Applying Schooling Genetic Algorithms to Evolutionary Design

Patrick T. Wanko, Paul M. Stanfield
Department of Industrial and Systems Engineering
NCA&T SU, Greensboro, North Carolina 27411, US

Abstract

Evolutionary design is a lifecycle-based incremental and iterative design process. Recent changes in product nature (configurability, programmability, ambient intelligence, adaptability, networkability) provide new opportunities in terms of evolutionary design. Schooling genetic algorithms (SGA) provide a metaheuristic tool that imitate fish schooling dynamics with a genetic algorithms (GA) underpinning. SGA was designed to work on problems where grouping, evolvability, and sustainability are valuable in the solution. In this paper, we first present an evolutionary design concept we call the product life cycle product design (PLPD). Then we apply SGA to product life cycle management following the PLPD concept. The results are presented and more insights derived.

Keywords
Systems engineering, schooling genetics, genetic algorithms, life cycle, evolutionary design

1. Introduction

In this paper, evolutionary design is defined as a lifecycle-based incremental and iterative design process resulting from the ongoing durable product evolution. The design of durable products, such as automobiles and aircraft, has expanded from traditional mechanical design to include more biologically inspired capabilities – learn, morph, communicate, and sustain. The trend of using analogies to biological systems to develop solutions for engineering problems, also called biologically inspired design, is somewhat new and keeps gaining importance as a wide-spread movement in design for sustainable development [1-4]. The changes have resulted in terms such as “Evolving Parts/Products Families” [5-6] to address and describe the changes occurring to those product families as mutations, with product features losses and gains through generations, and the appearance of new families of products. The change in product nature, whether in terms of configurability, programmability, ambient intelligence, or adaptability, provides new opportunities in terms of design. As the transition from mechanical to biologically inspired product design progresses, new processes and mathematical tools are necessary to integrate a product’s enhanced data capabilities across the life cycle to minimize cost of ownership, extend life, and enhance sustainability.

This paper is structured as follow. First, the limits of the traditional systems engineering (SE) views of design are explored and a new product design approach called product lifecycle product design (PLPD) is introduced. After that, schooling genetic algorithms (SGA), a new GA-based enhanced metaheuristic search approach introduced by Wanko and Stanfield in 2011 [7] is reviewed and updated. Next, an application of SGA to PLPD is presented and the obtained results interpreted. The paper ends with a conclusion.

2. Product Design

Systems engineering (SE) as a scientific approach, has been around since the 1940s and has significantly evolved from its prior engineering approaches. Traditional SE is a mature field with well-established design processes.

2.1 Current Limits

Traditionally, SE has emphasized: (1) design optimization into a fixed configuration, (2) system decomposition in order to facilitate system analysis, and (3) the guiding role of systems engineers to design and maintain systems. Such an emphasis does not account for “intelligent” products and/or product parts, and tend to make SE heavily rely on the design engineer’s specialized knowledge and expertise. These limitations, and the increasingly shortened life cycle of products [8] make it difficult for engineers to innovate and to sustain their design. Given the nature of smart systems, the systems engineering process can be enhanced by building in biologically-observable processes supporting evolvability, grouping and sustainability. The challenge is to characterize and
represent the life cycle engineering problem and then create the methods to incorporate product evolvability (enabled by modularity, interoperability, and software level configurability), grouping (enabling system efficiency due to the economy of scale), and sustainability (requiring minimal intervention)? The product life cycle product design (PLPD) approach is summarized as a means for problem characterization and schooling genetic algorithms (SGA) applied as the method using the lifecycle data to enable product evolvability, grouping, and sustainability.

2.2 Product Life cycle Product Design
A general categorization of lifecycle data was made. The categorization was based on parameters factors, from conception to retirement or recycling, that are key indicators of a product’s performance. Life cycle parameters, the identified factors that impact the performance of a product, are divided into three categories namely design, operational, and environmental. Figure 1 shows the relationships between the parameters and the performance within the systemic view of a product.

![Systemic view of a product](image)

Having the performance of a product as a function (simulated by a model or observed in real system) of the interactions between that product design, operational, and environmental parameters offers new opportunities. Figure 1 also illustrates individual learning by a product through user-product, product-environment, and product-product interactions. Table 2 shows a sample of parameters. Engineers set the design parameters of any product before production. The design parameters are infrequently changed. Operational parameters are those that can be changed by the user to meet a specific use or need. Operational parameters are frequently changed throughout the life cycle of a product. The environmental parameters are tied to the environment and are out of the designers and the users’ control.

<table>
<thead>
<tr>
<th>Design</th>
<th>Operational</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
<td>Type of use</td>
<td>Temperature</td>
</tr>
<tr>
<td>Material</td>
<td>Frequency of use</td>
<td>Humidity</td>
</tr>
<tr>
<td>Functions</td>
<td>Type of Maintenance</td>
<td>Culture</td>
</tr>
<tr>
<td>Sensors</td>
<td>Frequency of Maintenance</td>
<td>Infrastructure</td>
</tr>
<tr>
<td>Size</td>
<td>Custom settings</td>
<td>Regulations</td>
</tr>
<tr>
<td>Shape</td>
<td>Number of resets</td>
<td>Similar Products</td>
</tr>
<tr>
<td>Dimension</td>
<td></td>
<td>Climate</td>
</tr>
<tr>
<td>Manufacturing Process</td>
<td></td>
<td>Location</td>
</tr>
</tbody>
</table>
Building on the above proposed LCE data classification of life cycle data, a metaheuristic tool named schooling genetic algorithms (SGA) was created.

3. Schooling Genetic Algorithms
There are few, if any, generalized metaheuristic methods dealing with problems where grouping, evolvability, and sustainability are part of the characteristics of the solution. A schooling genetic algorithm (SGA) is a GA-based metaheuristic model that enacts process and operator adaptability to mimic fish schooling. SGAs represent a new type of metaheuristic where operators behave differently depending on fish schooling dynamics. SGA is new, and has evolved since its introduction [7] into a more stable and complete (both concept and implementation wise) metaheuristic.

3.1 SGA Procedure
A fish, representing candidate solutions to a problem in SGA, randomly but collectively “swim” and discover new places to feed within the sea. The sea is the search domain. A school can be as small as one fish, or as wide as the entire population of fish within the sea. A school is any set of fish that are “similar” enough to belong to the same group. Within the SGA, each fish must belong to one school. School membership is binary.
SGA takes from fish schooling the concepts of “food foraging”, “predator avoidance”, and “school maintenance”. All concepts are implemented via the use of the crossover and the mutation operators. The crossover and mutation operators are used according to the way the immediate local search domain is “perceived” by a school of fish. The spatial perception by a school of fish is what ultimately set the behavioral mode of that school. The spatial perception of any school is assessed by comparing the fitness value of the center of mass of that school to the average of all the existing schools center of mass fitness values. Figure 2 shows how the behavioral modes are assigned. $Performance_{Group}$ and $Performance_{Average}$ represent the fitness value of a school’s center of mass fitness, and the average of all the schools’ center of mass fitness respectively. $low\_cut\_off$ and $high\_cut\_off$ represent arbitrary constant values used to set the threshold for the assignment of the schools’ behavior mode. The SGA implements three behavioral modes, each named after the fish schooling concept listed above. Effects such as “school division” and “school formation” occur as a consequence of the contextual application of the mutation and crossover operators.

![Figure 2: Fish school behavior assignment](image-url)
In food foraging mode, a school looks for feeding grounds. Each time a new feeding ground is found, the school consumes it, and then moves on to look for another (not yet fed on) location to exploit. When in school foraging mode, a school has a high crossover rate (number of offspring to result from the crossover process), and a low mutation rate (number of offspring to result from the mutation process). The act of leaving a location after feeding from it is to prevent any school, for time to time, from being trapped in an optimum (whether local or global) solution. The behavior of depleting the food when feeding is also intended for the schools to explore new areas of the sea in which the global solution may be discovered.

The implementation of the food depletion relies on the periodic use of the fish clustering and a recency tabu list. Performing school clustering periodically enables schools in a given behavioral mode to stay in that mode for a given (short) period of time. The recency-based tabu list can serve many purposes, with one of them being providing medium-term knowledge of the search history. During the metaheuristic search process, all known better solutions are stored in a tabu list. The solutions are updated or removed to the list on a first-entered first-removed basis. While a solution is still in the tabu list, no school is allowed within a given close proximity of that solution.

During the schooling process, schools may encounter predators and are forced to escape using various strategies. SGA mimics such a behavior by first defining a predator as any region of the search domain where solutions are worse than the currently obtained average solutions. A predator avoidance mechanism built into each school allows the fish of that school to swiftly escape predators while looking for feeding grounds. During a predator avoidance maneuver, an escapee fish is first selected. Then, the mutation operator is used to create “mutant” fish in the direction of the escapee fish. Last, the crossover operator is used to breed fish between the “mutant” fish, and the original school. Figure 3 shows how the predator avoidance maneuver operates. During predator avoidance, the mutation magnitude (defined here as the length of the phenotype to be affected by the mutation process) is set high, and the application of the mutation operator precedes that of the crossover.

School maintenance is the default behavioral mode in SGA. When fish are not foraging or escaping predators, they are foraging. In foraging mode, both the crossover rate, and the mutation rate of a school should be the same. The sea can host many schools at a time. With SGA, within school and between schools’ interactions are behavior based. Applying genetic operators such as mutation and crossover carries out the interactions between fishes and between schools. The SGA has a dynamic nature due to the way (when and how) both the mutation and the crossover operators are used, that is, according to the size of a school, the overall average fitness of that school, and the relative perception of the local search domain by the school.

3.2 Applying SGA to PLPD

The life cycle data characterization can be used to encode product entity in a form suitable for use in the SGA algorithm as shown in Figure 4. An experimentation was carried to find out the potential benefits of using SGA and
PLPD in the design process of a system/product. Within the designed SGA test bed created for this paper, values for $n$, $m$, and $p$ representing the number of genes to encode the design, operational, and environmental parameters respectively, were set to 1.

Being able to assess the effects of grouping on performance when running SGA is important. Such an assessment assists in interpreting the solutions returned by SGA, and possibly helps objectively mapping the fitness values of the final solutions. A solution quality assessment indicator was defined as the trait performance indicator (TPI). The TPI was defined such that its value would indicate how important (0.0 to 1.0) any given phenotype is to the observed performance values. Considering all the final solutions returned by the SGA implementation, the wider the spread over the range of permissible values for a given phenotype, the higher the importance of that phenotype. Let $BP$ be the set of all the final solutions returned by the SGA, and $P^j$ any element of $BP$. $P^j$ is a solution candidate. Let $P^j_i$ designate a trait $i$ of the solution $P^j$. Ideally, each trait has a range of values (constraints on trait) that fall between a minimum and a maximum. Let $E$ designate the set of minima and maxima for all the traits of the phenotype, with $E_i^m/E_i^M$ representing the minimum/maximum value for trait $i$. For a phenotype subset of $n$ traits, the TPI value is computed in a way similar to a relative error, giving an indication of how important the selected set of traits is to the observed performance, relatively to the whole phenotype:

$$TPI^j = \frac{\sum_{i=1}^{n} (\max\{P^j_i \in BP\} - \min\{P^j_i \in BP\})}{\sum_{i=1}^{n} (E_i^M - E_i^m)}$$

The TPI values account for all the solutions returned by the SGA. TPI values should inform the systems engineer about a stricter constraint that can be imposed on a given parameter without sacrificing the system overall performance. The possible implications of a TPI value are as follow:

- The corresponding or involved component of the system/product can be changed to another less accurate, sensitive, or performing without hurting the system performance.
- The corresponding or involved attribute can be pushed from one category (design, operational, environmental) to another without hurting the system performance.

3.3 SGA Approach to the PLPD Model

Using probability distributions, constrained continuous random values were generated that represent the design, operational and environmental parameters of a hypothetical product. Solutions were randomly assigned to one of two environments. The proportion of the number of products available for each environment over the number of products that is available overall was set and kept constant throughout each simulation. Simulations were carried using one or more parameters of each LCE type. LCE data was generated to test PLPD with SGA for the following scenario: environment parameter driving design for performance.
3.4 Experimental Design
For the experiments, each candidate solution was a 3-dimensional vector of the search space. Each parameter was coded with a 10-bit string value (actual decimal values falling within the range of -500 to 500). The size of the population was set and maintained at 120 fish. The proportion for each environment was 55% for environment A, and 45% for environment B. The stopping criteria was set to be either the maximum number of generations (set to 1000), or the lack of improvement of the average fitness value of the population for three consecutive generations, whichever came first. The above described experiment was repeated ten times and the recorded metrics were:

- the number of generations it took for SGA to converge (if convergence occurs)
- the distribution of the behavioral states
- the best solutions obtained
- the best school, and
- the TPI values of each dimension (allele) of the product.

3.5 Experimental Setup
The objective of the experiments carried here was to determine the ability of SGA to effectively use grouping to converge on separate environments, and to check the ability of SGA to characterize the life cycle parameters’ relationship. The relation used for the experiments was linear as follows:

\[
f([D, O, E]) = \begin{cases} 
1 - \frac{D}{\text{max}_D} - 0.7 & \text{if } E \text{ is } A \\
1 - \frac{D}{\text{max}_D} - 0.3 & \text{otherwise}
\end{cases} \quad -500 \leq D, O \leq 500
\]

Using a linear relationship, the expected characterization is vertical lines within the (D, O) plane. The lines should contain the best solutions of the problem. This test is for the case of a product with a simple (linear) performance function that depends on just two life cycle attributes.

3.6 Experimental Results and Interpretation
Plotting the contents of the experiment’s tabu list showed that the SGA was able to capture the nature of the relationship between the LCE parameters. Figure 5 shows the plot of the tabu list contents. The best performance for environment A (red dotted line) and environment B (blue dotted line) are both plotted. The contents of the tabu list tend to follow the line that the plot of the performance within (D, O) plan would return for each environment. The observed trend gives an idea of the embedded relationship between performance and LCE parameters.

Figure 5: Tabu list contents plot for one experiment
Table 3 shows the contents of the SGA’s tabu list for experiment 1. As a reminder, the tabu list is recency based with elements representing the best feeding locations recorded by SGA during the search process. Along with the locations, are listed the composition, environment wise, of the number of fish that made that school.

<table>
<thead>
<tr>
<th>Design</th>
<th>Centre of Mass</th>
<th>Tabu Performance</th>
<th># Fish From A</th>
<th># Fish From B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>209.2005</td>
<td>-169.253392 A</td>
<td>0.881598961</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>156.6719</td>
<td>248.6509379 A</td>
<td>0.986656224</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>146.0206</td>
<td>-229.1916984 A</td>
<td>0.992041137</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>135.1326</td>
<td>64.83714354 A</td>
<td>0.970265136</td>
<td>19</td>
<td>7</td>
<td>26</td>
</tr>
<tr>
<td>148.7408</td>
<td>257.8771119 A</td>
<td>0.997481554</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151.2075</td>
<td>118.6229351 A</td>
<td>0.99758492</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>148.7791</td>
<td>-77.94622939 A</td>
<td>0.997558292</td>
<td>43</td>
<td>0</td>
<td>43</td>
</tr>
<tr>
<td>150.5193</td>
<td>84.01986896 A</td>
<td>0.998961384</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>350.6195</td>
<td>115.1523153 B</td>
<td>0.998761058</td>
<td>0</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>354.9097</td>
<td>209.0342458 B</td>
<td>0.990180656</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>348.7395</td>
<td>-84.6879403 B</td>
<td>0.99747906</td>
<td>0</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>349.7957</td>
<td>98.50342012 B</td>
<td>0.99991313</td>
<td>0</td>
<td>19</td>
<td>19</td>
</tr>
</tbody>
</table>

The center of mass of each cluster was calculated, as the centroid of all the continuous parameters of the fish making up the school. The assignment of a school’s center of mass to a given environment, only discrete parameter in this case, depended on the environment the majority of the fish of the school belonged to. In case of equal provenance of fish from each school, the environment of the center of mass was randomly picked. The assignment of the center of mass is critical as the fitness values of the centers of masses are what the SGA uses to compute the baseline necessary to set the behavior of all the existing schools.

The contents of the tabu list were used to calculate the TPI. TPI values are available in Table 4. The interpretation of those values is straightforward. The TPI for the design parameter (0.2523) indicates that the recorded best performance was achieved for values of the design parameter occupying just 25.23% of the available range [-500, 500]. This means that the best performances recorded can still be achieved, even if the designers was to only consider that small range of values for the design parameters. The TPI value for the environment (1.0) reflects the fact that the search process was concurrently operated on two environments, and that the performance function was environment dependent.

Table 5 shows a sample of the most diverse solution from the final population as returned by the SGA. Unlike Table 3, Table 5 lacks diversity. The lack of diversity is attributed to the convergence of the SGA.

Table 1: Table of Experiment 1

<table>
<thead>
<tr>
<th>Design</th>
<th>Operational</th>
<th>Environmental</th>
<th>Tabu</th>
<th># Fish From A</th>
<th># Fish From B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>350</td>
<td>-85.75016356</td>
<td>B</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>350</td>
<td>-85.75016356</td>
<td>B</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>-79.03928056</td>
<td>A</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>-79.03928053</td>
<td>A</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>-79.03928051</td>
<td>A</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 6 shows a plot of the final results for experiment 1. From Figure 6, it can be observed that as the population’s average fitness improves, the number of schools decreases to a final count of 2, matching the number of environments.

Still from Figure 6, the lower right graph represents the number of schools that were on a given behavioral mode during a given generation. School maintenance (SM) is in red (R), predator avoidance (PA) is in green (G), and food foraging (FF) is in blue (B). Although all behavioral modes were used early during the simulation, by the time the final count of 2 is reached, improvements per generation to the population’s average fitness became too small and school maintenance became the predominant behavior for the schools. Such a change in the overall behavioral pattern of the schools was expected as the population gets more uniform as the number of generations grows. Similar behavior was predicted of the number of schools that went down to 2.

4. Conclusion
Within the paper, a new product design approach for evolutionary design named product life cycle product design (PLPD) was presented. Schooling genetic algorithms (SGA), a new metaheuristic created to work on problems where grouping, evolvability, and sustainability are characteristics of the solution was explored, and was applied to the PLPD methodology of continuous product design. Simulations of SGA were carried out according to a set of factors including the relationship between life cycle parameters, the number of parameters, and the nature of the search domain. The quality of the solutions returned by SGA was assessed using different metrics including the TPI values for each LCE parameter type, and a graph for the distribution of behavioral mode.

Running experiments where the LCE factors were bound by a linear relationship helped to better understand how SGA built-in grouping shapes both the search process and the interpretation of the returned results. As the number of generations increased, a point convergence was not observed for each environment. Instead, convergence along the line describing the relationship between the LCE parameters was observed. Proceeding further with the testing of the SGA metaheuristic, more experiments with new LCE parameters relationships and shapes of the search spaces will be carried to further our assessment of the suitability of SGA to evolutionary design.

Acknowledgements
This research is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
References