

Reconsidering Fitness and Convergence in Co-evolutionary Design

MARY LOU MAHER AND PETER XIANGHUA WU

Department of Architectural and Design Science
The University of Sydney, NSW 2006, Australia

Abstract Computational evolution provides a mechanism for searching a space of potential solutions according to a specified fitness function. In design, the search for potential solutions is often interleaved with changes in the requirements for the solution. We propose a model of design that uses a co-evolutionary computational process. Specifically, we present design as a parallel search for both design requirements and design solutions. Further, we develop a co-evolutionary process in which the interaction between requirements and solution redefine the current fitness function. The concepts of fitness and convergence in computational evolution do not necessarily have the same meanings in a co-evolutionary process in which the fitness function changes. An additional consideration not present in computational evolution is the interaction between the parallel search spaces. We present a mechanism for interaction which results in a kind of co-evolution called mutualism.

Keywords: Co-evolutionary computation, Design, Evolutionary computation, GAs, fitness, convergence

1. Introduction

Computational evolution provides a mechanism for searching a space of potential solutions according to a specified fitness function. However, these evolutionary models typically require that the metric by which a solution is measured is defined before the solution space is searched. A characteristic of designing is a reconsideration of the requirements when a design solution is suggested. To achieve the effect of a changing metric, several different fitness functions should be used in different runs of evolutionary design processes. An alternative to changing the metric manually as the evolutionary process searches for solutions is to consider a co-evolutionary process in which the problem space is searched in parallel to the solution space.

Co-evolution is the term used to identify the process in nature in which two or more species interact so intimately that their evolutionary fitness depends on each other. Biological co-evolution has been the inspiration for a class of computational algorithms called co-evolutionary computing (for example, see Hillis, 1991; Paredis J., 1995; and Richard N. L., 1995). Co-evolutionary design, as introduced in Maher (1994), is an approach to design problem solving in which the requirements and solutions of design evolve separately and affect each other. This paper firstly identifies co-evolutionary design by comparing it to design as search and design as exploration. Then an algorithm for co-evolutionary design is presented to highlight the process and parameters. A reconsideration of the purpose of the fitness function and its affect on convergence is necessary since the fitness function changes through the co-evolutionary cycles. The interactions between requirements and solutions of design may possibly add some new variables to both aspects of design, which may not only redefine the search space for requirements and solutions but also the fitness function. Based on the idea of mutualism, which is one of the three types of coevolution in nature (Smith M.J., 1989), the interacting populations raise the level of fitness in both, rather than the two populations competing with each other or one population living off the other.

2. Design as search, exploration and coevolution

Simon (1969) characterised design as a search process, allowing the design process to be understood as one of the “sciences of the artificial”. Since then the design research community has embraced the “design as search” model by formulating the goals, state spaces, and operators for various design domains and design problems. Although the search paradigm is very strong and still underpins much of problem solving, other models of design have been proposed that address the formalisation of design knowledge and design goals. Here we compare three related models: search, exploration, and co-evolution. The comparison highlights the difference between the three models in order to clarify co-evolutionary design.

2.1 DESIGN AS SEARCH

Design can be formalised as search when the goals of the design are well-defined before search commences and the focuses of design are not changed until a solution is found. Design as search can be modelled as:

$$\begin{cases} \mathbf{S} = \{S_1(\mathbf{P}), S_2(\mathbf{P}), S_3(\mathbf{P}), \dots, \dots\} \\ S_f = \text{best}\{\mathbf{S}\} \end{cases} \quad (1)$$

where

- \mathbf{P} is a given focus of design,
- \mathbf{P} is the space of problem requirements,
- \mathbf{S} is a set of design solution with $S_i \in \mathbf{S}$,
- $S_i(\mathbf{P})$ is the i th design solution with $S_i(\mathbf{P}) \in \mathbf{S}$,
- S_f is the final selected design solution.

In this model, \mathbf{P} is the focus of the design which is derived from the problem requirements \mathbf{P} . A solution \mathbf{S} is selected based on how well it satisfies \mathbf{P} . In computational evolution terms, \mathbf{P} is the fitness function. The fitness function is defined once and convergence occurs when the solution with the best fitness function is found.

2.2 DESIGN AS EXPLORATION

Design becomes exploration when the focus of the design changes as the process continues. A focus, derived from the requirement space, determines which part of a solution space is considered best. Design as exploration can be modelled as follows:

$$\begin{cases} S_i = \{S_{i1}(\mathbf{P}), S_{i2}(\mathbf{P}), S_{i3}(\mathbf{P}), \dots, \dots\} \\ S_{fi} = \text{best}\{S_i\} \\ S_f = \{S_{f1}, S_{f2}, S_{f3}, \dots, S_{fn}\} \end{cases} \quad (2)$$

where,

- \mathbf{P}_i is a given focus of design at time $i, i=1,2,\dots,n$,
- \mathbf{P} is a space of problem requirements,
- $S_{ij}(\mathbf{P})$ is the j th design solution corresponding to the focus \mathbf{P}_i ,
- S_{fi} is the best design solution of S_i corresponding to the space of \mathbf{P} with \mathbf{P}_i ,
- S_f is the final selected design solution.

Design as exploration uses the space of requirements as a problem definition and allows the search focus to change. Depending on different focuses, there are different sets of possible designs can be generated with different “best” designs. The idea of exploration can be illustrated using Figure 1, showing that a designer will see different parts of a solution space when he considers different focuses. Assuming that f is the fitness function, exploration allows the fitness function to change, but does not specific how it changes.

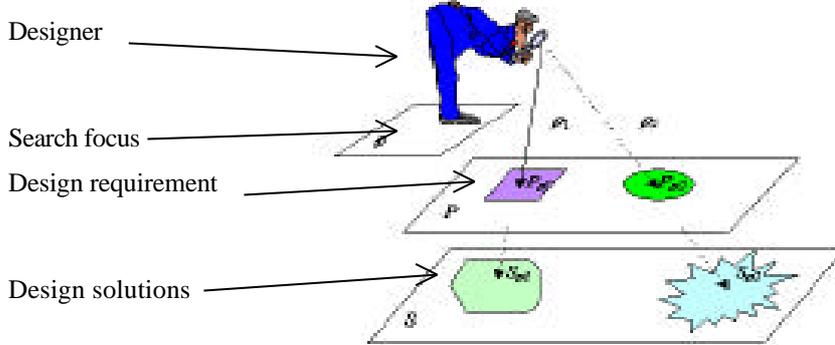


Figure 1. The idea of design as exploration

2.3 DESIGN AS COEVOLUTION

Design becomes co-evolution when the focus of the design can change, and the requirements space and solution space change through mutual interaction. The focus of the search is based on the requirements when searching the solution space, and based on the solutions when searching the requirement space. Design as co-evolution can be modelled as:

$$\left\{ \begin{array}{l} \begin{array}{l} f_i = f(P_i) \\ S_i = \{S_{i1}(P_i), S_{i2}(P_i), S_{i3}(P_i), \dots, \dots\} \\ S_{fi} = \text{best}\{S_i\} \\ S_f = \{S_{f1}, S_{f2}, S_{f3}, \dots, S_{fn}\} \end{array} \\ \\ \begin{array}{l} f'_i = f(S_i) \\ P_{fi} = \{P_{f1}(S_i), P_{f2}(S_i), P_{f3}(S_i), \dots, \dots\} \\ P_{f'fi} = \text{best}\{P_{fi}\} \\ P_f = \{P_{f'f1}, P_{f'f2}, P_{f'f3}, \dots, P_{f'fn}\} \end{array} \end{array} \right. \quad (3)$$

where,

- f_i is a given focus for the design solution at time i ,
- f'_i is a given focus for the design requirements at time i ,
- P_i is the i th space of problem requirements,
- S_i is the i th space of design solutions,
- $P_{f'ij}(S_i)$ is the j th design requirement corresponding to S_i and f'_i
- $S_{ij}(P_i)$ is the j th design solution corresponding to P_i and f_i ,
- S_i is a set of design solution corresponding to the space of P_i with f_i ,
- P_{fi} is a set of problem requirement corresponding to the space of S_i with f_i ,
- $P_{f'fi}$ is the best design requirement of P_{fi} ,
- S_{fi} is the best design solution of S_i ,
- S_f is the final selected design solution,
- P_f is the final selected design requirements.

The model of design as coevolution is illustrated in Figure 2. The interactions between two sets of spaces occur through the focus of the search. The focus for the solution space is based on the requirements in the problem space and the focus for the problem space is based on the solutions in the solution space. The first evaluation corresponds to the first downward arrow on diagram, from P to S . If no satisfactory solution is found with the stated design requirements, the solution space becomes the basis to derive the focus for searching the problem space. Hence, the second part of the first phase corresponds to certain changes in the fitness function when a design-space is given, i.e., the upward arrow from S to P . After searching for relevant requirements in the problem space, another fitness function is derived from the new requirement space to be the focus for the search for a solution (Maher and Poon, 1996). The revised fitness function has the possibility to change the search space by introducing new criteria to be measured.

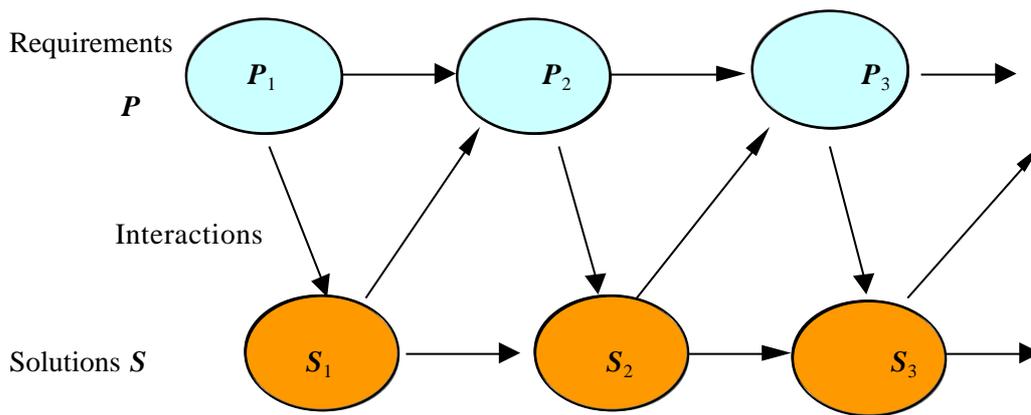


Figure 2. A model of design as exploration (after Maher, 1994)

2.4 COMPARISON OF SEARCH, EXPLORATION AND COEVOLUTION

Design as search finds the best design solutions from a part of the design space that satisfies the pre-defined requirements. Since design requirements are not changed during the design process, the model of design as search is suitable for well-defined design problems.

Design as exploration is able to search other parts of the solution space by changing the design focus. Viewed from different perspectives, different parts of the solution space are considered or possibly different solution spaces become relevant. The difficulty in design as exploration is determining a way to change the focus of design. For instance conventional evolutionary computation which assumes a fixed fitness function does not simply apply to this design model.

Through the interaction of the design requirements and design solutions, design as coevolution explores the spaces of design requirement and design solution simultaneously. Those interactions may add new variables into both design spaces. Usually, the requirements and solution of an ill-defined design problem can not be understood well enough to define a fixed fitness function. The search for potential solutions is often interleaved with changes in the requirements for the solution. Therefore, the model of design as co-evolution is appropriately used in conceptual design, where relatively little is known about the design problem. Figure 3 illustrates the input and output of the co-evolutionary model of design.

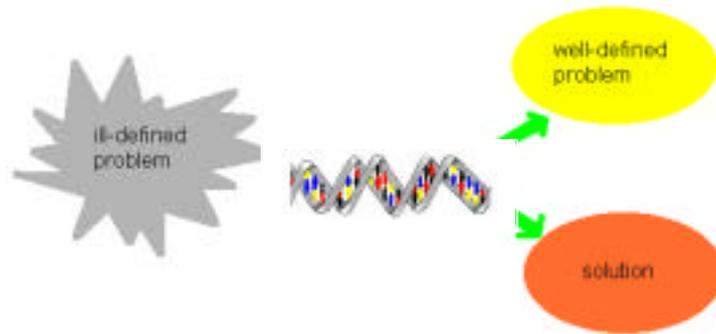


Figure 3. Input and output of co-evolutionary design.

3. Co-evolutionary Design Algorithm

Co-evolutionary design is characterised by having a search space of problem requirements and a search space of problem solutions. The algorithm has two phases, each one corresponding to a simple genetic algorithm. In the first phase, the problem space provides the basis for a fitness function used to evaluate alternatives in the design space. This corresponds to the downward arrow in Figure 2. In the second phase, the solution space provides the basis for a fitness function used to evaluate the problem space. This corresponds to the upward arrow in Figure 2. Each phase is essentially a goal-directed search process with a fixed goal.

First, the requirements space is used to develop a focus, as illustrated in Figure 4. The solution space is a population of alternative designs. Applying the evolution operators, the members of the population in the solution space go through repeated cycles of selection, reproduction, crossover, and mutation. At each generation, the solutions are checked for convergence. When the process has converged, and if the process has not yet reached the termination condition, the focus shifts and the search changes to the problem space.

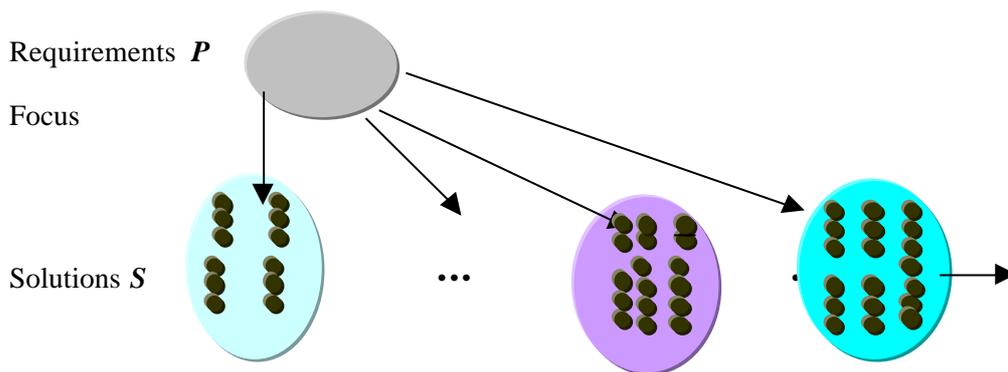


Figure 4. Evolutionary search for design solutions.

After convergence, the solution space is used to develop a focus for searching the problem space, as illustrated in Figure 5. The problem space is a population of alternative requirements. Applying the evolution operators, the members of the population in the problem space go through repeated cycles of selection, reproduction, crossover, and mutation. At each generation, the requirements are checked for convergence. When the process has converged,

the focus shifts and the search changes to the solution space. Logically, termination is not possible at this phase, only after the search for solutions.

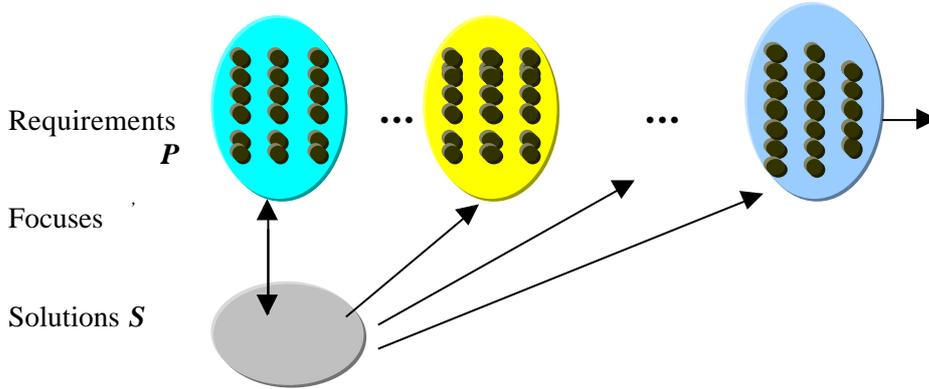


Figure 5. Evolutionary search for design requirements.

Figure 6 presents a general algorithm of coevolutionary design. Each of the two phases of coevolutionary design is a search process using a simple GA and unchanging fitness function, denoted f . Therefore, each phase corresponds to one design focus and a change in phase indicates a change in focus. Using this algorithm for co-evolutionary design we need to reconsider concepts of evolution, their counterpart in GAs and their meaning in the co-evolutionary design process. Fitness is used as the criteria for determining which members of the population are selected to participate in the next generation. Convergence is defined within each phase as the criteria for changing the focus of the search. This distinction ensures that fitness is used to measure and compare the performance of individuals with the same fitness function, and convergence indicates that the current search space is no longer going to be searched. Termination is not tied to fitness or convergence and may be determined by an analysis of the changing focus as recorded in the set P , by a fixed amount of time for which co-evolutionary design is allowed to continue, or by the human designer

```

/* CoDESIGN algorithm */
T = 1; /*co-evolutionary cycle counter */
vp = 1; /*local evolutionary counter of problem space*/
vs = 1; /*local evolutionary counter of solution space*/
u = 1; /*co-evolutionary phase counter*/
t = 1; /* generation counter */
f = {}; /* set of fitness functions*/
initialize genotypes of requirements Pt and solutions St;
While termination conditions, f(T, v), are not satisfied
{
    T = T + 1; /*update the evolutionary cycle counter*/
    u = 1; /*update the evolutionary phase counter*/
    while u < 3 /*two phase coevolutionary design process*/
        switch (u)
        {
            case u = 1:
                /* Phase 1: determine focus for new problem requirements, */
                /* redefine problem requirement space, and search for best problem requirement*/
                if T = 1 /*no evolution in problem space at the first cycle*/
                {
                    ft,T = f(St); /*fitness function*/
                    f = { ft,T }; /*search focus*/
                    Pt = g(Pt, ft,T); /*revised problem requirement space*/
                    t = 1; /*reset the evolutionary counter*/
                }

```

```

While convergency conditions,  $c(t, \dots)$ , are not satisfied
{
 $P_t ::=$  select genotypes in  $P_{t-1}$ ;
reproduce, crossover and mutate genotypes in  $P_t$ ;
calculate fitness of phenotypes in  $P_t$  using  $\tau$ ;
convergence using DNA testing,  $h(v, P_t, P_{t-1})$ ;
 $t=t+1$ ; /*update the evolutionary counter*/
 $v_p = v_p + 1$ ; /*update the local evolutionary counter of problem space */
} /*end of evolution search in problem space*/
} /*end of the phase one*/

case u=2:
/* Phase 2:determine focus for new design solution, redefine design
space*/
/*search for best design solution */
 $\tau_T = f(P_t)$ ; /*fitness function*/
 $\tau = \{ \tau_T \}$ ; /*search focus*/
 $S_t = g(S_t, \tau_T)$ ; /*revised design solution space*/
 $t=1$ ; /*reset the evolutionary counter*/
While convergency conditions,  $c(t, \dots)$ , are not satisfied
{
 $S_t ::=$  select genotypes in  $S_{t-1}$ ;
reproduce, crossover and mutate genotypes in  $S_t$ ;
calculate fitness of phenotypes in  $S_t$  using  $\tau$ ;
convergence using DNA testing,  $h(v, S_t, S_{t-1})$ ;
 $t=t+1$ ; /*update the evolutionary counter*/
 $v_s = v_s + 1$ ; /*update the local evolutionary counter of problem space */
} /*end of evolution search in solution space*/
break; /*end of the phase two*/
} /*end of a coevolutionary cycle*/
} /*end of coevolutionary design */

```

Figure 6. CoDESIGN: an algorithm for co-evolutionary design.

The algorithm is illustrated in Figure 7 graphically. The concept of a “generation” or “cycle” in co-evolution computing has different meanings from the simple GA since there is more than one population that is evolving. Therefore, we use three counters for time: T, t, v and u. The co-evolutionary cycle counter, T, indicates the change in generation from one full cycle of coevolution to another. T is updated after the solution space and the problem space have been searched and converged. The generation counter, t, indicates a new generation in the simple GA sense. t is updated after each generation of search in one of the spaces; that is, after a cycle of selection, crossover, and mutation. The local evolutionary counter, v, indicates the number of generations in each search of the problem or solution space. Its value is reset once a full cycle of coevolution is completed so that v is a local counter for evolutionary cycles. The co-evolutionary phase counter, u, indicates the current status of coevolution process. Its value is reset at the beginning of every coevolutionary cycle.

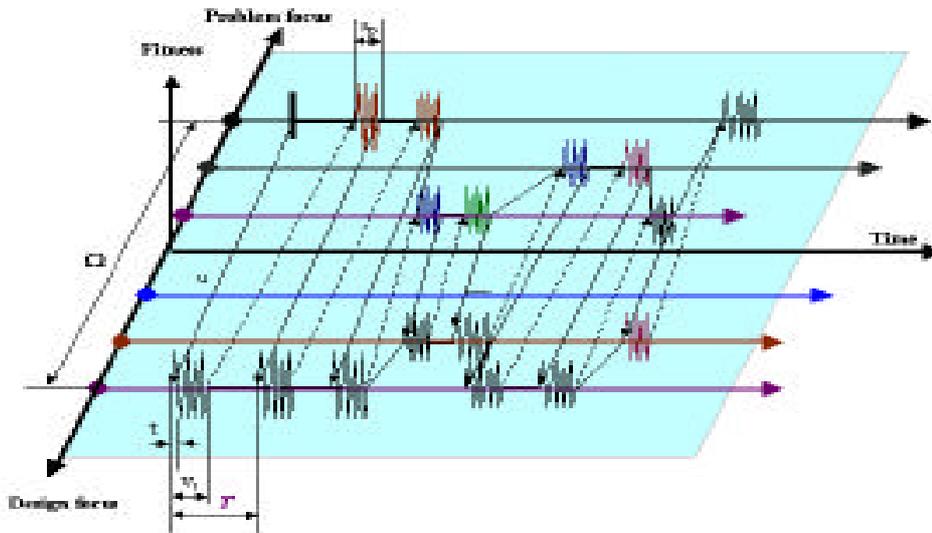


Figure 7. Illustration of the co-evolutionary design algorithm.

Figure 8 is a demonstration of the co-evolutionary algorithm. The first generation of search occurs in the solution space using the given set of requirements. During this first phase there are 0 generations in the problem space and 8878 generations in the solution space. Following the search in the solution space, the co-evolutionary design algorithm searches the problem space for 256 generations. The demonstration shows that there were 5 complete co-evolutionary cycles where each cycle has 2 phases.

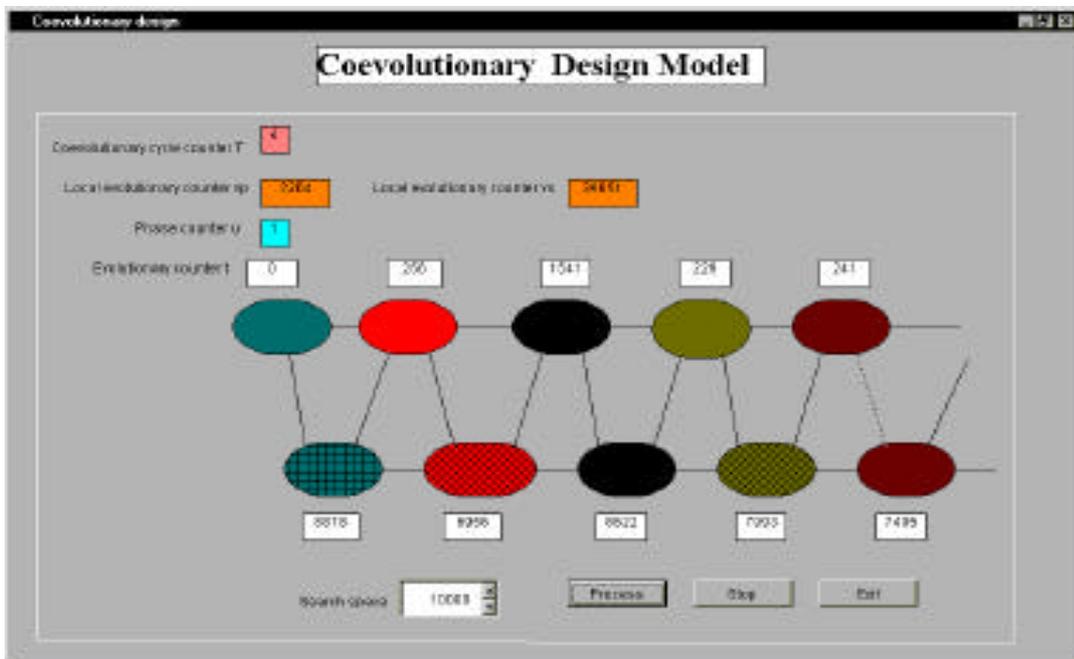


Figure 8. Demonstration of the coevolutionary algorithm.

The concepts of fitness, convergence and termination are discussed below.

Fitness: Survival of the fittest in evolution has been translated as a fitness function in simple GAs. This fitness function is the basis for the comparison of alternative solutions. In design, when we let the definition of the fitness function change, the value of the fitness function can no longer serve as the basis for comparison for all alternative designs. The performance of individuals in the solution space can only be compared when they are evaluated using the same fitness function. This makes it difficult to compare the performance of solutions across different phases of the co-evolutionary design process. The performance of individuals is used to determine which members of the population “survive”, or are selected to participate in the next generation of search. When searching a space, either the problem space P or the solutions space S , the performance is measured by how well the alternatives satisfy the focus. A focus for the search for a design solution is a function of the set of possible design requirements, and a focus for the design requirements is based on the current set of design solution alternatives. In the CoDESIGN algorithm, we represent the set of possible design requirements in the space P and the corresponding focus, or fitness function, as f_t . We represent the current set of design solution alternatives in the space S and the corresponding focus as f_T . The fitness function, f , is defined as a function of P , or $f(P)$. The subscripts of f are t and T , where t indicates which generation of P was used to derive f_t and T indicates which generation the f_T is used. There will be a new f_t for each T , but not for each t , since each phase may result in multiple generations of P or S but the focus stays the same.

Convergence: Convergence in evolutionary algorithms means that the search process has led to the “best” design in terms of the specified fitness function. Convergence is typically the criteria for termination of the evolutionary search process. Since the fitness function in coevolutionary design changes from one phase to another, the idea of convergence needs to be reconsidered. This requires a consideration of the purpose of coevolutionary design as compared to evolutionary search. The purpose of evolutionary search is to find the best solution based on a given environment, where the environment is effectively represented by the fitness function. The purpose of coevolutionary design is to explore both the problem and solution spaces, allowing both to change in reaction to each other, until a satisfactory combination of a problem statement and solution state is found. The exploratory nature of the coevolutionary process implies that the process should search until the potential for new ideas is reduced. We propose then, that convergence is not related to fitness, but to the similarity of the members of the population. A population in which there is little change in the genotypes of the members when compared to the previous population indicates that the search process has converged.

Termination: The link between convergence and termination in evolutionary algorithms occurs because the convergence to the “best” solution indicates that the search should be terminated. In coevolutionary design, convergence is determined for each phase of the search, that is, for a given focus, and following the convergence for one focus, another focus is determined and search commences in the other space. This indicates a separation of termination and convergence. We use termination to indicate when the coevolutionary process should stop, and convergence to indicate when the search in a given space for a given focus should stop. One criterion for termination is the number of cycles of the coevolution process. This is indicated in the algorithm as T . This criterion is equivalent to setting a time limit for the design process. Often, the time limit is a major criterion for signalling when exploration of changes in problem and solution should stop. Another criterion for termination is similar to the convergence criterion above – there are no new fitness functions being found. With each change in phase, the fitness function is appended to a list of fitness functions, labelled f_t in the algorithm. A criterion for termination is when there are no new fitness functions added to f_t .

The significance of this criterion is that the algorithm is not able to identify a different focus for the design and therefore, new ideas have been exhausted.

Interaction. Interaction is relevant only in the context of coevolution, since there are two search spaces. Interaction provides a mechanism for transferring knowledge from one space to another, with the potential to expand the boundaries of a search space during the design process. Interaction between the requirements and solution spaces can be modeled as passing variables from one space to the other. The vehicle for passing variables is the focus, or fitness function, .

4. Summary

This paper formalises the model of co-evolutionary design using the terminology of search and exploration. Traditional evolutionary computing can be considered as a case of coevolution. Coevolutionary design has the following features:

- Co-evolutionary design consists of two spaces/populations: requirements of design and solution of design.
- Co-evolutionary design consists of two iterative phases: search for design requirements and search for design solutions.
- The focus of search in one space is determined by the current population in the other space.
- Interactions may add new variables to both spaces, which may lead to unexpected design requirements or solutions.
- Genetic changes in co-evolutionary design consist of two aspects: changes in design focus and changes in genotype.
- Fitness is local and changes in each phase.
- The fitness value is not comparable across different phases.
- There is no relationship between convergence and fitness: fitness is used to determine which individuals survive and convergence occurs when new ideas can not be found.
- The termination conditions do not rely on the fitness of individuals.

The model of evolutionary design has been used to explain real design projects such as the Sydney Opera House [Maher and Poon, 1996] and as the basis for developing a computational model of the design of braced frames [Maher, et al, 1995] and floor plans [Poon and Maher, 1997].

Acknowledgments

This work is supported by a grant from the Australian Research Council.

References

- Hillis, Daniel W.: 1991, Co-evolving parasites improve simulated evolution as an optimization procedure. In: Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen (eds) *Artificial Life II*, Santa Fe Institute Studies in the Sciences of Complexity, vol. X, Addison-Wesley, pp 313-324.
- Gero, J.: 1992, *Creativity, emergence and evolution in design* in Gero J. and F. Sudweeks (eds) *Preprints Computational Models of Creative Design*, University of Sydney, pp 1-28.
- Maher, M.L.: 1994, *Creative Design Using A Genetic Algorithm*, *Computing in Civil Engineering*, ASCE, pp 2014-2021.

- Maher, M.L., Poon, J., Boulanger, S.:1995, Formalising design exploration as co-evolution: A combined gene approach, in J.S.Gero and F. Sudweeks (eds), *Advances in Formal Design Methods for CAD*, Chapman & Hall, pp 1-28.
- Maher, M.L. and Poon, J.:1996, Modelling design exploration as co-evolution, *Microcomputers in Civil Engineering*, 11: pp 195-210.
- Mendelson,:1963,: *Introduction to Topology*, Blackie & Son Limited.
- Paredis J.: 1995, *Artificial Coevolution, Explorations in Artificial Life*, AI Expert Presents, Miller Freeman Inc
- Richard N. L.,; 1995, *The Coevolution of Technology and Organization in the Transition to the Factory System*, Report-no: 95-153, Department of Economics, University of Connecticut, 1995
- Simon, H.A.:1969, *The Sciences of the Artificial*, MIT Press.
- Smith M.J.: 1989, *Evolutionary Genetics*, Oxford University Press.