# Accelerating Strategies in Evolutionary Algorithms

Vassil G. Guliashki[1], Leoneed Kirilov[2]

*Abstract* – **We present three strategies designed to accelerate the convergence during the search process of evolutionary algorithms for convex integer optimization problems. The strategies realize a systematic diversification of the search. They are compared with performance of scatter search and particle swarm optimization.**

***Keywords*** **– evolutionary algorithms, convex integer optimization problems, accelerating strategies.**

## I. INTRODUCTION

We consider the convex integer programming problem in the form:

$$\textbf{Min } F(x) \tag{1}$$

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

$$l_j \le x_j \le u_j; \quad j = 1,\ldots,n; \tag{3}$$

$$x \in \mathbf{Z}^n, \tag{4}$$

where $x$ is an $n$-dimensional vector of integer variables $x_j$, $j = 1,\ldots,n$. By $l_j$ and $u_j$ are denoted the bounds (lower and upper) of $x_j$, and $F(x)$ is the multimodal objective function. $F(x)$ may not possess derivatives in an explicit analytical form. The functions $g_i(x)$, $i = 1,\ldots,m$; are convex nonlinear functions and $m$ is the number of nonlinear constraints **(2)**.

The convex integer problems (see [6, 18]) belong to the class of NP-hard optimization problems. 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 (see [8,11,14,17,19,21]).

To solve problem (1-4) many algorithms which mimic the natural evolution process of species have been designed in order to obtain a global optimum. They could be classified as “evolutionary” or “population based” algorithms (see [14]). The most familiar and powerful among them are Genetic Algorithms (GA) (see [11, 15]), Scatter Search (SS) (see [7, 9]), Tabu Search (TS) (see [8, 9, 10]), Ant Systems (AS) (see [1, 2, 3]) and Particle Swarm Optimization (PSO) (see [5, 16, 17]). The evolutionary algorithms use a population of feasible solutions (or characteristics of solutions), called *individuals, trial (dispersed) points, ants, particles* etc. In this paper is used the term *individuals*.

The evolutionary algorithms usually use an improvement sub-procedure to intensify the search process in some regions of the search space. One such sub-procedure is the local search. It is supposed that each intensification period finishes with a found local optimum. To escape from the local optimality a diversification of the search process is necessary after the intensification period.

To be efficient an evolutionary algorithm for search a global optimal solution it should quickly perform the diversification of the search. Different ways for diversification of the search have been developed. For example, during the diversification phase the individuals could be modified independently – like the mutation in GA. But the results are unexpected in the sense that the modification does not lead necessarily to an improvement. A famous example for diversification is the Tabu list strategy used in the Tabu search algorithms. Some characteristics of solutions or movements (steps in given directions) are stored as forbidden (tabu) for certain number of iterations. In this manner the cycling and the trap of local optimality are avoided. Successful diversification of the search process is the use of non-convex combinations of parent solution vectors. In this way individuals that lie in new regions are systematically generated (see [7, 9]).

To achieve good convergence speed the successful global search methods combine usually two or more metaheuristics in hybrid methods. For example GA are combined with Tabu Search methods, or with a faster local search procedure, AS – with local search techniques (see [21]), GA – with clustering procedure (see[4]), SS – with TS or SS – with GA (see [9]). Tabu search can also be coupled with directional search approach. 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. This is an elitism – based approach for generating new offspring individuals (see for example [13]). A possible strategy is to combine the qualities of a directional type method with the good features of evolutionary algorithms. 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. Accelerating the systematic diversification is an open area for further development of search strategies.

In this paper three strategies for fast systematic diversification of the search process are proposed. The proposed accelerating strategies are described in Section II. An illustrative example is given in Section III. Some conclusions are drawn in Section IV.

[1]Vassil G. Guliashki is with the Institute of Information and Communication Technologies – BAS, “Acad. G. Bonchev” Str. Bl. 2, 1113 Sofia, Bulgaria, E-mail: vggul@yahoo.com

[2]Leoneed Kirilov is with the Institute of Information and Communication Technologies – BAS, “Acad. G. Bonchev” Str. Bl. 2, 1113 Sofia, Bulgaria, E-mail: leomk@abv.bg

## II. THE HEURISTIC ACCELERATING STRATEGIES

Considering the search process for global optimum there is no reason the search to be directed to the region of the best found so far (local optimal or near optimal) solution, because in the most cases it will not coincide with the global optimal solution. The same is valid for all known local optimal solutions, as well for all explored regions of the feasible domain. The exploration of the whole feasible domain means that there is a guaranteed systematic diversification of the search process. A hybrid method performing systematically diversified search (SDS-method) by means of separating the feasible domain in sub-regions (cones having a common vertex) is proposed in [12]. The systematic diversification of the search consists in exploring the cones obtained one by one.

Let the feasible domain be denoted by $X$ and let the Tchebicheff center (the point located at the maximal Euclidean distance from the constraint surfaces) be $x_{tch} \in X$. We assume that $x_{tch}$ is obtained by means of a method for solving convex problems with continuous variables. Then $x_{tch}$ is rounded off to the nearest integer point $i_{tch}$.

*A. Wave-spreading strategy*

**Step 1.** Generate a regular simplex with $n$+1 vertices, using $i_{tch}$ as one vertex. The other simplex vertices are generated in the following manner:

$$v^{(i)}_j = \begin{cases} i_{tch\,j} + \varphi_1 & \text{if } j \neq i \\ & i = 1,\ldots,n;\ j = 1,\ldots,n; \\ i_{tch\,j} + \varphi_2 & \text{if } j = i \end{cases} \quad \textbf{(5)}$$

$$\varphi_1 = \alpha.[\ \frac{\sqrt{(n+1)}+n-1}{n\sqrt{2}}\ ] \qquad \textbf{(6)}$$

$$\varphi_2 = \alpha.[\ \frac{\sqrt{(n+1)}-1}{n\sqrt{2}}\ ] \qquad \textbf{(7)}$$

Let $i_{tch}$ be denoted as $v^{(0)}$. Round off each $v^{(j)}$, $j = 1,\ldots,n$; to its nearest integer point. There are ($n$+1) combinations of $n$ vertices, correspondingly for each facet of the simplex.

**Step 2.** Calculate the components of the simplex weight center as follows:

$$cs_i = \frac{\sum_{j=0}^{n} v_i^{(j)}}{n+1}, \quad i=1,\ldots,n \qquad \textbf{(8)}$$

Round off each component $cs_i$ to its nearest integer value.

**Step 3.** Create an initial population $P_0$ around the weight center $cs$, containing $p$ uniform distributed solution vectors, generated by using deviation of ±δ, where δ is a constant % of corresponding component (for example δmax = ±1%).

**Step 4.** *Intensification phase*

Here is used a reflection like the idea in the simplex method by Nelder and Mead (see [20]).

**a)** Order the individuals (solutions) in the current population in increasing order of their $F$-values.

**b)** Calculate the weight center $P_c$ of first $k$ individuals:

$$P_c = \frac{1}{k}\sum_{j=1}^{k} x^{(j)} \qquad \textbf{(9)}$$

Here $k$ is chosen to cover about 10% to 50% of the individuals in the population.

**c)** Let $x_{w1}$ ,…, $x_{wk}$ be the individuals in the current population having the worst (i.e. the greatest) $F$-values. Calculate the steps:

$$\mathbf{y}_i = P_c - x_{wi}\ ,\ i = 1,\ldots,k; \qquad \textbf{(10)}$$

**d)** Reflect the $k$ worst individuals towards $P_c$ to generate $k$ new individuals (solutions):

$$x_{newi} = P_c + \mathbf{y}_i,\ i = 1,\ldots,k. \qquad \textbf{(11)}$$

Round off each $x_{newi}$ is rounded off to its near integer point. In case someone new solution is infeasible, i.e. the constraints **(2)-(3)** are violated, restrict the step length:

$$\mathbf{y} = \theta.\ \mathbf{y}, \qquad \textbf{(12)}$$

where $\theta \in (0,1)$.

**e)** In case someone of the so generated individuals is better than one of the current population, the better individual replaces the worse. If there aren't generated better individuals continue by **Step 5**, otherwise go to **a)**.

**Step 5.** *Diversification phase*

**a)** Make step $\beta(v^{(j)} - cs)$ along each ray starting at $cs$ and passing through the simplex vertices $v^{(j)}$, $j = 0,\ldots,n$; in outside direction, so that the new central solutions $cs^{(j)}$, $j = 0,\ldots,n$; are generated.

**b) A**round each point $cs^{(j)}$ are generated $p$ uniform distributed solutions' vectors like in **Step 3** and build ($n$+1) new populations $P_j$.

**c)** Perform the *Intensification phase* for each new population $P_j$ , $j = 0,\ldots,n$;.

**d)** Make step $\beta(cs - v^{(j)})$ along each ray starting at $cs$ in the opposite of sub-step **a)** direction, so that the new points $cs^{(j)}$, $j = 0,\ldots,n$; are generated. Perform the sub-steps **b)** and **c)**.

**Step 6.** Alternate the *Diversification* and the *Intensification* phase in the same way until reaching the boundaries of the feasible region.

**Step 7.** Perform simple local search around each found locally optimal solution to precisely locate all found optima.

*REMARKS:*

The initial simplex gets larger and larger in the search space like a wave raised by a stone in a lake.

The parameter β depends on the size of feasible region. For relative small domains the greatest component of $\beta(v^{(j)} - cs)$ is 10% of the greatest among the values $Q_j = u_j - l_j$, for $j = 1,\ldots,n$; For larger domains β should be chosen smaller.

For large feasible domains also the rays passing through $cs$ and through each of the ($n$+1) weight centers of simplex vertices determining each simplex facet should be explored in the way described above.

*B. Slicing strategy*

In this strategy the feasible domain will be separated (sliced) in $t$ sub-regions as follows:

**Step 1.** Compare the values $Q_j = u_j - l_j$, for $j = 1,\ldots,n$; and find out the maximal value $Q_j^{(\max)}$ for fixed $j = j_{\max}$. Let $q$ be the integer part of $Q_j^{(\max)}/t$:

$$q = \left\lfloor \frac{Q_j^{\max}}{t} \right\rfloor \qquad \textbf{(13)}$$

and let $l_j^1 = l_j$, $u_j^1 = l_j^1+q-1$, $l_j^i = u_j^{i-1}+1$; $u_j^i = u_j^{i-1}+q-1$; for $i$=2,…,$t$–1, and $l_j^t = u_j^{t-1}+1$; $u_j^t = u_j$, where $j = j_{\max}$.

**Step 2.** Divide the constraint system **(3)** into $t$ constraint sub-systems:

$$\begin{aligned} &\boldsymbol{l_j \le x_j \le u_j; \quad j = 1,\ldots,n; j \neq j_{\max};} \\ &\boldsymbol{l_j^i \le x_j^i \le u_j^i; \quad j = j_{\max}; i = 1,\ldots,t.} \end{aligned} \qquad \textbf{(14)}$$

Each sub-region is defined by the constraint systems **(2)**, **(4)** and by one constraint sub-system from **(14)**.

**Step 3.** Perform *diversification* of the search process by going from one sub-region to another, generating the initial population at random with uniform distribution around the Tchebicheff center of the current sub-region. Then perform the *Intensification phase* described in *Wave – spreading strategy* (**Step 4.**) in each sub-region.

*C. Hybrid strategy*

This strategy consists in slicing the feasible domain in the way described in *Slicing strategy*. After that the search procedure performs a *Wave – spreading strategy* in each sub-region.

## III. ILLUSTRATIVE EXAMPLE

Let us consider the following two-dimensional example.
Five sub-areas in the feasible domain are defined:

A1 = $\{0 \le x_1, 0 \le x_2, 21x_1 + 20x_2 - 84000 \le 0\}$

A2 = $\{0 \le x_1, x_2 \le 10000, 0 < 21x_1 + 20x_2 - 84000, 7x_1 - 5x_2 + 15000 \le 0\}$

A3 = $\{x_1 \le 7200, 0 \le x_2, x_2 \le 10000, 0 < 21x_1 + 20x_2 - 84000, 0 < 7x_1 - 5x_2 + 15000\}$

A4 = $\{7200 < x_1, x_1 \le 10000, x_2 \le 5900, 0 \le x_2\}$

A5 = $\{7200 < x_1, x_1 \le 10000, 5900 < x_2, x_2 \le 10000\}$

The optimization problem is:

$$\text{Min } \boldsymbol{F}(\boldsymbol{x}) = \begin{cases} 10 + (x_1 - 2500)^2 + (x_2 - 1000)^2 & \text{if } (x_1, x_2) \in \text{A1}; \\ 7 + (x_1 - 1500)^2 + (x_2 - 7000)^2 & \text{if } (x_1, x_2) \in \text{A2}; \\ 12 + (x_1 - 6100)^2 + (x_2 - 3400)^2 & \text{if } (x_1, x_2) \in \text{A3}; \\ 11 + (x_1 - 9800)^2 + (x_2 - 2100)^2 & \text{if } (x_1, x_2) \in \text{A4}; \\ 3 + (x_1 - 8100)^2 + (x_2 - 9700)^2 & \text{if } (x_1, x_2) \in \text{A5}; \end{cases}$$

subject to: $0 \le x_1 \le 10000$; $0 \le x_2 \le 10000$;

This problem has five local optima – one per each sub-area: $x^{(1*)}$= (2500, 1000), $x^{(2*)}$= (1500, 7000), $x^{(3*)}$= (6100, 3400), $x^{(4*)}$= (9800, 2100), $x^{(5*)}$= (8100, 9700);

The corresponding objective function values are:
$F(x^{(1*)})$=10; $F(x^{(2*)})$ = 7; $F(x^{(3*)})$ = 12; $F(x^{(4*)})$ = 11; $F(x^{(5*)})$ = 3. Hence the global optimal solution is $x^{(5*)}$.

*Wave-spreading strategy*

Starting at the Tchebicheff center $i_{tch}$ = (5000, 5000) the simplex with vertices (5000, 5000), (5002.588, 5009.659) and (5009.659, 5002.588) is generated. The weight center of the simplex is $cs$ = (5004, 5004).

The population $P^{(0)}$ includes 10 points (individuals):
$x^{(1)}$ = (4994, 4994), $x^{(2)}$ = (5004, 4994), $x^{(3)}$ = (5014, 4994), $x^{(4)}$ = (4985, 5004), $x^{(5)}$ = (4005, 5004), $x^{(6)}$ = (5005, 5004), $x^{(7)}$ = (5015, 5004), $x^{(8)}$ = (4994, 5014), $x^{(9)}$ = (5004, 5014), $x^{(10)}$ = (5014, 5014);.

The corresponding objective function values are:
$F(x^{(1)})$ = 3764084, $F(x^{(2)})$ = 3742064, $F(x^{(3)})$ = 3720244, $F(x^{(4)})$ = 3816053, $F(x^{(5)})$ = 3793853, $F(x^{(6)})$ = 3771853, $F(x^{(7)})$ = 3750053, $F(x^{(8)})$ = 3828244, $F(x^{(9)})$ = 3806224, $F(x^{(10)})$ = 3784404;.

For $k$ = 10% we choose the best individual: $x^{(3)}$. The worst individuals are $x^{(8)}$, $x^{(4)}$ and $x^{(9)}$. The worst individuals are reflected towards $x^{(3)}$. Three new better individuals are generated and they replace the worst individuals $x^{(8)}$, $x^{(4)}$ and $x^{(9)}$. Proceeding in this way until no better individuals are generated, and then performing a simple local search, the procedure finds out the locally optimal solution $x^{(3*)}$ = (6100, 3400) with objective function value $F(x^{(3*)})$ = 12.

The generated simplex has the following rounded off vertices: $v^{(0)}$ = (5000, 5000), $v^{(1)}$ = (5003, 5010) and $v^{(2)}$ = (5010, 5003). We will consider the performance of this strategy along one exploring ray, say ($v^{(0} - cs$) = (–4, –4). Proceeding with β = 250 at $v^{(0)}$ the search procedure creates consecutively 5 initial populations around the calculated central solutions $cs^{(1)}$ = (4000, 4000), $cs^{(2)}$ = (3000, 3000), $cs^{(3)}$ = (2000, 2000), $cs^{(4)}$ = (1000, 1000), $cs^{(5)}$ = (0, 0). The last population reaches the boundaries of the feasible domain, so that this direction is explored. The third generated population around $cs^{(3)}$ = (2000, 2000) comes in the sub-area A1, so that the intensification phase finds out the optimum $F(x^{(1*)})$. The same is repeated with the fourth and fifth generated population.

Then the procedure explores the opposite direction, creating again five initial populations around the calculated central solutions $cs^{(6)}$ = (6000, 6000), $cs^{(7)}$ = (7000, 7000), $cs^{(8)}$ = (8000, 8000), $cs^{(9)}$ = (9000, 9000), $cs^{(10)}$ = (10000, 10000). The last three populations come in the sub-area A5, so that the intensification phase finds out the optimum $F(x^{(5*)})$.

Going on along the other two exploring rays in both possible directions the search procedure finds out also the optima $F(x^{(4*)})$ and $F(x^{(2*)})$.

During the exploration of whole feasible domain 31 intensification phases are performed.

*Slicing strategy*

We choose the component $\boldsymbol{x_1}$ as slicing component. The created sub-areas are: B1: $0 \le \boldsymbol{x_1} < 1000$; $0 \le \boldsymbol{x_2} < 10000$;

B2: $1000 \le \boldsymbol{x_1} < 2000$; $0 \le \boldsymbol{x_2} < 10000$; …

B10: $9000 \le \boldsymbol{x_1} \le 10000$; $0 \le \boldsymbol{x_2} < 10000$;

In each sub-area is generated an initial population randomly with uniform distribution. Performing the above described *intensification phase* in each sub-area the search procedure finds out all locally optimal solutions. To explore the whole feasible domain 10 intensification phases are performed. Some of them are more time-consuming than the intensification phases performed by *Wave-spreading strategy*.

*Hybrid strategy*

This strategy also finds out all possible local optima. Here the value of parameter β remains the same like in the *Wave-spreading strategy*, because the component $\boldsymbol{x_2}$ keeps its variation interval unchanged. This leads to great steps along the exploring rays and in some directions already the first generated initial population is infeasible. For this reason β should be reduced in half and this is repeated until the generating a feasible population becomes possible.

*Comparisons with other algorithms*

The described strategies are compared with scatter search and with particle swarm optimization. Starting with the initial population $P^{(0)}$ these two algorithms are able to find out only one locally optimal solution, and this is $x^{(3*)}$. Better performance is achieved in case the initial population is enough dispersed. The scatter search has better chances to find out the global optimal solution in case the limit of iterations is large. In this case it needs more than 1000 generations.

The presented problem may be solved by genetic algorithm, using niches. In this way all optima can be found but this performance will be essentially more time consuming than the proposed solution procedures.

## IV. Conclusions

The presented strategies for fast systematic diversification of the search have the following advantages:

- They systematically diversify the search process, avoiding in this manner the trap of local minima.
- They explore roughly the whole feasible domain and have good chances to find out the global optimal solution.
- The applying of local search technique at the end of the search process guarantees the good quality of the obtained solution.
- The proposed strategies have a better convergence to the global optimum in comparison to other global search algorithms, in which the search process does not perform a systematic diversification.
- They are simpler and don’t require large computer memory and complex memory organization in comparison to other global search strategies like Tabu search.
- They use populations with relatively small size. This makes them efficient in solving large dimensional problems.
- The proposed strategies are easy for computer programming implementation.

## 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] Colorni A., M. Dorigo and V. Maniezzo, “Positive Feedback as a Search Strategy”, Department of Electronics, Politechnico di Milano, Italy, Working paper 91-16, 1991.

[2] Colorni A., M. Dorigo and V. Maniezzo, “ Distributed Optimization by Ant Colonies”, in Proceedings of the *First European Conference on Artificial Life (ECAL-91)*, F. J. Varela and P. Bourgine eds., The MIT Press: Cambridge, MA, 1991, pp. 134-142.

[3] Colorni A., M. Dorigo and V. Maniezzo, „An Investigation of Some Properties of an Ant Algorithm“, in Parallel Problem Solving from Nature 2, R, Männer and B. Mandrieck eds., North-Holland: Amsterdam, 1992, pp. 509-520.

[4] 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.

[5] Eberhart R. C. and J. Kennedy, “A new optimizer using particle swarm theory”, Proceedings of the *Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.

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

[7] Glover F., “Heuristics for integer programming using surrogate constraints”, *Decision Sciences* 8, 1977, pp. 156-166.

[8] Glover F., “Future path for integer programming and links to artificial intelligence”, *Computers & Operations Research* 13, 1986, pp. 533-549.

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

[10] Glover F., “Tabu search and adaptive memory programming – advances, applications and challenges”, in *Advances in Metaheuristics, Optimization and Stochastic Modeling Technologies*, Barr, Helgason and Kennington eds.-, Kluwer: Boston, MA., 1997, pp. 1-75.

[11] Goldberg D. E. *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison Wesley, Reading, Mass, 1989.

[12] Guliashki, V., (2007) “A Hybrid Population Based Method Solving Convex Integer Optimization Problems”, In: *Proceedings of ICEST2007*, (Editor Prof. Dr. Mitrovski, C.), June 24 – 27, 2007, Ohrid, Macedonia, vol. I, pp. 249-252.

[13] Guliashki V., C. Korsemov, H. Toshev, (2010) “Elitism Based Evolutionary Algorithm for Discrete Optimization Problems”, In: *Proceedings of ICEST2010*, (Editor Prof. Dr. Mitrovski, C.), June 23 – 26, 2010, Ohrid, Macedonia, vol. I, ISBN: 978-9989-786-57-0, pp. 281-284.

[14] 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.

[15] Holland J. H., „*Adaptation in Natural and Artificial Systems*”, The University of Michigan Press: Ann Arbor, MI, 1975, MIT Press, 1992.

[16] Kennedy J. and R. C. Eberhart, “Particle Swarm Optimization”, Proceedings of *IEEE International Conference on Neural Networks*, Piscataway, N. J., 1995, pp. 1942-1948.

[17] Krusienski D. J. and Jenkins W. K. “Design and performance of adaptive systems, based on structured stochastic optimization strategies”, *IEEE Circuits and Systems*, Vol.5, No.1, first quarter 2005, pp.8-20.

[18] Nemhauser G. L. and Wolsey L. A. “*Integer and Combinatorial Optimization*”, Wiley, New York. 1988.

[19] Pirlot M., “Heuristic Search Methods”, Oper. Res. Designing Practical Solutions; Tutorial and Research Review Papers, Euro XIII/OR 36, The joint EURO/Oper. Res. Society Conference; Univ. of Strathclyde, Glasgow, 19-22, VII. 1994. pp. 180-201.

[20] Reklaitis G.V., Ravindran A., Ragsdell K. M., *Engineering Optimization. Methods and Applications*, John Wiley & Sons, 1983.

[21] 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.