BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 3

Sofia • 2015 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0049

# Solving Multicriteria Optimization Problems with WebOptim Software System

*Boris Staykov*

*Institute of Information and Communication Technologies, 1113 Sofia*
*Email: bstaykov@iinf.bas.bg*

***Abstract:*** *This paper is a presentation of a web based decision support system "WebOptim" for solving single and multiple criteria optimization problems. It targets a wide range of user typeseducators, researchers, managers and business people. It also provides two types of communication interfaces user friendly graphical interface for human interaction and programming interface for machine communication with other third party software systems. The interfaces facilitate the problem solving process of different types of optimization problems, mainly single and multi-objective programming optimization problems with continuous or integer variables.*

***Keywords:*** *Multiple criteria optimization, web-based decision support system.*

## 1. Introduction

Computer aided decision support exists almost since the computers themselves [10]. Many real life problems in planning, controlling, analyzing and monitoring in economy, transportations, industrial production, infrastructures, ecology and other branches can be modeled as problems of multicriteria optimization [6, 8, 9]. One of the most important things today is the wise management of energy consumption and efficiency. Numerous scientific researches point out the need of continuous improvements in this area by means of automated process control and energy management systems [11]. This is a place where the use of decision support software systems is a must.

Multiple criteria optimization problems solving is applicable also in the widely used distributed computers and communication networks where the optimization itself is a critical task [13].

Defined by their nature, the optimization problems are related to huge mathematical computations that are practically impossible to be done in reasonable time without some kind of computer aid. Here comes the use of the software decision support systems.

At the beginning they were a single user type and implemented only one method or a few similar methods of one type. With the beginning of the Internet era, some network-based systems appeared [1]. But still most of them were problem-oriented and they could only solve one specific problem. For software implementation mainly Java applet technologies were used, but this approach has many limitations. This architecture was still basically client-side technology and it is limited by the hardware resources of the client machine, memory, data volume, CPU power and problem complexity.

Nowadays most of the software is network oriented and the most popular architecture is the WEB. When thinking about web, there are certain problems and limitations associated with it [2]. One of them is that the web architecture does not allow persistent communication between the client and the server. This automatically put limits on the response time and the amount of data exchanged between the servers and clients. When solving multiple criteria optimization problems, the computation time needed is highly undetermined which conflicts with these communication limitations. Despite of these problems, the tendency is that web technology takes place almost everywhere in all kinds of the software solutions. WebOptim itself is a web based software system that has its own solution of overcoming the communication limitations of the web architecture.

## 2. System structure

The technologies used for developing WebOptim system are entirely Microsoft specific:

- .NET 4.0
- Internet Information Services 7
- Microsoft SQL Server 2008R2.

WebOptim is built on a modular principle and contains the following main modules (Fig. 1):

- User interface
- User management and security module
- Solvers management and maintenance module
- Solvers
- Intermediate system for inter-modular communication
- Public API module for communication with other software systems.

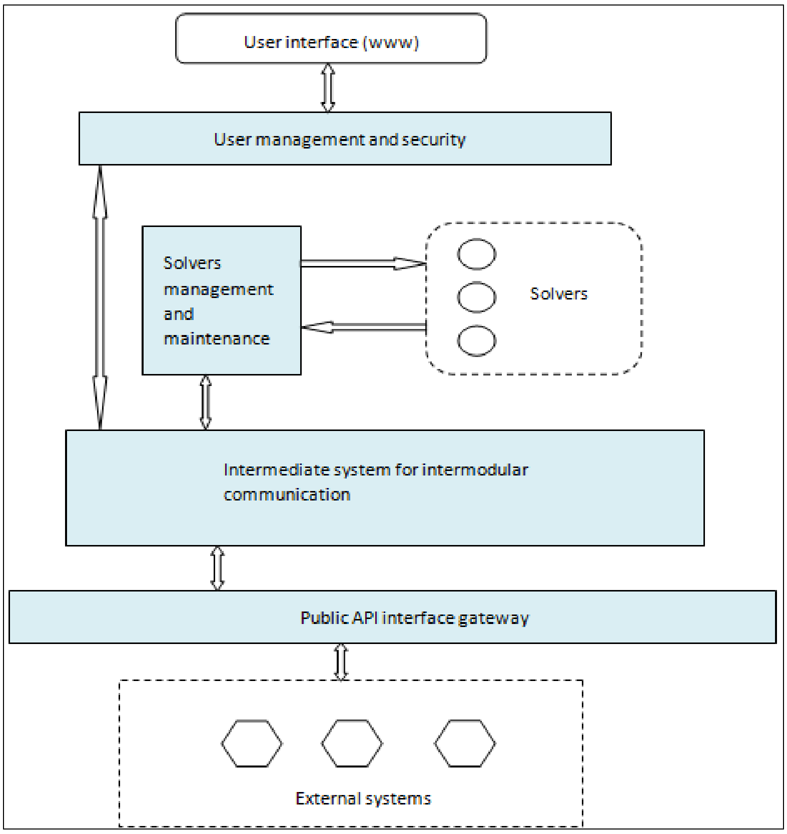
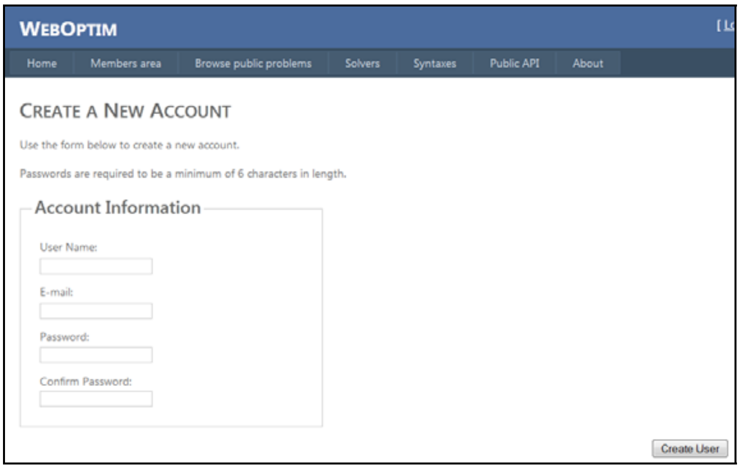

User interface (www)
User management and security
Solvers management and maintenance
Solvers
Intermediate system for intermodular communication
Public API interface gateway
External systems

Fig. 1. WebOptim system modules

### 2.1. User interface

The WebOptim system is a web application which means that its user interface presents entirely standard html elements that are supported by all contemporary web browsers on all operating systems. The most valuable advantage using this technology is that it automatically makes the whole system platform and operating system independent! All that the user needs in order to explore WebOptim. is a web browser and Internet connection.

The whole system is developed as MVC 4.0 Web Application [3]. The first step of using WebOptim system is the creation of a personal account. This is done by standard web forms with minimum information requirements – names, email and password (Fig. 2). In this way the user can create, store and solve optimization problems which are automatically associated with his profile. Any problem from the personal profile can be shared simply by making it public. Furthermore, the security module is responsible for taking care of system logging, permissions management and personal information maintenance.



Create a New Account

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

Account Information

User Name:

E-mail:

Password:

Confirm Password:

Fig. 2. WebOptim user registration

### 2.2. User management and security module

This part of the system manages the user profiles. This is where all user personal information is stored – login credentials, personal details, the optimization problems and their solutions, information about errors and specific problem solving details. Microsoft SQL relational database is used for information storage.

### 2.3. Solvers management and maintenance

The system architecture allows different methods and algorithms to be used for solving linear single or multi-objective optimization problems. With its common machine gateway interface, WebOptim enables using solvers written in different programming languages for different computer architectures. The solvers can be installed on one server together with WebOptim or on remote servers with the operating systems. This interface is realized by using standard protocols for data exchange – XML, WSDL, SOAP [5].

Solving optimization problems with a large number of variables and objective functions needs significant computer time. As a result, the solution cannot be obtained immediately. That is why each problem is stored in the database and marked as “sent to solver”. When the solution is ready, the solver returns the result of the problem based on its unique identification number and the status of the problem is changed to “solution obtained” or “error received”.

The users define their optimization problems by using one of the following standard linear optimization descriptions formats:

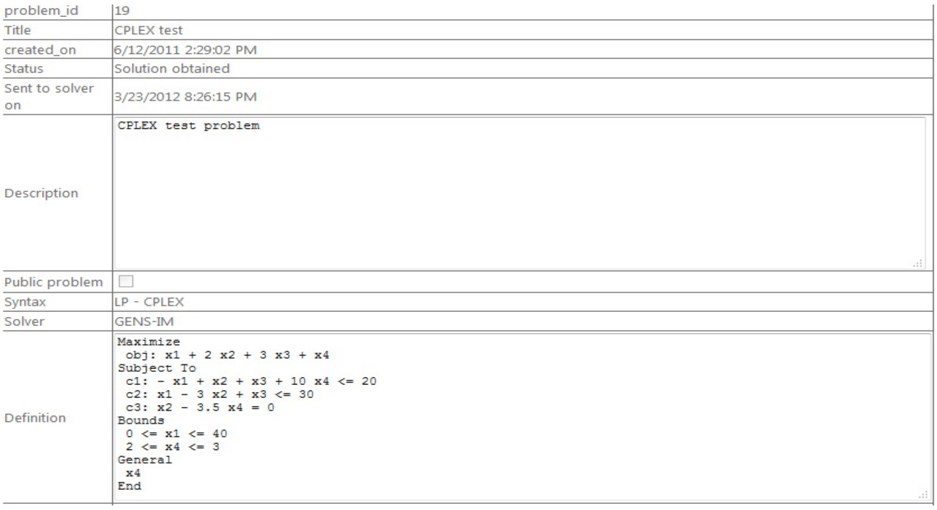
- MPS file format – Supported by most lp solvers, but it is old and difficult to be read by humans.
- LP format – Native format of the lpsolve open source library. Readable by humans and its syntax is very similar to the mathematical formulation of the problem.
- CPLEX – very readable and its syntax is similar to the mathematical formulation. Used mainly by the CPLEX solver.

- LINDO – developed by Lindo Systems and used for their solver. Human readable and close to the mathematical formulation.
- OSIL XML – This format is very recent and uses a XML layout.

## 2.4. Solvers

Currently WebOptim supports three solvers. Two of them are based on the open source LPSolve library [4].

The first is a solver responsible for solving single criteria linear optimization problems and it implements the GENS-IM algorithm. Figs 3 and 4 show the algorithm user interface. Again it consists of typical web forms elements where the user can define, solve and receive the solution of the problem.

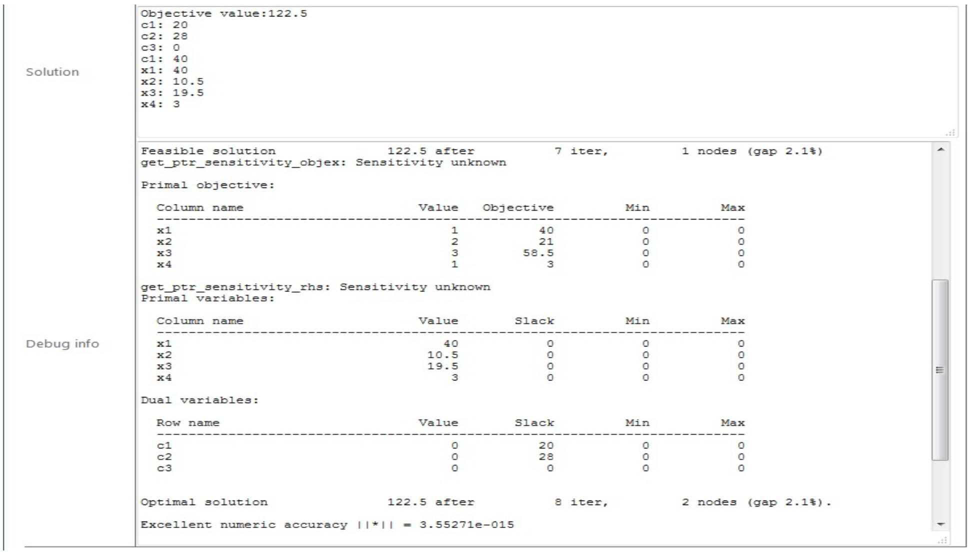


| | |
|---|---|
| problem_id | 19 |
| Title | CPLEX test |
| created_on | 6/12/2011 2:29:02 PM |
| Status | Solution obtained |
| Sent to solver on | 3/23/2012 8:26:15 PM |
| Description | CPLEX test problem |
| Public problem | □ |
| Syntax | LP - CPLEX |
| Solver | GENS-IM |

Definition

```
Maximize
 obj: x1 + 2 x2 + 3 x3 + x4
Subject To
 c1: - x1 + x2 + x3 + 10 x4 <= 20
 c2: x1 - 3 x2 + x3 <= 30
 c3: x2 - 3.5 x4 = 0
Bounds
 0 <= x1 <= 40
 2 <= x4 <= 3
General
 x4
End
```

Fig. 3. Solving single criterion optimization problems with GENS-IM solver – problem definition part



Solution

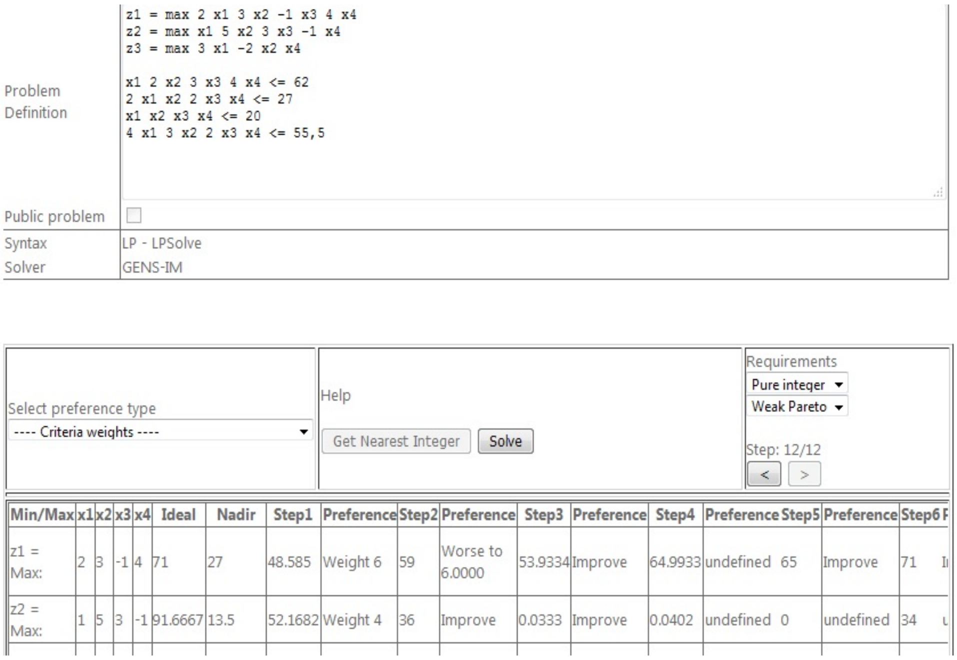
```
Objective value:122.5
c1: 20
c2: 28
c3: 0
c1: 40
x1: 40
x2: 10.5
x3: 19.5
x4: 3
```

Debug info

```
Feasible solution              122.5 after          7 iter,         1 nodes (gap 2.1%)
get_ptr_sensitivity_objex: Sensitivity unknown

Primal objective:

  Column name                    Value   Objective         Min         Max
  --------------------------------------------------------------------------
  x1                                 1          40           0           0
  x2                                 2          21           0           0
  x3                                 3        58.5           0           0
  x4                                 1           3           0           0

get_ptr_sensitivity_rhs: Sensitivity unknown
Primal variables:

  Column name                    Value       Slack         Min         Max
  --------------------------------------------------------------------------
  x1                                40           0           0           0
  x2                              10.5           0           0           0
  x3                              19.5           0           0           0
  x4                                 3           0           0           0

Dual variables:

  Row name                       Value       Slack         Min         Max
  --------------------------------------------------------------------------
  c1                                 0          20           0           0
  c2                                 0          28           0           0
  c3                                 0           0           0           0


Optimal solution               122.5 after          8 iter,         2 nodes (gap 2.1%).

Excellent numeric accuracy ||*|| = 3.55271e-015
```

Fig. 4. Solving single criterion optimization problems with GENS-IM solver – solution part

The second solver is again LPSolve based and implements the multicriteria GENS-IM algorithm. Its user interface is slightly different because this is an interactive multi-steps solver which requires input of additional problem specific information at each step (Fig. 5).



| | |
|---|---|
| Problem Definition | z1 = max 2 x1 3 x2 -1 x3 4 x4<br>z2 = max x1 5 x2 3 x3 -1 x4<br>z3 = max 3 x1 -2 x2 x4<br><br>x1 2 x2 3 x3 4 x4 <= 62<br>2 x1 x2 2 x3 x4 <= 27<br>x1 x2 x3 x4 <= 20<br>4 x1 3 x2 2 x3 x4 <= 55,5 |
| Public problem | □ |
| Syntax | LP - LPSolve |
| Solver | GENS-IM |

Select preference type: ---- Criteria weights ----

Help

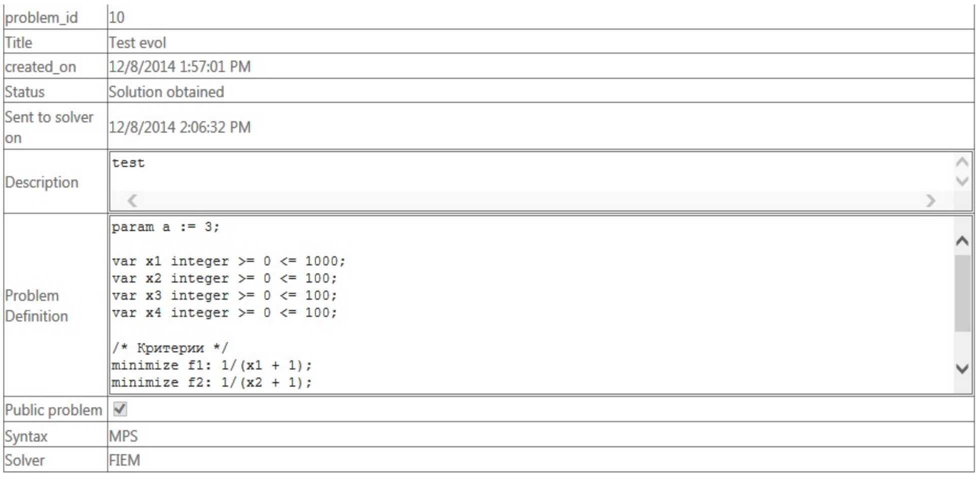
Requirements: Pure integer, Weak Pareto

Step: 12/12

| Min/Max | x1 | x2 | x3 | x4 | Ideal | Nadir | Step1 | Preference | Step2 | Preference | Step3 | Preference | Step4 | Preference | Step5 | Preference | Step6 | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| z1 = Max: | 2 | 3 | -1 | 4 | 71 | 27 | 48.585 | Weight 6 | 59 | Worse to 6.0000 | 53.9334 | Improve | 64.9933 | undefined | 65 | Improve | 71 | I |
| z2 = Max: | 1 | 5 | 3 | -1 | 91.6667 | 13.5 | 52.1682 | Weight 4 | 36 | Improve | 0.0333 | Improve | 0.0402 | undefined | 0 | undefined | 34 | u |

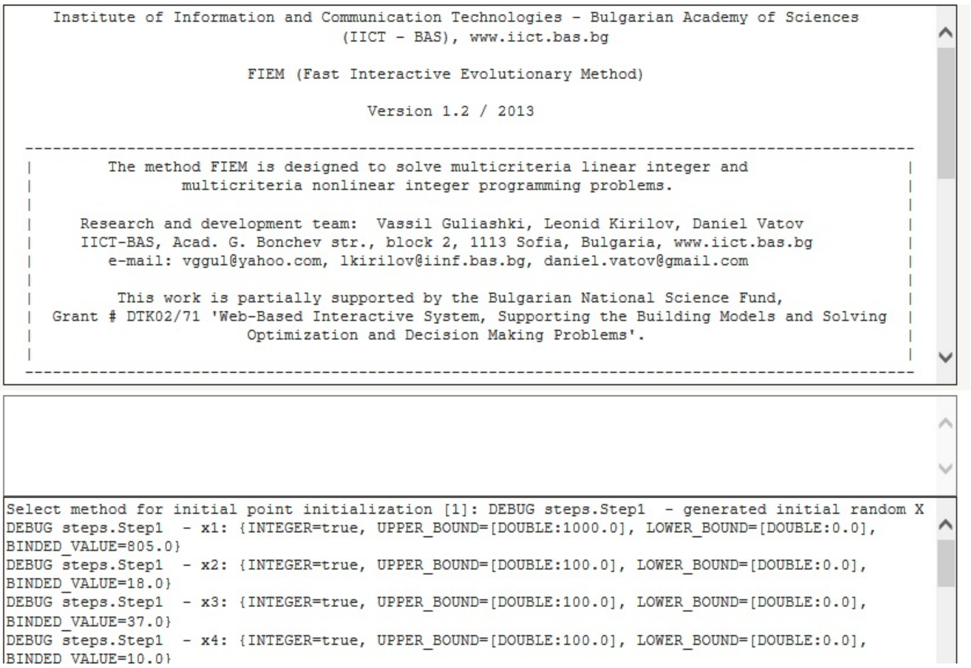
Fig. 5. Solving multicriteria optimization problems with GENS-IM solver

The third solver is most recent, evolutionary based and allows solving of non-linear multicriteria optimization problems. This is the FIEM solver [7] and it is still in a development and testing stage. The solving process with FIEM is also interactive and multistep (Figs 6, 7).



| | |
|---|---|
| problem_id | 10 |
| Title | Test evol |
| created_on | 12/8/2014 1:57:01 PM |
| Status | Solution obtained |
| Sent to solver on | 12/8/2014 2:06:32 PM |
| Description | test |
| Problem Definition | param a := 3;<br><br>var x1 integer >= 0 <= 1000;<br>var x2 integer >= 0 <= 100;<br>var x3 integer >= 0 <= 100;<br>var x4 integer >= 0 <= 100;<br><br>/* Критерии */<br>minimize f1: 1/(x1 + 1);<br>minimize f2: 1/(x2 + 1); |
| Public problem | ☑ |
| Syntax | MPS |
| Solver | FIEM |

Fig. 6. FIEM Solver – problem definition



```
    Institute of Information and Communication Technologies - Bulgarian Academy of Sciences
                                (IICT - BAS), www.iict.bas.bg

                        FIEM (Fast Interactive Evolutionary Method)

                                     Version 1.2 / 2013

 -----------------------------------------------------------------------------------------------
 |        The method FIEM is designed to solve multicriteria linear integer and               |
 |               multicriteria nonlinear integer programming problems.                        |
 |                                                                                            |
 |     Research and development team:  Vassil Guliashki, Leonid Kirilov, Daniel Vatov         |
 |     IICT-BAS, Acad. G. Bonchev str., block 2, 1113 Sofia, Bulgaria, www.iict.bas.bg        |
 |        e-mail: vggul@yahoo.com, lkirilov@iinf.bas.bg, daniel.vatov@gmail.com               |
 |                                                                                            |
 |         This work is partially supported by the Bulgarian National Science Fund,           |
 |  Grant # DTK02/71 'Web-Based Interactive System, Supporting the Building Models and Solving |
 |                       Optimization and Decision Making Problems'.                          |
 |                                                                                            |
 -----------------------------------------------------------------------------------------------
```

```
Select method for initial point initialization [1]: DEBUG steps.Step1  - generated initial random X
DEBUG steps.Step1  - x1: {INTEGER=true, UPPER_BOUND=[DOUBLE:1000.0], LOWER_BOUND=[DOUBLE:0.0],
BINDED_VALUE=805.0}
DEBUG steps.Step1  - x2: {INTEGER=true, UPPER_BOUND=[DOUBLE:100.0], LOWER_BOUND=[DOUBLE:0.0],
BINDED_VALUE=18.0}
DEBUG steps.Step1  - x3: {INTEGER=true, UPPER_BOUND=[DOUBLE:100.0], LOWER_BOUND=[DOUBLE:0.0],
BINDED_VALUE=37.0}
DEBUG steps.Step1  - x4: {INTEGER=true, UPPER_BOUND=[DOUBLE:100.0], LOWER_BOUND=[DOUBLE:0.0],
BINDED_VALUE=10.0}
```

Fig. 7. FIEM Solver – solving process

## 2.5. An intermediate system for inter-modular communication

WebOptim is designed to be open for extensions in the future. It means that new methods, solvers, problem types can be easily added at any time. This is achievable by using XML language for communication standard and web services. Here comes the role of an intermediate system module, which takes care of the communication between the end user interfaces, database storage and solvers. This module performs two main tasks: security and solvers information exchange. There are also two subtasks – sending requests to the solvers and handling solver responses for solutions of the problems. This is done by using a unified solver web service with SOAP communication [12].

## 2.6. Public API module

This module gives the program interface for machine communication with other external software systems. Its purpose is to provide functionalities for:

- Sending a problem for solving to the corresponding solver
- Sending a solution response to the external system (web service).

This is accomplished by using standard SOAP web services. Web services provide standard tools for inter-operability between different software applications, running on various platforms. The main advantage of the web services is that they allow integration between systems based on different program languages and different operating systems, regardless of their physical location.

Two web service interfaces ensure these module functionalities:

1. Problems receiving web service interface.

This web service endpoint allows remote posting of problems to the system by sending the following parameters:

- ProblemKey – Unique identifier for the problem
- Syntax – Name of the syntax that is used to define the problem
- Definition – Definition of the problem
- UserKey – Curently not used
- ReturnURL – URL address of the webservice endpoint where the solution will be returned. For security reasons the IP address of this URL address must match the sender IP address.

When the problem is posted, it goes to the target solver and waits for a solution response. After obtaining a solution (or an error), the role of the second public API interface s realized, that sends the response to the corresponding remote system where the problem comes from. This is done by using a target web service point that was sent with the "ReturnURL" parameter.

2. Solution postback service

The purpose of this module is to send the web service response to the remote system. It uses again a standard SOAP web services technology and sends the following response parameters:

- ProblemKey – Unique identifier for the problem (it has the same value that was sent by the remote system when posting the problem)
- Solution – Obtained solution (when no errors occurred).
- Verbose – Additional debug information.
- ErrorCode. This parameter has the following values and meanings:
  - 0 – Solution obtained
  - 1 – Solver does not support the given syntax
  - 2 – Parse (syntax) error
  - 3 – Internal solver error.

## 3. Solving multicriteria optimization problems

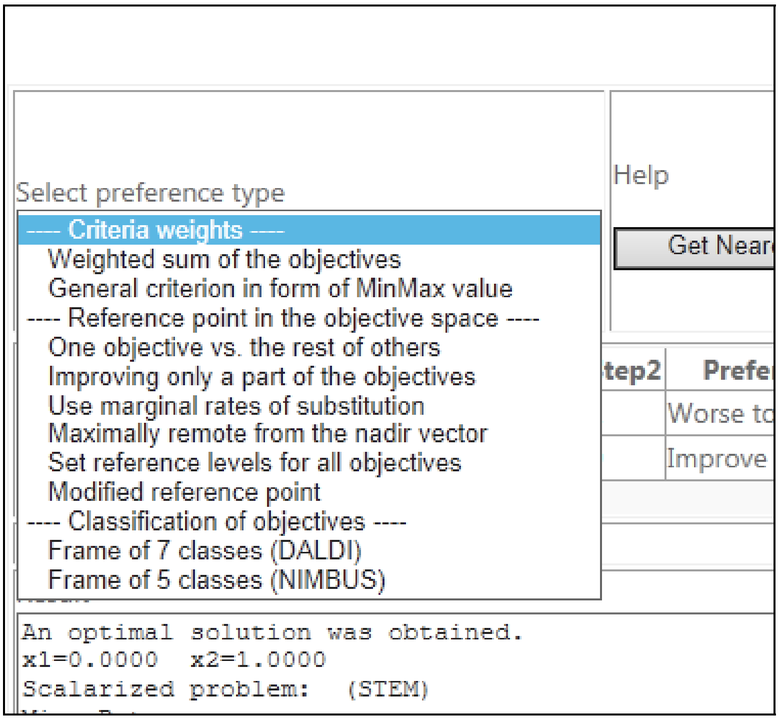
Decision making problems are non-formalized or weak formalized problems and their solution involves active participation of the so called Decision Makers (DM). The solutions depend on DM's preferences related to the current problem. It is an interactive process of finding solution, providing new preferences, finding another solution which satisfies these preferences.

For the purpose of this paper, an example using GENS-IM multicriteria solver is included.

The first step of the solution finding process is to choose how the user preferences will be provided to the system. The DM can choose among ten types of preferences and each of them implements a different method. Each method transforms the multiple criteria optimization problem to a single criterion one through the so called scalarization. Therefore, WebOptim provides a specific user interface for each preferences input choice.

There are three main classes for providing preferences to the system, which define what user interface input element will be provided (Fig. 8):

- By setting criteria weights
- By setting a reference point in the objective space
- Choosing between 7 classes of classification.


Select preference type
---- Criteria weights ----
Weighted sum of the objectives
General criterion in form of MinMax value
---- Reference point in the objective space ----
One objective vs. the rest of others
Improving only a part of the objectives
Use marginal rates of substitution
Maximally remote from the nadir vector
Set reference levels for all objectives
Modified reference point
---- Classification of objectives ----
Frame of 7 classes (DALDI)
Frame of 5 classes (NIMBUS)
Help
Get Near
tep2
Prefe
Worse to
Improve

```
An optimal solution was obtained.
x1=0.0000  x2=1.0000
Scalarized problem:  (STEM)
```

Fig. 8. Choosing the solving method (or how to provide the user preferences to the solver)

### 3.1. Providing preferences by criteria weights

The user must input certain weights for each criterion, which sets the importance of each objective function (Fig. 9). It includes two methods:

- *Weighted sum of the objectives:* Scalarizing problem of weighted sums.

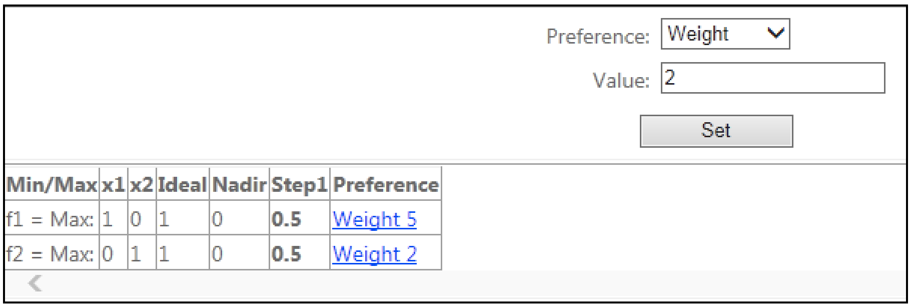
Input parameters: The DM sets weights for each objective function.

Output parameters: one supported Pareto optimal solution.

- *General criterion in the form of a min/max value:* Tchebycheff scalarizing problem (or augmented Tchebycheff scalarizing problem).

Input parameters: The DM sets weights for each objective function.

Output parameters: one (weak) Pareto optimal solution.



Preference: Weight

Value: 2

| Min/Max | x1 | x2 | Ideal | Nadir | Step1 | Preference |
|---|---|---|---|---|---|---|
| f1 = Max: | 1 | 0 | 1 | 0 | **0.5** | Weight 5 |
| f2 = Max: | 0 | 1 | 1 | 0 | **0.5** | Weight 2 |

Fig. 9. Providing preferences by setting criteria weights

### 3.2. Providing preferences by a reference point in the objective space

The user must define a reference point (or directions) for the next solution (Fig. 10). It includes six methods:

- *One objective vs. the rest*: Scalarizing problem of ε constraints method.

Input parameters: The DM sets one objective function for maximization (minimization) and lower (upper) bounds for the rest (ε-constraints).

Output parameters: one (weak) Pareto optimal solution.

- *Improving only a part of the objectives*: Scalarizing problem of STEM (STEP method).

Input parameters: The DM specifies the set of objectives to be relaxed and the amounts by which they are to be relaxed. It is looking for improvement of the rest of the objectives.

Output parameters: one (weak) Pareto optimal solution.

- *Using marginal rates of substitution*: Scalarizing problem of STOM (Satisfying Trade-Off Method).

Input parameters: The DM sets the reference point in the objective space. Its coordinates are the desired or acceptable objective values for the DM. The reference point must not dominate the ideal point.

Output parameters: one (weak) Pareto optimal solution.

- *Maximally remote from the Nadir vectors*: Scalarizing problem of GUESS method input parameters: The DM sets the reference point in the objective space. Its coordinates are the desired or acceptable objective values for the DM. The reference point must dominate the Nadir vector. Output parameters: one (weak) Pareto optimal solution.

- *Set reference levels for all objectives*: RP – achievement scalarizing problem (Reference Point method).

Input parameters: The DM sets the reference point in the objective space. Its coordinates are the desired or acceptable objective values for the DM. Output parameters: one (weak) Pareto optimal solution

- *Modified reference point*: MRP scalarizing problem.

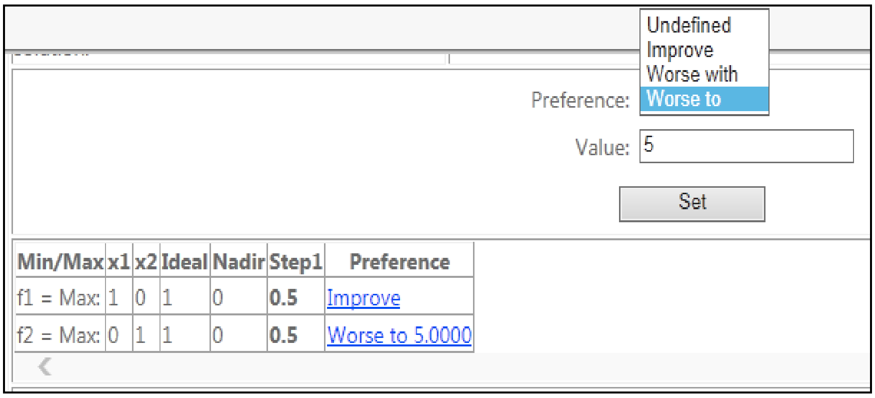
Input parameters: The DM classifies the objectives based on a frame of three classes:

1. The set of objectives to be improved by the desired values or towards desired aspiration levels

2. The set of objectives may be impaired by no more than a set value or towards a desired aspiration level.

3. The set of objectives to either preserve or improve their current values of the criteria.

Output parameters: one (weak) Pareto optimal solution.



Preference: Worse to

Value: 5

| Min/Max | x1 | x2 | Ideal | Nadir | Step1 | Preference |
|---|---|---|---|---|---|---|
| f1 = Max: | 1 | 0 | 1 | 0 | 0.5 | Improve |
| f2 = Max: | 0 | 1 | 1 | 0 | 0.5 | Worse to 5.0000 |

Fig. 10. Setting  the reference points or directions for each criterion

### 3.3. Providing preferences through classification of the objectives

Two methods are included in this section. Each criterion must be classified in one of the seven (DALDI) or five (NIMBUS) classes (Fig. 11).

- *Frame of seven classes* (*DALDI*). The DALDI scalarizing problem realizes the searching strategy “no great benefit-little loss”.
- *Input parameters*. The DM classifies each objective based on a frame of seven classes regarding its current value:

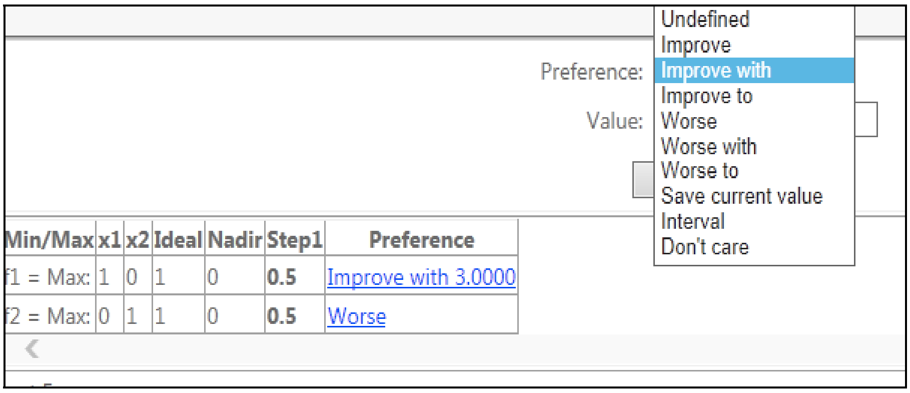
1. Improve as much as possible
2. Improve to a desired value or towards a desired aspiration level.
3. Preserve the current value.
4. Deteriorate the current value
5. Deteriorate by a desired values or towards a desired aspiration levels.
6. Keep the current value within the desired interval.
7. No limitations.

Output parameters: one (weak) Pareto-optimal solution.

- *Frame of five classes* (*NIMBUS*). The DM classifies each objective based on a frame of five classes regarding its current value:

1. Improve as much as possible.
2. Improve to a desired aspiration level.
3. Preserve the current value.
4. Keep the current value within a desired interval.
5. No limitations.

Output parameters: one (weak) Pareto optimal solution.



Preference: Improve with

Value:

| Min/Max | x1 | x2 | Ideal | Nadir | Step1 | Preference |
|---|---|---|---|---|---|---|
| f1 = Max: | 1 | 0 | 1 | 0 | **0.5** | Improve with 3.0000 |
| f2 = Max: | 0 | 1 | 1 | 0 | **0.5** | Worse |

Fig. 11. Classification of objectives

When the preferences are set, a new Pareto-optimal solution can be found. If it satisfies the decision maker preferences, it is taken as a final solution. If not – the process continues with setting new preferences and obtaining a new solution.

The goal of this interactive multistep solving process is to find a Pareto-optimal solution that satisfies the decision maker. Then it is taken as a final solution of the initial problem.

## 4. Conclusion

*WebOptim* system is a web-based software application designed to support decision makers in solving different multicriteria optimization problems. A key feature of the system is that it is designed to be easily expanded by using the latest web services technologies and communication protocols. Other important advantages of the system are:

- Targeted to a wide range of users from different professional fields and with different experience levels in the field of optimization.
- User friendly interface, accessible worldwide via the Web
- Implements one single criterion solver and two multicriteria interactive solvers
- Providing API interface for external use by third party developers

The future development of WebOptim will be concentrated on building new more efficient methods, algorithms and solvers for multicriteria optimization. The flexible design of the system makes it an ideal platform for implementing and testing such products.

## References

1. Bhargava, H. K., D. J. Power, D. Sun. Progress in Web-Based Decision Support Technologies. –Decision Support Systems, Vol. **43**, August 2007, Issue 4, pp. 1083-1095.
2. Bhargava, H. K., R. Krishnan, D. Kaplan. On Generalized Access to a www-Based Network of Decision Support Services. – In: Proc. of 3rd International Conference on DSS, Hong Kong, 22-23 June 1995.

3. Developing ASP.NET MVC4 Web Applications.
   **https://channel9.msdn.com/Series/Dev-ASP-MVC4-WebApps/01**
4. Introduction to lp_Solve 5.5.2.0.
   **http://web.mit.edu/lpsolve/doc/**
5. Paoli, J., T. Bray. The Annotated XML Specification. Internet Resource.
   **http://www.xml.com/pub/a/axml/axmlintro.html**
6. Kaleta, M., W. Ogryczak, E. Toczytowski, I. Zottowska. On Multiple Criteria Decision Support for Suppliers on the Competitive Electric Power Market. – In: Annals of Operations Research, Vol. **121**, 2003, No 1-4, pp. 79-104.
7. Kirilov, L., V. Guliashki. An Interactive Evolutionary Method FIEM for Solving Integer Multiple Objective Problems. – Compt. Rend. Acad. Bulg. Sci., Vol. **64**, 2011, No 2, pp. 203-212.
8. Korhonen, P. Multiple Objective Linear Programming in Supporting Forest Management. – In: Multiple Use of Forests and Other Natural Resources. F. Hellens, H. Andersen and L. Wichmann, Eds. Dordrecht, Kluwer, 1998, pp. 85-95. ISBN 1-4020-0277-7.
9. Monov, V. Energy Consumption and Efficiency in Industrial Processes. – In: John Atanasoff Celebration Days, Proceedings of International Conference “Robotics, Automation and Mechatronics’12”,15-17 October 2012, Sofia, pp. a9-a12.
10. Pardalos, P., M. Resende. Handbook of Applied Optimization. Oxford University Press, 2002. ISBN: 9780195125948.
11. Power, D. J. A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web. Internet Resource.
    **http://DSSResources.com/history/dsshistory.html**
12. Simple Object Access Protocol (SOAP) 1.1. Internet Resource.
    **http://www.w3.org/TR/2000/NOTE-SOAP-20000508/**
13. Tashev, T. D., V. M. Vorobiov. Generalized Net Model for Non-Conflict Switch in Communication Node. – In: Proc. of International Workshop “Distributed Computer and Communication Networks-DCCN’2007”, 10-12 September 2007, Russia, Moscow, IPPI Publ., 2007, pp.158-163.