Control and Cybernetics

vol. **25** (1996) No. 1

Clustering of parameters on the basis of correlations via simulated annealing

by

Gintautas Dzemyda

Institute of Mathematics and Informatics,
Akademijos St. 4, 2600 Vilnius, Lithuania,
e-mail: dzemyda@ktl.mii.lt

Abstract. The application of simulated annealing is considered in solving the problem of parameter clustering according to their correlation matrix. The problem is formulated as a problem of combinatorial optimization and attempted to be solved using the simulated annealing strategy. The algorithms, realizing such a strategy, are reviewed and investigated on the basis of test and real data.

Keywords: simulated annealing, data analysis, parameter clustering, combinatorial optimization.

1. Introduction

Any set of similar objects may be often characterized by common parameters (variables). The term "object" may cover, e.g., people, equipment, or produce of manufacturing. Any parameter may take some values. A combination of values of all parameters characterizes a concrete object from the whole set. The values obtained by any parameter depend on the values of other parameters, i.e., the parameters are correlated. The correlation matrix of parameters may be calculated during the analysis of objects composing the set. There exist groups of parameters characterizing different properties of the object. The problem is to find these groups.

One of the major objectives of various data analysis methods is to discover relations among the parameters. The methods analyzed in this paper are oriented to the analysis of correlation matrices and, in particular, to the clustering of parameters on the basis of correlations.

Examples of real correlation matrices:

1. The matrix of 8 physical parameters measured on 305 schoolgirls, Harman (1976).
2. The matrix of 11 parameters characterizing the development of agriculture in two Canadian provinces, Braverman and Muchnik (1983).

3. The matrix of 33 parameters of a tractor driver, Lumelsky (1970).
4. The matrix of 24 psychological tests on 145 pupils of the 7th and 8th forms in Chicago, Harman (1976).

The scope of this paper is out of search for the answer which method of parameter clustering (not realizations of the same method) is better, which criterion characterizing the partitioning quality is better. Investigator should have some methods at his disposal and integrate interactively their results. The paper deals with various simulated annealing strategies, which may be applied in solving combinatorial problems where the functional of partitioning quality is to be optimized.

2. The problem

The problem is to partition the parameters $x_1, \ldots, x_n$ into a fixed number p of non intersecting and homogeneous, in a certain sense, groups $A_1, \ldots, A_p$ by the correlation matrix $R = \{r_{x_i x_j}, i, j = \overline{1,n}\}$ characterizing the connections among the parameters ($r_{x_i x_j}$ is the correlation coefficient of parameters x_i and x_j). The covariance matrix may be used instead of the matrix R. However, the parameters with a greater variance will be more significant in the analysis. There is no *a priori* information regarding the number and size of groups.

Algorithms of parameter clustering are widely used to analyze the real data. There are two possibilities of such an analysis. The first one is to analyze the data matrix $Z = \{z_{ij}, i = \overline{1,t},\ j = \overline{1,n}\}$, where t is the number of objects, and any parameter x_s is characterized by the data vector $(z_{is},\ i = \overline{1,t})$. The algorithms for clustering objects are suitable here, because in this case parameters are interpreted as objects and objects as parameters (see Braveman and Muchnik, 1983). But sometimes t may be large or only the correlation matrix of parameters is known. In this case, the analysis of a set of $(n-1) \cdot n/2$ elements of correlation matrix (or $(n+1) \cdot n/2$ elements of covariance matrix) is made instead of $t \cdot n$ elements of the data matrix Z, i.e., the compressed information is used.

There is a variety of parameter clustering algorithms on the basis of correlations. The modifications of Harman's algorithm, Harman (1976), based on the analysis of correlations are included in SAS (1982). The algorithms of such a type are also presented, for example, in papers by Braveman and Muchnik (1983), Dzemyda (1990), Lumelsky (1970). All the algorithms start from some initial partition selected by some algorithm or by a certain knowledge about the problem. The goal of these algorithms is to recognize the internal structure of a system of parameters characterizing objects from a given set. But they do not try clustering the objects on the basis of parameter partition: there is a lot of efficient algorithms oriented to the clustering of objects (e.g., see Anderberg, 1973; Hartigan, 1975; Späth, 1980; Owsiński, 1986; Mezzich and Solomon, 1980). Some interaction between the clustering of objects and analysis of the groups of parameters may be found in the paper by Diday (1986). A review of

algorithms for a simultaneous clustering of parameters and objects is given by Mezzich and Solomon (1980).

In this paper we deal with the algorithms of extremal parameter grouping, Braveman and Muchnik (1983), Dzemyda (1987;1988;1990), Braveman (1970), Dzemyda and Valevičienė (1988), Dzemyda and Senkienė (1992), Dzemyda, Senkienė and Valevičienė (1990), based on the analysis of correlations and maximizing the partitioning quality

$$I_1 = \sum_{L=1}^{p} \sum_{x_i \in A_L} r^2_{x_i F_L},$$

where F_L is the factor with a unit variance, corresponding to the group A_L; $r_{x_i F_L}$ is the correlation coefficient of the parameter x_i and the factor F_L. The factors F_L, $L = \overline{1,p}$, are selected so that to maximize the sums

$$\sum_{x_i \in A_L} r^2_{x_i F_L}, \; L = \overline{1,p}. \tag{1}$$

Dzemyda (1987) proved that:

1. $$F_L = \sum_{x_i \in A_L} \alpha_i^L x_i / \sqrt{\lambda_L}, \tag{2}$$
$$r_{x_s F_L} = \sum_{x_i \in A_L} \alpha_i^L r_{x_i x_s} / \sqrt{\lambda_L},$$
where λ_L is the greatest eigenvalue of the matrix $R_L = \{r_{x_i x_j}, x_i, x_j \in A_L\}$, α_i^L are components of the normalized eigenvector of the matrix R_L corresponding to λ_L.
2. $r_{x_s F_L} = \sqrt{\lambda_L}\alpha_s$ as $x_s \in A_L$.
3. $I_1 = \sum_{L=1}^{p} \lambda_L$.

From (2) we observe that the factors F_L, $L = \overline{1,p}$, are linear combinations of parameters from the corresponding groups A_L, $L = \overline{1,p}$. The coefficients of the linear combinations are selected so that to maximize the sums (1). The values of coefficients of the linear combination are proportional to the elements of the eigenvector corresponding to the greatest eigenvalue λ_L of R_L, and $\lambda_L = \sum_{x_i \in A_L} r^2_{x_i F_L}$. In this manner the ideas of factor analysis are applied to the formulation of the problem of parameter grouping.

DEFINITION 2.1 *By the local maximum of the functional I_1 we shall call its value, corresponding to such a partition, where the squared correlation coefficient of any parameter with the factor, corresponding to the group including this parameter, is greater than or equal to that of the parameter with other factors, i.e., for any parameter x_s (let $x_s \in A_L$) the following inequality holds:*

$$r^2_{x_s F_L} \geq r^2_{x_s F_k}, \; k = \overline{1,p}, \; k \neq L.$$

Such a definition is useful in creating a strategy for maximizing I_1. Let us consider a parameter x_s ($x_s \in A_k$). If $r^2_{x_sF_L} > r^2_{x_sF_k}$, $L \neq k$, then the transfer of x_s into the group A_L will increase the value of I_1 (see Braveman and Muchnik, 1983). But by using the values of $r_{x_sF_L}$, $L = \overline{1,p}$, it is impossible to determine the group where the value of I_1 increases at most after transfering the parameter x_s (see Dzemyda, 1987).

The partition will also correspond to the local maximum defined above, if the transfer of any parameter from its group to another one will not increase the value of I_1. Such a situation occurs when for any parameter x_s (let $x_s \in A_L$) the following inequalities are satisfied:

$$\lambda_k^{+s} + \lambda_L^{-s} \leq \lambda_k + \lambda_L, \; k = \overline{1,p}, \; k \neq L,$$

where

λ_L^{-s} is the maximal eigenvalue of the matrix $R_L^{-s} = \{r_{x_ix_j}, x_i, x_j \in A_L^{-s}\}$, where the group A_L^{-s} is obtained from the group A_L by eliminating x_s;

λ_k^{+s} is the maximal eigenvalue of the matrix $R_k^{+s} = \{r_{x_ix_j}, x_i, x_j \in A_k^{+s}\}$, where the group A_k^{+s} is obtained from the group A_k by adding x_s.

The value of I_1 will increase after transferring x_s (let $x_s \in A_L$) into the group A_k if we succeed in finding such k ($k \neq L$) where $\lambda_k^{+s} + \lambda_L^{-s} > \lambda_k + \lambda_L$, $L \neq k$. This fact is also useful in creating a strategy for maximizing I_1.

The global maximum of I_1 belongs to the set of local maxima, too.

The problem can be formulated as a combinatorial optimization problem. Let $X^1, \ldots, X^n$ be variables taking discrete values from 1 to p, $K = \{X = (X^1, \ldots, X^n), \; X^i \in \{1, \ldots, p\}, \; i = \overline{1,n}\}$. Let us introduce a function $f(X^1, \ldots, X^n)$ that is related with the functional I_1 in such a manner:

$$f(X^1, \ldots, X^n) = I_1, \text{ where } x_i \in A_L \text{ as } X^i = L.$$

It means that any point from K corresponds to the fixed parameter partition, and any partition of parameters corresponds to some point in K.

The problem of parameter clustering can be formulated as follows:

$$\max f(X) \tag{3}$$

subject to

$$X = (X^1, \ldots X^n) \in K \tag{4}$$

$$\exists i: \; X^i = 1, \ldots, \exists i: \; X^i = p \tag{5}$$

A functional similar to I_1 is proposed and investigated by Dzemyda and Valevičienė (1988), Dzemyda (1990) for the clustering of objects. Instead of the correlation matrix of parameters, Dzemyda and Valevičienė (1988), Dzemyda (1990) use a matrix $\overline{K} = \{\overline{K}(Z_i, Z_j), i, j = \overline{1,t}\}$, where $Z_i = (z_{i1}, \ldots, z_{in})$, $i = \overline{1,t}$, are objects to be clustered into p non intersecting clusters,

$$\overline{K}(Z_i, Z_j) = e^{-\alpha \cdot \rho^2(Z_i, Z_j)},$$

α is a positive number, $\rho(Z_i, Z_j)$ is the generalized Euclidean distance between the objects Z_i and Z_j, Hartigan (1975):

$$\rho(Z_i, Z_j) = \sqrt{\sum_{k,l=1}^{n} \bar{r}_{kl}(z_{ik} - z_{jk})(z_{il} - z_{jl})},$$

where $\bar{R} = \{\bar{r}_{kl},\ k, l = \overline{1,n}\}$ is a non-negative defined symmetric matrix. In fact, some nonlinear transformation of the objects defined on the Euclidean space R^n into its subset $\bar{S}^n \subset R^n$ containing unit vectors is given above. Dzemyda and Valevičienė (1988), Dzemyda (1990) showed the efficiency of such transformation experimentally on the basis of well-known clustering methods and data sets.

Dzemyda (1990) proposed to transform the problem of parameter clustering into the clustering of points distributed in $\bar{S}^n$. The possibility to use the methods oriented to the clustering of objects (e.g., k-means – Späth, 1980) was grounded.

3. Deterministic algorithms

Deterministic algorithms, Braveman and Muchnik (1983), Dzemyda (1987;1988; 1990), Braveman (1970), Dzemyda and Valevičienė (1988), often find only the local maximum of I_1 which is not global. They are based on the analysis of parameters in consecutive order and on the search for a group of transferring the individual parameter with a view to increase the I_1 value. They use different strategies to determine when the parameter must be transferred from its group to another. The algorithms stop when the transfer of any parameter by the chosen strategy does not increase the value of I_1.

The investigations of deterministic algorithms, Dzemyda (1987), indicate that their efficiency depends substantially on the initial partition of parameters.

The performance of deterministic algorithms in this paper is illustrated on the basis of two algorithms proposed by Dzemyda (1987), representing different strategies of maximizing I_1. A1 is one of the fastest algorithms; A2 finds the greatest values of I_1 in comparison with other deterministic algorithms investigated by Dzemyda (1987).

When algorithm A1, Dzemyda (1987), Dzemyda and Senkienė (1992), considers the parameter x_s (let $x_s \in A_k$), it seeks the greatest squared correlation coefficient $r^2_{x_s F_L}$ among $r^2_{x_s F_j}$, $j = \overline{1,p}$, $j \neq k$. If $r^2_{x_s F_L} > r^2_{x_s F_k}$, then A1 transfers x_s into the group A_L and recalculates the factors F_k and F_L.

When algorithm A2, Dzemyda (1987), Dzemyda and Senkienė (1992), considers the parameter x_s (let $x_s \in A_k$), it seeks the group A_L, where the value of the functional I_1 increases most after transferring x_s, i.e., A2 looks for L maximizing $e = \lambda_L^{+s} + \lambda_k^{-s} - \lambda_L - \lambda_k$. If $e > 0$, then A2 transfers x_s into the group A_L and recalculates λ_k and λ_L.

4. Simulated annealing

Since Kirkpatrick et al (1983) much work has been done on simulated annealing for discrete variables and it has been used in a wide range of contexts. A brief description of different modifications to the discrete simulated annealing algorithm can be found in the paper by Eglese (1990). Simulated annealing was used in solving partitioning problems, also Trzebiatowski (1985), namely, partitioning of networks.

The problem (3)-(5) can be solved by using the simulated annealing strategy, too. It permits searching for the global maximum of I_1. The attempts to solve the problem by using simulated annealing were made by Dzemyda, Senkienė and Valevičienė (1990), Dzemyda and Senkienė (1992). Here we review and extend their results.

4.1. Algorithm

Let us consider the simulated annealing strategy in search of the global maximum of the combinatorial problem

$$\max f(X),$$

where $X = (X^1, \ldots, X^n) \in S = [A, B]^n \subset R^n$; $A = (A^1, \ldots, A^n)$; $B = (B^1, \ldots, B^n)$; X^i, A^i and B^i, $i = \overline{1,n}$, take integer values; $A^i \leq X^i \leq B^i$.

$A^k = 1$, $B^k = p$, $k = \overline{1,n}$, in the case of problem (3)-(5).

The performance of optimization algorithms based on simulated annealing can be generalized as follows. Let $m-1$ step be performed. The current point is X_{m-1}. The problem is to find the next current point X_m. It may be one of the neighbors of X_{m-1}. X_{m-1} can remain as the current point after m steps, too. The selection of X_m is divided into two stages. X_m is chosen from the neighbors of X_{m-1} in the first stage. Then X_m and X_{m-1} are compared in the second stage. X_{m-1} can become X_m with some probability.

The search for the global maximum of $f(\cdot)$ can be performed in such a manner:

the m-th step of the algorithm is as follows:

$$X_m^i = X_{m-1}^i + \xi^i, \; m = 1, 2, \ldots, \; i = \overline{1,n}, \tag{6}$$

where ξ^i, $i = \overline{1,n}$, are integers taking values with some probabilities:

a) ξ^i, $i = \overline{1,n}$, are random numbers taking integer values in the set $\{-1, 0, 1\}$; $P\{\xi^i = 0, i = \overline{1,n}\} = 0$, and the probability for any other combination of ξ^i, $i = \overline{1,n}$, to appear is equal to $1/(3^n - 1)$;

b) $\xi^i \in S^i = \{A^i - X_{m-1}^i, A^i + 1 - X_{m-1}^i, \ldots, B^i - X_{m-1}^i\} \setminus \{0\}$, $i = \overline{1,n}$, with the same probability $p_i = 1/(B^i - A^i)$, i.e., $N(X_{m-1}) = S \setminus \{X_{m-1}\}$, where $N(X_{m-1})$ is the set of neighbors of X_{m-1}.

The probability of transition to the point X_m is defined by the formula:

$$P\{X_m\} = \begin{cases} 1, & \text{as} \quad f(X_m) > f(X_{m-1}) \\ \exp\{[f(X_m) - f(X_{m-1})]/T_m\}, & \text{as} \quad f(X_m) \leq f(X_{m-1}) \end{cases} \tag{7}$$

i.e., $P\{X_m\} = 1$ as $f(X_m) > f(X_{m-1})$; in the other case a random number $\eta \in [0,1]$ is generated: the point X_m will be initial for a new step ($(m+1)$-st) of the algorithm and in formula (6) it will replace X_{m-1} if $\eta < \exp\{[f(X_m) - f(X_{m-1})]/T_m\}$, and the point X_{m-1} remains as the initial one for a new step, otherwise.

$$T_m = c/\ln(1 + m_0 + m), \tag{8}$$

or

$$T_m = c/\ln[\ln(1 + m_0 + m)], \tag{9}$$

$m = 1, 2, \ldots$ is the number of a step, c is a positive constant, m_0 is some constant from $[1, \infty)$.

The proof of convergence of the algorithm in probability to the global maximum of $f(\cdot)$ is based on the results of Mitra, Romeo and Sangiovanni-Vincentelli (1986).

The transition probability $P\{X_m\}$ with an unknown parameter c (see (7), (8) and (9)), may be modified into the form with an unknown parameter $\delta \in (0,1]$. If we use some initial probability $P\{X_1\} = \delta$ as $m = 1$ and if T_m has the form (8), then the constant c can be expressed:

$$c = [f(X_1) - f(X_0)] \cdot \ln(2 + m_0)/\ln\delta,$$

where X_0 and X_1 are such that $f(X_1) < f(X_0)$. Then (7) will have such a form (for $m = 2, 3, \ldots$):

$$P\{X_m\} = \\ = \begin{cases} 1, & \text{as} f(X_m) > f(X_{m-1}) \\ (1 + m_0 + m)^{\frac{[f(X_m) - f(X_{m-1})]}{[f(X_1) - f(X_0)]} \cdot \frac{\ln\delta}{\ln(2+m_0)}}, & \text{as} f(X_m) \leq f(X_{m-1}) \end{cases} \tag{10}$$

If T_m has the form (9), then the transition probability may be transformed as follows:

$$P\{X_m\} = \\ = \begin{cases} 1, & \text{as} f(X_m) > f(X_{m-1}) \\ [\ln(1 + m_0 + m)]^{\frac{[f(X_m) - f(X_{m-1})]}{[f(X_1) - f(X_0)]} \cdot \frac{\ln\delta}{\ln[\ln(2+m_0)]}}, & \text{as} f(X_m) \leq f(X_{m-1}) \end{cases} \tag{11}$$

Taking into account a specific character of the functional, characterizing the partitioning quality of parameters, we suggest restricting the set of neighbors of the current point. Thus, two additional special cases of ξ^i selection for (6) are used:

c) $$P\{\xi^k = -1\} = P\{\xi^k = 1\} = 1/2, \quad (12)$$
$$\xi^i = 0,\ i = 1, 2, \ldots, k-1, k+1, \ldots, n,$$
$$k = 1, 2, \ldots, n,$$

d) $$\xi^k \in S^k = \{A^k - X^k_{m-1}, A^k + 1 - X^k_{m-1}, \ldots, B^k - X^k_{m-1}\} \setminus \{0\} \quad (13)$$

with the same probabilities:

$$p_k = 1/(B^k - A^k),$$
$$\xi^i = 0,\ i = 1, 2, \ldots, k-1, k+1, \ldots, n,$$
$$k = 1, 2, \ldots, n.$$

Case c) is a restriction of case a), and case d) is that of b). The peculiarity of these two cases is that only the k-th coordinate of X_m and X_{m-1} differs, and different values of k correspond to the consequent steps. The relation of m and k may be defined in a more sophisticated way (see for an example of such a relation the chapter below).

4.2. Realizations

Algorithms SA1 and SA2 are concrete realizations of the algorithm proposed above. Their peculiarities are the following:

1. The algorithms start from the point $\bar{X}_0$; the m-th step of algorithm is as follows (initially $X_0 = \bar{X}_0$, $m = 1$, $P\{X_m\} = 1$):
$$X^i_m = X^i_{m-1} + \xi^i,\ i = \overline{1, n}, \quad (14)$$
where ξ^i, $i = \overline{1, n}$, are integers taking values by (12*) (i.e., by (12) or (13)). $X_0 = X_m$ as $f(X_m) \geq f(X_{m-1})$. $X_1 = X_m$ and further calculations are performed by formulae (10), (12*) and (14) starting from the $(m+1)$-st step (various strategies for a further m_0 and m selection are presented in the sixth peculiarity) as $f(X_m) < f(X_0)$.
2. The relation of m and k (the strategy of k changing) is such: $p-1$ step of the algorithm are performed for each fixed value of k. Thus, k corresponds to the number of the variable, whose value is changed, while the values of other variables are fixed. The totality of the calculations above, when the value of k runs from 1 to n, is called an *iteration* of the algorithm. One iteration requires no more than $n \cdot (p-1)$ calculation of the function $f(\cdot)$ values.
3. Restriction (5) is taken into account. Let a new value of k be fixed and the parameter x_k be the only one in its group. Then the algorithm passes to the next k value.
4. Let the value of k be fixed. There may be some coincidental argument points among $p-1$ point in which it is necessary to calculate the value of $f(\cdot)$. The calls for program realizing $f(\cdot)$ are not reiterated in such a situation, but the number of calculated values of $f(\cdot)$ is increased.
5. Only the necessary part of the function $f(\cdot)$ is recalculated when we need to compute the unknown value of $f(\cdot)$.

6. The following strategies for m_0 and m initial selection were investigated:
   a) $m_0 = 1$; m is equal to the number of the function $f(\cdot)$ values calculated, the calculation of $f(\bar{X}_0)$ value is also taken into account;
   b) $m_0 = 1$; $m = 1$;
   c) m_0 is the number of $f(\cdot)$ calculations used to obtain $f(X_0)$; $m = 1$.
7. SA1 uses (12), SA2 – (13).
8. $X_m^i = A^i$ as $X_{m-1}^i = B^i$ and $\xi^i = 1$, and $X_m^i = B^i$ as $X_{m-1}^i = A^i$ and $\xi^i = -1$ when we use (12).

Algorithms SA1 and SA2 use the probabilistic selection (described by (12) and (13), respectively) of a point for the next calculation of the objective function value. The algorithm below uses a deterministic selection. Let us denote it by SA3. Its peculiarities are the following:

1. This peculiarity differs from the first one of SA1 and that of SA2 like this: in (14) ξ^k takes the values from S^k in a deterministic way and $\xi^i = 0$ ($i = \overline{1,n}$, $i \neq k$) during calculations when the value of k is fixed.
2. For each fixed value of k the value of ξ^k runs from $A^k - X_{m-1}^k$ to $B^k - X_{m-1}^k$ with the exception of $\xi^k = 0$. The totality of above calculations, when the value of k runs from 1 to n, is called the *iteration* of SA3.
3. The third, fifth and sixth peculiarities of SA1 and SA2 remain the same for SA3.

Some properties of algorithms are presented below (see proofs in the paper by Dzemyda and Sienkienė, 1992).

REMARK 4.1 SA3 *is identical to* A2, *Dzemyda (1987), if* $\delta = 0$ *and if* $P\{X_m\} = 0$ *as* $f(X_m) = f(X_{m-1})$ *are used.*

Let us denote:

1. $X_M = (X_M^1, \ldots, X_M^n)$ is the current point of the algorithm after M iterations (X_M is not necessarily the point, where the maximum of $f(\cdot)$ is achieved after M iterations);
2. $S_1 = \bigcup_{i=1}^n [\{(X_M^1, \ldots, X_M^{i-1}, 1, X_M^{i+1}, \ldots, X_M^n), \ldots, (X_M^1, \ldots, X_M^{i-1}, p, X_M^{i+1}, \ldots, X_M^n)\} \setminus \{X_M\}]$;
3. $\bar{S}$ is the subset of S_1 consisting of the points, that satisfy the restriction (5).

REMARK 4.2 *Let* X_M *be the current point of* SA3 *after* M *iterations. If the current point remains the same after the* $(M+1)$*-st iteration (i.e.* $X_{M+1} = X_M$*),* X_M *corresponds to the parameter partition conforming to the local maximum of functional* I_1*, i.e., the transfer of any parameter from its group to another one does not increase the value of* I_1.

REMARK 4.3 *Let* X_M *be the current point of* SA2 *after* M *iterations. If it remains the same during consecutive iterations, the probability of calculation of* $f(\cdot)$ *values at all the points of* $\bar{S}$ *in consecutive iterations grows to* 1.

REMARK 4.4 *Let X_M be the current point of* SA2 *after M iterations. If it remains the same during some consecutive iterations and if $f(\cdot)$ values are calculated at all the points of $\bar{S}$ during these iterations, X_M corresponds to the parameter partition, conforming to the local maximum of I_1.*

The difference of SB-algorithms (SB1, SB2, SB3) from the corresponding SA-algorithms (SA1, SA2, SA3) is that SA-algorithms use (8) and SB-algorithms use (9). The goal of introduction of a double logarithm in the T_m expression was to slow down the cooling schedule. As a result the transition probability $P\{X_m\}$ has been changed: SA-algorithms use (10) and SB-algorithms use (11).

Remarks 4.1-4.4 concerning SA-algorithms are valid for the corresponding SB-algorithms.

5. Experimental investigation

5.1. The experiments

The investigation was divided into two stages:

1. **The search for an optimal value of** δ. The convergence of algorithms depends on the value of δ (or c). It is impossible to determine the value of δ theoretically. Thus, it remains only the experimental search.
2. **The comparison of the efficiency of algorithms**, when optimal δ (or c) is used.

The investigations were performed by IBM PC/AT computer. The third mode of m_0 and m selection was used because the optimal partitioning quality of all the modes turned out to be similar (except for the optimal value of δ). The termination condition of annealing algorithms was the limited number of iterations (a more complex termination condition may be selected for practical problems). Random correlation matrices were generated. The matrices were obtained by generating points, uniformly distributed in a certain subset of R^n, and calculating their correlation matrix. So the data structure was random, and the problems generated were multiextremal. $n = 20$. The search for the optimal number of groups is out of interest in this paper. Thus, it was set $p = 4$ and the next initial parameter partition was chosen: $A_1 = \{x_1, \ldots, x_5\}$, $A_2 = \{x_6, \ldots, x_{10}\}$, $A_3 = \{x_{11}, \ldots, x_{15}\}$, $A_4 = \{x_{16}, \ldots, x_{20}\}$. The first stage of investigations has been performed on the basis of 20 random matrices, and the second one – on 50 matrices. The results were averaged.

The search for optimal δ is illustrated in Fig. 1. It is not very convincing as to the choice of δ. But we observe that the greatest values of $f(\cdot)$ are achieved by all three algorithms for δ lying in the neighborhood of 0.3. This value of δ was set for all further investigations of the annealing algorithms.

Figures 2 (a-e) illustrate the dependence of optimization results on the number of iterations. The results of solving 50 problems are averaged. We note that the performance of SA-algorithms is similar to that of the corresponding SB-

algorithms. But the SB-algorithms are better for a smaller number of iterations. It is easy to see in Fig. 2c for SA3 and SB3.

Another criterion for comparison of the algorithms may be a number of cases from 50 when the algorithm β yielded a better result than the algorithm ρ. Let us denote this criterion by E^{ρ}_{β}. The integral criterion for comparing the algorithm β_L with β_i, $i = \overline{1,l}$, $i \neq L$, may be such:

$$E_{\beta_L} = \sum_{i=1\ i\neq L}^{l} E^{\beta_i}_{\beta_L}.$$

Figures 3 (a-e) show the dependence of integral criterion E on the number of iteration of all the annealing algorithms.

The dependence of E on the iteration number is not always a monotonous function. This nonmonotonicity must be taken into account when the annealing algorithms are used for the initial partition of parameters, and the deterministic algorithms (e.g., A1 or A2, Dzemyda, 1987) seek the local optimum. The number of annealing iterations is preferable to be correspondent to the greatest values of the integral criterion.

Some additional properties of the algorithms are illustrated in Table 1. The results of commutation of annealing and deterministic algorithms are presented, too.

In Table 1 IT is the averaged number of iterations, where the maximum of $f(\cdot)$ was obtained, NL is the number of iterations performed by the SA and SB-type algorithms, IE is the averaged number of calculations of maximal eigenvalues in an iteration (calculations of such a type play a great part in computational expenditure of the algorithms investigated), T is the averaged computer time in seconds used by an iteration. The performance of deterministic algorithms is illustrated on the basis of A1 and A2, Dzemyda (1987). The IT values presented for A1 and A2 rows mean the averaged number of performed iterations (running through all the parameters); T means the averaged computer time used by A1 and A2. The last five rows in Table 1 correspond to the case when simulated annealing was used for the initial partition of parameters, and afterwards the result was specified by the deterministic algorithms. Here T denotes the averaged computer time used by both algorithms.

All the algorithms (both simulated annealing and deterministic) were used to analyze the real data.

The first experiment was carried out using the correlation matrix of 8 physical parameters measured on 305 schoolgirls, Harman (1976), SAS (1982): height, arm span, length of forearm, length of lower leg, weight, bitrochanteric diameter, chest girth, chest width. The matrix is given in Appendix 1. The investigations of these classical test data divided parameters into two groups: $A_1 = \{x_1, \ldots, x_4\}$ and $A_2 = \{x_5, \ldots, x_8\}$: the parameters of the first group characterize shapeliness, while the parameters of the second group characterize plumpness of girls. This is an "ideal" partition. It means that this data set

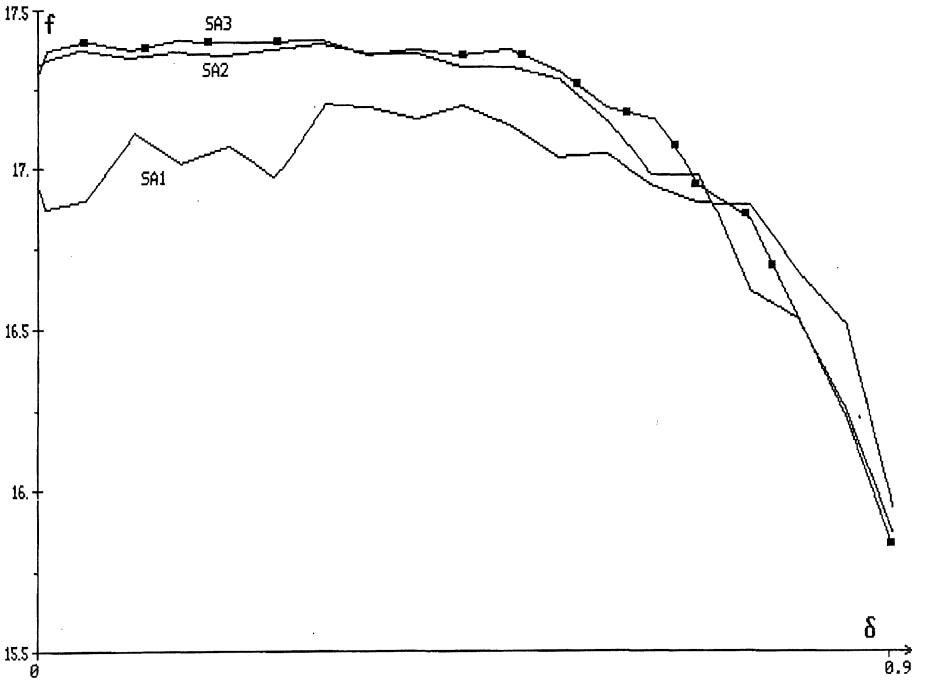

Figure 1. The dependence of SA1, SA2, and SA3 performance on δ

Algorithm	NL	IT	IE	T	$f(\cdot)$
SA1	10(20)	5.58 (9.52)	55.5	6.1	16.950 (17.050)
SA2	10(20)	6.18 (10.62)	62.1	6.5	17.283 (17.374)
SA3	10(20)	4.78 (8.74)	79.6	8.3	17.358 (17.394)
SB1	10(20)	4.22 (6.78)	55.2	6.0	16.862 (16.910)
SB2	10(20)	5.92 (8.52)	62.3	6.4	17.323 (17.361)
SB3	10(20)	4.54 (6.66)	79.7	8.3	17.351 (17.375)
A1	-	3.65	8.1	3.3	17.011
A2	-	4.68	80.0	37.3	17.291
SA1+A1	2	1.98	-	18.5	17.104
SB1+A1	2	1.96	-	17.6	17.121
SA3+A1	2	1.92	-	22.9	17.205
SB3+A1	2	1.84	-	22.2	17.245
SB3+A2	2	1.84	-	38.3	17.339

Table 1. The averaged results of test problem solving

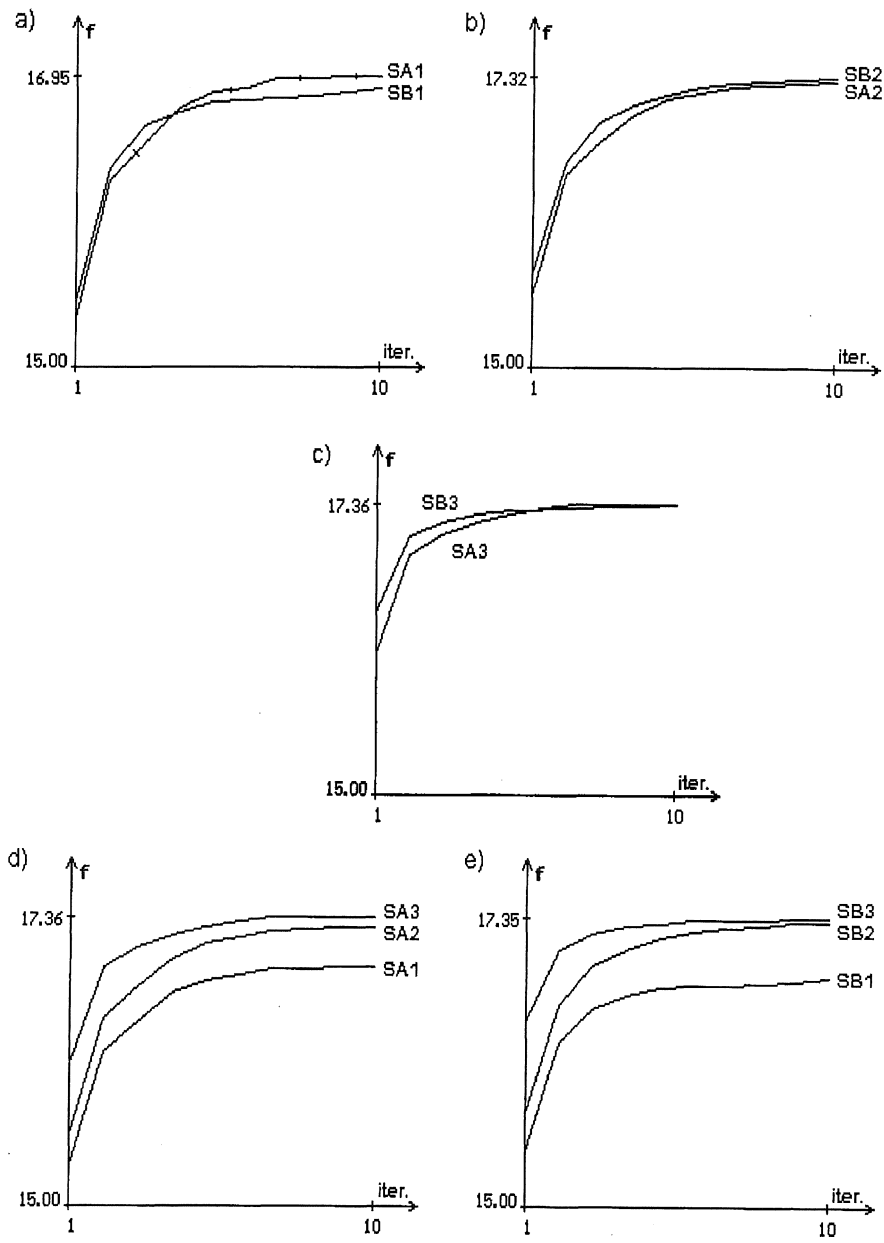

Figure 2. The dependence of optimization results on the iteration number

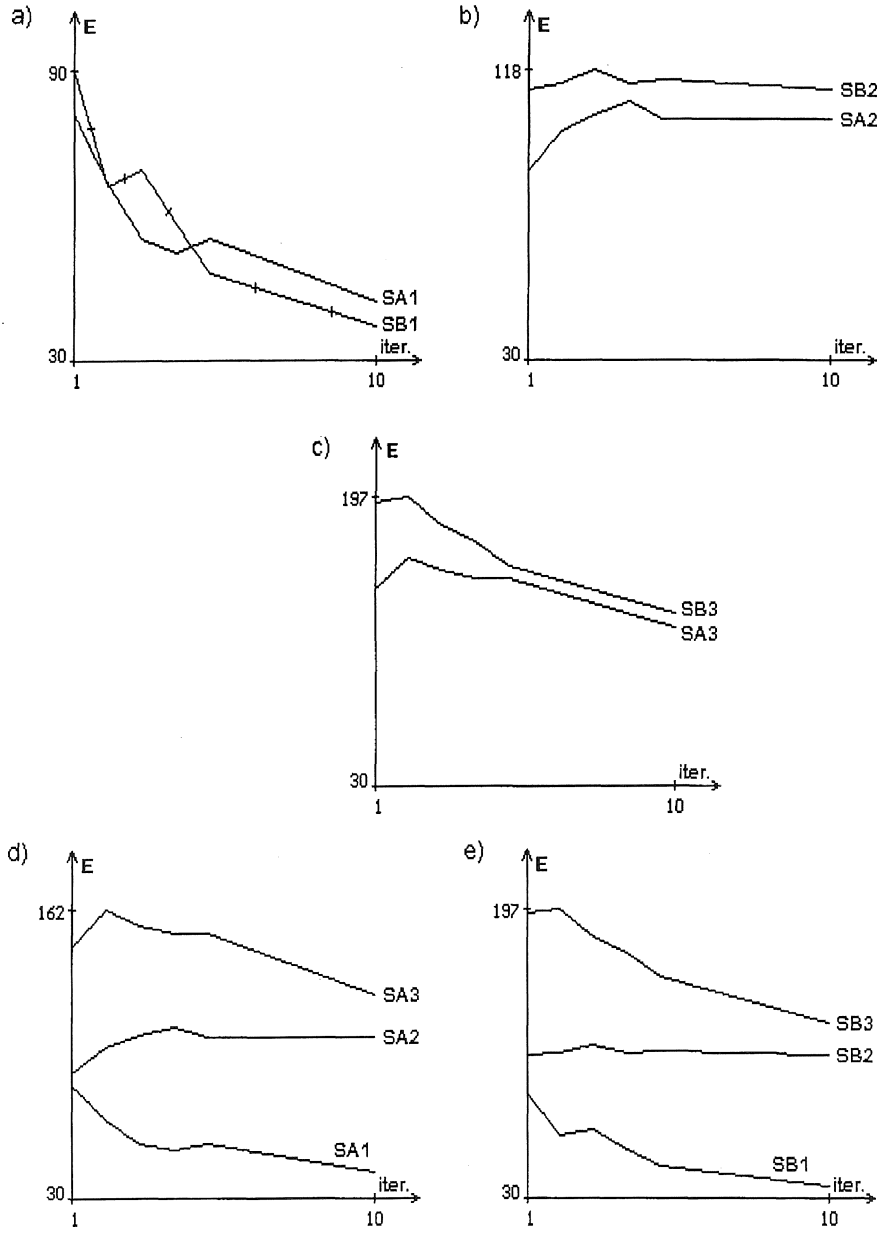

Figure 3. The dependence of integral criterion on the number of iteration

has a good degree of structure. Naturally, any fast algorithm (e.g., A1) may be successfully used for the analysis of such data, because the simulated annealing algorithms are oriented to difficult problems where the partition precision is essential.

The second experiment was carried out using the correlation matrix of 24 psychological tests on 145 pupils of the 7th and 8th forms in Chicago Harman (1976). The matrix is given in Appendix 2. There are five groups of tests:

1. Spatial perception $\{x_1, \ldots, x_4\}$.
2. Verbal tests $\{x_5, \ldots, x_9\}$.
3. The rapidity of thinking $\{x_{10}, \ldots, x_{13}\}$.
4. Memory $\{x_{14}, \ldots, x_{19}\}$.
5. Mathematical capabilities $\{x_{20}, \ldots, x_{24}\}$

The tests of the fifth group characterize a general development of the tested person. They do not characterize separate parts of his intellect. Thus, classifying all the tests into four groups the algorithms distribute the tests of the fifth group among the other four groups. The investigations indicated that the global maximum of I_1 is equal to 12.598 in case $A_1 = \{x_1, \ldots, x_4, x_{20}, x_{22}, x_{23}\}$, $A_2 = \{x_5, \ldots, x_9\}$, $A_3 = \{x_{10}, \ldots, x_{13}, x_{21}, x_{24}\}$, $A_4 = \{x_{14}, \ldots, x_{19}\}$.

The problem of I_1 maximization was solved starting from 50 random initial partitions of parameters. The results (the value of $f(\cdot)$ and the number GL of achieved global solutions) are presented in Table 2. NL is the number of iterations performed by the algorithms of SA and SB-type.

Simulated annealing algorithms were joined with the deterministic algorithms:

1. A1 was used in the fast local search for the nearest local maximum, because the simulated annealing not always stops at the local solution.
2. A2 was used to improve the initial partition.

The performance of various combinations of simulated annealing with the deterministic algorithms is illustrated in Table 2. The results of simulated annealing are put in brackets.

The results indicate that the simulated annealing algorithms are efficient in the analysis of the set of psychological tests. Good results are also obtained by A2 which is a partial case of SA3 (see Remark 4.1). The best results are obtained combining A2 and some iterations of SB3.

5.2. Conclusions from experimental investigations

The experimental investigation showed that by using simulated annealing one can find a better partition of parameters in comparison with that obtained by the deterministic algorithms.

We noticed a tendency to a significant improvement of the partition during initial iterations. Later on the results were specified. Thus, some iterations of simulated annealing algorithms can also be used for the initial partition of

Algorithm	$f(\cdot)$	NL	GL
SA1	12.220	10	15
SA2	12.598	10	49
SA3	12.595	10	48
SB1	12.148	10	9
SB2	12.596	10	49
SB3	12.592	10	47
SA1+A1	12.307 (12.158)	5	11 (9)
SA2+A1	12.561 (12.542)	5	36 (33)
SA3+A1	12.594 (12.594)	5	46 (46)
SB1+A1	12.234 (12.109)	5	7 (5)
SB2+A1	12.564 (12.547)	5	39 (35)
SB3+A1	12.586 (12.586)	5	47 (47)
A1	11.329	-	0
A2	12.573	-	43
A2+SB2	12.580	2	44
A2+SB2	12.596	3	48
A2+SB2	12.597	4	49
A2+SB3	12.591	2	47
A2+SB3	12.593	3	48
A2+SB3	12.598	4	50

Table 2. Experiments with the matrix of 24 parameters

parameters. Then deterministic algorithms, that are faster but require a good initial partition, can be used.

The practical use of SA1 (not for the initial partition) is doubtful because its partitioning quality is similar to that of A2, and A2 requires less computational expenditure and has no problems in choosing the value of a parameter (like δ) exerting some influence on the result. The initial partition has a great influence on the results of SA1.

The partitioning quality of SA3 is better than that of SA2. The improvement of partition stops after a larger number of SA2 iterations in comparison with SA3. However, one iteration of SA3 performs more calculations of maximal eigenvalues of symmetrical matrices. So it is difficult to determine which algorithm (SA2 or SA3) is better for use.

All the conclusions about SA-algorithms (SA1, SA2, and SA3) also apply to the corresponding SB-algorithms (SB1, SB2, and SB3).

The comparison of SA-algorithms with SB-algorithms indicates that SB-algorithms are better for the initial partition of parameters (using some iterations of annealing). SB2 and SB3 are a bit better than SA2 and SA3, respectively, when a greater number of iterations is used.

Various mixtures of deterministic and simulated annealing algorithms may speed up optimization, and yield better results.

6. Conclusions

Some algorithms for parameter clustering are proposed and investigated. They are more difficult than classical algorithms and require more computational resources. But they are oriented to the problems, not having a good degree of structure, and the cases when a high partition quality is required.

The algorithms can be modified for solving any clustering problem. Only the functional, characterizing the partitioning quality will be different.

As far as further research problems in this direction are concerned, it would be to analyze other criteria characterizing partitioning quality of parameters. The values of these criteria would be calculated significantly faster compared with calculation of I_1. The simulated annealing would be more effective in this case, because more sophisticated annealing strategies may be proposed and applied.

It would also be fruitful to study more precisely the application of simulated annealing to the clustering of objects, because the traditional criteria of partitioning quality of objects (e.g., k-means – Späth, 1980) are simpler and may be calculated significantly faster compared with I_1.

Acknowledgment

The author wishes to thank anonymous referees for many detailed and helpful comments and fruitful ideas for further investigations.

References

ANDERBERG, M.R. (1973) *Cluster Analysis for Applications.* Academic Press, New York.

BRAVERMAN, E.M. (1970) Methods of extremal grouping of parameters and the problem of apportionment of essential factors, *Avtomatika i Telemechanika*, **1**, 123-132 (in Russian).

BRAVERMAN, E.M. and MUCHNIK, I.B. (1983) *The Structural Methods for Empirical Data Processing.* Nauka, Moscow (in Russian).

DIDAY, E. (1986) Canonical analysis from the automatic classification point of view. In: J.W.Owsiński, ed., *Control and Cybernetics, Special Issue on Optimization Approaches in Clustering*, **15**, 2, 115-137.

DZEMYDA, G. (1987) On the extremal parameter grouping. In: A. Žilinskas, ed., *Teorija Optimal'nych Reshenij*, Vol. 12, Inst. Math. Cybern. Lith. Acad. Sci., Vilnius, 28-42 (in Russian).

DZEMYDA, G. (1988) The algorithms of extremal parameter grouping. In: A. Sydow, S.G. Tzafestas and R. Vichnevetsky, eds., *Mathematical Research*, Band 46, Akademie-Verlag, Berlin, 133-136.

DZEMYDA, G. (1990) Grouping on the sphere. In: *Teorija Optimal'nych Reshenij*, Vol. 14, Vilnius, 22-40 (in Russian).

DZEMYDA, G. and VALEVIČIENĖ, J. (1988) The extremal parameter grouping in cluster analysis. In: *Teorija Optimal'nych Reshenij*, Vol. 13, Vilnius, 36-53 (in Russian).

DZEMYDA, G. and SENKIENĖ, E. (1992) Simulated annealing for parameter grouping. In: *Trans. of the 11th Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes*, ACADEMIA, Prague, 373-383.

DZEMYDA, G., SENKIENĖ E. and VALEVIČIENĖ, J. (1990) On the problem of parameter grouping. In: H.-M. Voigt, H. Mühlenbein and H.-P. Schwefel, eds., *Evolution and Optimization'89. Selected Papers on Evolution Theory, Combinatorial Optimization, and Related Topics*, Akademie-Verlag, Berlin, 216-224.

EGLESE, R.W. (1990) Simulated annealing: a tool for operational research. *European Journal of Operational Research*, **46**, 271-281.

HARMAN, H.H. (1976) *Modern Factor Analysis*, 3rd ed. University of Chicago Press, Chicago.

HARTIGAN, J. (1975) *Clustering Algorithms.* Wiley - Interscience, New York.

KIRKPATRICK, S., GELATT, C.D. and VECCHI, M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 4598, 671-680.

LUMELSKY, V.Y. (1970) Grouping of parameters on the basis of communication matrix. *Avtomatika i Telemechanika*, **1**, 133-143 (in Russian).

MEZZICH, J.E. and SOLOMON, H. (1980) *Taxonomy and Behavioral Science, Comparative Performance of Grouping Methods.* Academic Press.

MITRA, D., ROMEO, F. and SANGIOVANNI-VINCENTELLI, A. (1986) Convergence and finite-time behavior of simulated annealing. *Adv. Appl. Prob.*, **18**, 3, 747-771.

OWSIŃSKI, J.W. ed. (1986) *Control and Cybernetics, Special Issue on Optimization Approaches in Clustering*, **15**, 2.

SAS User's Guide: Statistics (1982) SAS Institute Inc, Cary.

SPÄTH, H. (1980) *Cluster Analysis Algorithms for Data Reduction and Classification of Objects.* Ellis Horwood.

TRZEBIATOWSKI, G.W. (1985) Thermodynamic simulation procedure for partitioning problems. In: H.-M. Voigt, ed., *Informatis: Informationen-Reporte*, **1**, 12, Institut für Informatik und Rechentechnik, Berlin, 86-96.

Appendix 1

Correlation matrix of 8 physical parameters measured on schoolgirls

$i \setminus j$	1	2	3	4	5	6	7	8
1	1.000	0.846	0.805	0.895	0.473	0.398	0.301	0.382
2	0.846	1.000	0.881	0.826	0.376	0.326	0.277	0.415
3	0.805	0.881	1.000	0.801	0.380	0.319	0.237	0.345
4	0.859	0.826	0.801	1.000	0.436	0.329	0.327	0.365
5	0.473	0.376	0.380	0.436	1.000	0.762	0.730	0.629
6	0.398	0.326	0.319	0.329	0.762	1.000	0.583	0.577
7	0.301	0.277	0.237	0.327	0.730	0.583	1.000	0.539
8	0.382	0.415	0.345	0.365	0.629	0.577	0.539	1.000

Appendix 2

Correlation matrix of 24 tests

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.318	0.403	0.468	0.321	0.355	0.304	0.332	0.326	0.116	0.308	0.314	0.489	0.125	0.238	0.414	0.176	0.368	0.270	0.365	0.369	0.413	0.474	0.282
2	0.318	1.000	0.317	0.230	0.285	0.234	0.157	0.157	0.195	0.057	0.150	0.145	0.239	0.103	0.131	0.272	0.005	0.255	0.112	0.292	0.306	0.232	0.348	0.211
3	0.403	0.317	1.000	0.305	0.247	0.268	0.223	0.382	0.184	0.075	0.091	0.140	0.321	0.177	0.065	0.263	0.177	0.211	0.312	0.297	0.165	0.250	0.383	0.203
4	0.468	0.230	0.305	1.000	0.227	0.327	0.335	0.391	0.325	0.099	0.110	0.160	0.327	0.066	0.127	0.322	0.187	0.251	0.137	0.339	0.349	0.380	0.335	0.248
5	0.321	0.285	0.247	0.227	1.000	0.622	0.656	0.578	0.723	0.311	0.344	0.215	0.344	0.280	0.229	0.187	0.208	0.263	0.190	0.398	0.318	0.341	0.435	0.420
6	0.355	0.234	0.268	0.327	0.622	1.000	0.722	0.527	0.714	0.203	0.353	0.095	0.309	0.292	0.251	0.291	0.273	0.197	0.251	0.435	0.263	0.386	0.431	0.433
7	0.304	0.157	0.223	0.335	0.656	0.722	1.000	0.619	0.585	0.246	0.232	0.181	0.345	0.236	0.172	0.180	0.228	0.159	0.226	0.451	0.314	0.396	0.405	0.437
8	0.332	0.157	0.382	0.391	0.578	0.527	0.619	1.000	0.532	0.285	0.300	0.271	0.395	0.252	0.175	0.296	0.255	0.250	0.274	0.427	0.362	0.357	0.501	0.388
9	0.326	0.195	0.184	0.325	0.723	0.714	0.585	0.532	1.000	0.170	0.280	0.113	0.280	0.260	0.248	0.242	0.274	0.208	0.274	0.446	0.266	0.483	0.504	0.424
10	0.116	0.057	0.075	0.099	0.311	0.203	0.246	0.285	0.170	1.000	0.484	0.585	0.408	0.172	0.154	0.124	0.289	0.317	0.190	0.173	0.405	0.160	0.262	0.531
11	0.308	0.150	0.091	0.110	0.344	0.353	0.232	0.300	0.280	0.484	1.000	0.428	0.535	0.350	0.240	0.314	0.362	0.350	0.290	0.202	0.399	0.304	0.251	0.412
12	0.314	0.145	0.140	0.160	0.215	0.095	0.181	0.271	0.113	0.585	0.428	1.000	0.512	0.131	0.173	0.119	0.278	0.349	0.110	0.246	0.355	0.193	0.350	0.414
13	0.489	0.239	0.321	0.327	0.344	0.309	0.345	0.395	0.280	0.408	0.535	0.512	1.000	0.195	0.139	0.281	0.194	0.323	0.263	0.241	0.425	0.279	0.392	0.458
14	0.125	0.103	0.177	0.066	0.280	0.292	0.236	0.252	0.260	0.172	0.350	0.131	0.195	1.000	0.370	0.412	0.341	0.201	0.206	0.302	0.183	0.243	0.242	0.304
15	0.238	0.131	0.065	0.127	0.229	0.251	0.172	0.175	0.248	0.154	0.240	0.173	0.139	0.370	1.000	0.325	0.345	0.334	0.192	0.272	0.232	0.246	0.256	0.165
16	0.414	0.272	0.263	0.322	0.187	0.291	0.180	0.296	0.242	0.124	0.314	0.119	0.281	0.412	0.325	1.000	0.324	0.344	0.258	0.388	0.348	0.283	0.360	0.262
17	0.176	0.005	0.177	0.187	0.208	0.273	0.228	0.255	0.274	0.289	0.362	0.278	0.194	0.341	0.345	0.324	1.000	0.448	0.324	0.262	0.173	0.273	0.287	0.326
18	0.368	0.255	0.211	0.251	0.263	0.197	0.159	0.250	0.208	0.317	0.350	0.349	0.323	0.201	0.334	0.344	0.448	1.000	0.358	0.301	0.357	0.317	0.272	0.405
19	0.270	0.112	0.312	0.137	0.190	0.251	0.226	0.274	0.274	0.190	0.290	0.110	0.263	0.206	0.192	0.258	0.324	0.358	1.000	0.167	0.331	0.342	0.303	0.374
20	0.365	0.292	0.297	0.339	0.398	0.435	0.451	0.427	0.446	0.173	0.202	0.246	0.241	0.302	0.272	0.388	0.262	0.301	0.167	1.000	0.413	0.463	0.509	0.366
21	0.369	0.306	0.165	0.349	0.318	0.263	0.314	0.362	0.266	0.405	0.399	0.355	0.425	0.183	0.232	0.348	0.173	0.357	0.331	0.413	1.000	0.374	0.451	0.448
22	0.413	0.232	0.250	0.380	0.341	0.386	0.396	0.357	0.483	0.160	0.304	0.193	0.279	0.243	0.246	0.283	0.273	0.317	0.342	0.463	0.374	1.000	0.503	0.375
23	0.474	0.348	0.383	0.335	0.435	0.431	0.405	0.501	0.504	0.262	0.251	0.350	0.392	0.242	0.256	0.360	0.287	0.272	0.303	0.509	0.451	0.503	1.000	0.434
24	0.282	0.211	0.203	0.248	0.420	0.433	0.437	0.388	0.424	0.531	0.412	0.414	0.458	0.304	0.165	0.262	0.326	0.405	0.374	0.366	0.448	0.375	0.434	1.000