A Global Convergence Proof for a Class of Genetic Algorithms

Richard F. Hartl*

Abstract:

In this paper a variant of a genetic algorithm is investigated which combines the advantages of genetic algorithms and of simulated annealing. In particular, under rather mild assumptions a global convergence result is derived.

Key Words:

Genetic algorithms, Simulated annealing, Markov chains, Global convergence, Probabilistic programming, Simulation, Randomized optimization.

I. INTRODUCTION

For many NP-had combinatorial problems and for problems with "crazy" objective function (i.e., discontinuous, nonsmooth or having many local optima) stochastic search algorithms have been successfully applied and are particularly appreciated by practitioners. Beside simple Monte-Carlo methods there are two important stochastic approaches to solve deterministic problems namely "genetic (adaptive)algorithms" and "simulated annealing". Although the basic ideas in both algorithms are very similar, there is not much literature about the connection of the two approaches and about comparisons of efficiency. It is the purpose of this paper to present a modified genetic algorithm (GA, hereafter) which allows for all the flexibility of general genetic algorithms (population consisting of more than one individual, recombination, etc.). On the other hand, it has some similarities to simulated annealing which makes it possible to prove that under much less restrictive assumptions than required for simulated annealing the algorithm converges with probability one to the global optimal solution.

II. THE ALGORITHM

Let us consider the following mathematical model

$$F(x) \rightarrow \min \tag{1}$$

$$x \in M \tag{2}$$

where M is a finite set and F: M $\rightarrow$ E is the objective function. The assumption of a finite number of feasible solutions may seem somewhat restrictive. However, the methods can also be applied to more general problems while the finiteness assumption is required for the global convergence proof. On the other hand, if the set of feasible solutions is bounded and if the problem is discretized on a digital computer then the resulting feasible set is in fact finite.

Prof. Dipl.-Ing. Dr.techn. Richard F. Hartl
Chair of Production and Operations Management
Institute of Management at the University of Vienna
Bruennerstraße 72
A-1210 Vienna. Austria
Richard.Hartl@univie.ac.at

In order to describe the our genetic algorithm , we need the following definitions:

A neighborhood structure is a mapping N from M into 2 , i.e. for each solution x it defines a neighborhood N(x) of x and each $y \in N(x)$ is called a neighbor of x. In the case of the traveling salesman problem, the neighborhood of a given tour x can e.g., be defined as the set of tours which can be generated by the Lin (1965) 2-change or 3-change heuristic.

A generation mechanism is a rule of selecting a solution y from the neighborhood N(x) of a given solution x. In the context of genetic algorithms such a generation rule is usually called a mutation rule. The generation mechanism can be described by a transition probability matrix R such that

$$R(x,y) = P\{X_{t+1} = y | X_t = x\} \qquad (3)$$

where X_t denotes the state of the system at time (iteration) t. Clearly R(x, y) > 0 if and only if $y \in N(x)$. By (3) a Markov chain is defined over the set M of feasible solutions. However, in order to "solve" problem (1), (2) this Markov chain has to be modified by some acceptance criterion so that "good" solutions are selected more often or with higher probability than "bad" ones.

A local optimal solution is an $x \in M$ such that

$$F(x) \le F(y) \text{ for all } y \in N(x)$$

while a global optimal solution is defined by

$$F(x) \le F(y) \text{ for all } y \in M.$$

In order for the algorithm not to get stuck in a local optimum (which is not globally optimal) it is necessary to accept also deteriorations of the objective function with positive probability.

A state y is reachable from state x if there exists $z_1, ..., z_m \in M$ such that $z_1 \in N(x)$, $z_2 \in N(z_1)$, ..., $y \in N(z_m)$.

III. GENETIC ALGORITHMS

Genetic algorithms (GA, hereafter) imitate the evolution of nature in order to optimize a system. In each iteration, not a single current solution is considered but there is a population of p solutions which can be interpreted as the "parents". Let

$$X_t = \{x_t^1, ..., x_t^p\},$$

with $x_t^i \in E^n$, denote the parent population in iteration t. From this a number c of new solutions is generated which can be interpreted as "children". Let

$$\bar{X}_t = \{\bar{x}_t^1, ..., \bar{x}_t^c\},$$

denote the children population of iteration t.

There are two possibilities to generate children: Mutation and Recombination

In the context of GA the generation mechanism is usually called mutation rule. These c children are generated by mutation of arbitrarily selected parents or each parent generates a fixed number of individuals. In the latter case c is a multiple of p.

One of the advantages of GAs is the possibility of combining the information contained in two parent solutions in creating a new solution. In allusion to the evolution of nature this is usually called a recombination. In the simplest case, the recombination can be a "crossing over" of the genetic information, i.e. some (arbitrarily selected) coordinates of the newly created solution are taken from one parent and the rest is taken from the other parent. Several authors have reported significant improvements of the convergence of GAs when recombinations are introduced (see, e.g., Rechenberg 1973, or Ablay 1987).

Important references for GAs are, e.g., Fogel et al. (1966), Rechenberg (1973), Holland (1975), Schwefel (1977), and Goldberg (1988).

After generating the children population $\overline{X}_t$ from the parent population X_t by appropriate choice of the mutations and (possibly) recombinations the new parent population X for the next step has to be selected. This is usually done by one of the following selection rules.

(p+c)-rule: In this case the best p individuals out of all p+c parents and children $x_t^1,...,x_t^p,\overline{x}_t^1,...,\overline{x}_t^c$ are selected to form the new parent population $x_{t+1}^1,...,x_{t+1}^p$.

(p,c)-rule: In this variant the best p individuals out of the c children $\overline{x}_t^1,...,\overline{x}_t^c$ are selected, Clearly c > p in this case.

The difference is that according to the (p+c)-rule good individuals can live infinitely long while according to the (p,c)-rule each individual has a maximum life time of one generation. Since the (p,c)-rule implies that good solutions can "get lost", the convergence to a local optimum is usually a bit slower. However, to some extent it prevents the concentration of the population in a small area which improves the global convergence behavior in the case of multiple local optima.

Some authors, have reported considerable improvements of the convergence rate through destabilization phases; see, e.g., Ablay (1987) in the case of the traveling salesman problem.

IV. A COMBINATION OF SA AND GA

Let us consider a GA in the sense of the previous section. The algorithm starts from an initial population $X_0 = \left\{x_0^1,...,x_0^p\right\}$. In each iteration t, from the current parent population X_t a population $\overline{X}_t$ of c children is generated by mutations and/or recombinations, where at least one child is generated by a mutation. Then the following selection rule is chosen:

(a) Select x as the best of all the p+c individuals $x_t^1,...,x_t^p,\overline{x}_t^1,...\overline{x}_t^c$.

(b) Select x_{t+1}^2 arbitrarily among all the children $\overline{x}_t^1,...,\overline{x}_t^c$ hich have not already been selected in step (a) and ?

(c) Select $x_{t+1}^3,...,x_{t+1}^p$ if p > 2 by any selection rule.

In step (c) of rule (12) the following alternatives can be thought of.

(c1) Select those children (not already selected) with the best values of the objective function.

(c2) Select those individuals (parents or children not already selected) with the best value of the objective function.

(c3) Select p-2 individuals arbitrary among the children (not already selected).

(c4) Select p-2 individuals arbitrary among the parents and children (not already selected).

In the above selection rules, the statement "not already selected" prevents the population from getting concentrated in a small area by having many copies of the same individual. For the following convergence result this statement can also be omitted.

By selecting one solution arbitrary in step (b) one has the same effect of circumventing local optima as in the SA-rule (5). Furthermore, in the spirit of SA another possibility of part (c) of rule (12) can be formulated:

(c5) Select 100r % of the p-2 individuals arbitrarily and the remaining 100(1-r)% according to their objective function value. Choose r = 1 initially and let $r \to 0$ as $t \to \infty$. Since (p-2)r < 0.5 for t large enough, this implies that after some initial phase this rule works like (c1) or (c2).

In (c5), the parameter r plays a similar role as the "temperature" T in the SA-algorithm. It is, however, also possible to choose the following variant:

(c6) Choose the solutions according to rule (c5) with $0 < r < 1$ fixed.

We can now formulate the following global convergence result:

Theorem 4.1. *Let the set of feasible solutions M be finite and assume, that for all x,* $y \in M$ *the state y is reachable from x by the mutations considered. Then the GA with selection rule (12) in any of the variants (c1) - (c6) has the following property*

$$\lim P\{\text{At least one solution in } X_t \text{ is globally optimal}\} = 1. \qquad (13)$$

It should be noted that, compared to Theorem 2.1 much less restrictive assumptions have to be made. Essentially the only assumption is that the Markovian chain described by the mutation/generation-rule is irreducible. The proof of this result can be found in the next section.

V. PROOF OF THEOREM .1

Define a new Markov-chain Y_t as follows. The state $y \in M^P$ is given by the combined vector of p elements of the original state space M. However, for technical purposes we make the following exception for state y_0 : the system is in state y_0 if at least one globally optimal solution x is in the current population.

Now the sequence of populations computed by the modified GA of Section IV can be described by a Markov-chain in the above framework. The state Y_t in iteration t is the combined vector of all individuals in population X_t . On the other hand $Y_t = y_0$ if one of the elements $x^1,\ldots,x^p$ of X_t is a globally optimal solution.

According to part (a) of the selection rule in Section IV the state y_0 is absorbing. This is because once a globally optimal solution is found, it will always remain in the population. Furthermore, because of the assumptions of Theorem 4.1 each globally optimal solution is reachable from any other solution . According to part (b) of the selection rule this implies that (absorbing) state y_0 of the new Markov chain is reachable from any other state. This implies that every state $y \neq y_0$ is transient and

$$\lim_{t\to\infty} P\{Y_t = y_0\} = 1.$$

Thus (13) in Theorem 4.1 is proved. From the classical literature of Markov processes (e.g., Iosifescu, 1980) one can also obtain expressions for the time to absorption. We mention the following:

Let the transition matrix of the new Markov process $\{Y_t\}$ be given by $\tilde{P} = \begin{bmatrix} 1 & 0 \\ R & T \end{bmatrix}$, where R is a $k \times 1$ matrix, T is a $k \times k$ matrix, 0 is a $1 \times k$-zero matrix, and k + 1 is the dimension of the state space of $\{Y_t\}$. Then, with P(n) denoting the vector of probability that the time to absorption in state zero starting in the transient states is n, we get

$$P(n) = T^{n-1}(I - T)e \tag{14}$$

where e is the k-dimensional unity vector. Hence, the average time to absorption is

$$\pi\left[\sum_{n=0}^{\infty} nT^{n-1}\right](I - T)e = \pi\left[\sum_{n=0}^{\infty} T^n\right]e$$

where π is the initial distribution of Y. However, since k is extremely large in practical problems and therefore T cannot be computed any more, this formula does not seem to be of much practical use. For the same reason, we refrain from applying some further sophisticated results from the theory of absorbing Markov chains.

VI. CONCLUDING REMARKS

In this paper we have analyzed a genetic algorithm according to which in each generation not only the best individuals survive but also some randomly selected other ones regardless of their "fitness". This is in accordance with nature where in the long run the fittest individuals survive but in the short run also other individuals can survive. Furthermore, it circumvents the problem that the population concentrates in a narrow neighborhood of a local optimum.

As a consequence, it was possible to derive a global convergence result in the spirit of the simulated annealing result by Hajek (1988) under much weaker assumptions. An advantage of the algorithm considered is that it is robust, in the sense that it does not require much expertise to apply it whereas in the simulated annealing case the choice of the annealing rule is always a nontrivial problem.

Finally, we would like to mention that the above GA has successfully been applied to some classical textbook examples and to the unit commitment problem of a local power company; see Hartl and Petritsch (1990).

We should note that the above GA is more in the spirit of the German "Evolutionsstrategien" by Rechenberg (1973) and Schwefel (1977) than of the GAs by e.g. Goldberg (1989).

REFERENCES

Aarts, E., Korst, J. (1989) Simulated annealing and Boltzman machines: a stochastic approach to combinatorial optimization and neural computing. Chichester: John Wiley & Sons.

Ablay, P. (1987) Optimieren mit Evolutionsstrategien, Spektrum der Wissenschaften 7, 104-115.

Burkard, R.E., Rendl, F. (1984) A thermodynamically motivated simulation procedure for combinatorial optimization, European Journal of Operational Research 17, 169-174.

Cerny, V. (1985) Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, Journal of Optimization Theory and Applications 45, 41-45.

Fogel, L.J., Owens, A.J., Walsh, M.J. (1966) Artificial intelligence through simulated evolution, New York: Wiley.

Goldberg, D.E. (1989) Genetic algorithms in search, optimization, and ma- chine learning. MA: Addison-Wesley.

Goldberg, D.E. (1990) A note on Boltzman tournament selection for genetic algorithms and population oriented Simulated annealing, TCGA Report 90003 (May 1990), Univ. of Alabama.

Hajek, B. (1988) Cooling schedules for optimal annealing, Mathematics of Operations Research 13, 314-329.

Hartl, R.F., Petritsch, G. (1990) Kraftwerkseinsatzplanung mittels zufallsgesteuerter Suchstrategien, in: Operations Research Proceeding 1989, K.P. Kistner et al. (Eds.). Berlin: Springer, 95-102.

Holland, J.H. (1975) Adaptation in natural and artificial systems. Ann Arbor: Univ. of Michigan Press.

Iosifescu, M. (1980) Finite Markov Processes and their Applications, Chichester: Wiley.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983) Optimization by simulated annealing, Science 220, 671-680.

Lin, S. (1965) Computer solutions of the traveling salesman problem, Bell System Technical Journal 44, 2245-2269.

Rechenberg, I. (1973) Evolutionsstrategien: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Stuttgart: Frommann-Holzboog.

Schwefel, H.-P. (1977) Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhauser.