IDOL – An Introduction

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 denote a solution and $X_0 \subseteq \mathbb{R}^n$ the set of feasible solutions. Then the general multiobjective optimization problem is defined as:

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

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. $\mathbb{R}^k$ is called the objective space.

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 $\bar{x} \prec x$. Solution $\bar{x}$ 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).

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*) $\lambda_l > 0,\ l = 1, \ldots, k$,

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

- $y_l^* > \max_{x \in X_0} f_l(x)$ if the maximum exists and $y_l^* \geq \sup_{x \in X_0} f_l(x)$ if the maximum does not exist, $l = 1, \ldots, k$,

- 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 the existence of y^*, which is rather an obvious clause when it comes to numerical optimization calculations.

By formulation (2), the multiobjective optimization problem which 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 λ_l.

An equivalent formulation to (2) is

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

which may be suitable for other solvers than those suitable for formulation (2).

3 IDOL Inferface

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

Once the necessary input data is assembled as an *.lp file, Gurobi style, this file can be loaded to IDOL.

Together with the *.lp file it is possible, but not necessary, 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 a space. If *Weights.txt* file is absent, IDOL uses default weights all equal 1.

Likewise, it is possible, but not necessary, to upload the *Reference.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 a space. If *Reference.txt* file is absent, IDOL establishes y^* maximizing each objective function separately. To calculate y^* fast, the optimal values of each objective function is solved with 1% tolerance.

The successive step is performed in iterations, one iteration for each set of weights. In each iteration, the Chebychev scalarization is built with 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, parameter ρ is hard coded to 0.001.

On the starting page of IDOL (Figure 1) you opt to sign up with your chosen user name and a password, and next to login in. Once logged in, you can change the password, log out, or go to one of two pages: *Problems* or *My problems*. The functionalities of both pages are the same except files listed on the page *Problems* are visible to all users whereas files listed on the *My problems* page are visible only to the signed individual user.

You can also opt not to sign in and still use the service but in this case only the *Problems* page is available for you.

On *Problems* page (see Figure 2 representing an example population of this page by various problems) you can upload LP files, upload *Weights* and *Reference* 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^* are provided by Henry ("+") 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 reference elements) of the corresponding problem; upon completion of the process in the column *Status* the respective cell changes from *NOT SUBMIT-*

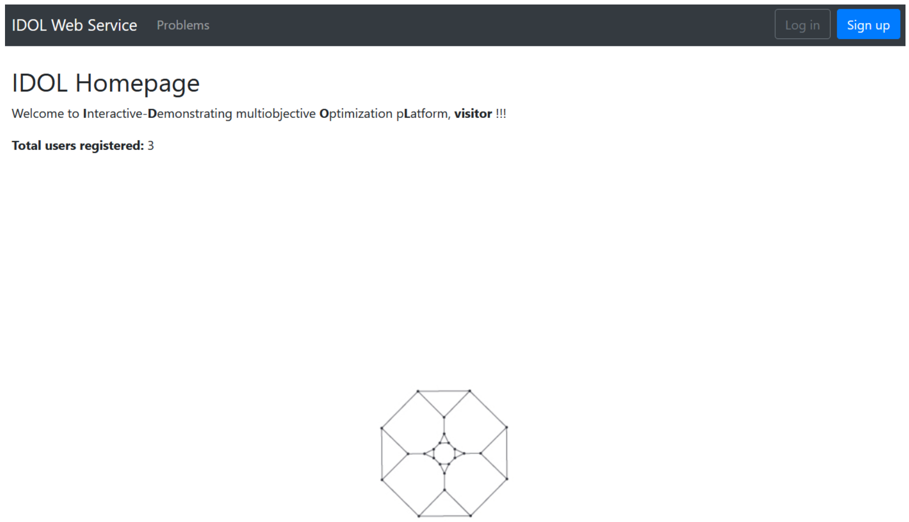


Figure 1: IDOL home page.

TED to *SUCCESS*, whereas clicking the respective black button *Download Zip* in column *Chebyshev* initiates downloading the archive containing the Chebyshev scalarization(s) file(s). Uploading set(s) of weights or y^* is initiated by clicking the corresponding blue button in the *Update* column.

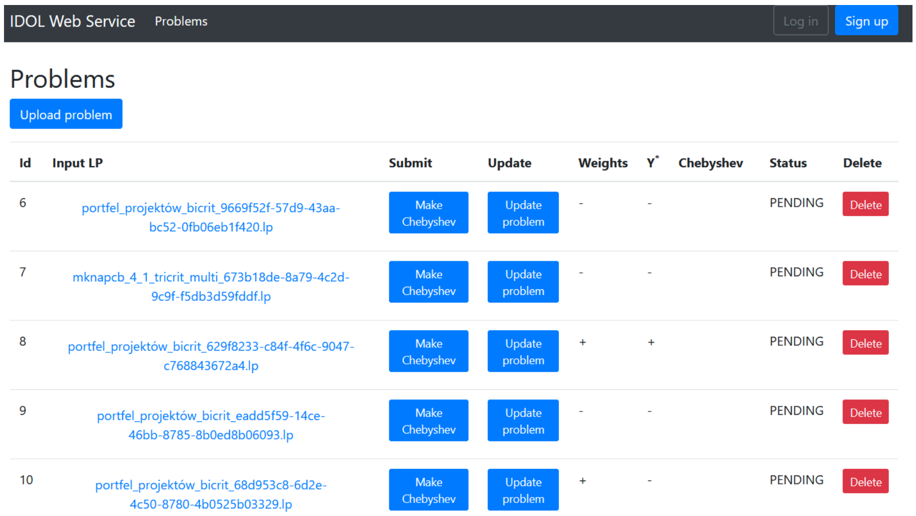
Page *My problems* replicates the logic of the *Problems* page.

If it happens that in column *Status* the message *Press 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 3).

Assume that after signing in and logging in, Henry opts for uploading the LP file to the individual *My problems* page without providing *Weights* and



IDOL Web Service Problems Log in Sign up

Problems

Upload problem

Id	Input LP	Submit	Update	Weights	Y*	Chebyshev	Status	Delete
6	portfel_projektów_bicrit_9669f52f-57d9-43aa-bc52-0fb06eb1f420.lp	Make Chebyshev	Update problem	-	-		PENDING	Delete
7	mknapcb_4_1_tricrit_multi_673b18de-8a79-4c2d-9c9f-f5db3d59fddf.lp	Make Chebyshev	Update problem	-	-		PENDING	Delete
8	portfel_projektów_bicrit_629f8233-c84f-4f6c-9047-c768843672a4.lp	Make Chebyshev	Update problem	+	+		PENDING	Delete
9	portfel_projektów_bicrit_eadd5f59-14ce-46bb-8785-8b0ed8b06093.lp	Make Chebyshev	Update problem	-	-		PENDING	Delete
10	portfel_projektów_bicrit_68d953c8-6d2e-4c50-8780-4b0525b03329.lp	Make Chebyshev	Update problem	+	-		PENDING	Delete

Figure 2: An example of IDOL *Problems* page.

```
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 3: The example problem LP file.

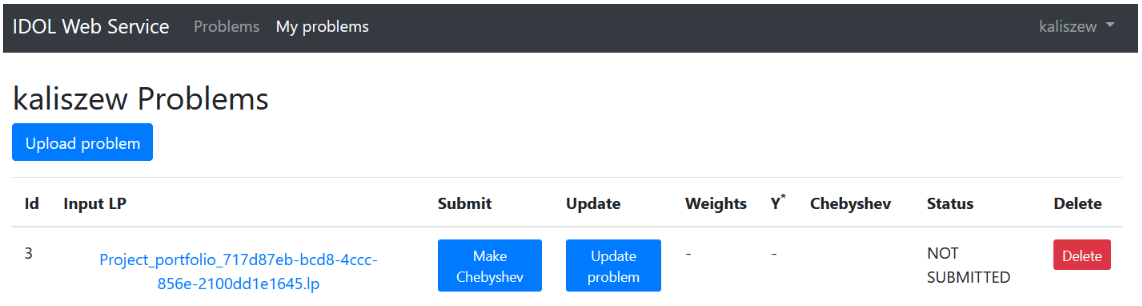


Figure 4: *My problems* page after *Project_portfolio.lp* file has been uploaded.

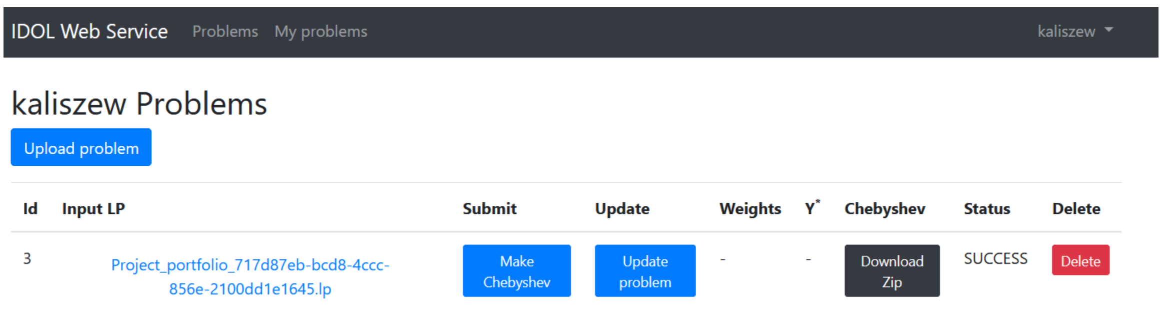


Figure 5: *My problems* page after the Chebyshev scalarization file has been produced.

Reference files. After the upload the page may look as in Figure 4 (no other files are present). The respective cell in *Status* column is *NOT SUBMITTED*.

Clicking *Make Chebyshev* button produces the Chebyshev scalarization which can be downloaded by clicking the corresponding *Download Zip* button. The respective *Status* cell changes to *SUCCESS* (Figure 5). The Chebyshev scalarization file is represented in Figure 6.

Suppose now you want to upload two sets of weights

0.2 0.8
0.8 0.2

```
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 6: The Chebyshev scalarization file to the example problem.

.

This is done by clicking *Update problem* button and uploading the file (Figure 7).

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 8. The blue color indicates that "+" is now linked to *Weights* file.

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

The Chebyshev scalarization file produced with the first set of uploaded weights is represented in Figure 9, and the Chebyshev scalarization file produced with the second set of uploaded weights represented in Figure 10.

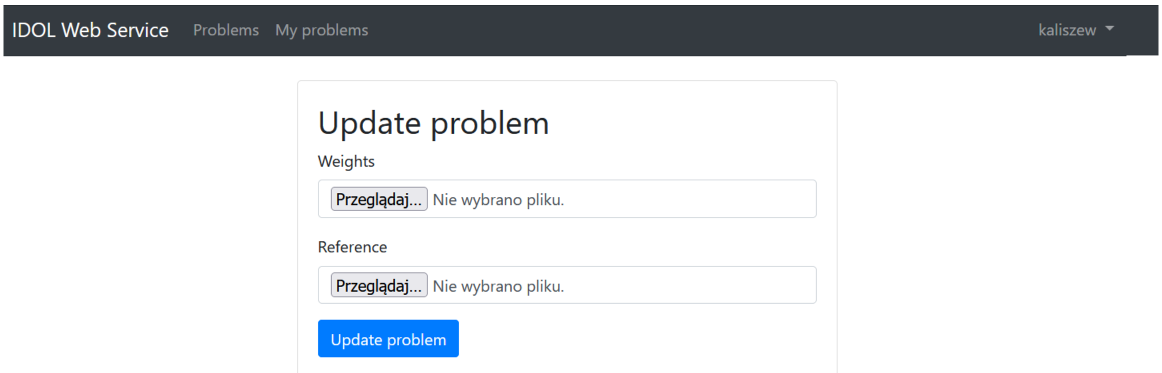


Figure 7: The form to give links to *Weights* file and *Reference* file.

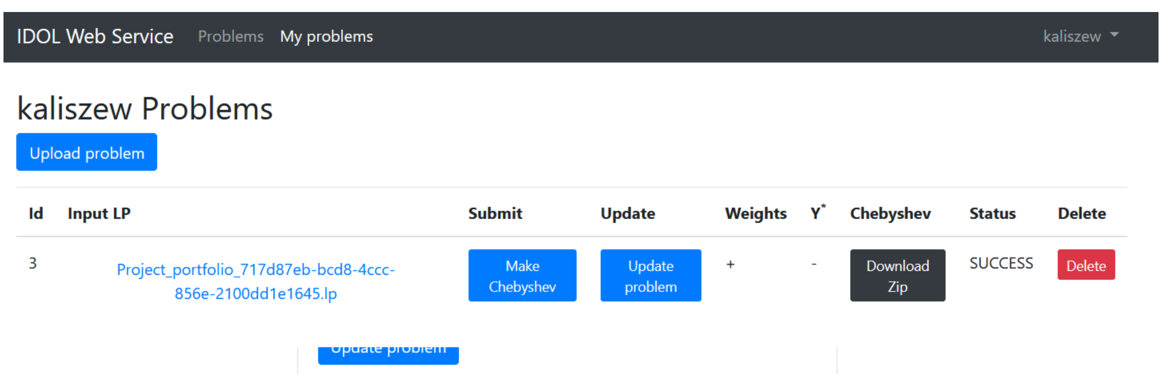


Figure 8: *My problems* page after file *Weights* has been uploaded.

```
Minimize
OBJROW: s
Subject To
constr1:  70 x1 + 12 x2 + 33 x3 + 40 x4 + 65 x5 + 75 x6 + 45 x7 <= 155
sum1:  - s -0.20100 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 9: 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 10: The Chebyshev scalarization file to the example problem with the second set of uploaded weights.

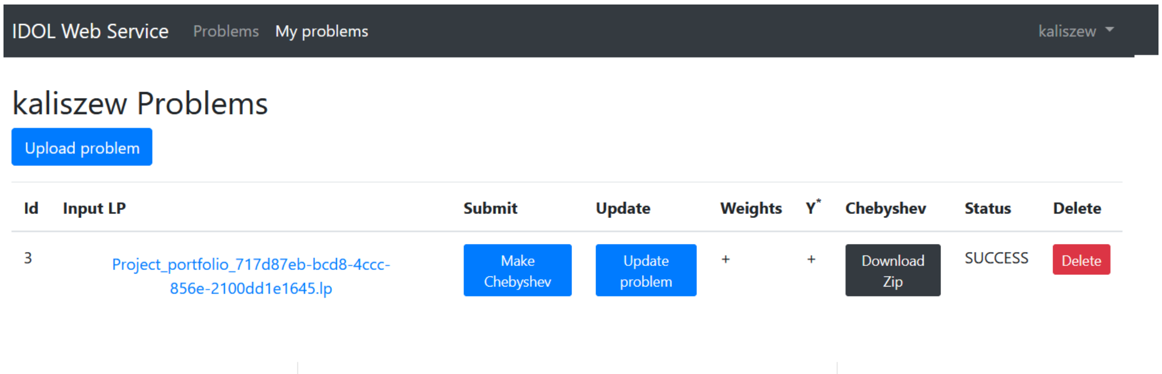


Figure 11: *My problems* page after file *Reference* has been uploaded.

Suppose now that you want to upload y^*

$$100\ 100$$

.

This is done by clicking *Update problem* button and uploading the respective file (Figure 7).

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 11. The blue color indicates that "+" is now linked to *Reference* 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 12, and the Chebyshev scalarization file produced with the second set of uploaded weights and the uploaded y^* is represented in Figure 13.

```
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 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 13: 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.