Applied Soft Computing 15 (2014) 42–56

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering

Kalyanmoy Deb[a,*], Sunith Bandaru[b], David Greiner[c], António Gaspar-Cunha[d], Cem Celal Tutum[a]

[a] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, 428 South Shaw Lane, 2120 EB, MI 48824, USA
[b] Virtual Systems Research Centre, University of Skövde, SE 54128 Skövde, Sweden
[c] Institute of Intelligent Systems and Numerical Applications in Engineering (SIANI), Universidad de Las Palmas de Gran Canaria, 35017, Spain
[d] Institute for Polymer and Composites – I3N, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

ARTICLE INFO

Article history:
Received 15 February 2012
Received in revised form
19 September 2013
Accepted 8 October 2013
Available online 31 October 2013

Keywords:
Multi-objective optimization
Innovization
Design principles
Extrusion process
Noise barrier optimization
Friction stir welding

ABSTRACT

Computational optimization methods are most often used to find a single or multiple optimal or near-optimal solutions to the underlying optimization problem describing the problem at hand. In this paper, we elevate the use of optimization to a higher level in arriving at useful problem knowledge associated with the optimal or near-optimal solutions to a problem. In the proposed *innovization* process, first a set of trade-off optimal or near-optimal solutions are found using an evolutionary algorithm. Thereafter, the trade-off solutions are analyzed to decipher useful relationships among problem entities automatically so as to provide a better understanding of the problem to a designer or a practitioner. We provide an integrated algorithm for the innovization process and demonstrate the usefulness of the procedure to three real-world engineering design problems. New and innovative design principles obtained in each case should clearly motivate engineers and practitioners for its further application to more complex problems and its further development as a more efficient data analysis procedure.

Published by Elsevier B.V.

1. Introduction

Quest for new knowledge about problems of interest to an engineer or scientist has always been of utmost importance. However, due to constraints on time and other resources, practitioners are most interested in arriving at a single solution that will suffice the requirements for the instant. In a routine problem solving scenario such as in a design or a process operation activity, practitioners often need to solve an identical problem repeatedly but for different parameter settings. In such activities, instead of repeatedly executing similar tasks (which can be somewhat monotonous to an intelligent mind), a more wise approach would be to gather useful knowledge and problem properties that constitute a high-performing solution. Such knowledge will go a long way in providing insights about the problem and making the person an expert in solving the problem under consideration. In this paper, we suggest and discuss a computational approach for arriving at such useful knowledge thorough the use of an optimization process.

The elicitation of knowledge can be different in different problems. Here, we are particularly interested in knowledge that may help a designer or a practitioner in understanding their problems better. Often such knowledge can be thumb-rules or other rules such as decision-trees or semantic nets involving a few decision variables and problem functionalities. The important constituent of our approach is that the knowledge being extracted must be true for, not any arbitrary solution set, but for high-performing solutions of the problem. High-performing solutions are the solutions that are optimal or near-optimal corresponding to one or more objectives of the problem. This is where the need for an optimization algorithm arises.

When an optimization problem is formed for a single objective function, usually there is a single optimal solution that most optimization applications attempt to find. What we propose here is a multi-objective optimization study in which at least two conflicting objectives are considered. For example, cost of fabricating a product and its quality are two usual conflicting objectives of design. The advantage of considering multiple conflicting objectives is that the resulting optimization problem gives rise to a set of trade-off Pareto-optimal solutions [1–3]. Each of these solutions is optimal (and hence high-performing) with respect to certain trade-off

* Corresponding author.
E-mail addresses: kdeb@egr.msu.edu (K. Deb), sunith.bandaru@his.se (S. Bandaru), dgreiner@iusiani.ulpgc.es (D. Greiner), agc@dep.uminho.pt (A. Gaspar-Cunha), tutum@egr.msu.edu (C.C. Tutum).

1568-4946/$ – see front matter. Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.asoc.2013.10.011

among the objectives. Since all these solutions are optimal and high-performing, an analysis of them may reveal important properties that they share. Such properties then can be considered as knowledge that high-performing solutions possess. Often, such knowledge brings in new concepts and innovative ideas of solving the problem optimally. Due to these possibilities, the task of searching for multiple trade-off solutions and identifying properties commonly appearing to these solutions is called as an *innovization* process – creating innovation through optimization.

There are some related studies in the data-mining and machine learning literature. However, most of these studies only provide information that can be perceived visually. For example, self-organizing maps have been used to project the multi-dimensional objective and design spaces onto a two-dimensional map, followed by hierarchical clustering to reveal clusters of similar design solutions [4]. Taboda and Coit [5] used *k*-means clustering on the trade-off solutions to simplify the task of analyzing them. Dendograms are used to depict strongly related decision variables in [6]. MODE or multi-objective design exploration [7] uses a combination of kriging and self-organizing maps to visualize the structure of the decision variables using the non-dominated solutions. Heatmap visualization inspired from biological micro-array analysis was proposed [8]. For many-objective problems, Walker et al. [9] proposed 'Pareto shells' and analyzed various methods for ordering the solutions. Oyama et al. [10] used proper orthogonal decomposition to decompose the design vector into the mean and fluctuation vectors.

Other studies aim at representing knowledge in the form of 'if-then' type rules using association rule-mining [11] or rough-set theory [12]. Such information, though very helpful in specific cases, is not compact. A typical multi-dimensional dataset may give rise to many such rules since no clustering procedure is invoked to highlight only the most important set of rules. The use of decision trees for representing the knowledge contained in large datasets also suffers from similar drawbacks [12]. Methods based on functional analysis of variance (ANOVA) [13] are only useful for considering each feature of the dataset one at a time. Correlations between different features are difficult to identify using ANOVA. While neural networks are very effective in modeling non-linear correlations between multiple inputs, the black box nature of the obtained networks may not be attractive to a practitioner. In this study, our goal is to extract knowledge from the trade-off dataset of any given multi-objective optimization problem. More importantly, the extracted knowledge should be simple, compact and significant. The methodology should be capable of identifying inter-dependencies between different problem entities (variables and objective functions) of the dataset and representing them in the form of closed-form mathematical expressions, so that any practitioner can remember them as thumb-rules for creating a good design in future design scenarios having a similar underlying structure.

It is important here to differentiate our method from data-modeling techniques such as regression, multivariate adaptive regression splines (MARS) [14], response surfaces and kriging [15] which also condense data in the form of closed-form mathematical expressions. Since the main purpose of these methods is to model the given data as closely as possible, they are not designed to look for abrupt changes in correlations. Being quite flexible, they are capable of adapting the resultant mathematical function to fit the differently correlated part of the data. An instance of this effect in MARS has been studied in [16]. On the other hand, our methodology has been tailored to weed out those parts of the input dataset that are either outliers or show an abrupt change in relationship.

The remainder of this paper is organized as follows. We briefly describe the proposed innovization process in Section 2. In Section 3, we describe a clustering based optimization technique for knowledge extraction, termed as automated innovization. All computer algorithms required for the integrated innovization process are outlined in this section. Thereafter, we consider three different real-world engineering design problems and apply the proposed automated innovization process and reveal useful knowledge about each problem. In all cases, the extracted knowledge provided new concepts of design which were not known before.

2. *Innovization*: Innovation through optimization

Designers and practitioners are often interested in solving their current problem at hand in order to meet deadlines and pre-specified targets. However, by virtue of their scientific bend of mind, they are always interested in gathering useful knowledge about their problem. The type and extent of knowledge can be different in different problems, but practitioners interested in engineering design problems would most likely be interested in knowing what design principles must a solution have in order for it be an optimal or high-performing solution. Such questions are vitally important to a designer as the answers to such questions provide deep insights among parameter interactions that would elevate a design to become optimal.

In the past few years, the first author has proposed a two-step procedure for unveiling such important information about a problem. The first step involves finding a set of high-performing solutions and the second step involves analyzing the obtained solutions to reveal important design principles. We discuss each of these two steps in the following paragraphs.

1 *Finding a set of high-performing solutions:* A design task usually involves a number of design variables each of which needs to be determined in order for the design to be feasible to be used and to achieve a certain goal. The goal is often to minimize the cost of fabrication, weight of the product, operation time, amount of harmful gas etc. The feasibility of a design is often checked by investigating if the design satisfies a number of pre-defined constraints, such as maximum stress developed due to loading is smaller than or equal to the strength of the material used or natural frequency of vibration is set well above the applied forcing frequency. Clearly, achieving such a feasible and optimal solution is not possible by manual (or trial-and-error) setting of variables, rather a computer-aided optimization algorithm is called for. Because of vagaries of design variables, constraints and goal functions, it becomes important to design or customize a suitable optimization algorithm for a particular problem. However, if a single goal is considered in the optimization task, the outcome would be a single optimal solution (we refer here as a high-performing solution). In the context of discovering design principles, we would require not one, but multiple high-performing solutions. An important question then to ask is where from multiple high-performing solutions will come? One way to look at the problem is again to follow what designers usually do in practice.

A designer in practice usually solves a similar problem repeatedly but with different parameter values. Let us take a typical scenario of an engineer who works in a pressure vessel design company. Today, the engineer may need to design a pressure vessel for a goal of minimum volume and for a particular internal pressure requirement for a refinery, tomorrow the same engineer may be designing another vessel for different internal pressure requirement for another petrochemical industry, and so on. By multiple solutions, we mean the optimal solution for each such scenario that the engineer is faced with every now and then in his/her work. One way to find multiple such solutions would be to treat the problem as a bi-objective optimization problem in which in addition to volume being a goal, we can include