MODIFIED HYBRID GENETIC ALGORITHM OF DISCREET OPTIMZATION PROBLEMS

A.O. Glukhov,
Vice professor, PhD,
Polotsk State University, PSU
Polotsk, Belarus,
alexey.glukhov@gmail.com

D.O. Glukhov,
Vice rector, PhD,
Polotsk State University, PSU
Polotsk, Belarus,
alexey.glukhov@gmail.com

V.V. Trofimov,
Chef of Informatics Department, professor,
St. Petersburg State Economics University, St. Petersburg, Russia,
tww@mail.ru

L.A. Trofimova
professor, Doctor of economics science,
St. Petersburg State Economics University, St. Petersburg, Russia,
L_Trofimova@bk.ru

Abstract*. The goal objective* **is to improve the efficiency of solving discrete optimization problems. The proposed method refers to the "fast" methods and was named the "Local genetic method". The peculiarity of this method is that the chromosomes do not encode the whole solution, but only a small part of the plan. Therefore, the method allows us introducing unary and binary operations that take into account the specific nature of the problem. The important feature of the method is the non-deterministic nature of the computation, which is due to the internal parallelism of computations and is expressed in the asynchronous action of various local strategies. In terms of speed, the proposed method in a number of experiments outperformed the traditional algorithm by more than 10 times and always found the best solution. The nature of the approximation to the optimum for these algorithms remained unchanged when solving any test cases.**

Keywords— genetic algorithms; discrete optimization

On the whole, the problem of planning (schedule, traveling salesman route, multifunctional production, etc.) is characterized by the extremely high time complexity and relates to the class of discrete optimization problems. It is effectively solved only by using approximate methods, such as genetic search or heuristic method. The proposed method also relates to "fast" methods and is named "Local genetic method" (LGM) [1]. The method can be classified as a modified hybrid genetic algorithm. The peculiarity of the method is the chromosomes encoding not the whole solution, but only a small part of the plan. Therefore, the method does not use traditional operations on chromosomes (crossover, inversion, etc.), but allows to introduce unary and binary operations that take the problem specificity into account. Such operations are purely heuristic, and their successful selection allows achieving the maximum search process efficiency in finding planning problem solutions.

To illustrate the LGM operation, we have chosen traveling salesman problem (TSP), as this problem got extensive literature coverage and the test sets of patterns are available. By using these sets, we can define the effectiveness factors of the new method as compared to the existing methods such as: heuristic methods, ant colony optimization, genetic methods, "taboo" search, neural networks, approximation, etc.

The genetic algorithm is based on the computer model of evolution [4,5,8,9]. Each individual is characterized by its chromosome, represented in the form of a symbol string:

$$X=\{s_1,s_2,\dots,s_g | s_i \in SYMBOL\} \in Chromosome.$$

Symbols are selected from a set of valid symbols (genes) - SYMBOL, encoding nodes (cities). With that, the length of chromosomes can vary within [1, ..., | SYMBOL |]. We will use the traditional terminology for the genetic algorithm, despite the differences in approaches. The optimization problem is usually formulated as the maximization of an individual's fitness function $f(X)$, i.e.

$f(X)\to max$ $X \in Chromosome$	$f(X)$: $Chromosome \to R^+$, Where R + is the set of nonnegative real numbers.

It is characteristic for heuristic algorithm to use another function for comparing the solutions' quality - namely the preference function $Q(X)$. The optimization problem in this case is formulated as $Q(X)\to max$. The preference function is usually constructed as a heuristic penalty function for the solution imperfection. In fact, there are no obstacles for switching over to the use of the heuristic preference function in the genetic algorithm instead of the fitness function. In doing so, we are able to operate with "incomplete" chromosomes as well.

To perform the search, we need to have a certain set of operations on chromosomes $ACTION=\{a_1,a_2,\dots,a_M\}$ and a certain set of conditions for performing these operations $CONDITION=\{c_1,c_2,\dots,c_M\}$.

We shall give the name of heuristic strategy σ to the pair consisting of an operation and its activation condition:

$\sigma_i=(a_i,c_i)\in STRATEGY.$

c_i: $Chromosomen^n \to B|\ n=1,2,\dots$ is a predicate, where $B=\{true,false\}$ is the space of logical variables.

a_i: $\text{Chromosomen}^n \to \text{Chromosome} \mid n=1,2,\ldots$ - unary, binary and n-ary operations.

Calculations using the above heuristic strategies require specifying a certain population of $P \subseteq \text{Chromosome}$. Before starting the calculations, we must specify the initial P_0 population, which, for example, can be constructed in a random fashion.

We formalize the local genetic algorithm of solving the TSP problem:

LGA: 1. create an initial population P_0;

2. Calculate Q(X), $X \in P_k$, where Q is the preference function;

3. Form the next-generation population $P_{k+1}=Sep(P_k \cup P_k^{/})$, where Sep is the selection operator, $P_k^{/}$ is the set of individuals resulting from the strategy set operation STRATEGY;

4. If the condition of ending the evolutionary search is not met, go to item 2;

The obvious LGA advantage is the reduction of the algorithm computational complexity due to the shortening of the analyzed and changed chains, and also to the fact that for heuristic operations, as a rule, calculating the value change in the preference function is possible without calculating its new value "from scratch ". Another LGA advantage is the increased flexibility as compared to the classical version of the genetic algorithm. This flexibility manifests itself by the joint action of various strategies, along with the ability to easily modify the heuristic strategies involved in it (to introduce new ones, enable or disable them).

The important LGA feature is the non-deterministic nature of the computations, resulting from the internal parallelism of computations. The latter is expressed in the asynchronous operation of various local strategies. Lack of artificial accident allows target-oriented movement in the problem solution space, proceeding only towards the "*potential improvement*" of the preference function value (this is how we understand the heuristic strategy). As the potential for significant “*improvement*" decreases with approaching *the optimum*, the character of the movement in LGA becomes more predictable, as compared to the traditional genetic algorithm.

The main drawback of the algorithm is its heuristic nature. Combination of different strategies allows overcoming a greater number of local minima than a conventional heuristic algorithm, but *does not guarantee* achieving the optimum. The second obvious drawback (the consequence of the first one) – is that we can have only statistical estimates of the algorithm effectiveness, thus requiring a vast number of experiments in various tasks for obtaining reliable information.

Strategy selection. In TSP problem the chromosome encodes the traveling salesman's route that can be estimated by using the following preference function:

$$Q(X) = Length + A \cdot (|X|-N) + B \cdot DoNotClose,$$

Where Length is the route length, with mandatory return of traveling salesman to the starting point, *A* is the penalty for missing the city, *B* is the penalty for incomplete route; DoNotClose is equal to 1 if S_i is a partial solution, otherwise 0. The following values are used for the penalty variables (A = 1000, B = 3000) for the geographic region model with the size of 500x400.

To formalize the heuristic strategies description, we shall introduce the generating operator for a new individual (chromosome) - $X_{PARENT} \Rightarrow X_{NEW}$. The heuristic strategies set STRATEGY will incorporate the following strategies $\{\sigma_{grow}, \sigma_{transition}, \sigma_{exchange}, \sigma_{rotation}\}$. Hereafter, the following is true for all the strategies:

$X, X^{/} \in \text{Chromosome}$, $COND = s_x \notin X \wedge s_x \in SYMBOL$.

Below are the formally defined strategies:

$\boldsymbol{\sigma_{grow}} = ((X \Rightarrow X^{/} = X \cup <s_x>), COND)$.

$\boldsymbol{\sigma_{transition}} = ((X = X_B \cup <s_i> \cup X_M \cup X_E \Rightarrow X^{/} = X_B \cup X_M \cup <s_i> \cup X_E), true)$.

$\boldsymbol{\sigma_{exchange}} = ((X = X_B \cup <s_i> \cup X_M \cup <s_m> \cup X_E \Rightarrow X^{/} = X_B \cup <s_m> \cup X_M \cup <s_i> \cup X_E), true)$.

$\boldsymbol{\sigma_{rotation}} = ((X = X_B \cup <s_i> \cup X_M \cup <s_m> \cup X_E \Rightarrow X^{/} = X_B \cup <s_i> \cup inverce(X_M) \cup <s_m> \cup X_E), true)$.

The inverce(X) function changes the sequence order of symbols in the X set for the reverse one.

All strategies allow calculating change of the preference function value for chromosomes, thus significantly reducing computational costs.

Results. Comparative effectiveness indexes of the traditional genetic algorithm - GA and LGA are given below in Table 1. The algorithm parameters selected for the experimental set up were the ones achieving the best results on the same test examples. So, for example, for GA – population size - 50, elite crossing - 50, mutation - 3, inversion – 8. For LGA - population size - 50, the transition - 6, torsion - 6, exchange - 6. The condition for the search termination was the lack of noticeable solution improvement within a certain period (about 15% of the calculation time).

Tabl.1

Size	GA		LGA	
	t, sec	Q(X)	t, sec	Q(X)
50	12	2288	3	2076
80	28	4408	4	2930
140	35	4441	12	4142
200	135	5103	17	4616
270	261	4755	36	4431
350	710	6021	51	5498

Data is obtained using the program LGA.exe

Individual experiments were also conducted on the test examples taken from TSPLIB. The calculation results are given in Table 2.

Tabl.2

Task	Size	current optimum (TSPLIB)	LGA	
		Q(X)	t, min: sec	Q(X)
eil51	51	429.983	3:04	428.982
rd100	100	7909.1	10:32	7908.04
eil76	76	545.388	4:14	545.388
pr76	76	108159	17:12	108159
cp150	150	6527.85	30:40	6566.58

Data is obtained using the program LGA.exe

In the first two problems we succeeded in improving the solution previously accepted as optimal until now.

Conclusion

The obtained results support the proposed algorithms. Their speed LGA and accuracy significantly exceed the corresponding characteristics of the traditional algorithm. So, in terms of speed, LGA in experimental series outperformed the traditional algorithms by more than tenfold and always found the best solution. The "spacing" in quality of solutions averaged about 9% -12%. The character of the optimum approximation for these algorithms remained unchanged with solving any test examples

It is also interesting that we have completely eliminated binary operations (for example, crossing) from the genetic algorithm, while its efficiency increased due to introducing sufficient amount of unary (mutational) special type operations. The latter ones, as a rule, allow us to estimate the amount of solution value change ΔQ, thus eliminating the need for a full price calculation of new solutions. So for $X^{/}=a_i(X)$ we have $Q(X^{/})=Q(X)+\Delta Q$. Thereby, the computational costs of the new algorithms go noticeably down. This allows us to assume that the effectiveness of local genetic algorithms will significantly exceed the traditional algorithm effectiveness, specifically in solving large and very large-scale problems. These are, for example, planning and scheduling problems, transport task with greater number of "cities", etc.

We view the improvement of search strategies in LGA and the possible introduction of adaptive behavior in algorithm as themes for future research.

REFERENCES

[1] Глухов А.О., Трофимов В.В., Трофимова Л.А. Аппроксимация сложных, многопараметрических, существенно нелинейных, динамических зависимостей на основе генетических алгоритмов. XIX Международная конференция по мягким вычислениям и измерениям (SCM-2016). Сборник докладов в 2-х томах. Санкт-Петербург. 25-27 мая 2016 г. Т.1. – С.477-480.

[2] Глухов А.О., Трофимов В.В. Использование рекурсивных эвристик при решении задач дискретной оптимизации большой размерности // Современные проблемы менеджмента: межвуз. сб. Выпуск 6. – СПб.: Изд-во СПбГУЭФ, 2003. – С. 99-105.

[3] Глухов А.О., Трофимов В.В., Глухов Д.О. Локальный генетический алгоритм планирования процесса многопрофильного производства / Экономическая кибернетика: системный анализ в экономике и управлении: Сб. научных трудов. Выпуск №5 – СПб.: СПбГУЭФ, 2002 – С. 50-56.

[4] Trofimov V.V., Hluhov A.O. The Optimal Schedule for the Technological Process of Semiconductor Production. Operations Research 2002: International Conference on Operations Research Sept.2-sept.5, 2002, Klagenfurt, Austria, Klagenfurt University, 2002. – P.148.

[5] Трофимов В.В., Глухов А.О. Метод решения задачи идентификации и моделирования самоорганизующихся систем на основе генетического подхода. Международная конференция по мягким вычислениям и измерениям, Санкт-Петербург, 25-27 июня 2001, Сборник докладов. СПб.: ЛЭТИ, 2001, том 1. – с.291-293.

[6] Trofimov V.V., Hluhov A.O. Construction of the Optimal Schedule for the Technological process of semiconducting production. 46. International Scientific Colloquium. “Multimedia – The Challenge for Science, Technology and Business”. 24-27.09.2001 Technique University of Ilmenau, Germany 2001.

.