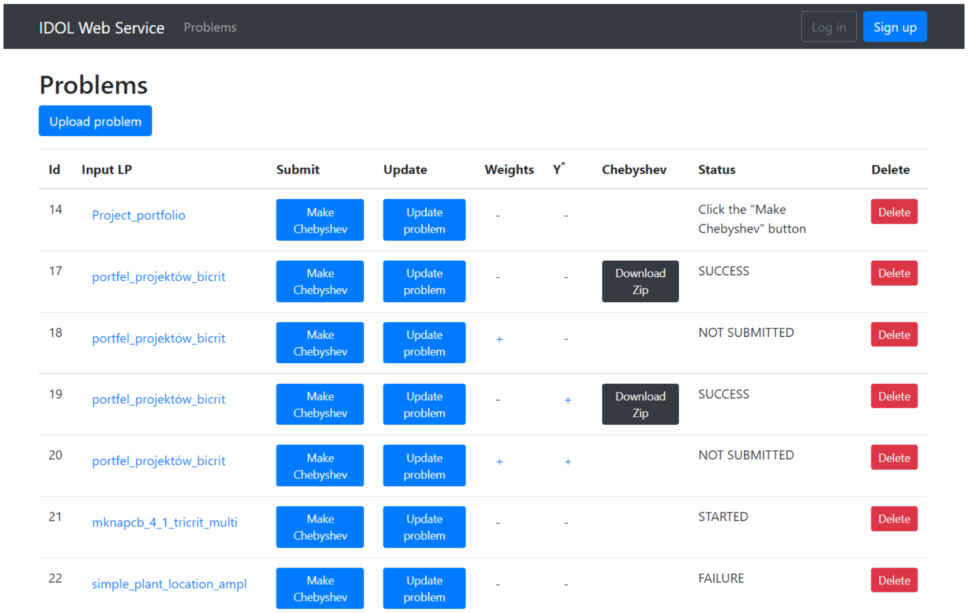
The successive step is performed in iterations. In each iteration, the Chebychev scalarization is built with consecutive y^* and the consecutive set of weights. All resulting LP files are put to a Zip archive and the archive can be download.

At present, parameters ρ and ε are hard coded to $\rho = 0.001$, $\varepsilon = 0.1$. To calculate y^* fast, the optimal values of each objective function are solved with 1% tolerance.

Each IDOL page contains the navigation bar to register and log in, go to *Problems* page, *My problems* page or home page.

Logged in users have the option to go to the individual *My problems* page. The content of this page is not visible to other users. Except this, *My problems* page replicates the logic of the *Problems* page.

On *Problems* page (see Figure 1 representing an example population of this page by various problems) you can upload LP files, upload *Weights* and y^* files, produce the file with Chebyshev scalarizations of the uploaded LP file, and delete all information related to the uploaded file together with the file itself. Cells of columns *Weights* and y^* indicate whether for the corresponding problems weights or Y^*(s) are provided by you ("+") or by IDOL ("-"). Clicking *Make Chebyshev* blue button starts producing the Chebyshev scalarizations (as many as the number of sets of weights times the number of y^* elements provided by you) of the corresponding problem; upon completion of the process, in the column *Status* the respective cell changes from *NOT SUBMITTED* to *SUCCESS*, whereas clicking the *Download Zip* black button in column *Chebyshev* initiates downloading the archive containing the Chebyshev scalarization(s) file(s). Uploading set(s) of weights or y^* is



Problems

Id	Input LP	Submit	Update	Weights	Y*	Chebyshev	Status	Delete
14	Project_portfolio	Make Chebyshev	Update problem	-	-		Click the "Make Chebyshev" button	Delete
17	portfel_projektów_bicrit	Make Chebyshev	Update problem	-	-	Download Zip	SUCCESS	Delete
18	portfel_projektów_bicrit	Make Chebyshev	Update problem	+	-		NOT SUBMITTED	Delete
19	portfel_projektów_bicrit	Make Chebyshev	Update problem	-	+	Download Zip	SUCCESS	Delete
20	portfel_projektów_bicrit	Make Chebyshev	Update problem	+	+		NOT SUBMITTED	Delete
21	mknapcb_4_1_tricrit_multi	Make Chebyshev	Update problem	-	-		STARTED	Delete
22	simple_plant_location_ampl	Make Chebyshev	Update problem	-	-		FAILURE	Delete

Figure 1: An example of IDOL *Problems* page.

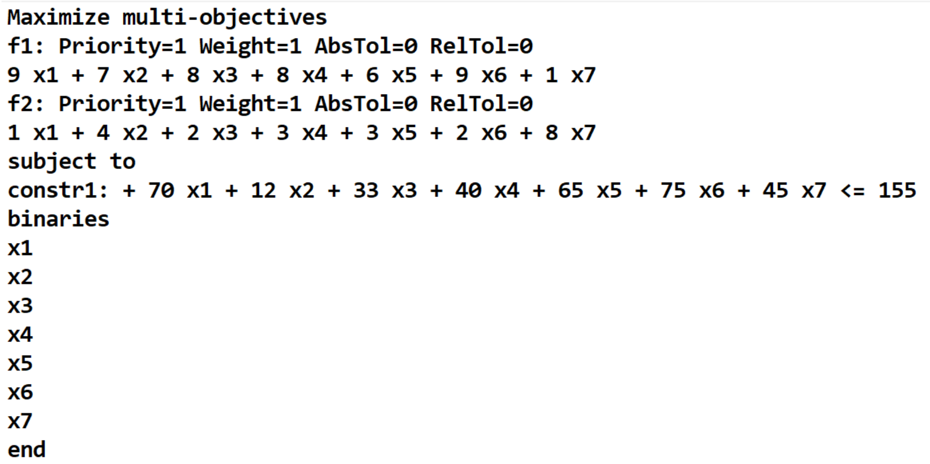
initiated by clicking the blue *Update problem* button in the *Update* column.

If it happens that in column *Status* the message *Click the "Make Chebyshev" button* appears, click *Make Chebyshev* button to rerun the Chebyshev scalarization.

4 A Use Case

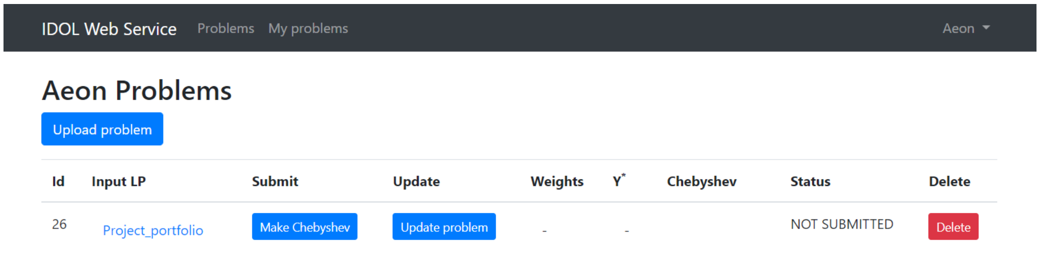
The functionalities of IDOL are illustrated with the problem LP file *Project_portfolio.lp* containing data of a small biobjective MIP problem (Figure 2).

Suppose that after logging in you opt for uploading the LP file to the individual *My problems* page without providing *Weights* and y^* files. After the upload the page may look as in Figure 3 (no other files are present). The respective cell in *Status* column shows *NOT SUBMITTED*.



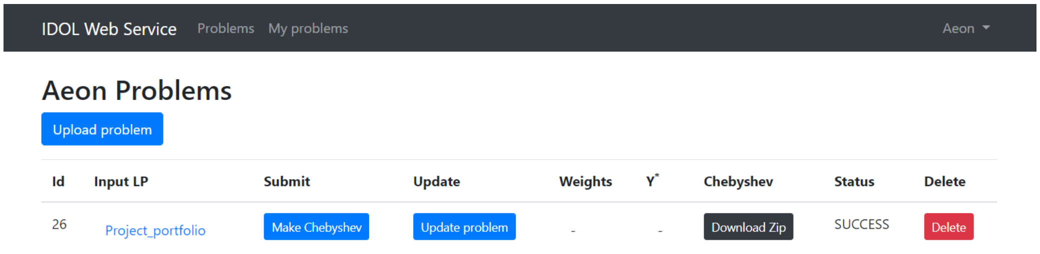
```
Maximize multi-objectives
f1: Priority=1 Weight=1 AbsTol=0 RelTol=0
9 x1 + 7 x2 + 8 x3 + 8 x4 + 6 x5 + 9 x6 + 1 x7
f2: Priority=1 Weight=1 AbsTol=0 RelTol=0
1 x1 + 4 x2 + 2 x3 + 3 x4 + 3 x5 + 2 x6 + 8 x7
subject to
constr1: + 70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
binaries
x1
x2
x3
x4
x5
x6
x7
end
```

Figure 2: The example problem LP file.



Id	Input LP	Submit	Update	Weights	Y^*	Chebyshev	Status	Delete
26	Project_portfolio	Make Chebyshev	Update problem	-	-		NOT SUBMITTED	Delete

Figure 3: *My problems* page after *Project_portfolio.lp* file has been uploaded.



Id	Input LP	Submit	Update	Weights	Y^*	Chebyshev	Status	Delete
26	Project_portfolio	Make Chebyshev	Update problem	-	-	Download Zip	SUCCESS	Delete

Figure 4: *My problems* page after the Chebyshev scalarization file has been produced.

Clicking *Make Chebyshev* button produces the Chebyshev scalarization file which is saved into *Chebyshev_Id.zip* archive, where Id is the unique identifier of the uploaded problem displayed in *Id* columnn. The archive can be downloaded by clicking the corresponding *Download Zip* button. The respective *Status* cell changes to *SUCCESS* (Figure 4). The Chebyshev scalarization file is represented in Figure 5.

Suppose now you want to upload two sets of weights

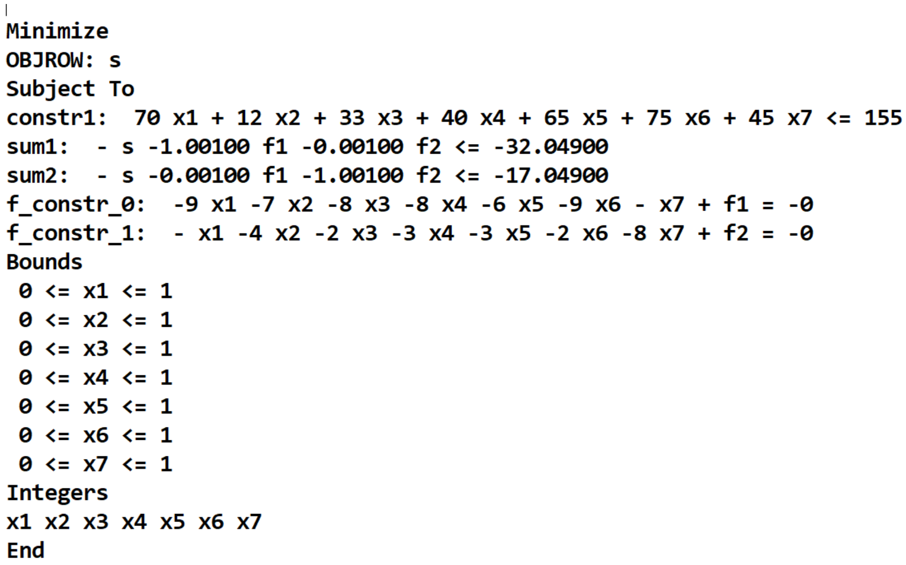
0.2 0.8
0.8 0.2

This is done by clicking *Update problem* button and uploading the file (Figure 6).

After the file is uploaded the status of the respective cell in the *Weights* column changes from ”-” (black) to ”+” (blue) and the page looks as in Figure 7. The blue color indicates that ”+” is now linked to *Weights* file.

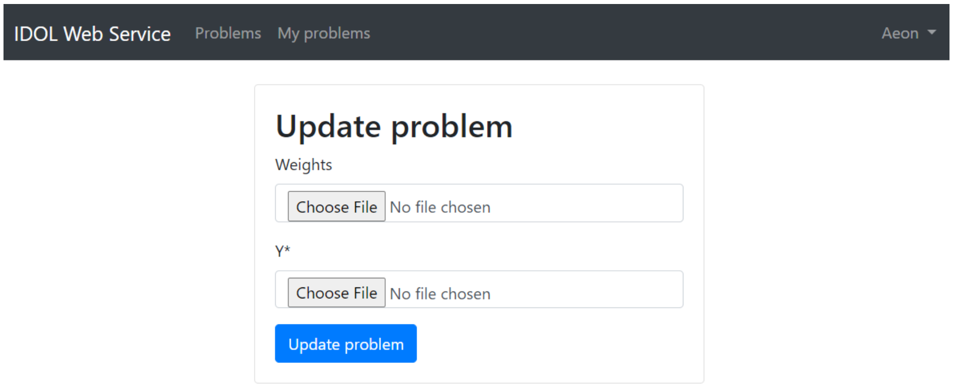
Clicking *Make Chebyshev* button produces two Chebyshev scalarization files, each with one of the two uploaded sets of weights, and the files are saved into *Chebyshev_Id.zip* archive.

The Chebyshev scalarization file produced with the first set of uploaded



```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -1.00100 f1 -0.00100 f2 <= -32.04900
sum2:  - s -0.00100 f1 -1.00100 f2 <= -17.04900
f_constr_0:  -9 x1 -7 x2 -8 x3 -8 x4 -6 x5 -9 x6 - x7 + f1 = -0
f_constr_1:  - x1 -4 x2 -2 x3 -3 x4 -3 x5 -2 x6 -8 x7 + f2 = -0
Bounds
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1
 0 <= x5 <= 1
 0 <= x6 <= 1
 0 <= x7 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7
End
```

Figure 5: The Chebyshev scalarization file to the example problem.

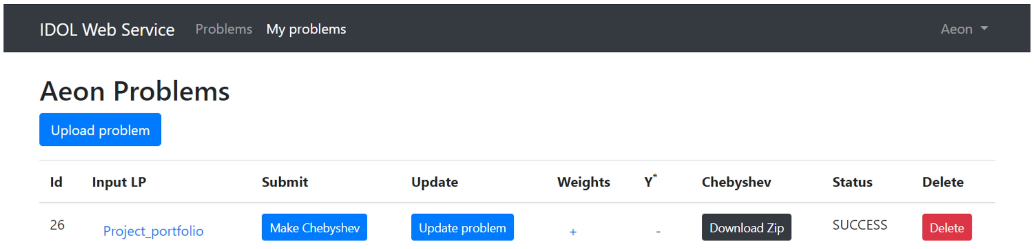


Update problem

Weights

Y*

Figure 6: The form to upload *Weights.txt* and *Y.txt* files.



Aeon Problems

Id	Input LP	Submit	Update	Weights	Y*	Chebyshev	Status	Delete
26	Project_portfolio	Make Chebyshev	Update problem	+	-	Download Zip	SUCCESS	Delete

Figure 7: *My problems* page after file *Weights* has been uploaded.

weights is represented in Figure 8, and the Chebyshev scalarization file produced with the second set of uploaded weights represented in Figure 9.

Suppose now you want to upload y^*

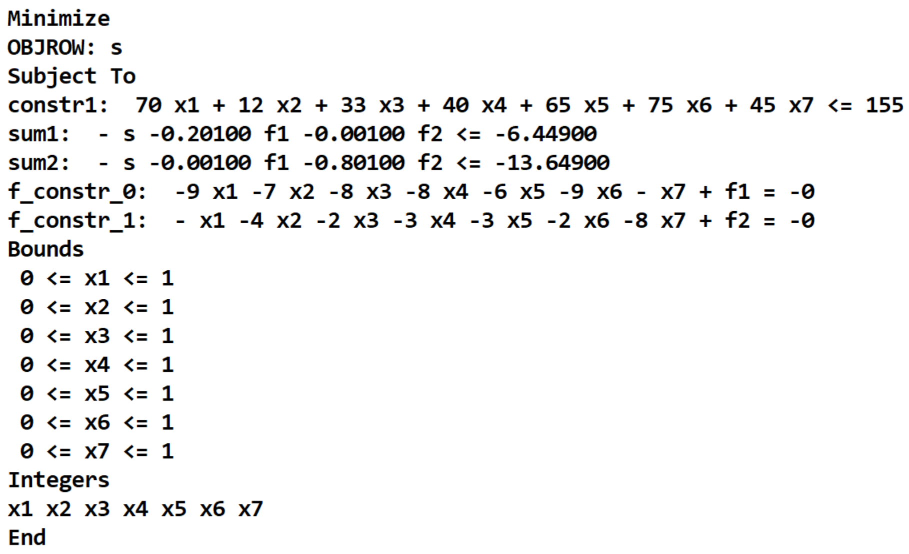
100 100

This is done by clicking *Update problem* button and uploading the respective file (Figure 6).

After the file is uploaded the status of the respective cell in the y^* column changes from "-" (black) to "+" (blue) and the page looks as in Figure 10. The blue color indicates that "+" is now linked to y^* file.

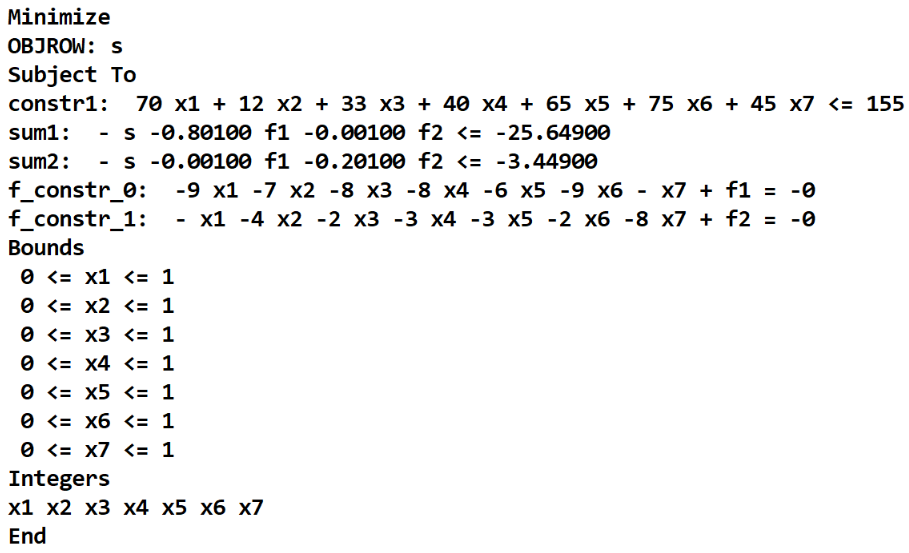
Clicking *Make Chebyshev* button produces two Chebyshev scalarizations each with one of the two previously uploaded sets of weights and the new y^*, and the files are put to *Download Zip* archive.

The Chebyshev scalarization file produced with the first set of uploaded weights and the uploaded y^* is represented in Figure 11, and the Chebyshev scalarization file produced with the second set of uploaded weights and the uploaded y^* is represented in Figure 12.



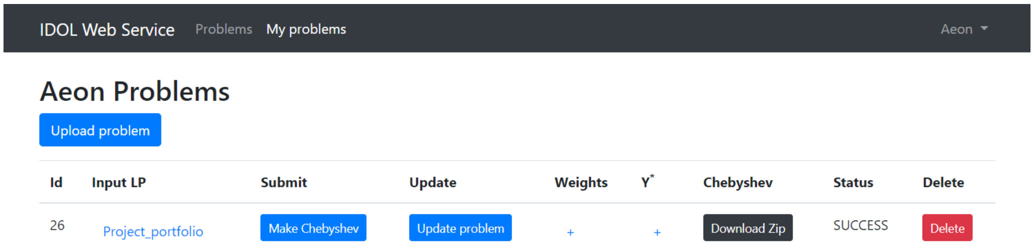
```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -0.20100 f1 -0.00100 f2 <= -6.44900
sum2:  - s -0.00100 f1 -0.80100 f2 <= -13.64900
f_constr_0:  -9 x1 -7 x2 -8 x3 -8 x4 -6 x5 -9 x6 - x7 + f1 = -0
f_constr_1:  - x1 -4 x2 -2 x3 -3 x4 -3 x5 -2 x6 -8 x7 + f2 = -0
Bounds
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1
 0 <= x5 <= 1
 0 <= x6 <= 1
 0 <= x7 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7
End
```

Figure 8: The Chebyshev scalarization file to the example problem with the first set of uploaded weights.



```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -0.80100 f1 -0.00100 f2 <= -25.64900
sum2:  - s -0.00100 f1 -0.20100 f2 <= -3.44900
f_constr_0:  -9 x1 -7 x2 -8 x3 -8 x4 -6 x5 -9 x6 - x7 + f1 = -0
f_constr_1:  - x1 -4 x2 -2 x3 -3 x4 -3 x5 -2 x6 -8 x7 + f2 = -0
Bounds
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1
 0 <= x5 <= 1
 0 <= x6 <= 1
 0 <= x7 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7
End
```

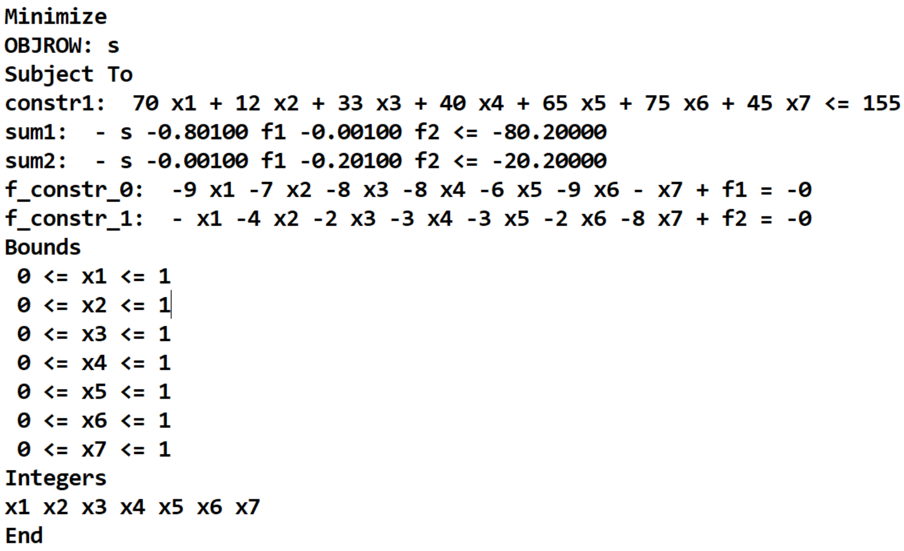
Figure 9: The Chebyshev scalarization file to the example problem with the second set of uploaded weights.



Aeon Problems

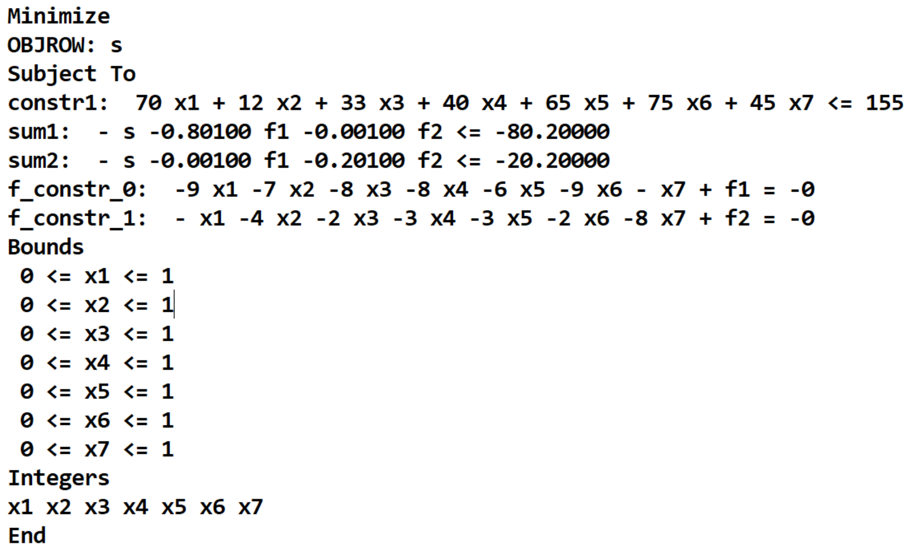
Id	Input LP	Submit	Update	Weights	Y^*	Chebyshev	Status	Delete
26	Project_portfolio	Make Chebyshev	Update problem	+	+	Download Zip	SUCCESS	Delete

Figure 10: *My problems* page after file *Y.txt* has been uploaded.



```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -0.80100 f1 -0.00100 f2 <= -80.20000
sum2:  - s -0.00100 f1 -0.20100 f2 <= -20.20000
f_constr_0:  -9 x1 -7 x2 -8 x3 -8 x4 -6 x5 -9 x6 - x7 + f1 = -0
f_constr_1:  - x1 -4 x2 -2 x3 -3 x4 -3 x5 -2 x6 -8 x7 + f2 = -0
Bounds
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1
 0 <= x5 <= 1
 0 <= x6 <= 1
 0 <= x7 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7
End
```

Figure 11: The Chebyshev scalarization file to the example problem with the first set of uploaded weights and the uploaded y^*.



```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -0.80100 f1 -0.00100 f2 <= -80.20000
sum2:  - s -0.00100 f1 -0.20100 f2 <= -20.20000
f_constr_0:  -9 x1 -7 x2 -8 x3 -8 x4 -6 x5 -9 x6 - x7 + f1 = -0
f_constr_1:  - x1 -4 x2 -2 x3 -3 x4 -3 x5 -2 x6 -8 x7 + f2 = -0
Bounds
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1
 0 <= x5 <= 1
 0 <= x6 <= 1
 0 <= x7 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7
End
```

Figure 12: The Chebyshev scalarization file to the example problem with the second set of uploaded weights and the uploaded y^*.

5 References

1. Gurobi Optimizer Reference Manual, 2020. As of September 1, 2020, https://www.gurobi.com/documentation/ .

2. Ehrgott, M. (2005). Multicriteria Optimization. Springer.

3. Kaliszewski, I. (2006). Soft Computing for Complex Multiple Criteria Decision Making. Springer, New York.

4. Miettinen, K.M. (1999). Nonlinear Multiobjective Optimization. Kluwer Academic Publishers.