ON THE UTILITY OF LINEAR TRANSFORMATIONS FOR POPULATION-BASED OPTIMIZATION ALGORITHMS

Petr Pošík

Czech Technical University, Faculty of Electrical Engineering, Department of Cybernetics
Technická 2, Prague 6
e-mail: posik@labe.felk.cvut.cz

Abstract: Many population-based real-valued optimization algorithms assume statistical independence of individual parts of solution. This assumption is only seldom fulfilled. In real domain, some coordinate transformations can be applied to reduce the dependency among variables making the optimization problem easier to solve. This article reviews two common linear transformations, principal and independent component analysis (PCA, ICA). Although ICA should work for our purposes better, it is shown that there are cases when PCA results in a better performance.

Keywords: optimization, parameter estimation

1. INTRODUCTION

Optimization problems can be found in many areas of human activities. In real domain, the parametric optimization presents the typical one. It arises e.g. when fitting a parametric model to the data set at hand. Population-based optimization algorithms (such as evolutionary algorithms, EAs) were found very successful solving similar problems. They do not need any information about the inner structure of the problem and they are able to escape from local optima.

However, many of these algorithms use such mechanisms which assume the individual features of the potential solutions to be statistically independent of each other. This unrealistic assumption is often broken. Nevertheless, there are some possibilities how to reduce the relationships among variables. This possibility is offered to us by certain coordinate transformations. The individuals can live in an environment where their lives are easier, i.e. where it requires less effort from the algorithm to breed new and better individuals than it would require when evolving in the original environment where the competing for survival takes place. This principle is not new in EAs, it can be thought of as a kind of genotype-phenotype mapping with the exception that this mapping is here adapted on purpose.

Inside many EAs various coordinate transformations can be used. The main requirement on such a transformation should be its *reversibility*. We need the 'forward' part of the transformation to reduce the dependencies among variables so that the evolution can take place in the transformed space. The 'backward' part of transformation (or the inverse transformation) is to transform newly generated offsprings back into the original space in order to be evaluated (the objective function is defined only in the original space).

Section 2 of this article describes two well-known linear coordinate transformations, principal components analysis (PCA) and independent component analysis (ICA). Their use in EAs in not new. In (Hansen and Ostermeier, 2001), the PCA is implicitly used in a mutative evolutionary strategy with covariance matrix adaptation. EA with ICA (similar to the one used in this article) was described in (Zhang *et al.*, 2000), and (Cho and Zhang, 2004) used even a finite mixture of ICA models. Although the name suggests that ICA should be more appropriate kind of transformation, the experiments in Sec. 3 show that the choice is not that easy and deserves a special care. Section 4 concludes the paper and suggests some explanations of the observed phenomenons and proposes directions for further research.

2. POPULATION PREPROCESSING

As already stated, the dependencies in the data set can be reduced by population preprocessing. In this section, two linear methods are described. They can be used with any real-valued population based algorithm. In this article, an univariate marginal distribution algorithm (UMDA) is used. This algorithm assumes the independence of individual variables. For each variable it builds a marginal probabilistic model, the joint probability is then given by a product of marginal histograms and new individuals are produced by sampling from this model. The algorithm used herein uses so-called equi-height histograms (see e.g. (Pošík, 2003)). The only difference is that before creating the probabilistic model of the population, the data points are preprocessed by PCA or ICA, and, of course, after creating new offsprings they are transformed back by the inverse transformation.

2.1 Principal Components Analysis

Perhaps the best known linear coordinate transformation is the so-called *principal components analysis* (PCA). It is used mainly for dimensionality reduction in multivariate analysis and its applications range from data compression, through image processing, to visualisation, pattern recognition, or time series prediction.

The PCA is most commonly defined as a linear projection which maximizes the variance in the projected space (Hotelling, 1933). We have to find such an orthogonal coordinate system, in which the variance of the data is maximized along the axes. It can be easily accomplished by performing the eigendecomposition of the data sample covariance matrix. One additional property of PCA is worth mentioning: among all orthogonal linear projections, the PCA projection minimizes the squared reconstruction error.

To formalize the computations, let us denote the set of centered data points at hand (the population) as $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, ...\mathbf{x}^N)$. The population matrix $\mathbf{X}$ is of size $D \times N$, where D is the dimension of the input space and N is the population size. The covariance matrix of size $D \times D$ is given by

$$\mathbf{C} = \frac{1}{N}\mathbf{X}\mathbf{X}^T. \tag{1}$$

If we find the eigendecomposition of this matrix, we find the linear transformation which decorrelates the components of individuals, i.e. we need to compute a diagonal matrix λ of order D and symmetric matrix $\mathbf{V}$ of size $D \times D$ such that the condition $\lambda\mathbf{C} = \mathbf{V}\mathbf{C}$ holds. Matrices λ and $\mathbf{V}$ contain the eigenvalues and eigenvectors of the covariance matrix $\mathbf{C}$, respectively. Then, the transformation

$$\mathbf{Y} = \mathbf{V} \times \mathbf{X} \tag{2}$$

rotates the coordinate system of the population matrix $\mathbf{X}$ in such a way that the coordinates of individual data points in matrix $\mathbf{Y}$ are not correlated. The inverse transformation can be done simply by inverting the eigenvectors matrix $\mathbf{V}$, i.e.

$$\mathbf{X} = \mathbf{V}^{-1} \times \mathbf{Y}. \tag{3}$$

2.2 Independent Components Analysis

The PCA described in Sec. 2.1 can be also described in these terms: it is a linear transformation which minimizes the correlations [1] (the 'first-order' dependencies) among variables. It would be very nice to have similar algorithm for creating a linear transformation minimizing a compound criterion which would take into account also the 'higher-order' dependencies among variables.

The *independent components analysis* (ICA) (see e.g. (Hyvärinen, 1999)) is a rather recent data analysis technique. Its primary aim is to find such a linear transformation which makes the transformed variables as independent of each other as possible. This goal makes the ICA a very appealing preprocessing technique for the use in EAs.

2.2.1. ICA Basics We can define the ICA in several ways. If we select the mutual information (MI) as the measure of dependency, we can define the ICA as a process of finding such a linear transformation which minimizes the MI.

[1] In fact, PCA not only minimizes the correlations, but puts them away completely.

Unfortunately, direct minimization of MI over possible linear transformations would require to estimate the density functions which is very hard (very uncertain and often very time consuming) work. In (Hyvärinen and Oja, 2000), it is shown that:

"...ICA estimation by minimization of mutual information is equivalent to maximizing the sum of non-gaussianities of the estimates, when the estimates are constrained to be uncorrelated. ..."

From the above citation it is clear that due to the constraint of uncorrelated projections, the ICA need not estimate the joint probability density — the problem is very simplified and can be solved just by searching for 1-dimensional subspaces with the greatest measures of non-gaussianities of the projections. The general measure of non-gaussianity is usually the *negentropy*. If $H(X)$ is the differential entropy of a random variable X, then the negentropy of a random variable $J(X)$ is defined as

$$J(X) = H(X_{Gauss}) - H(X), \tag{4}$$

where the X_{Gauss} is a random variable with normal distribution. It is well known fact, that a gaussian variable has the largest entropy among all random variables with equal variance, thus the negentropy is always nonnegative and zero for normal distribution, so that it can be used as a measure of non-gaussianity. Nevertheless, it still remains only theoretical measure because one still has to estimate the probability distributions.

In practice, we have to resort to some approximations of negentropy. Some of the approximations can be found in (Hyvärinen and Oja, 2000). They usually emphasize the differences from the normal distribution, i.e. they return greater values if the empirical distribution is spiky, multimodal or has heavy tails. Finding the most independent directions is judged only by the shape of the 1D projection distributions. During the EA run it can easily happen that the distribution in some direction has a more non-gaussian shape then the distribution in a direction which is really independent. This can mislead the EA that uses ICA as a preprocessing step.

2.2.2. ICA Features The fact that ICA can be performed by searching for the most non-gaussian directions is appealing also from an empirical point of view. The most non-gaussian projections (i.e. spiky, multimodal, clustered, etc.) are the most interesting ones. This principle was also used in a statistical method for visualization of the most interesting views of the data — in *projection pursuit* (Friedman, 1987).

In the ICA model, only one of the independent components can have a normal distribution. Greater number of gaussian variables would make the ICA model unidentifiable because all rotations of n-dimensional gaussian random cloud of data points are in fact equivalent.

The outcome of PCA and ICA can be visualized by means of contour plots of the linear components extracted from the data. Individual contours are parallel to each other and contours of the first component are perpendicular to the contour lines of the second component. An example of the difference between principal and independent components can be seen in Fig. 1. This data resemble the population in certain phases of evolution of the 2D Griewangk function. In this case, the PCA actually discovers the most independent components, while the ICA (operating on the basis of maximizing the nongaussianity of the projections) almost does not rotate the data. This is an example of a data set when the ICA fails to find the independent components and PCA gives better result (see Sec. 3.4). This will be also seen in the results of experiments in the next section.

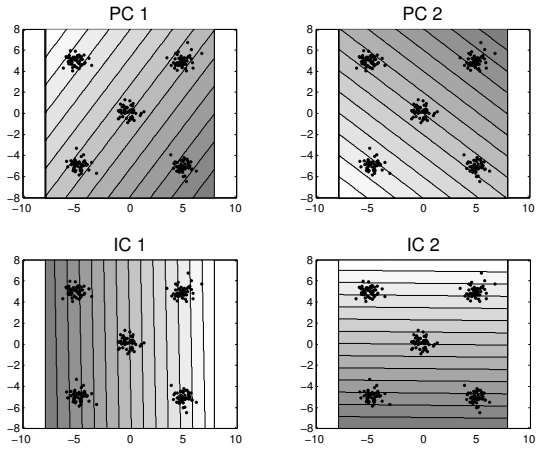

Fig. 1. Principal and independent components for a data set similar to a population when evolving 2D Griewangk function.

3. EXPERIMENTS WITH PCA AND ICA

Several experiments were carried out to demonstrate the influence of population preprocessing using PCA and ICA. The 2- and 10-dimensional Griewangk function with the 2- and 20-dimensional Two Peaks function were selected for this demonstration. The Griewangk function is non-separable function with one global optimum at point $(0, \ldots, 0)$ and several local optima surrounding and hiding the global one. The complexity of this test function decreases with dimensionality. The Two Peaks function is completely separable function with one global optimum at point $(1, \ldots, 1)$ and many local optima with much greater basins of attraction.

3.1 Evolutionary Model

For the comparison, the UMDA with marginal histogram models (see (Pošík, 2003)) was used. This kind of algorithm assumes the individual variables to be statistically independent of each other. In all experiments the following evolutionary model was used.

(1) Initialize and evaluate the population.
(2) Based on the current population, perform the PCA or ICA of the parents.
(3) Model the transformed parents by marginal histograms.
(4) Sample N new offsprings from the model.
(5) Transform the new offsprings to the original space using inverse PCA or ICA transformation.
(6) Evaluate them.
(7) Join the old and the new population to get a data pool of size $2N$.
(8) Use the truncation selection to select the better half of the data points (returning the population size back to N).
(9) If the termination criteria are not met, go to Step 2.

This cycle was repeated until the number of function evaluations exceeded 50,000. The first set of experiments was carried out without any preprocessing. The second and third set of experiments uses the PCA and ICA preprocessing, respectively.

3.2 Monitored Statistics

Population sizes of 20, 50, 100, 200, 400, 600, and 800 individuals were used. Each experiment was repeated 20 times with the same settings. During all experiments several measures of the efficiency were tracked:

- *BSF* (Best-so-far fitness). Average fitness of the best individual after 50,000 evaluations in all 20 runs.
- *StdevBSF*. The standard deviation of the best fitness after 50,000 evaluations in all 20 runs.
- *Found0.1* (*Found0.01*, *Found0.001*). As the evolution progresses, it is checked how many times (out of the 20 runs) the best solution is in the '0.1 neighborhood' (0.01, 0.001 respectively) of the global optimum. E.g. for 0.1 neighborhood, the condition $|x_i^{BSF} - x_i^{OPT}| < 0.1$ for all i must hold.
- *#Evals0.1* (*#Evals0.01*, *#Evals0.001*). The average number of evaluations needed to get to the 0.1 neighborhood (0.01, 0.001 respectively) is computed only from the runs in which the algorithm succeeded to get that close to the global optimum.
- *PopSizeUsed*. Population size for which the results are reported.
- *TimeElapsed*. The average length of one run in seconds. Only informative measure, because the number of PCA or ICA invocations depends on the population size.

3.3 Results and Discussion

The results are presented in Table 1. Reported statistics are chosen for that population size for which the algorithm gained the best average BSF score.

PCA and ICA rotate the population to find uncorrelated or the most independent version of the population, respectively, which can be a very hard and sometimes very unprecise work due to the finite samples. Let us first review influence of the preprocessing when optimizing the Griewangk function. Both types of preprocessing are useful for both versions of the function, 2D and 10D. The UMDA coupled with PCA or ICA is able to gain better solutions than UMDA without any preprocessing. However, there is a difference between PCA and ICA: for 2D Griewangk function, PCA preprocessing works better, while for the 10D version the ICA is preferable.

Different results can be seen for the Two Peaks function. The best choice here is not to use any preprocessing at all because the function is separable and any rotation makes it only harder. The results confirm this statement. Nevertheless, we can compare the PCA and ICA on this function and ICA seems to be much better transformation for this case — the UMDA with ICA outperformed the UMDA with PCA in the quality of the found solution, in the speed and in the reliability of finding a solution. Interesting results (although not reported) were observed for the UMDA with PCA when solving the 20D Two Peaks function. Almost independently of the population size, the average quality of the solution found after 50,000 evaluations was about 20. The reported case for the population size of 20 is the only exception. Probable reason of this behavior is that with only 20 data points the PCA cannot be estimated reliably and because of these 'errors' in estimation the algorithm can produce improving steps more often. With population size 50 and higher, the PCA produces more stable transformation which does not allow the algorithm to improve the solutions very often.

3.4 Example Test of Independence

In the previous subsection, it was experimentally shown that there are situations when PCA can

Table 1. Results of experiments. Statistics described in Sec. 3.2 are presented in each 'cell' in the following order: 1st row: BSF ± StdevBSF. 2nd row: Found0.1, Found0.01, Found0.001. 3rd row: #Evals0.1, #Evals0.01, #Evals0.001. 4th row: PopSizeUsed, TimeElapsed.

Alg.	2D Griewangk	10D Griewangk	2D Two Peaks	20D Two Peaks
UMDA	0.0024 ± 0.0026	$3.7 \cdot 10^{-4} \pm 0.0017$	0 ± 0	0.0027 ± 0.0059
	14 0 0	19 19 19	20 20 20	20 20 18
	3887 — —	17906 28012 37612	541 1621 2621	9941 18421 26890
	800, 42.4	600, 211.1	200, 27.6	400, 563.7
UMDA/PCA	$9.3 \cdot 10^{-6} \pm 4.1 \cdot 10^{-5}$	0 ± 0	0.44 ± 0.41	14.93 ± 2.008
	20 19 18	20 20 20	12 6 4	0 0 0
	3198 14537 17280	4793 7613 10413	4205 28960 21606	— — —
	600, 44.9	200, 214.0	600, 60.0	20, 67.9
UMDA/ICA	0.001 ± 0.0023	0 ± 0	0.0167 ± 0.0515	4.1 ± 4.3
	18 12 10	20 20 20	20 18 18	6 2 2
	3255 28904 33518	2913 4528 6353	2294 13319 17497	18436 18796 22596
	600, 61.5	100, 608.1	600, 59.6	200, 1236.4

produce more independent data points than ICA. It was the case of 2D Griewangk function. This fact is investigated in more theoretical way in this section.

Generally speaking, it is the independence of individual variables which plays the major role in the EA efficiency if we use the UMDA. However, it is very hard to test for the independence of continuous variables. The notion of independence is usually defined by the probability density functions (PDFs). Let the $p(X_1, X_2)$ be the joint PDF of variables X_1 and X_2. We say that X_1 and X_2 are independent if the joint PDF can be factorized as follows:

$$p(X_1, X_2) = p_1(X_1)p_2(X_2), \tag{5}$$

where p_1 and p_2 are the marginal PDFs. This definition can be extended to any number of variables. If we wanted to test the independence based on current data set directly using the definition 5, it would require to build some approximations of the PDFs. For the example purposes the following simplification is used. It is possible to discretize the domain of each variable into C bins. Then, we can treat the data points as measured on categorical scale and thus we can use the χ^2-test of independence for contingency tables. Of course, this test is only an approximation but its value lies mainly in the fact that it is relatively fast and not complicated.

Let us use this test of independence for a data pattern that can be observed during evolution of the 2D Griewangk function (the data points form five clusters in a pattern which evokes the number 5 on a dice). First, the output of ICA is analyzed. After that, the same analysis is carried out for the output of PCA.

Each of the two variables is divided to 3 equal intervals. The 100 data points are uniformly divided into the 5 clusters. The observed contingency table is then depicted in Fig. 2 (left).

20		20	40
	20		20
20		20	40
40	20	40	100

16	8	16	40
8	4	8	20
16	8	16	40
40	20	40	100

Fig. 2. Observed (left) and expected (right) contingency table

The contingency tables are matrices and their cells contain the number of data points which belong to the respective D-dimensional interval (2-dimensional in this example). Let the observed contingency table be $\mathbf{O}$ with the entries $O_{i,j}$, $i, j = 1, 2, 3$. Let us further describe the marginal sums of rows as $O_{i,:} = \sum_j O_{i,j}$ (the last column of the tables in Fig. 2) and the marginal sums of columns as $O_{:,j} = \sum_i O_{i,j}$ (the last row of the tables in Fig. 2). The lower right entry of the table is N — the overall number of data points, i.e. the sum of all table entries.

The χ^2-test only compares the observed frequencies with the expected ones. In order to use this test as the test of independence we have to construct the contingency table of frequencies expected when the assumption of independence would hold. Let the expected contingency table be $\mathbf{E}$. Its entries (see Fig. 2, right) can be easily computed as

$$E_{i,j} = \frac{O_{i,:}O_{:,j}}{N}. \tag{6}$$

The test statistic Chi^2 will then be a measure of how much the observed contingency table differs from the expected one and we can define it as follows:

$$Chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}. \tag{7}$$

The Chi^2 random variable has the χ^2 distribution. If the number of rows is I and the number of columns is J, the number of degrees of freedom for the χ^2 distribution is $(I-1)(J-1)$, i.e. for this example it is equal to 4. The p-value of this test (computed as $p = 1 - CDF\chi^2(dof, Chi^2)$) is the probability of observing this or greater Chi^2 if the assumption of independence holds. Thus, the p-value can be interpreted as a measure of independence of the two variables. If the p-value is close to zero we can be pretty sure that the variables are not independent. If the p-value is not close to 0, the variables can be considered as independent (strictly speaking, we do not have enough evidence to prove their dependence).

Coming back to our example, computing the test statistic and the p-value results to $Chi^2 = 4\frac{(20-16)^2}{16} + 4\frac{(0-8)^2}{8} + \frac{(20-4)^2}{4} = 4 + 32 + 64 = 100$ and $p = 1 - CDF\chi^2(4, 100) = 0$, thus we can say that based on our finite sample from the distribution, there is almost no chance that the variables are independent.

Now, we try to use the same test for the data rotated by the outcome of the PCA analysis. In that case the contingency tables look like this:

	20		20
20	20	20	60
	20		20
20	60	20	100

4	12	4	20
12	36	12	60
4	12	4	20
20	60	20	100

Fig. 3. Observed (left) and expected (right) contingency table for data rotated by PCA.

The results of the test are $Chi^2 = 4\frac{(0-4)^2}{4} + 4\frac{(20-12)^2}{12} + \frac{(20-36)^2}{36} = 16 + 21.33 + 7.11 = 44.44$ and $p = 1 - CDF\chi^2(4, 44.44) = 5.2 \times 10^{-9}$. We can see, that even in this case we can hardly describe the variables as independent in an absolute sense, but we can state that they are 'more independent' than in the previous case (preprocessed by ICA).

4. CONCLUSIONS

The results presented in this article and their discussion suggest that there is no general rule of the form 'preprocess the population with ICA and you will get better results for sure'. Although ICA preprocessing seems to work well on many problems, there are cases when PCA results are better (sometimes the best transformation is no transformation). Thus, the decision as to which preprocessing method to use should be taken on a case to case basis, or we need a tool which would evaluate the quality of transformations suggested by the PCA and ICA. Such a tool would enable us to choose among all possibilities (no transformation, PCA, ICA, ...) the 'best' one during the run of the algorithm. This role can be played by some tests of independence similar to that suggested in Sec. 3.4. These tests are subject to further research.

The population preprocessing can be also thought of as a kind of simple linkage learning. Although other methods with the linkage learning capability are likely to do their jobs better, they create rather complex models of dependency structures among variables and require special methods for creating new individuals. On contrary, population preprocessing with linear transformations has limited potential for reducing the dependencies, but allows us to use almost all conventional methods of crossover and mutation.

ACKNOWLEDGMENTS

The project was partially supported by the Czech Ministry of Education with the grant No. MSM6840770012 entitled "Transdisciplinary Research in Biomedical Engineering".

REFERENCES

Cho, Dong-Yeon and Byoung-Tak Zhang (2004). Evolutionary continuous optimization by distribution estimation with variational bayesian independent component analyzers mixture model. In: *Parallel Problem Solving from Nature VIII* (Xin Yao et al., Ed.). LNCS 3242. Springer Verlag. Birmingham, UK. pp. 212–221.

Friedman, Jerome H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association* **82**(397), 249–266.

Hansen, Nikolaus and Andreas Ostermeier (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* **9**(2), 159–195.

Hotelling, Harold (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**, 417–441.

Hyvärinen, Aypo (1999). Survey on independent component analysis. *Neural Computing Surveys* **2**, 94–128.

Hyvärinen, Aypo and Erkki Oja (2000). Independent component analysis: A tutorial. *Neural Networks* **13**(4–5), 411–430.

Pošík, Petr (2003). Comparing various marginal probability models in evolutionary algorithms. In: *MENDEL 2003* (Pavel Ošmera, Ed.). Vol. 1. Brno University. Brno. pp. 59–64. ISBN 80-214-2411-7.

Zhang, Quingfu, Nigel M. Allison and Hijun Jin (2000). Population optimization algorithm based on ICA.