Elitism Based Evolutionary Algorithm for Discrete Optimization Problems

Vassil Guliashki[1], Chavdar Korsemov[2] and Hristo Toshev[3]

Abstract – **In this paper different elitism strategies are considered. They are combined in an evolutionary algorithm, designed to solve discrete optimization problems. The computational complexity of the proposed algorithm is investigated and evaluated. Conclusions about the use of elitism are drawn.**

Keywords – **Evolutionary algorithms, Elitism, Discrete optimization.**

I. INTRODUCTION

In this paper is considered integer programming problem, which can be stated in the following form:

$$\mathbf{Min}\ F(x) \quad (1)$$

$$\textbf{subject to: } g_i(x) \le 0; \quad i = 1,\dots,m; \quad (2)$$

$$l_j \le x_j \le u_j; \quad j = 1,\dots,n; \quad (3)$$

$$x \in \mathbf{Z}^n, \quad (4)$$

where x is an n-dimensional vector of variables x_j, $j = 1,\dots,n$; which accept discrete values only. By l_j and u_j are denoted the bounds (lower and upper) of x_j, and $F(x)$ is the multimodal objective function. There is no necessary $F(x)$ to posses accessible to calculation derivatives in an explicit analytical form. The functions $g_i(x)$, $i = 1,\dots,m$; are convex nonlinear functions and m is the number of nonlinear constraints **(2)**

The integer programming problems belong to the class of NP-hard optimization problems (see [2, 6]). There does not exist an exact algorithm, which can solve these problems in time, depending polynomially on the problem input data length or on the problem size. For this reason many efficient approximate evolutionary algorithms and metaheuristic methods have been created to find out the global optimum of such complex optimization problems. The objective function **(1)** with integer variables is usually multimodal function. The approximate algorithms may not be able to find out the exact global optimal solution. A possible strategy to solve such complex optimization problems is to combine the qualities of a directional type method with the good features of a population based (evolutionary) algorithm. The directional type steps may accelerate the convergence in regular regions of the search space, while the evolutionary algorithms are able to escape the trap of local optima, exploring the whole feasible domain. For precisely locating the optimal solution(s) some kind of local search procedure may be included in the optimization algorithm.

The population based algorithms (see [5]) handle a population of individuals and make them evolve according to some rules that are clearly specified for each algorithm. At each iteration periods of *self-adaptation* (intensification of the search process in some region of the search space) alternate with periods of *co-operation* (information collective gathered during the search process is used to direct further the search). The periods of self-adaptation correspond to execution of mutation, improvement or local search procedure, and the periods of co-operation are connected with the selection, crossover, trace updating or generation of trial points, i.e. with some (explicit or implicit) sharing among the individuals useful information gathered during the search. The population based algorithms are organized according the following general scheme:

Generate an initial population of individuals;
While no stopping condition is met **do**
Co-operation,
Self-adaptation,
EndWhile.

Figure 1: General scheme of a population based algorithm

Many metaheuristic algorithms correspond to this framework and can be described within it.

One major difficulty with the creation of global search algorithms is to overcome the premature convergence towards local optima. To obtain the global optimum of the problem at hand a diversification phases of the search process are necessary, so that new areas of feasible domain, that are remained still unexplored, can be investigated. This means that after each intensification phase and locating a local optimum the search procedure has to steer the individuals away from the just explored region and from the found local optimum, as well as away from all other known local optima and their corresponding regions. During the diversification phase the individuals are modified independently but with unexpected results in the sense that they are not necessarily improved. A famous example for diversification is the tabu list strategy in the tabu search algorithms, where some characteristics of solutions or movements (steps in given directions) are stored as forbidden (tabu) for certain number of iterations. In this manner are avoided the cycling and the trap of local optimality. In general the metaheuristic algorithms

[1]Vassil Guliashki is with the Institute of Information Technologies, “Acad. G. Bonchev” str. Bl. 29A, 1113 Sofia, Bulgaria, E-mail: vggul@yahoo.com

[2]Chavdar Korsemov is with the Institute of Information Technologies, “Acad. G. Bonchev” str. Bl. 29A, 1113 Sofia, Bulgaria, E-mail: chkorsemov@iinf.bas.bg

[3]Hristo Toshev is with the Institute of Information Technologies, “Acad. G. Bonchev” str. Bl. 29A, 1113 Sofia, Bulgaria, E-mail: toshev@iinf.bas.bg

alternate phases of intensification with diversification phases. To obtain guaranteed the global optimal solution these algorithms should perform a search process, exploring the whole feasible domain, i.e. they should perform systematically diversification after each intensification period.

Another major difficulty with the creation of global search algorithms is to obtain good convergence speed, because they could converge very slowly. This could be achieved by means of some kind hybridization. To create successful and efficient global search methods the researchers very often make combinations of two or more metaheuristics in hybrid methods. Such techniques have been proposed in [1, 3, 4, 7]. For example tabu search can be coupled with directional search and scatter search approaches. Another important way to accelerate the performance of an evolutionary algorithm is to use the features of the best individuals obtained during the search process and the historically good information they have accumulated, combining them to generate new offspring individuals for the next population, i.e. to use some kind of elitism.

In this paper we propose an evolutionary algorithm for solving the problem (1)-(4), which includes different elitism strategies for selection the parent individuals, generating the next population in the search process.

II. Elitism strategies

A. Scatter search elitism strategy

Such kind strategy has been proposed for the first time by F. Glover (see [3]). Let we have a population P containing p individuals or solutions in the feasible domain for the considered optimization problem (1-4). The initial population P can be randomly generated around the Tchebicheff center x^{tch} of the feasible domain, determined by the constraints (2)-(3). Let the feasible domain be denoted by X. The Tchebicheff center $x_{tch} \in X$ is the point located at the maximal Euclidean distance from the constraint surfaces. We assume that the Tchebicheff center is obtained by means of a method for solving the relaxed convex problem with continuous variables. Then x^{tch} is rounded to the nearest integer point ix^{tch}.

The objective function values should be calculated for each individual in the population P. Using this information the most attractive elements of P can be selected and used to generate new elements that replace previous members of P. Elite elements (those with especially good objective function values) become part of a special historical collection that is used at the final stage of the search to contribute to the generation of final solution. The following sets are referenced and updated at each iteration:

T – a set of trial solutions;

S – a set of s selected current generators;

H – a set of elite historical generators, consisting of h best (with highest evaluation) solutions generated historically throughout the search. On the first iteration H will consist of the h best elements of S.

The scatter search strategy consists of the following steps:

Step 1. Set $i = 0$ and $itlim = itl$, where itl is a positive integer. Generate initial set of trial solutions T(i).

Step 2. Select the best s solutions from T(i) and create S(i). Create set of elite historical generators H(i), including the h best elements of S(i) in H(i).

Step 3. Let x and y are two solutions, $x \in$ S and $y \in$ S. For each pair $(x, y)_j$ consider the line through x and y, given by the expression

$$l(t) = x + t(y-x), \qquad \textbf{(5)}$$

where t is scalar weight.

The solutions $l(-1/3)$, $l(1/3)$, $l(2/3)$, $l(4/3)$ are generated. Each of them is rounded of to its nearest integer feasible solution and are obtained four new solutions: $xj^{(1)}$, $xj^{(2)}$, $xj^{(3)}$, $xj^{(4)}$;

Calculate for each generated offspring solution its objective function value.

Step 4. Update the set H(i) as follows: If some generated offspring solution has better objective function value than some solution in H(i) replace the corresponding solution in H(i) by this solution.

Step 5. Set $i = i+1$. If $i > itlim$, then STOP (the procedure terminates), else use the trial solutions T(i-1) to generate new trial solutions T(i) and go to Step 2.

B. Neighbor S_i *sets – based elitism strategy*

The main idea of this strategy is to select a given percent of the best solutions included in j different neighbor sets S_i, $i=1,\ldots j$; (i.e. in j different neighbor areas of the search space) and to create a new population PS sharing their accumulated information. Then each pair solutions (x, y), where $x \in PS$ and $y \in PS$, is used as a parent pair to generate four new solutions analog to Step 3. in the Scatter search strategy. The corresponding procedure is described by the following steps:

Step 1. Let s be the cardinality of each set Si. Select $\lfloor s/j \rfloor$ best solutions from each of neighbor sets S_i, $i=1,\ldots j$; (here $\lfloor v \rfloor$ denotes the integer part of v) and create the new population PS. Create set of elite historical generators $H^* \equiv PS$.

Step 2. Let x and y are two solutions, $x \in PS$ and $y \in PS$. For each parent pair $(x, y)_p$ consider the line through x and y, given by the expression **(5)**: $l(t) = x + t(y-x)$, where t is scalar weight.

The solutions $l(-1/3)$, $l(1/3)$, $l(2/3)$, $l(4/3)$ are generated. Each of them is rounded of to its nearest integer feasible solution and are obtained four new solutions: $xp^{(1)}$, $xp^{(2)}$, $xp^{(3)}$, $xp^{(4)}$;

Calculate for each generated solution its objective function value.

Step 3. Update the set H* as follows: If some generated offspring solution has better objective function value than some solution in H* replace the corresponding solution in H* by this solution. STOP (the procedure terminates).

III. The proposed elitism based evolutionary algorithm EEA

The idea of the proposed EEA algorithm is the following:

Let the feasible domain be denoted by X. The Tchebicheff center $x_{tch} \in X$ is the point located at the maximal Euclidean distance from the constraint surfaces. We assume that the Tchebicheff center is obtained by means of a method for solving convex problems with continuous variables. Then x_{tch} is rounded to the nearest integer point i_{tch}. A regular simplex with n+1 vertices is generated around the solution i_{tch}. There are (n+1) combinations of n vertices, correspondingly for each facet of the simplex. The feasible domain X is divided in (n+1) sub-regions as follows: The rays starting at the weight center of all simplex vertices cs and passing through the vertices $v(j)$, j=1,…,n; belonging to each facet determine $K^{(i)}$ cones, i = 1,…, n+1; in the feasible domain. Similar division of the feasible region is proposed in [4]. The iterative procedure generates at each iteration k a number i of trial points sets $T(i)_k$ for each facet of the simplex i=1,…,n+1; Each set T(i) includes obtained by rounding off (to the nearest integer solutions) of points along the rays r_j starting at cs and passing trough v(j), as well as along the ray r_{ci}, starting at cs and passing through the weight center cv_i of the current facet. By means of each T(i) a set Si of s selected current generators is created and the described scatter search strategy is performed on it. All Si sets at the k-th iteration are used that to perform the Si sets – based elitism strategy. Than the sets of elite historical generators H_k and H^*_k are compared and in case H_k contains some better solutions than a solution in H^*_k, the better solution replaces the worse. At the end of all iterations the procedure finishes by applying the scatter search strategy on the final H*.

The steps of EEA algorithm are presented below:

Step 1. Find the Tchebicheff center x_{tch} and round off it to the nearest feasible integer solution i_{tch}.

Step 2. Generate a regular simplex with n+1 vertices, using i_{tch} as one vertex. Generate the other vertices on the base of the elementary geometry in the following manner:

$$v^{(i)}_j = \begin{cases} i_{tch\,j} + \varphi_1 & \text{if } j \neq i \\ i_{tch\,j} + \varphi_2 & \text{if } j = i \end{cases} \qquad \textbf{(6)}$$

$$\varphi_1 = \alpha.\left[\frac{\sqrt{(n+1)} + n - 1}{n\sqrt{2}} \right] \qquad \textbf{(7)}$$

$$\varphi_2 = \alpha.\left[\frac{\sqrt{(n+1)} - 1}{n\sqrt{2}} \right] \qquad \textbf{(8)}$$

Let i_{tch} be denoted as $v^{(0)}$. Round off each $v^{(j)}$, j = 1,…,n; to its nearest integer point.

Step 3. Calculate the weight center of the simplex:

$$cs = \frac{\sum_{j=0}^{n} v^{(j)}}{n+1} \qquad \textbf{(9)}$$

Round off cs to its nearest integer point. Set i = 1.

Step 4. For i = 1,…, n+1; explore the cone $K^{(i)}$ as follows: Create a set of trial points $T^{(i)}$, including in it the simplex vertices of the current simplex facet and their weight center cv, where

$$cv = \frac{\sum_{v^{(i)} \in K^{(i)}} v^{(i)}}{n} \qquad \textbf{(10)}$$

Round off cv to its nearest integer point.
Calculate the objective function values of the trial solutions $x \in T^{(i)}$. Create the set Si of the best s members $x \in T^{(i)}$.

Step 5. Perform the described Scatter search strategy on the set Si. Update the set of elite historical generators Ho.

Step 6. Perform the described Si –based elitism strategy. Update the set of elite historical generators Ho*.
Let: $xold^{(j)} = v^{(j)}$, j = 1,…, n; and $xold^{(0)} = cv$.
Calculate $p^{(j)} = \gamma(xold^{(j)} - cs)$, j = 1,…, n; and $p^{(0)} = \gamma(cv - cs)$, where γ is a scale multiplier, tuned according to the concrete problem.

Step 7.

ITERATION

Let the iteration has the current number ik. Calculate the points $xnew^{(j)} = xold^{(j)} + ik.p^{(j)}$, j= 0,…, n; Round off each $xnew^{(j)}$ to its nearest integer point. In case there is a violated constraint from the system **(2)-(3)** reduce the corresponding $p^{(j)}$ as follows:

- If a constraint of type $\boldsymbol{x_k + a = 0}$ is violated, where $\boldsymbol{a}$ can have positive or negative value, then the corresponding component p_k of $p^{(j)}$, is used to reduce $p^{(j)}$:

$$p^{(j)} = \left| \frac{a - p_k}{p_k} \right| p^{(j)} \qquad \textbf{(11)}$$

- If a constraint of type $\boldsymbol{g_i(x) \le 0}$ is violated then $p^{(j)} = 0.8\ p^{(j)}$. If it is necessary repeat this reduction until the rounded off integer $xnew^{(j)}$ becomes feasible.

- If there are more than one constraints, violated by $p^{(j)}$, then chose the most reduced vector $p^{(j)}$, so that the rounded off integer $xnew^{(j)}$ becomes feasible.

Use the generated solutions $xnew^{(j)}$ as trial solutions. For each facet include the corresponding trial solutions in the set $T^{(i)}_{ik}$. Calculate also ik new trial points on each segment seg_i between cv and $v(i)$, where each segment seg_i is divided to (ik+1) parts of equal length. The new generated ik solutions on each segment seg_i are rounded off to the corresponding nearest integer feasible point and are included in $T^{(i)}_{ik}$. Evaluate the objective function values of the trial solutions $x \in T^{(i)}_{ik}$.
Perform the described Scatter search strategy on the set Si. Update the set of elite historical generators H_{ik}.
Perform the described Si –based elitism strategy. Update the set of elite historical generators H_{ik}*.

ENDofITERATION

Repeat the *ITERATION* until it is not possible to generate any new feasible points by means of $p^{(j)}$, j=0,…, n;

Step 8. Compare all obtained H_{ik} and H^*_{ik}. In case H_{ik} contains some better solutions than a solution in H^*_{ik}, the better solution replaces the worse. Create a final population H* of the solutions in all H^*_{ik}. Perform the described Scatter search strategy on the set H*. Find the best generated solution x^*. STOP.

The computational complexity of EEA algorithm is considered and evaluated as follows:

Theorem: The EEA algorithm has a polynomial computational complexity $O(n^4)$.

Proof: There are n+1 different cones as described in Step 4. On each facet are generated n+1 trial solutions and additional $ik.n$ trial solutions, where ik is the iteration number. There are $s(s–1)/2$ combinations of two solutions belonging to Si, hence $s(s–1)/2$ segments will be considered. The scatter search strategy evaluates 4 solutions for each segment, i.e. $2s(s–1)$ new solutions. The same number of new solutions is generated also in the Si – based elitism strategy. The scatter search strategy is applied n+1 times at each iteration and the Si – based elitism strategy – only once (see Step 7.). Hence at each iteration are evaluated no more than $2(n+2)s(s–1)$ solutions, where s depends linear of n (see Step 4). There are performed ik iterations. Depending on the feasible domain we could expect that $ik \sim n$. Then the total number of evaluated solutions during the performance of EEA algorithm is proportional to $(n-1)(n+2)n^2$. Hence the Theorem is proved. ■

IV. Conclusions

The presented new evolutionary method EEA has the following good features and advantages:

- During the exploration of each sub-region the EEA algorithm systematically diversifies the search process, avoiding in this manner the trap of local minima.
- The EEA algorithm performs search in all defined sub-regions of the search space, so that the whole feasible domain is explored.
- The formed final population contains diverse enough individuals, so that it is expected that the final search phase would lead to the global optimal solution.
- The EEA algorithm can be efficient in comparison to other global search algorithms, because it is an elitism based polynomial algorithm. The non elitism procedure based on the same steps would evaluate 2 or three times more solutions at each iteration. Hence it will perform two or three time slower than EEA algorithm.
- The populations used in EEA algorithm have relatively small size, so that no great memory will be necessary for its implementation.
- The EEA algorithm guarantees the feasibility of the obtained solutions.
- A great part of the integer points located near or on the rays forming each cone, which have been explored during the search in the corresponding sub-region at the current iteration, can be used during the exploration of the next sub-region. This may be used for creation of efficient program realizations of EEA algorithm.

The EEA algorithm will be tested on a set of test examples and the results will be compared with that of non elitism based evolutionary algorithms.

Acknowledgement

The authors gratefully acknowledge the support of Bulgarian National Science Fund, Grant No DTK02/71 “Web-Based Interactive System, Supporting the Building Models and Solving Optimization and Decision Making Problems”.

References

[1] Damavandi N. and Safavi-Naeini S., “A hybrid evolutionary programming method for circuit optimization”, In Proc. *IEEE Transactions on circuits and systems – I: Regular papers*, vol. 52, No. 5, May 2005, pp. 902-910.

[2] Garey M. R. and Johnson D. S., “*Computers and Intractability: A Guide to the Theory of NP-Completeness*”, W. H. Freeman, San Francisco 1979.

[3] Glover F., “Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms)”, *Discrete Applied Mathematics,* “Viewpoints on Optimization”, 1991.

[4] Guliashki, V., (2007) “A Hybrid Population Based Method Solving Convex Integer Optimization Problems”, In: Proceedings of Papers of the XLII International Scientific Conference on Information, Communication and Energy Systems and Technologies ICEST2007, (Editor Prof. Dr. Mitrovski, C.), June 24 – 27, 2007, Ohrid, Macedonia, vol. I, pp. 249-252.

[5] Hertz A., Kobler D., „A Framework for Description of Population Based Methods“, Tutorials and Research Reviews, 16-th European Conference on Operational Research Brussels, Belgium, 1998, pp. 48-59.

[6] NemhauserG. L. and Wolsey L. A. “*Integer and Combinatorial Optimization*”, Wiley, New York. 1988.

[7] Taillard E. D., Gambardella L. M., Gendreau M. and J.-Y. Potvin, “Adaptive Memory Programming: A Unified View of Metaheuristics”, *Tutorials and Research Reviews*, 16-th European Conference on Operational Research Brussels, Belgium, 1998, pp. 30-38.