An Efficient Topology/Parameter Control in Evolutionary Design for Multi-domain Engineering Systems

Kisung Seo

Dept. of Electronics Engineering
Seokyeong University

Abstract

This paper suggests a control method for an efficient topology/parameter evolution in a bond graph-based GP design framework that automatically synthesizes designs for multi-domain, lumped parameter dynamic systems. We adopt a hierarchical breeding control mechanism with fitness-level-dependent differences to obtain better balancing of topology/parameter search - biased toward topological changes at low fitness levels, and toward parameter changes at high fitness levels. As a testbed for this approach in bond graph synthesis, an eigenvalue assignment problem, which is to find bond graph models exhibiting minimal distance errors from target sets of eigenvalues, was tested and showed improved performance for various sets of eigenvalues.

Key Words : Topology/parameter Control, Evolutionary Design, Genetic Programming, Multi-domain, Hierarchical Breeding Control

1. Introduction

Engineering design of dynamic systems usually involves discovering topological connections of components and/or optimizing of their numeric parameters in open-ended manner[1-4]. Evolutionary computation has been used many times to automate the creation of engineering designs[3,4]. Especially, genetic programming has been used for design of several patented electrical circuits, controllers, and antennas [5,6].

The topology specifies the system's structure, which consists of the number and type of components and their interconnections. On the other hand, parameter optimization seeks the best numerical values for given a topology. Some design problems are concentrated on topology only under given parameter values such as topology optimization in truss design[1,2], where the connectivity of members in a truss is to be determined. On the other hand, there exists parameter only optimization with given topologies, such as sizing optimization of trusses [3], where cross-sectional areas of members are considered as design variables.

Most engineering design problems involve making topological connections of components and optimizing their numeric values in an open-ended manner. That means both topology and parameter values should be optimized[4-6]. However, no definitive approach has yet been introduced for topology/parameter evolution. Most approaches to finding better designs are limited to using huge populations.

The key idea of our approach is to provide different breeding probabilities for topology and parameter operations according to the fitness level of each subpopulation in a genetic programming with fitness-stratified populations. Additionally, more topology-altering operations are executed in earlier generations and more parameter-altering operations are executed in later generations.

The Bond Graph / Genetic Programming (BG/GP) design methodology[7] has been developed to overcome limitations of single-domain design approaches and enable open-ended search, based on the combination of these two powerful tools - bond graph[8] and genetic programming[5]. It has been tested for a few applications - an analog filter[10], printer drive mechanism[11]. BG/GP worked efficiently for these applications.

As a test class of design problems for topology/parameter control in bond graph synthesis, we have chosen one in which the objective is to realize a design having a specified set of eigenvalues. The eigenvalue assignment problem is well defined and has been studied effectively using linear components with constant parameters. Section 2 discusses the nature of the topology/parameter design problem and bond graph synthesis. Section 3 describes genetic construction for bond graph model. Section 4 explains hierarchical probability control method. Section 5 presents results for eigenvalue design problems, and Section 6 concludes the paper.

2. Topology/Parameter Design in Bond Graph Synthesis

2.1 The nature of topology/parameter design in engineering problem

Topology connection is represented as a directed graph G with vertex set V and edge set E, where $V=\{v_0, v_1, \dots, v_{n-1}\}$ and $E(G)=\{e_{ij}=\{v_i, v_j\} \mid v_i \in V, v_j \in V\}$. Given

Manuscript received May. 24, 2005; revised Jun. 10, 2005.

parameter values are represented as a function $G(f)=\{(v_{i,}\ f(v_i))|v_i\in V,\ f(v_i)\in R\}$.

Some design problems are to determine topologies only under given parameter values or parameters optimization with given topologies(Fig. 1).

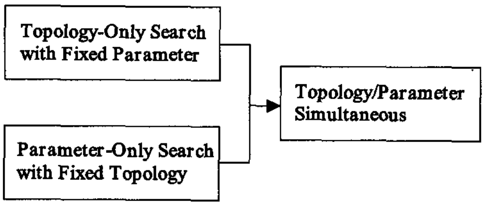
Topology-Only Search with Fixed Parameter
Parameter-Only Search with Fixed Topology
Topology/Parameter Simultaneous

Fig. 1. Topology/Parameter design problem

However, most engineering design problems involve discovering good topological connections of components and optimizing their numeric values, simultaneously in an open-ended manner. In search for good designs for topologically open dynamic systems, no topology can be evaluated in the absence of an associated set of parameters; and conversely, no set of parameter values can be evaluated except within the context of a given topology.

An obvious approach is to allocate to each new topology whatever amount of parameter search effort is needed to find at least a locally optimal parameterization, before judging the quality of the topology. A second strategy might be to allocate a fixed amount of search effort to each topology, adequate to optimize the parameters in many cases. However, such strategies may consume far more search effort than is practical during a simultaneous topology/parameter search, and when to stop each parameter search is difficult to determine.

2.2 Bond Graph Synthesis

Bond graph modeling[8,9] is a powerful method that enables a unified approach to the analysis, synthesis and evaluation of dynamic system. It represents the common energy processes of multi-domain systems - electrical, mechanical, fluid, and thermal systems - in one graphical notation, as shown in Fig. 2.

The Bond Graph / Genetic Programming (BG/GP) design methodology has been developed to overcome limitations of single-domain design approaches and enable open-ended search, based on the combination of these two powerful tools and tested for a few applications[7,10,11].

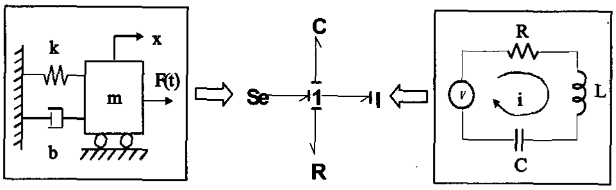

Fig. 2. The same bond graph model for two different domains

The scheme of topology/parameter search in bond graphs is as follows. Fig. 3 shows component types typically used in bond graphs, including junctions, sources of effort or flow, one-port elements like resistors, capacitors and inductors, 2-port elements like transformers and gyrators, and higher-level modules composed of lower-level primitive elements. "Below the line" are associated parameters.

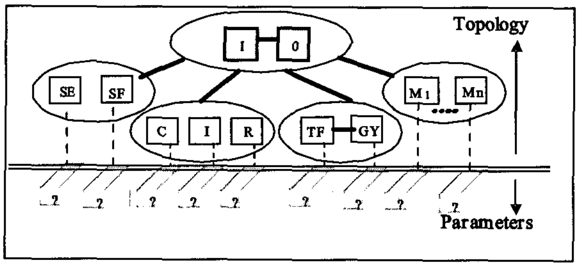

Fig. 3. The scheme of topology/parameter space in bond graphs

3. Genetic constructor for bond graph synthesis

The BG/GP system used GP functions and terminals for bond graph construction as follows. There are four types of functions: *add* functions that can be applied only to a junction and which add a C, I, or R element; *insert* functions that can be applied to a bond and which insert a 0-junction or 1-junction into the bond; *replace* functions that can be applied to a node and which can change the type of element and corresponding parameter values for C, I, or R elements; and *arithmetic* functions that perform arithmetic operations and can be used to determine the numerical values associated with components (Table 1). Details of function definition and GP process are illustrated in [7].

Table 1. GP terminals and functions

Name	#Args	Description
add_C	4	Add a C element to a junction
add_I	4	Add an I element to a junction
add_R	4	Add an R element to a junction
insert_J0	3	Insert a 0-junction in a bond
insert_J1	3	Insert a 1-junction in a bond
replace_C	2	Replace the current element with a C element
replace_I	2	Replace the current element with an I element
replace_R	2	Replace the current element with an R element
+	2	Add two ERCs
-	2	Subtract two ERCs
enda	0	End terminal for add element
endi	0	End terminal for insert junction
endr	0	End terminal for replace element
erc	0	Ephemeral random constant (ERC)

In this approach, GP functions in Table 1 are classified into three categories - topology operation, intermediate topology operation, and parameter operation. The topology level corresponds to insert junctions(0, 1) and add elements(C, I, and R) operations, which determine topological connections among junctions and elements. such as insert_J0, insert_J1, add_C, add_I, and add_R. Some of them are illustrated in Fig. 4 and 5.

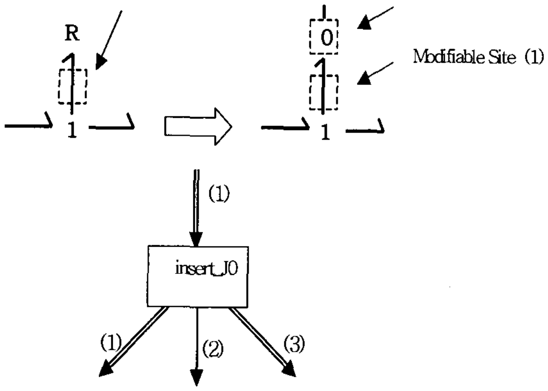

Fig 4. Insert_J0 function

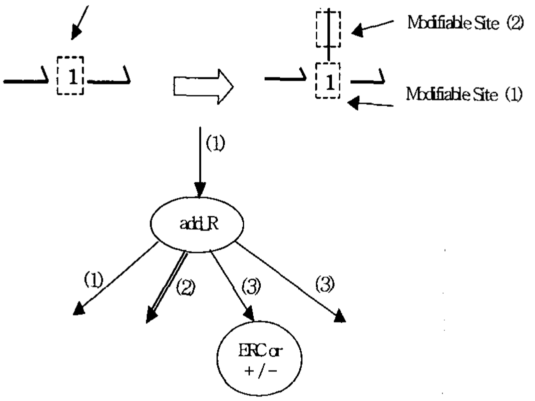

Fig 5. add_R function

The intermediate topology level has replacing a C, I and R elements with another elements, such as replace_C, replace_I, and replace_R. The replace_C operation replaces current element with C element, as shown in Fig. 6. The third parameter level includes numerical operations to fill in values of elements, such as C (capacitor), I (inductor), and R (resistor) as shown in Fig. 7(where, ERC means ephemeral random constant).

4. Hierarchical breeding control method

In this paper, a hierarchical breeding control mechanism is adopted to obtain better performance based on differential balancing of topology-altering operations and parameter-altering operations according to fitness level, in a fitness-structured multi-population model. The basic idea for this control mechanism arises from observing the human design process. Usually, preliminary or conceptual design involves more structural modification, and final or detailed design involves more parameter tuning - i.e., there is greater concentration on design topology in the early stage and more on parameter tuning in the later stage.

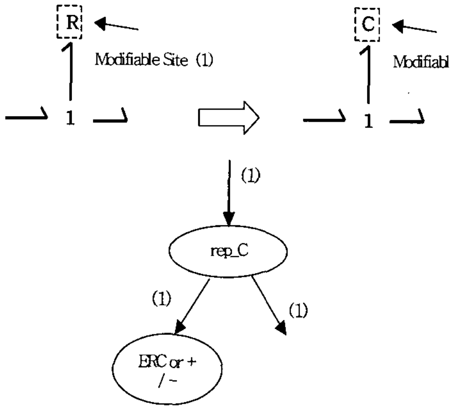

Fig. 6. replace_C function

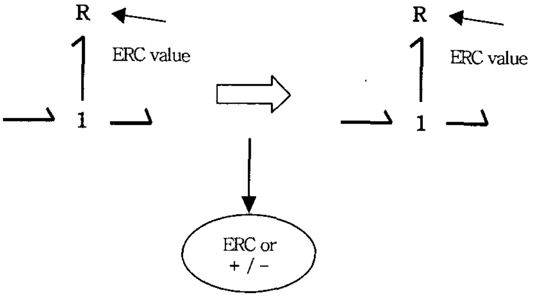

Fig. 7. ERC numerical operation

Therefore, the key concept is to provide different breeding probabilities for topology-altering and parameter-altering operations according to fitness level of the subpopulation (Fig. 8). Subpopulations are organized in a hierarchy with ascending fitness levels. In other words, topology-altering operations have higher probability than parameter-altering operations at low fitness levels, and vice-versa. Additionally, more topology-altering operations are executed in earlier generations and more parameter-altering operations are executed in later generations(Fig. 9).

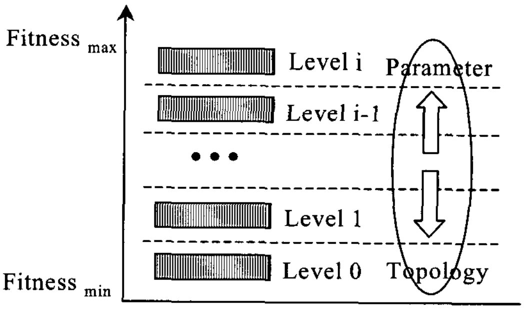

Fig. 8. Hierarchical breeding control structure.

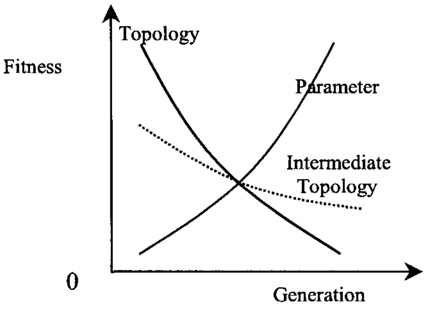

Fig. 9. Distribution of breeding control probability

5. Experiments and Analysis

To evaluate and compare the proposed approach with the previous one, the eigenvalue assignment problem is used, for which the design objective is to find bond graph models with minimal distance errors from a target set of eigenvalues(Fig. 10). It is a classical "inverse" design problem - seeking something with given behavior, rather than the "forward" analysis problem of calculating the behavior of a given artifact.

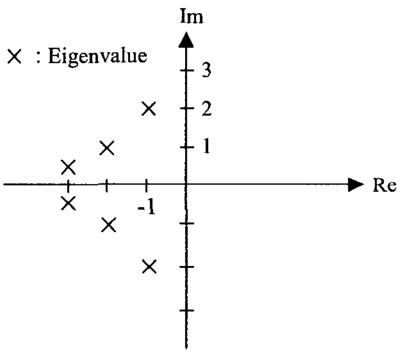

Fig. 10. An example of a target set of eigenvalues

Fig. 11 gives an example of the solution eigenvalues obtained for a typical run with targets -1± 2j, -2± j. The corresponding bond graph model obtained is shown on the right side of Fig. 11. Same result for an six-eigenvalues is shown in Fig. 12.

Eigenvalues
Target eigenvalues : -1 ± 2j, -2 ± j Solution eigenvalues:-1.028 ± 1.929j, -2.008 ± 0.995j Average distance error:0.043
Bond Graph Structure
added junctions : 0 added elements : 5 added bonds : 5
Bond Graph Parameters
C1 : 0.0029 C2 : 7.499 I 1 : 121.581 I 2 : 0.027 R3 : 0.033

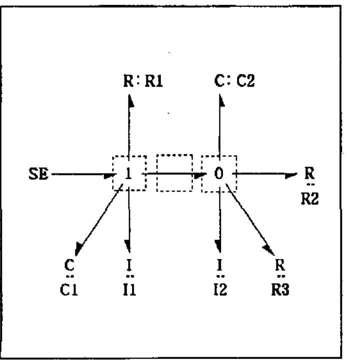

Fig. 11. An four-eigenvalue result

Eigenvalues
Target eigenvalues : -1 ± 2j, -2 ± j , -3±0.5j Solution eigenvalues:-1.014 ± 2.012j, -2.053 ± 1.016j -2.998 ± 0.503j Average distance error : 0.18
Bond Graph Structure
added junctions : 3 added elements : 10 added bonds : 13
Bond Graph Parameter
C1 : 0.106, C2 : 0.309 C3 : 0.000 I 1 : 100.574, I2 : 10.0 I3 : 570.908 R3 : 21.39, R4 : 40.393 R5 : 1.116, R6 : 0.002

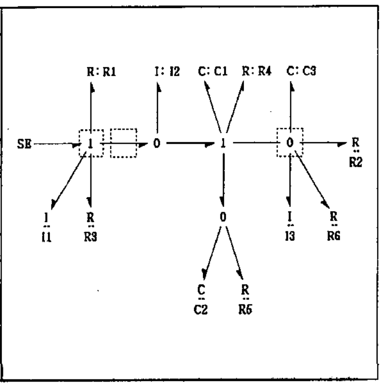

Fig. 12. An six-eigenvalue result

The following sets (consisting of various 6-, and 10-eigenvalue target sets, respectively) were used for topology/parameter control runs.

Eigenvalue sets used in experiments:

1) {-1± 2j, -2± j, -3± 0.5j}
2) {-10± j, -1± 10j, -3± 3j}
3) {-20± j, -1± 20j, -7± 7j}
4) {-1, -2, -3, -4, -5, -6}
5) {-20± j, -1± 20j, -7± 7j, -12± 4j, -4± 12j}
6) {-1, -2, -3, -4, -5, -6, -7, -8, -9, -10}

The fitness function is defined as follows: pair each target eigenvalue one: one with the closest one in the solution; calculate the sum of distance errors between each target eigenvalue and the solution's corresponding eigenvalue, divide by the order, and perform hyperbolic scaling. Relative distance error (normed by the distance of the target from the origin) is used. We used a strongly-typed version of lilgp [12] to generate bond graph models. These examples were run on a single Pentium IV 2.8GHz PC with 512MB RAM. The GP parameters were as shown below.

Number of generations : 500
Population sizes : 100 in each of ten subpopulations
Initial population: half_and_half
Initial depth : 3-6
Max depth : 12 (with 800 max_nodes)
Selection : Tournament (size=7)
Crossover : 0.9
Mutation : 0.1

The results of 6- and 10-eigenvalue runs are provided in Fig. 13 and 14, showing average distance error for each set across 10 experiments. Fig. 13 illustrates the comparison between the basic approach (without topology/parameter control) and the hierarchical topology/parameter breeding control on typical complex conjugate and real six-eigenvalue target sets. In all four sets, numbered 1)-4), the average error in the hierarchical topology/parameter breeding control approach is smaller than that of the basic approach. Fig. 14 represents the results on two 10-eigenvalue sets, set numbers 5) and 6) above, and shows that the new approach also outperforms the basic approach on these problems.

There are other factors to be determined to obtain optimal results, such as distribution of breeding control rate for topology- and parameter-altering operations, the control rate for the intermediate topology, and the ratio between fitness and generation etc. Current results are due to simple setting of control factors based on a few preliminary experiments. Therefore, much improvement is expected if optimal values of these control factors are found through further experiments and analysis.

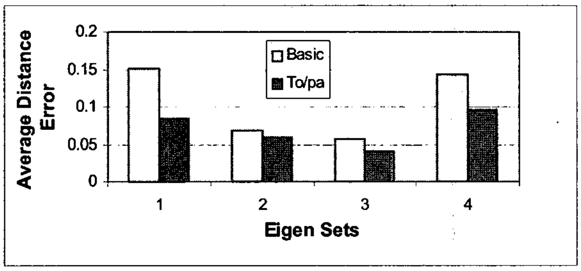

Fig. 13. Results for 6 eigenvalues

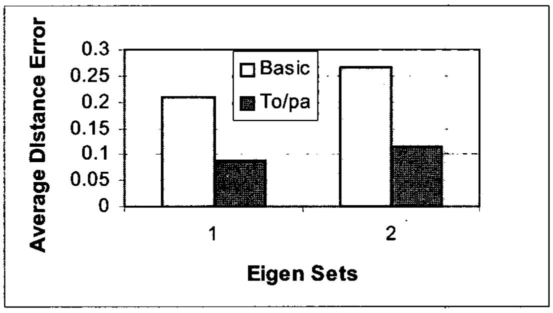

Fig. 14. Results for 10 eigenvalues(set 1 is eigenvalue set 5 above, etc.)

6. Conclusion

This paper has introduced a hierarchical breeding control method for efficient topology/parameter evolution in bond-graph-based GP design. We adopt a hierarchical breeding control mechanism, implemented in a set of subpopulations separated hierarchically according to fitness levels, to obtain better performance based on balancing of topology/parameter search using a given set of switched modular primitives. Topology-altering operations are given higher probability in high-fitness subpopulations, and parameter-altering operations get higher probability in lower-fitness subpopulations. Simultaneously, in all subpopulations, the percentage of topology-altering operations is reduced as the number of generations increases.

As a proof of concept for this approach, the eigenvalue assignment problem, which is to synthesize bond graph models with minimum distance errors from pre-specified target sets of eigenvalues, was used. Results showed better performance for all tested eigenvalue sets when the new topology/parameter control method was used. This tends to support the conjecture that a carefully tailored representation and sophisticated topology/parameter control method will improve the efficiency of GP search.

References

[1] Krish, U., "Optimal topologies of truss structures", *Computer Methods in Applied Mechanics and Engineering*, 72, pp.15-28, 1989

[2] Ringertz, U. T., "On topology optimization of trusses", *Engineering Optimization*, 9, pp.209-218, 1985

[3] Goldberg, D.E. and Samtani, M. P., "Engineering optimization via genetic algoritms", *Proceedings of the Ninth Conference on Electronic Computations, ASCE*, pp.471-482, 1986

[4] Deb, K, Gulati, S, "Design of Truss-Structures for Minimum Weight using Genetic Algorithms", *Journal of Finite Elements in Analysis and Design*, vol. 37, 5, pp.447-465, 2001

[5] Koza, J. R. et al., *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic Publishers, 2003

[6] Koza, J. R., Bennett F. H., Andre D., Keane M. A., *Genetic Programming III, Darwinian Invention and Problem Solving*, Morgan Kaufmann Publishers, 1999

[7] Seo K., Hu, J., Fan, Z., Goodman E. D., Rosenberg R. C., "Automated Design Approaches for Multi-Domain Dynamic Systems Using Bond Graphs and Genetic Programming", *International Journal of Computers, Systems and Signals*, 3(1). 2002 pp. 55-70

[8] Karnopp, D. C., Rosenberg R. C., Margolis, D. L., *System Dynamics, A Unified Approach, 3nd ed.*, John Wiley & Sons (2000)

[9] Sharpe J. E. E. E. , Bracewell R. H. , "The Use of Bond Graph Reasoning for the Design of Interdisciplinary Schemes", *1995 International Conference on Bond Graph Modeling and Simulation*, 1995, pp. 116-121

[10] Fan Z., Hu, J., Seo, K., Goodman E. D., Rosenberg R. C., Zhang, B., "Bond Graph Representation and GP for Automated Analog Filter Design", *Genetic and Evolutionary Computation Conf. Late-Breaking Papers*, San Francisco (2001) 81-86

[11]. Fan Z., Seo, K., Rosenberg R. C., Hu J., Goodman E. D., "Exploring Multiple Design Topologies using Genetic Programming and Bond Graphs", *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2002*, New York (2002) 1073-1080.

[12] Zongker, D., Punch, W., *lil-gp 1.1 User's Manual.* Michigan State University, (1996)

Kisung Seo

He received B.S. M.S. and Ph.D. degree in Electrical Engineering in 1982, 1986 and 1993, respectively, from Yonsei Univeristy. From 1993-1998, he was a full time lecturer and assistant professor in Industrial Engineering at Seokyeong University. From 1999-2002, he was a research associate in GARAGe(Genetic Algorithms Research and Applications Group) at Michigan State University, and during 2002-2003, visiting assistant professor in Electrical & Computer Engineering at Michigan State University.

He is currently an associate professor in Electronics Engineering at Seokyeong University. His research interests include evolutionary design, genetic programming, bond graph and multi-energy domain system, and evolvable hardware.