Lecture 11: Memetic Algorithms - II

Qiangfu Zhao and Yong Liu, “Memetic Algorithms,” Handbook on Computational Intelligence, Chapter 17, https://doi.org/10.1142/9789814675017_0017
Contents

• Definition of memetic algorithms
• Definition of memetic evolution
• Hybrids that are not memetic algorithms
• 1\textsuperscript{st} order memetic algorithms
• 2\textsuperscript{nd} order memetic algorithms
• Summary
Standard genetic algorithm

DefineType Agent {
    Genotype: g;
    Fitness : f;
};

Agent P[NumberOfAgent];
P=Initialize();
For t=1 to NumberOfGeneration
    For i=1 to NumberOfAgent
        p=Decoding(P[i].g);
        P[i].f=Evaluation(p);
    End
    if (Termination) break;
    P=Reproduction(P);
End

This the base algorithm!
Sequential Hybrid Algorithm (SHA)

P=Initialize();
For t=1 to NumberOfGeneration
    For i=1 to NumberOfAgent
        p=Decoding(P[i].g);
        P[i].f=Evaluation(p);
    End
    If (Termination) break;
    P=Reproduction(P);
End

x=FindBest(P);
p=Decoding(x.g);
p=LocalSearch(p);
Two-layered hybrid algorithm (TLHA)

\[ P = \text{Initialize}(); \]
\[ \text{For } t = 1 \text{ to NumberOfGeneration} \]
\[ \quad \text{For } i = 1 \text{ to NumberOfAgent} \]
\[ \quad \quad p = \text{Decoding}(P[i].g); \]
\[ \quad \quad p = \text{LocalSearch}(p); \]
\[ \quad \quad P[i].f = \text{Evaluation}(p); \]
\[ \text{End} \]
\[ \text{If (Termination) break;} \]
\[ P = \text{Reproduction}(P); \]
\[ \text{End} \]

Local search can be considered as one of the mutation operators.
Difference between SHA and TLHA

• There is an important difference between local searches in SHA and TLHA.
• In SHA, local search is usually *full*, that is, it should be conducted until convergence, but in TLHA, local search can be *partial*, that is, we may just run local search for several iterations without waiting for convergence.
• In fact, full local search for TLHA can be harmful because the evolution process may fall into local minimum easily.
• Using TLHA we may not have to employ complex algorithms for local search.
  – For example, to solve the TSP or other combinatorial problems, simple local searches like swapping and visiting neighboring nodes is sufficient.
SHA and TLHA are not memetic algorithms!

- In SHA or TLHA, the memeplex, that is, the local search strategy, is pre-defined or pre-designed.
- There is no process or mechanism for producing, selecting, varying, preserving good memeplexes in both algorithms.
- Therefore, they are NOT memetic algorithms, although they are hybrid algorithms.
- **We call these algorithms 0-th order MAs.**
Adaptive Two-layered Hybrid Algorithm (a-TLHA)

Agent P[NumberOfAgent];
Parameter par;

P=Initialize();
par=InitializeParameter();
For t=1 to NumberOfGeneration
   For i=1 to NumberOfAgent
      p=Decoding(P[i].g);
      pp=LocalSearch(p);
      P[i].f=Evaluation(pp);
      Gain=P[i].f-Evaluation(p);
      par=Adaptation(par,p,pp,Gain);
   End
   If (Termination) break;
   P=Reproduction(P);
End

Update the local search parameters to increase the gain!
Heuristics for parameter adaptation

• For many problems, we may just use simple neighborhood based search strategies, and the parameters can be adapted as follows:
  – If the fitness gain obtained through local search is high, that is, if the local search is successful, we may increase the step size, or increase the neighborhood radius, to accelerate the search process.
  – If the fitness gain is negative, that is, if the local search fails, we may decrease the step size, reduce the neighborhood radius, and increase the number of iterations, to search more carefully.
Generalizations

• We can generalize in 3 directions:
  – Each agent possesses a parameter set, so that good memes can be preserved.
  – The parameter set can specify not only the parameter values, but also the type of local search strategy.
  – The parameter set can specify a sequence of local search strategies, with different parameter values, provided that the computing resource consumption is allowed.
A Generalized adaptive two-layered hybrid algorithm (Ga-TLHA)

DefineType Agent {
  Genotype : g;
  Fitness : f;
  Parameter: par;
};

P=Initialize();
For t=1 to NumberOfGeneration
  For i=1 to NumberOfAgent
    p=Decoding(P[i].g);
    pp=LocalSearch(p,P[i].par);
    P[i].f=Evaluation(pp);
    Gain=P[i].f-Evaluation(p);
    P[i].par=Adaptation(P[i].par,p,pp,Gain);
  End
  If (Termination) break;
  P=Reproduction(P);
End

Each search agent has a parameter set, but the set does not evolve!
1st order memetic algorithms

• In general, the adaptive hybrid algorithms are 1st order MAs.
• They change the memes through adaptation (innovation) during the search process, but they do not “evolve” the memes.
• That is, memes are not produced through evolution in these algorithms.
DefineType Agent {
Genotype: [g1, g2];
Fitness : f;
}
Agent P[NumberOfAgent];

P=Initialize();
For t=1 to NumberOfGeneration
  For i=1 to NumberOfAgent
    p=PhenotypeDecoding(P[i].g1);
    par=MemeplexDecoding(P[i].g2);
    p=LocalSearch(p,par);
    P[i].f=Evaluation(p);
  End
  If (Termination) break;
  P=Reproduction(P);
End

There is only vertical memetic evolution!
Canonical evolutionary memetic algorithm (C-EMA)

P=Initialize();
For t=1 to NumberOfGeneration
  For i=1 to NumberOfAgent
    p=PhenotypeDecoding(P[i].g1);
    m=P[i].g2;
    If(IMITATION)
      j=MemeSelection(P);
      m=Imitation(P[j],m);
    If(VARIATION) m=Variation(m);
    par=MemeplexDecoding(m);
    p=LocalSearch(p,par);
    P[i].f=Evaluation(p);
  End
  If (Termination) break;
End
P=Reproduction(P);

Crossover of memes
Mutation of memes
Co-evolutionary memetic algorithm (Co-EMA)

```
(P, M) = Initialize();
For t = 1 to NumberOfGeneration
  For i = 1 to NumberOfAgent
    p = PhenotypeDecoding(P[i].g);
    (m1, m2) = MemeSelection(M);
    newMeme[i] = Variation(m1, m2);
    par = MemeplexDecoding(newMeme[i]);
    pp = LocalSearch(p, par);
    P[i].f = Evaluation(pp);
    Gain = P[i].f - Evaluation(p);
    newMeme[i].f = MemeEvaluation(Gain, p, pp);
  End
  If (Termination) break;
  (P, M) = Reproduction(P, M, newMeme);
End
```

DefineType Agent {
  Genotype: g;
  Fitness: f;
};
DefineType Spirit {
  Memotype: m;
  Fitness: f;
}
Agent P[NumberOfAgent];
Spirit M[NumberOfSpirit];
Spirit newMeme[NumberOfAgent];
General evolutionary memetic algorithm (G-EMA)

DefineType Agent {
    Genotype: \([g_1, g_2]\);
    Fitness : f;
};
DefineType Spirit {
    Memotype: m;
    Fitness : f;
}
Agent P[NumberOfAgent];
Spirit M[NumberOfSpirit];
Spirit newMeme[NumberOfAgent];
For $t=1$ to $\text{NumberOfGeneration}$
  For $i=1$ to $\text{NumberOfAgent}$
    $p = \text{PhenotypeDecoding}(P[i].g1)$;
    $m = \text{MemeSelection}(M)$;
    $\text{newMeme}[i] = \text{Variation}(P[i].g2, m)$;
    $\text{par} = \text{MemeplexDecoding}(\text{newMeme}[i])$;
    $\text{pp} = \text{LocalSearch}(p, \text{par})$;
    $P[i].f = \text{Evaluation}(\text{pp})$;
    $\text{Gain} = P[i].f - \text{Evaluation}(p)$;
    $\text{newMeme}[i].f = \text{MemeEvaluation}(\text{Gain}, p, \text{pp})$;
  End
  If (Termination) break;
  $(P, M) = \text{Reproduction}(P, M, \text{newMeme})$;
End
Summary - 1

• So far, memetic algorithms (MAs) have been considered as hybrid algorithms, including ANY hybrids of ANY existing algorithms.
• Here, MAs are re-defined as algorithms that possess three abilities:
  – Ability of producing the memes;
  – Ability of imitating the memes; and
  – Ability of preserving good memes.

In practice, full MAs may not be the best. If we have some a priori domain knowledge, we may find the best solution more efficiently and more effectively without trying to evolve the memes.
Summary - 2

- This lecture provided a more understandable classification of existing MAs algorithms (and their generalizations), using memetic evolution as a thread.
- Detailed discussions (e.g. detailed methods for local search, for encoding and decoding, etc.) are omitted because these parts can be found in textbook related to evolutionary computation.
- Also, we have focused on evolving good memeplexes for optimization or search. We may also study MAs for evolving or improving good human culture(s).
Summary - 3

• Memeplexes and Phenotype are two sides of the same thing:
  – Memeplexes are spirits for controlling the body, and the phenotype defines the body itself.
  – Phenotype is YIN, and Memeplexes are YANG.
  – Many spirits work together to form the mental brain, or the SELF of the body.
  – We human being is producing a more powerful media (internet) for the memeplexes to evolve, and we might be forgotten in a near future!