AN ARTIFICIAL LIFE APPROACH TO EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION

Srinivas Shiva Kar Vulli
Department of Computer Science

Sanjeev Agarwal
Department of Electrical Engineering

ABSTRACT
Current evolutionary multiobjective optimization techniques while effective in approximating the shape of the Pareto front, suffer from high computational complexity due to explicit diversity management, archiving of good solutions, and Pareto ranking for selection and recombination. In this paper, an evolutionary algorithm inspired by artificial life (ALife) and naturally occurring ecosystems is proposed for multiobjective optimization. The algorithm uses a predator-prey model, wherein the multiobjective problem at hand is encoded into the prey and predator’s genotype. The solution space is represented by prey. The fitness is evaluated through interactions between predators and prey. The proposed algorithm mimics a self-sustaining ecosystem, i.e., even after converging to the Pareto front, the interactions between predators and prey continue without any adverse affect on the solution. As a result the proposed algorithm appears promising for dynamic multiobjective optimization problems.

1. INTRODUCTION
Over the last two decades, evolutionary algorithms have emerged as a prominent approach for multiobjective optimization [1-3]. Although existing methods are effective in approximating the shape of the Pareto front, they suffer from high computational complexity due to the very mechanism employed to achieve good results, namely, explicit diversity management, archiving of good solutions, population wide Pareto ranking for parent selection and recombination at every generation. Popular Multiobjective Evolutionary Algorithms (MOEAs) which employ these techniques include NSGA-II [4], SPEA [5], and e-MOEA[6].

Several efforts have been directed towards moving away from the above trend and demonstrated reasonable results. Laumanns et. al. [7], were the first to consider a predator-prey interaction scheme for multiobjective optimization (MOO). In the algorithm, Preys representing possible solutions were placed at vertices of a graph and were stationary. Predators traversed the graph randomly, killing prey only within their immediate neighborhood and according to one of the objective. Since each prey is evaluated by predators representing different objective only those prey which performed well with respect to all the objectives survived. Any vertex vacated due to killing of a prey is repopulation by recombining prey randomly in the neighborhood. The results although proved that this approach can be used for MOO, were not comparable to the existing MOEAs. Grimme et. al. [8], and later Deb et. al. [9], improved upon the original predator-prey algorithm to incorporate new reproduction, diversity preservation and elitism operators.

Berry and Vamplew [10] proposed an artificial life based MOO approach excluding the need for population wide Pareto ranking and explicit diversity management. The model consisted of two distinct components: agents representing solutions in the search space and an environment responsible for distributing resources for the various objectives. An agent survives on the resources which are granted based on the performance of the agent for that particular objective which the resource represents. This model was later improved to include an accretion based reproduction [11]. In both the cases, the model was reported to perform comparable to traditional MOEAs.

As evident from the previous literature, the idea of ALife models for MOO is not a novel one. However, the approach presented in this paper is inspired by self-sustaining and self-regulating nature of naturally occurring ecosystems and is considerably different to the existing literature. From a theoretical perspective, the algorithm appears suitable for dynamic multiobjective optimization; however, further research is required to validate any such claims.

2. MULTIOBJECTIVE OPTIMIZATION
Multiobjective optimization refers to the class of optimization problems involving multiple, often conflicting objectives. Since conflicting objectives cannot be satisfied simultaneously, optimal solutions, in these cases refers to solutions where no further improvement with respect to one objective can be achieved without deteriorating with respect to some other objective. Therefore, MOO problems have not one, but a set of optimal solutions, called Pareto-optimal set or Pareto-front.

Definition 1: A general multiobjective minimization problem with \(m\) decision variables (parameters) and \(n\) objectives is written as:

Minimize \(y = f(x) = (f_1(x), \ldots, f_n(x))\)
\[
\begin{align*}
\text{where} \quad x &= (x_1, \ldots, x_m) \in X \\
y &= (y_1, \ldots, y_n) \in Y
\end{align*}
\]

where \(x\) is the decision vector, \(X\) is the decision space, \(y\) is the objective vector, and \(Y\) is the objective space. A decision vector \(a \in X\) is said to dominate a decision vector \(b \in X\), written as \(a \prec b\), if and only if
\[
\forall i \in \{1, \ldots, n\}: f_i(a) \leq f_i(b) \quad \land \quad \exists j \in \{1, \ldots, n\}: f_j(a) < f_j(b)
\]

Based on the above definition, a set of solutions \(X^*\) is called a global Pareto-optimal set if and only if
\[
\forall a' \in X^*: \nexists a \in X: a \prec a'
\]

3. MODEL SETUP

The model consists of three components – predators (non-evolving), prey (evolving) and the environment in which predators and prey exist. The decision vector of the optimization problem encoded as the genotype of the prey. The objective vector is employed by the predators to identify and kill weak prey. At each time instant the prey population represents the current approximation of the Pareto front. The quality or fitness of the solutions is evaluated through the interactions between the predator and prey. Preys move around randomly in the environment, interact with each other for mate selection and reproduction and die, either due to old age or due to predation. In this model, the predators do not evolve, reproduce or die, but simply move around the environment, killing weak prey.

3.1. Reproduction

A density dependent offspring size is adopted in this algorithm, i.e., a prey produces more offspring if it has experienced during its lifetime a lower prey population density and lesser number of offspring if it has experienced a higher prey population density.

At every time step \(t\), each prey senses the number of preys (including itself) \(n_i^t\) within an interaction area \(I_a\). An approximation of the current density with respect to the current individual, \(o_i^t\), called observed density is defined as the ratio \(n_i^t / I_a\). The average population density experienced by the individual, \(i\), at time step \(t\) is therefore given by
\[
d_i^t = d_i^{t-1} + \lambda (o_i^t - d_i^{t-1})
\]

where \(d_i^t\) is population density experienced by the individual, \(i\), at time \(t\), \(o_i^t\) is the observed density at time step \(t\) and \(\lambda\) is the update rate.

It is assumed that each individual has the knowledge of the desired equilibrium population density. This number is called genetic density \(C_d\) and relates to the maximum population size setup for the experiment. Based on the local population density sensed by an individual prey, the number of offspring produced by that prey is given as
\[
b_i^t = \begin{cases} 
P_b \left(1 + \alpha \left(1 - \frac{d_i^t}{G_d}\right)\right) & \text{if } d_i^t < \frac{1 + \alpha}{\alpha} G_d \\
0 & \text{otherwise}
\end{cases}
\]

Where \(b_i^t\) is the number of offspring, \(\alpha\) is the rate parameter and \(P_b\) is the birth rate such that
\[
P_b = \frac{P_d}{(1 - P_d)^m}
\]
where \(P_d\) is a constant death rate and \(m\) is the maturity age beyond which a prey can reproduce.

Each mature prey during its reproduction stage searches for each offspring spawned, a suitable mate (mature) within its interaction area. If a suitable mate is found, then the offspring genome is derived using the Simulated Binary Crossover (SBX) [12] recombination operator. Mutation is applied to the offspring's genome with a small probability of \(P_{\mu}\).

3.2. Death

Prey can either die probabilistically or due to predation. At each time step, each prey faces a constant probability \(P_d\) of natural death.

During each step, the predators search the environment within a predation radius \(R\), and attempt to kill \(N_k\) young (immature) dominated prey with a constant probability of \(P_p\).

3.3. Initialization

At the start of the simulation run, \(N\) preys are randomly placed in the environment, along with \(N_p\) predators. The prey genotype (decision vector) is initialized randomly within the valid range of the decision vector. Table 1 provides an overview and values of the parameters used in the model.

4. ALGORITHM

Listing 1, Listing 2 and Listing 3 show the main, prey and predator loops for the algorithm. The prey loops are executed for each prey pseudo-parallelly, i.e., although the prey loop is executed one prey at a time, the order of preys is randomly selected each time step to eliminate any boundary effects. The same is true for predator loop. At the end of each time step the age of all surviving preys and predators is incremented by one.
Each prey is associated with five behaviors (or functions) it executes each time step, namely, move, interact, select mate, reproduce, death.

Each predator is associated with three behaviors, namely, move, evaluate prey, kill prey.

5. BENCHMARK PROBLEMS

Several aspects of a problem which may cause difficulties for multiobjective optimization algorithms were identified by Deb [13]. According to these suggestions, Zitzler et al. [14] suggested six test problems, namely convex (ZDT1), nonconvex (ZDT2), discontinuous fronts (ZDT3), multi-modal (ZDT4), deceptive binary coded (ZDT5) and non uniform (ZDT6). Of these the proposed algorithm was tested on ZDT1 – ZDT3. All the test function have two objectives, with the first objective $f_1$ being a function of only the first decision variable, and the second objective $f_2$ being a function of $f_1$ and the rest of the decision variables. The details of the test problems are as follows:

5.1. ZDT1

This test function has a convex Pareto-optimal front:

$$f_1(x_1) = x_1$$
$$g(x_2 \ldots x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^{n} x_i}{(m-1)}$$
$$h(f_1, g) = 1 - \sqrt{f_1 / g}$$
$$f_2(x) = g(x_2, \ldots, x_n) h(f_1(x_1), g(x_2, \ldots, x_n))$$

where $m = 30$ and $x_i \in [0,1]$. The Pareto-optimal front is formed with $g(x) = 1$.

5.2. ZDT2

This test function has a nonconvex Pareto-optimal front:

$$f_1(x_1) = x_1$$
$$g(x_2 \ldots x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^{n} x_i}{(m-1)}$$
$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2$$
$$f_2(x) = g(x_2, \ldots, x_n) h(f_1(x_1), g(x_2, \ldots, x_n))$$

where $m = 30$ and $x_i \in [0,1]$. The Pareto-optimal front is formed with $g(x) = 1$.

5.3. ZDT3

This test function's Pareto-optimal front consists of several noncontiguous convex parts:

$$f_1(x_1) = x_1$$
$$g(x_2 \ldots x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^{n} x_i}{(m-1)}$$
$$h(f_1, g) = 1 - \sqrt{f_1 / g} - \left(f_1 / g\right) \sin(10 \pi f_1)$$
$$f_2(x) = g(x_2, \ldots, x_n) h(f_1(x_1), g(x_2, \ldots, x_n))$$

where $m = 30$ and $x_i \in [0,1]$. The Pareto-optimal front is
formed with \( g(x) = 1 \). The introduction of the sine function in \( h \) causes discontinuities in the Pareto-optimal front.

6. RESULTS

Figure 1–3 show the union of non-dominated fronts achieved for the three test problems over five runs each. The Pareto front in each case is also plotted for reference.

As can be seen from the results, the algorithm performed well in all the tested cases. These results are comparable to those of other MOEAs tested using this benchmark and can be found in [14].

Figure 4 shows the Pareto front approximation achieved for ZDT3 test problem in one run. Even without explicit diversity management, archiving or population wide Pareto ranking for mate selection and reproduction, the algorithm is able to cover much of the Pareto front, even in a single run, in some cases.

7. CONCLUSIONS

An artificial life inspired evolutionary algorithm was presented for multiobjective optimization. Tests on several
multiobjective benchmark problems show promising results. As future work, the proposed algorithm will be tested on the remaining benchmarks problems. Also the algorithm's performance on dynamic multiobjective optimization problems will be tested. Further research is also required to investigate the performance of algorithm with other recombination and fitness evaluation mechanisms.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Intelligent Systems Center.

9. REFERENCES


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Habitat Width (cells) ((W))</td>
<td>128</td>
</tr>
<tr>
<td>Habitat Height (cells) ((H))</td>
<td>128</td>
</tr>
<tr>
<td><strong>Prey Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Initial number of prey ((N))</td>
<td>1000</td>
</tr>
<tr>
<td>Interaction radius (cells) ((R_{prey}))</td>
<td>5</td>
</tr>
<tr>
<td>Interaction area (cells) ((I_{area}))</td>
<td>81</td>
</tr>
<tr>
<td>Birth rate (%) ((P_b))</td>
<td>20</td>
</tr>
<tr>
<td>Death rate (%) ((P_d))</td>
<td>10</td>
</tr>
<tr>
<td>Genetic density ((G_d))</td>
<td>(N_p/(W \times H))</td>
</tr>
<tr>
<td>Mobility (cells) ((M_{prey}))</td>
<td>1</td>
</tr>
<tr>
<td>Mutation rate (%) ((P_m))</td>
<td>0.03</td>
</tr>
<tr>
<td>Crossover rate(%) ((P_c))</td>
<td>90</td>
</tr>
<tr>
<td>Maturity age (time steps) ((m))</td>
<td>4</td>
</tr>
<tr>
<td>Initial prey population age</td>
<td>(U(0, 3 \times P_m))</td>
</tr>
<tr>
<td>Initial prey interactions</td>
<td>(\text{Round}{ P_a \times G_d \times I_{area} } )</td>
</tr>
<tr>
<td><strong>Predator Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Initial predator population ((N_p))</td>
<td>6</td>
</tr>
<tr>
<td>Predation radius (cells) ((R))</td>
<td>32</td>
</tr>
<tr>
<td>Predation attempts ((N_x))</td>
<td>8</td>
</tr>
<tr>
<td>Predation success rate (%) ((P_s))</td>
<td>25</td>
</tr>
<tr>
<td>Mobility (cells) ((M_{pred}))</td>
<td>5</td>
</tr>
</tbody>
</table>