Decipherment Complexity in 1:1 Substitution Ciphers

Malte Nuhn and **Hermann Ney**
Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University, Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

In this paper we show that even for the case of 1:1 substitution ciphers—which encipher plaintext symbols by exchanging them with a unique substitute—finding the optimal decipherment with respect to a bigram language model is NP-hard. We show that in this case the decipherment problem is equivalent to the quadratic assignment problem (QAP). To the best of our knowledge, this connection between the QAP and the decipherment problem has not been known in the literature before.

1 Introduction

The decipherment approach for MT has recently gained popularity for training and adapting translation models using only monolingual data. The general idea is to find those translation model parameters that maximize the probability of the translations of a given source text in a given language model of the target language.

In general, the process of translation has a wide range of phenomena like substitution and reordering of words and phrases. In this paper we only study models that substitute tokens—i.e. words or letters—with a unique substitute. It therefore serves as a very basic case for decipherment and machine translation.

Multiple techniques like integer linear programming (ILP), A^* search, genetic algorithms, and Bayesian inference have been used to tackle the decipherment problem for 1:1 substitution ciphers. The existence of such a variety of different approaches for solving the same problem already shows that there is no obvious way to solve the problem optimally.

In this paper we show that decipherment of 1:1 substitution ciphers is indeed NP-hard and thus explain why there is no single best approach to the problem. The literature on decipherment provides surprisingly little on the analysis of the complexity of the decipherment problem. This might be related to the fact that a statistical formulation of the decipherment problem has not been analyzed with respect to n-gram language models: This paper shows the close relationship of the decipherment problem to the quadratic assignment problem. To the best of our knowledge the connection between the decipherment problem and the quadratic assignment problem was not known.

The remainder of this paper is structured as follows: In Section 2 we review related work. Section 3 introduces the decipherment problem and describes the notation and definitions used throughout this paper. In Section 4 we show that decipherment using a unigram language model corresponds to solving a linear sum assignment problem (LSAP). Section 5 shows the connection between the quadratic assignment problem and decipherment using a bigram language model. Here we also give a reduction of the traveling salesman problem (TSP) to the decipherment problem to highlight the additional complexity in the decipherment problem.

2 Related Work

In recent years a large number of publications on the automatic decipherment of substitution ciphers has been published. These publications were mostly dominated by rather heuristic methods and did not provide a theoretical analysis of the complexity of the decipherment problem: (Knight and Yamada, 1999) and (Knight et al., 2006) use the EM algorithm for various decipherment problems, like e.g. word substitution ciphers. (Ravi and Knight, 2008) and (Corlett and Penn, 2010) are able to obtain optimal (i.e. without search errors) decipherments of short cryptograms given an n-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 615–621,
Sofia, Bulgaria, August 4-9 2013.

gram language model. (Ravi and Knight, 2011), (Nuhn et al., 2012), and (Dou and Knight, 2012) treat natural language translation as a deciphering problem including phenomena like reordering, insertion, and deletion and are able to train translation models using only monolingual data.

In this paper we will show the connection between the decipherment problem and the linear sum assignment problem as well as the quadratic assignment problem: Regarding the linear sum assignment problem we will make use of definitions presented in (Burkard and ela, 1999). Concerning the quadratic assignment problem we will use basic definitions from (Beckmann and Koopmans, 1957). Further (Burkard et al., 1998) gives a good overview over the quadratic assignment problem, including different formulations, solution methods, and an analysis of computational complexity. The paper also references a vast amount of further literature that might be interesting for future research.

3 Definitions

In the following we will use the machine translation notation and denote the **ciphertext** with $f_1^N = f_1 \dots f_j \dots f_N$ which consists of cipher tokens $f_j \in V_f$. We denote the **plaintext** with $e_1^N = e_1 \dots e_i \dots e_N$ (and its vocabulary V_e respectively). We define

$$e_0 = f_0 = e_{N+1} = f_{N+1} = \$ \tag{1}$$

with "$" being a special sentence boundary token. We use the abbreviations $\overline{V}_e = V_e \cup \{\$\}$ and $\overline{V}_f$ respectively.

A **general substitution cipher** uses a table $s(e|f)$ which contains for each cipher token f a probability that the token f is substituted with the plaintext token e. Such a table for substituting cipher tokens $\{A, B, C, D\}$ with plaintext tokens $\{a, b, c, d\}$ could for example look like

	a	b	c	d
A	0.1	0.2	0.3	0.4
B	0.4	0.2	0.1	0.3
C	0.4	0.1	0.2	0.3
D	0.3	0.4	0.2	0.1

The **1:1 substitution cipher** encrypts a given plaintext into a ciphertext by replacing each plaintext token with a unique substitute: This means that the table $s(e|f)$ contains all zeroes, except for one "1.0" per $f \in V_f$ and one "1.0" per $e \in V_e$. For example the text

abadcab

would be enciphered to

BCBADBC

when using the substitution

	a	b	c	d
A	0	0	0	1
B	1	0	0	0
C	0	1	0	0
D	0	0	1	0

We formalize the 1:1 substitutions with a bijective function $\phi : V_f \rightarrow V_e$. The general **decipherment goal** is to obtain a mapping ϕ such that the probability of the deciphered text is maximal:

$$\hat{\phi} = \arg\max_{\phi} p(\phi(f_1)\phi(f_2)\phi(f_3)...\phi(f_N)) \tag{2}$$

Here $p(\dots)$ denotes the **language model**. Depending on the structure of the language model Equation 2 can be further simplified.

Given a ciphertext f_1^N, we define the **unigram count** N_f of $f \in \overline{V}_f$ as[1]

$$N_f = \sum_{i=0}^{N+1} \delta(f, f_i) \tag{3}$$

This implies that N_f are integer counts > 0. We similarly define the **bigram count** $N_{ff'}$ of $f, f' \in \overline{V}_f$ as

$$N_{ff'} = \sum_{i=1}^{N+1} \delta(f, f_{i-1}) \cdot \delta(f', f_i) \tag{4}$$

This definition implies that

(a) $N_{ff'}$ are integer counts > 0 of bigrams found in the ciphertext f_1^N.

(b) Given the first and last token of the cipher f_1 and f_N, the bigram counts involving the sentence boundary token $ need to fulfill

$$N_{\$f} = \delta(f, f_1) \tag{5}$$
$$N_{f\$} = \delta(f, f_N) \tag{6}$$

(c) For all $f \in V_f$

$$\sum_{f' \in V_f} N_{ff'} = \sum_{f' \in V_f} N_{f'f} \tag{7}$$

must hold.

[1] Here δ denotes the Kronecker delta.

Similarly, we define language model matrices S for the unigram and the bigram case. The **unigram language model** S_f is defined as

$$S_f = \log p(f) \tag{8}$$

which implies that

(a) S_f are real numbers with

$$S_f \in [-\infty, 0] \tag{9}$$

(b) The following normalization constraint holds:

$$\sum_{f \in V_f} \exp(S_f) = 1 \tag{10}$$

Similarly for the **bigram language model matrix** $S_{ff'}$, we define

$$S_{ff'} = \log p(f'|f) \tag{11}$$

This definition implies that

(a) $S_{ff'}$ are real numbers with

$$S_{ff'} \in [-\infty, 0] \tag{12}$$

(b) For the sentence boundary symbol, it holds that

$$S_{\$\$} = -\infty \tag{13}$$

(c) For all $f \in \overline{V_f}$ the following normalization constraint holds:

$$\sum_{f' \in \overline{V_f}} \exp(S_{ff'}) = 1 \tag{14}$$

4 Decipherment Using Unigram LMs

4.1 Problem Definition

When using a unigram language model, Equation 2 simplifies to finding

$$\hat{\phi} = \arg\max_{\phi} \prod_{i=1}^{N} p(\phi(f_i)) \tag{15}$$

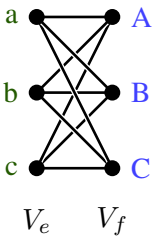
which can be rewritten as

$$\hat{\phi} = \arg\max_{\phi} \sum_{f \in V_f} N_f S_{\phi(f)} \tag{16}$$

When defining $c_{ff'} = N_f \log p(f')$, for $f, f' \in V_f$, Equation 16 can be brought into the form of

$$\hat{\phi} = \arg\max_{\phi} \sum_{f \in V_f} c_{f\phi(f)} \tag{17}$$

Figure 1 shows an illustration of this problem.



c_{ij}	A	B	C
a	$N_A \log p(a)$	$N_B \log p(a)$	$N_C \log p(a)$
b	$N_A \log p(b)$	$N_B \log p(b)$	$N_C \log p(b)$
c	$N_A \log p(c)$	$N_B \log p(c)$	$N_C \log p(c)$

Figure 1: Linear sum assignment problem for a cipher with $V_e = \{a, b, c\}$, $V_f = \{A, B, C\}$, unigram counts N_f, and unigram probabilities $p(e)$.

4.2 The Linear Sum Assignment Problem

The practical problem behind the linear sum assignment problem can be described as follows: Given jobs $\{j_1, \ldots, j_n\}$ and workers $\{w_1, \ldots, w_n\}$, the task is to assign each job j_i to a worker w_j. Each assignment incurs a cost c_{ij} and the total cost for assigning all jobs and workers is to be minimized.

This can be formalized as finding the assignment

$$\hat{\phi} = \arg\min_{\phi} \sum_{i=1}^{n} c_{i\phi(i)} \tag{18}$$

The general LSAP can be solved in polynomial time using the Hungarian algorithm (Kuhn, 1955). However, since the matrix c_{ij} occurring for the decipherment using a unigram language model can be represented as the product $c_{ij} = a_i \cdot b_j$ the decipherment problem can be solved more easily: In the Section "Optimal Matching", (Bauer, 2010) shows that in this case the optimal assignment is found by sorting the jobs j_i by a_i and workers w_j by b_j and then assigning the jobs j_i to workers w_j that have the same rank in the respective sorted lists. Sorting and then assigning the elements can be done in $\mathcal{O}(n \log n)$.

5 Decipherment Using Bigram LMs

5.1 Problem Definition

When using a 2-gram language model, Equation 2 simplifies to

$$\hat{\phi} = \arg\max_{\phi} \left\{ \prod_{j=1}^{N+1} p(\phi(f_j)|\phi(f_{j-1})) \right\} \tag{19}$$

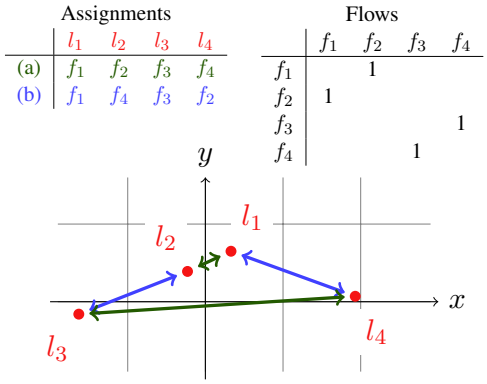


Figure 2: Hypothetical quadratic assignment problem with locations $l_1 \dots l_4$ and facilities $f_1 \dots f_4$ with all flows being zero except $f_1 \leftrightarrow f_2$ and $f_3 \leftrightarrow f_4$. The distance between locations $l_1 \dots l_4$ is implicitly given by the locations in the plane, implying a euclidean metric. Two example assignments (a) and (b) are shown, with (b) having the lower overall costs.

Using the definitions from Section 3, Equation 19 can be rewritten as

$$\hat{\phi} = \arg\max_{\phi} \left\{ \sum_{f \in V_f} \sum_{f' \in V_f} N_{ff'} S_{\phi(f)\phi(f')} \right\} \quad (20)$$

(Bauer, 2010) arrives at a similar optimization problem for the "combined method of frequency matching" using bigrams and mentions that it can be seen as a combinatorial problem for which an efficient way of solving is not known. However, he does not mention the close connection to the quadratic assignment problem.

5.2 The Quadratic Assignment Problem

The quadratic assignment problem was introduced by (Beckmann and Koopmans, 1957) for the following real-life problem:

Given a set of facilities $\{f_1, \dots, f_n\}$ and a set of locations $\{l_1, \dots, l_n\}$ with distances for each pair of locations, and flows for each pair of facilities (e.g. the amount of supplies to be transported between a pair of facilities) the problem is to assign the facilities to locations such that the sum of the distances multiplied by the corresponding flows (which can be interpreted as total transportation costs) is minimized. This is visualized in Figure 2.

Following (Beckmann and Koopmans, 1957) we can express the quadratic assignment problem as finding

$$\hat{\phi} = \arg\min_{\phi} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\phi(i)\phi(j)} + \sum_{i=1}^{n} c_{i\phi(i)} \right\} \quad (21)$$

where $A = (a_{ij}), B = (b_{ij}), C = (c_{ij}) \in \mathbb{N}^{n \times n}$ and ϕ a permutation

$$\phi : \{1, \dots, n\} \to \{1, \dots, n\}. \quad (22)$$

This formulation is often referred to as Koopman-Beckman QAP and often abbreviated as $QAP(A, B, C)$. The so-called *pure* or *homogeneous* QAP

$$\hat{\phi} = \arg\min_{\phi} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\phi(i)\phi(j)} \right\} \quad (23)$$

is obtained by setting $c_{ij} = 0$, and is often denoted as $QAP(A, B)$.

In terms of the real-life problem presented in (Beckmann and Koopmans, 1957) the matrix A can be interpreted as distance matrix for locations $\{l_1 \dots l_n\}$ and B as flow matrix for facilities $\{f_1 \dots f_n\}$.

(Sahni and Gonzalez, 1976) show that the quadratic assignment problem is strongly NP-hard.

We will now show the relation between the quadratic assignment problem and the decipherment problem.

5.3 Decipherment Problem $\preceq$ Quadratic Assignment Problem

Every decipherment problem is directly a quadratic assignment problem, since the matrices $N_{ff'}$ and $S_{ff'}$ are just special cases of the general matrices A and B required for the quadratic assignment problem. Thus a reduction from the decipherment problem to the quadratic assignment problem is trivial. This means that all algorithms capable of solving QAPs can directly be used to solve the decipherment problem.

5.4 Quadratic Assignment Problem $\preceq$ Decipherment Problem

Given $QAP(A, B)$ with integer matrices $A = (a_{ij})$, $B = (b_{ij})$ $i, j \in \{1, \dots, n\}$ we construct the count matrix $N_{ff'}$ and language model matrix $S_{ff'}$ in such a way that the solution for the decipherment problem implies the solution to the

quadratic assignment problem, and vice versa. We will use the vocabularies $\overline{V}_e = \overline{V}_f = \{1, \ldots, n+3\}$, with $n+3$ being the special sentence boundary token "$". The construction of $N_{ff'}$ and $S_{ff'}$ is shown in Figure 3.

To show the validity of our construction, we will

1. Show that $N_{ff'}$ is a valid count matrix.
2. Show that $S_{ff'}$ is a valid bigram language model matrix.
3. Show that the decipherment problem and the newly constructed quadratic assignment problem are equivalent.

We start by showing that $N_{ff'}$ is a valid count matrix:

(a) By construction, $N_{ff'}$ has integer counts that are greater or equal to 0.

(b) By construction, $N_{ff'}$ at boundaries is:

- $N_{\$f} = \delta(f, 1)$
- $N_{f\$} = \delta(f, n+2)$

(c) Regarding the properties $\sum_{f'} N_{ff'} = \sum_{f'} N_{f'f}$:

- For all $f \in \{1, \ldots, n\}$ the count properties are equivalent to

$$\tilde{a}_{f*} + \sum_{f'} \tilde{a}_{ff'} = \tilde{a}_{*f} + \sum_{f'} \tilde{a}_{f'f} + \delta(f, 1) \tag{24}$$

  which holds by construction of $\tilde{a}_{*f}$ and $\tilde{a}_{f*}$.
- For $f = n+1$ the count property is equivalent to

$$1 + \sum_{f'} \tilde{a}_{f'*} = 2 + \sum_{f'} \tilde{a}_{*f'} \tag{25}$$

  which follows from Equation (24) by summing over all $f \in \{1, \ldots, n\}$.
- For $f = n+2$ and $f = n+3$, the condition is fulfilled by construction.

We now show that $S_{ff'}$ is a valid bigram language model matrix:

(a) By construction, $S_{ff'} \in [-\infty, 0]$ holds.

(b) By construction, $S_{\$\$} = -\infty$ holds.

(c) By the construction of $\tilde{b}_{f*}$, the values $S_{ff'}$ fulfill $\sum_{f'} \exp(S_{ff'}) = 1$ for all f. This works since all entries $\tilde{b}_{ff'}$ are chosen to be smaller than $-log(n+2)$.

We now show the equivalence of the quadratic assignment problem and the newly constructed decipherment problem. For this we will use the definitions

$$\tilde{A} = \{1, \ldots, n\} \tag{26}$$
$$\tilde{B} = \{n+1, n+2, n+3\} \tag{27}$$

We first show that solutions of the constructed decipherment problem with score $> -\infty$ fulfill $\phi(f) = f$ for $f \in \tilde{B}$.

All mappings ϕ, with $\phi(f) = f'$ for any $f \in \tilde{A}$ and $f' \in \tilde{B}$ will induce a score of $-\infty$ since for $f \in \tilde{A}$ all $N_{ff} > 0$ and $S_{f'f'} = -\infty$ for $f' \in \tilde{B}$. Thus any ϕ with score $> -\infty$ will fulfill $\phi(f) \in \tilde{B}$ for $f \in \tilde{B}$. Further, by enumerating all *six* possible permutations, it can be seen that only the ϕ with $\phi(f) = f$ for $f \in \tilde{B}$ induces a score of $> -\infty$. Thus we can rewrite

$$\sum_{f=1}^{n+3} \sum_{f'=1}^{n+3} N_{ff'} S_{\phi(f)\phi(f')} \tag{28}$$

to

$$\underbrace{\sum_{f \in \tilde{A}} \sum_{f \in \tilde{A}} N_{ff'} S_{\phi(f)\phi(f')}}_{(AA)} + \underbrace{\sum_{f \in \tilde{A}} \sum_{f' \in \tilde{B}} N_{ff'} S_{\phi(f)f'}}_{(AB)} + \underbrace{\sum_{f \in \tilde{B}} \sum_{f' \in \tilde{A}} N_{ff'} S_{f\phi(f')}}_{(BA)} + \underbrace{\sum_{f \in \tilde{B}} \sum_{f' \in \tilde{B}} N_{ff'} S_{ff'}}_{(BB)}$$

Here

- (AB) is independent of ϕ since

$$\forall f \in \tilde{A}, f' \in \{n+1, n+3\} : S_{ff'} = S_{1f'} \tag{29}$$

  and

$$\forall f \in \tilde{A} : N_{f,n+2} = 0 \tag{30}$$

- (BA) is independent of ϕ since

$$\forall f' \in \tilde{A}, f \in \tilde{B} : S_{ff'} = S_{f1} \tag{31}$$

- (BB) is independent of ϕ.

$$N_{ff'} = \left(\begin{array}{cccc|c|c|c} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} & \tilde{a}_{1*} & 0 & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} & \tilde{a}_{2*} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} & \tilde{a}_{n*} & 0 & 0 \\ \hline \tilde{a}_{*1} & \tilde{a}_{*2} & \cdots & \tilde{a}_{*n} & 0 & 2 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & \cdots & 0 & 0 & 0 & 0 \end{array}\right)$$

$$S_{ff'} = \left(\begin{array}{cccc|c|c|c} \tilde{b}_{11} & \tilde{b}_{12} & \cdots & \tilde{b}_{1n} & \varepsilon_2 & \tilde{b}_{1*} & \varepsilon_2 \\ \tilde{b}_{21} & \tilde{b}_{22} & \cdots & \tilde{b}_{2n} & \varepsilon_2 & \tilde{b}_{2*} & \varepsilon_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \tilde{b}_{n1} & \tilde{b}_{n2} & \cdots & \tilde{b}_{nn} & \varepsilon_2 & \tilde{b}_{n*} & \varepsilon_2 \\ \hline \varepsilon_1 & \varepsilon_1 & \cdots & \varepsilon_1 & -\infty & \varepsilon_1 & -\infty \\ \varepsilon_2 & \varepsilon_2 & \cdots & \varepsilon_2 & \varepsilon_2 & -\infty & \varepsilon_2 \\ \hline \varepsilon_0 & \varepsilon_0 & \cdots & \varepsilon_0 & -\infty & -\infty & -\infty \end{array}\right)$$

$$\tilde{a}_{ff'} = a_{ff'} - \min_{\tilde{f}\tilde{f}'}\{a_{\tilde{f}\tilde{f}'}\} + 1$$

$$\tilde{a}_{f*} = \max\left\{\sum_{f'=1}^{n} a_{f'f} - a_{ff'}, 0\right\} + \delta(f,1)$$

$$\tilde{a}_{*f'} = \max\left\{\sum_{f=1}^{n} a_{ff'} - a_{f'f}, 0\right\}$$

$$\tilde{b}_{ff'} = b_{ff'} - \max_{\tilde{f}\tilde{f}'}\{b_{\tilde{f}\tilde{f}'}\} - \log(n+2)$$

$$\tilde{b}_{f*} = \log\left(1 - \sum_{f'=1}^{n} \exp(\tilde{b}_{ff'}) - \frac{2}{n+2}\right)$$

$$\varepsilon_i = -\log(n+i)$$

Figure 3: Construction of matrices $N_{ff'}$ and $S_{ff'}$ of the decipherment problem from matrices $A = (a_{ij})$ and $B = (b_{ij})$ of the quadratic assignment problem $QAP(A, B)$.

Thus, with some constant c, we can finally rewrite Equation 28 as

$$c + \sum_{f=1}^{n}\sum_{f'=1}^{n} N_{ff'} S_{\phi(f)\phi(f')} \tag{32}$$

Inserting the definition of $N_{ff'}$ and $S_{ff'}$ (simplified using constants c', and c'') we obtain

$$c + \sum_{f=1}^{n}\sum_{f'=1}^{n} (a_{ff'} + c')(b_{\phi(f)\phi(f')} + c'') \tag{33}$$

which is equivalent to the original quadratic assignment problem

$$\arg\max\left\{\sum_{f=1}^{n}\sum_{f'=1}^{n} a_{ff'} b_{\phi(f)\phi(f')}\right\} \tag{34}$$

Thus we have shown that a solution to the quadratic assignment problem in Equation 34 is a solution to the decipherment problem in Equation 20 and vice versa. Assuming that calculating elementary functions can be done in $\mathcal{O}(1)$, setting up $N_{ff'}$ and $S_{ff'}$ can be done in polynomial time.[2] Thus we have given a polynomial time reduction from the quadratic assignment problem to the decipherment problem: Since the quadratic assignment problem is NP-hard, it follows that the decipherment problem is NP-hard, too.

5.5 Traveling Salesman Problem $\preceq$ Decipherment Problem

Using the above construction we can immediately construct a decipherment problem that is equivalent to the traveling salesman problem by using the quadratic assignment problem formulation of the traveling salesman problem.

Without loss of generality[3] we assume that the TSP's distance matrix fulfills the constraints of a bigram language model matrix $S_{ff'}$. Then the count matrix $N_{ff'}$ needs to be chosen as

$$N_{ff'} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \tag{35}$$

which fulfills the constraints of a bigram count matrix.

[2] This is the case if we only require a fixed number of digits precision for the log and exp operations.

[3] The general case can be covered using the reduction shown in Section 5.

This matrix corresponds to a ciphertext of the form

$$\$abcd\$ \tag{36}$$

and represents the tour of the traveling salesman in an intuitive way. The mapping ϕ then only decides in which order the cities are visited, and only costs between two successive cities are counted.

This shows that the TSP is only a special case of the decipherment problem.

6 Conclusion

We have shown the correspondence between solving 1:1 substitution ciphers and the linear sum assignment problem and the quadratic assignment problem: When using unigram language models, the decipherment problem is equivalent to the linear sum assignment problem and solvable in polynomial time. For a bigram language model, the decipherment problem is equivalent to the quadratic assignment problem and is NP-hard.

We also pointed out that all available algorithms for the quadratic assignment problem can be directly used to solve the decipherment problem.

To the best of our knowledge, this correspondence between the decipherment problem and the quadratic assignment problem has not been known previous to our work.

Acknowledgements

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

Friedrich L. Bauer. 2010. *Decrypted Secrets: Methods and Maxims of Cryptology*. Springer, 4th edition.

Martin J. Beckmann and Tjalling C. Koopmans. 1957. Assignment problems and the location of economic activities. *Econometrica*, 25(4):53–76.

Rainer E. Burkard and Eranda ela. 1999. Linear assignment problems and extensions. In *Handbook of Combinatorial Optimization - Supplement Volume A*, pages 75–149. Kluwer Academic Publishers.

Rainer E. Burkard, Eranda ela, Panos M. Pardalos, and Leonidas S. Pitsoulis. 1998. The quadratic assignment problem. In *Handbook of Combinatorial Optimization*, pages 241–338. Kluwer Academic Publishers.

Eric Corlett and Gerald Penn. 2010. An exact A* method for deciphering letter-substitution ciphers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1040–1047, Uppsala, Sweden, July. The Association for Computer Linguistics.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 266–275, Jeju Island, Korea, July. Association for Computational Linguistics.

Kevin Knight and Kenji Yamada. 1999. A computational approach to deciphering unknown scripts. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, number 1, pages 37–44. Association for Computational Linguistics.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the Conference on Computational Linguistics and Association of Computation Linguistics (COLING/ACL) Main Conference Poster Sessions*, pages 499–506, Sydney, Australia, July. Association for Computational Linguistics.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2(1-2):83–97.

Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 156–164, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 812–819, Honolulu, Hawaii. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 12–21, Portland, Oregon, USA, June. Association for Computational Linguistics.

Sartaj Sahni and Teofilo Gonzalez. 1976. P-complete approximation problems. *Journal of the Association for Computing Machinery (JACM)*, 23(3):555–565, July.