Instituto de Engenharia de Sistemas e Computadores de Coimbra
Institute of Systems Engineering and Computers
INESC – Coimbra

Gilbert Laporte and Marta Pascoal

The pipeline and valve location problem

No. 2 2010

ISSN: 1645-2631

Instituto de Engenharia de Sistemas e Computadores de Coimbra
INESC – Coimbra
Rua Antero de Quental, 199; 3000 - 033 Coimbra; Portugal
www.inescc.pt

The pipeline and valve location problem

GILBERT LAPORTE[1,2] and MARTA M. B. PASCOAL[3,4]

[1]Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
E-mail: `Gilbert.Laporte@cirrelt.ca`

[2]Canada Research Chair in Distribution Management, HEC, Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[3]Departamento de Matemática da Universidade de Coimbra, Apartado 3008, EC Universidade, 3001-454 Coimbra, Portugal
E-mail: `marta@mat.uc.pt`

[4]Institute for Systems and Computers Engineering – Coimbra (INESCC)

May 2010

Abstract

This paper proposes an exact algorithm for the problem of locating a pipeline between two points of a network, as well as a set of safety valves which help control the damage caused by possible spills along the pipeline. A labelling approach is developed to determine simultaneously the optimal pipeline and valve locations, with the objective of optimizing an impact measure that depends on the average number of accidents and their cost. An upper limit is imposed on the maximum number of valves. The algorithm includes a dominance test as well as the definition of bounds which allow useless labels to be discarded at an early phase. Computational experiments on grid and random instances are presented in order to evaluate the algorithm's performance and to compare its results to the solutions provided by sequential approaches.

Keywords: Multicriteria path problems, labelling, environmental studies.

1 Introduction

Pipelines are used to transport different gaseous or liquid products, one of the most common being crude oil. This equipment is robust, but spills and other accidents are possible and can have a major environmental impact, as well as expensive implications. To minimize the impact of accidents, one can determine the best route for the pipeline in terms of risk. For example, the route of the Eastern Siberia-Pacific Ocean (ESPO) oil pipeline between Angarsk, Russia, and Daqing, China, had to be moved 40 km to the north of Lake Baikal in response to the protests of environmental organizations [10]. A standard protection mechanism for pipelines is the installation of shutoff or safety valves at appropriate locations. These valves are activated when a leak occurs in order to isolate the section of the pipeline affected by the leak between two consecutive valves, thus limiting the impact of accidents.

The literature on this problem is rather limited. Grigoriev and Grigorieva [7] have developed a dynamic, as well as a binary search algorithm to locate safety valves along a given route in order to minimize a bottleneck objective function measuring the environmental impact of accidents. More recently Bodlaender, Hendriks, Grigoriev and Grigorieva [1] have considered the problem of placing a given number of valves in a network in such a way that the maximum possible spill is minimized. They have shown that the problem is NP-hard, even for particular graph topologies like series-parallel graphs, and graphs in which all edge costs are equal to one. They have also shown that the problem can be solved in polynomial time when the network is a simple path, a cycle or a tree. In addition, Laporte and Pascoal [8] have dealt with the simultaneous location of a path and relays under a cost minimization objective. They have used a labelling approach involving the minimization of an additive function subject to a resource constraint. Some research has also been carried out on the determination of a route for the transport of hazardous materials, under the minimization of different risk functions between two given points of a network, like in Erkut and Verter [6, 9]. Díaz-Báñez, Gómez and Toussaint [4] have studied the point-to-point shortest path problem avoiding a set of obstacles in a continuous space, and Cáceres, Mesa and Ortega [2] have used Voronoi diagrams for the location of undersea waste pipelines in order to minimize the environmental impact.

Our purpose is to simultaneously determine the pipeline route and the valve locations under a risk minimization objective. The problem is modelled as a multicriteria path problem, and solved by means of a labelling procedure. The algorithm is empirically tested and compared to the sequential approach of finding a route first and choosing the valve locations afterwards.

The remainder of the paper is divided into four sections. Section 2 introduces the *Pipeline and Valve Location Problem* (PVLP) in more detail and presents a mathematical model. Section 3 proposes a labelling algorithm for the PVLP. It describes a dominance test and pruning techniques that can be used to reduce the determination of dominated paths. Section 4 evaluates the behaviour of the algorithm based on a set of empirical experiments, and compares the results with those obtained by a sequential approach. Finally, the last section draws some conclusions.

2 Mathematical model

We represent the space where the pipeline is to be located by a directed network $(\mathcal{N}, \mathcal{A})$ with a set $\mathcal{N} = \{1, \ldots, n\}$ of nodes and a set $\mathcal{A} = \{1, \ldots, m\}$ of arcs. Two nodes are distinguished in $\mathcal{N}$: an origin s and a destination t. A pipeline between these two nodes is represented by a path $p = \langle v_1, v_2, \ldots, v_\ell \rangle$, where $v_1 = s$, $v_\ell = t$, $v_i \in \mathcal{N}$, $i = 1, \ldots, \ell$, and $(v_i, v_{i+1}) \in \mathcal{A}$, $i = 1, \ldots, \ell - 1$, along with a subset of nodes $V(p) = \{\widehat{s}_1, \ldots, \widehat{s}_k\}$ at which valves are located. For simplicity we write $i \in p$ if i is a node in the sequence p, $(i, j) \in p$ if i immediately precedes j in p, and $q \subset p$ if the sequence of nodes in a path q is a subsequence of the nodes of p. We denote by $\mathcal{P}$ the set of paths from s to t in $(\mathcal{N}, \mathcal{A})$. Given $i, j \in p$, the subpath of p between nodes i and j is denoted by $\sigma_p(i, j)$. If a path p_1 ends at a node where path p_2 starts, then their concatenation is denoted by $p_1 \diamond p_2$.

With each arc $(i, j) \in \mathcal{A}$ are associated three values:

- the average number $\lambda_{ij} \geq 0$ of accidents (spills) on (i, j) per time period,
- the amount $b_{ij} \geq 0$ (in barrels) of oil transported along a pipe linking node i to node j, and

- the environmental damage $d_{ij} \geq 0$ produced by a leak along the pipe between i and j measured in monetary units per barrel of spilled oil.

The total impact of choosing a certain location for the pipeline and a sequence of safety valves, in monetary units and within a time period, is represented by a function r. To simplify the presentation let $c_{ij} = b_{ij}d_{ij}$ denote the cost of damage on arc (i, j) in case of an accident on this arc, and for an arbitrary path q

$$\lambda(q) = \sum_{(i,j)\in q} \lambda_{ij} \text{ and } c(q) = \sum_{(i,j)\in q} c_{ij}.$$

On a pipeline, or path p, containing no shutoff valve the total impact along a time period is then

$$r(p) = \lambda(p)c(p). \tag{1}$$

More generally, given p a path from x to y with $\widehat{s}_1, \ldots, \widehat{s}_k$ as the sequence of nodes with a valve, this impact is expressed as

$$r(p) = r(\sigma_p(s, \widehat{s}_1)) + \sum_{i=2}^{k} r(\sigma_p(\widehat{s}_{i-1}, \widehat{s}_i)) + r(\sigma_p(\widehat{s}_k, t)).$$

To illustrate the impact of an accident, in Figure 1, if there is a valve at node 2, then a leak occurring on one arc affects that area only, so the total impact is

$$r(\langle 1, \widehat{2}, 3\rangle) = (\lambda_{12} \times b_{12} \times d_{12}) + (\lambda_{23} \times b_{23} \times d_{23}) = 6 + 15 = 21.$$

If there is no valve between 1 and 3, then there are on average $\lambda_{12} + \lambda_{23} = 8$ accidents per period. A leak along that pipe leads to a damage of $b_{12} \times d_{12} = 2$ in the area surrounding arc $(1, 2)$, whereas in the area surrounding arc $(2, 3)$ the damage is $b_{23} \times d_{23} = 3$. Therefore, the impact of this path is

$$r(\langle 1, 2, 3\rangle) = (\lambda_{12} + \lambda_{23})(b_{12} \times d_{12} + b_{23} \times d_{23}) = 8 \times (2 + 3) = 40,$$

as expected greater than when no safety valve is used.

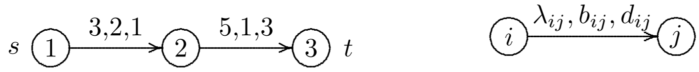

Figure 1: A pipeline from s to t

Because shutoff valves are very expensive devices, only a limited number W can be installed on the pipeline, in addition to the two valves always installed at s and t. Let $v(p) = |V(p)| \leq W$ represent the number of safety valves along a pipeline associated with path p.

The purpose of the PVLP is to determine feasible paths between s and t, i.e., paths containing at most W valves, with minimum global impact. In other words, denoting the set of feasible paths in $\mathcal{P}$ by

$$\overline{\mathcal{P}} = \{p \in \mathcal{P} : \ v(p) \leq W\},$$

the PVLP aims to find a path from s to t in $(\mathcal{N}, \mathcal{A})$ satisfying

$$\min_{p \in \overline{\mathcal{P}}} \{r(p)\}, \tag{2}$$

that is, to find a path $p \in \overline{\mathcal{P}}$ such that $r(q) \leq r(p)$ for any other feasible path $q \in \overline{\mathcal{P}}$. Unless otherwise stated, in the sequel we will assume the optimal pipeline and the valve locations are determined simultaneously. As already mentioned, an alternative related problem can be defined following a sequential strategy consisting of two distinct phases:

$$\min_{p \in \mathcal{P}} \{R(p)\} \tag{3}$$

and

$$\min_{p \in p^*} \{r(p)\}, \tag{4}$$

where R represents an objective function that measures the accidents impact along a certain path without valves, and path p^* is an optimal solution of problem (3). Note that different R functions may be used. For example, let γ be a function that assigns $\gamma(p) = \sum_{(i,j) \in p} \lambda_{ij} c_{ij}$ to any path p in $(\mathcal{N}, \mathcal{A})$. Taking $R = \gamma$ corresponds to the *optimistic* assumption that a valve can be installed at every node of the chosen path, while taking $R = \lambda c$ corresponds to the *pessimistic* assumption that no valves are installed.

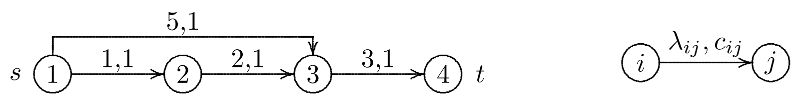

Figure 2: Finding the best pipeline and valve location

To illustrate the difference between simultaneous and sequential pipeline and valve location, consider the network depicted in Figure 2. For $W = 1$ the path $p = \langle 1, \widehat{3}, 4 \rangle$ is the optimal solution of (2). If $R = \gamma$, then the path chosen for pipeline location is $p^* = \langle 1, 2, 3, 4 \rangle$ $(R(p^*) = 6)$, and the single valve that can be used is located at node 3, thus $q = \langle 1, 2, \widehat{3}, 4 \rangle$ becomes the solution provided by (4). Then $r(p) = 8 < r(q) = 9$, therefore p is a better option than q. A similar conclusion can be obtained for $R = \lambda c$. For $W = 2$, $p = \langle 1, \widehat{2}, \widehat{3}, 4 \rangle$ is the best solution of (2). Following the pessimistic approach the best path for the pipeline is $p^* = \langle 1, 3, 4 \rangle$ $(R(p^*) = \lambda(p^*) c(p^*) = 16)$, which after valve location leads to the solution $q = \langle 1, \widehat{3}, 4 \rangle$. However, $r(p) = 6 < r(q) = 8$, therefore the solution obtained by simultaneous location is, again, better than the one obtained by sequential location.

3 Methodological approach

If the number of valves to be located is not constraining, then the solution is to use one at every node. In this case, the objective function is $r = \gamma$ and the PVLP can be solved by a shortest path algorithm where $\lambda_{ij} c_{ij}$ is the cost of arc (i, j). However, the imposition of an upper bound on the number of valves can produce optimal paths which contain non-optimal subpaths, and thus a straightforward labelling approach is not valid in this context. For instance, $p^* = \langle 1, \widehat{3}, 4 \rangle$ is again the PVLP optimal solution in the network in Figure 2 when $W = 1$, with $r(p^*) = 8$ and $v(p^*) = 1$. However, $q = \langle 1, \widehat{2}, 3\rangle$ and $\sigma_{p^*}(1,3)$ are two paths with the same number of valves, therefore both feasible, but q is a better choice than the latter because $r(q) = 3$ and $r(\sigma_{p^*}(1,3)) = 5$. This shows that looking only at the impact of a path is not sufficient to compare different paths. The algorithm presented in the following associates a label with each feasible path from s to any node. This is accompanied by a dominance test, to compare labels pairwise taking into consideration all factors that influence the impact of a path, and other pruning conditions that predict bounds for the impact of the extension of paths and use them to discard labels as early as possible. At the end of the section some remarks about the determination of a sequential solution of the problem are added.

Focusing on the simultaneous determination of route and valve locations, the set of paths with the best impact for any feasible number of valves can be found by means of a labelling algorithm. The algorithm computes paths from s to every node, and represents each one, denoted by $p(l_x)$, by an indexed label of the form $l_x = [\pi^r_x, \pi^\lambda_x, \pi^c_x, \pi^v_x, \xi_x, \beta_x]$, where

- π^r_x is the impact of the path from s until the latest node of $p(l_x)$ with a valve,
- π^λ_x and π^c_x denote the number of accidents in the segment after the latest valve, and the damage resulting from a leak on that segment, respectively,
- π^v_x denotes the number of valves in $p(l_x)$,
- ξ_x is the label index associated with the path extended to produce $p(l_x)$, and
- β_x is the network node which corresponds to the last node in $p(l_x)$, that is, the network node corresponding to index x.

Given a label l_x such that $\beta_x = i \in \mathcal{N}$ and $(i,j) \in \mathcal{A}$, $j \neq t$, two new labels are created:

$$l_y = [\pi^r_x + (\pi^\lambda_x + \lambda_{ij})(\pi^c_x + c_{ij}),\ 0,\ 0,\ \pi^v_x + 1,\ x,\ j], \quad (5)$$

$$l_z = [\pi^r_x,\ \pi^\lambda_x + \lambda_{ij},\ \pi^c_x + c_{ij},\ \pi^v_x,\ x,\ j]. \quad (6)$$

The first case corresponds to using arc (i,j) following $p(l_x)$ and locating a valve at node j, while l_z corresponds to the same path, but with no valve at j. The terminal node is treated differently because it is the last and, by assumption, it already holds a valve. Thus, if $j = t$ then the only created label is

$$l_y = [\pi^r_x + (\pi^\lambda_x + \lambda_{ij})(\pi^c_x + c_{ij}),\ 0,\ 0,\ \pi^v_x,\ x,\ t]. \quad (7)$$

This procedure generates as many labels associated with a network node as the number of feasible paths generated from s up to that node. Two policies are used to bound the number of labels: a dominance test which compares labels associated with the same network node, and a pruning technique which compares the impact and the number of valves of the paths from s to t. Both are described in the following.

3.1 Dominance test

Let $j \neq t$ be a network node and l_x, l_y be the label indices of two paths from s to j. Because the PVLP does not satisfy the optimality principle, comparing the impacts of the two paths is not sufficient to guarantee an optimal solution until t is reached. However, if $\pi^r_x \leq \pi^r_y$, $\pi^\lambda_x \leq \pi^\lambda_y$, $\pi^c_x \leq \pi^c_y$, $\pi^v_x \leq \pi^v_y$ and at least one of the inequalities is strict, then no feasible path starting with $p(l_y)$ and ending at t has a better impact than an analogous path starting with $p(l_x)$. In this case l_x *dominates* l_y, or l_y *is dominated by* l_x. Moreover, a label l_x is said to be dominated if and only if there is another one l_y, with $\beta_x = \beta_y$, that dominates it.

Dominated candidate labels should be discarded, and a label that becomes dominated by a newer one should be deleted. In other words, a new label l_y is considered only if there is no other l_x such that $\beta_x = \beta_y$ and $\pi^v_x \leq \pi^v_y$, for which $\pi^r_x \leq \pi^r_y$, $\pi^\lambda_x \leq \pi^\lambda_y$, $\pi^c_x \leq \pi^c_y$ and one inequality is strict. Furthermore, if l_y is considered, then all the remaining l_x such that $\beta_x = \beta_y$ with $\pi^v_x \geq \pi^v_y$ and $\pi^r_y \leq \pi^r_x$, $\pi^\lambda_y \leq \pi^\lambda_x$, $\pi^c_y \leq \pi^c_x$, with one strict inequality, can be eliminated.

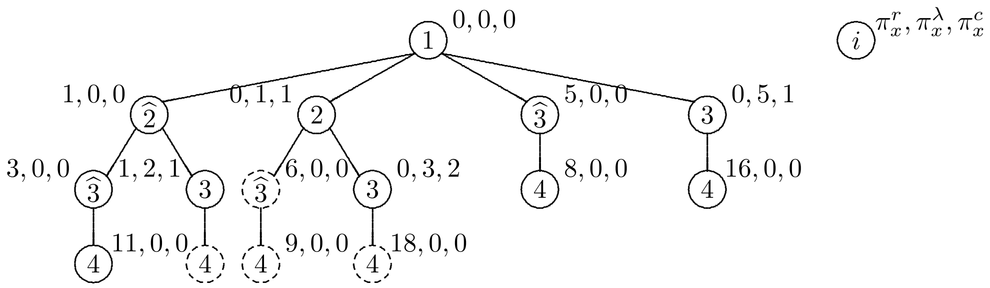

Figure 3: Pipeline and valve location in the network of Figure 2. The dominated labels are marked by a dashed line.

As an example, Figure 3 depicts the tree of all paths from 1 to 4 in the network in Figure 2. The labels associated with these paths are indexed in breadth first order. Some of them, those marked by a dashed line in the picture, are dominated, namely

- $l_8 = [6, 0, 0, 1, 3, 3]$ (corresponding to $p(l_8) = \langle 1, 2, \widehat{3} \rangle$) is dominated by $l_4 = [5, 0, 0, 1, 1, 3]$ (corresponding to $p(l_4) = \langle 1, \widehat{3} \rangle$),
- $l_{13} = [11, 0, 0, 1, 7, 4]$ and $l_{14} = [9, 0, 0, 1, 8, 4]$ (corresponding to $p(l_{13}) = \langle 1, \widehat{2}, 3, 4 \rangle$ and $p(l_{14}) = \langle 1, 2, \widehat{3}, 4 \rangle$, respectively) are dominated by $l_{10} = [8, 0, 0, 1, 4, 4]$ (corresponding to $p(l_{10}) = \langle 1, \widehat{3}, 4 \rangle$),
- $l_{15} = [18, 0, 0, 0, 9, 4]$ (corresponding to $p(l_{15}) = \langle 1, 2, 3, 4 \rangle$) is dominated by $l_{11} = [16, 0, 0, 0, 5, 4]$ (corresponding to $p(l_{11}) = \langle 1, 3, 4 \rangle$).

Therefore none of these labels, l_8, l_{13}, l_{14} nor l_{15}, is included by the algorithm.

3.2 Label pruning

Other useless labels can be discarded by the algorithm. In addition to comparing them directly, which amounts to comparing partial paths starting at s, additional information about what the rest of the path up to t can become is also taken into account. We present lower and upper bounds for the impact of paths in the context of the PVLP labelling algorithm.

As mentioned in Section 2, the impact along a certain sequence of arcs can vary between two extreme cases. The impact function, r, has some useful properties:

- $r(p) = \gamma(p)$, if p has one valve at every intermediate node;
- $r(p) = \lambda(p)c(p)$, if none of the nodes of path p has a valve;
- $\gamma(p) \leq r(p) \leq \lambda(p)c(p)$, for any path p.

Note that, contrary to r, functions γ and λc are independent from the valves that p might have. This means that lower and upper bounds for the impact of a path can be derived from the special case of paths with one valve per node and no valve at all. Because a label corresponds to a path from s to a certain node, and the rest of the path up to t is initially unknown, the properties just stated cannot be applied directly. Yet, minimizing γ from any node to t can be done by means of a shortest path algorithm that considers the cost of each arc $(i, j) \in \mathcal{A}$ to be $\lambda_{ij}c_{ij}$. Let $\mathcal{T}_t^\gamma$ denote the tree of the best path from any node to t in terms of γ, let $\mathcal{T}_t^\gamma(i)$ represent the path from i to t in $\mathcal{T}_t^\gamma$ and $\pi_i^\gamma = \gamma(\mathcal{T}_t^\gamma(i))$, $i \in \mathcal{N}$. The following result holds.

Proposition 1. *Let l_x be a label such that $\beta_x = i \in \mathcal{N}$. Then,*

- $\pi_x^r + \pi_i^\gamma$, *if i has a valve in $p(l_x)$, or*
- $\pi_x^r + \min_{(i,j)\in\mathcal{A}}\{(\pi_x^\lambda + \lambda_{ij})(\pi_x^c + c_{ij}) + \pi_j^\gamma\}$, *otherwise,*

are lower bounds on the impact of a path starting with $p(l_x)$.

Proof. We will prove the first result. The other one can be proved analogously. If a valve is located at node i, then

$$\pi_x^r + \pi_i^\gamma = r(p(l_x)) + \pi_i^\gamma \leq r(p(l_x)) + \gamma(p),$$

for any path p from i to t, because $\mathcal{T}_t(i)$ is optimal. By Property 3,

$$r(p(l_x)) + \gamma(p) \leq r(p(l_x)) + r(p)$$

and, again because there is a valve at node i,

$$r(p(l_x)) + r(p) = r(p(l_x) \diamond p),$$

which concludes the proof. □

Whereas the lower bounds in Proposition 1 can be obtained by computing a shortest path from any node in $\mathcal{N}$ to t, finding similar upper bounds for the impact would require minimizing the function λc. These solutions may not provide upper bounds for the impact of every path, but their impact can in fact be used as an upper bound on the impact of the PVLP solutions for every feasible number of valves. Let $p^*_{\lambda c}(i)$ represent the best path from any node i to t in $(\mathcal{N}, \mathcal{A})$, in terms of λc.

Proposition 2. *Let l_x be a label such that $\beta_x = i$ and $\pi_x \leq W$, $i \in \mathcal{N}$. Then,*

- $\pi_x^r + \lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i))$, *if i has a valve in $p(l_x)$, or*
- $\pi_x^r + \lambda(p^*_{\lambda c}(j))c(p^*_{\lambda c}(j))$, *where j is the last node in $l(p_x)$ with a valve, otherwise,*

are upper bounds on the impact of every feasible path with minimum impact starting as $p(l_x)$.

Proof. We only prove the first part, the other proof being similar. First assume that node i contains a valve. Then, because we can consider, with no loss of generality, that $p^*_{\lambda c}(i)$ does not contain valves in intermediate nodes, and by Property 2,

$$\pi_x^r + \lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i)) = r(p(l_x)) + r(p^*_{\lambda c}(i)) = r(p(l_x) \diamond p^*_{\lambda c}(i)),$$

and

$$v(p(l_x) \diamond p^*_{\lambda c}(i)) = v(p(l_x)) = \pi_x^v \leq W,$$

since l_x is feasible. Thus $p(l_x) \diamond p^*_{\lambda c}(i)$ is a feasible path with an impact $\pi_x^r + \lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i))$. If there is another feasible path with the same form $p(l_x) \diamond p$ but such that $r(p^*_{\lambda c}(i)) < r(p)$, then as i contains a valve

$$r(p(l_x) \diamond p) = r(p(l_x)) + r(p) > \pi_x^r + \lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i)) = r(p(l_x) \diamond p^*_{\lambda c}(i)),$$

therefore $p(l_x) \diamond p$ does not have minimum impact. □

Again the objective function λc is not additive and its minimization does not satisfy the principle of optimality. However, it is simple to show that at least one of its optimal solutions is also a non-dominated path of the bicriteria problem defined by

$$\min_{p \in \mathcal{P}} \{\lambda(p), c(p)\}. \tag{8}$$

Let $\lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i))$ be the optimal value of λc for paths from i to t and assume that any solution p with the same objective value is dominated. This means that for a given p there is another path q such that $\lambda(q) \leq \lambda(p)$ and $c(q) < c(p)$, or $\lambda(q) < \lambda(p)$ and $c(q) \leq c(p)$. If λ and c are both positive this implies $\lambda(q)c(q) < \lambda(p)c(p)$, thus p is not optimal. Otherwise $\lambda(q) = 0$ or $c(q) = 0$, and there is another solution, simultaneously optimal for λc and non-dominated (just take q such that $\lambda(q) = 0$ and $c(q)$ is minimum, or $c(q) = 0$ and $\lambda(q)$ is minimum).

Finding the non-dominated solutions of (8) can be very demanding. Instead of computing all these solutions, others can be determined efficiently and provide another upper bound for the impact of feasible paths from s to t (possibly not as tight as the minimum of λc in $\mathcal{P}$). For instance, let p^*_λ and p^*_c be the lexicographically best paths in terms of (λ, c) and in terms of (c, λ), respectively. These are two particular non-dominated solutions of (8), which can be computed by a shortest path labelling algorithm adapted in a way that a new label is considered if it improves the first objective function, or else it improves the second while there is tie on the first. Moreover, $\lambda(p^*_\lambda)c(p^*_\lambda)$ and $\lambda(p^*_c)c(p^*_c)$ are both upper bounds on $\lambda(p)c(p)$, for any $p \in \mathcal{P}$. Finally note that

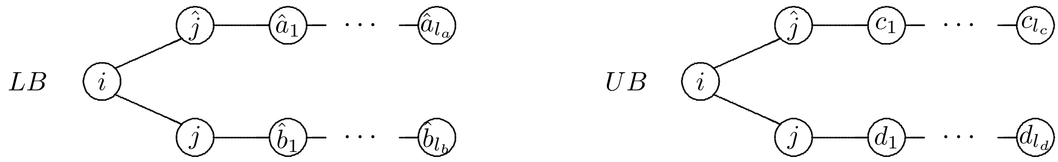

Figure 4: Paths corresponding to the bounds generated from scanning a given label.

paths p^*_λ and p^*_c can still be replaced simply by the optimal solutions with respect to λ or with respect to c, respectively.

For the integration of bounds in the algorithm, the values LB_x and UB_x, a lower bound and an upper bound on the impact of the paths with l_x as initial part, are stored for any index x. These values are updated according to Propositions 1 and 2, and to Figure 4. Their aim is to predict the least and the largest possible impact of a path for which an intermediate label is known, so that it can be compared to others. Let l_x be a label associated with a node i, and l_y, l_z be the new labels produced as previously for $j \neq t$. The best impact value for the paths starting with $p(l_x) \diamond \langle i, \widehat{j} \rangle$ and $p(l_x) \diamond \langle i, j \rangle$ are at least

$$\pi^r_x + (\pi^\lambda_x + \lambda_{ij})(\pi^c_x + c_{ij}) + \pi^\gamma_j$$

and

$$\pi^r_x + \min_{(j,k)\in\mathcal{A}} \{(\pi^\lambda_x + \lambda_{ij} + \lambda_{jk})(\pi^c_x + c_{ij} + c_{jk}) + \pi^\gamma_k\}.$$

Therefore these are impact lower bounds for the paths produced from labels l_y and l_z, LU_y and LU_z, respectively. Recall that these values may correspond to paths with more valves than what is allowed and thus may be unreachable lower bounds. As for the greatest impact value of a PVLP optimal solution, assuming it contains the paths $p(l_y)$, $p(l_z)$, it does not exceed

$$\pi^r_x + (\pi^\lambda_x + \lambda_{ij})(\pi^c_x + c_{ij}) + \pi^{\lambda c}_j$$

and

$$UB_x,$$

respectively, where $\pi^{\lambda c}_i$ represents an upper bound for $\lambda(p^*_{\lambda c}(i))c(p^*_{\lambda c}(i))$ (or the exact optimal value if known). Contrarily to the lower bounds, these values are always assigned to feasible paths although there may exist others with less impact, and thus better solutions.

The impact bounds information can be used in two ways: first to predict and compare the impact of paths from s to t, as shown above, and second to keep record of the best impact of the paths which can be generated from a set of partial labels for each feasible number of valves. In the first case,

- a new label l_y is considered only if there is no other l_x such that $\beta_x = \beta_y$ and $\pi^v_x \leq \pi^v_y$, for which $UB_x > LB_y$;
- if the decision be to take l_y then other labels l_x such that $\beta_x = \beta_y$ with $\pi^v_x \geq \pi^v_y$ for which $LB_x > UB_y$ can be deleted.

As for the second case, let $\bar{r}_v$ represent the greatest impact value that a best solution found with v valves can have, for $v = 0, \dots, W$. Before the labelling begins $\bar{r}_v$ is set to $\pi_s^{\lambda c}$, $v = 0, \dots, W$. Afterwards a new label l_x is inserted only if $LB_x \leq \bar{r}_v$ for all $v = 0, \dots, \pi_x^v$, and in this case $\bar{r}_v$ is updated if UB_x improves it, for all $v = \pi_x^v, \dots, W$.

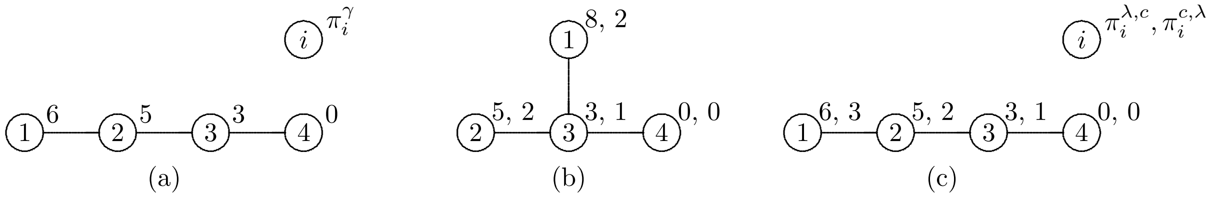

Figure 5: Trees of the best path from 1 to every node in the network of Figure 2, in terms of (a) γ, (b) (λ, c) and (c) (c, λ).

Going back to the previous example, in order to apply the pruning just described, consider the trees depicted in Figure 5. These show the solutions to the best path problem from any node to t in the network $(\mathcal{N}, \mathcal{A})$, for the objective function γ (Figure 5(a)), for function c among the optimal paths for λ, that is, for (λ, c) (Figure 5(b)), and for function λ among the optimal paths for c, that is, (c, λ) (Figure 5(c)). This provides the following information on the values of π^γ, $\pi^{\lambda c}$ and $\bar{r}$:

i	1	2	3	4
π_i^γ	6	5	3	0

i	1	2	3	4
$\pi_i^{\lambda c}$	18	10	3	0

v	0	1	2
$\bar{r}_v$	16	16	16

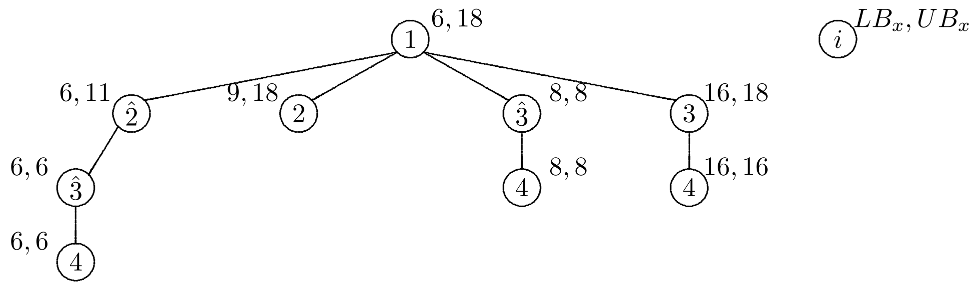

Figure 6: Label bounds generated by the pipeline and valve location in the network of Figure 2.

Figure 6 shows the tree of labels obtained applying the dominance test together with the pruning techniques to the network of Figure 2. Besides the dominated labels already indicated in Figure 3, others are also eliminated:

- l_7 (corresponding to $p(l_7) = \langle 1, \widehat{2}, 3 \rangle$), because $UB_7 = 11 > UB_4 = 8$ and l_4 is associated with a path with one valve only too, and
- l_9 (corresponding to $p(l_9) = \langle 1, 2, 3 \rangle$), because $p(l_9)$ has no valves and $UB_9 = 18 > \bar{r}_0 = 16$.

The paths $\langle 1, \widehat{2}, \widehat{3}, 4 \rangle$, $\langle 1, \widehat{3}, 4 \rangle$ and $\langle 1, 3, 4 \rangle$ are the PVLP solutions. Note also that $\bar{r}$ is updated as new labels are inserted.

3.3 Algorithm

As mentioned earlier in this section. We solve the PVLP by means of a labelling algorithm adapted from classical labelling algorithms for the shortest path problem, like Dijkstra [5] for instance. Each path starting in s is associated with a label, and with the bounds for the impact of the paths it can generate up to t. Initially the path containing s as the single node is taken. In the following iterations a label is selected to be scanned and is then used to extend the corresponding path according to rules (5)–(7). Extensions are accompanied by the dominance and pruning tests. The dominance and pruning tests are applied to the extensions, prior to their acceptance as valid non-dominated candidate labels. In case of acceptance, analogous tests are used to trim other useless temporary labels. The labelling, dominance test and pruning have already been defined.

Given that most label comparisons depend on the number of safety valves in the corresponding paths, and considering that the maximum feasible number of valves, W, is one of the problem's input, and generally a small value, the storage of labels is made with disjoint sets for each number of valves $v = 0, \ldots, W$, and each node $i \in \mathcal{N}$. A $(W+1) \times n$ matrix L is defined. For every feasible number of valves $v = 0, \ldots, W$ and node j, L_{vj} contains the paths from s to j with v valves, which are not dominated by other path with the same number of valves. Each position in L is usually associated with more than one path.

The pseudo-code of Algorithm 1 summarizes the method described for the PVLP. The working variable X represents a set with the labels that have not been scanned. A last note to reduce the number of labels generated is that it is sufficient to store a single label associated with t, the best in terms of r.

Algorithm 1. *Determination of feasible pipeline and valves locations*

Pre-processing
$nX \leftarrow 1$; Insert $l_{nX} = [0,0,0,0,-,s]$ and bounds $(\pi_s^{\gamma}, \pi_s^{\lambda c})$
`While` $X \neq \emptyset$ `Do`
  $x \leftarrow$ element in X; $X \longleftarrow X - \{x\}$; $i \leftarrow \beta_x$
  `For any` $(i,j) \in \mathcal{A}$ `Do`
    `If` $j = t$ `Then`
      `If` $\pi_x^r + (\pi_x^{\lambda} + \lambda_{it})(\pi_x^c + c_{it}) \leq \pi_y^r$ for all $l_y \in L_{dt}$, $d = 0, \ldots, \pi_x^v$ `Then`
        $nX \leftarrow nX + 1$
        Insert $l_{nX} = [\pi_x^r + (\pi_x^{\lambda} + \lambda_{it})(\pi_x^c + c_{it}), 0, 0, \pi_x^v, x, t]$ and bounds (π_{nX}^r, π_{nX}^r)
    `Else`
      $aux \leftarrow \pi_x^r + \max_{(j,k)\in\mathcal{A}}\{(\pi_x^{\lambda} + \lambda_{ij} + \lambda_{jk})(\pi_x^c + c_{ij} + c_{jk}) + \pi_k^{\gamma}\}$
      `If` Check label $(\pi_x^r, \pi_x^{\lambda} + \lambda_{ij}, \pi_x^c + c_{ij}, aux, UB_x)$ `Then`
        $nX \leftarrow nX + 1$
        Insert $l_{nX} = [\pi_x^r, \pi_x^{\lambda} + \lambda_{ij}, \pi_x^c + c_{ij}, \pi_x^v, x, j]$ and bounds (aux, UB_x)
        Eliminate labels dominated by l_{nX} and update $\bar{r}$
      `If` $\pi_x^v + 1 \leq W$ `Then`
        $aux \leftarrow \pi_x^r + (\pi_x^{\lambda} + \lambda_{ij})(\pi_x^c + c_{ij})$
        `If` Check label $(aux, 0, 0, aux + \pi_j^{\gamma}, aux + \pi_j^{\lambda c})$ `Then`
          $nX \leftarrow nX + 1$
          Insert $l_{nX} = [aux, 0, 0, \pi_x^v + 1, x, j]$ and bounds $(aux + \pi_j^{\gamma}, aux + \pi_j^{\lambda c})$
          Eliminate labels dominated by l_{nX} and update $\bar{r}$

Algorithm 1 is followed by the subroutines described in Procedures 1 – 4. The first deals with the computation of the initial bounds, prior to the labelling process, the second is used to verify whether a candidate label should be further considered and added to the set X or not, and finally Procedures 3 and 4 implement operations to manage X, namely the addition of new elements, the removal of dominated labels, and an update of the bounds on the impact of paths with any feasible number of valves.

Procedure 1. ***Pre-processing***

$\mathcal{T}_t^{\gamma} \leftarrow$ tree of the best paths from any node to t with respect to γ
$\mathcal{T}_t^{\lambda,c} \leftarrow$ tree of the best paths from any node to t with respect to (λ, c)
$\mathcal{T}_t^{c,\lambda} \leftarrow$ tree of the best paths from any node to t with respect to (c, λ)
`For` $j \in \mathcal{N}$ `Do`
  $\pi_j^{\gamma} \leftarrow \gamma(\mathcal{T}_t^{\gamma}(j))$; $\pi_j^{\lambda,c} \leftarrow \lambda(\mathcal{T}_t^{\lambda,c}(j))c(\mathcal{T}_t^{\lambda,c}(j))$; $\pi_j^{c,\lambda} \leftarrow \lambda(\mathcal{T}_t^{c,\lambda}(j))c(\mathcal{T}_t^{c,\lambda}(j))$; $\pi_j^{\lambda c} \leftarrow \max\{\pi_j^{\lambda,c}, \pi_j^{c,\lambda}\}$
  $L_{vj} \leftarrow \emptyset$, for any $v \in \{0, \ldots, W\}$
$\bar{r}_v \leftarrow \min\{\pi_s^{c,\lambda}, \pi_s^{\lambda,c}\}$, for any $v \in \{0, \ldots, W\}$

Procedure 2. ***Check label*** $(\pi_x^r, \pi_x^{\lambda}, \pi_x^c, \pi_x^v, LB_x, UB_x)$

`If` $(\pi_x^r, \pi_x^{\lambda}, \pi_x^c)$ is not dominated in $L_{d\beta_x}$ `and` $LB_x \leq UB_y$ for all $l_y \in L_{d\beta_x}$ `and` $LB_x \leq \bar{r}_d$, $d = 0, \ldots, \pi_x^v$
`Then` Return `True`
`Else` Return `False`

Procedure 3. ***Insert*** $l_x = [\pi_x^r, \pi_x^{\lambda}, \pi_x^c, \pi_x^v, \xi_x, \beta_x]$ ***with bounds*** (LB_x, UB_x)

Set l_x and (LB_x, UB_x); $L_{\pi_x^v j} \leftarrow L_{\pi_x^v j} \cup \{x\}$; $X \leftarrow X \cup \{x\}$

Procedure 4. ***Eliminate labels dominated by*** l_x ***and update*** $\bar{r}$

`For` $d = \pi_x^v, \ldots, W$ `Do`
  $\bar{r}_d \leftarrow \min\{\bar{r}_d, UB_x\}$
  `For` $l_y \in L_{d\beta_x} - \{x\}$ `Do`
    `If` $(\pi_x^r, \pi_x^{\lambda}, \pi_x^c)$ dominates l_y `or` $UB_x < LB_y$ `Then`
      $X \leftarrow X - \{y\}$; $L_{d\beta_x} \leftarrow L_{d\beta_x} - \{y\}$

3.4 Sequential determination of pipeline and valve locations

The first part of a sequential approach consists in finding the pipeline route. If an optimistic impact function $R = \gamma$ is considered this first part can be solved by a shortest path algorithm in terms of arc costs given by $\lambda_{ij}c_{ij}$, $(i, j) \in \mathcal{A}$. Otherwise, if a pessimistic impact $R = \lambda c$ is used, then the route to locate the pipeline can be determined by means of an adaptation of a bicriteria path algorithm, summarized in Algorithm 2.

Algorithm 2. ***Determination of pipeline location with pessimistic impact function***

$nX \leftarrow 1$; Insert label $l_{nX} = [0, 0, -, s]$
$BestT \leftarrow -$; $BestImpact \leftarrow \pi_s^{\lambda c}$
`While` $X \neq \emptyset$ `Do`
  $x \leftarrow$ element in X; $X \longleftarrow X - \{x\}$; $i \leftarrow \beta_x$
  `For` any $(i, j) \in \mathcal{A}$ `Do`
    `If` $j = t$ `Then`
      `If` $(\pi_x^{\lambda} + \lambda_{it})(\pi_x^c + c_{it}) < BestImpact$ `Then`
        $nX \leftarrow nX + 1$; Insert $l_{nX} = [\pi_x^{\lambda} + \lambda_{it}, \pi_x^c + c_{it}, x, t]$
        $BestT \leftarrow nX$; $BestImpact \leftarrow (\pi_x^{\lambda} + \lambda_{it})(\pi_x^c + c_{it})$

Else

If $(\pi_x^\lambda + \lambda_{ij}, \pi_x^c + c_{ij})$ is not dominated Then

$nX \leftarrow nX + 1$; Insert $l_{nX} = [\pi_x^\lambda + \lambda_{ij}, \pi_x^c + c_{ij}, x, j]$

Eliminate labels dominated by l_{nX}

$p^* \leftarrow$ path associated with $BestT$

The second part of the sequential approach is dedicated to the valve locations along a certain path and does not depend on the impact function considered in the first part. The process differs from the one used for the simultaneous approach in that only the arcs of the given path can be used. A pseudo-code to solve this problem is shown in Algorithm 3. Note that, if the trees $\mathcal{T}_t^\lambda$ and $\mathcal{T}_t^c$ of the best paths from any node to t with respect to λ and c, respectively, are known, then we do not have to consider the labels such that $(\pi_x^\lambda + \lambda_{ij} + \lambda(\mathcal{T}_t^\lambda(j)))(\pi_x^c + c_{ij} + c(\mathcal{T}_t^c(j)))$ is greater than other bounds obtained for the paths up to t. In Algorithm 3 the use of lower and upper bounds to discard some of the labels produced can be similar to Algorithm 1.

Algorithm 3. ***Determination of valves location along pipeline*** p^*

$nX \leftarrow 1$; Insert label $l_{nX} = [0, 0, 0, 0, -, s]$

While $X \neq \emptyset$ Do

$x \leftarrow$ element in X; $X \longleftarrow X - \{x\}$; $i \leftarrow \beta_x$

For any $(i, j) \in p^*$ Do

If $j = t$ Then

If $\pi_x^r + (\pi_x^\lambda + \lambda_{it})(\pi_x^c + c_{it}) \leq \pi_y^r$ for all $l_y \in L_{dt}$, $d = 0, \ldots, \pi_x^v$ Then

$nX \leftarrow nX + 1$; Insert $l_{nX} = [\pi_x^r + (\pi_x^\lambda + \lambda_{it})(\pi_x^c + c_{it}), 0, 0, \pi_x^v, x, t]$

Else

If Check label $(\pi_x^r, \pi_x^\lambda + \lambda_{ij}, \pi_x^c + c_{ij})$ Then

$nX \leftarrow nX + 1$

Insert $l_{nX} = [\pi_x^r, \pi_x^\lambda + \lambda_{ij}, \pi_x^c + c_{ij}, \pi_x^v, x, j]$

Eliminate labels dominated by l_{nX}

If $\pi_x^v + 1 \leq W$ Then

If Check label $(\pi_x^r + (\pi_x^\lambda + \lambda_{ij})(\pi_x^c + c_{ij}), 0, 0)$ Then

$nX \leftarrow nX + 1$; Insert $l_{nX} = [\pi_x^r + (\pi_x^\lambda + \lambda_{ij})(\pi_x^c + c_{ij}), 0, 0, \pi_x^v + 1, x, j]$

Eliminate labels dominated by l_{nX}

3.5 Examples

To illustrate the algorithm, we first consider the real case discussed in [7]. We then apply the algorithm to an artificial instance of a grid network representing a small area.

The ESPO pipeline [10] is a 4130 km long pipeline constructed to export Russian crude oil to Japan, China and Korea. It crosses very sensitive natural areas. Grigoriev and Grigorieva [7] have studied a 47.5 km long linear segment of this pipeline, consisting of 97 pipes of approximately 500 m each. The terrain surface around these segment is depicted in Figure 8, where the altitudes at the extremes of each pipe are in hundreds of meters and the abscissae are measured in kilometers. The ecological damage around each pipe is defined as

$$c_{ij} = \begin{cases} 10 & \text{if } (i,j) \text{ crosses small rivers,} \\ 5 & \text{if } (i,j) \text{ crosses a water stream,} \\ 3 & \text{if } (i,j) \text{ crosses a road or railway,} \\ 1 & \text{otherwise,} \end{cases} \qquad (i,j) \in \mathcal{A},$$

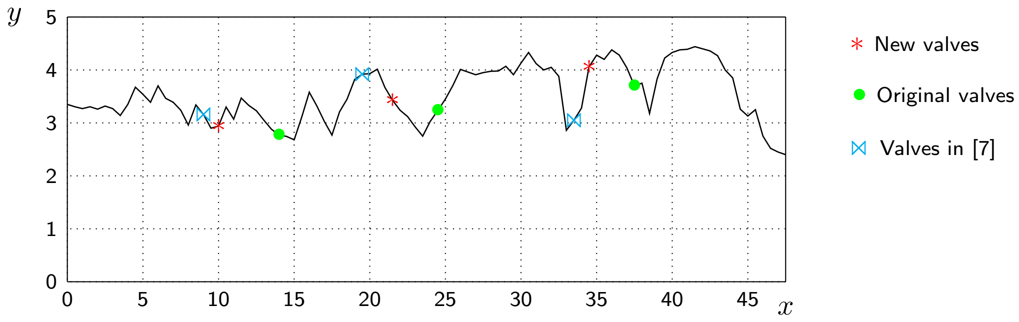

Figure 7: Terrain characteristics of a segment of the ESPO oil pipeline

and are presented in Figure 8. For rivers and lakes wider than 30 m, the installation of valves on both sides is recommended, even though this situation does not occur in the example. Because it is assumed that the pipeline is already installed, only the optimal valve location has to be determined. Figure 7 shows the solution that is currently implemented, the solution obtained with the procedure developed in [7], and the solution obtained by solving problem (4) with a technique similar to that presented in [8]. Note that [7] considers a bottleneck function to measure the environmental damage, whereas our algorithm uses an additive measure, which explains the different locations found. The partial and total values of the environmental impact given be these three approaches are summarized in Table 1. As Grigoriev and Grigorieva stress, in the solution currently implemented the valves are almost equidistant, whereas their solution installs mores valves in areas with higher potential damage. Besides the fact that [7] and Algorithm 1 consider different impact functions, the first model does not take the number of accidents into account, whereas the second includes it in the objective function, which results in slightly different valve locations.

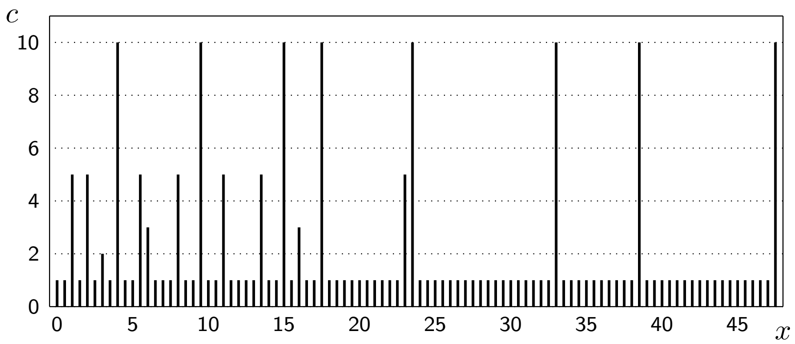

Figure 8: Weights of ecological damage along the pipeline depicted in Figure 7

As a second example we consider the area depicted in the left part of Figure 9, which can be represented by the grid network on the right. The values associated with each arc in the picture represent c_{ij}, whereas we assume that $c_{ij} = 1$ for the remaining arcs where the value is omitted, $\lambda_{ij} = 1$ for any arc (i, j) and $W = 1$. The solution returned by Algorithm 1 is $p^* = \langle 11, 12, 13, \widehat{14}, 15, 10, 5\rangle$, with only one valve installed at node 14. The total impact of this path is $r(p^*) = 24$, and it can be seen that this is not the only optimal solution. For instance,

	Original	[6]	This paper
$r(\sigma_p(s, v_1))$	5.516	3.600	4.359
$r(\sigma_p(v_1, v_2))$	3.802	4.056	3.663
$r(\sigma_p(v_2, v_3))$	2.992	3.910	3.693
$r(\sigma_p(v_3, t))$	2.998	3.743	3.593
$r(p)$	375.063	368.000	364.481

Table 1: Impact values of the ESPO oil pipeline,p, after the installation of valves v_1, v_2 and v_3

$p^* = \langle 11, 6, 7, 8, \widehat{9}, 10, 5\rangle$ has also only one valve and yields $r(q) = 24$. Moreover, in this case the same solution is obtained whether a sequential (for $R = \gamma$ or $R = \lambda c$) or a simultaneous approach is used.

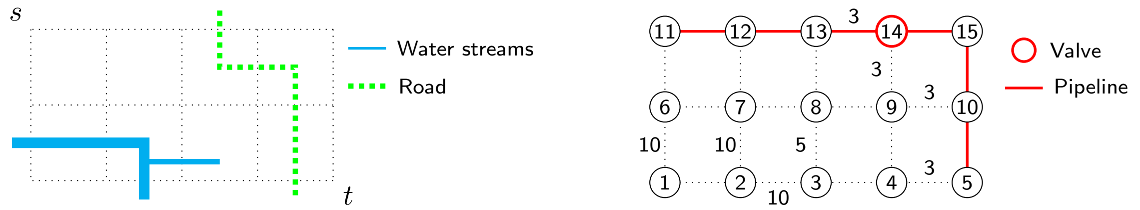

Figure 9: Example of a problem defined on a grid network (left) and solution (right)

4 Computational results

Algorithm 1 was coded in C, compiled with `gcc 4.2.1` and optimization option `-O3`, and tested on randomly generated networks, in order to evaluate its empirical performance. This program is also compared with implementations of Algorithms 2 and 3, under the same conditions, to complement the examples on the previous section. The tests were performed on a Pentium 4 at 3 GHz, with 512 Kb of cache memory and 1Gb of RAM, with an openSUSE Linux 10.3 operative system. In the following the average results of 20 different instances for each dimension of the data sets are reported.

4.1 Test bed

In order to obtain realistic networks, we have generated square grid graphs of the form $[1, p] \times [1, p]$, where p is given. The nodes in the network are the intersection between horizontal and vertical lines along the grid. The horizontal and vertical lines, together with diagonal, represent arcs that link each node to all its neighbours in the grid. Such a grid, depicted in Figure 10, has $n = 4p^2$ nodes and $m = 4(2p - 1)(p - 1)$ arcs. In the experiments performed $p = 4, 6, \dots, 24$ and $W = 10$. The source node is the bottom border vertex of the grid, whereas the top border is the sink.

Initially, the average number of spills λ_{ij}, edge number of oil barrels b_{ij} and edge damages d_{ij} are all equal to 1. These values are then modified as follows (where all parameters are integer):

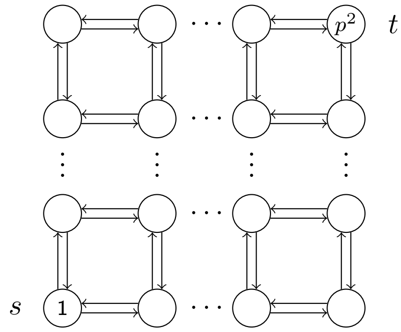

Figure 10: Grid graph

1. Randomly select a point (x, y) of the grid and generate a square with side of length 2α centered at (x, y), where α is randomly selected in $[1, p/10]$.
2. Select β randomly in $[1, 2]$.
3. For all $(i, j) \in \mathcal{A}$ contained in the square centered at (x, y), do: $\lambda_{ij} \leftarrow \lambda_{ij} + \beta$.
4. Repeat Steps 1 to 3 for $\lfloor m/4 \rfloor$ times.

The numbers of barrels and damage parameters are modified similarly, the first with $\beta = 1$ and the second with β chosen uniformly in $[0, 2]$.

In a second set of tests, random networks were considered. To obtain these instances we have modified the network generator code `sprand.c` which is part of the library SPLIB, publicly available at `http://www.avglab.com/andrew/soft.html`, [3]. The modifications aimed at generating all the parameters associated with each arc and making the network undirected. The sizes of the instances obtained were determined by $n = 3\,000, 5\,000, 7\,000, 9\,000$ and $m = an$ arcs, for average degrees $a = 2, 4, 5, 10, 20$. The arc parameters λ_{ij} were uniformly generated between 1 and $\Lambda = 1, 5, 10$, whereas c_{ij} were uniformly calculated between 1 and $C = 1, 5, 10, 20$, $(i, j) \in \mathcal{A}$. Moreover $W = 10$.

4.2 Test results

When running the tests for grid networks the number of labels generated by Algorithm 1 increased very quickly with p, and so did the running times, summarized in Table 2. For instance, it took more than 30 minutes to find the pipeline and valve locations in 24×24 grids.

p	4	6	8	10	12	14	16	18	20	22	24
	0	1	7	38	130	929	3 198	10 668	42 324	131 237	141 960

Table 2: Average running times (in milliseconds) for grid networks of size $p \times p$.

The algorithm's performance was very different for the set of random networks, as the CPU times in Tables 3 and 4 show. All of these instances were solved in less than one minute. The running times increased with the size of the network, depending on both n and a, although the growth seems to be more sensitive to the variation of the average degree rather than to the number of nodes. The range of the parameters associated with each arc is also a factor to be taken into account. The times observed increase with Λ and C, and it can also be noted that they are very similar regarding cases of the same size and with the same value of ΛC.

		$n = 3\,000$					$n = 5\,000$				
Λ	C	$a = 2$	$a = 4$	$a = 5$	$a = 10$	$a = 20$	$a = 2$	$a = 4$	$a = 5$	$a = 10$	$a = 20$
1	1	2	3	3	7	20	2	7	7	16	36
1	5	2	4	4	8	24	3	8	10	17	44
1	10	2	3	4	9	36	4	8	10	18	56
1	20	3	4	4	9	33	4	10	10	21	65
5	1	2	3	3	8	21	3	7	10	20	47
5	5	2	3	4	10	37	4	10	11	22	70
5	10	3	3	5	11	56	4	10	12	25	74
5	20	3	4	5	15	73	4	11	12	31	104
10	1	2	4	5	9	24	4	9	10	24	65
10	5	3	3	5	12	22	4	11	12	24	84
10	10	2	4	5	13	63	5	12	14	30	109
10	20	4	4	6	16	81	4	13	14	31	154

Table 3: Average running times (in milliseconds) for random undirected networks.

		$n = 7\,000$					$n = 9\,000$				
Λ	C	$a = 2$	$a = 4$	$a = 5$	$a = 10$	$a = 20$	$a = 2$	$a = 4$	$a = 5$	$a = 10$	$a = 20$
1	1	5	11	14	32	59	7	18	21	47	92
1	5	5	12	16	31	64	8	17	24	56	105
1	10	5	12	16	36	76	9	18	24	58	117
1	20	5	13	16	38	78	8	20	28	66	130
5	1	5	12	17	36	76	9	18	25	60	116
5	5	5	14	16	41	93	10	21	28	63	122
5	10	7	14	19	45	105	8	20	30	68	126
5	20	5	14	22	49	131	9	23	32	76	148
10	1	5	13	18	38	106	7	21	30	67	127
10	5	6	15	20	45	125	10	22	31	70	127
10	10	6	14	21	46	148	10	20	33	73	155
10	20	7	17	21	52	239	10	21	32	80	157

Table 4: Average running times (in milliseconds) of for random undirected networks.

In order to compare the results obtained by the simultaneous resolution of the PVLP in Algorithm 1 and the sequential approaches in Algorithms 2 and 3, the latter algorithm was only applied to a subset of the instances just described. Tables 5 and 6 summarize some of the results obtained, the first one comparing the quality of the solutions provided, and the second showing the running times of the two considered sequential approaches. For most of the random instances considered in the general tests for Algorithm 1 the number of valves installed was smaller than $W = 10$. Under such conditions the number of valves is not constraining, and thus the optimistic sequential and the simultaneous approaches return the same solution. Tables 5 and 6 show the results on a subset of the original test bed (thus for $W = 10$), however by fixing W to a lower value the differences observed between the two algorithms are expected to increase. In these cases the impact of solutions determined by Algorithm 2 and by Algorithm 1 is still the same for many of the instances. Otherwise, the values are close but the impact computed by Algorithm 2 is never better. It is also worth noting that when the impacts obtained are different, the optimistic sequential approach always uses more valves than the simultaneous algorithm. As for the pessimistic sequential approach, the impact of the optimal solution was always much worse than with the simultaneous technique, more than double in many of the cases, and slightly smaller for instances with a higher average degree. Also the number of valves used in the solutions computed with this method was larger than with a simultaneous computation.

		$n = 7\,000$										$n = 9\,000$									
		$a = 5$					$a = 10$					$a = 5$					$a = 10$				
		S	O		P		S	O		P		S	O		P		S	O		P	
Λ	C	(2)	(1)	(2)	(1)	(2)	(2)	(1)	(2)	(1)	(2)	(2)	(1)	(2)	(1)	(2)	(2)	(1)	(2)	(1)	(2)
5	10	2.0	1.00	2.0	2.14	4.8	3.4	1.00	3.4	1.30	4.0	0.6	1.01	1.6	2.51	5.2	2.5	1.00	2.5	2.13	5.3
5	20	2.7	1.02	3.7	2.08	5.0	3.8	1.00	3.8	1.42	4.3	1.6	1.00	1.6	2.59	5.2	2.8	1.00	2.8	2.07	5.3
10	10	2.1	1.00	2.1	2.18	4.8	4.2	1.01	4.3	1.27	4.1	1.6	1.00	1.6	2.77	3.4	2.1	1.00	2.1	2.36	3.5
10	20	3.6	1.00	3.6	2.14	5.1	5.1	1.01	5.2	1.55	3.8	1.6	1.00	1.6	2.87	3.8	2.6	1.00	2.6	2.34	3.9

Legend: S: simultaneous approach. O: optimistic sequential approach. P: pessimistic sequential approach. (1) Ratio between the impact obtained by the sequential and the simultaneous approaches. (2) Number of valves.

Table 5: Average results of the sequential approaches for random undirected networks.

Finally Table 6 reports the running times of the two sequential methods. These methods are simpler, and therefore faster, than the simultaneous approach. In particular, the optimistic version applies a modified shortest path algorithm in order to find the pipeline location, whereas the pessimistic version solves a variant of a bicriteria shortest path problem, which is why the former is generally quicker than the latter. However, this advantage seems to diminish for larger instances.

		$n = 7\,000$				$n = 9\,000$			
		$a = 5$		$a = 10$		$a = 5$		$a = 10$	
Λ	C	O	P	O	P	O	P	O	P
5	10	5	8	10	11	9	12	22	16
5	20	6	10	11	11	10	12	23	17
10	10	6	9	11	9	10	12	20	16
10	20	6	11	11	11	11	14	21	24

Legend: O: optimistic sequential approach. P: pessimistic sequential approach.

Table 6: Average CPU times (in milliseconds) of the sequential approaches for random undirected networks.

5 Conclusions

We have introduced and solved the problem of determining the best location for a pipeline and safety valves between two nodes of a network. Our algorithm associates pipelines to paths in the network. Because the problem does not satisfy the principle of optimality, we have developed a labelling method based on partial values for the number and impact of accidents along the pipeline, while controlling the number of valves installed on the partial paths. The algorithm includes a dominance test and computes bounds on non-dominated labels to prune the set of generated paths. We have also proposed an algorithm that sequentially locates the pipeline and the valves under both an optimistic and a pessimistic evaluation of the risk. Implementations of the three alternative methods were extensively tested on randomly generated instances. The sequential approaches were faster than the simultaneous algorithm but their solutions exhibited a higher risk value. The running times of the simultaneous location algorithm increase with the density of the graph because of the larger number of labels that need to be considered. However, the running time of this algorithm is still modest, around two seconds for undirected networks with 7 000 nodes and 14 000 arcs, or 9 000 nodes and 18 000 arcs.

Acknowledgments The authors thank Alexander Grigoriev and Nadejda Grigorieva for providing the data related with the ESPO oil pipeline. This research was partially funded by the Canadian Natural Sciences and Engineering Research Council under grant 39682-10, and by the Portuguese FCT project POSC/U308/1.3/NRE/04. This support is gratefully acknowledged.

References

[1] H. L. Bodlaender, A. Hendriks, A. Grigoriev, and N. V. Grigorieva. The valve location problem in simple network topologies. To appear in *INFORMS Journal on Computing*, 2009.

[2] T. Cáceres, J. A. Mesa, and F. A. Ortega. Locating waste pipelines to minimize their impact on marine environment. *European Journal of Operational Research*, 179:1143–1159, 2007.

[3] B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shorthest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73:129–196, 1996.

[4] J. M. Díaz-Báñez, F. Gómez, and G. T. Toussaint. Computing shortest paths for transportation of hazardous materials in continuous spaces. *Journal of Food Engineering*, 70:293–298, 2005.

[5] E. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:395–412, 1959.

[6] E. Erkut and V. Verter. Modelling of transport risk for hazardous materials. *Operations Research*, 46:625–642, 1998.

[7] A. Grigoriev and N. V. Grigorieva. The valve location problem: Minimizing environmental damage of a spill in long oil pipelines. *Computers & Industrial Engineering*, 57:976–982, 2009.

[8] G. Laporte and M. M. B. Pascoal. Minimum cost path problems with relays. To appear in *Computers & Operations Research*, 2010.

[9] V. Verter and E. Erkut. Hazardous materials logistics: An annotated bibliography. In C. Carraro and A. Haurie, editors, *Operations Research and Environmental Management*, pages 221–267. Kluwer, Boston, 1996.

[10] Wikipedia, the free enciclopedia. Eastern Siberia – Pacific Ocean oil pipeline, 2010. (http://en.wikipedia.org/wiki/Eastern_Siberia_-_Pacific_Ocean_oil_pipeline) Retrieved 18.02.10.