A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues

Katalio Krasnogor and Jim Smith

IEEE Tran. on Evolutionary Computation vol.9 no.5 2005

Three Kinds of Evolution Theory

- Lamarck evolution: If an organism changes during life in order to adapt to its environment, those changes are passed on to its offspring.
- Darwin evolution: The desires of animals have nothing to do with how they evolve, and that changes in an organism during its life do not affect the evolution of the species.
- Baldwin effect: It proposed that the ability of individuals to learn can guide the evolutionary process, facilitating evolution by smoothing the fitness landscape.

Evolution Algorithms and Local Search

- EAs are a class of search and optimization techniques based on Darwinian Evolution.
- Solutions are encoded as so-called chromosomes. Crossover, mutation, and selection are proceeded on chromosomes to obtain a better solution.
- Pure EAs are not well suited to fine tuning search in complex combinatorial spaces. Hybridization with other techniques can greatly improve the efficiency of search.

Memetic Algorithms

- The combination of EAs with local search (LS) was named memetic algorithms (MAs).
- The choice of name is inspired by Richard Dawkins' concept of a "meme", which represents a unit of cultural evolution that can exhibit local refinement.
- In the literature, MAs have also been named hybrid genetic algorithms (GAs), genetic local searches, Lamarkian GAs, and Baldwinian GAs.

Goals, Aims, and Methods

- This paper aims to begin the process of placing MA design on a sounder footing.
- The first goal is to define a syntactic model which enables a better understanding of the interplay between the different component parts of an MA.
- With such a model , we can construct a taxonomy of MAs, the second goal of this paper.

Design Issues for MAs

- Where and when should local search be applied within evolutionary cycle?
- Which individuals in the population should be improved by local search, and how should they be chosen?
- How much computational effort should be allocated to each local search?
- How can the genetic operators best be integrated with local search in order to achieve a synergistic effect?

TSP Problem definition

- Instance: A set C of m cities, and there are some paths between cities.
- Solution: a tour of C (a permutation from $[1 \ldots \mathrm{m}]$ to $[1 \ldots \mathrm{m}]$).
- Measure: the length of the tour.
- Aim: minimum length tour.

1997 Handbook of Evolutionary Computation
T.Back, D.Fogel, and Z.Michalewicz

```
Genetic_Local_Search(P ∈ S^l)
Begin
  /* λ, μ, m ≥ 1 */
  For i := 1 To μ Do
    Iterative_Improvement (s_i);
  Od
  stop_criterion:= false
  While (¬ stop_criterion) Do
    P' := ∅;
    For i := 1 To λ Do
      /* Mate */
      M_i ∈ P^m;
      /* Recombine */
      s_i ∈ H_m(M_i);
      Iterative_Improvement (s_i);
      P' := P' ∪ {s_i};
    Od
    /* Select */
    P :∈ (P ∪ P')^μ;
    evaluate stop_criterion
  Od
End.
```

Local Search

No Mutation

$(\mu+\lambda)$-selection

1999 Memetic Algorithms using guided local search: a cast study D.Corne, F.Glover, and M.Dorigo

1. The local search is used after every genetic operators.
2. Also (μ + λ)-selection.
3. Double function evaluation

```
GLS_Based_Memetic_Algorithm
Begin
  Initialize population;
  For i := 1 To sizeOf(population) Do
    individual := population_i;
    individual := Local − Search − Engine(individual);
    Evaluate(individual);
  Od
  Repeat Until (termination_condition)
Do
    For j := 1 To #recombinations Do.
      selectToMerge a set S_par ⊆ population;
      offspring = Recombine(S_par, x);
      offspring = Local − Search− Engine(offspring);
      Evaluate(offspring);
      Add offspring to population;
    Od
    For j := 1 To #mutations Do
      selectToMutate an individual in population;
      Mutate(individual);
      individual = Local − Search−; Engine(individual)
      Evaluate(individual);
      Add individual to population;
    Od
    population=SelectPop(population);
    If (population has converged) Then
      population=RestartPop(population);
    Fi
  Od
End.
```

1996 A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems
B.Freisleben and P.Merz

1. Good operator or good MA?
2. Local search between mutation and crossover

```
STSP-GA
Begin
  Initialize pop P with
Nearest − Neighbor(...);
  For i := 1 To popsize(P) Do
    Lin − Kernighan − Opt(individual_i), i ∈ P;
  od
  Repeat Until (converged) Do
    For i := 0 To #crossover Do
      Select two parents i_a, i_b ∈ P
      randomly;
      i_c = DPX − STSP(i_a, i_b);
      Lin-Kernighan-Opt (i_c);
      With probability m_p do
      Mutation-STSP (i_c);
      Replace an individual of P by i_c;
    od
  od
End.
```

Special crossover

Special mutation

QAP Problem definition

- Instance: A,B matrices of $n \times n$
- Solution: A permutation π on n
- Measure: The cost of permutation $C(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{\pi(i),\pi(j)}$
- Aim: Munimum cost permutation

1993 Genetic hybrids for the quadratic assignment problem
C.Fleurent and J.Ferland

1. H1 and H2 both are Tabu search
2. Cull is a $(\mu+\lambda)$-selection.
3. No mutation

```
Genetic_Hybrid_Algorithm(H1, H2)
Begin
  P := ∅;
  For i := 1 To m Do
    generate a random permutation p;
    Add H1(p) to P;
  Od
  Sort P;
  For i := 1 To number_of_generations Do
    For j := 1 To num_offspring_per_
    generation Do
      select two parents p1, p2 from P;
      child := crossover(p1, p22);
      Add H2(child) to P;
    Od
    Sort P;
    Cull(P, num_offspring_per_generation);
  Od
  Return the best p ∈ P;
End.
```

1999 A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment

C.Fleurent and J.Ferland

```
QAP_MA
Begin
  Initialize population P;
  For i:=1 To sizeof(P) Do
    individual := P_i;
    individual:=Local_Search(individual);
  Od
  Repeat Until (terminate=True) Do
    For i:=1 To #recombinations Do
      Select two parents i_a,i_b ∈ P
      randomly;
      i_c := Recombine(i_a,i_b);
      i_c := Local_Search(i_c);
      Add individual i_c to P;
    Od
    P := Select(P);
    If (Pconverged) Then
      For i:=1 To sizeof(P), i ≠ index(Best)
Do
        individual := P_i;
        individual:=Local_Search(Mutate
        (individual));
      Od
    Fi
  Od
End.
```

1. Use mutation to prevent convergence
2. $(\mu+\lambda)$-selection

MGC Problem definition

- Instance: Graph $G = (V, E)$
- Solution: A coloring of G
- Measure: cardinality k of coloring
- Aim: Munimum k coloring

1998 A new genetic local search algorithm for graph coloring C.Fleurent and J.Ferland

```
GL_for_Coloring
Begin
  /* f, F*: fitness function and */
  /* best value encountered so far */
  /* s*: best individual encountered so
  far */
  /* best(P): returns the best
  individual */
  /* of the population P */
  i = 0;
  generate(P_0);
  s* := best(P_0);
  f* := f(s*);
  While (f* > 0 and i < maxIter) Do
    P'_i := crossing(P_i, T_x);
    /* using specialised crossover */

    P_{i+1} = mutation(P'_i);
    /* using Tabu search */
    If (f(best(P_{i+1})) < f*) Then
      s* := best(P_{i+1});
      f* := f(s*);
    Fi
    i := i + 1;
  Od
End.
```

1.Mutation is replaced by Tabu search
2.For some large instance, have the best results
3.No survivor selection

PSP Problem definition

- Instance: A simplified protein sequence of length l
  i.e., a string $s \in \{G, P\}^l$
- Solution: A self-avoiding path p which embeds s into a two or three dimensional lattice.
- Measure: Potential energy
  $E(p) = -\sum_{i=1}^{l} \sum_{j=i+1}^{l} D_{ij} | (D_{ij} = 1) \wedge (s_i = s_j = H)$
- Minimum energy solution

2000 A memetic algorithm with self-adaptive local search: TSP as a case study
K.Krasnogor and J.Smith

```
PF_MA
Begin
  Random initialize population Parents;
  Repeat Until (Finalization_cri-
teria_met) Do
    Local_Search(Parents);
    mating_pool=Select_mating(Parents);
    offsprings:=Cross(mating_pool);
    Mutate(offsprings)
    Parents:=Select(Parents+offsprings)
  Od
End.
```

1.$(\mu+\lambda)$-selection

Syntactic Model

- Simple EAs
- Coordinating local search with crossover and mutation
- Coordinating local search with population management
- Incorporation historical information

Taxonomy

$D(A) = b_{mS} b_{cS} b_R b_M$

b_{mS}: provide history data to its local search

b_{cS}: provide population statics to its local search

b_R: corporate with recombination

b_M: corporate with mutation

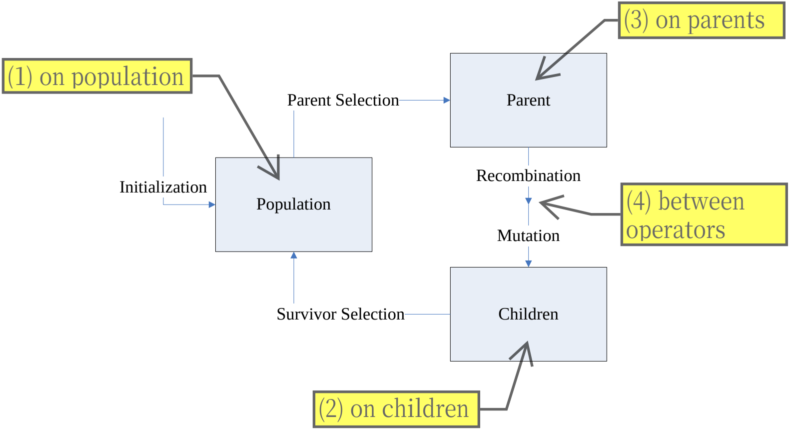
TABLE I
CLASSIFICATION OF ALGORITHMS DISCUSSED IN SECTION III ACCORDING TO PROBLEM AND D

9 - 15							
8							[21]
7						[76]	[82]
6		[16]		[47]			
5							[83]
4	[57]				[57], [59]	[26], [74], [84], [85]	
3	[13], [38], [44]					[86]	[87], [88]
2	[37], [42], [43], [45], [46]	[9]	[50], [51]				
1			[49]				[89], [90]
0							
D	TSP	QAP	MGC	BPQ	*PFP* and *Protein Docking*	General Studies	Other Applications

Choice of Local Search Operators

- The answer is "it depends". Even within a single problem class (like TSP), the choice of LS operators was instance-dependent.
- It is trivially true that a point which is locally optimal with respect to one operator may not be with respect to another. Multiple local search operators may be work (variable neighborhood search).
- Using multiple operator may be inefficiency and adapting the parameters of LS is useful. But adaptation of parameters produces noise in a first-ascent approach.

Integration Into EA Cycles



Managing the Global-Local Search Tradeoff

- Most MAs apply local search to every individual in every generation of the EA.
- Hart(1994) and Land(1998) suggest various mechanisms to choose individuals to be optimized, the intensity of local search and the probability of performing LS.
- Land(1998) proposes the use of fine-grain schedulers that sample the basin of attraction. Only those solutions that are in promising basin of attraction will be assigned later.

Conclusion

- The syntactical model obtained allowed for the definition of an index number D.
- When plotting the D index for a number of papers, we were able to identify classes of MAs that had received a lot of attention and other classes that were little explored.