Genetic programming through bi-objective genetic algorithms with a study of a simulated moving bed process involving multiple objectives

Brijesh Kumar Giri[a], Jussi Hakanen[b], Kaisa Miettinen[b], Nirupam Chakraborti[a],*

[a] Department of Metallurgical & Materials Engineering, Indian Institute of Technology, Kharagpur 721 302, India
[b] Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

ABSTRACT

A new bi-objective genetic programming (BioGP) technique has been developed for meta-modeling and applied in a chromatographic separation process using a simulated moving bed (SMB) process. The BioGP technique initially minimizes training error through a single objective optimization procedure and then a trade-off between complexity and accuracy is worked out through a genetic algorithm based bi-objective optimization strategy. A benefit of the BioGP approach is that an expert user or a decision maker (DM) can flexibly select the mathematical operations involved to construct a meta-model of desired complexity or accuracy. It is also designed to combat *bloat* – a perennial problem in genetic programming along with over fitting and under fitting problems. In this study the meta-models constructed for SMB reactors were compared with those obtained from an evolutionary neural network (EvoNN) developed earlier and also with a polynomial regression model. Both BioGP and EvoNN were compared for subsequent constrained bi-objective optimization studies for the SMB reactor involving four objectives. The results were also compared with the previous work in the literature. The BioGP technique produced acceptable results and is now ready for data-driven modeling and optimization studies at large.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Real-world optimization problems involving computationally expensive simulation models are getting more and more common. This sets requirements for optimization tools used in order to get reliable solutions without spending a lot of time in the solution process. To reduce the time spent in solving computationally costly optimization problems, many meta-modeling techniques have been developed (see e.g. [1,2] and references therein). New challenges are introduced when we are dealing with problems having multiple conflicting objective functions to be optimized simultaneously. Solving these kinds of multi-objective optimization problems involves dealing with so-called Pareto optimal solutions, where no objective function can be improved without impairing some other objective(s). There are various ways to handle computationally costly problems. Among others, meta-modeling techniques have been developed for multi-objective optimization, see e.g. [3].

Earlier, in a study related to an industrial blast furnace, an evolutionary neural network (EvoNN) was proposed for meta-modeling [4]. Over the years, it has been progressively modified and applied

successfully to numerous engineering optimization problems of significant complexity [5–7]. In this paper, we provide a complementary approach, where the basic strategy of EvoNN has been extended to genetic programming (GP) [8,9]. The idea is to come up with a bi-objective evolutionary version of GP, a strategy that is increasingly becoming a competitor and a potential alternate for neural networks. This bi-objective approach would provide a GP user the flexibility of constructing models of desirable complexities, as the users of EvoNN are currently capable of. Such a feature enables one to construct models with the right *fit*, and offers remedies in situations frequently encountered in GP learning where its constituent trees tend to grow out of bounds without any noticeable differences in accuracy. Keeping this in view, here we elaborate the basic strategic development of the concept of this bi-objective evolutionary genetic programming (BioGP) and apply it in the optimization task of a fructose and glucose separation process based upon simulated moving bed technique [10]. The same task is also assigned to the EvoNN method for comparison.

Simulated moving bed (SMB) reactors [10,11] are widely used in the chemical and pharmaceutical industries for chromatographic separation of various constituents in a liquid stream flowing through a packed bed. Since moving the bed itself would be cumbersome and impractical, the SMB process ingeniously creates the same relative motion between the solid and the fluid by sequentially switching the inlet and outlet points for the liquid that moves in and out of the bed. A typical arrangement analyzed in the

---
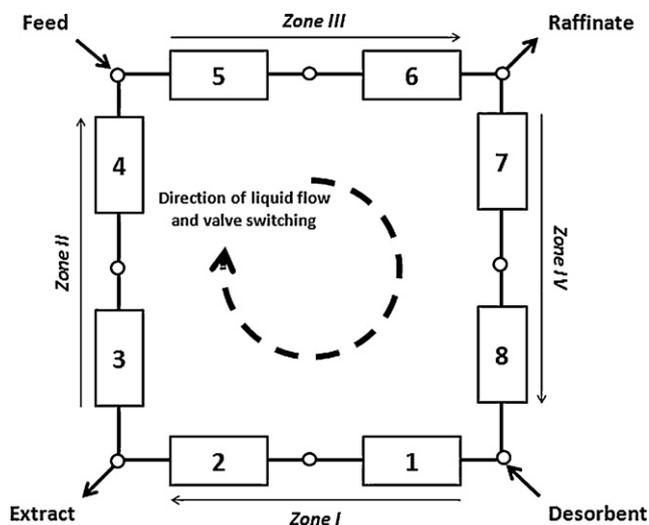
Fig. 1. Schematics of simulated moving bed process.

Fig. 2. Data partitioning in evolutionary neural network showing the data subsets used for training.

earlier studies [11] is shown schematically in Fig. 1, consisting of a number of separating columns connected in a loop with two inlet streams (feed and desorbent) and also two outlet streams (raffinate and extract).

Fairly comprehensive analytical models are available for the SMBreactors [10,11]. However, coupling the direct numerical solutions of the complicated model equations with a multi-objective optimization scheme, particularly if using population based evolutionary computation, would lead to a very heavy, and often unmanageable, computational burden. This has been done, e.g., in [12], where the authors apply an evolutionary multi-objective optimization algorithm (NSGA) to the direct synthesis of methyl tertiary butyl ether from tertiary butyl alcohol using e.g. SMB techniques. They consider problems having only two or three objectives due to computational aspects, although they acknowledge that the number of objectives could be higher. They do not report anything on how much time it took to solve those problems. The remedy for the computational burden would be to come up with some efficient meta-models that could be effectively used during the optimization task instead of the original time-consuming models.

The rest of this paper is organized as follows: A brief overview of evolutionary neural network is provided in Section 2 followed by the details of the proposed BioGP technique in Section 3. After that, a multi-objective optimization problem related to the SMB process is described. Furthermore, results obtained by applying BioGP are presented and compared to those obtained with EvoNN as well as previous research. The paper is concluded in Section 5.

2. Essentials of evolutionary neural network

The evolutionary neural network [4] is an attempt to bring in the process of neural network construction in the realm of evolutionary multi-objective optimization. Here we deal with a population of neural networks and attempt to construct a trade-off between the two conflicting criteria: complexity of the network and its training error. The trade-off curve between these two criteria consists of a set of distinct neural networks, from which a suitable one is picked up by applying an information criterion, usually the *Corrected Akaike Information Criterion* (AICc) ([13,14]) computed as:
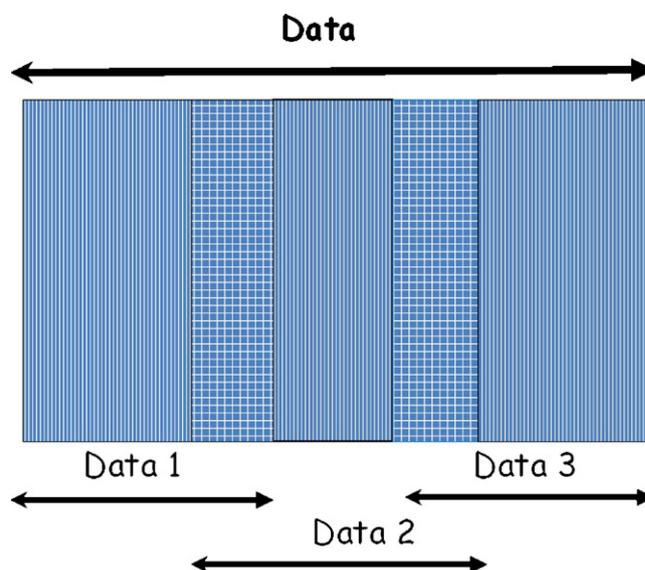
$$AICc = AIC + \frac{2k(k+1)}{n-k-1}, \tag{1}$$

where AIC is the Akaike information criterion defined as [14]

$$AIC = 2k + n \ln\left(\frac{RSS}{n}\right), \tag{2}$$

where k indicates the number of parameters used in the model. Here this would be determined by the total number of connections in both its upper and lower parts of the network, including the biases. The total number of observations is denoted by n, and RSS is the residual sum of squares of the errors for the model being evaluated.

The configuration of this network is flexible in its lower part as neither the number of weights used nor the number of hidden nodes is fixed. EvoNN need not use all the input variables, and, furthermore, it can very well do without some of the possible connections to the hidden layer. It differs further from the traditional neural networks by not requiring a separate training algorithm, since the lower part of the network evolves through a multi-objective genetic algorithm. A predator-prey algorithms being employed for that [4,15]. Also, here we combine the training and testing procedures by dividing the total data set into a number of overlapping subsets, as elaborated in Fig. 2 for a data set with three partitions. Networks trained for each of the subsets are tested on the rest to ensure that the model works acceptably in every segment of data. The learning process is initiated with a random population of neural networks, with each member having its unique set of topology and weight vector while they use the same data set. The member networks are subjected to a fast ranking following the standard Fonseca–Flemming procedure [16] and new crossover and mutation procedures were required here as the variable space represents a neural network architecture. During the crossover, two parent neural networks give rise to a pair of children, by exchanging a randomly selected hidden node between the two parents along with the pertinent connections, as shown schematically in Fig. 3.

A linearly self-adaptive mutation is applied on the weights and a mutated version of any weight belonging to a population member m, connecting an input i to a hidden node j is obtained as

$$W_{ij}^{m\mu} = W_{ij}^m + \lambda_m(W_{ij}^k - W_{ij}^l), \tag{3}$$

where the superscript μ denotes the mutated version of the weight W_{ij}. The weights with superscripts k and l are randomly picked from