Efficient clustering with a self-adaptive genetic algorithm

Juha Kivijärvi  
Turku Centre for Computer Science  
University of Turku  
FIN-20014 Turku, Finland  
Email: juhkkivij@utu.fi  
Tel: +358 2 333 8797

Pasi Fränti  
Department of Computer Science  
University of Joensuu  
Finland  
PB 111  
Email: franti@cs.joensuu.fi  
Tel: +358 13 251 3103

Olli Nevalainen  
Turku Centre for Computer Science  
University of Turku  
FIN-20014 Turku, Finland  
Email: olli.nevalainen@utu.fi  
Tel: +358 2 333 8631

We give a self-adaptive genetic algorithm (SAGA) for the clustering problem (Kaufman and Rousseeuw, 1990). We assume that the number of clusters is fixed and the objects to be clustered are k-dimensional vectors in an Euclidean space.

Genetic algorithms (GAs) have turned out to be very effective in solving the clustering problem (Fränti et al., 1997). However, they have many parameters, the optimal selection of which depends on the problem instance. Furthermore, their optimal values may change during the execution of the algorithm.

Our aim is to give an effective and straightforward algorithm which obtains good solutions for the problem without explicit parameter tuning. The self-adaptation takes place at individual level (Hinterding et al., 1997), which means that each individual of the GA population contains a set of parameters. These are used in the reproduction process. The parameters we adapt are the crossover method γ, mutation probability ψ and noise range ν. Thus, an individual i is of the form i = (ωi, γi, ψi, νi) where ωi is a solution to the clustering problem.

The general structure of SAGA is following:

1. Generate S random individuals to form the initial generation.
2. Iterate the following T times.
   (a) Select SB surviving individuals for the new generation.
   (b) Select S − SB pairs of individuals as the set of parents.
   (c) For each pair of parents (iα, iβ) do:
      i. Determine the strategy parameter values (γα, ψα, να) for the offspring iα by crossing the strategy parameters of the two parents.
      ii. Mutate the strategy parameter values of iα with the probability Ψ.
      iii. Create a new solution ωα by crossing the solutions of the parents. The crossing method is determined by γα.
      iv. Mutate the solution of the offspring with the probability ψα.
      v. Add noise to ωα. The maximal noise is να.
      vi. Apply k-means iterations to ωα.
      vii. Add iα to the new generation.
3. Output the best solution of the final generation.

Our experiments show that the results of SAGA are fully comparable with those of a carefully refined non-adaptive genetic algorithm. The major benefit of SAGA is that the refining phase can be omitted completely. The algorithm finds the best or at least a satisfying parameter combination and is able to obtain a very good solution practically always. A drawback of the method is its rather large running time.

References

