Theoretical foundations of order-based genetic algorithms

Jakub Wróblewski
Institute of Mathematics
Warsaw University
Warsaw, Poland
jakubw@jakubw.pl

Abstract

This is a shorter version of an article presented in:

Fundamenta Informaticae, vol. 28 (3, 4), pp: 423-430. IOS Press, 1996.

A lot of research on genetic algorithms theory is concentrated on classical, binary case. However, there are many other types of useful genetic algorithms (GA), e.g. tree-based (genetic programming), or order-based ones. This paper shows, that many of classical results can be transferred into the order-based GAs. The analysis includes the Schema Theorem and Markov chain modelling of order-based GA.

1 Introduction

Some combinatorical problems can be solved using genetic algorithm (GA) with order (permutational) representation. The classical example of Travelling Salesman Problem [2] is the best known one. This type of genetic algorithms is also widely used as the driving force of "hybrid algorithms", a powerful tool for solving problems such as: graph colouring problem [1], short reduct finding [10] or template finding in databases [4].

The order-based genetic algorithms are rich in applications. However, there are still no theoretical analysis of some kinds of order-based GA. Next sections are the trial of translation of classical theoretical results to the order-based case.

2 Order-based genetic algorithm

In the **order-based genetic algorithms** the chromosome is a n-element permutation σ, represented by the sequence of numbers: $\sigma(1)\ \sigma(2)\ \sigma(3)\ \ldots\ \sigma(\mathrm{n})$.

Mutation of order-based individual means one random transposition of its genes. There are various methods of recombination considered in literature. In [2] such methods as PMX (Partially Matched Crossover), CX (Cycle Crossover) and OX (Order Crossover) are described. But we will use another type of crossing-over operator for further analyse: **MOX (Modified Order Crossover)**.

This recombination operator affects two parent chromosomes and replaces them by two children. First, we choose one gene in first parent's chromosome at random. This gene will be the end of a matching section, starting at the beginning of chromosome. Identical matching section is marked on second parent's chromosome. Then, we leave the matching sections unchanged, but the rest of genes of the first chromosome is set in the order of appearance in the second chromosome. We perform the same operation on the second parent. For example:

$$\begin{array}{ccccccccccccccc} 1 & 2 & 3 & | & 4 & 5 & 6 & \xrightarrow{MOX} & 1 & 2 & 3 & | & 4 & 6 & 5 \\ 4 & 2 & 1 & | & 3 & 6 & 5 & & 4 & 2 & 1 & | & 3 & 5 & 6 \end{array}$$

Vertical line indicates the end of matching section.

3 Order-based Schema Theorem

Suppose we have the hybrid algorithm similar to these described in [1], [4], [10] (see section 5). The deterministic, heuristic algorithm is used to finding the solutions, but its efficiency depends on the order of considering the elements of the search space. The genetic algorithm is used to generate the optimal order.

The notion of schema should refer to this internal structure of algorithm. None of the order-based schemata considered in [2] is fitted enough to our situation. The optimal schema notion should refer only to relative position of genes in a chromosome, with no respect to the defining length of schema nor to the relative distances between values. We will consider another type of order-based schema: **relative dispersed (rd)**, similar to the one described in [1].

Definition 3.1 *Relative dispersed schema* $rd^n(\ a_1\ \ a_2\ \ldots\ a_k\ )$ *expands to the set of all individuals of length n with genes* $a_1\ \ldots\ a_k$ *located in this order (not necessarily sequentially). The* **order** $o(\cdot)$ *of an rd-schema is defined to be equal to the number k.*

In [7] N. J. Radcliffe introduced seven design principles - characteristics of various schema notions and recombination operators. The MOX operator and the notion of rd-schema obey the principle of respect (if both of the parents match the schema, children will match the schema as well). Moreover, the MOX operator is in some sense strictly transmitting: if a value x precedes a value y in the child chromosome, the same order we observe in at least one of its parents. This is the theoretical reason of usage MOX operator in the discussion - the practical one will be presented in section 5.

To formulate a theorem similar to the classical Schema Theorem, we should calculate probability of disruption of schemata by the genetic operators. Let $l = o(S)$ be an order of given schema S. Suppose that the first parent matches the schema S. There are two situations in which at least one of children will match schema S:

a) all values from S are in matching section of first parent;

b) all values from S are beyond matching section of second parent.

These two events are independent. Let $1 - P_1$ be the probability of a), let $1 - P_2$ be the probability of b). Let Pc be the probability of disruption of schema S by the MOX operation, let k be the size of matching section. We have:

$$P_1 = 1 - \frac{\binom{k}{l}}{\binom{n}{l}} = \frac{\binom{n}{l} - \binom{k}{l}}{\binom{n}{l}}$$

$$P_2 = 1 - \frac{\binom{n-k}{l}}{\binom{n}{l}} = \frac{\binom{n}{l} - \binom{n-k}{l}}{\binom{n}{l}}$$

$$Pc \leq \frac{1}{n+1}\sum_{k=0}^{n} P_1 P_2 = \frac{1}{n+1}\sum_{k=0}^{n} \frac{\binom{n}{l} - \binom{k}{l}}{\binom{n}{l}} \cdot \frac{\binom{n}{l} - \binom{n-k}{l}}{\binom{n}{l}}$$

There are exactly two situations in which schema S will be disrupted by mutation:

a) both positions chosen during mutation belongs to S;

b) one position is in S, second is placed between the genes from S.

Probability of a) is equal to:

$$P_a = \frac{l}{n} \cdot \frac{l-1}{n-1}$$

Probability of b) is equal to:

$$P_b = \left(\frac{l}{n} \cdot \frac{n-l}{n-1} + \frac{n-l}{n} \cdot \frac{l}{n-1}\right) P'$$

where P' denotes the probability, that between two randomly chosen positions in chromosome is at least one value from S:

$$P' = \sum_{k=1}^{n-2} \left(\frac{n-(k+1)}{\left(\frac{n\cdot(n-1)}{2}\right)} \right) \cdot \left(1 - \frac{\binom{n-l-1}{k}}{\binom{n-2}{k}} \right)$$

Now, we can calculate the probability Pm of disruption of schema S by the mutation:

$$Pm = P_a + P_b = \frac{l}{n\cdot(n-1)} \left(l - 1 + \frac{2\cdot(n-l)}{n\cdot(n-1)} \sum_{k=1}^{n-2} 2\,(n-k-1)\, \frac{\binom{n-2}{k} - \binom{n-l-1}{k}}{\binom{n-2}{k}} \right)$$

Let p_{cross} be the probability of crossing-over, let p_{mut} be the probability of mutation of individual. Let N_t be the number of individuals matching S in the population after t steps of evolution, let f be an average value of fitness function in population, let $\overline{f}$ be an average value of fitness function of individuals matching S. The **Schema Theorem** can be formulated in the way similar to the classical case:

$$E\left(N_{t+1}\right) \geq N_t \frac{f}{\overline{f}} \left(1 - p_{cross} \cdot Pc - p_{mut} \cdot Pm\right)$$

In fact, we should take into account that if both parents match the schema, there is no chance of disruption, so the theorem can be reformulated as follows:

$$E\left(N_{t+1}\right) \geq N_t \frac{f}{\overline{f}} \left(1 - p_{cross} \cdot Pc \cdot \left(1 - \frac{N_t}{M}\right) - p_{mut} \cdot Pm\right)$$

The results are nearly identical with the classical case. Therefore we may use classical results suchlike the building block hypothesis, the analogy with k-armed bandit, theory of deceptiveness.

4 Convergence theorems

In this section we will show, that the results described in [5] and [9] can be generalised to the order-based genetic algorithms.

In [5] authors model a simple genetic algorithm using a Markov chain with N states - each of it corresponds to one possible population. In case of order-based GA, number N can be calculated by:

$$N = \binom{M + n! - 1}{n! - 1}$$

where M is the number of individuals in population, n - size of chromosome (permutation). The size of transition matrix Q of the chain is $N \times N$. There are exact values of elements of Q calculated in [5] for classical case. We do not need to calculate corresponding values for our case, because the size of transition matrix is too large for direct analysing. But we need to prove that the chain is ergodic.

Lemma 4.1 *The Markov chain modelling the behaviour of order-based GA is regular.*

Proof 1 *We need to prove than for all i, j: $Q^m_{i,j} > 0$, i.e. we can transit between any two populations with positive probability. Suppose the probability of mutation and crossover to be greater than 0 and less than 1, and suppose the fitness function to be positive. Note that since fitness function is positive, the probability, that the "roulette wheel" algorithm causes no change to a population, is greater than 0. On the other hand, we can transit from any individual to any other by no more than $n-1$ mutations (transpositions). Therefore, for any i, j we have $Q^{n-1}_{i,j} > 0$.*

Since the Markov chain is ergodic, there exist a steady state distribution:

$$\pi Q = \pi$$

In general, the steady state distribution gives positive probability to all populations - not only to these containing the optimal individual. There are two methods assuring that the order-based GA will spend asymptotically all time in population containing the individual with maximum fitness value. The first method is presented in [5] for classical case and bases on the asymptotic behaviour of GA with population size going to infinity. These results can be used for order-based case with small changes. The second method, described in [9] for classical case, bases on finite population size and assumes some changes in selection algorithm.

Let us introduce the modified elitist strategy to our model: let the best (by means of fitness function) individual be copied with no change after each step of evolution. The population consists of $M+1$ individuals now. Let Q be the transition matrix of Markov chain modelling such an algorithm.

For strictness, let us introduce the priority function on individuals: let the best (globally) individual has priority value of 1, the next - 2 etc., the worst individual - $n!$. We will consider the populations in order of the priority of the best individual. The shape of Q can be depicted as follows:

$$Q = \begin{bmatrix} Q(1) & 0 & \cdots & 0 \\ + & Q(2) & & 0 \\ \vdots & & \ddots & \vdots \\ + & + & \cdots & Q(n!) \end{bmatrix}$$

where $Q(s)$ is the transition submatrix of size $N(s) \times N(s)$, corresponding to the populations with the individual of priority s as the best individual. The size$N(s)$ can be calculated as:

$$N(s) = \left(\begin{array}{c} M + n! - s \\ n! - s \end{array} \right)$$

Now we can observe (as in case of classical GA, [9]) that Q represents the Markov chain with absorbing class $Q(1)$. Moreover, this is easy to see, that $Q(1)$ is regular (we can omit $(M+1)$-th individual and repeat the proof presented above). So, the steady state distribution on Q exists and gives positive probabilities only on $Q(1)$ i.e. on populations containing globally optimal individual.

Now we will approximate the algorithm's convergence rate. Let $d(i,j)$ be a distance measure on individuals (analogous to Hamming distance) defined as the minimal number of mutations (transpositions) transforming i-th individual into j-th one. Let r be the maximal distance in any population. Since every permutation is a combination of at most $n-1$ transpositions, we have $r = n-1$. Now, let $Q' = Q^r$ be a transition matrix of Markov chain describing the behaviour of our algorithm every r steps. We will continue to analyse Q' instead of Q. It is easy to see, that Q' has the similar form than Q, with submatrices $Q'(s)$ on its diagonal.

Let $pr(i)$ be a priority value for individual i, or a priority value of the best individual in case i is a population. In [9] the following fact was proved:

$$\sum_{pr(k)=1} q_k^{(n)} \geq 1 - C\,|\lambda^*|^n$$

where $q^{(n)}$ is the probability distribution on populations after n steps, C is a constant, and:

$$\lambda^* = \max_{s,t}(\lambda_{s,t}), \qquad s = 2, \ldots n!, \quad t = 1, \ldots N(s)$$

where $\lambda_{s,t}$ is a t-th eigenvalue of $Q(s)$.

We can write the same for Q'. Now we will estimate λ^*, which determines convergence rate of algorithm.

Lemma 4.2 *The value of $|\lambda^*|$ can be estimated as follows:*

$$|\lambda^*| \leq 1 - A \cdot n^{-2\delta} p_{mut}^{\delta} (1 - p_{mut})^{r-\delta}$$

where A is a constant (irrespective of the mutation probability), and:

$$\delta = \max_{i \in \{1 \ldots n!\},\, pr(i)>1} (\min_{j \in \{1 \ldots n!\},\, pr(j)<pr(i)} d(i,j))$$

Proof 2 *As in [9] (appendix C), we have:*

$$|\lambda^*| \leq \max_{i=1\ldots N(s^*)} \sum_{j=1}^{N(s^*)} Q'(s^*)_{i,j}$$

The sum on the right hand side of the inequality is equal to the probability, that the population still has the best individual with priority s^* after one step of Q' (i.e. after r steps of Q). This probability can be estimated as follows:

$$|\lambda^*| \leq \max_{i=1\ldots N(s^*)} (1 - A_i B_i)$$

where A_i is a probability, that the best individual will not be affected by crossover and reproduction during r steps of Q, and B_i is a probability, that this individual will be affected by proper number of proper mutations (so that it will be transformed into the individual with higher fitness value).

The value of A_i can be estimated irrespective of the probability of mutation and i value:

$$A_i \geq A = (1 - p_{cross})^r \left(\frac{f_{\min}}{M \cdot f_{\max}} \right)^r$$

where f_{min} and f_{max} means the minimal and maximal fitness values (positive by assumption).

Suppose we know the minimal number of mutations transforming the best individual in the i-th population into the better one. Let δ_i be such a number. Since there are n^2 possible transpositions, the value of B_i can be estimated as follows:

$$B_i \geq (n^{-2})^{\delta_i} \cdot p_{mut}^{\delta_i} (1 - p_{mut})^{r - \delta_i}$$

Since $\delta_i \geq \delta$, the proof is completed.

We have successfully estimated the convergence rate of order-based GA. Now we can translate the last result from [9] into the order-based case:

Corollary 1 *The optimal (by means of convergence rate) mutation probability is given by:*

$$p_{mut} = \frac{\delta}{r} = \frac{\delta}{n-1}$$

This evaluation is as useless as in classical case - we should know exactly the fitness function to evaluate δ.

Note that all these results describe the convergence rate of Q'. The Markov chain Q converges r times slower.

5 Example of application and results

The problem of **finding a minimal reduct** for a given information system is one of the NP-hard problems (see [8]) used in knowledge discovery. On the other hand, we can use short reducts to build effective decision algorithms.

We can introduce classical, binary representation for subsets of the set of attributes and use genetic algorithm to generate reducts. We have performed such an experiment and obtained good reducts after short time [10]. However, these results were still not satisfactory and the second method of generating of short reducts was introduced: a **hybrid algorithm**.

In the hybrid algorithm [10] we use simple, deterministic method for reduct generation: we start with the set of all attributes, then we attempt to remove attributes in some order. If we can find a pair of objects which is not discerned by any of the remaining attributes, we insert the removed attribute back to the subset. At last we obtain a subset which is a reduct. This deterministic algorithm can give any of the reducts of the information system, supposing the proper order of attributes. The genetic algorithm is used to generate this proper order. To claculate the function of fitness for a given permutation (order of attributes) we have to perform one run of the deterministic algorithm and calculate the lenght of the shortest reduct.

The hybrid algorithm described above performs about four times slower that the classical one. On the other hand, the reducts obtained by this algorithm are usually shorter. Moreover, the hybrid algorithm generates from 50 to 500 different reducts in comparison with 5 to 50 reducts generated by the classical GA at the same time.

The final result of the deterministic algorithm depends only on the order of attributes, with no respect to the absolute position of any of them. To demonstrate this properity of the hybrid algorithm, we have compared the average efficiency (the average lenght of obtained reducts) using three recombination operators: PMX, OX and MOX. The results obtained using OX and MOX operator were significantly better, than those obtained using PMX operator (based on absolute positions of attributes). The difference between OX and MOX operators was not so significant, although OX operator was susceptible of premature convergence. These results show, that we should use rather relative than absolute o-schemata. The high efficiency of MOX operator suggests, that rd-schemata create a good tool for analysing our hybrid algorithm.

6 Conclusion

This paper shows that the most of theoretical results concerning classical GA - both the schemata and the Markov chain analysis - can be transferred with small changes into the order-based case using the notion of rd (relative dispersed) schemata. We have introduced new MOX operator concerned with this notion of schema. This type of schemata was rarely used in literature ([1]) but never named nor analysed. We can conclude that the order-based genetic algorithms are based on as reliable theoretical foundations as the classical GAs are. On the other hand, these foundations are still not satisfactory from the practical point of view - in both classical and order-based case.

References

[1] Davis L. (ed.), 1991. Handbook of Genetic Algorithms. New York, Van Nostrand Reinhold.

[2] Goldberg D.E., 1989. GA in Search, Optimisation, and Machine Learning. Addison-Wesley.

[3] Holland J.H. 1975. Adaptation in natural and artificial systems. The MIT Press, Cambridge, 1992.

[4] Nguyen S. H., Nguyen T. T., Polkowski L., Skowron A., Synak P., Wróblewski J., 1996. Decision Rules for Large Data Tables. Proc. of Computational Engineering in Systems Applications, CESA'96, Lille, France.

[5] Nix A., Vose M. D., 1991. Modelling genetic algorithms with markov chains. In: Annals of Mathematics and Artificial Intelligence.

[6] Pawlak Z., 1991. Rough sets: Theoretical aspects of reasoning about data. Kluwer, Dordrecht 1991.

[7] Radcliffe N. J., 1991. Forma Analysis and Random Respectful Recombination. In: Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, pp. 222 - 229.

[8] Skowron A., Rauszer C., 1992. The Discernibility Matrices and Functions in Information Systems. In: Slowiński (ed.): Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dordrecht 1992, pp. 331 - 362.

[9] Suzuki J., 1993. A Markov Chain Analysis on A Genetic Algorithm. In: Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, pp. 146 - 153.

[10] Wróblewski J., 1995. Finding minimal reducts using genetic algorithms. Proc. of the Second Annual Join Conference on Information Sciences, pp.186-189, September 28-October 1, 1995, Wrightsville Beach, NC. Also in: ICS Research report 16/95, Warsaw University of Technology.

[11] Wróblewski J., 1996. Rozwiązywanie niektórych problemów NP-trudnych za pomocą algorytmów genetycznych: podstawy teoretyczne i zastosowania. Master's Thesis, Warsaw University.