

Working Paper No. 35  
ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and  
ECONOMIC COMMISSION FOR EUROPE  
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION  
STATISTICAL OFFICE OF THE  
EUROPEAN COMMUNITIES (EUROSTAT)**

**Joint ECE/Eurostat work session on statistical data confidentiality**  
(Luxembourg, 7-9 April 2003)

Topic (v): Risk assessment

## **A GRAPH THEORETICAL APPROACH TO RECORD LINKAGE**

**Contributed paper**

Submitted by the Statistisches Bundesamt, Germany<sup>1</sup>

---

<sup>1</sup> Prepared by Rainer Lenz (rainer.lenz@destatis.de).

# A graph theoretical approach to record linkage

Rainer Lenz

## Abstract

Let  $A$  and  $B$  be two computer files of observations, corresponding to subsets of a given set of objects. Both files contain records of variables and share a nonempty set  $\{v_1, \dots, v_k\}$  of common variables. The aim is to decide for each pair  $(a, b)$  of records taken from  $A \times B$  whether  $a$  and  $b$  belong to the same object. The paper will describe a 3-phase algorithm with polynomial run-time. In the first phase of the algorithm, for each pair  $(a, b)$  of records weights  $w_i(a, b), i = 1, \dots, k$ , are calculated componentwise to build up a complete vector-weighted, bipartite graph. In phase 2, the resulting multi-objective linear program is linearly extended and transformed into a single objective assignment problem. The idea for phase 3 is to use for this purpose two greedy heuristics and a modification of Kuhn's algorithm [1]. The paper will conclude with a small, illustrative application of record linkage methods to data of the german structure of costs survey.

## 1 Introduction

The software package  $\mu$ -ARGUS offers a variety of methods to produce safe micro-data files. Users of the package are offered a choice between methods, or they must select suitable parameters when applying a method to a data set. Regarding a particular method, or setting of parameters, two aspects have to be taken into account: information loss and data protection. An anonymization method may perform well with respect to information loss, but it may not protect the data sufficiently according to the user's requirement. In order to assess the performance of a disclosure limitation method, it is not sufficient to consider its behaviour regarding the loss of information alone. The disclosure risk associated to the protected data set must also be taken into account.

Record linkage is a practical disclosure risk assessment methodology, applicable to every masking method offered by the  $\mu$ -ARGUS package. It has therefore been decided to add a record linkage tool to the package. While work on development of this tool is still in an early stage, this paper will describe the approach currently foreseen for the implementation.

## 2 Definitions and notations

Throughout the paper we will use the following denotation: A (finite) graph  $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$  is a relational structure, consisting of a (finite) set  $V(\mathcal{G})$ , the elements of which are called **vertices** (or points), and a set  $E(\mathcal{G}) \subseteq V(\mathcal{G})^2$  of unordered pairs of vertices, called **edges** (or lines) of  $\mathcal{G}$ . We denote these sets by  $V$  or  $E$  when there is no possibility of confusion. We consider undirected graphs, fulfilling the implication  $(a, b) \in E \implies (b, a) \in E$ . That is,  $E$  determines a symmetric binary relation. The edge  $(x, y) \in E$  is said to be **incident** with the vertices  $x$  and  $y$ , and  $x, y$  to be **adjacent**.

A graph  $\mathcal{S} = (V(\mathcal{S}), E(\mathcal{S}))$  is called **subgraph** of  $\mathcal{G}$ , if  $V(\mathcal{S}) \subseteq V(\mathcal{G})$  and  $E(\mathcal{S}) \subseteq E(\mathcal{G})$  holds.

$\mathcal{G}$  is called **bipartite graph** with bipartition  $(X, Y)$ , if  $V(\mathcal{G})$  is a disjoint union  $V = X \cup Y$ , such that every edge  $e$  is incident with both an  $x \in X$  and an  $y \in Y$ . Moreover, if every  $x \in X$  is connected to every  $y \in Y$ , the graph  $\mathcal{G}$  is said to be **complete**.

A **matching**  $\mathcal{M}$  of  $\mathcal{G}$  is a subgraph with the property that no two edges are adjacent in  $\mathcal{M}$ . If  $v$  is a vertex of  $\mathcal{M}$ , then  $\mathcal{M}$  **saturates**  $v$ . Moreover, if every  $v \in V$  is saturated,  $\mathcal{M}$  is called **perfect matching**.

A **vector-weighted graph**  $\mathcal{G}$  is a graph combined with a weight function

$$\begin{aligned} w : E(\mathcal{G}) &\longrightarrow \mathbb{R}^k, \\ e &\longmapsto (w_1(e), \dots, w_k(e)), \end{aligned}$$

which maps every edge  $e$  to a  $k$ -tuple of real numbers.

## 3 Multiobjective linear programming

The first part of the algorithm translates the problem of linking records which correspond to the same object into the problem of finding a minimum perfect matching in a vector-weighted bipartite graph with bipartition  $(A, B)$ . For this purpose, we need an appropriate weight function  $w$  reflecting the “similarity”  $w(a, b)$  for each pair  $(a, b)$  of records.

Let  $k$  be the number of common variables in the records of  $A$  and  $B$ . We consider the component functions  $w_i : E \longrightarrow \mathbb{R}^+ \cup \{0\}$ ,  $i = 1, \dots, k$ , to be distances of the  $i^{\text{th}}$  variables of  $a$  and  $b$ . In this sense, a record pair can be regarded as a “good” candidate for a match, if the corresponding distance  $w(a, b)$  is small in relation to the other distances. Concerning numerical variables  $v_r$ , the standardization

$$\tilde{w}_r(a, b) := \frac{w_r(a, b) - \min_{e \in E} w_r(e)}{\max_{e \in E} w_r(e) - \min_{e \in E} w_r(e)}$$

is strongly recommended. For string variables, we use as distances the classical LEVENSTHEIN-metric (the distance of two strings  $s$  and  $t$  equals the number of deletions, insertions or substitutions required to transform  $s$  into  $t$ ), the Soundex and the more efficient method of  $n$ -grams (cf. [2,3]). For a detailed overview of these and other techniques see [5].

In the following let  $n = |A| = |B| = m$ . Otherwise consider w.l.o.g. the case  $m < n$ . We then define new objects  $b_{m+1}, \dots, b_n$  which induce new pairs  $(a_i, b_j)$  for  $i = 1, \dots, n$  and  $j = m + 1, \dots, n$ , being weighted with

$$w(a_i, b_j) := (\max_{e \in E} w_1(e), \max_{e \in E} w_2(e), \dots, \max_{e \in E} w_k(e)).$$

We obtain the multiobjective linear program described below:

$$\begin{aligned}
 \text{(MOLP)} \quad & \text{Minimize} \quad \begin{cases} \sum_{i=1}^n \sum_{j=1}^n w_1(a_i, b_j) x_{ij} \\ \sum_{i=1}^n \sum_{j=1}^n w_2(a_i, b_j) x_{ij} \\ \vdots \\ \sum_{i=1}^n \sum_{j=1}^n w_k(a_i, b_j) x_{ij} \end{cases} \\
 \text{s.t.} \quad & x_{ij} \in \{0, 1\} \quad \text{for } i, j = 1, \dots, n, \\
 & \sum_{j=1}^n x_{ij} = 1 \quad \text{for } i = 1, \dots, n \quad \text{and} \\
 & \sum_{i=1}^n x_{ij} = 1 \quad \text{for } j = 1, \dots, n.
 \end{aligned}$$

This problem can be interpreted graph-theoretically as follows: Let  $\mathcal{G}$  be a vector-weighted bipartite graph with bipartition  $(A, B)$ , where  $w(a_i, b_j)$  are the  $k$ -dimensional weights of the edges  $(a_i, b_j)$ . The constraints enforce a subgraph connecting every  $a_i$  with exactly one  $b_j$  and vice versa. That is,  $x_{ij} = 1$  if and only if  $a_i$  is connected with  $b_j$ .

**Definition 1** [4] *A perfect matching  $\mathcal{M}$  of a vector-weighted, bipartite graph is called **efficient**, if no further perfect matching  $\mathcal{M}'$  exists with*

$$w(\mathcal{M}') := (w_1(\mathcal{M}'), \dots, w_k(\mathcal{M}')) < (w_1(\mathcal{M}), \dots, w_k(\mathcal{M})) =: w(\mathcal{M}),$$

where  $w_i(\mathcal{M})$  is the sum  $\sum_{e \in \mathcal{M}} w_i(e)$  of the  $i^{\text{th}}$  components of weights of all edges contained in the matching  $\mathcal{M}$ .

In general, there is no unique solution to the (MOLP). Several efficient matchings have to be expected with pairwise non-comparable vector-weights. Therefore, the decision maker must decide between  $k$  objectives and prefers some of them to the other. This is done by use of the so-called preference functions.

## 4 Single objective assignment problem

Let  $\Lambda = (\lambda_1, \dots, \lambda_k) \in (\mathbb{R}^+)^k$  be a  $k$ -tuple of positive real numbers. For a record  $r = (r^{(1)}, \dots, r^{(s)})$ , where  $s$  is the number of all variables, let w.l.o.g. the entries  $r^{(1)}, \dots, r^{(k)}$  be the values of common variables. We define a **linear preference function**  $f_\Lambda : \mathbb{R}^k \rightarrow \mathbb{R}$  by

$$f_\Lambda(x_1, \dots, x_k) = \sum_{i=1}^k \lambda_i x_i.$$

Setting  $\sum_{i=1}^k \lambda_i = 1$  and hereby  $\lambda_k = 1 - \sum_{i=1}^{k-1} \lambda_i$ , we may reduce the set of parameters to  $\{\lambda_1, \dots, \lambda_{k-1}\}$ . The permutation  $\tau$ , defined in such a way that

$$\lambda_{\tau(1)} > \lambda_{\tau(2)} > \dots > \lambda_{\tau(k)},$$

can be understood as an individual ranking of variables by the decision maker. In the theory of multicriteria optimization, linear preference functions are used to turn a multiobjective optimization problem into a single objective one.

Let  $(E; \leq_k)$  be the partially ordered set (**poset**) of vector-weighted edges, ordered lexicographically. Every linear preference function  $f$  defines a **linear preference extension**  $(L_f(E); \leq)$  by  $e_i \leq e_j$  if and only if

$$f(w_1(e_i), \dots, w_k(e_i)) \leq f(w_1(e_j), \dots, w_k(e_j)).$$

The poset  $(L_f(E); \leq)$  is ordered linearly, since for all  $e_i, e_j \in E$  it follows either  $e_i \leq e_j$  or  $e_j < e_i$ .

**Definition 2** *A perfect matching  $\mathcal{M}$  is called a **preference matching**, if there is a preference function  $f_\Lambda$  such that  $f_\Lambda(w(\mathcal{M})) \leq f_\Lambda(w(\mathcal{M}'))$  holds for every perfect matching  $\mathcal{M}'$ .*

Note that every preference matching is an efficient matching [4]. But in general there are efficient matchings which do not result from preference functions. If it holds  $f_\Lambda(w(\mathcal{M})) = f_\Lambda(w(\mathcal{M}'))$ , then  $\mathcal{M}$  and  $\mathcal{M}'$  are said to be **equivalent** matchings.

A single edge  $(a, b)$  can be regarded as a (non perfect) matching. The preference function involves for every  $(a, b) \in A \times B$  its distance

$$f_\Lambda(w(a, b)) = \sum_{i=1}^k \lambda_i w_i(a, b).$$

This expression may be regarded as a weighted sum over all common variables. We apply  $n$ th roots in order to recollect some classical metrics, to which we will refer in

section 6. Note that the  $n$ th root defines an isotone transformation. The well-known MINKOWSKI-metrics  $L_q$  are defined by

$$d_q(x, y) = \sqrt[q]{\sum_{i=1}^k |x^{(i)} - y^{(i)}|^q}.$$

For example, in the case  $q = 1$  one obtains the **city-block metric**

$$d_1(x, y) = \sum_{i=1}^k |x^{(i)} - y^{(i)}|$$

and for  $q = 2$  the **euclidean metric**

$$d_2(x, y) = \sqrt{\sum_{i=1}^k (x^{(i)} - y^{(i)})^2}.$$

Additional weights  $\lambda_1, \dots, \lambda_k$  then yield to

$$d(x, y) = \sqrt{\sum_{i=1}^k \lambda_i (x^{(i)} - y^{(i)})^2}.$$

The latter metric is implemented using a diagonal matrix  $D = \text{diag}(\lambda_1, \dots, \lambda_k)$ , that is

$$d(x, y) = \sqrt{(x - y)^T D (x - y)}.$$

Replacing  $D$  by an arbitrary symmetric, positive definite matrix  $C$ , we obtain the **general matrix metric**

$$d(x, y) = \sqrt{(x - y)^T C (x - y)}.$$

$C$  is often chosen to be the inverse  $S^{-1}$  of the empirical covariance matrix.

Defining  $d(a, b) := f_\Lambda(w(a, b))$  and abbreviated  $d_{ij} := f_\Lambda(w(a_i, b_j))$  for the edges  $(a_i, b_j)$ ,  $i, j = 1, \dots, n$ , the problem of finding a preference matching is turned into a single objective assignment problem:

$$\begin{aligned} & \text{Minimize} && \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}, && \text{(AP)} \\ & \text{s.t.} && x_{ij} \in \{0, 1\} && \text{for } i, j = 1, \dots, n, \\ & && \sum_{j=1}^n x_{ij} = 1 && \text{for } i = 1, \dots, n \quad \text{and} \\ & && \sum_{i=1}^n x_{ij} = 1 && \text{for } j = 1, \dots, n. \end{aligned}$$

In other words, we have to look for a permutation  $\pi$  of  $\{1, \dots, n\}$ , which minimizes the sum  $\sum_{i=1}^n d_{i,\pi(i)}$ . That is, in order to solve (AP), we might produce all  $n!$  perfect matchings of  $\mathcal{G}$  and select one of minimum weight. However, this algorithm will certainly not be efficient and thus does not justify the transition from problem (MOLP) to (AP). A way out is use of the Hungarian Method, introduced in [1,6] and sketched below in section 4.2.

## 4.1 Greedy assignment procedures

We consider two different assignment procedures. For these procedures the record sets  $A$  and  $B$  do not have to be of the same size, i.e. generation of new pairs – as in section 3 – can be omitted.

INPUT: Set of distances

$$\{d(a_i, b_j) \mid i = 1, \dots, n, j = 1, \dots, m\}.$$

OUTPUT: Perfect matching  $\mathcal{M}$

### **Procedure 1:**

We present an algorithm suggested in [7], slight modified such that the resulting assignment becomes unique:

```

begin{PROC 1}
   $\mathcal{M} := \emptyset$ 
   $i := 1$ 
  While ( $i \leq n$  and  $B \neq \emptyset$ ) do
     $b' := \arg \min_{b \in B} d(a_i, b)$ 
     $\mathcal{M} := \mathcal{M} \cup \{(a_i, b')\}$ 
     $B := B \setminus \{b'\}$ 
     $i := i + 1$ 
end{PROC 1}

```

Obviously, the result depends on the enumeration of  $a_1, \dots, a_n$ . This draw back is improved by procedure 2 below, where the enumeration has a smaller impact.

**Procedure 2:**

Sort the	elements $d_{ij}$ in an ascending list $\mathbf{L}$
While $\mathbf{L}$	is nonempty do
	Consider the first element $d_{ij}$ of $\mathbf{L}$ and assign $(a_i, b_j)$ .
	Delete all elements $d_{rs}$ , where $r = i$ or $s = j$ .

Naturally, both procedures make erroneous assignments in order to simplify the implementation. Let w.l.o.g.  $a_1, \dots, a_r$  be assigned to  $b_{\pi(1)}, \dots, b_{\pi(r)}$ . In step  $r + 1$  the target object  $a$  is associated with a record  $b$  of minimum distance to  $a$ . Record  $b$  is one of the remaining  $m - r$  records in  $B$ , which have not been assigned at this stage.

The following subsection describes how to find a solution to the problem (AP), minimizing the total sum of distances.

## 4.2 Optimum assignment

We give a short description of the **Hungarian Method**, originally proposed for maximum matchings, in order to find a minimum perfect matching or a preference matching, respectively.

Let us consider a complete weighted, bipartite graph. A **feasible vertex labeling**  $l$  is a mapping from the set  $V(\mathcal{G})$  of vertices of  $\mathcal{G}$  into the real numbers, where

$$l(a) + l(b) \leq d(a, b).$$

The number  $l(v)$  is then called **label** of  $v$ . The **equality subgraph**  $\mathcal{G}_l$  is a subgraph of  $\mathcal{G}$  which includes all vertices of  $\mathcal{G}$  but only those edges  $(a, b)$  fulfilling

$$l(a) + l(b) = d(a, b).$$

A connection between equality subgraphs and matchings of minimum weight is provided by the following theorem.

**Theorem** *Let  $l$  be a feasible vertex labeling of  $\mathcal{G}$ . If the equality subgraph  $\mathcal{G}_l$  possesses a perfect matching  $\mathcal{M}$ , then  $\mathcal{M}$  is a minimum perfect matching of  $\mathcal{G}$ .*

**Proof:** Let  $\mathcal{M}$  be a perfect matching of  $\mathcal{G}_l$  and  $\mathcal{M}'$  be any perfect matching of  $\mathcal{G}$ . Then it holds

$$\begin{aligned} d(\mathcal{M}') &:= \sum_{(a,b) \in \mathcal{M}'} d(a,b) \geq \sum_{v \in V(\mathcal{G})} l(v) \quad (\text{since } \mathcal{M}' \text{ saturates all vertices}) \\ &= \sum_{(a,b) \in \mathcal{M}} d(a,b) \quad (\text{by definition of } \mathcal{M}) \\ &=: d(\mathcal{M}). \end{aligned}$$

Hence,  $\mathcal{M}$  is a minimum perfect matching of  $\mathcal{G}$ .  $\diamond$

When applying the algorithm, we use two vectors of labels,  $(l(a_1), \dots, l(a_n))$  and  $(l(b_1), \dots, l(b_n))$ , to select admissible edges. Initially, we set

$$\begin{aligned} l(a_i) &= 0 && \text{for } i = 1, \dots, n \\ \text{and } l(b_j) &= \min_{1 \leq i \leq n} d(a_i, b_j) && \text{for } j = 1, \dots, n. \end{aligned}$$

Using the concept of **augmenting paths**, we find a matching  $\mathcal{M}$  of  $\mathcal{G}_l$  which saturates as many vertices as possible. If  $\mathcal{M}$  is perfect, by the above theorem  $\mathcal{M}$  is a minimum matching of  $\mathcal{G}$  and the algorithm stops.  $\mathcal{M}$  is then uniquely determined up to equivalence. Else, if  $\mathcal{M}$  does not determine a perfect matching, we relax the values for some  $l(a)$  and  $l(b)$  so that new edges will be admissible.

## 5 Record linkage algorithm

Based on the considerations above we propose the following algorithm:

- 1) Input: Sets  $A, B$  of records.
- 2) Partition the problem into sub-problems by use of blocking variables  $B \subset \{v_1, \dots, v_k\}$ .
- 3) Calculate the distances to construct a vector-weighted bipartite graph  $\mathcal{G}$ .
- 4) Choose a linear preference extension  $(L_f; \leq)$  of the poset  $(E; \leq_k)$  of vector-weighted edges.
- 5) To solve (AP), apply alternatively
  - the Hungarian Method or
  - one of the procedures presented in section 4.1.
- 6) Output:  $(1 - 1)$ -assignment between  $A$  and  $B$ .

## 6 Illustrative example

To illustrate the record linkage algorithm previously mentioned, we consider a small example. We link original data  $A = \{a_1, \dots, a_4\}$  with masked data  $B = \{b_1, \dots, b_4\}$ , where the objects are associated by five common variables  $v_1, \dots, v_5$ .

objects\variables	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
$a_1$	14008906	755187	907264	6582133	4794809
$a_2$	14309437	673189	1179713	8111720	5407676
$a_3$	14330083	567300	920065	4871720	1667078
$a_4$	14780637	567553	1026861	5313029	3654241
$b_1$	14825332	563928	913631	4978410	1711353
$b_2$	14045802	724071	1040229	7064023	5078378
$b_3$	13945802	682110	973631	7378984	508494
$b_4$	14996199	563928	1050673	5252164	3871084

Enumeration of all 24 perfect matchings then leads to:

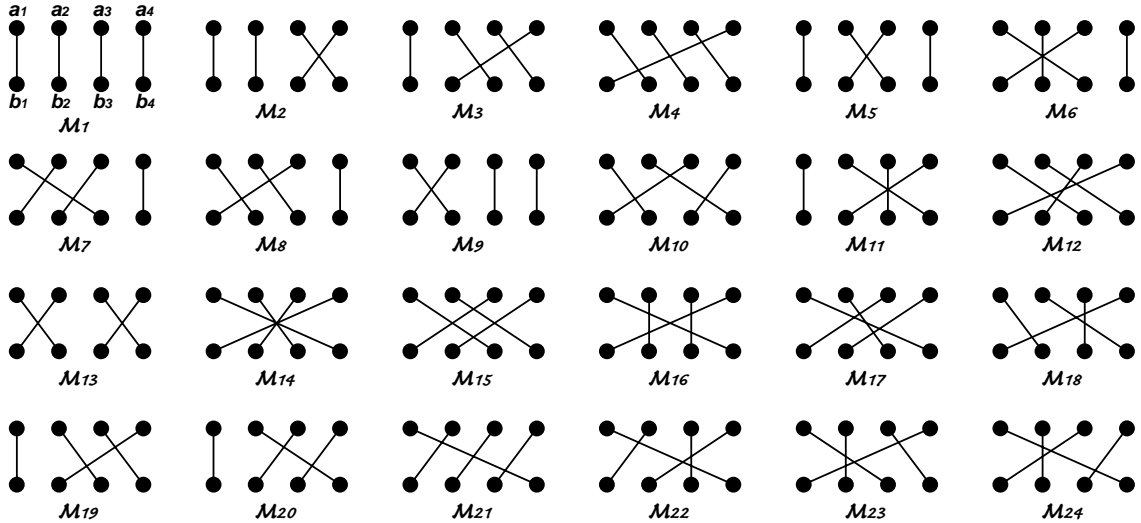


Figure 1: Perfect matchings

Using the standardized euclidean or square distances for each component, one obtains the following Hasse diagram of perfect matchings, ordered lexicographically by their vector-weights

$$w(\mathcal{M}) = \left( \sum_{(a_i, b_j) \in \mathcal{M}} w_{ij}^{(1)}, \dots, \sum_{(a_i, b_j) \in \mathcal{M}} w_{ij}^{(5)} \right), \text{ where } w(a_i, b_j) = (w_{ij}^{(1)}, \dots, w_{ij}^{(5)}).$$

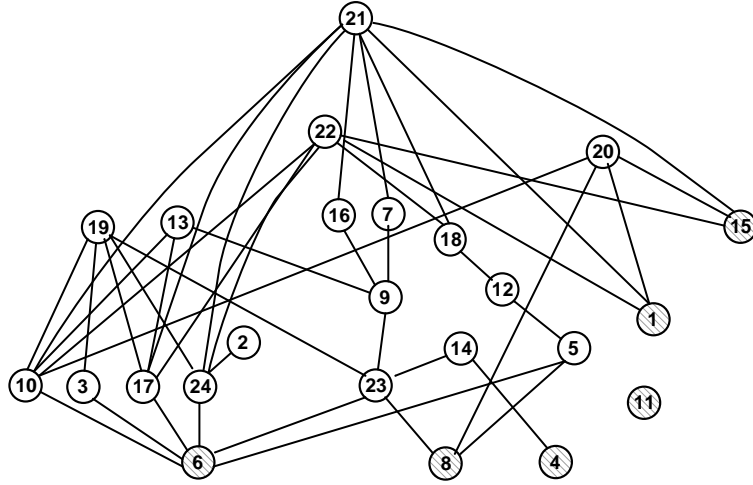


Figure 2: Poset of vector-weighted perfect matchings

The minimum elements  $\mathcal{M}_1, \mathcal{M}_4, \mathcal{M}_6, \mathcal{M}_8, \mathcal{M}_{11}$  and  $\mathcal{M}_{15}$  of this poset are efficient matchings. Taking figure 2 as a basis, we may intuitively pick  $\mathcal{M}_6$ , since it is covered by all inefficient perfect matchings. This intuition is confirmed below.

We standardize the metrics and consider  $\Lambda = (\frac{1}{5}, \dots, \frac{1}{5})$ , that is, no objective is preferred to the other. The following table compares the results obtained by combining the metrics  $L_1, L_2$  with the procedures **PROC 1**, **PROC 2** and the **Hungarian Method**. The first entry in each cell refers to the number of true assignments, the second to the total sum of distances.

metrics \ procedure	<b>PROC 1</b>	<b>PROC 2</b>	<b>HM</b>
$L_1$	2; 2.51	4; 2.33	4; 2.33
$L_2$	4; 1.34	4; 1.34	4; 1.34

Nearly all combinations led to the true assignment  $\mathcal{M}_6$ . By modification of  $\Lambda$ , application of the corresponding  $f_\Lambda$  to the vector-weights and choice of the Hungarian Method, the decision maker is able to find another efficient matching, e.g.  $\Lambda_1 = (\frac{1}{3}, 0, \frac{2}{3}, 0, 0)$ ,  $\Lambda_4 = (0, 1, 0, 0, 0)$ ,  $\Lambda_8 = (0, 0, 0, 1, 0)$ ,  $\Lambda_{11} = (0, 0, 1, 0, 0)$  and  $\Lambda_{15} = (0, 0, \frac{2}{3}, 0, \frac{1}{3})$  yield the efficient matchings  $\mathcal{M}_1, \mathcal{M}_4, \mathcal{M}_8, \mathcal{M}_{11}$  and  $\mathcal{M}_{15}$ .

In practice, however, we expect even for files of medium size very few true matches. Since the Hungarian Method finds an optimum assignment w.r.t. the distances, the success of the algorithm essentially depends on the quality of the estimated distances, which are in general not very reliable. Therefore, it is of particular relevance to consider other techniques to determine distances, such as the probabilistic EM-method [8], studied in this context in [2].

This work was partially supported by the EU project IST-2000-25069 Computational Aspects of Statistical Confidentiality.

The author gratefully acknowledges Sarah Gießing for many helpful comments.

## References

- [1] H.W.Kuhn, "The hungarian method for the assignment problem", *Naval Res. Logist. Quart.* **2** (1955), pp. 83-97.
- [2] J.Domingo-Ferrer, V.Torra, "Record linkage methods for multidatabase data mining", *Information Fusion in Data Mining, Springer-Verlag, Berlin* (2003), pp. 99-130 (in press), ISSN 1434-9922, ISBN 3-540-00676-1.
- [3] M.A.Jaro, "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida", *Journal of the American Statistical Association* **89** (1989), pp. 415-435.
- [4] D.Schweigert, "Vector Weighted Matchings", *Combinatorial Advances (ed. C.J.Colbourn, E.S.Mahmoodian), Kluwer* 1995, pp. 267-276.
- [5] G.A.Stephen, "String Searching Algorithms", *World Scientific Publishing*, 1994.
- [6] J.Munkres, "Algorithms for the assignment and transportation problem", *L. Soc. Indust. Appl. Math.* **5** (1957), pp. 32-38.
- [7] D.Pagliuca, G.Seri, "Some Results of Individual Ranking Method on the System of Enterprise Accounts Annual Survey", *Esprit SDC Project, Deliverable MI-3/D2*, 1976.
- [8] A.P.Dempster, N.M.Laird, D.B.Rubin, "Maximum Likelihood From Incomplete Data Via the EM Algorithm", *Journal of the Royal Statistical Society* **39** (1971), pp. 1-38.
- [9] D.Schweigert, "Order and Clustering", *Human centered processes, 10<sup>th</sup> Mini Euro Conference*, Brest 1999.
- [10] I.P.Fellegi, A.B.Sunter, "A Theory for Record Linkage", *Journal of the American Statistical Association* **64** (1969), pp. 1183-1210.

Rainer Lenz, Statistisches Bundesamt, Gustav-Stresemann-Ring 11, 65189 Wiesbaden, Germany, *e-mail*: rainer.lenz@destatis.de.