Application of Neural Network to Interactive Physical Programming

Hongzhong Huang[1] and Zhigang Tian[2]

[1] School of Mechatronics Engn.,
University of Electronic Science and Technology of China,
Chengdu, Sichuan 610054, China
hzhhuang@dlut.edu.cn

[2] Department of Mechanical Engineering, University of Alberta,
Edmonton, Alberta, T6G 2G8, Canada,
ztian@ualberta.ca

Abstract. A neural network based interactive physical programming approach is proposed in this paper. The approximate model of Pareto surface at a given Pareto design is developed based on neural networks, and a map from Pareto designs to their corresponding evaluation values is built. Genetic algorithms is used to find the Pareto design that best satisfies the designer's local preferences. An example is given to illustrate the proposed method.

1 Introduction

Physical programming developed by Messac [1], has been successfully applied to high-speed-civil-transport plane design [1], control, structure design [2], interactive design [3], [4] and robust design [5]. Interactive physical programming is based on physical programming. It takes into account the designer's preferences during the optimization process, and allows for design exploration at a given Pareto design.

Based on the Tappeta, Renaud, and Messac's work [4], this paper mainly obtains the following achievements: (1) The approximation to the Pareto surface around a given Pareto design is developed using neural network for design exploration. (2) A map from Pareto designs to their corresponding evaluation values, called the designer's local preferences model, is built using neural networks. (3) Genetic algorithms is used in a optimization process with the designer's local preferences model as objective function to search for the Pareto design that best satisfies the designer's local preferences. The obtained Pareto design is further used as the aspiration point in a compromise programming problem [4] to obtain the final optimal design.

2 Interactive Physical Programming Based on Neural Networks

Interactive physical programming takes into account the designer's preferences during the optimization process, which enables the designer to partly control the optimization process. The flow chart of interactive physical programming is shown in Figure 1, with detailed explanations given as follows.

J. Wang, X. Liao, and Z. Yi (Eds.): ISNN 2005, LNCS 3496, pp. 725–730, 2005.

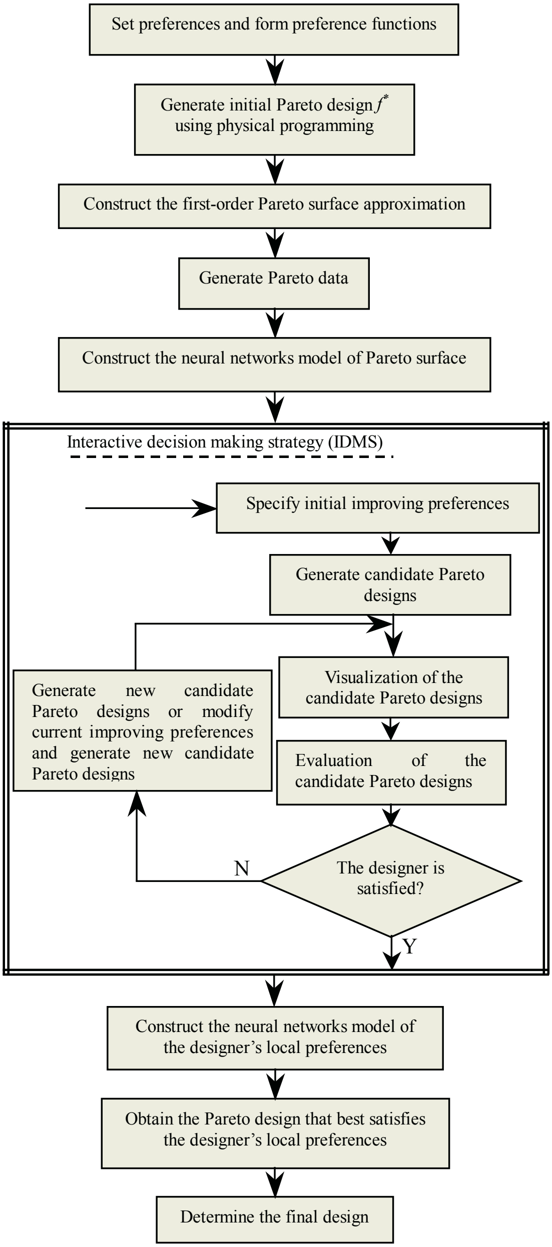

Fig. 1. Flow chart of interactive physical programming.

2.1 First-Order Pareto Surface Approximation around a Given Pareto Design

The initial Pareto design that best satisfies the designer's initial preferences, f^*, is generated by solving a physical programming problem. The next step is trying to achieve the approximation to the Pareto surface around f^* as accurately as possible, so that the designer can explore other Pareto designs in the following interactive decision making process. We make sensitivity analysis at Pareto design f^*, and generate the first-order Pareto surface approximation [4]. The first-order Pareto surface approximation is represented by a linear equation describing the relationship among all the objective functions.

2.2 Pareto Designs Generation and Neural Network Model of Pareto Surface

To represent the Pareto surface more accurately, more information about other Pareto designs around f^* is required. Usually, to a multiobjective optimization problem with m objectives, at least $m(m-1)/2$ Pareto designs are required. A projection method including two steps, the predictor step and the corrector step, is used to generate these Pareto designs [4].

Pareto surface is highly nonlinear, nonsmooth, and has discontinuity. Tappeta et al. used second-order polynomial to represent the Pareto surface [4], however, it can't describe the attributes of the Pareto surface mentioned above. Neural networks [6], [7] is good at representing complex nonlinear model, and it can describe the Pareto surface more accurately. There are interior relationships among the objective functions through design variables, and these relationships should be embodied as much as possible between inputs and outputs. Therefore, the neural network model of the Pareto surface is built with $(f_1, f_2, \ldots, f_{m-1})$ as input and $\sqrt[m+1]{f_1 \cdot f_2 \cdot \ldots \cdot f_{m-1} \cdot f_m^2}$ as output.

2.3 Interactive Decision Making Strategy

The flow chart of the interactive decision making strategy is shown in Figure 1. The Pareto design f^* is Pareto optimal, and there's no other feasible design that can improves all the objective functions. But the designer may want to improve some objectives at the expense of some other objectives, this is called the designer's improving preferences. The improving preferences can be specified by qualitative sentences, e.g., improve f_i, f_j, and sacrifice f_k.

After specifying the improving preferences, a set of candidate Pareto designs that satisfy the improving preferences are generated around the current Pareto design f^* using the neural network model of Pareto surface. They are presented to the designer using the Pareto visualization tool, which will help the designer to evaluate these candidate Pareto designs.

These candidate Pareto designs are evaluated using qualitative-quantitative analysis [8]. The qualitative part is evaluating the candidate Pareto designs with Analytic Hierarchy Process (AHP). The quantitative part is evaluating the candidate Pareto designs with quantitative criteria based on the preference functions of all the objectives. We combine four proposed quantitative criteria with the AHP approach to evaluate the candidate Pareto designs, and determine an evaluation value with respect to each candidate Pareto design to represent the designer's preference on the Pareto design.

After examining the candidate Pareto designs presented, the designer can generated a different set of approximate Pareto designs with current improving preferences, or modify current improving preferences and generate new approximate Pareto designs, or select one of the candidate designs presented if he is satisfied with it and then turn to the next step.

Finally, with the objective functions of the Pareto designs as inputs, the corresponding evaluation values as outputs to train a neural network, the neural network model of the designer's local preferences can be built. The neural network model of the map is called the neural network model of the designer's local preferences. The method developed in reference [4] to determine the Pareto design that best satisfies the designer's local preferences is just selecting the Pareto design that best satisfies the designer's local preferences from the candidates Pareto designs already generated. There're infinite Pareto designs around f^*, and it's obvious that the designer can't inspect all of them. Thus, the neural network based model gives us a continuous and more accurate model of the designer's local preference.

2.4 Determine the Final Design

With $(f_1, f_2, ..., f_{m-1})$ as design variables, the evaluation value corresponding to the Pareto design as objective function, genetic algorithms [9] is used to search for the Pareto design f_{local} that minimizes the evaluation value. f_{local} is thus the Pareto design that best satisfies the designer's local preferences. f_m is calculated via the design variables $(f_1, f_2, ..., f_{m-1})$ using the neural network model of the Pareto surface, and then the corresponding evaluation value can be calculated using the neural network model of the designer's local preferences.

The obtained Pareto design that best satisfies the designer's local preferences, f_{local}, is on the neural network model of the Pareto surface, and not on the real Pareto surface (although there's minor difference between them). With f_{local} as the aspiration point, a compromise programming problem is solved [4], and the final design x_{final} can be obtained. The objective functions vector corresponding to x_{final}, f_{final}, is on the real Pareto surface.

3 Example

A symmetrical pinned-pinned sandwich beam that supports a motor is considered [2]. A vibratory disturbance (at 10Hz) is imparted form the motor onto the beam. The mass of the motor is ignored in the following analysis. The objectives of this problem are fundamental frequency, cost, width, length, height and mass. The design variables are $x = \{d_1, d_2, d_3, b, L\}$, where L denotes the length of the beam, b is the width, and d_1, d_2 and d_3 represent the heights of the three pieces of the beam [2].

The region limits of the design objectives' preference functions are shown in Table 1. The steps are shown as follows.

(1) A physical programming problem is solved to obtained the initial Pareto design f^* and the corresponding design variables

$$f^* = (162.94, 346.27, 0.8747, 3.9967, 1867.6, 0.3599)\,. \tag{1}$$

$$x^* = (0.3233, 0.3333, 0.3599, 0.8747, 3.9967)\,. \tag{2}$$

(2) Through the sensitivity analysis of the Pareto surface at f^*, the first-order Pareto surface approximation around f^* is obtained. 30 Pareto designs around f^* are obtained. The neural network model of the Pareto surface is built.

(3) Through the interactive decision making process, the candidate Pareto designs are generated, visualized and evaluated. The neural network model of the designer's local preferences is built.

(4) Genetic algorithms is used in the optimization process to obtain the Pareto design that best satisfies the designer's local preferences. Then, a corresponding compromise programming problem is solved, and the final design can be obtained

$$f_{\text{final}} = (160.3097, 325.7527, 0.7799, 3.9479, 1834.2, 0.3477)\,. \tag{3}$$

$$x_{\text{final}} = (0.2954, 0.3230, 0.3477, 0.7799, 3.9479)\,. \tag{4}$$

Table 1. Physical programming region limits table.

Design objectives	Class type	g_{i5}	g_{i4}	g_{i3}	g_{i2}	g_{i1}
Fundamental frequency f/Hz	2-S	100	110	120	150	200
Cost c/\$·m^{-3}	1-S	2000	1950	1900	1800	1000
Width b/m	2-S	0.30	0.35	0.40	0.45	0.55
Length L/m	2-S	3.0	3.3	3.8	4.0	6.0
Mass m/kg	1-S	2800	2700	2600	2500	2000
Height h/m	1-S	0.60	0.55	0.50	0.40	0.30

4 Conclusions

The interactive nature of the proposed interactive physical programming approach enables the designer to partly control the optimization process, which can improve the design efficiency and design result, and avoid wasting lots of time in the wrong directions during the design process. Neural networks, a powerful nonlinear modeling tool, is used to construct the Pareto surface model and the designer's local preferences model, which makes them more accurate and reasonable. The continuous model of the designer's local preferences can be obtained in this way, and thus the continuous optimization can be implemented. From the view of continuous optimization, the design that best satisfies the designer's preferences can be obtained.

Acknowledgements

This research was partially supported by the National Excellent Doctoral Dissertation Special Foundation of China under the contract number 200232.

References

1. Messac, A., Hattis, P.D.: High Speed Civil Transport (HSCT) Plane Design Using Physical Programming. AIAA/ASME/ASCE/AHS Structures, Structural Dynamics & Materials Conference-collection of Technical Papers, **3** (1995) 10-13
2. Messac, A.: Physical Programming: Effective Optimization for Computational Design. AIAA Journal, **34** (1996) 149-158
3. Messac, A., Chen, X.: Visualizing the Optimization Process in Real-Time Using Physical Programming. Engineering Optimization, **32** (2000) 721-747
4. Tappeta, R.V., Renaud, J.E., Messac, A.: Interactive Physical Programming: Tradeoff Analysis and Decision Making in Multiobjective Optimization. AIAA Journal, **38** (2000) 917-926
5. Chen, W., Sahai, A., Messac, A.: Exploration of the Effectiveness of Physical Programming in Robust Design. Journal of Mechanical Design, Transactions of the ASME, **122** (2000) 155-163
6. Yan, P.F., Zhang, C.S.: Artificial Neural Network and Simulated Evolution Computation. Tsinghua University Press, Beijing (2000)
7. Huang, H.Z., Huang, W.P., Wang, J.N.: Neural Network and Application to Mechanical Engineering. Mechanical Science and Technology, **14** (1995) 97-103
8. Wang, Y.L.: System Engineering: Theory, Method and Application. Higher Education Press, Beijing (1998)
9. Huang, H.Z., Zhao, Z.J.: Genetic Algorithm Principle, Realization and Their Application Research, Prospect in Mechanical Engineering. Journal of Machine Design. **17** (2000) 1-6