Control
and Cybernetics
VOL. 10 (1981) No. 3-4

Partitioning a Set of Entities Due to Their Mutual Dissimilarities

by

TADEUSZ STACHOWIAK
WIESŁAW STAŃCZAK

Institute of Computer Science
Polish Academy of Science
Warsaw, Poland

The paper deals with a problem of partitioning a set of entities into subsets. The decomposition is performed by taking into consideration the reciprocal dissimilarities between the entities. Both the number of subsets and the way of partitioning are determined. First, the goal function is defined and interpreted. The problem is to generate a partition such that it maximizes the value of the function mentioned. Then some basic properties of the goal function are formulated and proved. These properties provide a basis for decomposing the problem into two disjoint subproblems. One of them consists in consecutive enumeration over a smaller space of partial solutions. The second one is to generate the next solution on the basis of the previous result. The exact solution of the second subproblem is given and a fast heuristical procedure which generates (sub) optimal results is proposed. The procedure makes it possible to obtain an approximate solution of the general problem in time proportional to n^2, where n denotes the number of entities in the set considered.

1. Introduction

In many real life situations a non-trivial mathematical description of any real problem under consideration leads to a multidimensional and large scale model. Investigations performed on such a model are very complex and time-consuming. For simplification we often decompose the model into several pairwise disjoint parts (submodels). The way of the decomposition depends upon the kind of interconnections assigned to respective pairs of parts of the model and to pairs of entities. We propose a general classification of these interconnections, which divides them into two groups. To the first one we assign interconnections, which have the nature of similarity. The second group consists of interconnections with the opposite nature, i.e. the dissimilarity. To obtain a "good" decomposition of the model with interconnections of the first type we have to fulfil the two following conditions: every pair of entities with a relatively high value of similarity should belong to the same part, and two objects with a relatively small mutual similarity

Now, it is obvious that the maximization proceeds over both aspects of the decomposition, i.e. the number of subsets in the optimal partition and the way of partitioning the entities of X into the subsets mentioned.

3. Some properties of the global partition index

By the definition of partition and due to the formula (7) we obtain

$$M(P)=S(X, X)=M=\text{const.}, \tag{9}$$

for each $P \in \mathscr{P}$ (we omit here the subscript k, because it is out of interest. This same will be done in some further formulae). Hence, a simpler form of the formula (6) is as follows

$$G(P_k)=\frac{1}{(k-1)\,k}\left[M-\frac{k+1}{2}\,V(P_k)\right] \tag{10}$$

Now, we construct a partition P_{k+1} on the basis of the partition P_k, $|X|>k\geqslant 2$. Let $P_{k+1}=\{\hat{X}_i: i=1, 2, ..., k+1\}$. We also assume that the following conditions hold

$$\hat{X}_i=X_i, \text{ for each } i=1, 2, ..., k, \; i\neq q; \tag{11}$$

$$\hat{X}_q \cup \hat{X}_{k+1}=X_q, \tag{12}$$

where $X_i \in P_k$ for each $i=1, 2, ..., k$. The index q is chosen in such a way that $|X_q|\geqslant 2$, $q=1, 2, ..., k$. Since

$$V(P_{k+1})=V(P_k)-2S(\hat{X}_q, \hat{X}_{k+1}) \tag{13}$$

then one obtains

$$G(P_{k+1})-G(P_k)=\frac{1}{2(k-1)\,k\,(k+1)}\,[(k+3)\,V(P_k)+2(k+2)(k-1)\,S_k^q-4M],$$

where we denote $S_k^q=S(\hat{X}_q, \hat{X}_{k+1})$, for short. In the similar way one can show that

$$G(P_k)-G(P_{k-1})=\frac{1}{2(k-2)(k-1)\,k}\,[(k+2)\,V(P_{k-1})+2(k+1)(k-2)\,S_{k-1}^r-4M],$$

for $3\leqslant k\leqslant |X|$, $r=1, 2, ..., k-1$, $P_{k-1} \in \mathscr{P}_{k-1}$. Here $P_{k-1}=\{\bar{X}_i: i=1, 2, ..., k-1\}$, $\bar{X}_i=X_i$ for each $i=1, 2, ..., k-1$, $i\neq r$ and $\bar{X}_r=X_r \cup X_k$, $X_r, X_k \in P_k$. Then, we immediately have the following theorem.

THEOREM 1. *Let P_{k-1}, P_k and P_{k+1} be as defined before, $3\leqslant k<|X|$. If*

$$G(P_k^*)=\max\{G(P): P \in \mathscr{P}\} \tag{14}$$

then the two following conditions must hold

$$4M\geqslant(k+3)\,V(P_k)+2(k+2)(k-1)\,S_k^q, \tag{15}$$

$$4M\leqslant(k+2)\,V(P_{k-1})+2(k+1)(k-2)\,S_{k-1}^r. \tag{16}$$

for each $q=1, 2, ..., k$ and $r=1, 2, ..., k-1$.

Let a partition $P'_{k+1}=\{X'_i: i=1, 2, ..., k+1\}$ be given, which is defined as follows: for each $i=1, 2, ..., k$, $i\neq r$, we have $X'_i=X_i\in P_k$ and $X'_r\cup X'_{k+1}=X_k\in P_k$. The index r is here arbitrarily chosen, but $r=1, 2, ..., k$. Thus, one can calculate

$$G(P_{k+1})-G(P'_{k+1})=\frac{k+2}{k(k+1)}[S(\hat{X}_q, \hat{X}_{k+1})-S(X'_r, X'_{k+1})]$$

Then, we have the following theorem

THEOREM 2. *Let P_k and P_{k+1} be in the same form as in Theorem 1, $2\leqslant k<|X|$. Moreover, let P_k maximize G over $\mathscr{P}_k$. The partition P_{k+1} fulfills the relation*

$$G(P_{k+1})=max\{G(P): P\in\mathscr{P}_{k+1}\}$$

if and only if the following condition

$$S_k^q\geqslant S(X'_r, X'_{k+1})$$

holds for each $P'_{k+1}\in\mathscr{P}_{k+1}$, where P'_{k+1} is as described above.

Theorems 1 and 2 give us a simple method for seeking the solution of the problem considered. Initially we look for the partition $P_2^*=\{X_1, X_2\}$ of the set X such that it maximizes $S(X_1, X_2)$. Then, we compute two new subpartitions (which will be called bipartitions), namely $\{X'_1, X'_1{}'\}$ and $\{X'_2, X'_2{}'\}$, where $X'_1\cup X'_1{}'=X_1$, $|X_1|\geqslant 2$, and $X'_2\cup X'_2{}'=X_2$, $|X_2|\geqslant 2$. The former one maximizes $S(X'_1, X'_1{}')$ and the second one maximizes $S(X'_2, X'_2{}')$. Then, we choose the greater number out of these two values obtained and the corresponding bipartition. Thus, we have $P_3^*=\{\bar{X}_1, \bar{X}_2, \bar{X}_3\}$, which fulfills the formula (14) for $k=3$, etc. If we do not want to attain the global maximum, then we stop when the inequalities (15) and (16) hold for some k. Otherwise we continue up to $k=|X|$.

It seems that the above described idea for solving the problem is very natural and simple. It is obvious that the most important problem is here to obtain an effective and efficient procedure for seeking a bipartition of any specified set $X, |X|\geqslant 3$, into two subsets X_1, X_2 that maximizes $S(X_1, X_2)$. This bipartition is said to be the bipartition with the greatest value. Next sections of the paper are mainly devoted to the derivation of a procedure for seeking the bipartition considered.

4. Maximal bipartitions

From now on we consider partitions of a set X into two disjoint and nonempty subsets, i.e. bipartitions. Let a partition $P=\{X_1, X_2\}$ be given, and its value, i.e. $S(X_1, X_2)$, be equal to S. If there exists no X_1^0, $X_1^0\neq X_1$, such that

$$X_1\subset X_1^0\subset X \text{ and } S(X_1^0, X-X_1^0)\geqslant S \tag{17}$$

and for every $\bar{X}\subset X_1$ we have $S(\bar{X}, X-\bar{X})\leqslant S$, then the ordered pair of sets $\langle X_1, X_2\rangle$ is said to be a maximal bipartition. Moreover, a bipartition $\{X_1, X_2\}$, where $\langle X_1, X_2\rangle$ is a given maximal bipartition is called the bipartition corresponding to a maximal

bipartition. (Note that the bipartition corresponding to a maximal bipartition and a bipartition with the greatest value can be, but need not to be the same!). The set of all the maximal bipartitions for the considered X is denoted by $\mathscr{P}^0$.

In order to simplify later notations we write

$$S_i(x)=S(\{x\}, X_i-\{x\}),\ i=1, 2; \tag{18}$$

$$S(x)=S(\{x\}, X-\{x\}) \tag{19}$$

It is evident that for any fixed bipartition $\{X_1, X_2\}$ we have

$$S_1(x)+S_2(x)=S(x) \tag{20}$$

LEMMA 1. *Let* $\langle X_1, X_2\rangle \in \mathscr{P}^0$. *The necessary and sufficient condition for* $x \in X_1$ *is to fulfil the inequality*

$$S_1(x) \leqslant S_2(x) \tag{21}$$

Proof. We assume that $x \in X_1$. Let $S_1(x)>S_2(x)$. Then, we have $S(X_1-\{x\}, X_2 \cup \cup\{x\})=S(X_1, X_2)+S_1(x)-S_2(x)>S(X_1, X_2)$, which is a contradiction, because $\langle X_1, X_2\rangle \in \mathscr{P}^0$.

Now, let the inequality (21) hold. We assume that $x \in X_2$. Then $S(X_1 \cup \{x\}, X_2+ -\{x\})=S(X_1, X_2)+S_2(x)-S_1(x)>S(X_1, X_2)$, hence a contradiction which completes the whole proof. Q.E.D. ∎

In a similar way we can prove the next lemma.

LEMMA 2. *Let* $\langle X_1, X_2\rangle$ *be a maximal bipartition. The relation* $x \in X_2$ *is equivalent to the following inequality*

$$S_1(x)>S_2(x) \tag{22}$$

Now, we formulate a more general property of maximal bipartitions.

THEOREM 3. $P=\langle X_1, X_2\rangle$ *is a maximal bipartition, if and only if the formula* (21) *holds for each* $x \in X_1$, *and each* $x \in X_2$ *satisfies the inequality* (22).

Proof. The necessity for $P \in \mathscr{P}^0$ directly results from Lemmas 1 and 2.

Now, we consider the sufficiency. We deal with a bipartition $\{X_1, X_2\}$ and assume that the relation (21) holds for each $x \in X_1$. Moreover, let the inequality (22) be fulfilled for every $x \in X_2$. We consider a set A, $A \subset X_1$, $A \neq X_1$. We have $S(X_1+ -A, X_2 \cup A)=S(X_1, X_2)+[S(X_1, A)-S(X_2, A)]-S(A, A)$. We observe that for some C, $C \subset X$,

$$S(X_1, C)-S(X_2, C)=\sum_{x \in C}[S_1(x)-S_2(x)] \tag{23}$$

In this case we obtain $[S(X_1, A)-S(X_2, A)] \leqslant 0$, due to (21), and then the bipartition $\{X_1-A, X_2 \cup A\}$ has the value not greater than $S(X_1, X_2)$. Next, we consider a set B, $B \subset X_2$, $B \neq X_2$. By a similar argument we obtain the following inequality $S(X_1 \cup B, X_2-B)<S(X_1, X_2)$, which accomplishes the proof. Q.E.D. ∎

Then, we will investigate the following algorithm.

ALGORITHM 1.

1. START.
2. Initialize the value of bipartition $S:=0$.
3. For each $x \in X$ calculate $S(x)$.
4. For each $x \in X$ take $a(x):=S(x)$.
5. Initialize the working lists $Y:=X$, $Z:=\emptyset$.
6. Take any $x \in Y$ and the corresponding $a(x)$.
7. $S:=S+a(x)$.
8. $Y:=Y-\{x\}$, $Z:=Z \cup \{x\}$.
9. For each $y \in X$ substitute $a(y):=a(y)-2f(x, y)$. Moreover, if $a(y)<0$, then $Y:=Y-\{y\}$.
10. For all the elements y from the list Z test, whether $a(y)<0$. If so, then $Z:=Z-\{y\}$, $S:=S-a(y)$ and, moreover, for each $z \in X$ substitute $a(z):=$ $=a(z)+2f(z, y)$. Otherwise, go to Step 12.
11. Test, whether $a(z) \geqslant 0$ for any $z \in X-(Z \cup Y)$. If so, then $Y:=Y \cup \{z\}$.
12. Check, whether $Y=\emptyset$. If so, then $X_1:=Z$, $X_2:=X-X_1$. Otherwise pass to Step 6.
13. STOP.

In the above description we use the symbol of substitution: =. Its meaning is here analogous as in ALGOL.

A simple numerical example for the above algorithm is given later in this section. The following lemma concerns one of the most important properties of Algorithm 1.

LEMMA 3. *Let* $\langle X_1, X_2 \rangle$ *be an ordered pair of sets generated by Algorithm* 1. *Then, for each* $x \in X_1$ *the condition* (21) *holds and each* $x \in X_2$ *fulfills the inequality* (22).

Proof. Due to the 4th, 9th, 10th and 12th steps we notice that

$$a(x)=S(x)-2\sum_{z \in X_1} f(x, z)=S(x)-2S_1(x) \tag{24}$$

From (20) and (24) it follows that $a(x)=S_2(x)-S_1(x)$.

According to Step 12 we have $X_1=Z$. From Steps 5, 8, 9, 10 and 11 it follows that the list Y contains the elements $x \in X$ such that $x \notin Z$ and $a(x) \geqslant 0$. Due to Step 12 the procedure terminates, if and only if $Y=\emptyset$. Hence, after having performed Step 13, there is no element $x \in X$ outside the list Z, such that $a(x) \geqslant 0$. Thus, we have $a(x)<0$ for each $x \in X_2=X-X_1$ (Step 12). Due to Steps 6, 8, 10 and 11 each element taken from X_1 has a nonnegative $a(x)$. It is obvious that the last property is equivalent to the formula (21). From the relation $a(x)<0$ there follows the formula (22). Q.E.D. ∎

Lemma 3 and Theorem 3 immediately imply the following property of Algorithm 1.

THEOREM 4. *Algorithm* 1 *generates a maximal bipartition.*

There can exist more than one maximal bipartition. Moreover, they may have distinct values. It is shown in the following simple example which also gives an idea how Algoritm 1 works.

Example.

Let $X=\{1, 2, 3, 4, 5, 6, 7\}$. The values of f are given in the matrix below.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 3 & 1 \\ 1 & 0 & 1 & 2 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 0 & 1 & 1 \\ 3 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Algorithm 1 is initiated in this case with $S=0$ (Step 2) and $\boldsymbol{a}=[9, 6, 5, 7, 7, 6, 6]$ (Step 4), where in the xth element of the row vector $\boldsymbol{a}$ we store the current value of $a(x)$. Let $x=1$ be taken in Step 6. Then $S=9$, $\boldsymbol{a}=[9, 4, 3, 5, 3, 0, 4]$, $Y=\{2, 3, 4, 5, 6, 7\}$ and, $Z=\{1\}$. We pass to Step 6 again and let $x=2$. Then $S=13$, $\boldsymbol{a}=[7, 4, 1, 1, 1, 0, 2]$, $Y=\{3, 4, 5, 6, 7\}$, $Z=\{1, 2\}$, and we go to Step 6. Let $x=3$. Thus $S=14$, $\boldsymbol{a}=[5, 2, 1, -1, -1, 0, 0]$, $Y=\{6, 7\}$, $Z=\{1, 2, 3\}$, and we pass to Step 6. Let $x=6$. Hence $S=14$, $\boldsymbol{a}=[-1, 2, 1, -3, -3, 0, -2]$, $Y=\emptyset$, $Z=\{1, 2, 3, 6\}$, but now $1 \in Z$ and $a(1)=-1$. After having performed Step 10 and Step 11 we obtain $S=15$, $\boldsymbol{a}=[-1, 4, 3, -1, 1, 6, 0]$, $Y=\{5, 7\}$, $Z=\{2, 3, 6\}$. We pass to Step 6 again and take $x=5$. Thus $S=16$, $\boldsymbol{a}=[-5, 2, 1, -3, 1, 4, -2]$, $Y=\emptyset$, $Z=\{2, 3, 5, 6\}$ and the procedure terminates. We obtain the maximal bipartition $\langle\{1, 4, 7\}, \{2, 3, 5, 6\}\rangle$ with the value $S=16$. In this case we have

$$\begin{aligned} & S_1(1)=2 \leqslant S_2(1)=7, \\ & S_1(4)=2 \leqslant S_2(4)=5, \\ & S_1(7)=2 \leqslant S_2(7)=4, \\ \text{and } & S_1(2)=4 > S_2(2)=2, \\ & S_1(3)=3 > S_2(3)=2, \\ & S_1(5)=4 > S_2(5)=3, \\ & S_1(6)=5 > S_2(6)=1. \end{aligned}$$

If we take the consecutive xs in Step 6 in the order 3, 4, 6, 7, then we obtain another maximal bipartition $\langle\{3, 4, 6, 7\}, \{1, 2, 5\}\rangle$ with the value $S=14$. Here we have

$$\begin{aligned} & S_1(3)=2 \leqslant S_2(3)=3, \\ & S_1(4)=3 \leqslant S_2(4)=4, \\ & S_1(6)=2 \leqslant S_2(6)=4, \\ & S_1(7)=3 \leqslant S_2(7)=3, \\ \text{and } & S_1(1)=6 > S_2(1)=3, \\ & S_1(2)=4 > S_2(2)=2, \\ & S_1(5)=4 > S_2(5)=3. \end{aligned}$$

5. Bipartitions with the greatest value

There are some situations when we have more than one bipartition with the greatest value. As a simple example we can investigate such bipartitions in the case when to each pair $\{x, y\}$ of distinct elements of X we assign $f(x, y)=f_0=$ $=\text{const}>0$. Similar situations appear in the set of maximal bipartitions for a given X, i.e. $\mathscr{P}^0$. There can exist two or more maximal bipartitions with distinct values, as it was shown in the preceding section. Now, we try to estabilish a relation between maximal bipartitions and bipartitions with the greatest value. Initially, we formulate some properties of bipartitions with the greatest value.

LEMMA 4. *Let* $\{X_1, X_2\}$ *be a bipartition with the greatest value. If* $x \in X_1$, *then the inequality* (21) *holds. If* $S_1(x)<S_2(x)$, *then* $x \in X_1$.

Proof. The proof of the first part of the lemma can be accomplished in the same way as used for proving the first part of Lemma 1.

Now, let $S_1(x)<S_2(x)$ and let $x \in X_2$. Then $S(X_1 \cup \{x\}, X_2-\{x\})=S(X_1, X_2)+$ $+[S_2(x)-S_1(x)]>S(X_1, X_2)$, hence a contradiction, which terminates the proof. Q.E.D. ■

In a similar way we obtain the next property.

LEMMA 5. *Let* $\{X_1, X_2\}$ *be a bipartition with the greatest value. If* $x \in X_2$, *then the inequality*

$$S_1(x) \geqslant S_2(x) \tag{25}$$

holds. Further, if the relation (22) *is satisfied, then* $x \in X_2$.

The above lemmas are analogous to Lemma 1 and Lemma 2. Let $S(\{x\}, A)=$ $=S(\{x\}, B)$, $A \cap B=\emptyset$, $A \cup B=X-\{x\}$, for some $x \in X$. It must be pointed out that if $\langle A \cup \{x\}, B\rangle$ is a maximal bipartition, then $\langle A, B \cup \{x\}\rangle$ cannot be a maximal bipartition due to its definition given in Section 4. On the other hand, if $\{A \cup$ $\cup \{x\}, B\}$ is a bipartition with the greatest value, then $\{A, B \cup \{x\}\}$ is also a bipartition with the greatest value, because $S(A \cup \{x\}, B)=S(A, B)+S(\{x\}, B)=$ $=S(A, B)+S(A, \{x\})$ (due to the above assumptions and the symmetry of $S(\cdot, \cdot)$), and hence $S(A \cup \{x\}, B)=S(A, B \cup \{x\})$. This is the reason for differences in weak and strict inequalities in Lemma 4, Lemma 5 and the Lemmas 1 and 2.

Here the question arises, whether any maximal bipartition is a bipartition with the greatest value. The following theorem gives the answer.

THEOREM 5. *For each bipartition with the greatest value there exists a maximal bipartition with the same value.*

Proof. Let $P=\{X_1, X_2\}$. Due to the lemmas 4 and 5 the condition (21) is satisfied for each $x \in X_1$. For $x \in X_2$ the inequality (25) also holds. If for every $x \in X_2$ we have $S_1(x) \neq S_2(x)$, i.e. the relation (22) rather than (25) is satisfied, then P is also a bipartition corresponding to a maximal bipartition which results from Theorem 3. This terminates the first part of the proof.

Now, we make further assumptions. Let A, $A \subset X_2$, $A \neq \emptyset$, be the set of all the elements for which we have

$$S_1(x) = S_2(x) \tag{26}$$

Let B, $B \subset A$, be a set, such that $S(B, B) = 0$. Since $A \neq \emptyset$, then this set is also non-empty. Indeed, if $x \in A$, then it is sufficient to take $B = \{x\}$, and obviously we have $S(\{x\}, \{x\}) = f(x, x) = 0$. Furthermore, let B be maximal in this sense that for each proper superset C of B there is $S(C, C) \neq 0$ (do not confuse it with a maximal bipartition!). Then, using the formula (23) we obtain $S(X_1 \cup B, X_2 - B) = S(X_1, X_2)$. It means that the bipartition considered is not a bipartition corresponding to a maximal bipartition. Moreover, the bipartition $\{X_1 \cup B, X_2 - B\}$ is also a bipartition with the greatest value. It remains to prove that the new bipartition is maximal.

Let us denote for simplicity $X_1 \cup B = X_1^*$ and $X_2 - B = X_2^*$. We calculate

$$S(\{x\}, X_1^* - \{x\}) = S_1^*(x) = S_1(x) + S(\{x\}, B), \tag{27}$$

$$S(\{x\}, X_2^* - \{x\}) = S_2^*(x) = S_2(x) - S(\{x\}, B), \tag{28}$$

Case 1. Let $x \in B \subset X_1^*$. Then, according to (26) and to the previously assumed properties of the set B, we have $S(\{x\}, B) = 0$. It implies that $S_1^*(x) = S_2^*(x)$ which confirms the condition (21).

Case 2. Let $x \in X_1 \subset X_1^*$. Since the bipartition $\{X_1^*, X_2^*\}$ is that with the greatest value, then Lemma 4 results in $S_1^*(x) \leqslant S_2^*(x)$, which also agrees with the condition (21).

Case 3. Let $x \in A - B \neq \emptyset$. Since B is a maximal set in the sense that $S(C, C) \neq 0$ for every C, $C \neq B$, $C \supset B$, then $S(\{x\}, B) > 0$. It follows that $S_1^*(x) > S_1(x) \geqslant S_2(x) > S_2^*(x)$, due to (25), (27) and (28), as well. Hence, $S_1^*(x) > S_2^*(x)$ which confirms the inequality (22).

Case 4. Let $x \in X_2^* - A$. We notice that $X_2^* - A = X_2 - A$. From the description of A it results that the condition (22) holds. Since $S(\{x\}, B) \geqslant 0$ then $S_1^*(x) > S_2^*(x)$ as in the previous case.

We see that the bipartion $\{X_1^*, X_2^*\}$ fulfills the assumptions of Theorem 3. Thus, there directly results from this theorem that $\langle X_1^*, X_2^* \rangle$ is also a maximal bipartition, which completes the whole proof. Q.E.D. ∎

As a natural consequence of the above theorem we obtain a simple idea for the generation a bipartition with the greatest value. It is sufficient to seek this bipartition in the set of maximal bipartitions for a given X. Let $\mathscr{P}^*$ be the set consisting of all the bipartitions corresponding to maximal bipartitions with the greatest value, i.e. $\mathscr{P}^* = \{\{X_1, X_2\} : X_1 = A, X_2 = B, \langle A, B \rangle \in \mathscr{P}^0, S(X_1, X_2) = \max \{S(A, B) : \langle A, B \rangle \in \mathscr{P}^0\}\}$. By $\mathscr{P}_2^{0*}$ we denote the set of all the bipartitions with the greatest value for a given X, i.e. $\mathscr{P}_2^{0*} = \{\{C, D\} : \{C, D\} \in \mathscr{P}_2, S(C, D) = \max \{S(A, B) : \{A, B\} \in \mathscr{P}_2\}\}$, and by $\mathscr{P}_2^*$ we denote the set of all the bipartitions maximizing the value of $G(P_2)$ over $\mathscr{P}_2$. Then we have the following corollary.

COROLLARY 1.

$$\mathscr{P}_2^* = \mathscr{P}_2^{0*} \supset \mathscr{P}^* \tag{29}$$

Proof. We notice that the maximization of $G(P_2)$ over $\mathscr{P}_2$ means the search of the minimal value of $V(P_2)$ over $\mathscr{P}_2$, due to (10). The last problem is equivalent to seeking the maximum value of $M-V(P_2)$ on $\mathscr{P}_2$. If we denote $P_2=\{X_1, X_2\}$, then the equality $M-V(P_2)=S(X_1, X_2)$ directly follows from the formulae (7) and (8). It means that $\mathscr{P}_2^*=\mathscr{P}_2^{0*}$. Then, there is evidently no possibility to construct a maximal bipartition with the value greater than the greatest value. Q.E.D. ∎

In the general case Corollary 1 gives the best result in this sense, that the inclusion cannot be replaced by the equality. It follows from the proof of Theorem 5. However, it should be mentioned that there exists some class of ordered pairs $\langle X, f\rangle$ (i.e. some class of problems defined by the set of entities X and the function of dissimilarity), for which such a replacement is possible.

6. Algorithm for the determination of a bipartition with the greatest value

Let us consider Algorithm 1 again. It is evident that the content of both sets forming a maximal bipartition depends on the method of choosing the elements from the list Y in Step 6. It seems that to be sure to obtain a maximal bipartition with the greatest value we have to generate all the permutations over the set X, i.e. $|X|!$ permutations. But this is not true. Let us assume, that the first permutation has been constructed. It results in a maximal bipartition, say $\langle X_1, X_2\rangle$. Due to Theorem 3 and Theorem 4 it is evident that for each $x \in X_1$ the inequality (21) holds and for each $x \in X_2$ the relation (22) is satisfied. According to the formula (24) the sign of $a(x)=S_2(x)-S_1(x)$ does not depend on the order of the elements in the set X_1 and the same holds for the points of the set X_2. Since any permutation on the set X_2 gives us a bipartition with the same value, then the above mentioned permutation represents the equivalence class of permutations with the same value of bipartition. From these remarks it follows that the cardinality of this equivalence class is equal at least to $2|X_1|!|X_2|!$. Then, the next permutation we have to inspect should be taken from outside of the mentioned class. This new permutation implies a new bipartition and then another equivalence class, etc. This gives the idea of exact algorithm for searching a bipartition with the greatest value. The search is based on the controlled inspection of equivalence classes of permutations rather than on the examination of the set of all the permutations. To obtain a systematic way of inspection, we can apply the method of enumerating all the permutations proposed by Even [1]. But his algorithm is not sufficiently fast for our purposes, because its efficiency is proportional to $|X|!$. Hence, we must consider some heuristic method.

We start with some remarks giving the possibility of preliminary diminishing the dimension of the problem. One can observe that if for some pair of distinct x, y the equality $S(x)=f(x, y)=S(y)$ holds, then surely either $x \in X_1$ and $y \in X_2$ or $x \in X_2$ and $y \in X_1$. Hence, for example, we can arbitrarily assume that $x \in X_1$ and $y \in X_2$, and remove x, y from X. The other special case appears when for some

$x \in X$ the relation $S(x)=0$ holds. The definition of maximal bipartition yields that such an element must belong to X_1. It means that we can start the algorithm by including this x in X_1.

Furthermore, each element x must belong either to X_1, or to X_2, for any bipartition $\{X_1, X_2\}$. Then, we can arbitrarily choose the first element for testing. For example, we can assume that this is the element x_1, which maximizes the value of $S(x)$. Many reasons support this choice. The most important one is that it makes very probable the exclusion of some elements from Y (see Step 9) in the first iteration.

The above remarks are taken into account in the construction of a heuristic algorithm. It is based on Algorithm 1. The problem of choosing the element from the working list Y is solved by taking that which maximizes the current value of $a(x)$, $x \in Y$. It assures the property of local maximum. In other words in each iteration we achieve the greatest possible increase of the value of the bipartition generated. This idea is similar to the concept of gradient methods. It also results in an increase of efficiency and speed of the algorithm. In the case when we have more than one x, $x \in Y$, with the greatest value of the current $a(x)$ we proceed as follows. We arbitrarily choose any x, say x_1, and save the other ones in the computer memory. Then, we continue the algorithm until the determination of a maximal bipartition. If some element saved before belongs to the computed X_1, then we delete it from the memory. If no element is saved, then we terminate the computations. Otherwise, we go back to the situation described before, choose a new element, say x_2, delete it from the computer memory and proceed as above, etc. After the inspection of all the possibilities we take that of the greatest value.

Let us now assume that the elements of X are distinct natural numbers. Then the above described concept can be presented in a more formal way, as follows.

A HEURISTIC ALGORITHM.

1. START.
2. Compute $S(x)$, for each $x \in X$.
3. Substitute $a(x) := S(x)$, for every $x \in X$.
4. $Q := \{x.\ x \in X,\ a(x)=0\}$.
5. $T := X-Q$.
6. $K_0 := \{x: x, y \in T,\ a(x)=f(x, y)=a(y),\ x<y\}$.
7. $K_1 := \{x: x, y \in T,\ a(x)=f(x, y)=a(y),\ x>y\}$.
8. $S_K := \sum_{x \in K_0} a(x)$.
9. $T := T-K_0-K_1$.
10. $p := 0$; $t := 0$; $w := 0$.
11. $Y_p := T$; $Z := \emptyset$, $Z_p := \emptyset$.
12. $S_p := 0$; $n_p := 0$.
13. $B_0 := \{x: x \in Y_0,\ a(x)=\max\{a(y): y \in Y_0\}\}$.
14. $q := \min\{x: x \in B_0\}$.
15. Go to Step 26.

16. $B_p: =\{x: x \in Y_p,\ a(x)=\max\{a(y): y \in Y_p\}\}$.
17. If $p=0$, then go to Step 20.
18. Determine $L_r: =B_r \cap B_p$, for each $r=0, 1, ..., p-1$.
19. $B_r: =B_r-L_r$; $n_r: =n_r-|L_r|$, for each $r=0, 1, ..., p-1$.
20. $q: =\min\{x: x \in B_p\}$.
21. $B_p: =B_p-\{q\}$.
22. Substitute $b_p: =|B_p|$.
23. If $b_p=0$, then go to Step 26.
24. $n_p: =b_p$; $n_{p+1}: =0$; $t: =t+1$; and for each $y \in T$ define $a_p(y): =a(y)$.
25. $Y_{p+1}: =Y_p$; $Z_{p+1}: =Z_p$; $S_{p+1}: =S_p$; $p: =p+1$.
26. $Y_p: =Y_p-\{q\}$; $Z_p: =Z_p \cup \{q\}$; $S_p: =S_p+a(q)$.
27. For each $y \in T$ substitute $a(y): =a(y)-2f(q, y)$ and if the new $a(y)<0$, then $Y_p: =Y_p-\{y\}$.
28. For each $y \in Z_p$ test whether $a(y)<0$, and if so, then $Z_p: =Z_p-\{y\}$; $S_p: =S_p-a(y)$; for each $z \in T$ substitute $a(z): =a(z)+2f(z, y)$, and if also $z \notin Z_p$ and $a(z) \geqslant 0$, then $Y_p: =Y_p \cup \{z\}$. Go back to checking further elements of Z_p.
29. If $Y_p \neq \emptyset$, then go to Step 16.
30. If $w>S_p$, then $S_p: =w$ and $Z_p: =Z$.
31. If $p \neq 0$ or $t \neq 0$, then go to Step 33.
32. $X_1: =Z_p \cup K_0 \cup Q$; $X_2: =T-Z_p \cup K_1$; $S: =S_p+S_K$; go to Step 41.
33. $w: =S_p$; $Z: =Z_p$.
34. If $n_p=0$, then $p: =p-1$ and $t: =t-1$.
35. $n_p: =n_{p-1}$.
36. Substitute $a(x): =a_p(x)$, for each $x \in T$.
37. $q: =\min\{x: x \in B_p\}$.
38. $B_p: =B_p-\{q\}$.
39. If $n_p>0$, then go to Step 25.
40. Go to Step 26.
41. The bipartition has the form $\{X_1, X_2\}$, and its value is equal to S.
42. STOP.

Now we present a computational example. Let $X=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23\}$. The values of f are shown in Table 1. The first run of Algorithm is shown in Table 2. The current values of $a(x)$ for the elements actually belonging to the list Z are given in brackets. The asterisk means that it is possible to have another choice than it has been made. If the elements labelled with the asterisks are finally included in X_1, then the asterisks disappear. In our example this situation refers to the 19th entity. As the result we obtain the following bipartition: $\{\{1, 4, 7, 8, 10, 11, 12, 13, 18, 19, 20, 22\}, \{2, 3, 5, 6, 9, 14, 15, 16, 17, 21, 23\}\}$, with the value $S=595$. Since we have still two asterisks, then the algorithm has to take into account two additional possibilities. First, it returns to the 10th iteration of the previous run. Now, in this iteration there is chosen the 14th entity instead of the 12th one. It results in the bipartition $\{\{1, 4, 7, 8, 10, 11, 13, 14, 17, 19, 20, 22\}, \{2, 3, 5, 6, 9, 12, 15, 16, 18, 21, 23\}\}$, with the value $S=600$.

Table 1. The values of dissimilarities

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	10	10	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	10	0	20	30	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	1	0
3	10	20	0	40	0	0	10	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	30	40	0	65	0	0	0	0	8	0	0	0	1	0	1	0	0	0	0	0	0	0
5	0	0	0	65	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	1
6	10	0	0	0	0	0	10	0	10	0	0	0	0	0	0	0	7	7	6	0	0	0	0
7	0	0	10	0	0	10	0	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	15	0	0	0	50	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	10	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	14	0	8	13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	5	5
11	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	0	0	5	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	10	0	10	10	10	10	2	3	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	10	10	0	10	10	10	0	0	0	0	7	0	0
14	0	0	0	1	0	0	0	0	0	0	10	10	10	0	10	10	0	4	4	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	10	10	10	10	0	10	0	0	0	5	0	0	0
16	0	0	0	1	0	0	0	0	0	0	10	10	10	10	10	0	0	0	0	8	0	0	0
17	0	0	0	0	0	7	0	0	0	0	0	2	0	0	0	0	0	30	20	0	0	0	5
18	0	0	0	0	0	7	0	0	0	0	0	3	0	4	0	0	30	0	20	0	0	5	0
19	0	0	0	0	0	6	0	0	0	1	5	0	0	4	0	0	20	20	0	0	5	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	8	0	0	0	0	17	0	18
21	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	5	17	0	19	0
22	0	1	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	5	0	0	19	0	18
23	0	0	0	0	1	0	0	0	0	5	0	0	0	0	0	0	5	0	0	18	0	18	0

Table 2. The performance of Heuristic Algorithm

$x=$	$S(\{x\}, X-\{x\})=$	The number of the iteration											
		1	2	3	4	5	6	7	8	9	10	11	12
		The current value of $a(x)$											
1	30	30	30	30	30	30	30	30	(30)	(30)	(30)	(30)	(30)
2	75	15	15	15	15	15	13	13	—7	—35	—35	—35	—35
3	95	15	—5	—5	—5	—5	—5	—5	—25	—25	—55	—55	—55
4	145	(145)	(145)	(145)	(145)	(145)	(145)	(145)	(145)	(129)	(129)	(129)	(129)
5	79	—51	—51	—51	—51	—51	—51	—51	—51	—77	—77	—77	—77
6	50	50	30	16	16	16	16	16	—4	—4	—4	—4	—16
7	120	120	(120)	(120)	(120)	(120)	(120)	(120)	(120)	(120)	(20)	(20)	(20)
8	115	115	15	15	15	15	15	15	15	15	(15)	(15)	(15)
9	110	110	10	10	10	10	10	10	10	10	—90	—90	—90
10	46	30	30	30	30	30	20	20	20	(20)	(20)	(20)	(18)
11	55	55	55	55	35	35	35	(35)	(35)	(35)	(35)	(15)	(5)
12	55	55	55	49	29	29	29	9	9	9	9	(9)	(9)
13	57	57	57	57	(57)	(57)	(57)	(37)	(37)	(37)	(37)	(17)	(17)
14	59	57	57	49	29	29	29	9	9	9	9	*—11	*—19
15	55	55	55	55	35	25	25	5	5	5	5	—15	—15
16	59	57	57	57	*37	*21	*21	*1	*1	*1	*1	*—19	*—19
17	64	64	64	4	4	4	4	4	4	4	4	0	—40
18	69	69	69	(69)	(69)	(69)	(59)	(59)	(59)	(59)	(59)	(53)	(13)
19	61	61	61	21	21	21	21	11	11	9	9	*9	(9)
20	48	48	48	48	48	(48)	(48)	(48)	(48)	(48)	(48)	(48)	(48)
21	48	48	48	48	34	0	—38	—38	—38	—38	—38	—38	—48
22	48	48	48	38	38	38	(38)	(38)	(38)	(28)	(28)	(28)	(28)
23	47	47	47	47	47	11	—25	—25	—25	—35	—35	—35	—35
The number of the entity added to X_1		4	7	18	13	20	22	11	1	10	8	12	19

Table 3. The performance of Heuristic Algorithm (const.)

$x=$	The number of the iteration											
	11′	12′	12_1	13′	4″	5″	6″	7″	8″	9″	10″	11″
	The current value of $a(x)$											
1	(30)	(30)	(30)	(30)	30	30	30	30	(30)	(30)	(30)	(30)
2	—35	—35	—35	—35	15	15	15	15	—5	—33	—33	—33
3	—55	—55	—55	—55	—5	—5	—5	—5	—25	—25	—55	—55
4	(127)	(127)	(127)	(127)	(143)	(143)	(143)	(143)	(143)	(127)	(127)	(127)
5	—77	—77	—77	—77	—51	—51	—53	—53	—53	—79	—79	—79
6	—4	—18	—4	—16	16	16	16	16	—4	—4	—4	—4
7	(20)	(20)	(20)	(20)	(120)	(120)	(120)	(120)	(120)	(120)	(20)	(20)
8	(15)	(15)	(15)	(15)	15	15	15	15	15	15	(15)	(15)
9	—90	—90	—90	—90	10	10	10	10	10	10	—90	—90
10	(20)	(20)	(20)	(18)	30	30	20	20	20	(20)	(20)	(20)
11	(15)	(15)	(15)	(5)	35	35	35	(35)	(35)	(35)	(35)	(15)
12	—11	—15	—9	—9	29	29	29	9	9	9	9	—11
13	(17)	(17)	(17)	(17)	37	23	23	3	3	3	3	—17
14	(9)	(9)	(17)	(9)	29	29	29	9	9	9	9	—11
15	—15	—15	—15	—15	35	35	*35	*15	*15	*15	*15	(15)
16	—19	—19	—19	—19	(57)	(57)	(57)	(37)	(37)	(37)	(37)	(17)
17	4	(4)	(64)	(24)	4	4	—6	—6	—6	—6	—6	—6
18	(51)	(—9)	—9	—49	(69)	(69)	(69)	(69)	(69)	(69)	(69)	(69)
19	1	—39	1	(1)	21	11	11	1	1	—1	—1	—1
20	(48)	(48)	(48)	(48)	32	—2	—38	—38	—38	—38	—38	—48
21	—38	—38	—38	—48	48	(48)	(48)	(48)	(48)	(48)	(48)	(48)
22	(28)	(28)	(38)	(38)	38	0	—36	—36	—36	—46	—46	—46
23	—35	—45	—45	—45	47	47	(47)	(47)	(47)	(37)	(37)	(37)
the number of the entity added to X_1	14	17	—18	19	16	21	23	11	1	10	8	15

We notice that in the case considered we delete from Z the 18th entity, which was previously included in Z (see the 12'th and the $12'_1$th iterations in Table 3). Then, we return to the conflict, which occurs in the 4th iteration of the initial run. Now, we include in Z the 16th element instead of the 13th one. The solution takes the following form: $\{\{1, 4, 7, 8, 10, 11, 15, 16, 18, 21, 23\}, \{2, 3, 5, 6, 9, 12, 13, 14, 17, 19, 20, 22\}\}$, with the value $S=601$. Hence, this result is the best one. The algorithm terminates, because there is no asterisk. Let us notice that the final result was obtained after 23 iterations. The performance of the algorithm in two additional cases is shown in Table 3.

7. Some remarks on the efficiency

First, let us consider the heuristic algorithm. If in each iteration we find in the set Y_p only one element with the greatest value of the current $a(x)$, then we have no more than $n-1$ iterations, where n denotes the cardinal number of X. Let us assume that to each pair of distinct x, y we assign the same weight, i.e. $f(x, y)=f_0=$ $=\text{const.}>0$. Thus we have no more than n iterations (neglecting the fact, that in this case we can determine the partition beforehand and with no computer algorithm). The case when for each p the inequality $|B_p-\{\min\{x: x\in B_p\}\}|>0$ holds is less probable in practice. This fact is confirmed by the results of many computations we have performed. Then, we can assume that the number of iterations is proportional to n, say is equal to bn, where b is some nonnegative constant.

Now, we return to the main problem defined in Section 2. It is solved in no more than $n-1$ steps labelled by the consecutive natural numbers starting from $k=2$. If $k=2$, then we search only one bipartition. In the kth step, $2<k<n$, we generate two bipartitions referred to some specified subsets of X, as it was mentioned in Section 3. For the last, nth, step we have in explicit form $G(P_n)=\dfrac{M}{n(n-1)}$, and then we can omit this step in our considerations. Hence, we have to perform

$$m=bn+b\sum_{k=3}^{n-1}(n_{k_1}+n_{k_2}) \tag{30}$$

iterations of the heuristic algorithm. The symbols n_{k_1} and n_{k_2} denote the cardinalities of subsets of X, which are divided in the kth step. It is obvious that in the worst case $n_{k_1}+n_{k_2}=n-k$. Thus, from the formula (30) one can obtain that

$$m\leqslant b\frac{n^2-3n+6}{2}$$

It means that in practice the number of iterations of the heuristic algorithm increases no faster than with the second power of $|X|$. We should note that this estimation was obtained making many simplifying assumptions. They concern the heuristic algorithm (see the beginning of this section) and there were given some

average estimations corresponding to a practical rather than the worst case. In this sense this quadratic estimation of the efficiency of the algorithm solving the whole problem has to be meant as an estimation for the most probable situation rather than the worst one.

8. Final remarks

The problem considered in this paper concerns a large family of problems consisting in partitioning a set into mutually disjoint subsets. There is a rich literature devoted to this subject, but there is no general method for solving this family of problems. It is a result of different nature of interconnections joining each pair in the set of entities under consideration, what was emphasized in the introduction. The solution of some problem with the interconnections of similarity type was proposed in previous papers of one of the authors [2, 3]. This paper is devoted to the problem which we partially solved. It means that we do not give any efficient and exact algorithm determining the solution, because we do not know any good exact algorithm generating a bipartition with the greatest value. We hope that the lemmas and theorems developed and proved in this paper would help to obtain an exact algorithm, which would be efficient enough from the practical point of view. The other way is to derive a fast heuristic algorithm, which gives also the estimation of exactness of the solution.

References

[1] EVEN S. Algorithmic combinatorics. The Macmillan Comp., 1973.
[2] KACPRZYK J., STAŃCZAK W. On an extension of the method of minimally interconnected subnetworks. *Control and Cybernetics* **5** (1976) 4, 61–77.
[3] KACPRZYK J., STAŃCZAK W. On a further extension of the method of minimally interconnected subnetworks, *Control and Cybernetics*, 7 (1978) 2, 17–31.
[4] MAURIN K. Analysis. I. Elements (in Polish). PWN, 1971.
[5] NOWICKI T., STAŃCZAK W. Partitioning a set of elements into subsets due to their similarity. Secondes Journées Internationales Analyse des Données et Informatique, Versaillès, 1979 (in Data Analysis and Informatics, Diday E. et al. eds., North-Holland, 1980, 583–591).

Received, October 1980.

Podział zbioru obiektów z uwagi na ich wzajemne niepodobieństwo

W artykule rozważamy zagadnienie podziału zbioru obiektów na podzbiory w oparciu o znajomość wzajemnego niepodobieństwa między obiektami. Wyznaczamy zarówno liczbę podzbiorów jak i sposób dokonywania podziału. W tym celu formalizujemy zadanie przez wprowadzenie funkcji celu, a przy okazji podajemy jej interpretację. Zadanie sprowadza się do wygenerowania takiego podziału, który maksymalizuje wartość tej funkcji. Wyprowadzamy i dowodzimy pod-

stawowe właściwości funkcji celu. Właściwości te pozwalają na zdekomponowanie problemu na dwa oddzielne podproblemy. Jeden z nich polega na kolejnym porównywaniu niewielkiej liczby rozwiązań częściowych. Drugi podproblem dotyczy sposobu tworzenia kolejnego rozwiązania częściowego na podstawie uzyskanych już rezultatów. Autorzy podają ścisłą metodę rozwiązania drugiego zadania, a także proponują szybką heurystyczną procedurę generującą rozwiązania (sub) optymalne. Przy użyciu tej procedury możemy otrzymać przybliżone rozwiązanie pierwotnego zadania w czasie proporcjonalnym do n^2, gdzie n oznacza liczbę obiektów w rozpatrywanym zbiorze.

Декомпозиция множества объектов на основании их взаимного различия

В статье рассматривается разбиение множества объектов на подмножества на основании их взаимного различия. Определяется число подмножеств и метод осуществления разбиения. С этой целью задача формализуется путем введения целевой функции и ее интерпретации. Теперь задача заключается в том, чтобы получить такую декомпозицию, для которой целевая функция принимает максимальное значение. Вводятся и доказываются основные свойства этой функции. Эти свойства позволяют провести декомпозицию задачи на две подзадачи. Одна из них заключается в сравнивании небольшого числа частных решений. Вторая подзадача сводится к определению очередного частного решения на основе полученных уже результатов.

Авторы представляют точный метод решения второй подзадачи, а также предлагают быструю эвристическую процедуру, которая генерирует (почти) оптимальные результаты. С помощью этой процедуры можно получить приблизительное решение первоначальной задачи за время, которое пропорционально n^2, где n — число объектов рассматриваемого множества.