Modelling Optimal Location of Traffic Counting Points at Urban Intersections in CLP(FD)

Ana Paula Tomás

Technical Report Series: DCC-2001-4

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6078830 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

Modelling Optimal Location of Traffic Counting Points at Urban Intersections in CLP(FD)

Ana Paula Tomás*
DCC & LIACC, Univ. do Porto
R. do Campo Alegre, 823
4150-180 Porto, Portugal
apt@ncc.up.pt

May 2001

Abstract

An application of Constraint Programming for finding the minimum number and location of count-posts at urban roundabouts so as to obtain origin-destination data at minimum cost is given. By finding interesting mathematical properties, we were able to model this problem as a constraint satisfaction problem in finite domains.

1 Introduction

Traffic studies for network planning, design, management and control, require origin-destination (OD) demand data whose acquisition is difficult and expensive. Important, but often insufficient, data is collected by counting vehicles passing at specific points of the traffic network, either manually or automatically. In the past two decades, there has been a considerable amount of research on the difficult problem of OD matrix estimation from (link) traffic counts, with economic issues as a major motivation. Either based on traffic modelling concepts or statistical inference approaches, typically all models for OD matrix estimation use some prior information on the OD matrix, which may be an old, probably outdated, matrix [1]. Some authors have examined various errors that may arise in the estimation, but with very limited discussion about how to identify the subset of network links where flow information should be collected and used [12]. The optimisation of the number and location of traffic counting points for traffic networks is the subject of two recent publications [4, 12]. While in [12], the emphasis is put on the quality of the estimation, where some heuristics for locating count-posts are examined, in [4], a model and heuristic based algorithms are given for solving the minimum-cost sensor location problem. This problem is that of finding the minimum number and location of count-posts in order to obtain the complete set of traffic flows on a transport network at minimum cost. In these works, it is assumed either some knowledge about the turning probabilities at junctions or path flow information.

Far less complex is the case of urban intersections, and in particular that of roundabouts, for which vehicles counts at exits, entries and cross-sections of the circulatory roadway may determine, in some situations, the complete set of OD flows. Traffic studies at intersections are not only quite common, but of prime importance to aid traffic operation for ascertaining the need for traffic control devices, geometric changes, tuning signal phasing, and so on. In order to obtain OD data, it is typically necessary to carry out OD surveys, such as manually recording of registration numbers or roadside video surveys, in combination with some number plate tagging scheme.

*The work presented here has been partially supported by funds granted to LIACC through the *Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia* and *Programa POSI*.

In this work, we present some of the results from our investigation on the problem of finding the minimal number of exits and entries where OD surveys must be made, when some particular subset of traffic counts can be obtained at a relatively negligible cost. This can be viewed as a variant of the above mentioned minimum-cost sensor location problem but for the case of urban intersections. As in the work by Confessore et al. [4], the analysis of specific mathematical properties of the problem also plays a crucial role in our model and solving method. The model we propose is that of a constraint satisfaction problem in finite domains. Clearly, no prior knowledge of turning movement coefficients is assumed or used in our work, which makes the problem distinct from that in [4].

This paper appears as a complement to a previous work [13], for which programs were implemented (in C) to perform an exhaustive case analysis of hypothetical roundabouts, seeking for possible patterns for the optimal solutions for some given cost functions. It is motivated by the observation that the cost criteria should be more flexible to encompass specific features of the particular roundabout in study [2]. In our previous work, we have briefly considered the use of a Constraint Programming system aiming at reducing the implementation effort. However, difficulties in modelling the problem in a suitable way, led us to write the programs in C and to implement search strategies based on usual techniques in Combinatorial Optimization and Constraint Programming. Relevant mathematical properties of the problem were put in evidence by an extensive analysis of the programs output. In particular, we were able to prove the existence of a quite well-defined pattern for the unique optimal solution when the estimated cost for measuring some directional flow is given by the number of roads between the origin and destination roads [2, 13].

Undoubtedly more important than that characterization in itself, was the fact that the solution had a greedy character. This led us to identify that problem as an instance of that of computing a maximum-weight independent subset in a linear matroid, which may always be solved by a Greedy Algorithm (see e.g. [5]). In brief, if the cost of measuring each OD flow is known, overall cost is minimized if the OD flows that should not be measured are selected in non-decreasing order of cost to form an independent set.

To check this independence — in fact, the linear independence of the columns of the system matrix relating to the selected OD flows — Gaussian elimination was implemented. Due to floating-point errors, an output solution was found with a wrong pattern, which finally led us to develop a rather simple method to test for independence. This method is a crucial tool in the work we now describe.

The paper is organized as follows. In Section 2, we give a formal description of the problem and some of its mathematical properties. Further results relating to our test for independence are stated in Section 3. To provide some insight on the problem, a particular situation is studied in Section 4. Finally, we present our model and discuss aspects of its implementation in CLP(FD) systems.

2 The Problem

Without loss of generality, we identify each hypothetical roundabout where n roads (streets) intersect by a string $R_1 R_2 \ldots R_n \in \{E, S, D\}^\star$, with $R_1 \neq E$ and $R_n \neq S$. This latter convention shall be of use in Sections 4 and 5.1. Each $R_i \in \{\mathrm{E}, \mathrm{D}, \mathrm{S}\}$ indicates whether road i is just an entry (E), just an exit (S) or both an entry and exit (D) road. Thus, we assume a numeration of the roads in the way traffic circulates such that the ordered sets $\mathcal{O}$ and $\mathcal{D}$, of origins and destinations, are defined as $\mathcal{O} = \{\imath_1, \ldots, \imath_e\}$ and $\mathcal{D} = \{\jmath_1, \ldots, \jmath_s\}$, where $\jmath_1 = 1$, $\imath_e = n$, $e = |\mathcal{O}|$ and $s = |\mathcal{D}|$. It is worth observing that all the strings obtained from $R_1 R_2 \ldots R_n$ by rotation denote exactly the same roundabout, but for different numerations of the roads.

The traffic flow from the entry i to the exit j is denoted by q_{ij}, for $i \in \mathcal{O}$ and $j \in \mathcal{D}$. These directional traffic volumes are related to the total volumes at entries, exits and passing through the cross-sections of the circulatory roadway in frontal alignment with the intersecting roads (respectively, O_i, D_j and F_k) by (1)–(3).

$$\sum_{j \in \mathcal{D}} q_{ij} = O_i, \text{ for } i \in \mathcal{O} \tag{1}$$

$$\sum_{i \in \mathcal{O}} q_{ij} = D_j, \text{ for } j \in \mathcal{D} \tag{2}$$

$$\sum_{i \in \mathcal{O} \setminus \{k\}} \; \sum_{j \in \mathcal{D},\, k \prec j \preceq i} q_{ij} \;=\; F_k, \quad \text{for } 1 \leq k \leq n \tag{3}$$

In (3), $j \in \mathcal{D},\ k \prec j \preceq i$ stands for the exits between road k and road i, being k excluded. Vehicles are assumed to exit the roundabout and, in addition, loops are disallowed (i.e., vehicles do not overpass their destination). As schematically illustrated in Fig. 1, other cross-sections may be considered, as

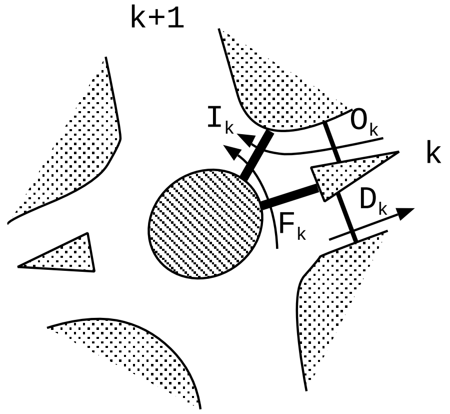


Figure 1: Cross-sections of the circulatory roadway

those standing strictly between two consecutive roads. But, clearly, the traffic volume I_k, that passes between roads k and $k+1$, is related to F_k and O_k by $I_k = F_k + O_k$, which means that counting I_k may be redundant. Determining which total counts to collect may also be of some practical interest, but for the purpose of this work we make the following assumption.

> **Assumption.** Counting the volumes O_i's, D_j's, F_k's and I_k's is of negligible cost when compared to other measuring tasks.

This means that to obtain the desired OD data at optimal cost, some maximal independent set of such counts is used, but it does not matter which one. Quite unfortunately, (4) always holds, and so at most $|\mathcal{O}| + |\mathcal{S}|$ (that is, $e + s$) total counts are globally non-redundant.

$$F_k \;=\; F_1 + \sum_{1 \leq i < k,\ i \in \mathcal{O}} O_i - \sum_{1 < i \leq k,\ i \in \mathcal{D}} D_i, \quad \text{for all } k \tag{4}$$

Based on the well-known theoretical result for Transportation Problems in Linear Programming (our textbook was [8]), that the matrix of the subsystem (1)–(2) has rank $e + s - 1$, we have shown the following property [13].

Property 1 *The matrix of the system defined by (1)–(3), in the variables q_{ij}, has rank $e+s$ if and only if none of equations in (3) is of the form $F_k = 0$, being $e+s-1$ otherwise. Furthermore, the rank is $e+s-1$ if and only if the string $R_1R_2\ldots R_n$ that identifies the roundabout, as defined above, is not in the language described by the regular expression $S^\star(D+SE)E^\star$.*

From Property 1, it is trivial to obtain the rank of (1)–(3). Even if the following states a quite strong condition in real practice, we consider that the traffic counts are error-free, an usually accepted condition [1, 4, 12].

> **Assumption.** Each of the required traffic counts can be obtained with accuracy during a certain period of time, meaning that the system that results from replacing such counts in (1)–(3) is consistent.

Under this assumption, and by a classical property of systems of linear equations, we conclude that (1)–(3) is equivalent to each of its subsystems, that consist of equations (1)–(2) and any of the equations in (3), say that of F_k.

Let $\mathbf{P}'$ be the matrix of such subsystem, being $\mathbf{p}'_{ij}$ the column of the variable q_{ij}. Let $\mathbf{P}$ be the matrix of the subsystem consisting of equations (1)–(2) and $\mathbf{p}_{ij}$ be the column of q_{ij}. Clearly, $\mathbf{p}'_{ij}$ has just one more element than $\mathbf{p}_{ij}$, namely the coefficient of q_{ij} in the equation defining F_k. We denote such element just by σ_{ij}, for almost always we refer to the case $k = 1$ (where we take F_1).

Now, it is known that any of the first $e + s$ equations is redundant, being the last equation redundant if and only if the system has rank $e + s - 1$. Provided $es - \text{rank}(\mathbf{P}')$ of the q_{ij}'s are known (that means, counted), the subsystem admits a unique solution for the remaining $\text{rank}(\mathbf{P}')$ variables if and only if the columns of $\mathbf{P}'$ relating to them are linearly independent.

Our problem. To find a basis $\mathcal{B}'$ consisting of $\text{rank}(\mathbf{P}')$ independent columns of $\mathbf{P}'$, such that to measure the OD flows q_{ij} with $\mathbf{p}'_{ij} \notin \mathcal{B}'$, surveys are undertaken at a minimum number of exits and entries. Without further assumptions, it is not difficult to show that the optimal cost would be, in general, $(e-1) + (s-1)$, meaning that the registration should take place at all the entries but one and at all the exits but one. What we want to study is whether this cost may be reduced by having some of the q_{ij} counted by other means, which are supposed to be less expensive, whatever that may mean (e.g., in some situations, some q_{ij}'s may be fully obtained by direct observation).

3 Checking Linear Independence by Hand

Lemma 1 below, which gives a simple graphical method to check the linear independence of a subset of columns of $\mathbf{P}$, refers to the following representation of the matrix $\mathbf{P}$ in tableau form.

$$\begin{array}{cccc} \mathbf{p}_{\imath_1\jmath_1} & \mathbf{p}_{\imath_1\jmath_2} & \cdots & \mathbf{p}_{\imath_1\jmath_s} \\ \mathbf{p}_{\imath_2\jmath_1} & \mathbf{p}_{\imath_2\jmath_2} & \cdots & \mathbf{p}_{\imath_2\jmath_s} \\ \vdots & & & \vdots \\ \mathbf{p}_{\imath_e\jmath_1} & \mathbf{p}_{\imath_e\jmath_2} & \cdots & \mathbf{p}_{\imath_e\jmath_s} \end{array}$$

Together with Lemma 2, it states well known results of Transportation type problems in LP, which may be found in [8].

Lemma 1 *A subset $\mathcal{B}$ of the columns of $\mathbf{P}$ is free if and only if the graph G built in the following way is acyclic. Its vertices correspond to the elements of $\mathcal{B}$, and the edges are obtained by linking each vertex in a given row (respectively, column) to the vertex in the same row (respectively, column) that is closer to that one in the tableau, if any exists. Moreover, $\mathcal{B}$ is a basis of the subspace spanned by the columns of $\mathbf{P}$ if and only if G is acyclic, connected, and contains at least a vertex in each column and in each row of the tableau.*

Lemma 2 *If $\mathcal{B}$ is a basis of the columns of $\mathbf{P}$, the linear combination of the columns in $\mathcal{B}$ that gives $\mathbf{p}_{ij} \notin \mathcal{B}$ is of the form*

$$\mathbf{p}_{ij} = \mathbf{p}_{ij_1} - \mathbf{p}_{i_1j_1} + \mathbf{p}_{i_1j_2} - \cdots - \mathbf{p}_{i_kj_k} + \mathbf{p}_{i_kj}$$

In terms of the tableau, $\mathbf{p}_{ij}$ and the columns in $\mathcal{B}$ that occur on the right-hand side form a simple cycle, consecutive edges being orthogonal.

To illustrate these properties, we analyse the examples in Fig. 2, where the columns in $\mathcal{B}$ are those in framed boxes. The conclusion is that the subset shown on the left is a basis of the subspace spanned by the columns of $\mathbf{P}$, whereas the one on the right is not even a free subset. For the example on the left, $\mathbf{p}_{12} = \mathbf{p}_{14} - \mathbf{p}_{24} + \mathbf{p}_{22}$, whereas on the right, we have $\mathbf{p}_{11} = \mathbf{p}_{12} - \mathbf{p}_{22} + \mathbf{p}_{23} - \mathbf{p}_{33} + \mathbf{p}_{31}$. If we remove $\mathbf{p}_{11}$ from these selected columns, we also obtain a basis.

Now, Corollary 1 states our extension of the above results to $\mathbf{P}'$, providing a powerful tool for practical use as it leads to a means of almost solving some problems by hand. Its proof [13], follows

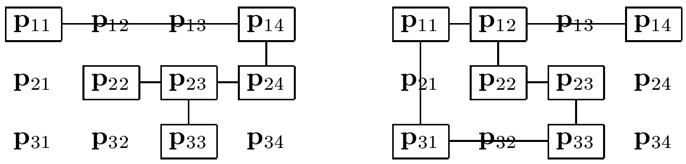


Figure 2: Quickly testing some $\mathbf{p}_{ij}$'s for independence

almost immediately from Lemmas 1 and 2, using the fact that $rank(\mathbf{B}) \leq rank(\mathbf{B}') \leq rank(\mathbf{B}) + 1$, where $\mathbf{B}'$ and $\mathbf{B}$ are the submatrices consisting of the columns in $\mathcal{B}'$ and $\mathcal{B}$, respectively.

Corollary 1 *Let $\mathcal{B}'$ be a subset of the columns of $\mathbf{P}'$ and $\mathcal{B}$ the set of the corresponding columns in $\mathbf{P}$. If some column $\mathbf{p}_{ij} \notin \mathcal{B}$ is written in an unique way as a combination of the columns in $\mathcal{B}$ by $\mathbf{p}_{ij} = \mathbf{p}_{ij_1} - \mathbf{p}_{i_1j_1} + \mathbf{p}_{i_1j_2} - \cdots - \mathbf{p}_{i_kj_k} + \mathbf{p}_{i_kj}$ then $\mathbf{p}'_{ij}$ is free relatively to $\mathcal{B}'$ if and only if*

$$\sigma_{ij} \neq \sigma_{ij_1} - \sigma_{i_1j_1} + \sigma_{i_1j_2} - \cdots - \sigma_{i_kj_k} + \sigma_{i_kj}$$

Thus, if $\mathcal{B}$ is not a free set, then $\mathcal{B}'$ is free if and only there exists some $\mathbf{p}_{ij}$ such that $\mathcal{B} \setminus \{\mathbf{p}_{ij}\}$ is free and the equality that expresses $\mathbf{p}_{ij}$ as a combination of the columns in $\mathcal{B} \setminus \{\mathbf{p}_{ij}\}$ does not hold for the corresponding columns in $\mathcal{B}'$. So, the graph G associated to $\mathcal{B}$ may have at most a simple cycle when $\mathcal{B}'$ is free.

As a consequence, when $\mathcal{B}'$ is free, the graph G associated to $\mathcal{B}$, cannot have cycles as those illutrated in Fig. 3, where each symbol $\circ$ denotes a $\mathbf{p}'_{ij} \in \mathcal{B}'$. Hence, a necessary condition for $\mathcal{B}'$ to

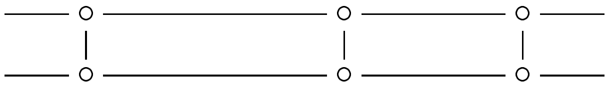


Figure 3: Example of linear dependence.

be a basis is that, when $e \geq 3$, the $\mathbf{p}'_{ij} \in \mathcal{B}'$ do not fill two or more columns of the tableau, neither do fill two or more rows when $s \geq 3$, whichever the basis $\mathcal{B}'$ is. In other words, when there are at least three entries and three exits, the optimal cost is given by $(e-1)+(s-1)$, if no q_{ij} is supposed to be fully obtained by direct observation.

We shall now go through an instance of our problem that has some theoretical interest. The fact that we had spent some time showing that it may be completely solved by hand, was of some help to design the general model.

4 Studying a Particular Cost Function

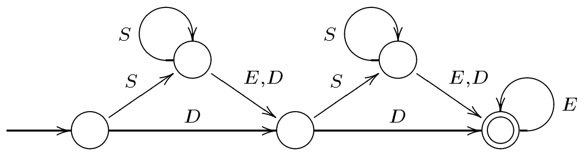
While studying hypothetical roundabouts, it makes some sense to consider the directional flows $q_{i\,i+1}$ as the ones that may be counted using some less expensive means. Here, $i+1$ refers to the road that immediately follows road i, in the ordering naturally induced by the way traffic circulates in the roundabout, which is the road 1 if $i = n$. By our hypothesis that $R_1 \neq E$ and $R_n \neq S$, the OD flow q_{n1} always appears. Notice that $q_{i\,i+1}$ occurs iff $R_iR_{i+1} \in \{ED, ES, DD, DS\}$.

The output from our programs, in this case, showed no regular patterns, so that it is almost of no use trying to abstract a form for the optimal solutions from the output data. Nevertheless, by following a case-based approach, using Corollary 1 as a guide to identify the relevant cases, we were able to exactly characterize the optimal cost for each type of hypothetical roundabout $R_1R_2 \ldots R_n$. There are relatively few cases to analyse, not only due to the same roundabout being represented by all the strings in a subset closed under string rotation, but also for a kind of duality we have found between certain patterns.

Another crucial factor is that when both $e \geq 5$ and $s \geq 5$, the optimal cost is necessarily $(e-1)+(s-1)$. Only in that way cycles as those illustrated in Fig. 3 are avoided, since there is at most one $q_{i\,i+1}$ in each line of the tableau.

Each case is nothing but a class of roundabouts with exactly the same optimal solutions, being possible to describe their patterns by some *regular expression* (RE). Naturally, the closure properties of string rotation extend to the possible identifying expressions. In broad terms, classes are distinguished by four major features, namely the numbers of entries and of exits (respectively, e and s), the number of directional flows $q_{i\,i+1}$ and the location of the $\mathbf{p}_{i\,i+1}$'s in the tableau.

Since, only the cases when either $e < 5$ or $s < 5$ remained to study, by designing some *deterministic finite state automata* (DFA) [9] we found a first coarse description of the classes. An example is given in Fig. 4. Notice that $\imath_1$ refers to the first entry road. Thus, $\imath_1 + 1 \in \mathcal{D}$ means that the first row of the tableau contains one of the $\mathbf{p}_{i\,i+1}$'s.



$$S^\star(SE+D)S^\star(SE+D)E^\star$$

Figure 4: DFA and RE for roundabouts with two $q_{i\,i+1}$'s and $\imath_1 + 1 \in \mathcal{D}$.

To illustrate some of the ideas and results, we go through a small example.

4.1 An interesting example

For the roundabout SSEDE, we have $\mathcal{O} = \{3,4,5\}$, $\mathcal{D} = \{1,2,4\}$, being q_{34} and q_{51} the directional flows of the form $q_{i\,i+1}$. The rightmost table in Fig. 5 contains the coefficients σ_{ij} when we take the equation defining F_1 as the last one in the subsystem. The positions of the variables $q_{i\,i+1}$ are marked with $\star$. The rank of the system matrix is $e+s=6$, implying that we can avoid counting six of the q_{ij}'s. To discard registration at entry i (respectively, exit j) we must select all the flows entering from road i (respectively, exiting at road j), with the possible exception of those marked with $\star$, whose observation is supposed of small cost. Thus, the problem may be seen as that of placing six $\circ$'s in the tableau, so as to cover, with $\circ$'s and $\star$'s, the largest number of columns and rows, while ensuring the linear independence of the $\mathbf{p}'_{ij}$ selected to $\mathcal{B}'$. These elements are those covered by $\circ$'s.

To obtain a basis, we cannot cover exit 1 and exit 2 at the same time, neither both entries 3 and 4, because of the cycle (*square*) shown below on the left.

$$\begin{array}{ccc} 0 & 1 & \star \\ 0 & 1 & 1 \\ \star & 1 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & \star \\ 0 & 1 & 1 \\ \star & 1 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & \star \\ 0 & & 1 \\ \star & 1 & 1 \end{array}$$

	1	2	4
3	$\mathbf{p}_{31}$	$\mathbf{p}_{32}$	$\mathbf{p}_{34}$
4	$\mathbf{p}_{41}$	$\mathbf{p}_{42}$	$\mathbf{p}_{44}$
5	$\mathbf{p}_{51}$	$\mathbf{p}_{52}$	$\mathbf{p}_{54}$

	1	2	4
3	0	1	0
4	0	1	1
5	0	1	1

	1	2	4
3	0	1	$\star$
4	0	1	1
5	$\star$	1	1

Figure 5: An example — SSEDE

The same can be said about exits 2 and 4, and entries 4 and 5. Moreover, the cycle shown on the right makes it impossible to cover two exits and two entries at the same time. This problem has the following six optimal solutions

$\circ$	$\circ$	$\circ$
$\circ$		$\circ$
$\star$		$\circ$

$\circ$		$\circ$
$\circ$	$\circ$	$\circ$
$\star$		$\circ$

$\circ$		$\circ$
$\circ$		$\circ$
$\star$	$\circ$	$\circ$

$\circ$	$\circ$	$\circ$
$\circ$		
$\star$	$\circ$	$\circ$

$\circ$	$\circ$	$\circ$
	$\circ$	
$\star$	$\circ$	$\circ$

$\circ$	$\circ$	$\circ$
		$\circ$
$\star$	$\circ$	$\circ$

The optimal cost is thus given as $(e-1)+(s-2)+1^\star$ and $(e-2)+(s-1)+1^\star$, where $1^\star$ means that one $q_{i\,i+1}$ must be obtained by direct observation. The kind of symmetry between entries and exits that such cost expressions suggest, is due to the duality between certain patterns that we mentioned above. This is made clear in Proposition 1 and by Fig. 6.

Proposition 1 *For any given roundabout $r = R_1R_2\ldots R_n$, let $\Psi(r)$ denote the string that we obtain if we exchange E with S in r and read the resulting expression from right to left. The roundabouts r and $\Psi(r)$ yield exactly the same system of equations, if q^r_{ij} is replaced by $q^{\Psi(r)}_{n-j+1\,n-i+1}$, and the constraints are re-ordered. The optimal cost wrt r is $(e_r - n_1) + (s_r - n_2) + n_3^\star$ iff the optimal cost for $\Psi(r)$ is given by $(e_{\Psi(r)} - n_2) + (s_{\Psi(r)} - n_1) + n_3^\star$, where $e_r = s_{\Psi(r)}$ and $s_r = e_{\Psi(r)}$ stand for the numbers of entries and exits.*

$SDESEE$

	1	2	4
2	0	1	0
3	0	1	$\star$
5	0	1	1
6	$\star$	1	1

wrt F_1^r

$SSESDE = \Psi(SDESEE)$

	1	2	4	5
3	1	1	$\star$	0
5	1	1	1	1
6	$\star$	0	0	0

wrt $F_n^{\Psi(r)}$

$\xrightarrow{\text{that is}}$

5	0	1	0
4	0	1	$\star$
2	0	1	1
1	$\star$	1	1
	6	5	3

Figure 6: Mapping tableau representations for r and $\Psi(r)$.

By Proposition 1, it is sufficient to analyse the cases of roundabouts with $e \leq s \leq 5$. As regards the roundabout SSEDE, it holds $\Psi(\text{SSEDE}) = \text{SSEDE}$, and thus there are two alternative expressions for the optimal cost.

Notice also that for the roundabouts that define the same tableau (wrt F_1) as SSEDE, we have $e = s = 3$, $\mathcal{O} = \{\imath_1, \imath_2, \imath_3\}$, $\mathcal{D} = \{1, 2, \imath_1 + 1\}$ and $\imath_1 \geq 2$. Since the tableau has two $\star$'s, one appearing in its first row, this class is part of the set described by $S^\star(SE+D)S^\star(SE+D)E^\star$, as seen in Fig. 4. The condition $e = s = 3$ holds for two instances of this expression, namely $S(SE+D)(SE+D)E$ and $(SE+D)S(SE+D)E$, which give different classes. The optimal cost for the latter is $(e-2)+(s-2)+2^\star$.

5 Modelling the General Problem as a CSP

Praça da República at Porto is a roundabout of type SEESDSE, for which a more realistic scenario is as shown in Fig. 7 on the left, as claimed in [2]. The intended meaning is that the OD flows q_{24} and q_{25} can be fully obtained by direct observation in site, by some person located at entry 2. The same applies to the pair q_{34} and q_{35}, and to q_{71}. By contrast, the geometry of the roundabout makes it too difficult to obtain q_{56} by direct observation. The mathematical model we are going to formulate applies to optimization problems as this one. The solution given in Fig. 7 is one of the eight optimal solutions found by our prototype implementations of the following model in SICStus

	1	4	5	6
2		$\star$	$\star$	
3		$\star$	$\star$	
5				
7	$\star$			

	1	4	5	6
2	$\circ$	$\star$	$\star$	$\circ$
3	$\circ$	$\star$	?	
5	$\circ$	$\circ$		
7	$\star$	$\circ$	$\circ$	$\circ$

Figure 7: A real world example — Praça da República

Prolog [11, 3] and ECLiPSe [6]. The symbol `?` indicates that there are two alternative solutions to obtain q_{35} (since we do not perform `labeling` on X, the domain variable X_{23} is not instantiated, but found in $\{-1,1\}$). The sets $\mathcal{O}$, $\mathcal{D}$, $\Gamma = \{(i,j) \mid q_{\imath_i \jmath_j}$ may be collected by direct observation$\}$ and $\Theta = \{i \mid (i,j) \in \Gamma$ for some $j\}$ are given as input.

The decision variables.

$$\begin{array}{rcl}
E_i & \in & \{0,1\}, \text{ is 1 iff a sensor is located at entry } \imath_i, \text{ with } 1 \leq i \leq e \\
S_j & \in & \{0,1\}, \text{ is 1 iff a sensor is located at exit } \jmath_j, \text{ with } 1 \leq j \leq s \\
V_i & \in & \{0,1\}, \text{ is 1 iff an observer stands at entry } \imath_i, \text{ with } i \in \Theta \\
P_{ij} & \in & \{0,1\}, \text{ is 1 iff } \mathbf{p}'_{\imath_i \imath_j} \text{ would be in } \mathcal{B}', \text{ with } 1 \leq i \leq e,\ 1 \leq j \leq s. \\
X_{ij} & \in & \{0,1,-1\} \text{ if } (i,j) \in \Gamma. \text{ Otherwise, } X_{ij} \in \{0,1\}, \text{ for all } i,\ j.
\end{array}$$

$$X_{ij} = \begin{cases} 1 & \text{if} \quad q_{\imath_i \jmath_j} \text{ is collected by sensor} \\ -1 & \text{if} \quad (i,j) \in \Gamma \text{ and } q_{\imath_i \jmath_j} \text{ is directly collected at entry } \imath_i \\ 0 & \text{if} \quad q_{\imath_i \jmath_j} \text{ is not collected} \end{cases}$$

The objective function. It is defined by (5)

$$\text{Minimize} \qquad Cost = \sum_{i=1}^{e} e_i E_i + \sum_{j=1}^{s} s_j S_j + \sum_{i \in \Theta} o_i V_i \tag{5}$$

where e_i, s_j, and o_i denote the defined costs for sensors and observers, the latter being assumed relatively insignificant. Given these costs, by putting a suitable upper bound on $Cost$, we may declare that the trivial solution — to locate sensors at all the entries but one and all the exits but one – as nogood.

The constraints. For all $1 \leq i \leq e$ and $1 \leq j \leq s$,

1. To record some flow $q_{\imath_i \jmath_j}$, sensors must be located at entry $\imath_i$ and exit $\jmath_j$

$$E_i + S_j \quad \geq \quad 2X_{ij} \tag{6}$$

   Notice that, if $E_i + S_j \in \{0,1\}$ then $X_{ij} \in \{0,-1\}$.

2. To obtain $q_{\imath_i \jmath_j}$ by direct observation, there must be an observer at entry $\imath_i$.

$$V_i \quad \geq \quad -X_{ij} \tag{7}$$

3. If $q_{\imath_i \jmath_j}$ is not explicitly collected then $\mathbf{p}'_{\imath_i \jmath_j} \in \mathcal{B}'$.

$$X_{ij} = 0 \quad \Leftrightarrow \quad P_{ij} = 1 \tag{8}$$

   Equivalently, we may write (9) and (10).

$$X_{ij} + P_{ij} = 1, \qquad \text{for all } (i,j) \notin \Gamma \tag{9}$$

$$X_{ij} P_{ij} = 0 \ \wedge \ X_{ij} + P_{ij} \neq 0, \qquad \text{for all } (i,j) \in \Gamma \tag{10}$$

4. We adopt the following conventions.

   (a) An observer is placed at entry $\imath_i$ only if it has to count some OD flow $q_{\imath_i \jmath_j}$, with $(i,j) \in \Gamma$.

$$V_i = 1 \quad \Rightarrow \quad \exists_{(i,j)\in\Gamma}\ X_{ij} = -1 \tag{11}$$

   (b) A sensor is located at entry $\imath_i$ only if some OD flow $q_{\imath_i \jmath_j}$, with $(i,j) \notin \Gamma$, has to be counted. The same holds for sensors at exits.

$$\sum_{j=1,(i,j)\notin\Gamma}^{s} X_{ij} \geq E_i \qquad \text{and} \qquad \sum_{i=1,(i,j)\notin\Gamma}^{e} X_{ij} \geq S_j \tag{12}$$

5. The set $\mathcal{B}' = \{\mathbf{p}'_{\imath_i \jmath_j} \mid P_{ij} = 1\}$ must be a basis to $\mathbf{P}'$. Equivalently, $\mathcal{B}'$ has cardinality rank$(\mathbf{P}')$,

$$\sum_{i=1}^{e}\sum_{j=1}^{n} P_{ij} \quad = \quad \mathrm{rank}(\mathbf{P}') \tag{13}$$

   and $\mathcal{B}'$ is a free set (i.e. its elements are linearly independent), which corresponds to (14), where the variables $\beta_{ij} \in \mathbb{R}$.

$$\sum_{\mathbf{p}'_{\imath_i \jmath_j}\in\mathcal{B}'} \beta_{ij}\mathbf{p}'_{ij} = \mathbf{0} \quad \Rightarrow \quad \beta_{ij} = 0, \text{ for all such } (i,j) \tag{14}$$

With the exception of (11) and (14), these constraints can be straightforwardly encoded in CLP(FD). But, (11) may be implemented by *cardinality* operators — in SICStus Prolog, as

```
count(-1,XiStr,#>=,Vi)
```

where `XiStr` is the list of X_{ij} with $(i,j) \in \Gamma$ — and, in ECLiPSe, as `#(Vi,CtrXi,NSti)`, where `CtrXi` is a list of constraints $X_{ij} = -1$ and `NSti` its length, for $(i,j) \in \Gamma$.

To take advantage of consistency-based approaches to solving CSPs [10], we would like to replace (14) by other constraints that may be used to prune the search space in a more effective way. Two necessary conditions on $\mathcal{B}'$ are deduced from the results in Section 3. The first one is translated by (15) and says that each row and column of the tableau must contain at least one element of $\mathcal{B}'$.

$$\sum_{j=1}^{s} P_{ij} \geq 1 \qquad \text{and} \qquad \sum_{i=1}^{e} P_{ij} \geq 1 \tag{15}$$

The other one is what we call *the Square-Free Property* (SQF-Property, for short), which asserts that, for all $1 \leq i < i' \leq e$ and $1 \leq j < j' \leq s$, if

$$\sigma_{\imath_i \jmath_j} - \sigma_{\imath_i \jmath_{j'}} + \sigma_{\imath_{i'} \jmath_{j'}} - \sigma_{\imath_{i'} \jmath_j} = 0$$

then (16) must be satisfied,

$$P_{ij} + P_{ij'} + P_{i'j'} + P_{i'j} \quad < \quad 4 \tag{16}$$

and, if in addition, $\{(i,j),(i,j'),(i',j'),(i',j)\} \cap \Gamma = \emptyset$, meaning that none of these positions contains a $\star$, then (17) has to hold.

$$E_i + E_{i'} \geq 1 \qquad \text{and} \qquad S_j + S_{j'} \geq 1 \tag{17}$$

Its terms of the tableau, (17) says that the pair of entries $\imath_i$ and $\imath'_i$ cannot be discarded at the same time, neither can the exits $\jmath_j$ and $\jmath'_j$. Being (17) binary constraints, arc-consistency techniques may be employed. Nevertheless, as we shall see in Section 5.1, we can achieve at least the same pruning while possibly defining fewer and more global constraints.

5.1 Revisiting the SQF-Property

Given two rows i and i' of the tableau, we say that i *and* i' *are incompatible* if and only if $E_i + E_{i'} \geq 1$ is entailed by the SQF-Property. In a similar way, we define j and j' as incompatible columns if $S_j + S_{j'} \geq 1$ must hold.

Of great importance is the observation that, when we write the tableau wrt to F_1, the first column contains only 0's. Furthermore, for each row that contains two or more 0's, only 0's occur to the right of the second 0. As regards columns, there are only 1's below each 1. In fact, in view of the intuitive meaning of the total traffic volume F_1, it is not difficult to see that $\sigma_{\alpha\beta} = 0$ iff $\beta = 1$ or $\beta > \alpha$.

So, let $b_i = \min(s+1, \{j \mid \jmath_j \in \mathcal{D} \text{ and } \jmath_j > \imath_i\})$ and $\Gamma_i = \{j \mid (i,j) \in \Gamma\}$. Informally, b_i identifies the column of the second 0 in row i, if any exists, whereas Γ_i contains the columns of the $\star$'s in row i. The following lemmas are of help to implement the SQF-Property. Their proofs follow quite immediately from our previous observation.

Lemma 3 *For all $(i,i'), (j,j')$ such that $1 \leq i < i' \leq e$ and $1 \leq j < j' \leq s$, it holds $\sigma_{\imath_i \jmath_j} - \sigma_{\imath_{i'} \jmath_{j'}} + \sigma_{\imath_{i'} \jmath_{j'}} - \sigma_{\imath_{i'} \jmath_j} = 0$ if and only if either $j, j' \in [1, b_i[\cup [b_{i'}, s]$ or $j, j' \in [b_i, b_{i'}[$.*

Lemma 4 *For all (j,j') with $1 \leq j < j' \leq s$, the columns j and j' are incompatible if and only if $j, j' \in ([1, b_i[\cup [b_{i'}, s]) \setminus (\Gamma_i \cup \Gamma'_i)$ or $j, j' \in [b_i, b_{i'}[\setminus (\Gamma_i \cup \Gamma'_i)$, for some (i,i') such that $1 \leq i < i' \leq e$.*

Now, let $\mathcal{I}_1^{(i,i')} = ([1, b_i[\cup [b_{i'}, s]) \setminus (\Gamma_i \cup \Gamma'_i)$ and $\mathcal{I}_{b_i}^{(i,i')} = [b_i, b_{i'}[\setminus (\Gamma_i \cup \Gamma'_i)$ be the two sets of incompatible columns we obtain by enforcing the SQF-Property wrt rows i and i'.

Lemma 5 *The constraints $S_j + S_{j'} \geq 1$ that SQF-Property may impose are satisfied if instead we have (18) for all (i,i') such that $1 \leq i < i' \leq e$.*

$$\sum_{j \in \mathcal{I}_1^{(i,i')}} S_j \geq |\mathcal{I}_1^{(i,i')}| - 1 \qquad \textit{and} \qquad \sum_{j \in \mathcal{I}_{b_i}^{(i,i')}} S_j \geq |\mathcal{I}_{b_i}^{(i,i')}| - 1 \tag{18}$$

Lemma 6 *The constraints on P that the SQF-Property entails are equivalent to $P_{ij} + P_{ij'} + P_{i'j'} + P_{i'j} < 4$, for all $1 \leq i < i' \leq e$ and $1 \leq j < j' \leq s$ such that $j, j' \in [1, b_i[\cup [b_{i'}, s]$ or $j, j' \notin [1, b_i[\cup [b_{i'}, s]$.*

Lemma 7 *Two rows i and i', such that $1 \leq i < i' \leq e$, are incompatible if $|\mathcal{I}_1^{(i,i')}| \geq 2$ or $|\mathcal{I}_{b_i}^{(i,i')}| \geq 2$.*

An algorithm for establishing the SQF-Property can be deduced from the previous Lemmas, in which the main task is to compute the intervals $[1, b_i[\cup [b_{i'}, s]$, $\mathcal{I}_1^{(i,i')}$ and $\mathcal{I}_{b_i}^{(i,i')}$, which is not difficult at all. By doing that, we get enough information about sets of incompatible columns and pairs of incompatible rows.

To obtain more global constraints on the E_i's than those in (17), we may consider the complementary relation — defined by the pairs (i,i') of compatible rows — and find all *maximal independent sets with at least two elements* in the undirected graph that represents it. Although this is not a great idea in general, since finding *all* those sets is known to be an NP-complete problem (e.g. [7]), it works quite well in our specific case, as the graph is either small or sparse.

In SICStus, `findall(Set,independent_set(UGraph,2,Set),LSets)`, does that job for us using `library(ugraphs)`, for any given undirected graph `UGraph`. In our implementation, we start from a complete graph `CGraph` with vertices `1..s`, from which we subsequently remove edges that link incompatible columns, while constructing the graph `RGraph` of the compatibility relation for the rows and imposing (16).

As seen in Section 4.1 for the roundabout SSEDE, the SQF-Property may be insufficient to ensure that the computed solution is a basis. Nevertheless, it often allows relevant propagation for this specific problem, and thus, we decided to first label P and then to check that it defines a basis. So, in SICStus, we had a call to `minimize(solution(Vars,P,Rank,Ne,Ns,Sigmas),Cost)` with `solution/6` defined as follows, `Vars` consisting of all the decision variables but the X_{ij}'s, with the variables P_{ij} in its tail.

```
solution(Vars,P,Rank,Ne,Ns,Sigmas) :-
   labeling([],Vars),   % with no preferential strategy
   basis(P,Rank,Ne,Ns,Sigmas).
```

Further details on `basis/5` are given in following section.

5.2 Detecting troublesome cycles

Let $\mathcal{B}' = \{\mathbf{p}'_{\imath_i\imath_j} \mid P_{ij} \text{ is labelled to } 1\}$ and let $\mathcal{N}_{\mathcal{B}'} = \{(i,j) \mid \mathbf{p}_{\imath_i\imath_j} \in \mathcal{B}'\}$. By using the results in Section 3, we can define `basis/5` with little effort. The main idea is to obtain the largest subset $\mathcal{S} \subseteq \mathcal{B}'$ such that the graph $G_\mathcal{S}$ (as defined in Lemma 1) is a tree. Notice that if $\mathcal{B}'$ is a basis then either $\mathcal{S} = \mathcal{B}' \setminus \{\mathbf{p}'_{\imath_o\imath_d}\}$ or `Rank` is $e+s-1$ and $\mathcal{S} = \mathcal{B}'$. Moreover, $\mathcal{S}$ can be found by a deterministic completion procedure, in which the possible resulting $\mathbf{p}'_{\imath_o\imath_d}$ is the first node found to violate the tree-shape. It can be done iteratively — at iteration i, we try to link the chain consisting of the nodes $(i,j) \in \mathcal{N}_{\mathcal{B}'}$ selected in row i. To do that we adopt the idea of the Kruskal's Algorithm for minimum spanning trees [7].

```
forest([],Trees,Trees,_,[]).
forest([I-NodesI|Nodes],Trees,Treesf,LPod,[I-BasisI|Basis]) :-
   deleting(NodesI,Trees,RTrees,LTrees,I,LPod,BasisI),
   flatten(LTrees,NewTree),
   forest(Nodes,[NewTree|RTrees],Treesf,LPod,Basis).
```

In calls to `forest/5`, the first argument is a list of terms `I-NodesI` that represent the selected nodes in row `I`, being `NodesI` a list of terms `n(J,SigmaIJ)`. Initially, the argument `LPod` is bounded to `[]` or `[Pod]`, depending on `Rank`, and `Trees` is `[[1],[2],..., [Ns]]`, where `Ns` equals s, the number of exits. As suggested by its name, `deleting/7` implements deletion to pick up the tree where each node `n(J,SigmaIJ)` is imagined to be linked (i.e., the one that contains `J`), gathering them in `LTrees`. It naturally fails when two nodes in the same row are to be attached to the same tree, because that tree is removed when the first node is found. This predicate is deterministic, leaving no choice points.

If the program call to `forest/5` succeeds then, when `Rank` is $e+s$, we still have to check whether $\mathbf{p}'_{\imath_o\jmath_d}$ (which `Pod` identifies) can be written as a linear combination of the elements in $\mathcal{S}$ (which is represented by the last argument of `forest/6`). That can be done by first solving a CSP in finite domains

$$\sum_{\mathbf{p}'_{\imath_i\imath_j} \in \mathcal{S}} \beta_{ij}\mathbf{p}_{\imath_i\imath_j} = \mathbf{p}_{\imath_o\jmath_d}$$

in $e+s$ variables β_{ij} such that $\beta_{ij} \in \{0,1,-1\}$, which has a *unique* solution. Notice that $\mathbf{p}_{\imath_i\jmath_j} = \mathbf{e}_i + \mathbf{e}_{e+j}$, where $\mathbf{e}_i$ and $\mathbf{e}_{e+j}$ are the unit vectors of the real space of dimension $e+s$. Finally, we conclude that $\mathcal{B}'$ is a basis iff the following condition holds for the computed β_{ij}.

$$\sum_{\mathbf{p}'_{\imath_i\jmath_j} \in \mathcal{S}} \beta_{ij}\sigma_{\imath_i\jmath_j} \neq \sigma_{\imath_o\jmath_d}$$

This was the strategy we adopted to quickly define `basis/5`. Possibly, our prototype implementations can be improved.

5.2.1 Remembering nogoods

In our previous work [13] (to minimize $\sum_{ij} c_{ij}X_{ij}$) a depth-first search strategy with chronological backtracking (combined with branch-and-bound) was implemented to find bases. To check whether some $\mathbf{p}'_{ij}$ was free from those already in a set, we applied Gaussian elimination. This method is fairly adequate to handle the incremental change of the matrix formed by such columns in an efficient way. When a variable q_{ij} is selected, the program simply applies to $\mathbf{p}'_{ij}$ the transformations done in the previous steps in precisely the same order. This allows us to detect conflicts that may arise near the root of the search tree. By using a marking scheme, we prevented q_{ij} from being selected until the conflicting variable is removed through backtracking. For the current work, we could have adapted `forest/6` to get some new information from which other constraints on the P_{ij} could be deduced. One possibility in ECLiPSe, is to use a non-logical variable `cycles` to collect lists of `(I,J)`'s resulting in troublesome cycles, which may be used to add new constraints. But it does not seem worthwhile the implementation effort for our specific problem. In real-life, we cannot usually expect an observer to be able to count more than three OD flows even if their destinations are in sight. An extensive simulation for randomly generated roundabouts, with $4 \leq n \leq 10$ and less than four $\star$'s per row, shows that nogood solutions were found just in less than 5% of the cases.

Further Research Ongoing work aims at studying whether Constraint Programming may be successfully applied to sensor location problems for Transportation Networks.

References

[1] Abrahamsson T.: Estimation of Origin-Destination Matrices using Traffic Counts – A Literature Survey. IR-98-021/May, IIASA, Austria, 1998.

[2] Andrade M.: *Métodos e Técnicas de Recolha de Dados de Tráfego – Algoritmo para a Definição da Matriz Origem-Destino.* Msc. Thesis, Faculdade de Engenharia da Universidade do Porto, 2000.

[3] Carlsson M., Ottosson G., Carlson B.: An Open-Ended Finite Domain Constraint Solver. In *Proceedings of PLILP'97*, LNCS 1292, 191-206, Springer-Verlag, 1997.

[4] Bianco L., Confessore G., and Reverberi P.: Optimal Location of Traffic Counting Points for Transport Network Control. In *Proceedings of IFAC'97.* Extended version to appear in *J. Transportation Science.*

[5] Cook W., Cunningham W., Pulleyblank W., and Schrijver A.: *Combinatorial Optimization.* John Wiley & Sons, 1998.

[6] ECLiPSe User Manual Version 5.1.2, IC-PARC, Inperial College, London, 2001.

[7] Cormen H., Leiserson C., Rivest R.: *Introduction to Algorithms*, MIT Press, 1990.

[8] Hadley G.: *Linear Programming.* Addison-Wesley, 1969.

[9] Hopcroft J. E., Ullman J. D.: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[10] Marriott K., and Stuckey P.: *Programming with Constraints – An Introduction*, The MIT Press, 1998.

[11] SICStus Prolog User Manual Release 3.8.6, SICS, Sweden, 2001.

[12] Yang H., and Zhou J.: Optimal Traffic Counting Locations for Origin-Destination Matrix Estimation. *J. Transportation Research*, 32B(2), 109–126, 1998.

[13] Tomás A. P., Andrade M., and Pires da Costa A.: Obtaining Origin-Destination Data at Optimal Cost at Urban Roundabouts, Internal Report, DCC-FC & LIACC, Universidade do Porto, 2001 (in preparation).