Applied Mathematics, 2012, 3, 1572-1582
http://dx.doi.org/10.4236/am.2012.330217 Published Online October 2012 (http://www.SciRP.org/journal/am)

Scientific Research

Reactive Search Optimization; Application to Multiobjective Optimization Problems

Amir Mosavi[1], **Atieh Vaezipour**[2]
[1]Faculty of Informatics, University of Debrecen, Debrecen, Hungary
[2]Jönköping University, Jönköping, Sweden
Email: amir.allen.hu@gmail.com

Received August 2, 2012; revised September 25, 2012; accepted October 4, 2012

ABSTRACT

During the last few years we have witnessed impressive developments in the area of stochastic local search techniques for intelligent optimization and Reactive Search Optimization. In order to handle the complexity, in the framework of stochastic local search optimization, learning and optimization has been deeply interconnected through interaction with the decision maker via the visualization approach of the online graphs. Consequently a number of complex optimization problems, in particular multiobjective optimization problems, arising in widely different contexts have been effectively treated within the general framework of RSO. In solving real-life multiobjective optimization problems often most emphasis are spent on finding the complete Pareto-optimal set and less on decision-making. However the complete task of multiobjective optimization is considered as a combined task of optimization and decision-making. In this paper, we suggest an interactive procedure which will involve the decision-maker in the optimization process helping to choose a single solution at the end. Our proposed method works on the basis of Reactive Search Optimization (RSO) algorithms and available software architecture packages. The procedure is further compared with the excising novel method of Interactive Multiobjective Optimization and Decision-Making, using Evolutionary method (I-MODE). In order to evaluate the effectiveness of both methods the well-known study case of welded beam design problem is reconsidered.

Keywords: Stochastic Local Search; Real-Life Application; Multi Criteria Decision Making; Multiobjective Optimization; Reactive Search Optimization

1. Introduction

In the modern-day of optimal design and decision-making, optimization plays the main role [1,2]. Yet most studies in the past concentrated in finding the optimum corresponding to only a single design objective. However in the real-life design problems there are numerous objectives to be considered at once.

Efficient multiobjective optimization algorithms facilitate the decision makers to consider more than one conflicting goals simultaneously. Some examples of such algorithms and potential applications could be found in [3-7].

Within the known approaches to solving complicated optimal design problems there are different ideologies and considerations in which any decision-making task would find a fine balance among them.

1.1. Statement of the General form of the Multiobjective Optimization Problems

According to [8,9] the general form of the Multiobjective optimization problems is stated as; Minimize $\boldsymbol{f}(\boldsymbol{x})=\{f_1(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x})\}$, Subjected to $\boldsymbol{x}\in\Omega$, where $\boldsymbol{x}\in\mathbb{R}^n$ is a vector of n decision variables; $\boldsymbol{x}\subset\mathbb{R}^n$ is the feasible region and is specified as a set of constraints on the decision variables; $\boldsymbol{f}:\Omega\to\mathbb{R}^m$ is made of m objective functions subjected to be minimization. Objective vectors are images of decision vectors written as $\boldsymbol{z}=\boldsymbol{f}(\boldsymbol{x})=\{f_1(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x})\}$. Yet an objective vector is considered optimal if none of its components can be improved without worsening at least one of the others. An objective vector $\boldsymbol{z}$ is said to dominate $\boldsymbol{z}'$, denoted as $\boldsymbol{z}\prec\boldsymbol{z}'$, if $z_k\le z'_k$ for all k and there exist at least one h that $z_h\le z'_h$. A point $\hat{\boldsymbol{x}}$ is Pareto optimal if there is no other $\boldsymbol{x}\in\Omega$ such that $\boldsymbol{f}(\boldsymbol{x})$ dominates $\boldsymbol{f}(\hat{\boldsymbol{x}})$.

The task of optimal design, described above, is devided into two parts: 1) An optimization procedure to discover trad-off conflicting goals of design; 2) A decision-making process to choose a single preferred solution from them. Although both processes of optimization and decision-making are considered as two joint tasks, yet they are often treated as a couple of independent activities. For instance evolutionary multiobjective optimization (EMO) algorithms [10,11] have mostly concentrated on the optimization aspects, developing efficient methodologies of finding a set of Pareto-optimal solutions in a problem. However finding a set of trade-off optimal solutions is just half the process of optimal design. This has been the reason why EMO researchers were looking to find ways to efficiently integrate both optimization and decision making tasks in a user-friendly and flexible way [12].

On the other hand, in multiple criteria decision making (MCDM) algorithms [13,14] often the single optimal solution is chosen by collecting the decision-maker preferences where multiobjective optimization and decision making tasks are combined for obtaining a point by point search approach [13,15,16]. In addition in multiobjective optimization and decision-making, the final obtained solutions must be as close to the true optimal solution as possible and the solution must satisfy the preference information. Towards such a task, an interactive analysis tool to try various preferences to arrive at a solution is essential. This fact has motivated some researches to carry on the important task of integration between multiobjective optimization and multiple criteria decision-making [10,12,17,18]. Although in multiobjective optimi- zation, interactions with a decision-maker can come ei- ther during the optimization process, such as in the interactive EA optimization [11], or during the decision-making process [19,20]. As a multiobjective optimization task is not complete without the final decision-making activity and in this sense there exist a number of interacttive multiobjective optimization methods in the MCDM literature [13,16,18,21-23].

2. Combination of EMO and MCDM

According to [10] there are two different ways by which EMO and MCDM methodologies can be combined together; either EMO followed by MCDM or, MCDM integrated in an EMO. In the first way, an EMO algorithm is applied to find a set of Pareto-optimal solutions. Then, a single preferred solution is chosen from the obtained set by using a MCDM procedure. The former EMO application helps a decision maker (DM) to analyze different trade-off solutions before choosing a single solution. However the DM has to analyze many different solutions to be able to make the final decision. In this case if the DM has some preferences before hand, it is not possible to use these information in this procedure. Therefore the DM is forced to analyze too many solutions at the end.

In the second way, a MCDM procedure is integrated within an EMO to find a preferred set of Pareto-optimal solutions. In this way, the search is concentrated on an important region of the Pareto-optimal front. This allows the optimization task to value the preferences of the DM. The research and publications on interactive evolutionary algorithms (EA) and applications are numerous. The researches in the field and various problem domains in which an EA simulation is carried out by the involvement of the DM reviewed by Takagi [24]. Additionaly a summary can be found in the text by Miettinen [14]. Some of the popular approaches are interactive surrogate worth trade-off method [25], the reference point method [26] and the NIMBUS approach [13]. Each method is different from each other, but all have a typical characteristic. All procedures require a DM to provide critical information about the preferences in different ways [10]. This information is utilized to constitute a simpler search problem. A point-by-point search methodology is then used to find the optimum of the derived single objective task. This procedure is repeated many times until the DM is satisfied with the obtained final solution.

In [27] an EMO procedure is applied to a complicated design problem and then a human-interactive methodology is employed to choose a single solution. Moreover in [17] the reported human-driven qualitative and functionality-driven quantitative objectives, which come up with trade-off solutions in a multiobjective optimization set-up, are reviewed. In [18], EMO is combined with MCDM procedures, and an interactive procedure is suggested. This method is called I-MODE, which stands for interactive multiobjective optimization and decision-making using evolutionary methods. The work later in [18] was extended by involving more MCDM tools and integrations with the other software packages such as MATLAB, for providing better working on more real-world case studies [10]. In the integrated interactive procedure of I-MODE the EMO methodologies are combined with a certain and efficient MCDM technique to constitute a search-cum-decision-making procedure systematically. In the I-MODE procedure unlike the classical interactive methods, e.g. the one presented in [13], in addition to providing the ideal and nadir points of the problem, a good estimation of the shape of the Pareto-optimal frontier is created, in which helps to concentrate on a particular region on the front. Furthermore in the research on I-MODE procedure [18] it is concluded that when a scheme is best suited for one problem it may be inadequate in another problem. Thus, while developing such an integration, a procedure may involve more than one optimization and decision making tool in it so that any optimization and any decision-making scheme may be combined to constitute a viable problem solving procedure [28]. The research on I-MODE, procedure, and applications has motivated other EMO and MCDM researches, including our article, to develop such integration schemes further.

3. The Proposed Approach; RSO Integrated Visualization; an Effective Approach to MCDM

Visualization is an effective approach in the operations research and mathematical programming applications to explore optimal solutions, and to summarize the results into an insight, instead of numbers [29,30]. Fortunately during past few years, it has been a huge development in combinatorial optimization, machine learning, intelligent optimization, and reactive search optimization (RSO) [8,9,21], which have moved the advanced visualization methods forward [30]. Previous work in the area of visualization for MCDM [30] allows the DM to better formulate the multiple objective functions for large optimization runs. Alternatively in our research utilizing RSO and visualization [21], which advocates learning for optimizing, the algorithm selection, adaptation and integration, are done in an automated way and the user is kept in the loop for subsequent refinements. Here one of the crucial issue in MCDM is to critically analyzing a mass of tentative solutions, which is visually mined to extract useful information [31-33]. In developing RSO in terms of learning capabilities there has been a progressive shift from the decision maker to the algorithm itself, through machine learning techniques [8].

Concerning solving the MCDM problems, utilizing RSO, the final user is not distracted by technical details, instead concentrates on using his expertise and informed choice among the large number of possibilities. Algorithms with self-tuning capabilities like RSO make life simpler for the final user. And to doing so the novel approach of RSO is to integrat the machine learning techniques, artificial intelligence, reinforcement learning and active or query learning into search heuristics. According to the original literature [21] during a solving process the alternative solutions are tested through an online feedback loop for the optimal parameters' tuning. Therefore the DM would deal with the diversity of the problems, stochasticity, and dynamicity more efficiently. The RSO approach of learning on the job is contrasted with off-line accurate parameter tuning [22,23] which automatically tunes the parameter values of a *stochastic local search* (*SLS*) *algorithm*.

4. Stochastic Local Search

The aim of local search is to find the minimum of a combinatorial optimization function *f*, so called fitness function, on a set of discrete possible input values *X*. To doing so the focus would be on a local search, hinting at RSO with internal self-tuning mechanisms, and *Brain-Computer Optimization* (*BCO*), with a DM in an interacttive problem-solving loop. Accordingly in this context the basic problem-solving strategy would consists of starting from an initial tentative solution modifying the optimization function. According to [9] local search starts from an acceptable configuration $X(0)$ and builds a search trajectory $X^{(0)},\cdots,X^{(t+1)}$. Where *X* is the search space, $X^{(t)}$ is the current solution at iteration *t* time. $N\left(X^{(T)}\right)$ is the neighborhood of point $X^{(t)}$, obtained by applying a set of basic moves $\mu_0,\mu_1,\cdots,\mu_M$ to the current configuration:

$$N\left(X^{(t)}\right)=\left\{X\in x \text{ Such that } X=\mu_i\left(X^{(t)}\right), i=0,\cdots,M\right\}.$$

If the search space is given by binary strings with a given length $L: X=\{0,1\}^L$, the moves can be those changing the individual bits, and therefore *L* is equal to the string length *M*. The successor of the current point is a point in the neighborhood with a lower value of the function *f* to be minimized. If no neighbor has this property, *i.e.* if the configuration is a local minimizer, the search stops [8].

$$X^{(t+1)}=\begin{cases}Y & \text{if } f(Y)<X^{(t)}\\ X^{(t)} & \text{otherwise}(\text{search stops})\end{cases} \quad [9]$$

IMPROVING-NEIGHBOR returns an improving element in the neighborhood. Here `the local search works very effectively. This is mainly because most combinatorial optimization problems have a very rich internal structure relating the configuration *X* and the *f* value [8]. The analogy when the input domain is given by real numbers in $\mathbb{R}^n$ is that of a continuously differentiable function $f(x)$ with continuous derivatives. In the neighborhood the vector containing the partial derivatives is the gradient, and the change of *f* after a small displacement is approximated by the scalar product between the gradient and the displacement [34].

5. From Local Search to RSO by Learning

In problem-solving methods of Stochastic Local Search, where the free parameters are tuned through a feedback loop, the user is considered as a crucial learning component in which different options are developed and tested until acceptable results are obtained. As suggested in [8] by inserting the Machine Learning the human intervenetion is eliminated by transfering intelligent expertise into the algorithm itself. Yet in order to optimize the outcome setting the parameters and observing the outcome, a simple loop is performed where the parameters in a strategic and intelligent manner changed until a suitable solution is identified. Additionally to operate efficiently, RSO uses memory and intelligence, to recognize ways to improve solutions in a directed and focused manner.

5.1. Brain-Computer Optimization (BCO): The User in the Loop

In the RSO approach of problem solving the brain-computer interaction is simplified. This is done via learning-optimizing process which is basically the insertion of the machine learning component into the solution algorithm. The strengths of RSO are associated to the human brain characteristics *i.e.* learning from the past experience, learning on the job, rapid analysis of alternatives, ability to cope with incomplete information, quick adaptation to new situations and events [8,9]. The term of intelligent optimization in RSO refers to the online and offline schemes based on the use of memory, adaptation, incremental development of models, experimental algorithmics applied to optimization, intelligent tuning and design of heuristics. In this context with the aid of advanced visualization tools implemented within the software architecture packages the true meaning of numbers, and the conveyed information, are considered for better solutions. Therefore the integration of visualization and automated problem solving and optimization would be the center of attention.

6. Characteristics of the Proposed Approach

During the process of solving the real-world problems exploring the search space, utilizing RSO, many alternative solutions are tested and as the result adequate patterns and regularities appear. While exploring, the human brain quickly learns and drives future decisions based on the previous observations and searching alternatives. For the reason of rapidly exploiting the most promising solutions the online machine learning techniques are inserted into the optimization engine of RSO [9]. Furthermore with the aid of inserted machine learning a set of diverse, accurate and crucial alternatives are offered to the DM. The complete series of solutions are generated rapidly, better and better ones are produced in the following search phases. After the exploration of the design space, making the crucial decisions, within the multiple existing criteria, totally depends on several factors and priorities which are not always easy to describe before starting the solution process. In this context the feedbacks from the DM in the preliminary exploration phase can be incurporated so that a better tuning of the parameters takes the preferences into account [8]. Some relevant characteristics of RSO within the context of local search based processes, according to [9], could be summerized as learning on the job, rapid generation, and analysis of many alternatives, flexible decision support, diversity of solutions and anytime solutions.

7. Applications

A number of complex optimization problems arising in widely different contexts and applications have been effectively treated by the general framework of RSO. This include the real-life applications, computer science and operations research community combinatorial tasks, applications in the area of neural networks related to machine learning and continuous optimization tasks. In the following we summarize some applications in real-life enginering application areas which are the main interests of this research. In the area of electric power distribution there have been reported a series of real-world applications [35]. An open vehicle routing problem [36], as well as the pickup and delivery problem [37] both with the time and zoning constraints is modeled where the RSO methodology is applied to the distribution problem in a major metropolitan area. Alternatively to solve the vehicle routing problem with backhauls a heuristic approach based on a hybrid operation of reactive tabu search is proposed in [38]. By utilizing the RSO the flexible jobshop scheduling [39], the plant location problem [40], the continuous flow-shop scheduling problem [41], adaptive self-tuning neurocontrol [42] and the real-time dispatch of trams [43] were effectively solved. Moreover various applications of RSO focused on problems arising in telecommunication networks, internet and wireless in terms of optimal design, management and reliability improvements e.g. [44]. The multiple-choice multi-dimensional knapsack problem with applications to service level agreements and multimedia distribution is studied in [45]. In the military related applications, in optimal designing of an unmanned aerial vehicle routing system [46] and in finding the Underwater Vehicle Trajectories [47], RSO worked wonder. The problem of active structural acoustic control [48] and visual representation of data through clustering [49] are also well treated. Additionaly the solution of the engineering roof truss design problem is discussed in [50]. An application of RSO for designing barreled cylinders and domes of generalized elliptical profile is studied in [51]. Overall a series of successful projects acomplished with the aid of RSO could be found in [52-57]. Further applications of RSO are listed in [21,23] and the stochastic local search book [58].

8. Software Architectur Packages for the Proposed Reactive and Interactive MCDM

Grapheur and LIONsolver [8,9,22,23,59] are two implementaions of RSO. The software implements a strong interface between a generic optimization algorithm and DM. While optimizing the systems produce different solutions, the DM is pursuing conflicting goals, and tradeoff policies represented on the multi-dimensional graphs [21,23]. During multi-dimensional graphs visualization in these software packages, it is possible to call user-specific routines associated with visualized items. This is intended as the starting point for interactive optimization or problem solving attempts, where the user specifies a routine to be called to get information about a specific solution. These implementions of RSO are based on a three-tier model, independent from the optimization algorithm, effective and flexible software architecture for integrating problem-solving and optimization schemes into the integrated engineering design processes and optimal design, modeling, and decision-making.

For solving problems with a high level of complexity, modeling the true nature of the problem is of importance and essential. For this reason a considerable amount of efforts is made in modeling the MOO problems in Scilab (available in the appendix) which later will be integrated into optimizer package. Here, as an alternative to the previous approaches [17,18,28] the robust and interactive MOO algorithm of RSO is proposed in order to efficiently optimize all the design objectives at once in which couldn't be completely considered in the previous attempts. In this framework the quality of the design, similar to the previous research workflows, is measured using a set of certain functions, then an optimization algorithm is applied in order to optimize the function to improve the quality of the solution. Once the problem is modeled in Scilab it is integrated to the optimizer via advanced interfaces to the RSO algorithm and its brain-computer evolutionary multiobjective optimization implementations and visualization. In this framework the application of learning and intelligent optimization and reactive business intelligence approaches in improving the process of such complex optimization problems is accomplished. Furthermore the problem could be further treated by reducing the dimensionality and the dataset size, multidimensional scaling, clustering and visualization tools.

8.1. Creating the Model with Scilab

The Scilab file contains a string definition, *i.e. g_name*, inluding a short, mnemonic name for the model as well as two 8-bit integers, *i.e. g_dimension* and *g_range*, defining the number of input and output variables of the model. Additionally the file has two real-valued arrays; *i.e. g_min* and *g_max*, containing the minimum and maximum values allowed for each of the input and output variables. The following is a simple definition of a function that can be understood by utilized software package [59].

```
g_name = "ZDT1";
g _dimension = int8(2);
g _range = int8(2);
g _min = [0, 0, 0, 0];
g _max = [1, 1, 1, 1];
g _names = ["x1", "x2", "f1", "f2"];
function f = g_function(x)
f1_x1 = x(1)
g_x2 = 1 + 9 * x(2)
h = 1 - sqrt(f1_x1 / g_x2)
f = [ 1 - f1_x1, 1 - g_x2 * h ]
endfunction;
```

9. Case Study: Welded Beam Design

The problem of welded beam design [28] is a wellknown example of some complex designs issues arising in structural engineering, dealing with designing the form of steel beams and with connecting them to form complex stuctures. This study case has been used by many experts as a benchmark problem of single and also multiobjective design optimization. The problem of designing an optimal welded beam consists of dimensioning a welded steel beam and the welding length in order to minimize the cost subjected to bending stress, constraints on shear stress, the buckling load on the bar, the end the deflection of the beam, and side constraints. There are four design variables *i.e. h*, *l*, *t*, *b* shown in the **Figure 1**. Structural analysis of the welded beam leads to two nonlinear objective functions subjected to five nonlinear and two linear inequality constraints. The objectives are: the fabrication cost and the end deflection of the beam. In our case, the aim is to reduce fabrication cost without causing a higher deflection. Decision-making on the preferred solution among the Pareto-optimal set requires the intelligent participation of the designer, to identify the trade-off between cost and deflection.

As it is shown in the above figure the beam is welded on another beam carrying a certain load P. The problem is well studied as a single objective optimization problem [28], but we have transformed the original single objecttive problem into a two objective problem for more flexible design. In the original study the fabrication cost $\left(f_1(\mathrm{x})\right)$ of the joint is minimized with four nonlinear constraints related to normal stress, shear stress, buckling limitations and a geometry constraint. With the following formulation we have introduced one more objective *i.e.* minimization of the end deflection $\left(\delta(x)\right)$ of the structure. The problem has four decision variables presented in the optimization formulation, *i.e.* thickness of the beam *b*, width of the beam *t*, length of weld *l*, and weld thickness *h*. The overhang portion of the beam has a length of 14 in and F ¼ 6; 000 lb force is applied at the

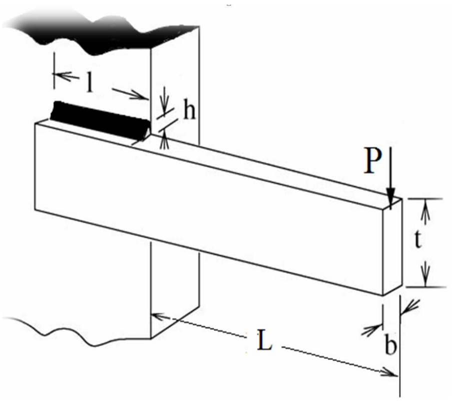
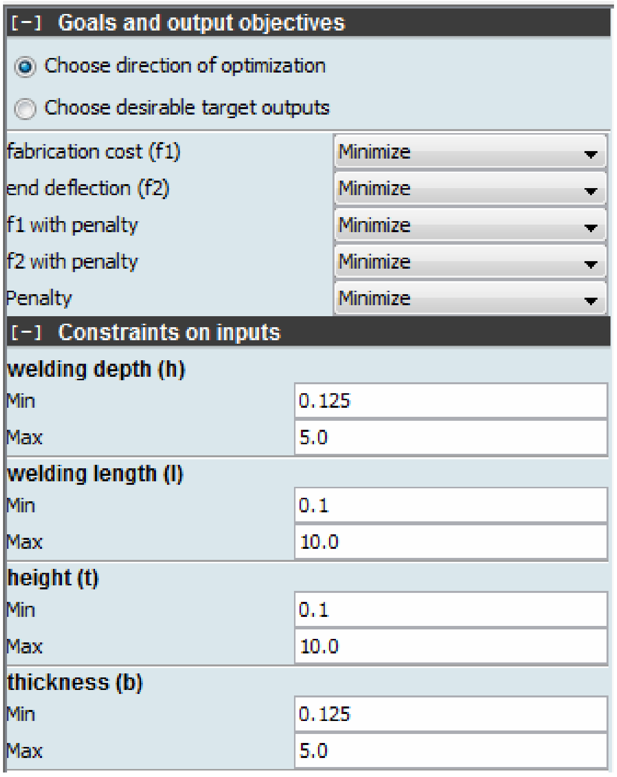


Figure 1. The welded beam optimal design problem.



[-] Goals and output objectives

● Choose direction of optimization

○ Choose desirable target outputs

fabrication cost (f1)	Minimize
end deflection (f2)	Minimize
f1 with penalty	Minimize
f2 with penalty	Minimize
Penalty	Minimize

[-] Constraints on inputs

welding depth (h)	
Min	0.125
Max	5.0
welding length (l)	
Min	0.1
Max	10.0
height (t)	
Min	0.1
Max	10.0
thickness (b)	
Min	0.125
Max	5.0

Figure 2. Description of the welded beam design problem in the software architecture of RSO multiobjective optimization; pointing out the objectives and constraints.

end of the beam. The mathematical formulation of the problem is given as;

$$\text{Minimize } f_1(x) = 1.104711h^2 l + 0.04811tb(14.0 + l),$$

$$\text{Minimize } f_2(x) = \delta(x) = \frac{2.1952}{t^3 b}$$

$$\text{Subjected to } g_1(x) \equiv 13,600 - \tau(x) \geq 0,$$

$$g_2(x) \equiv 30,000 - \sigma(x) \geq 0,$$

$$g_3(x) \equiv b - h \geq 0,$$

$$g_4(x) \equiv P_c(x) - 6,000 \geq 0$$

$$0.125 \leq h, b \leq 5.0,\ 0. \leq l, t \leq 10.0$$

Among the four constraints, g_1 deals with the shear stress developed at the support location of the beam which is meant to be smaller than the allowable shear strength of the material (13,600 psi). The g_2 guarantees that normal stress developed at the support location of the beam is smaller than the allowable yield strength of the material (30,000 psi). The g_3 makes certain that thickness of the beam is not smaller than the weld thickness from the standpoint. The g_4 keeps the allowable buckling load of the beam more than the applied load P for safe design. A violation of any of the above four constraints will make the design unacceptable. More on adjusting the constraints would be available in [22,23]. Additionaly on the stress and buckling terms calculated in [24] worth mentioning that they are highly non-linear to design variables.

In applying I-MODE framework to real-world design tasks of [22,23] in which real decision makers will be involved there exists a number of shortcomings which we needs further attention. I-MODE software implementtation can consider a maximum of three objectives due to limitation of visual representation of the Pareto-optimal solutions.

9.1. Setting up the RSO Software

Here the RSO software architecture of LIONsolver [21, 23,59] helps the designer to become aware of the different posibilities and focus on his preferred solutions, within the boundary of constraints. Consequently the constraints are transformed into a penalty function which sums the absolute values of the violations of the constraints plus a large constant. Unless the two functions are scaled, the effect of deflection in the weighted sum will tend to be negligible, and most Pareto-optimal points will be in the area corresponding to the lowest cost. Therefore each function is devided by the estimated maximum value of each function in the input range [28]. The Pareto-optimal solutions of the multiobjective optimization and MCDM corresponding to fabrication cost vs. end deflection of the beam are visually presented in the graph of **Figure 3**.

By associating a multidimentional graph for an advanced visualization, available in **Figure 4**, and a parallel chart, available in **Figure 5**, to the results table, the MCDM problem very clearly comes to the consideration and the final decision is very confidently made. Further it is observed that the welding length *l* and depth *h* are inversely proportional, the shorter the welding length, the larger the depth has to be, and that height *t* tends to be close to its maximum allowed value [18].

These observations can inspire a problem simplification, by fixing the height to its maximum value and by expressing the length as a function of depth, therefore eliminating two variables from consideration in the future explorations this optimal design problem.

Further, **Figure 2** describes an implementation of objectives and constraints of the welded beam design problem in the software architecture of RSO multiobjective optimization.

10. Conclusions

The proposed method which is developed on the basis of Reactive Search Optimization algorithms is compaired with the existing novel method of Interactive Multiobjective Optimization and Decision-making, using Evolutionary method in solving engineering optimal design problems. In order to evaluate the effectiveness of both

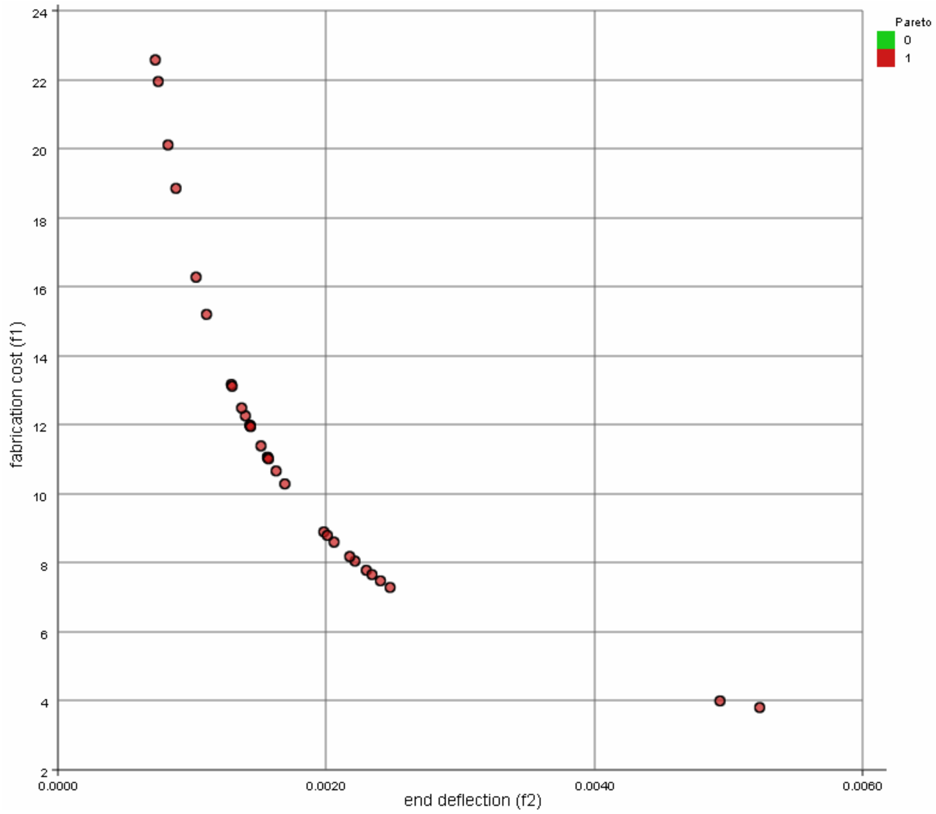


Figure 3. Set of Pareto-optimal solutions, fabrication cost vs. end deflection of the beam.

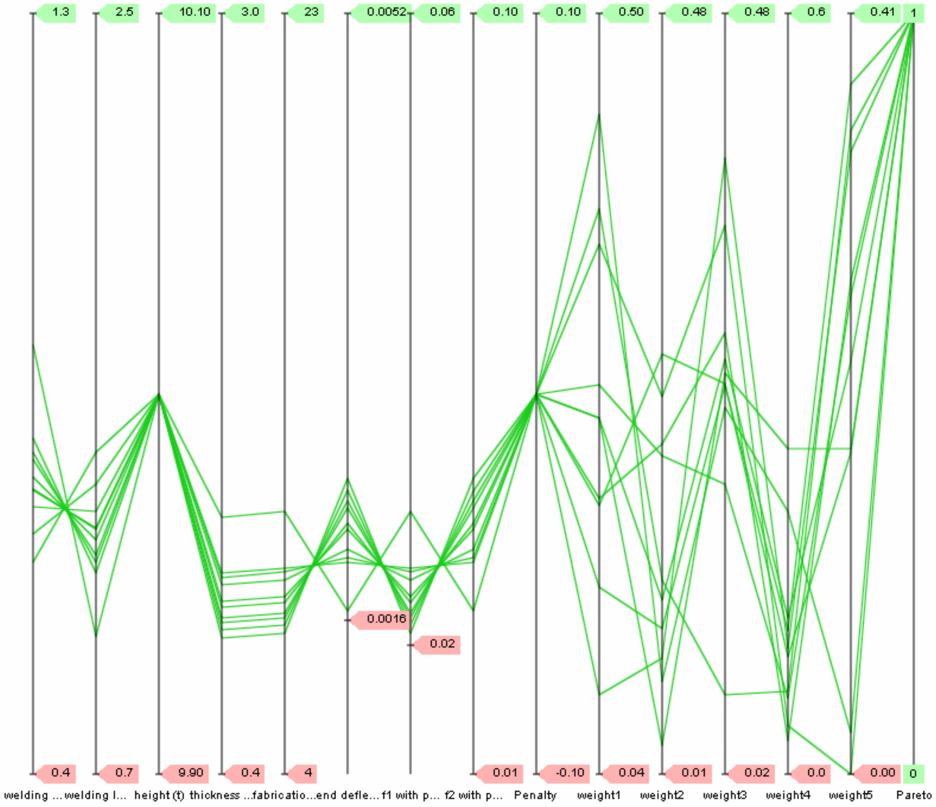


Figure 4. Parallel chart including all variables, constraints and optimization objectives.

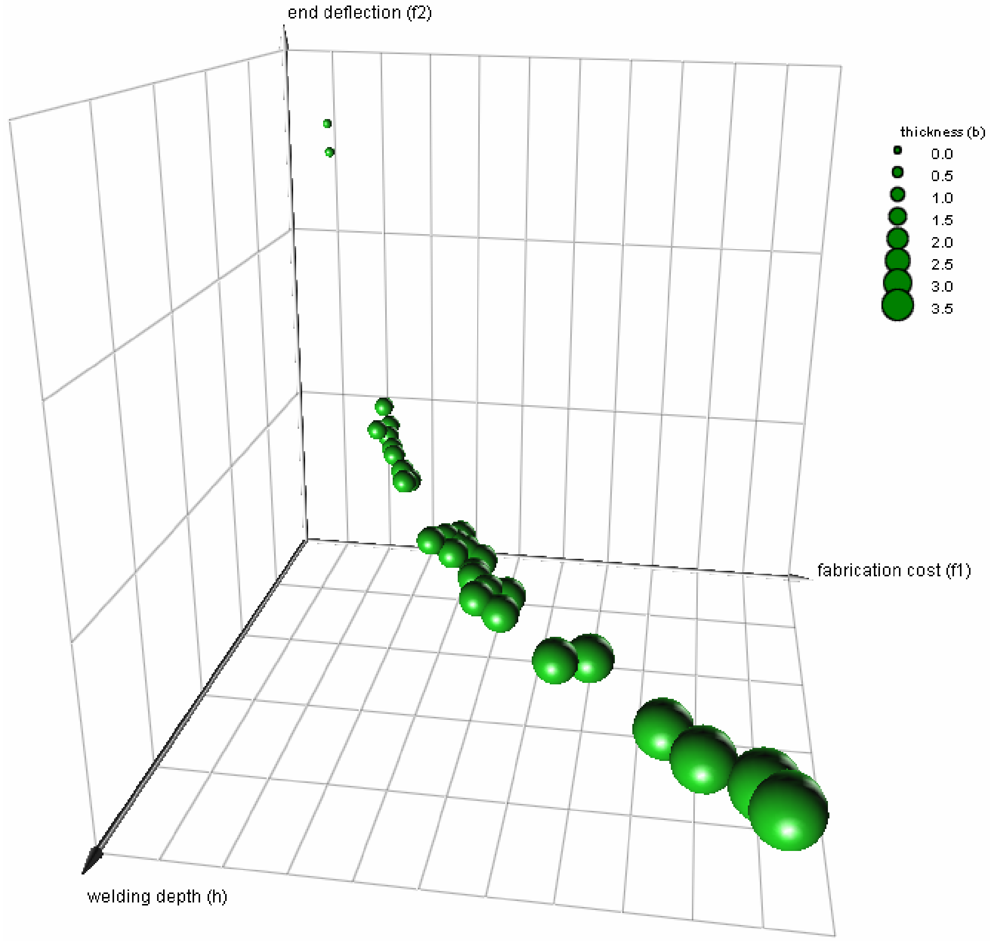


Figure 5. Multidimentional graph for an advanced visualization; the fabrication cost vs. end deflection of the beam.

methods the well-known study case of welded beam design problem is reconsidered. The preliminary tests of the software environment in the concrete context of optimal designing the welded beam design problem have shown the effectiveness of the approach in rapidly reaching a design preferred by the decision maker via advanced visualization tools and brain-computer novel interactions.

Further the utilization of the proposed software architecture for multiobjective optimization and decision-making, with a particular emphasis on supporting flexible visualization is discussed. The applicability of the software can be easily customized for different problems and usage contexts. For instance the geometrical optimization problems e.g. curves and surfaces [53,54], and skinning problem [60], would be in particular the future research interests of the authors.

10. Acknowledgements

Authors would like to thank Professor Miklos Hoffmann for his technical supports, valuable time and critical advices. Additionally the financial support of Jönköping University is highly acknowledged.

REFERENCES

[1] A. Mosavi, “The Multiobjective Optimization Package of IOSO; Applications and Future Trends,” In: CSCS, *Conference of PhD Students in Computer Science*, University of Szeged, Szeged, 2010, p. 55

[2] A. Mosavi, “Application of Multiobjective Optimization Packages in Design of an Evaporator Coil,” *World Academy of Science*, *Engineering and Technology*, Vol. 61, 2010, pp. 25-29.

[3] M. Esmaeili and A. Mosavi, “Variable Reduction for Multi-Objective Optimization Using Data Mining Techniques; Application to Aerospace Structures,” *Proceeding of ICCET*, *the* 2nd *IEEE International Conference on Computer Engineering and Technology*, Vol. 5, Chengdu, 16-18 April 2010, pp. V5-333-V5-337.

[4] A. Mosavi, “Computer Design and Simulation of Built Environment; Application to Forest,” *Proceeding of ICECS*’09, *The Second IEEE International Conference on Environmental and Computer Science*, Dubai, 28-30 December 2009, pp. 81-85.

[5] A. Mosavi, “Hydrodynamic Design and Optimization: Application to Design a General Case for Extra Equipments on the Submarine’s Hull,” *Proceeding on IEEE International Conference on Computer Technology and*

Development, *ICCTD*'09, Vol. 2, Kota Kinabalu, 13-15 November 2009, pp. 139-143.

[6] A. Mosavi, "Parametric Modeling of Trees and Using Integrated CAD/CFD and Optimization Tools: Application to Creating the Optimal Planting Patterns for New Forests," *Proceedings of* 2nd *International Conference Wind Effects on Trees*, Albert-Ludwigs-University of Freiburg, Freiburg, 2009.

[7] H. Afaq and S. Saini, "Swarm Intelligence Based Soft Computing Techniques for the Solutions to Multiobjective Optimization Problems," *International Journal of Computer Science Issues*, Vol. 8, No. 3, 2011.

[8] R. Battiti and P. Campigotto, "Reactive Search Optimization: Learning While Optimizing. An Experiment in Interactive Multiobjective Optimization," In: S. Voss and M. Caserta, Eds., *Proceedings of MIC* 2009, *VIII Metaheuristic International Conference*, *Lecture Notes in Computer Science*, Springer Verlag, Berlin, 2010.

[9] R. Battiti, M. Brunato, "Reactive Business Intelligence. From Data to Models to Reactive Search," Reactive Search Srl, Trento, 2011.

[10] S. Chaudhuri and K. Deb, "An Interactive Evolutionary Multi-Objective Optimization and Decision Making Procedure," *Applied Soft Computing*, Vol. 10, No. 2, 2010, pp. 496-511. doi:10.1016/j.asoc.2009.08.019

[11] J. Branke, "Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization," Springer, Berlin, 2008, pp. 157-178.

[12] A. Mosavi, "On Engineering Optimization the Splined Profiles," *Proceedings of International Conference on Engineering Optimization and International Mode Frontier Users*' *Meeting*, Trieste, 2010.

[13] K. Miettinen and M. M. Makela, "Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: Nimbus," *Optimization*, Vol. 34, No. 3, 1995, pp. 231-246. doi:10.1080/02331939508844109

[14] K. Miettinen, "Nonlinear Multiobjective Optimization," Kluwer, Boston, 1999.

[15] A. Mosavi, "The Large Scale System of Multiple Criteria Decision Making; Pre-Processing," *Large Scale Complex Systems Theory and Applications*, Vol. 9, 2010, pp. 354-359.

[16] A. Mosavi, "Applications of Interactive Methods of MOO in Chemical Engineering Problems," *Global Journal of Researches in Engineering*, Vol. 10, No. 3, 2010, p. 8.

[17] K. Deb and H. Gupta, "Searching for Robust Pareto-Optimal Solutions in Multi-Objective Optimization," *Proceedings of the Third Evolutionary Multi-Criteria Optimization* (*EMO*-05) *Conference* (*Lecture Notes on Computer Science*), Vol. 3410, 2005, pp. 150-164.

[18] K. Deb and S. Chaudhuri, "I-MODE: An Interactive Multi-Objective Optimization and Decision-Making Using Evolutionary Methods," *Evolutionary Multi-Criterion Optimization*, Vol. 4403, 2007, pp. 788-802.

[19] K. C. Tan, T. H. Lee, D. Khoo and E. F. Khor, "A Multiobjective Evolutionay Algorithm Toolbox for Computer-Aided Multiobjective Optimization," *IEEE Transactions on Systems*, *Man and Cybernetics—Part B*: *Cybernetics*, Vol. 31, No. 4, 2001, pp. 537-556. doi:10.1109/3477.938259

[20] A. Mosavi, "Multiobjective Optimization of Spline Curves Using Mode Frontier," *Proceedings of International Conference on Engineering Optimization and International Mode Frontierusers*' *Meeting*, Trieste, 2010.

[21] R. Battiti, M. Brunato, F. Mascia, "Reactive Search and Intelligent Optimization," *Operations research/Computer Science Interfaces*, Springer Verlag, Berlin, 2008.

[22] R. Battiti and M. Brunato, "Reactive Search Optimization: Learning While Optimizing," In: *Handbook of Metaheuristics*, 2nd Edition, Springer, Berlin, 2009.

[23] R. Battiti, M. Brunato and F. Mascia, "Reactive Search and Intelligent Optimization," *Operations Research/Computer Science Interfaces*, Vol. 45, Springer Verlag, Berlin, 2008.

[24] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation," *Proceedings of IEEE*, Vol. 89, No. 9, 2001, pp. 1275-1296. doi:10.1109/5.949485

[25] V. Chankong and Y. Y. Haimes, "Multiobjective Decision Making Theory and Methodology," North-Holland, New York, 1983.

[26] A. P. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization," In: G. Fandel and T. Gal, Eds., *Multiple Criteria Decision Making Theory and Applications*, Springer-Verlag, Berlin, 1980, pp. 468-486. doi:10.1007/978-3-642-48782-8_32

[27] R. Kamalian, H. Takagi and A. Agogino, "Optimized Design of Mems by Evolutionary Multi-Objective Optimization with Interactive Evolutionary Computation," *Genetic and Evolutionary Computation* (*GECCO*), Vol. 3103, 2004, pp. 1030-1041.

[28] G. V. Rekliatis, A. Ravindrab and K. M. Ragsdell, "Engineering Optimisation Methods and Applications," Wiley, New York, 1983.

[29] C. V. Jones, "Feature Article—Visualization and Optimization," *INFORMS Journal on Computing*, Vol. 6, No. 3, 1994, pp. 221-229. doi:10.1287/ijoc.6.3.221

[30] P. Piero, R. Subbu and J. Lizzi, "MCDM: A Framework for Research and Applications," *IEEE Computational Intelligence Magazine*, Vol. 4, No. 3, 2009, pp. 48-61.

[31] A. Mosavi, "Multiple Criteria Decision-Making Preprocessing Using Data Mining Tools," *International Journal of Computer Science Issues*, Vol. 7, No. 2, 2010, pp. 26-34.

[32] A. Adejuwon and A. Mosavi, "Domain Driven Data Mining; Application to Business," *International Journal of Computer Science Issues*, Vol. 7, No 2, 2010, pp. 41-44.

[33] A. Mosavi, "Application of Data Mining In Multiobjective Optimization Problems," *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 4, 2010.

[34] G. Anzellotti, R. Battiti, I. Lazzizzera, P. Lee, A. Sartori, G. Soncini, G. Tecchiolli and A. Zorat, "Totem: A Highly Parallel Chip for Triggering Applications with Inductive Learning Based on the Reactive Tabu Search," AI-HENP95, Pisa, IT, 1995.

[35] T. Genji, T. Oomori, K. Miyazato, N. Hayashi, Y. Fukuyama and K. Co, "Service Restoration in Distribution Systems Aiming Higher Utilization Rate of Feeders," *Proceedings of the 5th Metaheuristics International Conference* (*MIC*2003), 2003.

[36] R. Russell, W. Chiang, D. Zepeda, "Integrating Multi-Product Production and Distribution in Newspaper Logistics," *Computers and Operations Research*, Vol. 35, No. 5, 2008, pp. 1576-1588. doi:10.1016/j.cor.2006.09.002

[37] W. Nanry and J. Wesley Barnes, "Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search," *Transportation Research Part B*, Vol. 34, No. 2, 2000, pp. 107-121. doi:10.1016/S0191-2615(99)00016-8

[38] A. Login and S. Areas, "Reactive Tabu Adaptive Memory Programming Search for the Vehicle Routing Problem with Backhauls," *Journal of the Operational Research Society*, Vol. 58, 2007, pp. 1630-1641. doi:10.1057/palgrave.jors.2602313

[39] J. Chambers and J. Barnes, "New Tabu Search Results for the Job Shop Scheduling Problem," The University of Texas, Austin, Technical Report Series ORP96-06, Graduate Program in Operations Research and Industrial Engineering, 1996.

[40] H. Delmaire, J. Dıaz, E. Fernandez and M. Ortega, "Reactive GRASP and Tabu Search Based Heuristics for the Single Source Capacitated Plant Location Problem," INFOR 37, 1999, pp. 194-225.

[41] A. Fink and S. Voß, "Solving the Continuous Flow-Shop Scheduling Problem by Metaheuristics," *European Journal of Operational Research*, Vol. 151, No. 2, 2003, pp. 400-414. doi:10.1016/S0377-2217(02)00834-2

[42] P. Potocnik and I. Grabec, "Adaptive Self-Tuning Neurocontrol," *Mathematics and Computers in Simulation*, Vol. 51, No. 3-4, 2000, pp. 201-207. doi:10.1016/S0378-4754(99)00117-2

[43] T. Winter and U. Zimmermann, "Real-Time Dispatch of Trams in Storage Yards," *Annals of Operations Research*, Vol. 96, 2000, pp. 287-315.

[44] M. Magdon-Ismail, M. Goldberg, W. Wallace and D. Siebecker, "Locating Hidden Groups in Communication Networks Using Hidden Markov Models," *Lecture Notes in Computer Science*, Vol. 2665, 2003, pp. 126-137. doi:10.1007/3-540-44853-5_10

[45] M. Hifi, M. Michrafy and A. Sbihi, "A Reactive Local Search-Based Algorithm for the Multiple-Choice Multi-Dimensional Knapsack Problem," *Computational Optimization and Applications*, Vol. 33, No. 2, 2006, pp. 271-285. doi:10.1007/s10589-005-3057-0

[46] B. Hu and G. R. Raidl, "Variable Neighborhood Descent with Self-Adaptive Neighborhood Ordering," In: C. Cotta, A. J. Fernandez and J. E. Gallardo, Eds., *Proceedings of the 7th EU/MEeting on Adaptive*, *Self-Adaptive*, *and Multi-Level Metaheuristics*, Malaga, 2006.

[47] J. Ryan, T. Bailey, J. Moore and W. Carlton, "Reactive tabu Search in Unmanned Aerial Reconnaissance Simulations," *Proceedings of the 30th Conference on Winter Simulation*, 1998, pp. 873-880.

[48] R. Kincaid and K. Laba, "Reactive Tabu Search and Sensor Selection inActive Structural Acoustic Control Problems," *Journal of Heuristics*, Vol. 4, No. 3, 1998, pp. 199-220. doi:10.1023/A:1009681732632

[49] P. Hansen and N. Mladenovic, "Variable Neighborhood Search," In: E. Burke and G. Kendall, Eds., *Search Methodologies*: *Introductory Tutorials in Optimization and Decision Support Techniques*, Springer, Berlin, 2005, pp. 211-238.

[50] K. Hamza, H. Mahmoud and K. Saitou, "Design Optimization of N-Shaped Roof Trusses Using Reactive Taboo Search," *Applied Soft Computing Journal*, Vol. 3, No. 3, 2003, pp. 221-235. doi:10.1016/S1568-4946(03)00036-X

[51] J. Błachut, "Tabu Search Optimization of Externally Pressurized Barrels and Domes," *Engineering Optimization*, Vol. 39, No. 8, 2007, pp. 899-918. doi:10.1080/03052150701512604

[52] A. Mosavi, A. S. Milani, M. Hoffmann and M. Komeili, "Multiple Criteria Decision Making Integrated with Mechanical Modeling of Draping for Material Selection of Textile Composites," *Paper and poster in Proceeding of 15th Eeuropean Conference on Composite Materials*, Venice, 24-28 June 2012.

[53] A. Mosavi, M. Hoffmann and A. S. Milani, "Adapting the Reactive Search Optimization and Visualization Algorithms for Multiobjective Optimization Problems; Application to Geometry," *Conference of PhD Students in Computer Science*, Szeged, June 2012

[54] A. Mosavi, M. Hoffmann and A. S. Milani, "Optimal Design of the NURBS Curves and Surfaces Utilizing Multiobjective Optimization and Decision Making Algorithms of RSO," *Conference of PhD Students in Mathematics*, Szeged, Jnue 2012.

[55] E. Földi, A. Delavar, A. Mosavi, K. N. Hewage, A. S. Milani, A. A. Moussavi and M. Yeheyis, "Reconsidering the Multiple Criteria Decision Making Problems of Construction Projects; Using Advanced Visualization and Data Mining Tools," *Conference of PhD Students in Computer Science*, Szeged, 28-30 June 2012.

[56] A. Mosavi, M. Azodinia, Abbas S. Milani, Kasun N. Hewage and M. Yeheyis, "Reconsidering the Multiple Criteria Decision Making Problems of Construction Workers with the Aid of Grapheur," *Newsletter of Engin Soft CAE Conference* 2011, No. 4, 2011.

[57] A. Mosavi, M.Azodinia Abbas S. Milani, Kasun N. Hewage and M.Yeheyis, "Reconsidering the Multiple Criteria Decision Making Problems of Construction Workers with the Aid of Grapheur," *International ANSYS and EnginSoft Conference*, 2011.

[58] R. Lenne, C. Solnon, T. Stutzle, E. Tannier and M. Birattari, "Reactive Stochastic Local Search Algorithms for the Genomic Median Problem," *Lecture Notes in Computer Science*, Vol. 4972, 2008, pp. 266-276. doi:10.1007/978-3-540-78604-7_23

[59] http://www.lionsolver.com

[60] R. Kunkli and M. Hoffmann, "Skinning of Circles and Spheres," *Computer Aided Geometric Design*, Vol. 27, 2010, pp. 611-621.

Appendix

Welded beam design problem implementation in Scilab [59];

```
g_name = “weldedBeam”;
g_dimension = int8(4);
g_range = int8(5);
g_min = [0.125, 0.1, 0.1, 0.125, 0.0, 0.0, 0.0, 0.0, 0.0];
g_max = [5.0, 10.0, 10.0, 5.0, 350.0, 0.05, 1.0, 1.0, 10000.0];
g_names = [“welding depth (h)”, “welding length (l)”, “height (t)”, “thickness (b)”, “fabrication cost (f1)”, “end deflection (f2)”, “f1 with penalty”, “f2 with penalty”, “Penalty”];
P = 6000.0; L = 14.0; E = 3.0e7;
deltaMax = 0.25;
G = 12.0e6;
tauMax = 13600.0;
sigmaMax = 30000.0;
function f=g_function(x)
    h = x(1), l = x(2), t = x(3), b = x(4)
    //objectives
    f1 = 1.10471*h*h*l + 0.04811*t*b*(14.0+l)
    f2 = 4*P*(L^3) / (E*b*t^3)
    //constraints
    Penalty = 0
    tau1 = P/(sqrt(2)*h*l)
    M = P * (L + 0.5*l)
    R = sqrt(.25 * (l*l + (h+t)^2))
    J = 2.0*(h*l/sqrt(2))*(l*l/12.0 + .25*(h+t)^2)
    tau2 = M * R / J
    tauX = sqrt(tau1*tau1 + ((tau1*tau2*l)/R) + tau2*tau2)
    if tauX > tauMax then
        Penalty = Penalty + (tauX-tauMax)/tauMax
    end
    sigmaX = 6.0*P*L/(b*t*t)
    if sigmaX > sigmaMax then
        Penalty = Penalty + (sigmaX-sigmaMax)/tauMax
    end
    if h > b then
        Penalty = Penalty + (h - b) / b
    end
    PcX = (4.013*sqrt(E*G*t*t*b^6/36.0)/(L*L)) * (1-t/(2*L)*sqrt(E/(4.0*G)))
    if PcX < P then
        Penalty = Penalty + (P - PcX) / P
    end
    if Penalty > 0 then
        f1p = g_max(5) + Penalty
        f2p = g_max(6) + Penalty
    else
        f1p = f1
        f2p = f2
    end
    f = [f1, f2, f1p/g_max(5), f2p/g_max(6), Penalty]
endfunction;
```