СДІТ

МЕТОДИ АНАЛІЗУ ТА УПРАВЛІННЯ
СИСТЕМАМИ В УМОВАХ
РИЗИКУ І НЕВИЗНАЧЕНОСТІ

УДК 519.85, 539.3
DOI: 10.20535/SRIT.2308-8893.2017.4.10

MULTIVARIATE CONVERGENCE-TARGETED OPERATOR FOR THE GENETIC ALGORITHM

O. SHADURA, A. PETRENKO, S. SVISTUNOV

Abstract. Optimization of complex particle transport simulation packages could be managed using genetic algorithms as a tuning instrument for learning statistics and behavior of multi-objective optimisation functions. Combination of genetic algorithm and unsupervised machine learning could significantly increase convergence of algorithm to true Pareto Front (PF). We tried to apply specific multivariate analysis operator that can be used in case of expensive fitness function evaluations, in order to speed-up the convergence of the "black-box" optimization problem. The results delivered in the article shows that current approach could be used for any type of genetic algorithm and deployed as a separate genetic operator.

Keywords: machine learning, genetic algorithm, Pareto Front, principle component analysis, transport particle simulations.

INTRODUCTION

A set of scientific researches that required data verification or generating big set of datasets like the studies in cosmology, high energy physics (HEP), biology and genetics, require the development of new approaches and methods for their efficient analysis on modern computer platforms.

In the point of the work on analyzing and optimizing the performance of the GeantV code [1], which is the prototype of the next-generation particle transport simulation software intended to succeed to Geant4 [2], which is the current golden standard in high energy physics (HEP) and beyond. Geant4 is a toolkit for simulation of the passage of particles through different kinds of matter, with application including high energy and nuclear physics, accelerator physics, medicine and space science and it is widely used in HEP experiments at the Large Hadron Collider (LHC) located at CERN (Geneva, Switzerland).

As a history, GeantV project had been started in 2013 with an R&D phase focused on optimal exploitation of instruction level parallelism for particle transport simulation both on CPU and on accelerators such as GPUs and Intel Xeon Phi [3].

GeantV is based on a specially developed vectorized computational solid geometry (CSG) modeler, which provides a set of optimized shape primitives and highly parallel geometry navigator and necessary *ray-tracing* functionality for the efficient propagation of particles through the target geometry [4].

The goal of GeantV project is to optimize the simulation algorithms to get maximum benefit from highly massive parallel SIMD/MIMD architectures [5] while finding the optimal point for factors focused on computational performance (floating-point performance, off-chip memory bandwidth, usage of cache and memory hierarchy and etc.). As a consequence, a large number of parameters have to be optimized and GeantV optimization task can be treated as a black-box problem.

DTLZ [6] set of benchmarks is covering cases in convex and non-convex, separable and non separable and multimodal functions with degenerate Pareto optimal fronts or disconnected Pareto optimal fronts, and disconnected Pareto optimal sets. These helps us to prototype of behavior of our algorithm in case of different set of realistic functions. The objective of this work is to observe whether, by using unsupervised machine learning, we can accelerate the process of finding a Pareto front.

Also using genetic algorithm together with machine learning approach we will try to analyze convergence and fixed points of evolutionary systems, trying to accelerate convergence rate of algorithm for "black-box" optimization. Before going to optimize Geant-V simulations, we will try to prototype algorithm's performance on a set of numerical DTLZ benchmarks [6] in order to accelerate their convergence to the true Pareto front via the integration of multi-objective search/optimisation (MOO) algorithms and unsupervised machine learning Principal Component Analisis (PCA) and kernel PCA.

GENETIC ALGORITHMS AS A DYNAMIC SYSTEM

Genetic algorithm is one of the widely used evolutionary algorithms for studies of various optimization problems, in the same time the theory of genetic algorithms (GA) was a subject of research for the last decades. The easiest model for studying GA is a simple model of genetic algorithm (SGA)[7], that could be used as a prototype of evolutionary system. This model is describing genetic algorithm (GA) as a dynamical system with accurate mathematical definitions and well studied in a literature.

In the model for description of GA as a Markov chain is used next definitions where states are populations and transition are operated by sets of genetic operators: selection, crossover and mutation [8]. Mutation ensures that the Markov chain is connected, therefore there is an unique equilibrium distribution over populations, the probability to produce a particular population in one generation depends only on the previous generation external influencing factors. This randomized process is described by a Markov chain, characterized by a transition matrix $\Theta_{\vec{q},\vec{p}}$ from the population $\vec{p}$ to the population $\vec{q}$.

Dynamical systems describe the evolution of individuals in the finite space of possible populations of fixed size m, where m is number of measurements during the experiment. While rethinking the genetic algorithms as a discrete dynamical system, many interesting mathematical objects like fixed points could be found. These objects are apparently not only generic for simple genetic algorithms, but also general for optimization problems. Let's briefly recall the results presented in [7] and establish the possible links with the task of optimizing our parameters.

We have a population of N different types of individuals in search sample space Ω. Each element of Ω can be thought of as a "unique individual" with a given fitness value defined by the cost function.

A population consists of m-subsets ($m = N$) each of which contains v_{α_i} of the α_i-type individual where $i = 1,...,m$ and defined by vector

$$\vec{b} = (b_{\alpha_1}, b_{\alpha_2}, ..., b_{\alpha_m})^t ,$$

where $\alpha_i \in \Omega$. The size of the population is $\overline{m} = \sum_{i=1}^{m} b_{\alpha_i}$.

We can redefine the population vector in the following form

$$\vec{p} = (p_1, p_2, ..., p_N)^t ,$$

where $p_\alpha (p_{\alpha_i} = b_{\alpha_1} / \overline{m})$ is the probability of occurrence α-th individual in the population vector $\vec{b}$.

In the mentioned before representation the repeated application of the genetic algorithm gives a sequence of vectors $\vec{p} \in \Lambda$ where

$$\Lambda = \{(p_1, p_2, ..., p_N)^t \in R^N \,|0 \le p_\alpha \le 1, \sum_{\alpha=1}^{N} p_\alpha = 1\}.$$

Λ is a set of admissible states for the populations. We can consider Λ as a $(N-1)$-dimensional simplex (a hyper-tetrahedron).

$G_\alpha(\vec{p})$ is a certain probability of producing individual α in the next generation if the previous population was $\vec{p}$ and define map $G : \Lambda \to \Lambda$, where $G(\vec{p}) = \prod_{\alpha \in \Omega} G_\alpha(\vec{p})$, and $G(\vec{p}) \in \Lambda$ could be considered as heuristic function. $G(\vec{p})$ is GA procedure on $\vec{p} \in \Lambda$ and the map G is actually the composition of three different maps: selection, mutation and crossover.

Let define genetic selection operator $F : \Lambda \to \Lambda$, where $F(\vec{p}) = \prod_{\alpha \in \Omega} F_\alpha(\vec{p})$ and the α-th component, $F_\alpha(\vec{p})$, represents the probability of the appearance of an individual of type α if the selection is applied to $\vec{p} \in \Lambda$. A selection operator chooses individuals from the current population using the cost function vector, $\vec{f} = \{f_\alpha\} \in R^N$, where $f_\alpha = f(\alpha)$, $\alpha \in \Omega$. This generic type of selection collects elements with probability proportional to their fitness. This corresponds to a heuristic function

$$F(\vec{p}) = \frac{\operatorname{diag}(f) \cdot \vec{p}}{\vec{f}^{\,t} \cdot \vec{p}},$$

where $\vec{p} \in \Lambda$ is the population vector, and $\operatorname{diag}(\vec{f})$ is the matrix with entries from $\vec{f}$ along the diagonal and zeros elsewhere.

The mutation operator $U : \Lambda \to \Lambda$ is an $N \times N$ real valued matrix with (α,β)-th entry $u_{\alpha,\beta} > 0$ for all α, β, and $u_{\alpha,\beta}$ represents the probability that individual $\beta \in \Omega$ mutates into $\alpha \in \Omega$. Then $(U \cdot \vec{p})_\alpha$ is the appearance of an individual of type α after applying a mutation to the population $\vec{p}$.

Crossover operator is defined $C:\Lambda \to \Lambda$ as

$$C(\vec{p}) = (\vec{p}^{\,t} \cdot \hat{C}_1 \cdot \vec{p}, ..., \vec{p}^{\,t} \cdot \hat{C}_N \cdot \vec{p}),$$

for $\vec{p} \in \Lambda$, where $\hat{C}_1, ..., \hat{C}_N$ is a sequence of symmetric non-negative $N \times N$ real-valued matrices. Here $\hat{C}_\alpha(\vec{p})$ represents the probability that an individual α is generated by applying the crossover to population $\vec{p}$.

Combining the selection, mutation and crossover maps we obtain the complete operator $\hat{G}$ for the genetic algorithm (GA map)

$$\hat{G}:\Lambda \to \Lambda, \quad \hat{G}(\vec{p}) = \hat{C} \circ \hat{U} \circ F(\vec{p}).$$

If we know the heuristic function G, we can write the transition matrix which is stochastic and based on the probability of transforming the population $\vec{p}$ into the population $\vec{q}$:

$$\Theta_{\vec{q},\vec{p}} = \overline{m}! \prod_{\alpha \in \Omega} \frac{\left(G_\alpha(\vec{p})\right)^{\overline{m}q_\alpha}}{\left(\overline{m}q_\alpha\right)!}, \qquad (1)$$

where $G_\alpha(\vec{p})$ is probability of producing individual α in the next generation and $\overline{M}q_\alpha$ is the number of copies of individuals α in the population $\vec{q}$, $\overline{m}$ is the size of the population.

As a brief review, the convergence properties of the simple genetic algorithm evolution schema was properly explored in the work [9]. While [10] showed that the convergence rate of the genetic algorithm is determined by the second largest eigenvalue of the transition matrix (1). The details of the proof was performed for diagonalizable transition matrices and transferred to matrices in Jordan normal form.

Another remarkable feature of the SGA is the presence of a rich structure of fixed and metastable points (for a detailed discussion see [8]).

Describing GA model through Markov chain representation we try to discover "hotspots" and find algorithmic or data patterns that could be used for improvement of the GA.

For the optimization of the GeantV simulation, we identify a set of optimization parameters important for the performance of particle transport simulations (e.g. the size of vector of particles to be transported or other significant design features) and build the data matrix $X_{\alpha,i} = \{(\vec{x}_\alpha)_i\} = \{\vec{x}_\alpha\}$ which contains the values of these parameters. In this matrix index i enumerates the tuning parameters ($i = 1, ..., n$) and index α enumerates the number of measurements of the parameters ($\alpha = 1, ..., M$ for M measurements), while in terms of GA index α enumerates M individuals and the population vector is constituted by $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_M)$

Recall the data and probabilistic sample representation. In the first case we can associate the vector based on the measurements of the i-th parameter $\vec{x}_i^{'} = \{(\vec{x}_i^{'})_\alpha\} = \{(\vec{x}_i^{'})_1, (\vec{x}_i^{'})_2, ..., (\vec{x}_i^{'})_M\}$, where the component $(\vec{x}_i^{'})_\alpha$ corresponds to the value of the i-th parameter in the α-th measurement with the population vector $(\vec{x}_1^{'}, \vec{x}_2^{'}, ..., \vec{x}_n^{'})$.

In the second case $P_i(x)$ be the probability distribution function of the measurements of the i-th parameter, with normalization

$$\int_{-\infty}^{\infty} dx P_i(x) = 1.$$

Using the previous strategy we associate the population vector

$$(\vec{p}_1, \vec{p}_2, ..., \vec{p}_n) \quad \text{with} \quad (\vec{x}_1', \vec{x}_2', ..., \vec{x}_n'),$$

where

$$\vec{p}_i = \{(\vec{p}_i)_1, (\vec{p}_i)_2, ..., (\vec{p}_i)_M\},$$

and the component $(\vec{p}_i)_\alpha$ is the probability to measure of the i-th parameter value $(\vec{x}_i')_\alpha$ in the α-th measurement.

One of the challenges of a Markov chains is to determine the evolution of the components along an appropriate direction for faster convergence to equilibrium. Using Principal Component Analysis (PCA) allows to check the genetic algorithm parameter sensitivity and the possible correlation between parameters. For this we introduce a operator that will be based on PCA and inverse PCA noise reduction operation for a genetic algorithm's optimisation of set of parameters.

We considered a possibility to improve the convergence rate by adding to a standard set of GA operator's (selection, mutation, crossing), a new operator $\hat{P}$ performing uncentered PCA on the GA populations. We will analyze the result of the implementation of the operator on the uncentered data matrix on standard GA performance benchmarks. From the experimental output we see that as in the SGA case [10], the convergence rate of genetic algorithm depends on the eigenvalues following the highest one, and for this reason the proposed operator $\hat{P}$ was applied on them.

UNCENTERED PCA AS A SVD APPROXIMATION FOR POPULATION DATA MATRIX

In PCA, we usually manipulate with centered data matrix in order to reduce a complex data set (in our case performance measurements data) to a lower dimensional set through analyzing the covariance matrix. Here is presented a way that a "sort of PCA" could be implemented on an uncentered data matrix. This is particularly convenient in the case of transformations of constrained data measurements using genetic algorithms, which are in our case highly constrained and multi-scaled performance parameters. As a basis of ideas about the connection between the centered and uncentered data matrix was used ideas from [11, 12].

PCA for centered data matrix and SVD

Let briefly recall PCA for the centered data matrix. The elements of the data matrix $\hat{X}$ of size $m \times n$ are described through m-samples of data from an n-dimensional space. In our case m is the number of individuals in the generation and n is the size of the individual (n is the dimension of vector of genes $\vec{x} = \{x_i\}(1 \le i \le n)$.

Let $\vec{x}_\alpha = \{(\vec{x}_\alpha)_i\}(1 \le \alpha \le m, 1 \le i \le n)$ is α -th individual of the population and

$$\hat{X} \ = \{X_{\alpha,i}\} = \{(\vec{x}_\alpha)_i\},$$

be a uncentered data matrix, size $m \times n$. Let us define the centered data matrix $\hat{Y}$:

$$\hat{Y} \ = \{Y_{\alpha,i}\} = \{X_{\alpha,i} - \mu_i\} = \{(\vec{y}_\alpha)_i\},$$

where μ_i is mean over M -individuals of i -th component of the gene:

$$\mu_i = \frac{1}{m}\sum_{\alpha=1}^{m} X_{\alpha,i}, \quad 1 \le i \le n, \quad \vec{\mu} = \{\mu_i\}.$$

The centered data matrix defines the covariance matrix $\hat{\Sigma}$:

$$\hat{\Sigma} = \frac{1}{m}\hat{Y}^t \cdot \hat{Y} \ = \{\Sigma_{i,j}\} = \frac{1}{m} Y^t_{i,\alpha} Y_{\alpha,j}$$

with the matrix multiplication repeated induces imply summation. Similarly for the uncentered data matrix we obtain the matrix of non-central second moments,

$$\hat{T} = \frac{1}{m}\hat{X}^t \cdot \hat{X} \ = \{T_{i,j}\} = \frac{1}{m} X^t_{i,\alpha} X_{\alpha,j} .$$

In standard PCA terms the first principal component (PC) $\vec{v}_1 = \{v_{\alpha,1}\}(\alpha = 1,...,m)$ is the linear combination

$$v_{\alpha,1} = \vec{y}^{\,t}_\alpha \cdot \vec{u}_1 = \sum_{i=1}^{n} Y_{\alpha,i} u_{i,1}, \quad \vec{u}^{\,t}_1 \cdot \vec{u}_1 = 1,$$

where the orthonormal n-dimensional vector $\vec{u}^{\,t}_1 = (u_{1,1},...,u_{n,1})^t$ is defined from condition that the first principal component has the largest variance

$$\sigma^2_{u1} = \frac{1}{m}\sum_{\alpha=1}^{m}\left[\sum_{i=1}^{n} Y_{\alpha,i} u_{i,1}\right]^2 .$$

The second principal component is the linear combination with the second largest variance and orthogonal to the first principal component, and so on.

To calculate PC, it is more comfortable to review the variational problem. For $\vec{v} \ = \{v_\alpha\} = \{Y_{\alpha,i} u_i\}$ we have

$$\mathrm{Var}(\vec{v} \ ) = \frac{1}{m}\vec{u}^{\,t} \cdot \hat{Y}^t \cdot \hat{Y} \ \cdot \vec{u} = \vec{u}^{\,t} \cdot \hat{\Sigma} \cdot \vec{u} \tag{2}$$

and the Lagrangian for the variational problem

$$\mathrm{L} = \vec{u}^{\,t} \cdot \hat{\Sigma} \cdot \vec{u} + \lambda(\vec{u}^{\,t}\vec{u} - 1).$$

The stationary condition is

$$\frac{\partial L}{\partial \vec{u}} = 2\hat{\Sigma} \cdot \vec{u} - 2\lambda\vec{u} = 0, \quad \hat{\Sigma} \cdot \vec{u} = \lambda\vec{u}.$$

This matrix equation has n solutions

$$\hat{\Sigma} \cdot \vec{u}_j = \lambda_j^{(c)} \vec{u}_j, \quad 1 \le j \le n,$$

where $\vec{u}_j$ are eigenvectors of $\hat{\Sigma}$ with the eigenvalue λ_j and $\vec{u}_j$ satisfy the orthonormality condition

$$\vec{u}_i^{\,t} \cdot \vec{u}_j = \delta_{i,j}, \quad 1 \le i, j \le n, \tag{3}$$

and

$$\vec{u}_j^{\,t} \cdot \hat{\Sigma} \cdot \vec{u}_j = \lambda_j. \tag{4}$$

Then the direction with maximum variance is the eigenvector with the largest eigenvalue. This procedure can be iterated to get the second largest variance projection (orthogonal to the first one), and so on.

From (2) it follows that the variance of the i-th centered principal component

$$\mathrm{Var}(\vec{v}_i) = \vec{u}_i^{\,t} \cdot \hat{\Sigma} \cdot \vec{u}_i = \lambda_i$$

and the covariance of the i-th and j-th centered principal components

$$\mathrm{Cov}(\vec{v}_i, \vec{v}_j) = \vec{u}_i^{\,t} \cdot \hat{\Sigma} \cdot \vec{u}_j = 0, \quad i \ne j.$$

Defining the matrix as $U_{i,j} = \vec{u}_j = (u_i)_j$, which consists from the eigenvectors of the covariance matrix $\hat{\Sigma}$. From (3) this matrix satisfies the orthogonality condition

$$U_{i,i'}^t U_{i',j} = \delta_{i,j}. \tag{5}$$

Then from (4) we have

$$\hat{U}^t \cdot \hat{\Sigma} \cdot \hat{U} = \hat{\Lambda} \ , \quad \Lambda_{i,j} = \lambda_i \delta_{i,j}, \tag{6}$$

Let define the matrix $V_{\alpha,j} = \{\vec{v}_j\} = \{(v_\alpha)_j\}$, where $\vec{v}_j$ — j-th centered principal compoent. Then

$$V_{\alpha,j} = Y_{\alpha,i} U_{i,j}, \quad 1 \le \alpha \le m, \tag{7}$$

and the first principal component $\vec{v}_1$

$$v_{\alpha,1} = V_{\alpha,1} = Y_{\alpha,i} U_{i,1} = \vec{y}_\alpha^{\,t} \cdot \vec{u}_1$$

if λ_1 is the largest eigenvalue. From (5), (6) we have

$$V_{i,\alpha}^t V_{\alpha,j} = m \Lambda_{i,j} = m \lambda_i \delta_{i,j}.$$

It is convenient to define the new matrix $\widetilde{V}_{\alpha,j}$

$$V_{\alpha,j} = \sqrt{m} \widetilde{V}_{\alpha,i} \Lambda_{i,j}^{1/2}, \quad \Lambda_{i,j}^{1/2} = \lambda_i^{1/2} \delta_{i,j},$$

which satify the condition

$$\widetilde{V}_{i,\alpha,}^t \widetilde{V}_{\alpha,j} = \delta_{i,j},$$

The matrix $\widetilde{V}_{\alpha,j} = \{(\widetilde{v}_j)_\alpha\}$ consists from eigenvectors $(\widetilde{v}_j)_\alpha$ of the matrix $\hat{K} = \hat{Y} \cdot Y^t$ of the size $m \times m$

$$K_{\alpha,\beta}(\widetilde{v}_j)_\beta = Y_{\alpha,k}Y^t_{k,\beta}(\widetilde{v}_j)_\beta = \lambda_j(\widetilde{v}_j)_\alpha.$$

with the same eigenvalues as in (4). From (7) we have

$$Y_{\alpha,i} = V_{\alpha,j}U^t_{j,i},$$

and obtain the Singular Value Decomposition (SVD) [13] for the centered data matrix

$$Y_{\alpha,i} = \sqrt{m}\widetilde{V}_{\alpha,i}\Lambda^{1/2}_{i,j}U^t_{j,i}. \qquad (8)$$

We suppose that the covariance matrix $\hat{\Sigma}$ has $(n-p)$ smallest eigenvalues $\lambda_j = 1, p+1 \le j \le n$. Then we can apply the dimension reduction and after the reverse PCA, we obtain the output data matrix $\overline{Y}_{\alpha,i}$:

$$\overline{Y}_{\alpha,i} = \sqrt{m}\widetilde{V}_{\alpha,i}\widetilde{\Lambda}^{1/2}_{i,j}U^t_{j,i} = \sqrt{m}\,(\lambda^{1/2}_1\widetilde{V}_{\alpha,1}U^t_{1,i} + \ldots + \lambda^{1/2}_p\widetilde{V}_{\alpha,p}U^t_{p,i}).. \qquad (9)$$

The approximation of matrix $Y_{\alpha,i}$ is the matrix $\overline{Y}_{\alpha,i}$ of reduced rank $m < n$. This transformation is also known as the discrete Karhunen-Loéve or the Hotelling transformation [16].

Using the SVD representation (8) and (9) for the centered data matrix we can calculate the mean square error (the standard error)

$$\eta_m = \frac{1}{nm}\sum_{\alpha=1}^{m}\sum_{i=1}^{n}(Y_{\alpha,i} - \overline{Y}_{\alpha,i})^2 =$$

$$= \frac{1}{nm}\sum_{\alpha=1}^{m}\sum_{i=1}^{n}\left(\sqrt{m}\sum_{k=p+1}^{n}\sqrt{\lambda_k}\widetilde{V}_{\alpha,k}U^t_{k,i}\right)^2 = \frac{1}{n}\sum_{k=p+1}^{n}\lambda_k.$$

Thus the minimum error is obtained if the covariance matrix $\hat{\Sigma}$ has $(n-p)$ smallest eigenvalues $\lambda_j, p+1 \le j \le n$ and the Hotelling transformation can be considered as the "*eigenvalue control parameter*" approximation.

PCA for uncentered data matrix and SVD

Next step is to apply the PCA method for the uncentered data matrix $\hat{X}$.

Vectors $\vec{w}_j (1 \le j \le n)$ are eigenvectors of the matrix of non-central second moments

$$\hat{T} = \frac{1}{m}\hat{X}^t \cdot \hat{X},$$

with the corresponding eigenvalues t_j

$$\hat{T} \cdot \vec{w}_j = t_j\vec{w}_j, \quad 1 \le j \le n, \qquad (10)$$

and satisfy the orthonormality condition

$$\vec{w}_i^{\,t} \cdot \vec{w}_j = \delta_{i,j}, \quad 1 \le i, j \le n.$$

Then

$$\vec{w}_j^{\,t} \cdot \hat{T} \cdot \vec{w}_j = t_j. \tag{11}$$

We define matrix $W_{i,j} = \{\vec{w}_j\} = (w_i)_j$ that satisfies the orthogonality condition

$$\hat{W}^t \cdot \hat{W} = \hat{I}. \tag{12}$$

From (11) we have

$$\hat{W}^t \cdot \hat{T} \cdot \hat{W} = \hat{\Delta} \ , \quad \Delta_{i,j} = t_i \delta_{i,j}. \tag{13}$$

Let $\vec{\theta}_j = \{(\theta_j)_\alpha\} = \{\vec{x}_\alpha^{\,t} \cdot \vec{w}_j\}(\alpha = 1,...,m)$ is j-th uncentered principal component. By analogy with (7) we define the matrix $\Theta_{\alpha,j} = \{\vec{\theta}_j\} = \{(\theta_j)_\alpha\}$

$$\Theta_{\alpha,j} = X_{\alpha,i} W_{i,j}, \quad 1 \le \alpha \le m, \tag{14}$$

which from (12) and (13) satisfy the condition

$$\Theta_{i,\alpha}^t \Theta_{\alpha,j} = m\Delta_{i,j} = mt_i\delta_{i,j}. \tag{15}$$

For the variance of j-th uncentered principal component we obtain

$$\mathrm{Var}(\vec{\theta}_j) = \sigma_{\theta,j}^2 = \frac{1}{m}\sum_{\alpha=1}^{m}\left[\sum_{i=1}^{n}(X_{\alpha,i} - \mu_i)W_{i,j}\right]^2 = t_j - \vec{\mu}^2 \cos^2(\vec{\mu}, \vec{w}_j),$$

For case of uncentered matrix we do not have a simple relationship between the eigenvalues t_j and the variance j-th uncentered principal component $(\sigma_{\theta,j})^2$ as for the centered data matrix. However, this property is not essential for the usage of the PCA method for the GA and in this case it is convenient to apply the "eigenvalue control parameter" approximation. The idea is to use the PCA method for the SVD representation of the uncentered data matrix.

We define the matrix $\widetilde{\Theta}_{\alpha,j}$

$$\Theta_{\alpha,j} = \sqrt{m}\,\widetilde{\Theta}_{\alpha,i}\Delta_{i,j}^{1/2}, \quad \Delta_{i,j}^{1/2} = t_i^{1/2}\delta_{i,j}. \tag{16}$$

From (15) and (16) we obtain:

$$\widetilde{\Theta}_{i,\alpha}^t \widetilde{\Theta}_{\alpha,j} = \delta_{i,j}.$$

Using (16), (15) and (11) it is not hard to show that $\widetilde{\Theta}_{\alpha,j} = \{\widetilde{\vec{\theta}}_j\}$ is the matrix of eigenvectors $(\widetilde{\theta}_j)_\alpha$ of the matrix $\widetilde{K} = \hat{X} \cdot \hat{X}^t$ of size $m \times m$

$$\widetilde{K}_{\alpha,\beta}(\widetilde{\theta}_j)_\beta = X_{\alpha,k} X_{k,\beta}^t (\widetilde{\theta}_j)_\beta = t_j (\widetilde{\theta}_j)_\alpha.$$

From (14) we obtain the representation for the uncentered data matrix

$$X_{\alpha,i} = \Theta_{\alpha,j} W_{j,i}^t,$$

from which we get the SVD representation for the uncentered data matrix

$$X_{\alpha,i} = \sqrt{m}\,\widetilde{\Theta}_{\alpha,k}\,\Delta^{1/2}_{k,j}W^{t}_{j,i}.$$

If the matrix of non-central second moments $\hat{T}$ has $(n-q)$ smallest eigenvalues $t_j = 1, q+1 \le j \le n$ we can use the “eigenvalue control parameter” approximation and get the output data matrix $\widetilde{X}_{\alpha,j}$ of rang q

$$\widetilde{X}_{\alpha,i} = \sqrt{m}\,\widetilde{\Theta}_{\alpha,k}\,\widetilde{\Delta}^{1/2}_{k,j}W^{t}_{j,i} = \sqrt{m}\,(t_1^{1/2}\widetilde{\Theta}_{\alpha,1}W^{t}_{1,i} + \ldots + t_q^{1/2}\widetilde{\Theta}_{\alpha,q}W^{t}_{q,i}), \qquad (17)$$

where the eigenvalue matrix $\widetilde{\Delta}_{k,j}$ has rang q $(t_{q+1} = t_{q+2} = \ldots = t_n = 0)$.

We approximate $X_{\alpha,i}$ with rank n by the matrix $\widetilde{X}_{\alpha,i}$ which has rank q. This is the analog of the Hotelling transformation.

Using the SVD representation we can estimate the mean square error η_q for this approximation:

$$\eta_q = \frac{1}{mn}\sum_{\alpha=1}^{m}\sum_{i=1}^{n}(X_{\alpha,i} - \widetilde{X}_{\alpha,i})^2 =$$

$$= \frac{1}{mn}\sum_{\alpha=1}^{m}\sum_{i=1}^{n}\left(\sqrt{m}\sum_{k=q+1}^{n}\sqrt{t_k}\,\widetilde{\Theta}_{\alpha,k}W^{t}_{k,i}\right)^2 = \frac{1}{n}\sum_{k=q+1}^{n} t_k.$$

The minimum error is obtained in the case if the matrix of non-central second moments $\hat{T}$ has $(n-q)$ smallest eigenvalues $t_j, q+1 \le j \le n$.

The second case we can get this approximation using a projector $\hat{P}$, which projects the data matrix $\hat{X}$ onto the subspace spanned by the principal axes with largest eigenvalues t_j, $1 \le j \le q$. Let define matrix $\widetilde{W}_{i,k'} = \{\vec{w}_{k'}\} = (w_{k'})_i$ $(1 \le k' \le q)$ of the size $n \times q$. This matrix consists from the first q largest eigenvectors $\vec{w}_{k'}$ (20).

The projector $\hat{P}^{(1,q)}$ is defined the following way

$$\hat{P}^{(1,q)}_{i,j} = \widetilde{W}_{i,k'}\widetilde{W}^{t}_{k',j}, \quad \hat{P}^{(1,q)} \cdot \hat{P}^{(1,q)} = \hat{P}^{(1,q)}.$$

Then it is not hard to show that the approximation $\widetilde{X}_{\alpha,i}$ in (17) can be written using the projector $\hat{P}^{(1,q)}$

$$\widetilde{X}_{\alpha,j} = X_{\alpha,i}\hat{P}^{(1,q)}_{i,j} = t_1\Theta_{\alpha,1}W^{t}_{1,j} + \ldots + t_q\Theta_{\alpha,q}W^{t}_{q,j}.$$

Analysis of eigenvalues in SVD representation of the uncentered input data matrix $X_{\alpha,i}$ used as population in GA can significantly accelerate the processes of finding the Pareto front for the MOP. We verified this hypothesis for the standard GA test problems [6].

Eigenvectors with the largest eigenvalues likely determine the subspace of solutions of the MOP in which lies the Pareto front. Using an iterative procedure for uncentered data matrix from MOP we can faster converge to the optimal solution subspace.

PCA-based genetic operator $G_P(\vec{p}) = \hat{P} \circ \hat{C} \circ \hat{U} \circ F(\vec{p})$ allows to check the genetic algorithms parameter sensitivity and the possible correlation between parameters. We introduced a new algorithmic step applied to generation modification step that performs data transformation based on PCA and inverse PCA noise reduction operation the set of parameters used for GA.

EVOLUTIONARY SCHEMA PERFORMANCE IMPROVEMENT FOR NSGA-II

In the article we propose to modify NSGA-II [14] as one of the most common GAs with specific operator shown on figure 1 that can be regarded as a denoising factor for faster approximation and convergence to the true Pareto front consisting of ideal individuals, we can apply orthogonal transformation to be able to discover strong patterns in data set. NSGA-II features fast non-dominance sorting procedure of population and preservation of a good convergence rate to the optimal Pareto set and it preserves a spread of best individuals is using a diversity preservation operation called crowding distance and non-dominated ranking procedure. In case of NSGA-III [15] as a evolution of NSGA-II has more specific algorithm schema based on reference point's selection procedure.

On a Fig. 1 is shown how was integrated operator performing UPCA in the algorithm. It is particularly important to notice that authors tested more combinations, and case of schema described on Fig. 1, we got maximum of benefits in speedup and algorithm convergence caused by increased size of population matrix used for tournament selection pool.

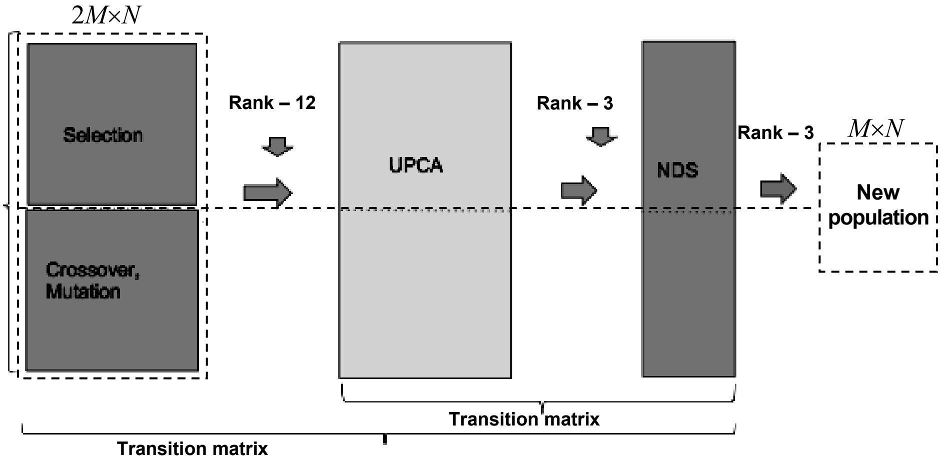
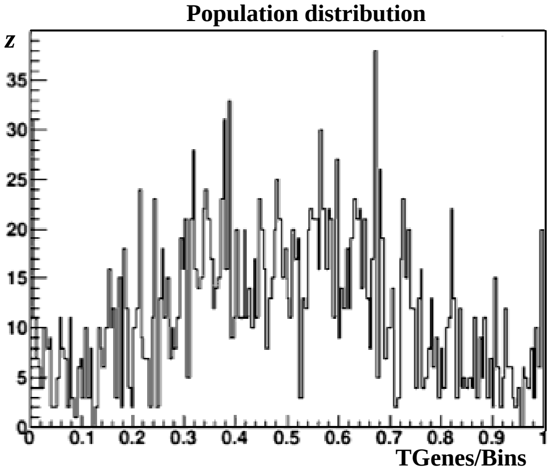


Fig. 1. New schema of algorithm

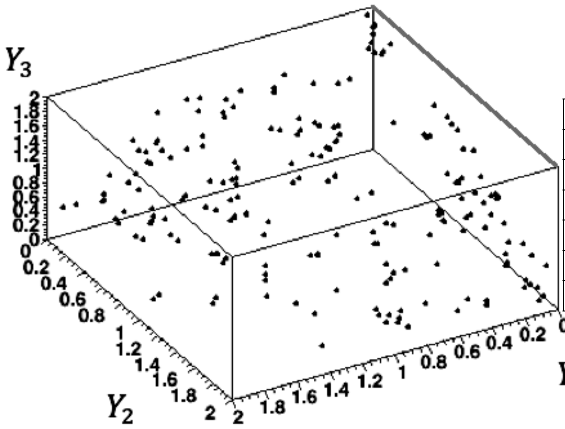
RESULTS OF RUNNING DTLZ BENCHMARKS

The DTLZ problems [6] are a set of numerical MOP benchmarks that are used for comparing and validating results from different GA algorithms. We present results of the DTLZ benchmarks [6] for NSGA-II and NSGA-II with PCA noise cleanup operator. We recognized that currently NSGA-III is outperforming NSGA-II but here results are provided as a proof of concept. On Figure 2, 5 are presented the parameter distribution (mean and standard deviation values) and cost function values behavior depending on used algorithms.



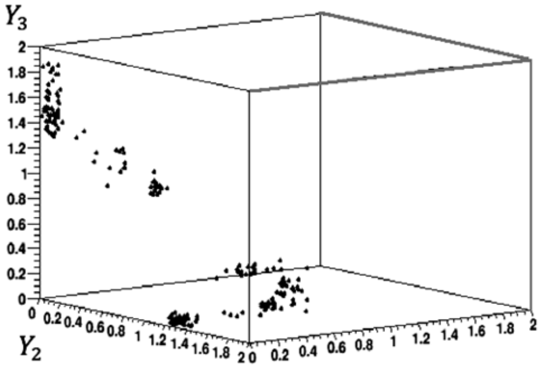
PopDist10	
Entries	2400
Mean	0,4911
Std deviation	0,2422

Fig. 2. Population distribution on 10th generation - NSGA-II - DTLZ2



	DTLZ2 NSGA–2
Entries	200
Mean Y_1	0,7881
Mean Y_2	0,824
Mean Y_3	0,754
Std deviation Y_1	0,563
Std deviation Y_2	0,5814
Std deviation Y_3	0,4796

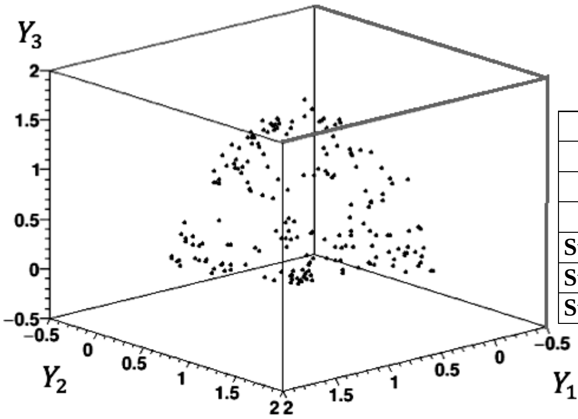
Fig. 3. Pareto Front on 10th generation of NSGA-II - DTLZ2



	DTLZ4 NSGA–2
Entries	200
Mean Y_1	0,6303
Mean Y_2	0,4336
Mean Y_3	0,6171
Std deviation Y_1	0,5334
Std deviation Y_2	0,5928
Std deviation Y_3	0,7106

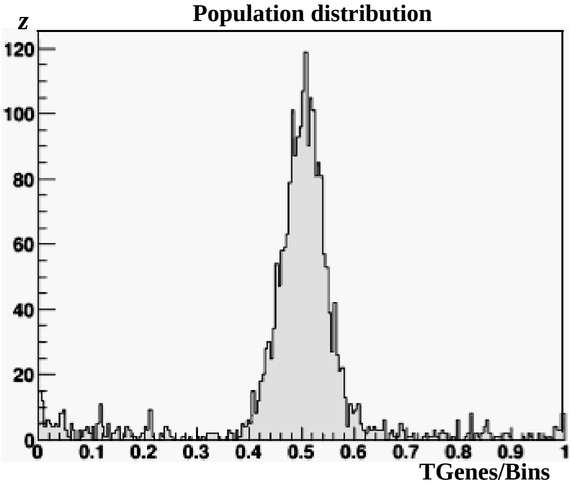
Fig. 4. Pareto Front on 40th generation of NSGA-II - DTLZ4

Comparing Fig. 5, 6 and Fig. 7 where was applied noise-removing procedure and Figure 2, 3 and Figure 4 where was not, we can observe faster convergence to the ideal values of the parameters in the first case. Fig. 7 and Fig. 6 show the first approach to Pareto front in combination with correct set of parameters.



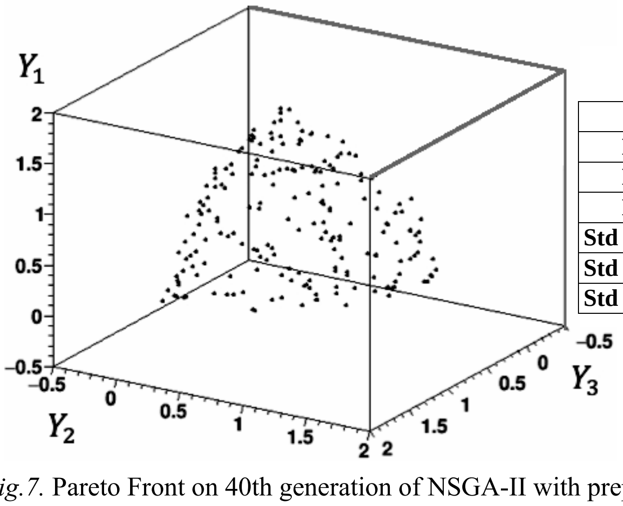
	DTLZ2 NSGA2UPSA
Entries	200
Mean Y_1	0,615
Mean Y_2	0,6235
Mean Y_3	0,5528
Std deviation Y_1	0,384
Std deviation Y_2	0,4275
Std deviation Y_3	0,4366

Fig. 5. Population distribution on 10th generation — NSGA-II with preprocessing of data -DTLZ2



PopDist10	
Entries	2400
Mean	0,487
Std deviation	0,4

Fig. 6. Pareto Front on 10th generation of NSGA-II with preprocessing of data - DTLZ2



	DTLZ4 NSCA2UPCA
Entries	200
Mean Y_1	0,6407
Mean Y_2	0,5633
Mean Y_3	0,6273
Std deviation Y_1	0,3989
Std deviation Y_2	0,4217
Std deviation Y_3	0,4117

Fig.7. Pareto Front on 40th generation of NSGA-II with preprocessing of data - DTLZ4

The next steps of our work will be to agree our concept with the existence of fixed points in dynamical systems, to re-evaluate a possible speedup comparing to other algorithms together with the “black-box” benchmarks [17] and port a new algorithm as a part of the optimization framework for GeantV particle transport simulations code.

CONCLUSIONS

In this work we tried to explore the possibility to combine genetic algorithms and unsupervised machine learning (PCA/UPCA/SVD/KPCA) to obtain a powerful combination that speedup existing GA algorithms. Usage of this algorithm for performance optimization of simulation of particle physics with clearly give benefit in finding optimal value with smaller number evolutions with costly fitness function. Next step of work is to implement this algorithm as a part of optimization routine in Geant-V project and test a benefits in full scale mode system.

REFERENCES

1. *Amadio G.* GeantV: from CPU to accelerators / G. Amadio, A. Ananya, J. Apostolakis, A. Arora, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F. Carminati, L. Duhem, D. Elvira, A. Gheata, M. Gheata, I. Goulas, R. Iope, S. Jun, G. Lima, A. Mohanty, T. Nikitina, M. Novak, W. Pokorski, A. Ribon, R. Sehgal, O. Shadura, S. Vallecorsa, S. Wenzel, Y. Zhang // Journal of Physics: Conference Series. — 2016. — Vol. 762, N 1. — P. 012019.
2. *Allison J.* Geant4 developments and applications / J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce Dubois, M. Asai, G. Barrand, R. Capra and others // IEEE Transactions on Nuclear Science. — 2006. — Vol. 53, N 1. — P. 270–278.
3. *Amadio G.* The GeantV project: preparing the future of simulation / G. Amadio, J. Apostolakis, M.Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun and others // Journal of Physics: Conference Series. — 2015. — Vol. 664, N 7. — P. 072006.
4. *Apostolakis J.* Towards a high performance geometry library for particle-detector simulations / J.Apostolakis, M.Bandieramonte, G. Bitzes, R. Brun, P. Canal, F. Carminati and others // Journal of Physics: Conference Series. — 2015. — Vol. 608, N 1. — P. 012023.
5. *Apostolakis J.* Adaptive track scheduling to optimize concurrency and vectorization in GeantV / J.Apostolakis, M.Bandieramonte, G. Bitzes, R. Brun, P. Canal, F Carminati and others // Journal of Physics: Conference Series. — 2015. — Vol. 608, N 1. — P. 012003.
6. *Deb K.* Scalable Test Problems for Evolutionary Multi-Objective Optimization, Evolutionary Multiobjective Optimization: Theoretical Advances and Applications / K. Deb, L. Thiele, M. Laumanns, E. Zitzler // Advanced Information and Knowledge Processing - Evolutionary Multiobjective Optimization - Springer London. — 2015. — P. 105–145.
7. *Vose M.* The Simple Genetic Algorithm: Foundations and Theory / M. Vose // MIT Press. Cambridge. — 1999. — 251 p.
8. *Rowe J.E.* Genetic algorithm theory / J.E. Rowe // Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO2012). — New York, USA. — 2012. — P. 917–940.

9. *Rudolph G.* Convergence properties of evolutionary algorithms / G. Rudolph // Kovac, Hamburg. — 1997. — 286 p.
10. *Schmitt F.* On the Importance of the Second Largest Eigenvalue on the Convergence Rate of Genetic Algorithms / F. Schmitt, F. Rothlauf // Proceedings of the 14th Symposium on Reliable Distributed Systems. — 2001. — P. 559–564.
11. *Cadima J.* On Relationships between Uncentered and Column-centered Principal Component Analysis / J. Cadima, I. Jolliffe // Pakistan Journal of Statistics. — 2009. — Vol. 25(4). — P. 473–503.
12. *Honeine P.* An eigenanalysis of data centering in machine learning / P. Honeine // Preprint ArXiV ID: 1407.2904. — 2014. — 14 p.
13. *Baker K.* Singular value decomposition tutorial / K. Baker. — Available at: https://datajobs.com/data-science-repo/SVD-Tutorial-%5BKirk-Baker%5D.pdf
14. *Deb K.* A fast and elitist multiobjective genetic algorithm: NSGA-II / K. Deb, A. Pratap, S. Agarwal, T. Meyarivan // IEEE Transactions on Evolutionary Computation. — 2002. — Vol. 6. — P.182–197.
15. *Haitham S.* U-NSGA-III: A unified evolutionary algorithm for single, multiple, and many-objective optimization / S. Haitham, K. Deb // Lecture Notes in Computer Science. — 2015. — Vol. 6019. — P. 34–49.
16. *Annadurai S.* Fundamentals of digital image processing / S. Annadurai // Pearson Education India. — 2007. — 440 p.
17. *Hansen N.* COCO: A platform for Comparing Continuous Optimizers in a Black-Box Setting / N. Hansen, A. Auger, O. Mersmann, T. Tuar, D. Brockhoff // Preprint ArXiV ID: 1603.08785. — 2016. — 10 p.

Received 19.01.2017

From the Editorial Board: the article corresponds completely to submitted manuscript.