HYBRID EVOLUTIONARY COMPUTATIONS
Application for Industry Investment Problem

Tadeusz Dyduch
Institute of Computing Science, AGH University of Science and Technology, Mickiewicza 30, Krakow, Poland

Keywords: Evolutionary computation, mixed linear programming, investments planning.

Abstract: The paper presents a special type of hybrid evolutionary computation, named by the authors Two-Level Adaptive Evolutionary Computation (TLAEC). The method consists in combination of evolutionary computation with deterministic optimization algorithms in a hierarchical system. Novelty of the method consists also in a new type of adaptation mechanism. Post optimal analysis of the lower level optimization task is utilized in order to modify probability distributions for new genotype generating. The paper presents an algorithm based on TLAEC method, solving a difficult optimization problem. A mathematical model of this problem assumes the form of mixed discrete-continuous programming. A concept of the algorithm is described in the paper and the proposed, new adaptation mechanism that is implemented in the algorithm is described in detail. The results of computation experiments as well as their analysis are also given.

1 INTRODUCTION

Evolutionary Computation (EC) is one of the most important and emerging technology of the recent times. Numerous scientific works, conferences, commercial software offers and academic handbooks were devoted to this technology. Evolutionary computations can be classified as iterative optimization methods based on a partially stochastic search through an optimization domain. Evolutionary computation has become the standard term that encompasses all of the evolutionary algorithms.

In order to improve convergence of evolutionary computation, evolutionary strategies have been devised, where the adaptation mechanism have been introduced (Michalewicz, 1996), (Eiben, 1999).

The paper presents a modification of the special type of adaptive evolutionary method, named two-level adaptive evolutionary computation (TLAEC), proposed and developed in (Dyduch T., 2004), (Dyduch T., Dudek-Dyduch E., 2005). Although the adaptation mechanism is embedded in the method, the adaptation differs from the classical evolutionary strategy. Novelty of the presented method consists in a new adaptation mechanism that is embedded in it, which utilises the data from local optimal analysis.

2 TWO-LEVEL ADAPTIVE EVO-LUTIONARY COMPUTATION

A modified TLAEC method consists of the following stages.

1. Generation of Population

The primary optimization task is transformed to the form suitable for hierarchical two-level algorithms. Thus, the set of searched variables are divided into two disjoint subsets. Similarly, the constraints are divided into two disjoint subsets.

The first subset of variables corresponds to a genotype. Values of the variables are computed on the higher level by means of random procedures using mutations and/or crossover operators. Only the first subset of constraints is taken into account here. These constraints refer only to the genotype. The values of the genotype constitute parameters for the lower level optimization task.

The remained variables (second subset) are computed on the lower level as a result of deterministic optimization procedure. Only the second subset of constraints is taken into account here. Because the variables can be calculated when the genotype is known, they correspond to a phenotype (or part of a phenotype).

Dyduch T. (2006).
HYBRID EVOLUTIONARY COMPUTATIONS - Application for Industry Investment Problem.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 195-198
DOI: 10.5220/0001205201950198

2. Choice of a Parent Individual or Parent Individuals.

A parent individual is chosen from the population on a basis of a fitness function, i.e. the probability that an individual will become a parent is proportional to the value of its fitness function.

3. Post Optimal Analysis and Modification of Probability Distributions.

After each iteration post optimal analysis is done. Its aim is to gather information about the lower level solution (or solutions computed in the earlier iterations). The gather information is utilized to modify probability distributions for genetic operators on the upper level. Thus the adaptation mechanism differs from the adaptation mechanism of evolutionary strategies (1+1), (μ+λ), (μ,λ) (Michalewicz Z., 1996).

TLAEC method can be applied to optimization tasks, which could be transformed to the form suitable for two-level optimization algorithms (Findeisen W., 1974).

Thus, the task: to find $\hat{x} \in X$ minimizing function *f(x),* where X is a subset of linear space X' can be replaced by:
to find the pair $(\hat{u},\hat{v}) \in U \times V = X$ such that

$$f(\hat{u},\hat{v}) = \min_{(u,v)\in U\times V} f(u,v) = \\ = \min_{u\in U}(\min_{v\in V(u)} f(u,v)) \tag{1}$$

where *V(u)* denotes a set of feasible vectors *v* determined at a fixed value of vector *u.* This task does not generally have an analytical solution unless it is trivial. Because of that it must be solved iteratively. Let u^i, v^i denote the value of vectors *u*, *v* computed in *i*-th iteration. Vector u^i is determined on a higher level while vector v^i on a lower level.

Let's present the searched variables in the evolutionary computation terms.

Individual: pair of vectors (u^i, v^i), representing a temporary point in the optimization domain (solution in the *i*-th iteration),

Genotype: vector u^i representing a temporary point in the subspace of optimization domain, searched at random with an evolutionary algorithm.

Phenotype: vector v^i representing a temporary point in the subspace of optimization domain, here computed by a deterministic, lower level optimization algorithm. In some cases a pair of vectors (u^i, v^i) may be a phenotype.

When the genotype u^i is established on the higher level then the lower level optimization task is of the form : to find v^i such that

$$Q^i = f(u^i,v^i) = \min_{v\in V(u^i)} f(u^i,v) \tag{2}$$

where Q^i denotes value of evaluation function of individual (u^i, v^i). This task should be effectively computable. The solution of problem (1), or point of its vicinity $(\hat{u},\hat{v})$ can be reached with an iterative procedure (3)

$$f(\hat{u},\hat{v}) = \min_{(u,v)\in U\times V} f(u,v) = \\ = \lim_{i\to\infty}(\min_{v\in V(u^i)} f(u^i,v)) \\ \text{where} \quad u^i = \mathrm{rnd}(u^{i-1},u^{i-2},...) \tag{3}$$

The function „rnd" is a random function, the probability distributions of which are tuned in the successive iterations.

3 TLAEC ALGORITHM APPLIED TO INDUSTRY INVESTMENT PROBLEM

The industry investment problem (Dyduch T., 2004) can be formally described as a mixed discrete-continuous linear optimization task. It is easy to notice that when the discrete variables u are fixed, the rest of the searched variables x, y, z can be computed by means of linear programming procedure. Because of that the vector u is assumed to be a genotype. Thus, on the upper level the vector u that satisfies (4) is randomly generated.

$$(u-u^0)g \le G \tag{4}$$

Let u = u^i where u^i is the value of u generated in i-th iteration. The rest of variables constitute a phenotype. The phenotype is computed on the lower level as a result of solving the following linear programming task (8)-(11): to minimize

$$c_2x - c_1y + c_3z + \lambda(u^i - u) \tag{5}$$

under constraints:

$$Az + x - y = b \tag{6}$$

$$z_k \le u_k h_k \quad \text{for } k=1,2,..m \tag{7}$$

$$x, y, z, u \ge 0 \tag{8}$$

The optimization task (5)-(8) is a Lagrangian relaxation of the primary optimization task.

Let us notice that after solving a linear programming task also the Lagrange multipliers λ for equalities $u = u^i$ are known. This information is a basis for the adaptation mechanism applied on the upper level evolutionary algorithm. In order to modify probability distributions for generation of new vectors u, two vectors e and d are defined. Both are used to approximate the most promising direction of vector u changes.

Let e_k, $k=1..m$, be defined by (9).

$$e_k = \lambda_k \frac{h_k}{g_k} \tag{9}$$

Let d_k, $k=1..m$, be defined by (10).

$$d_k = \left(1 - \frac{z_k}{h_k u_k}\right)\frac{g_k}{h_k} \tag{10}$$

The introduced adaptation mechanism has a heuristic character. The new vector u^{i+1} is generated as follows:

1. Probabilistic choice of types of production lines, numbers of which are to be reduced in the new arrangement u^{i+1}. The probability p_k that the number of production lines of k-th type will decrease is proportional to $q_k(u^i)$ where vector q is defined ($r \in (0,1)$ is a parameter):

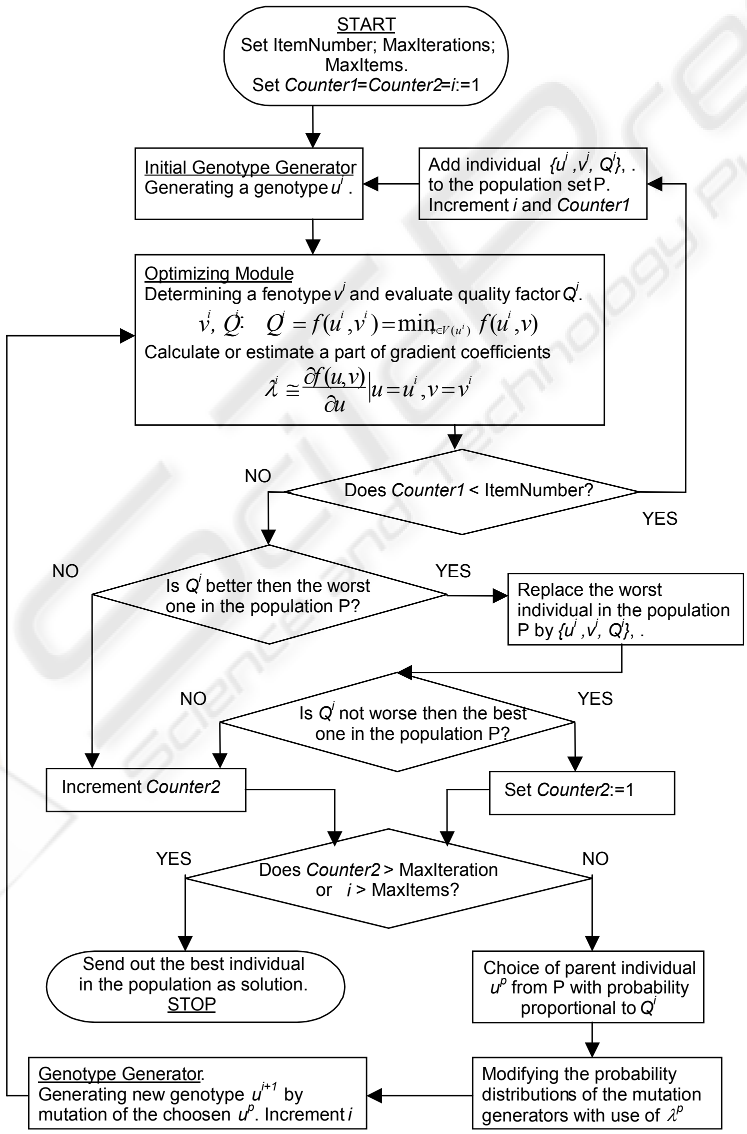

Figure 1: The simplified flow diagram of TLAEC method.

$$q(u^i) = \begin{cases} d(u^i) & if \quad i = 1 \\ (1-r)\cdot d(u^{i-1}) + r\cdot d(u^i) & if \quad i > 1 \end{cases} \quad (11)$$

$$p_k^i = \frac{q_k(u_k^i)}{\sum_{k=1}^{m} q_k(u_k^i)} \quad (12)$$

2. Probabilistic choice of new investments. The probability p_k that the number of production lines of k-th type will increase is proportional to $s_k(u^i)$. Vector u^{i+1} must belong to the neighborhood of u^i. ($r \in (0,1)$ is a parameter)

$$s(u^i) = \begin{cases} e(u^i) & if \quad i = 1 \\ (1-r)\cdot e(u^{i-1}) + r\cdot e(u^i) & if \quad i > 1 \end{cases} \quad (13)$$

$$p_k^i = \frac{s_k(u_k^i)}{\sum_{k=1}^{m} s_k(u_k^i)} \quad (14)$$

When the new vector u^{i+1} is generated then the new linear programming task is solved and value of function f, which plays a role of evaluation function in evolutionary algorithm, is calculated. Let this value is denoted as $f(u^{i+1})$.

If $f(u^{i+1})$ is better than the previous ones, then vector u^{i+1} replaces u^i. If value of evaluation function is not improved, the sampling is repeated.

Computations stop after a fixed number of iterations, when the best value of performance index does not change. The vectors $s(u)$, $q(u)$ and parameter r are used to improve the stability of the procedure.

4 RESULTS OF EXPERIMENTS

The presented algorithm has been tested for many exemplary tasks of different dimensions. The algorithm has been implemented in Matlab environment. The program uses a library procedure linprog(..) implemented in Matlab, that solves linear programming tasks. Procedure linprog(..) returns the value of the Lagrange multipliers.

Experiments conducted with the use of the proposed algorithm shown its high efficiency in searching for a global minimum of a discrete-continuous programming task.

In order to test ability of the algorithm for the real problems, the special generator of data has been designed and implemented. The semi-realistic investment problems of high dimensions were generated and tested. When parameters *r* and *MaxIteration* were fixed, the aim of further experiments was three-fold:

to test TLAEC algorithm convergence for different problems of the same size,

to test TLAEC algorithm efficiency for the different starting points when the data of problem are fixed,

to compare the efficiency of TLAEC algorithm with efficiency of TLEC algorithm, which has the adaptive ability withdrawn.

Table 1 shows some data collected during experiments on stability and accuracy of the TLAEC and TLEC methods, applied 8 times to each of 5 investment problems of 3x24x34 size. Q_0 are initial values of minimized criterion while Qopt are their optimal values.

Table 1

Q_0	3316,1	-88,4	-2860,3	5121,3	2981,2
TLEC	2856,8	-345,4	-3321,2	4767,0	2573,9
TLAEC	2562,5	-1362,7	-3671,0	4380,4	2360,7
Qopt	2491,1	-1396,0	-3671,0	4332,1	2334,6

The algorithm starting from different points found different but stable solutions, which fitness function values were similar. Contrary to it, TLEC algorithm (without the adaptive ability) computed unstable solutions.

REFERENCES

Dyduch T., 2004. Adaptive Evolutionary Computation of the Parametric Optimization Problem. *Lecture Notes In Artificial Intelligence* (3070), Springer-Verlag, Berlin , pp. 406-413

Dyduch T,.Dudek-Dyduch E., 2005. Two Level Adaptive Evolutionary Computation *Proc. of 23rd IASTED Int. Conf. Artificial Intelligence and Applications*, Insbruck, Austria pp. 42-47

Eiben A.E., Hinterding R., Michalewicz Z., 1999. Parameter Control in Evolutionary Algorithms, *IEEE Trans. On Evolutionary Computation*, vol.3, No 2, pp. 124-141

Findeisen W., 1974. *Multi-level control systems*. (in Polish) PWN Warszawa

Michalewicz Z., 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.