Charla invitada, Dpto. Matemática Aplicada,
a las 12:15 del lunes 12 de Septiembre de 2016
en la Sala de Grados B de Telecomunicaciones

Abstract:

Integer Linear Programming for Sequence Problems: A general approach to reduce the problem size

Peter Zörnig
University of Brasília, Department of Statistics
(e-mail: peter@unb.br, or peter.zoernig@gmx.net

Sequence problems belong to the most challenging interdisciplinary topics of the actuality. They are ubiquitous in science and daily life and occur, for example, in form of DNA sequences encoding all information of an organism, as a text (natural or formal) or in form of a computer program. Therefore, sequence problems occur in many variations in computational biology (drug development), coding theory, data compression, quantitative and computational linguistics (e.g. machine translation).

In recent years appeared some proposals to formulate sequence problems like the closest string problem (CSP) and the farthest string problem (FSP) as an Integer Linear Programming Problem (ILPP). In the present talk we present a general novel approach to reduce the size of the ILPP by grouping isomorphous columns of the string matrix together. The approach is of practical use, since the solution of sequence problems is very time consuming, in particular when the sequences are long.

About the invited speaker:

Peter Zörnig studied mathematics and physics at the University of Dortmund and received his PhD at Fernuniversität Hagen/Germany in 1990. He is currently associate professor at the Statistical Department of the University of Brasília/BRAZIL. He has published several books and research papers in the areas of linear programming, mathematics for economists, graph theory, probability and quantitative linguistics. More information is available on the home page of his department: www.est.unb.br

Integer Linear Programming for Sequence Problems:

A general approach to reduce the problem size

Peter Zörnig

University of Brasília, Department of Statistics

1 Introduction:

Examples for (information) sequences:

DNA: ACAGTCAGT.... (alphabet with 4 nucletides)
Codifies the entire information of an organism

Proteins: 2, 1, 15, 2, 20, 14, 12... (alphabet with 20 aminoacids)

Formalized text: 5, 4, 3, 4, 6, 2...
The elements represent e.g. the word type (adjective, verb...)

Computer program: ...

Areas of application:

Molecular Biology
(e.g. development of generic drugs and diagnostic methods)

Coding Theory
Data compression
Quantitative and Computational Linguistcs (e.g. automatic translation)

Given the alphabet $\Omega = \{1,\ldots\omega\}$ and a set of sequences (strings) $\Sigma = \{s^1,\ldots,s^n\} \subset \Omega^m$ with

$$\begin{array}{c} s^1 = (s_1^1,\ldots,s_m^1) \\ \vdots \quad\quad \vdots \\ s^n = (s_1^n,\ldots,s_m^n) \end{array}$$

One seeks for a sequence $t = (t_1,\ldots,t_m)$ that is as similar or as different as possible to the given sequences
(for example, an insecticide must be as harmful as possible to insects and as harmless as possible for persons and pets) .

2 Example of a String Problem

Definition: Let $\Omega = \{1,...\omega\}$ be an alphabet. For s, $t \in \Omega^m$ the *Hamming distance* $d(s, t)$ is defined as the number of positions at which s and t differ.

Example: $\Omega = \{1,...4\}$, m=8

s =(A, C, G, A, T, A, T ,G)
t = (A, T, G, A, G, C, T ,A) $d(s, t) = 4$

$d(s, s) = 0$
$d(s, t) = d(t, s)$
$d(s, t) + d(u, t) \geq d(s, u)$

Closest String Problem (CSP):

Given the string set $\Sigma = \{s^1,...,s^n\}$with $s^i = (s_1^i,..., s_m^i) \in \Omega^m$.

Determine a $t \in \Omega^m$ that minimizes $D(t) := \max_{i=1,...,n} d(s^{\mathrm{i}}, t)$.
The problem is NP-hard.

3 The conventional model

Example: Given the following CSP with n=3, $\omega = 4$ and m=10:

$$S = \begin{pmatrix} s^1 \\ s^2 \\ s^3 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 & 3 & 4 & 2 & 4 & 2 & 1 & 1 \\ 4 & 4 & 2 & 4 & 3 & 4 & 3 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 & 2 & 1 & 2 & 1 & 1 & 4 \end{pmatrix}$$

LetV_j be the set of characters appearing at position j (j=1,...,10).
For example, V_1 = {2, 3, 4}, V_2 = {3, 4}.

A sequence $t = (t_1,\ldots,t_{10})$ can only be a candidate for a solution of the CSP, if $t_j \in V_j$ for j = 1,…,10.

Such a candidate (feasible solution) is codified by means of the variables $x_{i,j}$.

For j = 1,…,10 and $i \in V_\mathrm{j}$ we define:

$x_{i,j}$ = 1 for $t_j = i$,
$x_{i,j}$ = 0 otherwise

For the above example we obtain the variables

$x_{3,1}$	$x_{4,2}$	$x_{4,3}$	$x_{3,4}$	$x_{4,5}$	$x_{2,6}$	$x_{4,7}$	$x_{2,8}$	$x_{1,9}$	$x_{1,10}$
$x_{4,1}$	$x_{3,2}$	$x_{2,3}$	$x_{4,4}$	$x_{3,5}$	$x_{4,6}$	$x_{3,7}$	$x_{1,8}$	$x_{3,9}$	$x_{4,10}$
$x_{2,1}$		$x_{1,3}$		$x_{2,5}$	$x_{1,6}$	$x_{2,7}$			

and e. g. the feasible solution t = (3,4,1,4,3,2,2,2,3,4) is codified by

1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0	1	1
0		1		0	0	1			

Integer Linear Programming Model

$$\min\ d$$
$$\text{s.t.}$$

$$d(t, s^1) = 10 - x_{3,1} - x_{4,2} - x_{4,3} - x_{3,4} - x_{4,5} - x_{2,6} - x_{4,7} - x_{2,8} - x_{1,9} - x_{1,10} \leq d$$
$$d(t, s^2) = 10 - x_{4,1} - x_{3,2} - x_{2,3} - x_{4,4} - x_{3,5} - x_{4,6} - x_{3,7} - x_{1,8} - x_{3,9} - x_{3,10} \leq d$$
$$d(t, s^3) = 10 - x_{2,1} - x_{3,2} - x_{1,3} - x_{4,4} - x_{2,5} - x_{1,6} - x_{2,7} - x_{1,8} - x_{1,9} - x_{4,10} \leq d$$

$$\sum_{i \in V_j} x_{i,j} = 1 \quad \text{for } j = 1,\ldots,10$$

$$x_{i,j} \in \{0; 1\}$$

Number of variables: $1 + \sum_{j=1}^{m} |V_j| = 26$

Number of constraints: $n + m = 13$

4 The improved model

Basic idea: “normalize” the problem and treat isomorphic columns simultaneously by assigning them to the same group

Definition: Two vectors $v = (v_1,\ldots,v_n)^T$ and $w = (w_1,\ldots,w_n)^T$ over Ω are called *isomorphic*, if and only if

$$v_i = v_j \quad \Leftrightarrow \quad w_i = w_j \quad \text{for all } i, j \in (1,\ldots,n\}.$$

Example:

$$\begin{pmatrix} 1 \\ 4 \\ 4 \\ 1 \\ 2 \end{pmatrix} \text{ isomorphic } \begin{pmatrix} 2 \\ 3 \\ 3 \\ 2 \\ 1 \end{pmatrix} \text{ not isomorphic } \begin{pmatrix} 2 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

An isomorphism class corresponds to a partition of the set {1,...,n} of lines.

For n=3 exist at most 4 non-isomorphic columns, represented by:

$$\begin{pmatrix}1\\1\\2\end{pmatrix}, \begin{pmatrix}1\\2\\1\end{pmatrix}, \begin{pmatrix}1\\2\\2\end{pmatrix} \text{ and } \begin{pmatrix}1\\2\\3\end{pmatrix}$$

By substituting the columns of a CSP by its representation vectors, we can normalize a CSP:

Matrix of a CSP and normalized problem

Position		1	2	3	4	5	6	7	8	9	10
S	s^1	**3**	**4**	4	3	4	**2**	4	**2**	1	1
	s^2	4	4	2	**4**	**3**	4	3	2	**3**	**4**
	s^3	2	3	**1**	4	2	1	**2**	1	1	4

Position		1	2	3	4	5	6	7	8	9	10
T	t^1	**1**	**1**	1	1	1	**1**	1	**1**	1	1
	t^2	2	1	2	**2**	**2**	2	2	1	**2**	**2**
	t^3	3	2	**3**	2	3	3	**3**	2	1	2

To a sequence s *de* S corresponds biuniquely a sequence t of T such that $d(s, s^i) = d(t, t^i)$ for all i $\Rightarrow$
A solution of T provides a solution of S.

Solution of a normalized CSP normalized (example):

By ordering the columns of *T* we obtain

Normalized problem with reordered columns

Position		1	2	3	4	5	6	7	8	9	10
T	t^1	1	1	1	1	1	1	1	1	1	1
	t^2	1	1	2	2	2	2	2	2	2	2
	t^3	2	2	1	2	2	3	3	3	3	3
isomorphism class		1		2	3		4				

A feasible solution *t* can be represented by means of the frequencies $y_{i,j}$ of the character *i* in group *j*.

For example, the feasible solution $t = (1,1\,|\,1\,|\,2,2\,|\,1,1,2,3,3)$ is codified by

$$
\begin{array}{llll}
y_{1,1} = 2, & y_{1,2} = 1, & y_{1,3} = 0, & y_{1,4} = 2, \\
y_{2,1} = 0, & y_{2,2} = 0, & y_{2,3} = 2, & y_{2,4} = 1, \\
 & & & y_{3,4} = 2.
\end{array}
$$

We obtain the **integer linear programming problem**

$$
\begin{array}{l}
\min d \\
\text{s.t.} \\
10 - y_{1,1} - y_{1,2} - y_{1,3} - y_{1,4} \le d \\
10 - y_{1,1} - y_{2,2} - y_{2,3} - y_{2,4} \le d \\
10 - y_{2,1} - y_{1,2} - y_{2,3} - y_{3,4} \le d \\
y_{1,1} + y_{2,1} = 2, \\
y_{1,2} + y_{2,2} = 1, \\
y_{1,3} + y_{2,3} = 2, \\
y_{1,4} + y_{2,4} + y_{3,4} = 5
\end{array}
$$

$y_{i,j}$ non-negative integers.

Obs.: For three given strings this model has only **10 variables and 7 constraints, independently of the sequence length!**

In the conventional model the size increases linearly with the sequence length.

Suppose e.g. that any column of the above table appears with frequency 300 (i.e. m=1200), the improved model solves the CSP easily, while the conventional has 2701 variables and 1203 constraints.

General case:

$$\min d$$

$$m- \sum_{j=1}^{k} y(t_j^i, j) \le d \quad \text{for i=1,…,}n$$

$$\sum_{i=1}^{v_i} y_{i,j} = m_{\text{j}} \quad \text{for j=1,…,}k$$

$y_{\text{i,j}}$ non-negative integer

Solution Process in two phases

1) Solve the Linear Relaxation
2) Determine an (approximate) integer solution via rounding

Obs.: Only a **small part** of the solutions of the relaxation is **non-integer.**

Test problems for n=4

No.	*m*	*N*	d_{relax}	d_{approx}	d_{opt}	It.LP	It. B&B
1	1554	6	683,25	684	684	16	50
2	1511	4	676	**677**	**676**	15	34
3	1584	4	701	**702**	**701**	15	28
4	1441	2	639,5	640	640	12	66
5	1336	6	596,25	597	597	15	42
6	1563	6	708,25	709	709	16	74
7	1751	6	800,75	801	801	17	31
8	1882	0	860	860	860	18	19
9	1973	2	890,5	891	891	16	24
10	1782	6	804,75	805	805	13	25
11	1616	4	728	**729**	**728**	12	91
12	1542	6	695,75	696	696	12	27
13	1425	4	641	**642**	**641**	12	31

14	1316	6	591,25	592	592	12	25
15	1231	2	553,5	554	554	12	25
16	1144	6	513,25	514	514	12	24
17	1028	4	461	**462**	**461**	13	18
18	1228	6	551,25	552	552	14	25
19	1356	4	608	**609**	**608**	14	34
20	1438	2	644,5	645	645	12	26

N: number of non-integer variables
d_{relax}: objective value of the “relaxed solution”
d_{approx}: objective value of the approximate solution obtained by rounding
d_{opt}: optimal value of the CSP

n=5 (In all cases it holds $d_{\text{approx}} = d_{\text{opt}}$!):

No	*m*	*N*	d_{relax}	d_{approx}	It.LP
1	2974	8	1494,6	1495	61
2	3058	8	1543,8	1544	63
3	3005	8	1515,6	1516	68
4	3085	8	1556,8	1557	66
5	3148	8	1589,8	1590	71
6	3235	8	1634,8	1635	63
7	3328	8	1681,2	1682	59
8	3278	6	1655,8	1656	63
9	3034	8	1540,6	1541	62
10	2723	6	1384,6	1385	69

5 Farthest String Problem (FSP):

Given $\Sigma = \{ s^1, \ldots, s^n \}$ with $s^i = (s^i_1, \ldots, s^i_m) \in \Omega^m$.

Determine a string t that maximizes $D(t) := \min_{i=1,\ldots,n} d(s^i, t)$

Case a) $t \in V_1 \times \ldots \times V_m$

Case b) $t \in V^m$ with $V = V_1 \cup \ldots \cup V_m$

Example:

$$S = \begin{pmatrix} 3 & 1 & 3 & 1 & 1 & 2 & 3 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 & 3 & 3 & 3 & 2 & 3 & 2 \\ 1 & 3 & 1 & 2 & 2 & 1 & 2 & 1 & 2 & 2 \end{pmatrix}$$

Case a) $t \in \{1,2,3\} \times \{1,3\} \times \{1,2,3\} \times \{1,2\} \times \{1,2,3\} \times \{1,2,3\} \times \{2,3\} \times \{1,2\} \times \{1,2,3\} \times \{1,2\}$

Case b) $t \in \{1,2,3\}^m$ V={1,2,3}

Def.: A column j is called *incomplete*, if $V_j \not\subset V$

Incomplete columns can be easily handled, so we can restrict ourselves to case a)

Model:

$$\max d$$

$$m - \sum_{j=1}^{k} y(t^i_j, j) \geq d \quad \text{for } i=1,\ldots,n$$

$$\sum_{i=1}^{v_i} y_{i,j} = m_j \quad \text{for } j=1,\ldots,k$$

$y_{i,j}$ non-negative integers

Test Problems

	no.	*m*	*N*	d_{relax}	d_{approx}	max. error
n=6, ω=5 *k*=15	21	2794	8	2328.333	2328	0
	22	3305	10	2754.167	2754	0
	23	4394	9	3515.5	3514	1
	24	5487	12	4398.333	4398	0
	25	6034	11	5068.833	5067	1
	26	7856	10	6284.667	6282	2
	27	8698	6	7393.333	7393	0
	28	9482	8	7775.167	7773	2
	29	10670	10	8962	8961	1
	30	11355	8	9084.833	9084	0
n=8, ω=7 *k*=28	31	5530	13	4838.75	4838	0
	32	10420	14	8336.625	8335	1
	33	19673	10	17705.875	17705	0
	34	38912	13	33075.375	33074	1
	35	78452	9	67468.75	67468	0
	36	150230	8	130700.13	130698	2
	37	230722	10	189192.25	189189	3
	38	295015	12	236012.38	236012	0
	39	450723	8	396636.88	396633	3
	40	950643	13	751007.75	751006	1

Conclusões

The reduction principle applies also to other tyypes of sequence problems:
---closest substring problem
---farthest substring problem
---far from most strings problem
---problems with multiple criteria

Open questions:

---test the models for large numbers of strings
---what is the distribution of the number of incomplete columns and
of the number *k* of isomorphism classes
---distributions of the parameter m_j (size of isomorphism class *j*)?

```
/* ZORN(2011), n=3<=w, Example 1 */

/* Objective function */
min: +d;

/* Constraints */
D1: 10-y11-y12-y13-y14 <= d;
D2: 10-y11-y22-y23-y24 <= d;
D3: 10-y21-y12-y23-y34 <= d;
E1: y11+y21=2;
E2: y12+y22=1;
E3: y13+y23=2;
E4: y14+y24+y34=5;

/* Integer definitions */
int d,y11,y12,y13,y14,y21,y22,y23,y24,y34;
```

```
/* ZORN(2011), n=3<=w, Example 2 */

/* Objective function */
min: +d;

/* Constraints */
D1: 1200-y11-y12-y13-y14 <= d;
D2: 1200-y11-y22-y23-y24 <= d;
D3: 1200-y21-y12-y23-y34 <= d;
E1: y11+y21=300;
E2: y12+y22=300;
E3: y13+y23=300;
E4: y14+y24+y34=300;

/* Integer definitions */
int d,y11,y12,y13,y14,y21,y22,y23,y24,y34;
```

```
/* ZORN(2011), n=4<=w, Example 3 */

/* Objective function */
min: +d;

/* Constraints */
D1: 1495-y11-y12-y13-y14-y15-y16-y17-y18-y19-y110-y111-y112-y113-y114
<= d;
D2: 1495-y11-y22-y23-y24-y25-y16-y17-y18-y29-y210-y211-y212-y213-y214
<= d;
D3: 1495-y21-y12-y23-y24-y15-y26-y17-y28-y19-y310-y211-y312-y313-y314
<= d;
D4: 1495-y21-y22-y13-y24-y15-y16-y27-y38-y39-y110-y311-y212-y313-y414
<= d;
E01: y11+y21=72;
E02: y12+y22=75;
E03: y13+y23=75;
E04: y14+y24=68;
E05: y15+y25=73;
E06: y16+y26=75;
E07: y17+y27=68;
E08: y18+y28+y38=141;
E09: y19+y29+y39=134;
E10: y110+y210+y310=138;
E11: y111+y211+y311=145;
E12: y112+y212+y312=142;
E13: y113+y213+y313=148;
E14: y114+y214+y314+y414=141;

/* Integer definitions */
int d,y11,y12,y13,y14,y15,y16,y17,y18,y19,y110,y111,y112,y113,y114,
      y21,y22,y23,y24,y25,y26,y27,y28,y29,y210,y211,y212,y213,y214,
      y38,y39,y310,y311,y312,y313,y314,y414;
```

Applied Mathematical Modelling 35 (2011) 5609–5617

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Improved optimization modelling for the closest string and related problems

Peter Zörnig *

Department of Statistics, Institute of Exact Sciences, University of Brasília, 70910-900 Brasília-DF, Brazil

ARTICLE INFO

Article history:
Received 3 February 2011
Received in revised form 6 May 2011
Accepted 8 May 2011
Available online 1 June 2011

Keywords:
Closest string problem
Motif finding
Computational biology
Mathematical programming
Modelling

ABSTRACT

We present a new integer linear programming model for the closest string problem. This model requires considerably less variables and constraints than the popular binary linear programming model used for this purpose. Due to the reduced size it is easier to handle rounding errors when an LP relaxation technique is used to solve the problem.

The proposed model is particularly useful for closest string problems where a small set of long sequences is given. In this case, the optimal string or a good approximate solution can be usually obtained by rounding the optimal solution of the LP relaxation to the nearest integers.

1. Introduction

The closest string problem (CSP), also known as the motif finding problem, represents an active research topic of combinatorial optimization with many practical applications, especially in computational biology and coding theory, see e.g. [1–9].

We start with an alphabet Ω, i.e. a finite set of elements, called characters. Without loss of generality we assume to be $\Omega = \{1,\ldots,\omega\}$ with $\omega \in \mathrm{IN}$. Let Ω^m denote the set of all sequences of length m with elements chosen from Ω. For any two sequences $s,t \in \Omega^m$ the *Hamming distance* $d(s,t)$ between s and t is defined as the number of positions in which s and t differ. It can be easily shown that $d(s,t)$ satisfies the properties of a distance, in particular the triangle inequality. The CSP is defined as follows:

Given a set $\Sigma = \{s^1,\ldots,s^n\}$ of sequences with $s^i = (s^i_1,\ldots,s^i_m) \in \Omega^m$ for $i = 1,\ldots,n$, find a sequence $t \in \Omega^m$ such that $D(t) = \max_{i=1,\ldots,n} d(s^i,t)$ is minimal.

From the triangle inequality it follows immediately that $D(t)$ can not be less than $\lceil HD(\Sigma)/2 \rceil$, where $HD(\Sigma) = \max_{s,t\in\Sigma} d(s,t)$ denotes the *Hamming diameter* of Σ [10, p. 2054], [11, Lemma 1]. Also, by $\lceil x \rceil$ and $\lfloor x \rfloor$ we will denote the smallest integer greater than or equal to x and the largest integer less than or equal to x, respectively.

2. The conventional model

The most widely used approach to solve the CSP which is known to be NP-hard consists in modelling it as an integer linear programming problem as follows, see e.g. [1,2,7,8]. Consider the CSP matrix

* Fax: +55 61 31076768.
E-mail address: peter@unb.br

0307-904X/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2011.05.015

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 22, Number 8, 2015
© Mary Ann Liebert, Inc.
Pp. 729–742
DOI: 10.1089/cmb.2014.0265

Reduced-Size Integer Linear Programming Models for String Selection Problems: Application to the Farthest String Problem

PETER ZÖRNIG

ABSTRACT

We present integer programming models for some variants of the farthest string problem. The number of variables and constraints is substantially less than that of the integer linear programming models known in the literature. Moreover, the solution of the linear programming-relaxation contains only a small proportion of noninteger values, which considerably simplifies the rounding process. Numerical tests have shown excellent results, especially when a small set of long sequences is given.

Key words: computational biology, far from most strings problem, farthest string problem, mathematical programming, modeling.

1. INTRODUCTION

String selection and comparison problems have numerous applications, principally in computational biology, but also in coding theory, data compression, and quantitative linguistics. For instance, genomic and proteomic data can be modeled as sequences (strings) over the alphabets of nucleotides or amino acids; see, for example, Blazewicz et al. (2005), Boucher (2010), and Pappalardo et al. (2013). The formalization of problems like motif recognition and similar tasks leads to diverse combinatorial optimization problems like the closest (sub) string problem, the farthest (sub) string problem, the close to most strings problem, the far from most strings problem (FFMSP), and the distinguishing string selection problem; see, for example, Soleimani-damaneh (2001), Lanctot et al. (2003), Meneses et al. (2005), Festa (2007), Zörnig (2011), and Ferone et al. (2013).

In the present article we study some variants of the farthest string problem (FSP) that are generally NP-hard. The solution of these problems is very difficult, in particular in most biological applications where the sequences are very long; see Blazewicz et al. (2005) and Zörnig (2011, p. 3). FSPs are frequently modeled as (zero–one) integer linear programming (ILP) problems; see, for example, Lanctot et al. (2003), Meneses et al. (2005), and Festa and Pardalos (2012). Our main objective is to generalize the size reduction approach of Zörnig (2011), which has so far never been addressed in the literature, and apply it to the FSP. We show that at least for a small set of sequences of arbitrary length, several variants of the FSP can be solved (exactly or with a very small error in the optimal value), by merely solving the linear programming (LP) relaxation of the ILP problem and subsequent rounding of noninteger solution values. After introducing the necessary concepts, we model the FSP as an integer linear programming problem, considering two cases of

Department of Statistics, Institute of Exact Sciences, University of Brasília, Brasília, Brazil.