



HAL
open science

Capacité opérative des réseaux de transfert de pétrole

Jorge Rojas d'Onofrio

► **To cite this version:**

Jorge Rojas d'Onofrio. Capacité opérative des réseaux de transfert de pétrole. Gestion et management. INSA de Lyon, 2011. Français. NNT : 2011ISAL0021 . tel-00668722

HAL Id: tel-00668722

<https://theses.hal.science/tel-00668722>

Submitted on 10 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL
DES SCIENCES APPLIQUÉES DE LYON
ÉCOLE DOCTORALE E.E.A
ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE

T H È S E

pour obtenir

Le grade de Docteur

de L'Institut National des Sciences Appliquées de Lyon

Spécialité : AUTOMATIQUE

Préparée par

Jorge ROJAS D'ONOFRIO

Capacité opérative des réseaux de transfert de pétrole

Thèse dirigée par Emmanuel BOUTLEUX et Eric NIEL

préparée au Laboratoire Ampère UMR CNRS 5005, Lyon et avec
la collaboration de PDVSA (Pétroles du Venezuela)

Jury :

<i>Rapporteurs :</i>	I. TODINCA	Professeur (LIFO, Université d'Orléans)
	A. TOGUYENI	Professeur (LAGIS, Ecole Centrale de Lille)
<i>Examineurs :</i>	E. ZAMAÏ	MCF, HDR (G-SCOP, INP Grenoble)
	J. AGUILAR	Professeur (ULA, Venezuela)
	B. ROUX	Professeur (ICJ, INSA de Lyon)
<i>Directeurs :</i>	E. BOUTLEUX	MCF (Ampère, Ecole Centrale de Lyon)
	E. NIEL	Professeur (Ampère, INSA de Lyon)

Capacité opérative des réseaux de transfert de pétrole

Résumé : Cette thèse étudie des systèmes locaux de gestion de transfert de pétrole ayant une architecture de réseau de canalisation. Pour leur représentativité, deux systèmes localisés au Venezuela et appartenant à l'entreprise PDVSA (Pétroles du Venezuela) ont été retenus pour illustrer les méthodes proposées et les valider : le Terminal Maritime de Pétrole de Guaraguao et le Centre de Stockage de Punta de Palmas. Dans ces réseaux des connexions, appelées "alignements", sont établies en ouvrant/fermant des vannes à travers d'un système SCADA (Supervisory Control and Data Acquisition). Le choix d'un alignement doit tenir compte de critères d'optimisation. La minimisation des interférences avec d'autres alignements, liée à la notion de capacité opérative, a été identifiée comme le critère de choix le plus important.

Les contributions de cette thèse reposent sur une modélisation sous forme de graphes, et sur des algorithmes appartenant au domaine de la recherche opérationnelle. Elles contribuent à fournir aux opérateurs de supervision des outils d'analyse permettant d'optimiser le choix des alignements. Des indicateurs permettant de quantifier l'impact des opérations d'alignement ou des défaillances, sur la capacité opérative du système, sont proposés. La minimisation de l'impact sur la capacité opérative, va correspondre à la minimisation des interférences avec des alignements potentiels. Un algorithme de calcul de ces indicateurs, est présenté, ainsi que des algorithmes de recherche de chemin, de détermination d'éléments critiques, et de recherche d'alignements utilisant des pompes. Ces algorithmes sont basés sur des algorithmes classiques s'adressant au problème du plus court chemin, du flot maximum et du nombre maximum de chemins disjoints. Cependant, ils utilisent des méthodes innovantes, comme l'ajout de contraintes considérant l'existence de sous-types d'alignements, le calcul dynamique des coûts des chemins à partir de son impact sur la capacité opérative, et la recherche de chemins via un point intermédiaire obligatoire. Les contributions sont potentiellement applicables dans des domaines autres que le transport de pétrole.

Les algorithmes ont été mis en œuvre en utilisant le langage Python et ont été testés en utilisant les données réelles des réseaux étudiés. L'objectif à moyen terme de ces travaux est le développement d'un logiciel d'assistance à la prise de décision.

Mots clés : Capacité opérative, transport de pétrole, réseau de canalisation, chemins disjoints, recherche opérationnelle, algorithmes de graphes

Operative Capacity of Crude Oil Local Transportation Networks

Abstract : This thesis studies local crude oil transportation systems with a pipe network architecture. Two representative systems, belonging to PDVSA (Venezuelan oil company), have been studied : the Guaraguao Crude Oil Seaport and the Punta de Palmas Tanks Yard. In this systems, connections, called "alignments", are established by opening/closing valves using a SCADA (Supervisory Control and Data Acquisition) system. Alignment choice is made based on optimization criteria. Interferences minimization with other alignments, related to the notion of operative capacity, has been identified as the most important criterion.

The contributions of this thesis are based on graph modelling and algorithms from operational research. The main goal is to provide analysis tools allowing alignment choice optimization. Indexes permitting the quantification of alignments or failures impact on the operative capacity of the system are proposed. Minimizing the impact on the operative capacity will correspond to minimizing interferences with potential alignments. An algorithm computing these indexes is presented, as well as complementary developments such as a path search algorithm, an algorithm for critical elements determination, and algorithm for alignments using pumps. These algorithms are based on classical algorithms for the shortest path problem, the maximum flow problem and the maximum disjoint paths problem. However, they use innovative methods such as adding constraints when considering alignment sub-types, the dynamic computation of path costs based on their impact on operative capacity, and path search considering an obligatory intermediate node. These contributions can potentially be applied in areas other than oil transportation.

The algorithms had been implemented in Python and had been tested using real data from the studied systems. The middle term goal of these works is the development of assistance software for decision making.

Keywords : Operative capacity, crude oil transportation, pipe network, disjoint paths, operational research, graph algorithms

Capacidad operativa de redes de tuberías para el transporte de petróleo

Resumen : Esta tesis estudia sistemas de transporte de petróleo con una arquitectura de red de tuberías. Dos sistemas representativos, pertenecientes a la compañía PDVSA (Petróleos de Venezuela) fueron estudiados : el Terminal Marino de Guara-guao, y el Patio de Tanques de Punta de Palmas. En estos sistemas, las conexiones, llamadas “alineaciones”, se establecen abriendo/cerrando válvulas, a través de un sistema SCADA (Supervisory Control and Data Acquisition). La escogencia de una alineación debe hacerse en base a criterios de optimización. La minimización de las interferencias con otras alineaciones, relacionada con la noción de capacidad operativa, fue identificada como el criterio más importante.

Las contribuciones de esta tesis están basadas en un modelado en forma de grafo y en algoritmos que pertenecen al área de investigación de operaciones. El objetivo es proveer herramientas de análisis a los operadores de supervisión para optimizar la escogencia de las alineaciones. Se proponen indicadores que permiten cuantificar el impacto de las alineaciones o de fallas sobre la capacidad operativa del sistema. La minimización del impacto sobre la capacidad operativa corresponderá a la minimización de la interferencia con potenciales alineaciones. Un algoritmo para el cálculo de estos indicadores es presentado, así como algoritmos de búsqueda de camino, de determinación de elementos críticos, y de búsqueda de alineaciones con dispositivos de bombeo. Estos algoritmos están basados en algoritmos clásicos que tratan los problemas del camino más corto, del flujo máximo y del número máximo de caminos disjuntos. Sin embargo utilizan métodos novedosos, como utilizar especificaciones adicionales para tener en cuenta sub-tipos de alineaciones, el cálculo dinámico del costo de caminos a partir del impacto que producen en la capacidad operativa, y la búsqueda de caminos con un punto intermedio obligatorio. Estas contribuciones son potencialmente aplicables en áreas diferentes al transporte de petróleo.

Los algoritmos fueron implementados en Python y fueron probados utilizando datos reales de los sistemas estudiados. El objetivo a mediano plazo de estos trabajos es el desarrollo de un software de asistencia a la toma de decisiones.

Palabras clave : Capacidad operativa, transporte de petróleo, redes de tuberías, caminos disjuntos, investigación de operaciones, algoritmos de grafos

Remerciements

Au Professeur Eric Niel et au Maître de Conférences Emmanuel Boutleux, directeurs de thèse, pour l'orientation et l'aide apportée à la réalisation de ce travail.

À José Gregorio González, tuteur industriel à la Raffinerie de Puerto La Cruz, PDVSA, et à Jack Márquez, tuteur industriel au DST (District Socialiste Technologique) de Mérida, PDVSA, pour la collaboration et l'orientation apportée.

À la Fondation Gran Mariscal de Ayacucho (Fundayacucho) et à l'Ambassade de France au Venezuela, pour le financement des études doctorales.

À l'équipe d'opérateurs du système SCADA du Terminal Maritime de Guaraguao, PDVSA, spécialement à Fidel Rivera, pour ses informations et le partage de son expertise.

À l'équipe de maintenance du Terminal Maritime de Guaraguao, spécialement à René Figueroa, pour son soutien et les informations apportées.

Au Professeur Bernard Roux pour les connaissances apportées au niveau théorique.

À l'équipe d'opérateurs du système SCADA du Centre de Stockage de Punta de Palmas, PDVSA, pour sa collaboration.

Table des matières

1	Introduction	1
1.1	Contexte	1
1.2	Premières approches	3
1.2.1	Étude des coups de bélier	4
1.2.2	Système Expert	4
1.2.3	Algorithme de recherche exhaustive des alignements possibles	5
1.3	Contribution : Indicateurs de Capacité Opérative	5
1.4	Organisation du manuscrit	6
2	Problématique et État de l'Art	7
2.1	Transport de pétrole	7
2.1.1	Propriétés du pétrole	8
2.1.2	Réseaux de canalisation	9
2.2	Transport de pétrole au Venezuela	11
2.2.1	Terminal Maritime de Guaraguao	11
2.2.2	Centre de Stockage de Punta de Palmas	13
2.3	Problématique des alignements	15
2.3.1	Définition d'un alignement	15
2.3.2	Contraintes sur les alignements	16
2.3.3	Critères de choix des alignements	18
2.4	SCADA	19
2.4.1	SCADA des systèmes de transfert de pétrole	20
2.4.2	Logiciel SCADA	20
2.5	État de l'art	22
2.5.1	Modélisation des réseaux de canalisation	23
2.5.2	Algorithmes de graphes	25
2.6	Conclusion	27
3	Cadre théorique	29
3.1	Modélisation sous forme de graphe	29
3.1.1	Définitions et concepts de base	30
3.1.2	Représentation des graphes	34
3.2	Algorithmes de graphes	37
3.2.1	Problème du plus court chemin	38
3.2.2	Problème du flot maximum	42
3.2.3	Problème du nombre maximum de chemins disjoints	48
3.3	Conclusion	48

4	Contributions	51
4.1	Modélisation des réseaux de canalisation sous forme de graphe non orienté	51
4.2	Indicateurs de Capacité Opérative	55
4.2.1	Relation d'équivalence entre des éléments du réseau	56
4.2.2	Définition	57
4.3	Algorithme de calcul de la Capacité Opérative	58
4.3.1	Modèle de graphe de flot pour des chaînes disjointes	59
4.3.2	Algorithme du flot maximum	64
4.3.3	Validation	65
4.3.4	Capacité Opérative pour des sous-types d'alignement	66
4.3.5	Ensemble d'indicateurs de Capacité Opérative	70
4.4	Développements complémentaires	72
4.4.1	Analyse de criticité	72
4.4.2	Algorithme de recherche de chemin maximisant la capacité opérative	77
4.4.3	Opérations d'alignements forcés	84
4.5	Conclusion	90
5	Mise en œuvre	93
5.1	Langage de programmation	93
5.2	Codage de la structure du réseau de canalisation	94
5.3	Codage des algorithmes	101
5.4	Résultats des tests	103
5.4.1	Terminal Maritime de Guaraguao	104
5.4.2	Centre de stockage de Punta de Palmas	106
5.5	Préconisations pour le développement d'un logiciel d'assistance aux opérateurs	107
5.6	Conclusion	110
6	Conclusion : Bilan, Extensions et Perspectives	111
6.1	Bilan	111
6.2	Extensions	112
6.3	Perspectives	115
A	Mise en œuvre sous Python	117
	Bibliographie	119

Table des figures

2.1	Schéma simplifié du réseau de transport de pétrole du Venezuela. Le centre de stockage de Punta de Palmas et le Terminal Maritime de Guaraguao (PLC) sont encerclés en rouge.	12
2.2	Structure générale du système de transport de pétrole du Terminal Maritime de Guaraguao.	13
2.3	Structure général du système de transport de pétrole du Centre de Stockage de Punta de Palmas.	14
2.4	Alignement (en gris) entre un réservoir et un bas de chargement. . .	16
2.5	Visualisation d'une section du réseau de canalisation du Terminal Maritime de Guaraguao avec le logiciel PlantScape.	21
2.6	Visualisation d'une section du réseau de canalisation du Centre de Stockage de Punta de Palmas avec le logiciel Realflex.	21
3.1	Représentation graphique d'un graphe orienté.	30
3.2	Représentation graphique d'un graphe non orienté.	31
3.3	Représentation graphique et matrice d'incidence d'un graphe orienté.	35
3.4	Représentation graphique et matrice d'incidence d'un graphe non orienté.	35
3.5	Représentation graphique et matrice d'adjacence d'un graphe orienté.	37
3.6	Représentation graphique d'un graphe simple.	37
3.7	Fonctionnement général de l'algorithme de Dijkstra pour le problème du plus court chemin (PCC).	39
3.8	Graphe orienté avec un nœud source 's', et un nœud terminal 't'. Chaque arc est affecté d'un coût positif.	41
3.9	Plus court chemin entre les nœuds 's' et 't'.	42
3.10	Graphe de flot et les valeurs de flot et de capacité de chaque arc (flot/capacité).	44
3.11	Fonctionnement général des algorithmes basés sur la méthode de Ford Fulkerson.	45
3.12	Graphe résiduel correspondant au graphe et à la fonction de flot de la figure 3.10.	46
3.13	Application de la méthode de Ford-Fulkerson à un graphe de flot avec un nœud source 's' et un nœud puits 'p'. A gauche : le graphe de flot et les valeurs de flot et de capacité de chaque arc (flot/capacité). A droite : le graphe résiduel correspondant avec les capacités résiduelles de chaque arc et le chemin d'augmentation utilisé.	49
4.1	Modèle d'automate à états d'une vanne du réseau.	52
4.2	En gris, un segment du réseau.	53

4.3	Modélisation sous forme de graphe non orienté d'un réseau de canalisation contenant des vannes.	54
4.4	Section d'un réseau de transport de pétrole. Les couleurs des réservoirs indiquent leur contenu : blanc : réservoir vide, gris clair : réservoir contenant du pétrole léger, gris foncé : réservoir contenant du pétrole lourd. La taille d'un réservoir indique sa capacité.	56
4.5	Transformations permettant de construire un graphe de flot à partir du modèle de graphe non orienté, tenant compte des contraintes des alignements.	60
4.6	Transformation d'un arc non orienté en deux arcs orientés opposés.	60
4.7	Pour des nœuds avec seulement un arc de sortie, l'assignation de capacités unitaires aux arcs interdit les convergences de flot (a), tandis que des capacités supérieures permettent des convergences du flot (b).	61
4.8	Un nœud avec trois arcs incidents et deux arcs sortants est remplacé par deux nœuds : un nœud connecté aux arcs entrants et un autre connecté aux arcs sortants, les deux nœuds connectés entre eux par un arc de capacité unitaire. Cette transformation interdit la convergence du flot.	61
4.9	Transformation d'un graphe non orienté en un graphe de flot pour des chemins disjoints. Tout les arcs du graphe de flot ont des capacités de un.	63
4.10	Fonction de flot maximale obtenue en utilisant un algorithme Ford Fulkerson dans un graphe de flot, et le graphe non orienté correspondant.	65
4.11	Alignements entre réservoirs et bras de chargement dans un réseau de canalisation. Le gris foncé et le gris clair des réservoirs correspondent à différents types de pétrole.	67
4.12	Partie du graphe de flot pour des chaînes disjointes construit à partir du réseau de la figure 4.11.	68
4.13	Partie du graphe de flot du réseau de la figure 4.11 construit d'après la méthode tenant compte des contraintes additionnelles sur le nombre de sous-types d'alignement.	69
4.14	Alignements trouvés en utilisant une modélisation qui restreint le nombre de réservoirs contenant le premier type de pétrole (gris foncé) à un et le nombre de réservoirs contenant le deuxième type de pétrole à deux.	69
4.15	Réseau de transport de pétrole.	70
4.16	Alignement entre un réservoir et un bras de chargement (en gris).	71
4.17	Opération de maintenance pour deux vannes du réseau.	72
4.18	Graphe non orienté et graphe de flot correspondants à un réseau de canalisation.	73
4.19	Graphes correspondants au cas où la vanne connectant les segments c et d de la figure 4.18 est une vanne indisponible fermée.	74

4.20	Graphes correspondants au cas où la vanne connectant les segments c et d de la figure 4.18 est une vanne indisponible ouverte.	75
4.21	Un réseau simple avec trois vannes critiques bloquantes (B) et une vanne critique permissive (P) pour les opérations de transport de pétrole des réservoirs vers les bras de changement.	76
4.22	Trois graphes correspondant à trois réseaux différents. Les opérations considérées sont les alignements entre les nœuds supérieurs et les nœuds inférieurs. Les nœuds critiques sont colorés en noir.	77
4.23	Fonctionnement général de l'algorithme de recherche de chemin maximisant la capacité opérative.	78
4.24	Chemins disjoints (lignes épaisses) entre les ensembles : (a,b,c,d) et (j,k,l,m) déterminés avec l'algorithme de calcul de la capacité opérative, dans un graphe non orienté.	83
4.25	Chemin entre x et y (pointillés) déterminé avec l'algorithme de recherche de chemin maximisant la capacité opérative pour les ensembles (a, b, c, d) et (j, k, l, m) . Les lignes épaisses correspondent à de possibles chemins disjoints résultants.	84
4.26	Modèle de graphe non orienté (droite) d'un réseau de canalisation (gauche). Le nœud du segment correspondant à la pompe a été supprimé.	85
4.27	Alignement entre le premier (gris foncé) et le deuxième réservoir (blanc) utilisant une pompe. Les vannes en noir sont fermées. Les points blancs montrent le flux de pétrole entre les réservoirs.	89
4.28	Réseau de canalisation. Une vanne fermée présente une défaillance (X).	89
4.29	Alignement entre le premier (gris foncé) et le deuxième réservoir (blanc) utilisant une pompe. Les vannes en noir sont fermées. Les points blancs montrent le flux de pétrole entre les réservoirs. Une vanne fermée présente une défaillance (X).	91
5.1	Section du multiple principal du réseau de Punta de Palmas.	96
5.2	affichage d'une section du réseau de canalisation de Guaraguao.	98
5.3	Réseau théorique avec seulement deux segments connectés aux mêmes éléments étiquetés.	99
5.4	Réseau simple avec 6 éléments étiquetés : les vannes V1, V2 et V3, le réservoir R, et les canalisations d'entrée/sortie E/S_1 et E/S_2.	100
5.5	Le calcul de la capacité opérative correspond à des alignements entre des réservoirs et les bras de chargement des quais.	104
5.6	Résultats de l'exécution des méthodes "flot_max" et "determiner_chemins" pour le graphe "G".	105
5.7	Résultats de l'exécution des méthodes "nœuds_critiques", "arcs_critiques" et "arcs_critiques_fusion" pour le graphe "G".	105
5.8	Résultats de l'exécution des méthodes "chemin_min_int" pour le graphe "G".	106

5.9	Résultats de l'exécution des méthodes "chemin_pompe" pour le graphe "G".	106
5.10	Le calcul de la capacité opérative correspond à des alignements entre les oléoducs d'entrée et des réservoirs.	107
5.11	Résultats de l'exécution des méthodes "flot_max", "determiner_chemins", "nœuds_critiques", "arcs_critiques", "arcs_critiques_fusion", "arcs_critiques_fusion" et "chemin_pompe" pour le graphe "P".	108
6.1	Circuit tally permettant le comptage du nombre des bits à "1" dans une séquence de bits.	114

Liste des tableaux

3.1	Valeurs des coûts estimés vers chaque nœud du graphe à chaque itération de l'algorithme de Dijkstra	42
4.1	Relations d'équivalence entre des réservoirs d'un réseau de canalisation	57
4.2	Ensemble d'indicateurs de capacité opérative (C.O.).	71
4.3	Ensemble d'indicateurs de capacité opérative (C.O.) résultants d'un alignement.	71
4.4	Ensemble d'indicateurs de capacité opérative (C.O.) résultants de deux défaillances.	71
5.1	Saisie des étiquettes d'un réseau de canalisation.	100

Liste des Algorithmes

1	Initialiser(G, s)	40
2	Actualiser(u, v, w)	40
3	Dijkstra(G, w, s)	40
4	Dijkstra2(G, w, s)	41
5	Ford-Fulkerson(G, s, p)	47
6	Graphe_de_flot(G, A, B)	62
7	Flot_ex(G, n, f)	80
8	Recalcul_flot_max(G, s, p, f)	81
9	Actualiser(u, v, G_u, f_u, s, p)	81
10	Initialiser(G, s, p, a)	82
11	Chemin_min_int(G, s, p, a, b)	82
12	Ajout_Pompe(G, s)	86

Introduction

Résumé : Dans ce chapitre nous introduisons de manière générale le contexte dans lequel se sont déroulés les travaux de cette thèse ainsi que la problématique abordée. En particulier nous présentons l'entreprise PDVSA (Pétroles du Venezuela), à laquelle appartiennent les systèmes étudiés. Nous décrivons le déroulement des périodes industrielles et les premières approches abordées au cours de nos travaux. Enfin, la contribution principale de cette thèse est introduite en explicitant son objectif spécifique et les difficultés associées.

Sommaire

1.1	Contexte	1
1.2	Premières approches	3
1.2.1	Étude des coups de bélier	4
1.2.2	Système Expert	4
1.2.3	Algorithme de recherche exhaustive des alignements possibles	5
1.3	Contribution : Indicateurs de Capacité Opérative	5
1.4	Organisation du manuscrit	6

1.1 Contexte

L'exportation de pétrole est l'activité économique la plus importante du Venezuela. Le gouvernement vénézuélien estime que la certification des réserves de pétrole de l'Orénoque (fleuve) réalisée actuellement, va faire passer le Venezuela au premier rang mondial, devant l'Arabie Saoudite, avec plus de 300 milliards de barils. Actuellement le Venezuela produit plus de trois millions de barils de pétrole par jour, dont un tiers est traité dans des raffineries nationales, le reste étant exporté.

L'exportation de pétrole vers les Etats Unis, l'Europe et plus récemment vers le Brésil, l'Argentine, la Chine et les Caraïbes, est réalisée par des navires tankers. Les puits de pétrole se trouvant à l'intérieur du pays, loin de la côte, il faut transporter le liquide vers les ports au bord de la mer des Caraïbes, à travers des pipelines. Parmi ces ports se trouve la ville de Puerto La Cruz, dans l'Est du pays. Dans cette ville il existe une raffinerie ayant une capacité de traitement de 200.000 barils par jour, ainsi que le Terminal Maritime de Guaraguao, qui gère les transferts de pétrole entre les pipelines, les tankers et la raffinerie. Il peut délivrer jusqu'à un million de barils par jour. Le système de transfert de pétrole du Terminal Maritime de Guaraguao est

l'objet d'étude de cette thèse. Il peut être vu comme un réseau de canalisations, avec plusieurs réservoirs de stockage, des vannes, des pompes et des bras de chargement situés sur des quais. Ces éléments sont surveillés et commandés à distance grâce à un système de supervision et contrôle SCADA (Supervisory Control and Data Acquisition).

Cette thèse est une coopération entre PDVSA, entreprise de l'état vénézuélien chargée de l'exploration, production, raffinement et distribution du pétrole et ses dérivés, et l'INSA de Lyon. Le projet est financé grâce au programme de bourses Fundayacucho - Ambassade de France. La Fondation Gran Mariscal de Ayacucho (Fundayacucho) est un organisme de l'état vénézuélien chargé du financement des études universitaires. Le projet inclut plusieurs stages dans les installations de PDVSA ainsi que des séjours à l'INSA de Lyon.

Les périodes industrielles dans les installations de PDVSA au Venezuela ont permis, dans un premier temps, de prendre contact avec la problématique du Terminal Maritime de Guaraguao et de délimiter l'objet d'étude de la thèse, puis d'obtenir des informations pertinentes sur le fonctionnement de ce système. Une période au District Socialiste Technologique (DST) de PDVSA (centre de développement de logiciels) a permis d'identifier la potentielle utilité d'intégrer le logiciel d'assistance au logiciel SCADA développé par le DST, profitant de l'architecture modulaire de ce dernier. Une visite au Centre de Stockage de Punta de Palmas de PDVSA, a été réalisée. Il s'agit d'un système ayant une problématique similaire au Terminal Maritime de Guaraguao, dans lequel le logiciel SCADA développé au DST a été installé.

L'objectif général du projet est l'utilisation de formalismes et d'outils mathématiques relevant de l'automatisation industrielle visant à optimiser les opérations de transfert dans le réseau de canalisations de Guaraguao. Le système a un fonctionnement qui peut être décrit avec une logique d'événements, car les temps de changement d'état des divers éléments (ouverture/fermeture des vannes, démarrage/arrêt des pompes) sont d'un ordre de grandeur nettement inférieur au temps de séjour des éléments (vannes, pompes) dans un état donné. Donc notre étude relève du domaine de l'automatique des systèmes à événements discrets. Les investigations scientifiques reposent sur des approches basées sur la théorie des graphes et sur des algorithmes du domaine de la recherche opérationnelle.

Pétroles du Venezuela (PDVSA)

Les systèmes de transfert de pétrole étudiés dans cette thèse appartiennent à l'entreprise PDVSA et se trouvent sur le territoire vénézuélien. Nous allons présenter de manière succincte cette entreprise, ce qui permettra de comprendre l'importance et la pertinence des problématiques abordées dans cette thèse ainsi que quelques spécifications concernant la mise en œuvre des propositions fournies.

Pétroles du Venezuela S.A. est la société de l'État de la République Bolivarienne du Venezuela chargée de l'exploration, production, traitement, transport et commercialisation des hydrocarbures. D'après la Constitution du Venezuela, la totalité

des actions de PDVSA appartiennent à l'État vénézuélien. L'entreprise dépend du Ministère de l'Énergie et du Pétrole.

PDVSA a été fondée en 1975 par la "Loi Organique réservant à l'État l'industrie et le commerce des hydrocarbures" et à travers un processus de nationalisation des activités pétrolières gérées jusqu'à cette année-là par des compagnies étrangères, parmi lesquelles les plus importantes étaient Shell, Exxon, Mobil et Gulf Oil. Les filières de ces compagnies au Venezuela et la Corporation Vénézuélienne du Pétrole (CVP, institution de l'État créée en 1960) sont devenues des filières de PDVSA. En 1977, une première réorganisation de l'entreprise réduit le nombre de filières de 14 à 7, avec l'absorption des plus petites. La même année une deuxième réorganisation réduit le nombre de filières à 5, puis en 1978 à 4, et en 1986 à 3. En 1997 une nouvelle restructuration élimine ces filières et crée des entreprises fonctionnelles d'affaires qui vont intégrer le groupe : PDVSA Pétrole et Gaz, PDVSA Exploration et Production, PDVSA Manufacture et Commercialisation, et PDVSA Services. En 2001, PDVSA Pétrole et Gaz est divisée en PDVSA Pétrole et PDVSA Gaz. Ces origines et les politiques successives de l'entreprise expliquent la diversité des technologies utilisées, des logiciels et des fournisseurs. Cet aspect sera important à prendre en compte lors de la conception d'un logiciel pour mettre en œuvre nos contributions.

En décembre 2002 un conflit politique entre des employés de PDVSA et le gouvernement vénézuélien a provoqué l'arrêt de la plupart des activités de la compagnie pendant deux mois, ainsi que des actions de sabotage conduisant à d'énormes pertes et se répercutant en une forte diminution de l'activité économique du Venezuela. Durant ces événements, le contrôle des systèmes informatiques par des compagnies étrangères a joué un rôle très important, rendant difficile la gestion des activités. Cette dépendance technologique extérieure a été en partie la cause de la création du District Socialiste Technologique (DST) en 2006, ayant comme but le développement de logiciels et de solutions technologiques pour les besoins spécifiques de PDVSA.

Pendant cette crise de 2002-2003 les activités de PDVSA ont été effectuées dans des conditions très difficiles, avec un personnel très réduit, et souvent en réalisant des opérations hors du commun. Dans le cas particulier des systèmes de transfert de pétrole, des installations conçues pour l'exportation de pétrole ont été employées pour l'importation. Ces événements ont confirmé le besoin d'avoir des logiciels d'assistance aux opérateurs lors de situations d'urgence ou anormales.

1.2 Premières approches

Les travaux de thèse ont commencé par délimiter l'objet d'étude et par établir une problématique appartenant au domaine de l'automatique et applicable aux activités pétrolières. Plusieurs approches ont été étudiées avant d'aboutir à des solutions applicables. Pourtant il est convenable de présenter de manière succincte les approches n'ayant pas été retenues.

1.2.1 Étude des coups de bélier

Dans le contexte des réseaux de canalisation, un coup de bélier est un phénomène de choc produit par une onde de pression provoquée par le changement brusque de pression lors de la fermeture/ouverture d'une vanne ou le démarrage/arrêt d'une pompe. Les coups de bélier se propagent à haute vitesse à travers les canalisations et peuvent produire des fissures ou des ruptures dans les canalisations et les dispositifs. L'étude théorique du phénomène et des simulations utilisant le logiciel AMESim ont eu comme objectif la détermination d'une séquence optimale d'événements minimisant les effets des coups de bélier. Néanmoins nous avons trouvé que les séquencements permettant d'éviter des coups de bélier dangereux sont assez simples et ne constituent pas un apport du point de vue scientifique. Les développements futurs devront aborder ce problème au niveau des dispositifs utilisés ou au niveau de la conception du réseau plutôt qu'au niveau du contrôle et de la supervision.

1.2.2 Système Expert

Le développement d'un logiciel d'assistance a conduit à l'étude des systèmes experts, appartenant au domaine de l'intelligence artificielle. Un système expert est un logiciel permettant de donner des solutions à un problème dans lequel normalement l'expertise d'un humain est requise. Ce logiciel est habituellement constitué de trois niveaux [Badiru & Cheung, 2002] :

- Une base de connaissance : composée des règles de résolution des problèmes (base de règles) et des données intrinsèques du système (base de données). Nous avons envisagé principalement le développement de la base de connaissance du système expert. Les données intrinsèques du système correspondent à la constitution exacte du réseau de canalisation et au système de contrôle existant. La base de règles est une information plus générale, qui peut être appliquée à d'autres réseaux de transfert de pétrole, et même d'autres types de réseaux de transport.
- Une mémoire de travail : correspond aux données d'un problème spécifique à résoudre. Dans le cas du réseau de canalisation cette mémoire doit être alimentée en temps réel avec les données provenant du SCADA (état des vannes, de pompes, niveaux des réservoirs) et des utilisateurs (opérations envisagées, spécifications).
- Un moteur d'inférence : il s'agit d'un mécanisme logique permettant de trouver les solutions aux problèmes, en appliquant les connaissances de la base de connaissance aux cas particuliers enregistrés dans la mémoire de travail. Nous avons travaillé avec le moteur d'inférence du logiciel CLIPS (C Language Integrated Production System). Il s'agit d'un logiciel très répandu, gratuit et avec la capacité d'être intégré dans d'autres langages comme C, Ada et Visual Basic.

Cette approche est intéressante et constitue un possible projet pour à considérer pour le futur. Néanmoins, le manque d'un historique fiable des opérations réalisées,

des décisions prises et de leurs conséquences (qui aurait été très utile pour le développement de la base de règle du système expert) nous a amené à considérer d'autres approches.

1.2.3 Algorithme de recherche exhaustive des alignements possibles

Nous avons utilisé le langage de programmation et le moteur d'inférence de CLIPS pour le développement d'un algorithme de recherche exhaustive des alignements possibles entre deux éléments du système, présenté en 2008 dans le Congrès Latino-américain de Contrôle Automatique [Rojas et al., 2008]. Cet algorithme, étant exhaustif, a un temps de calcul qui augmente de manière exponentielle avec la taille du réseau. Son application est limitée aux réseaux très simples. Pourtant il utilise une heuristique intéressante qui permet l'application d'une méthode de calcul en parallèle pouvant réduire le temps d'exécution.

1.3 Contribution : Indicateurs de Capacité Opérative

Notre contribution est basée sur l'utilisation d'algorithmes de graphes. Cette approche utilise la notion de capacité opérative, liée à la quantité d'alignements simultanés possibles dans le réseau de canalisation. Le choix de cette approche a permis d'établir l'objectif spécifique de cette thèse : la définition d'indicateurs permettant de quantifier la capacité opérative du système, et le développement d'outils d'assistance aux opérateurs des réseaux de canalisation dans la prise de décision.

Les contributions de cette thèse reposent sur une modélisation sous forme de graphes, et sur des algorithmes appartenant au domaine de la recherche opérationnelle. Dans un premier temps, la difficulté principale de notre travail a été de traduire les demandes et les connaissances des opérateurs des systèmes, sous forme de contraintes et modèles formels. Une fois posé le problème sous forme mathématique, des algorithmes ont été développés pour trouver une solution à ce problème. Des indicateurs permettant de quantifier l'impact des opérations d'alignement (ou des défaillances) sur la capacité opérative du système, sont proposés. Un algorithme de calcul de ces indicateurs est présenté, ainsi que des algorithmes de recherche de chemins, de détermination d'éléments critiques et de recherche d'alignements utilisant des pompes. Ces algorithmes sont basés sur des algorithmes classiques de la recherche opérationnelle s'adressant au problème du plus court chemin, du flot maximum, et du nombre maximum de chemins disjoints. Cependant, ils utilisent des méthodes innovantes, comme l'ajout de contraintes considérant l'existence de sous-types d'alignements, le calcul dynamique des coûts des chemins à partir de leur impact sur la capacité opérative et la recherche de chemins via un point intermédiaire obligatoire.

Les algorithmes développés ont été mis en œuvre utilisant le langage Python et ont été testés en utilisant les données réelles des réseaux étudiés. Les temps de calculs, ainsi que les résultats trouvés sont satisfaisants. Plusieurs extensions de ces

travaux sont envisageables, en particulier l'incorporation de critères de choix basés sur la fiabilité dynamique des alignements, la minimisation du nombre d'éléments préalablement sollicités et son utilisation dans d'autres domaines comme la micro-électronique.

Les principales contributions de cette thèse ont été présentées à la European Control Conference 2009 [Rojas et al., 2009].

1.4 Organisation du manuscrit

Ce chapitre a permis de présenter le contexte général de déroulement de cette thèse et d'introduire des problématiques et des concepts qui sont abordés plus en détail dans les chapitres suivants.

Le chapitre 2 présente la problématique et l'état de l'art de cette thèse. Les systèmes de transport de pétrole sont présentés dans un contexte général, puis dans le cas spécifique du Venezuela. Les réseaux de canalisation et spécifiquement le réseau du Terminal Maritime de Guaraguao, ainsi que celui du Centre de Stockage de Punta de Palmas, sont décrits. La problématique des alignements est abordée plus en détail. Les systèmes SCADA, dans le cas particulier des réseaux de transfert de pétrole, sont présentés. L'état de l'art appartient à deux domaines : le domaine de la modélisation des réseaux de canalisation et le domaine des algorithmes de graphes. Des logiciels modélisant les réseaux de canalisation sont décrits. Dans le cas des algorithmes nous avons abordé l'étude du problème du plus court chemin, le problème du flot maximum, le problème des chemins disjoints, et les indices de centralité dans des graphes.

Les concepts, les modèles, et les algorithmes de référence du chapitre 3 servent à comprendre les contributions de cette thèse. Les concepts de base sur les graphes, et leurs représentations matricielles et sous forme de liste, sont introduits. La définition mathématique des problèmes de la théorie des graphes, ainsi que la présentation d'algorithmes classiques donnant des solutions à ces problèmes sont abordées.

Le chapitre 4 présente les contributions principales de cette thèse. Ce chapitre définit de manière formelle la notion de capacité opérative, et présente un algorithme de calcul des indicateurs de la capacité opérative d'un réseau de canalisation. Plusieurs algorithmes, représentant des développements complémentaires basés sur la notion de capacité opérative, sont proposés. Ces algorithmes sont des outils permettant une analyse de la criticité des éléments du réseau, la recherche d'alignements maximisant la capacité opérative, et la recherche d'alignements passant par une pompe.

Le chapitre 5 montre la mise en œuvre des algorithmes proposés dans le chapitre 4. Ces algorithmes sont codés avec le langage de programmation Python et testés en utilisant les données réelles des réseaux de canalisation étudiés. Une architecture modulaire du logiciel d'assistance aux opérateurs est proposée.

Finalement, le dernier chapitre permet de tirer un bilan du manuscrit, des possibles extensions des contributions de cette thèse, et des perspectives du projet.

Problématique et État de l'Art

Résumé : Dans ce chapitre nous introduisons le fonctionnement des systèmes de transport de pétrole, premièrement dans un contexte international, et deuxièmement dans le contexte vénézuélien. Nous allons détailler la problématique des alignements (ou mise en relation) d'éléments constitutifs des systèmes de transport de pétrole, spécifiquement pour deux systèmes situés au Venezuela : celui du terminal maritime de Guaraguao à l'est du pays, et celui du centre de stockage de Punta de Palmas à l'ouest du pays. Nous allons introduire les systèmes SCADA utilisés pour le contrôle et la supervision des systèmes de transport étudiés. Finalement nous présentons la bibliographie appartenant au deux domaines d'étude de cette thèse : la modélisation des réseaux de canalisation, et le développement d'algorithmes de graphes.

Sommaire

2.1	Transport de pétrole	7
2.1.1	Propriétés du pétrole	8
2.1.2	Réseaux de canalisation	9
2.2	Transport de pétrole au Venezuela	11
2.2.1	Terminal Maritime de Guaraguao	11
2.2.2	Centre de Stockage de Punta de Palmas	13
2.3	Problématique des alignements	15
2.3.1	Définition d'un alignement	15
2.3.2	Contraintes sur les alignements	16
2.3.3	Critères de choix des alignements	18
2.4	SCADA	19
2.4.1	SCADA des systèmes de transfert de pétrole	20
2.4.2	Logiciel SCADA	20
2.5	État de l'art	22
2.5.1	Modélisation des réseaux de canalisation	23
2.5.2	Algorithmes de graphes	25
2.6	Conclusion	27

2.1 Transport de pétrole

Le pétrole constitue l'un des principaux produits de l'économie mondiale, non seulement du point de vue du poids économique représenté par les activités pétrolières, mais surtout du fait de la dépendance de l'économie actuelle aux produits

dérivés du pétrole. Le pétrole se trouve de façon naturelle dans des zones spécifiques de la planète, la plupart concentrées dans quelques régions et en particulier au Moyen-Orient et au Venezuela. Pour être utilisé, le pétrole doit être traité et raffiné dans des installations qui sont très souvent placées à une distance considérable des champs d'extraction, cela pour satisfaire à des contraintes spécifiques de l'activité économique. La localisation d'une raffinerie doit tenir compte de facteurs très divers comme l'accès à la main d'oeuvre, l'accès à une source d'eau, la connexion au réseau de distribution, la sécurisation de la zone, l'impact environnemental, le contrôle étatique, etc. Actuellement le marché mondial du pétrole implique des opérations de transport à échelle planétaire.

Les premiers moyens de transport de pétrole, au dix-neuvième siècle, étaient les trains, pour les cheminements terrestres, et les navires, pour les cheminements fluviaux et maritimes. Le pétrole était transporté dans des barils de 159 litres, conditionnement retenu depuis lors comme unité de volume pour le pétrole. Le développement rapide de l'industrie pétrolière va avoir comme résultat l'apparition de technologies conçues spécialement pour le transport de pétrole. Des navires avec des réservoirs de pétrole intégrés, ou tankers, seront construits, ainsi que des canalisations pour le transport de pétrole. Les oléoducs, des canalisations à grand diamètre pour des longues distances seront la solution technologique utilisée aux États Unis pendant la Seconde Guerre Mondiale pour faire face aux attaques allemandes ciblant ses navires de transport [Trench, 2001]. Sur quelques décennies les tankers et les oléoducs vont devenir les principaux moyens de transport de pétrole à cause de leurs très faibles coûts et l'utilisation d'autres moyens de transport, comme les camions et les trains, sera fortement réduite. L'exploitation des tankers et des oléoducs va impulser d'une part le développement de dispositifs permettant de pomper (pompes) et d'acheminer (vannes) le pétrole, et d'autre part la construction d'installations spécialisées pour connecter les différents systèmes de transport entre eux, et avec les raffineries, les zones d'extraction, et les centres de stockage de pétrole.

2.1.1 Propriétés du pétrole

Le pétrole est un mélange d'hydrocarbures avec des propriétés physico-chimiques qui peuvent varier d'un gisement à l'autre. Le pétrole étant souvent assez homogène dans la même zone d'extraction, le nom de ces zones sert à classifier les types de pétrole et va souvent identifier ces types de pétrole. Les propriétés du pétrole vont être importantes du point de vue du transport car elles vont limiter les possibles destinations. Chaque centre de traitement ou de raffinage de pétrole est conçu pour traiter des types de pétrole avec des propriétés spécifiques, souvent avec une marge de tolérance assez grande. Parmi les propriétés les plus importantes nous pouvons citer :

La densité API : c'est une échelle permettant d'exprimer la densité du pétrole brut, en degrés API ($^{\circ}$ API), calculée à partir de la densité, par la formule suivante :

$$\text{Densité API} = (141.5 / (\text{densité à } 60 \text{ } ^{\circ}\text{F})) - 131.5$$

Plus un brut est léger, plus sa densité est faible, plus sa densité API est élevée. La plupart des bruts extraits ont des densités API comprises entre 20° (très lourd) et 60° (très léger). Cette formule fut créée par l'American Petroleum Institute (API) et le National Institute of Standards and Technology (NIST). Le prix du pétrole augmente avec les degrés API.

La composition chimique : le pétrole étant un mélange d'hydrocarbures, la composition chimique du pétrole va dépendre de la proportion des différents types d'hydrocarbures présents. Ainsi le pétrole peut être de type paraffinique, naphténique ou aromatique.

Le pourcentage d'eau : Les hydrocarbures sortent des puits souvent mélangés avec de l'eau. Fréquemment le pétrole est traité pour retirer la plupart de l'eau qu'il contient.

La teneur en soufre : très souvent le pétrole se trouve naturellement mélangé avec du soufre. Le soufre doit être retiré pendant le raffinage, alors la tâche de raffinage doit tenir compte de la teneur en soufre du pétrole.

La salinité : le pétrole se trouve naturellement mélangé avec du sel. Le sel doit être retiré pendant le raffinage, alors la conception des raffineries doit tenir compte de la salinité du pétrole.

La teneur en minéraux : à part le soufre et le sel, d'autres minéraux sont présents dans le pétrole. Raffiner coûte d'autant plus cher que la teneur en minéraux est élevée car il est nécessaire de les retirer.

La viscosité : chaque fluide oppose une résistance interne au flux. La viscosité du pétrole va affecter la facilité avec laquelle il peut être transporté, et donc les coûts de transport.

En général, plus les caractéristiques d'un type de pétrole brut entraînent un coût élevé de transport et de raffinage, moindre sera le prix de ce type de pétrole, et vice-versa. En conséquence de ces propriétés particulières, des installations spéciales de traitement de pétrole doivent souvent être envisagées dans la conception du réseau de transport ; c'est le cas des centres de traitement pour séparer l'eau du pétrole, et des centres de traitement pour transformer le pétrole extra-lourd afin de pouvoir le transporter en utilisant des oléoducs.

Chaque type de pétrole a un prix et des destinations spécifiques. Les raffineries étant conçues pour traiter certains types de pétroles, la conception du réseau de transport doit tenir compte aussi des types de pétrole disponibles, et des centres de raffinage compatibles.

2.1.2 Réseaux de canalisation

D'un point de vue transport, le pétrole a comme origine les champs d'extraction et comme destination finale les raffineries dans lesquelles le pétrole est transformé en d'autres produits. Entre ces deux sites, il existe en général plusieurs centres intermédiaires qui servent à stocker le pétrole, le traiter avant le raffinage, ou à connecter les différents moyens de transport (oléoducs, tankers, camions, trains). Ces centres intermédiaires sont très souvent des réseaux de canalisation dans lesquels

il est possible d'établir de multiples connexions. Les types d'éléments à connecter vont dépendre du centre concerné. La liste suivante présente les éléments les plus communs qui font partie de ces réseaux ou qui y sont directement connectés :

Vanne : les vannes permettent d'acheminer le pétrole en séparant des sections du réseau pour établir des connections entre des éléments tels que des réservoirs, pompes, bras de chargement, et oléoducs d'entrée/sortie. Les vannes permettent aussi d'isoler des éléments ou des sections du réseau pour réaliser des opérations de maintenance, réparation ou remplacement.

Pompe : les pompes permettent d'établir un flux de pétrole lorsque la différence de pression entre les éléments fournisseurs et les éléments receveurs est négative ou nulle. Dans le cas des oléoducs qui connectent plusieurs systèmes, les pompes sont très souvent placées dans le système qui délivre le pétrole.

Oléoduc d'entrée/sortie : les oléoducs sont de larges canalisations (environ un mètre de diamètre) qui connectent le réseau à d'autres systèmes sur plusieurs centaines de kilomètres, permettant le transport de pétrole entre ces systèmes. Ces oléoducs souvent vont servir uniquement soit à fournir soit à recevoir du pétrole au réseau. Dans le cas des oléoducs qui fournissent du pétrole, appelés oléoducs d'entrée, le pétrole arrive avec une haute pression grâce à des pompes qui sont placées hors du réseau considéré. Dans le cas des oléoducs qui reçoivent du pétrole, appelés oléoducs de sortie, le pétrole doit être propulsé par des pompes appartenant au réseau considéré.

Réservoir de stockage : des réservoirs de structure cylindrique sont très souvent utilisés pour stocker le pétrole. Les capacités de ces réservoirs sont de l'ordre de dizaines de milliers de barils de pétrole. Souvent ils sont équipés d'un système de mesure de niveau qui permet de déterminer la quantité de pétrole contenue dans le réservoir. L'utilisation de réservoirs peut avoir différents buts :

- Servir d'intermédiaires entre un flux continu de pétrole arrivant des oléoducs, et la livraison du pétrole par "lots" à des tankers.
- Faire face aux variations de la consommation.
- Servir de réserves stratégiques de pétrole.
- Mesurer le volume de pétrole transporté.
- Traiter le pétrole avant son raffinage. Par exemple, les réservoirs peuvent être utilisés pour retirer l'eau et la terre par gravité.
- Mélanger des types différents de pétrole pour produire un autre type de pétrole.

Bras de chargement : lorsque le réseau fait partie d'un terminal maritime ou fluvial de pétrole, le chargement ou le déchargement des navires est réalisé très souvent par des bras de chargement placés sur les quais du port. Ces bras de chargement sont des bras mécaniques reliés au réseau et qui se connectent aux réservoirs internes d'un navire arrivé à quai.

Mélangeur : comme nous l'avons expliqué auparavant, deux types différents de pétrole peuvent être mélangés avec des proportions spécifiques pour produire un autre type de pétrole. Ce processus de mélange peut être réalisé dans un réservoir, mais souvent un dispositif spécialement conçu pour mélanger du pétrole est utilisé.

Ce dispositif, appelé mélangeur, est une jonction de trois canalisations, deux canalisations d'entrée correspondant aux pétroles de base, et une canalisation de sortie correspondant au mélange. Un système de contrôle en boucle fermée constitué d'un régulateur automatique, de vannes et de capteurs de flux permet de produire un mélange avec des proportions beaucoup plus proches des spécifications qu'avec des méthodes de mélange utilisant des réservoirs. De plus le pétrole final peut être délivré directement sans attendre l'homogénéisation du mélange (nécessaire pour le mélange utilisant des réservoirs).

Multiple : un multiple est une portion du réseau vers laquelle convergent de nombreuses canalisations. Grâce à l'ouverture et à la fermeture de vannes, différentes connections peuvent être établies entre les canalisations. Les multiples permettent de regrouper de nombreuses vannes dans un endroit localisé, facilitant l'ouverture et la fermeture manuelle des vannes, ainsi que des opérations de maintenance, réparation et remplacement. Ces multiples offrent de la redondance.

Par la suite, ces termes seront utilisés pour décrire les différents systèmes réels étudiés ainsi que les exemples pédagogiques imaginés afin de faciliter la compréhension des contributions présentées.

2.2 Transport de pétrole au Venezuela

Le Venezuela produit actuellement plus de trois millions de barils de pétrole par jour. Un tiers de ce pétrole est raffiné dans le pays, et le reste est exporté par voie maritime. Le réseau de transport de pétrole national est constitué principalement par des puits de pétrole, des raffineries et des terminaux maritimes. Des centres de stockage et de traitement de pétrole avant raffinage complètent les types de systèmes constituant ce réseau national. Parmi ces systèmes, un terminal maritime de pétrole et un centre de stockage de pétrole ont été étudiés dans le cadre de cette thèse. La figure 2.1 présente le réseau de transport de pétrole du Venezuela, et les deux systèmes étudiés (encerclés en rouge).

Nous allons présenter plus en détail le réseau de transfert de pétrole propre à chacun de ces deux systèmes. Le Terminal Maritime de Guaraguao a été l'objet principal d'étude, tandis que le Centre de Stockage de Punta de Palmas a été choisi pour tester les développements du projet. Le reste des installations pétrolières de PDVSA comportant un réseau de transfert de pétrole a un fonctionnement similaire à l'un de ces deux systèmes étudiés. Ces exemples sont donc représentatifs des réseaux de canalisation de PDVSA et permettent de valider les contributions de ces travaux.

2.2.1 Terminal Maritime de Guaraguao

Le Terminal Maritime de Guaraguao se trouve dans la ville de Puerto La Cruz (500.000 habitants), dans l'est du Venezuela. Son système de transfert de pétrole a la capacité de gérer jusqu'à un million de barils par jour. Il reçoit du pétrole des oléoducs provenant des gisements localisés au sud. Le pétrole est stocké dans des

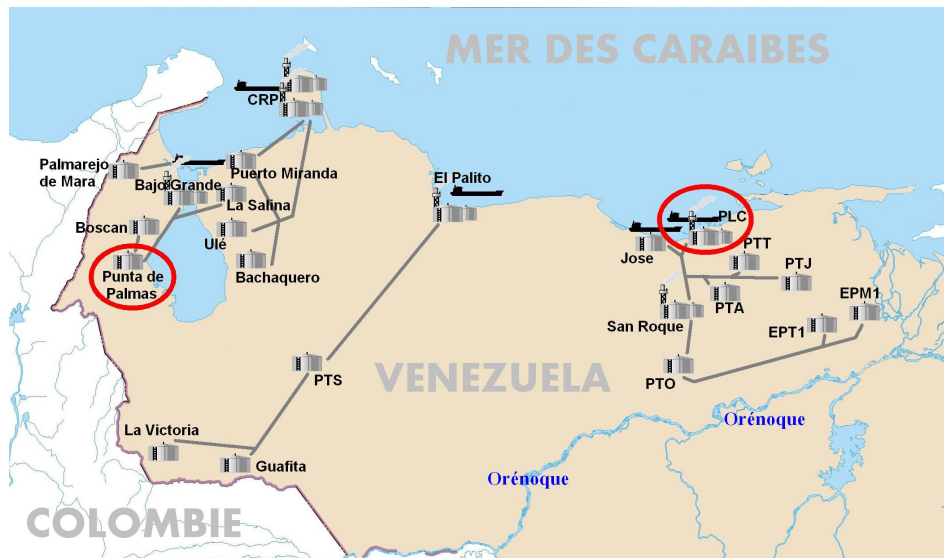


FIGURE 2.1 – Schéma simplifié du réseau de transport de pétrole du Venezuela. Le centre de stockage de Punta de Palmas et le Terminal Maritime de Guaraguao (PLC) sont encadrés en rouge.

réservoirs, puis, selon les besoins, il est transporté vers des tankers pour être exporté, ou bien il est destiné à la Raffinerie de Puerto La Cruz, laquelle a une capacité de traitement de 200.000 barils de pétrole par jour. Dans ce système la plupart des alignements connectent seulement deux éléments du réseau.

La figure 2.2 montre la structure générale du réseau de transport du Terminal Maritime de Guaraguao.

Le fonctionnement détaillé du système étant confidentiel pour l'entreprise, nous allons donner une vue plutôt générale du système concerné, d'après les manuels d'opérations et les descriptions données par le personnel de l'entreprise. Le réseau comporte les éléments suivants :

Oléoducs d'entrée : appelées «troncales», ces 4 lignes envoient constamment du pétrole grâce à la pression générée par des stations de pompes localisées hors du Terminal Maritime. Chaque ligne envoie un seul type de pétrole (“Merey”, “Mesa”, “Santa Barbara”, “Anaco Wax”), de composition (salinité, degré API, viscosité) spécifique.

Réservoirs : le pétrole est stocké dans plusieurs dizaines de réservoirs, représentant une capacité évaluée entre 100.000 et 300.000 barils de pétrole chacun, en attente de l'arrivée d'un tanker ou d'une demande de la raffinerie. Plusieurs de ces réservoirs se trouvent à 40 mètres d'altitude, ce qui permet d'embarquer le pétrole par gravité sans utiliser de pompes. Le niveau de remplissage de chaque réservoir est mesuré par des radars. Occasionnellement un réservoir peut être utilisé pour mélanger du pétrole.

Mélangeurs : en mélangeant deux types différents de pétrole, un cinquième

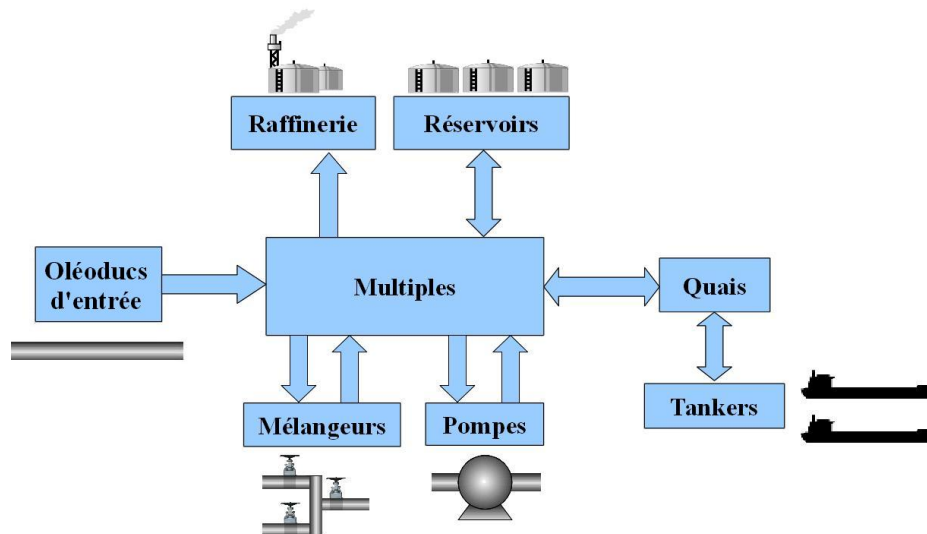


FIGURE 2.2 – Structure générale du système de transport de pétrole du Terminal Maritime de Guaraguao.

type, appelé “Leona” peut être produit dans les installations du Terminal Maritime. Cette transformation est réalisée pour satisfaire la demande des clients (les raffineries sont conçues pour travailler dans une fenêtre spécifique de degré API).

Bras de chargement : le transvasement du pétrole dans des tankers est réalisé par des bras robotisés qui s’accouplent aux tankers. Chaque tanker a une capacité semblable à celle d’un réservoir.

Multiples : le système possède plusieurs multiples interconnectés.

Pompes de transfert : l’arrivée du pétrole étant assurée par des stations de pompage qui se trouvent à l’intérieur du pays (à des dizaines de kilomètres), et le transvasement étant réalisé le plus souvent par gravité, les pompes sont utilisées soit pour effectuer des transferts entre les réservoirs, soit pour alimenter la raffinerie, soit, moins fréquemment, pour décharger des tankers.

Vannes : des centaines de vannes permettent de connecter, ou d’isoler, les différents éléments du système. Des vannes sont utilisées pour court-circuiter les pompes, réduisant la pression générée. Les vannes les plus utilisées sont motorisées et peuvent être pilotées à distance, depuis une salle de contrôle. Lors d’une situation d’urgence, des vannes manuelles peuvent être utilisées pour contourner des éléments défaillants (by-pass) ou pour isoler un élément lors d’une opération de maintenance.

2.2.2 Centre de Stockage de Punta de Palmas

Le centre de stockage Punta de Palma est localisé dans l’état du Zulia, dans l’ouest du Venezuela. Il reçoit du pétrole provenant des puits de pétrole du Lac de Maracaibo à travers trois oléoducs. Le pétrole est stocké dans des réservoirs, dans lesquels il est mis au repos jusqu’à ce que le pétrole et l’eau, avec laquelle il est

mélangé, soient séparés en deux couches. Puis, l'eau est retirée du pétrole grâce à des vannes. Une fois traité, le pétrole est expédié, en utilisant des pompes, vers des oléoducs de sortie connectés à des raffineries et des terminaux maritimes. Dans ce système la plupart des alignements connectent seulement deux éléments du réseau.

La figure 2.3 montre la structure générale de ce système :

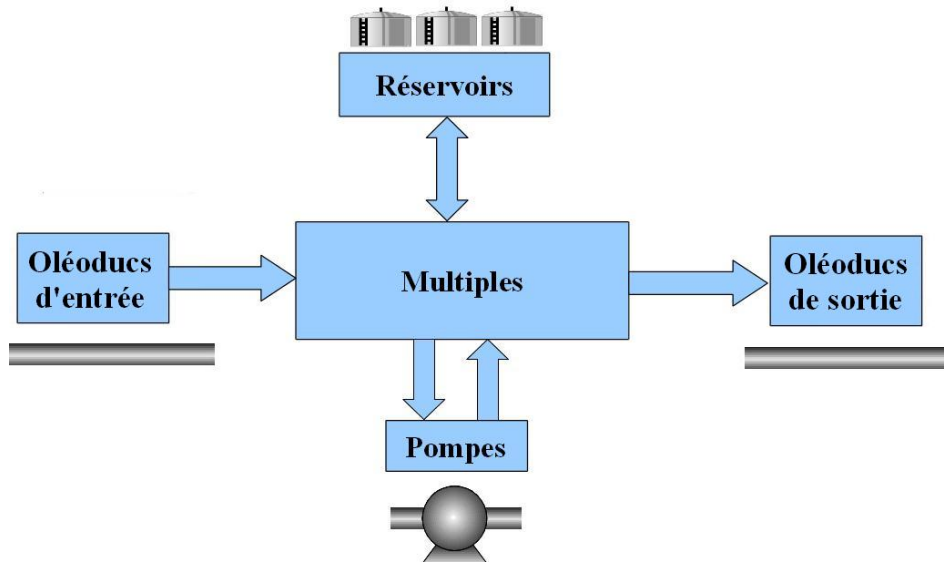


FIGURE 2.3 – Structure général du système de transport de pétrole du Centre de Stockage de Punta de Palmas.

Ce réseau est constitué des éléments suivants :

Oléoducs d'entrée : ces 4 lignes envoient constamment du pétrole grâce à la pression générée par des stations de pompes localisées hors du Centre de Stockage. Chaque ligne envoie un type de pétrole particulier ("Lagomar", "Lagomedio", "DZO Colon", "Boscan"), avec une composition (salinité, degré API, viscosité) spécifique.

Oléoducs de sortie : ces 2 lignes permettent d'envoyer du pétrole vers un Terminal Maritime (Puerto Miranda) et une raffinerie (Bajo Grande), situés à plusieurs dizaines de kilomètres. Les pompes du système doivent être utilisées pour établir ces flux de pétrole.

Réservoirs : le pétrole est stocké dans dix huit réservoirs, avec une capacité jugée entre 100.000 et 300.000 barils de pétrole chacun. Le niveau de chaque réservoir est mesuré par des radars. Chaque réservoir peut accomplir des fonctions de stockage, traitement (enlèvement d'eau) et mélange du pétrole.

Multiples : Le système possède un seul multiple.

Pompes : l'arrivée du pétrole étant assurée par des stations de pompes qui se trouvent à l'intérieur du pays, les pompes sont utilisées pour envoyer le pétrole hors du système à travers les oléoducs de sortie, ou pour faire des transferts entre les

réservoirs.

Vannes : des dizaines de vannes permettent, soit de connecter, soit d'isoler, les différents éléments du système. Les fonctions des vannes sont identiques à celles du système de transport de pétrole du Terminal Maritime de Guaragua.

2.3 Problématique des alignements

Plusieurs types d'opérations peuvent être réalisées dans les réseaux de canalisation considérés : alignements, opérations de maintenance, réparations, installation de nouveaux éléments, etc. Parmi ces opérations, les alignements sont les opérations permettant de transporter le pétrole entre les différents éléments du réseau.

2.3.1 Définition d'un alignement

Un alignement est défini comme un ensemble de canalisations connectant au moins deux éléments du réseau et permettant le transport de pétrole entre ces éléments. Ces canalisations doivent être isolées du reste du réseau par des vannes pour éviter le mélange de différents types de pétrole ou l'écoulement du pétrole vers d'autres éléments du réseau (ne faisant pas partie de l'alignement). Alors, l'alignement est établi lorsque les vannes connectant les canalisations faisant partie de l'alignement sont ouvertes et les vannes adjacentes à ces canalisations sont fermées. Donc, l'établissement de l'alignement nécessite l'ouverture et la fermeture de plusieurs vannes. Les éléments connectés par un alignement seront appelés éléments terminaux, ils peuvent être de deux types :

Élément fournisseur : c'est un élément qui fournit le pétrole aux éléments récepteurs. Il peut être considéré comme une source de flux. Exemples : oléoduc d'entrée, réservoir non vide, bras de chargement utilisé pour décharger des tankers.

Élément récepteur : c'est un élément qui reçoit le pétrole provenant des éléments fournisseurs. Il peut être considéré comme un puits de flux. Exemples : réservoir non plein, oléoduc de sortie, bras de chargement, raffinerie, tanker.

Normalement, les canalisations du réseau sont remplies de pétrole, lequel peut varier de type, ou être un mélange hors standards. Lorsqu'un alignement est établi, le pétrole provenant de l'élément fournisseur sera mélangé avec le pétrole contenu dans les canalisations. Cependant, la quantité de pétrole contenu dans les canalisations est négligeable par rapport à la quantité de pétrole qui sera transportée de l'élément fournisseur jusqu'à l'élément récepteur.

La figure 2.4 montre un alignement entre un réservoir (élément fournisseur) et un bras de chargement (élément récepteur).

Les canalisations marquées de gris appartiennent à un alignement constitué. Ces canalisations seront appelées route de l'alignement. Les vannes colorées en noir correspondent à des vannes qui doivent être fermées pour isoler l'alignement du reste du réseau. Les vannes non colorées qui se trouvent dans l'alignement (connectant deux canalisations en gris) doivent être ouvertes pour permettre le transport de pétrole entre les éléments. Une opération d'établissement d'un alignement correspondra

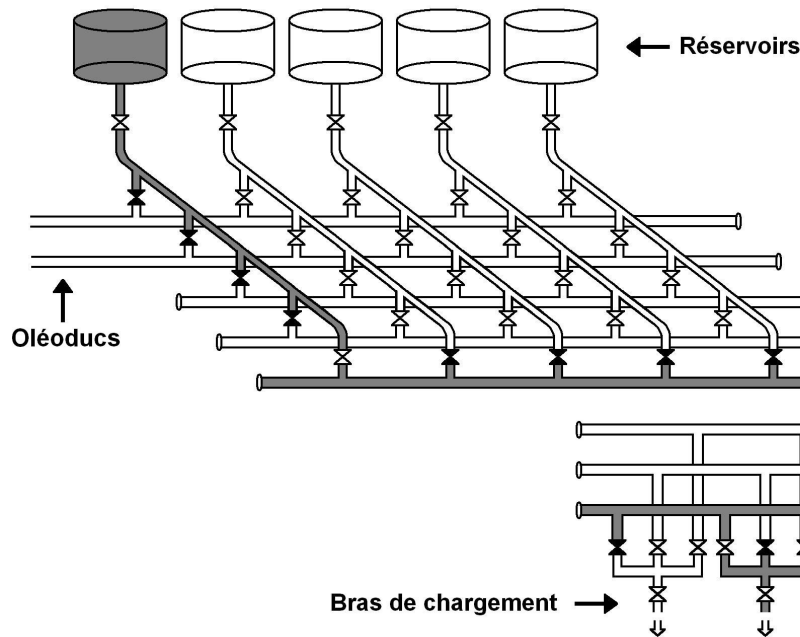


FIGURE 2.4 – Alignement (en gris) entre un réservoir et un bas de chargement.

alors à identifier, puis ouvrir ou fermer les vannes nécessaires pour transporter le pétrole.

2.3.2 Contraintes sur les alignements

Chaque réseau de canalisation doit satisfaire des demandes de pétrole de la part d'autres systèmes ou d'agents externes au réseau. Dans le cas du terminal maritime, les tankers sont souvent des clients. L'entreprise doit alors satisfaire, au risque de subir une pénalité, les conditions établies dans un contrat, dans lequel la composition du pétrole et les délais d'attente sont spécifiés. Dans les autres cas, les retards de livraison peuvent entraîner d'autres types de pertes pour l'entreprise (comme des pénalités dans une autre partie de la chaîne de transport), ainsi que des problèmes de logistique. Chaque demande va correspondre généralement aux spécifications suivantes :

- Récepteur
- Type de pétrole
- Quantité de pétrole
- Temps maximum d'attente
- Flux maximum du pétrole

Pour répondre à une demande de livraison de pétrole, l'alignement doit connecter des éléments permettant de satisfaire les spécifications de la demande. Les éléments fournisseurs, ainsi que les éléments récepteurs doivent pouvoir accomplir des tâches

particulières. Par exemple, considérons une demande de pétrole avec les spécifications suivantes :

- Récepteur : tanker de grande capacité
- Type de pétrole : Merey
- Quantité de pétrole : 250.000 barils
- Temps maximum d’attente : 10 heures
- Flux maximum du pétrole : 100.000 barils par heure

Un alignement doit être établi pour satisfaire cette demande. Ces spécifications vont probablement limiter les possibles éléments fournisseurs et récepteurs ainsi que la route de l’alignement. Dans l’exemple considéré, le récepteur étant un tanker de grande capacité, l’élément récepteur correspondra à un (ou plusieurs) bras de chargement placé(s) sur un quai ayant la capacité d’accueillir des tankers de la taille désirée. L’élément fournisseur doit pouvoir fournir du pétrole de type “Merey”, alors seulement des réservoirs contenant ce type de pétrole, ou des oléoducs d’entrée délivrant ce type de pétrole, seront utilisés. Si les éléments fournisseurs sont des oléoducs d’entrée, leur utilisation sera conditionnée par le flux de pétrole, lequel devra satisfaire les contraintes de temps maximum d’attente et de quantité de pétrole. Si les éléments fournisseurs sont des réservoirs, la quantité de pétrole totale contenue dans les réservoirs doit être supérieure ou égale à la quantité de pétrole demandée, et le flux de pétrole généré (soit par des pompes, soit par gravité, pour des réservoirs placés en hauteur) doit satisfaire les contraintes de temps d’attente maximum et de flux maximum du tanker.

Très souvent, divers éléments fournisseurs, récepteurs, ou diverses routes permettent de satisfaire une même demande. La problématique abordée est celle du choix des alignements permettant de satisfaire les demandes de manière optimale (selon des critères que nous définirons plus tard). Pour les deux sites étudiés (Guaraguao et Punta de Palmas) cette même problématique se présente lors des opérations de transport de pétrole. La proposition retenue est d’assister les opérateurs dans la prise de décision pour établir une route d’alignement. Cette problématique est commune à tous les réseaux de transport similaires ayant notamment les caractéristiques suivantes :

- Plusieurs types de pétrole : le système gère plusieurs types de pétrole et pour la plupart des opérations de transport considérées le mélange de différents types de pétrole doit être évité.
- Multiples : les multiples offrent de nombreuses combinaisons de connexion lors de l’acheminement du pétrole dans le réseau.
- Complexité du réseau : la complexité rend difficile la prise de décisions pour optimiser les opérations de transport. D’une part, l’expertise des opérateurs peut s’avérer insuffisante pour donner une réponse adéquate à toutes les situations possibles. D’autre part les algorithmes de force brute (trouvant des solutions après avoir calculé toutes les possibilités, au lieu d’appliquer une heuristique) peuvent s’avérer inutiles, car les temps de calcul les rendent inutilisables dans la pratique. Alors plus le réseau sera complexe, plus notre proposition sera utile.

Dans la pratique, la problématique du choix d'un alignement est simplifiée grâce à des caractéristiques particulières des réseaux de canalisation étudiés. En général, les systèmes sont conçus pour répondre aux demandes au travers d'alignements entre un seul élément fournisseur et un seul élément récepteur. L'établissement d'un alignement prend quelques minutes (plusieurs vannes pouvant être ouvertes/fermées en même temps), tandis que le transport de pétrole utilisant un alignement prend plusieurs heures. Les temps maximum d'attente des tankers vont être normalement respectés dans des conditions normales. Notre proposition cible en particulier des situations d'urgence, occasionnées par des défaillances, la congestion du système, des demandes extraordinaires, etc. La fréquence de ces situations est de quelques fois par an.

2.3.3 Critères de choix des alignements

Le choix d'un alignement parmi un ensemble de solutions possibles doit tenir compte de critères d'optimalité. Dans notre étude, les critères suivants ont été considérés :

Distance : la distance parcourue par le pétrole de l'élément fournisseur jusqu'à l'élément récepteur est un critère simple qui peut modifier la composition du pétrole, puisque cette distance augmente le volume de pétrole contenu initialement dans les canalisations (qui n'est pas forcément du même type que le pétrole délivré) et qui sera mélangé au pétrole délivré par l'élément fournisseur.

Pertes de charge : les pertes de charge représentent la résistance au flux du pétrole d'un alignement. Ces pertes de charge vont contraindre le flux de transport du pétrole, et augmenter le temps de transport du pétrole vers le récepteur.

Nombre de vannes concernées : le nombre de vannes faisant partie de l'alignement, ainsi que le nombre de vannes entourant l'alignement (et permettant de l'isoler) est une estimation du nombre d'éléments qu'il faudra commander pour établir un alignement. Chaque commande implique un risque de défaillance, ainsi qu'une dépense d'énergie. La minimisation de ces variables pourrait alors être recherchée.

Nombre de commutation de vannes : lorsque l'alignement est considéré en temps réel, et l'état des vannes (ouvertes/fermées) est connu, il est possible de tenir compte du nombre de vannes dont l'état doit être changé, c'est à dire, le nombre de vannes devant être ouvertes augmenté du nombre de vannes devant être fermées. Chaque commutation implique un risque de défaillance, ainsi qu'une dépense d'énergie. La minimisation de ces variables pourrait également être recherchée.

Interférences avec d'autres opérations : le choix d'un alignement va rendre plusieurs éléments du réseau indisponible pour d'autres opérations. Par exemple, une vanne faisant partie de la route d'un alignement doit être ouverte, alors elle ne peut pas être fermée pour isoler un autre alignement. Dans certains cas, cette indisponibilité de quelques éléments peut retarder la réponse à une demande de pétrole. Ce critère s'est avéré le plus critique pour les opérateurs.

Dans cette thèse le critère de minimisation du nombre d'interférences entre les

alignements a été retenu. Ceci nous amène à définir une notion de capacité opérative liée à la quantité d'alignements simultanés possibles dans le réseau de canalisation. Concernant les autres critères, soit ils sont négligeables pour être pris en compte lors du choix d'un alignement, soit il ne sont pas prioritaires au regard de la minimisation des interférences et pourront faire partie des extensions de cette thèse.

2.4 SCADA

Les systèmes de transfert étudiés sont contrôlés par des opérateurs via des systèmes de supervision appelés SCADA. SCADA est l'acronyme de l'anglais Supervisory Control and Data Acquisition (contrôle par supervision et acquisition de données). Un logiciel d'assistance aux opérateurs peut exploiter les données provenant des dispositifs du réseau et gérés par le système SCADA, et peut agir directement sur le système grâce à des actionneurs commandés à travers le SCADA. Le développement d'un logiciel d'assistance doit donc tenir compte d'une possible communication avec les systèmes SCADA des réseaux concernés.

Le terme SCADA sert à identifier des systèmes de supervision et de contrôle centralisés, permettant aux opérateurs de surveiller en temps réel des procédés industriels, au travers de capteurs, et d'agir sur ces procédés à travers d'actionneurs placés à distance (Stouffer et al., 2006). Ils sont implantés dans des systèmes tels que des réseaux de distribution d'eau, des oléoducs de pétrole et de gaz, des réseaux de distribution électrique, des réseaux ferroviaires. Le centre de contrôle du SCADA réalise une surveillance des sites à travers des réseaux de communication longue distance. Très souvent le SCADA est équipé avec un système d'alarmes et un système de traitement des données. Les dispositifs distribués dans le site permettent de contrôler des opérations locales comme l'ouverture et la fermeture des vannes, l'acheminement des données provenant des capteurs, et la surveillance des conditions environnementales locales pour évaluer les conditions d'alarme. Les systèmes SCADA sont constitués en général des éléments suivants :

RTUs (Remote Terminal Units) ou Terminaux distants : Ce sont des dispositifs connectés directement aux capteurs et aux actionneurs qui sont sur le terrain ou le site industriel. Très souvent, ce sont des contrôleurs programmables qui supervisent des dispositifs ou des processus locaux. Ils ont aussi la fonction de servir d'interface analogiques/numériques entre les capteurs, les actionneurs, et le réseau de communication.

MTU (Master Terminal Unit) ou Unité Centrale : c'est une unité logique placée dans le centre de contrôle du système, qui prélève périodiquement des données provenant des RTUs, qui envoie des commandes vers les RTU, et qui sert d'interface avec les opérateurs. Le MTU est souvent composé de plusieurs ordinateurs connectés par un réseau local.

Réseau de communication : les RTU sont connectés au MTU par à un réseau de communication qui peut avoir différentes architectures. Les protocoles de communication du logiciel SCADA utilisé doivent tenir compte du type d'architecture

de ce réseau (étoile, cercle).

Logiciel SCADA : un système SCADA doit être muni d'un logiciel permettant d'assurer la communication entre les éléments du SCADA et les opérateurs (interface homme-machine), et d'accomplir des tâches telles que le traitement des données, la sauvegarde des données, le déclenchement d'alarmes, et le contrôle automatique de processus de haut niveau.

2.4.1 SCADA des systèmes de transfert de pétrole

Dans les systèmes de transport étudiés, la gestion du transport de pétrole est réalisée par un système SCADA. Les opérateurs visualisent et commandent le transfert de pétrole à partir de terminaux placés dans une salle de contrôle. Les éléments du réseau de transport connectés au SCADA les plus probants du point de vue de la problématique des alignements présentée précédemment sont les suivants :

Vannes : les vannes sont connectées à des contrôleurs qui les commandent selon une valeur de pourcentage d'ouverture (ou de fermeture). De plus, ces contrôleurs communiquent avec le SCADA et envoient périodiquement le pourcentage d'ouverture de la vanne et des alarmes en cas de défaillance de la vanne.

Pompes : les pompes sont connectées à des contrôleurs qui permettent de commander ces dispositifs ainsi que de détecter une défaillance.

Capteurs de niveau : le niveau de pétrole des réservoirs peut être connu des opérateurs du SCADA grâce à des capteurs de niveau.

Fluxmètres : des fluxmètres sont souvent installés, notamment dans des oléoducs d'entrée, des oléoducs de sortie, des mélangeurs, et des bras de chargement.

D'autres dispositifs, comme des capteurs de température et de pression sont aussi reliés au SCADA, complétant la détection de fautes ou de défaillances dans le réseau.

2.4.2 Logiciel SCADA

Actuellement les logiciels SCADA utilisés dans les systèmes de PDVSA varient beaucoup d'un système à l'autre. Grâce à ces logiciels un réseau peut être surveillé et contrôlé à partir d'ordinateurs placés dans une salle de contrôle. Les réseaux sont souvent visualisés par sections. Chaque visualisation est une vue montrant une section du réseau, et contenant des objets animés représentant les dispositifs connectés au SCADA. Ces objets animés fournissent des informations en temps réel aux opérateurs. Les opérateurs peuvent agir sur les objets de type actionneurs. Par exemple, l'ouverture d'une vanne peut être commandée en cliquant dans la visualisation sur l'objet représentant cette vanne.

Dans le Terminal Maritime de Guaraguao, le logiciel Plantscape (figure 2.5), de Honeywell, est utilisé, tandis que dans le Centre de Stockage de Punta de Palmas, une ancienne version du logiciel Realflex (figure 2.6) est de moins en moins utilisée, pour raison de non compatibilité avec des nouveaux dispositifs installés.

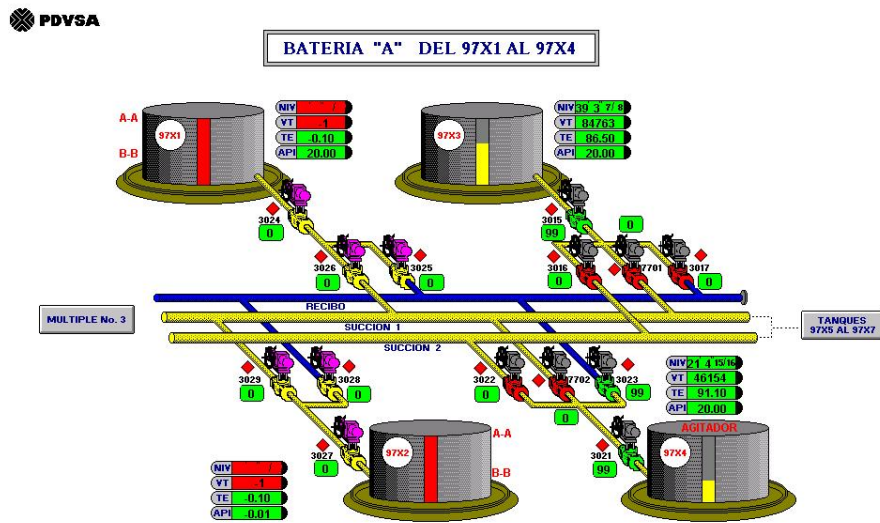


FIGURE 2.5 – Visualisation d’une section du réseau de canalisation du Terminal Maritime de Guaraguao avec le logiciel PlantScape.

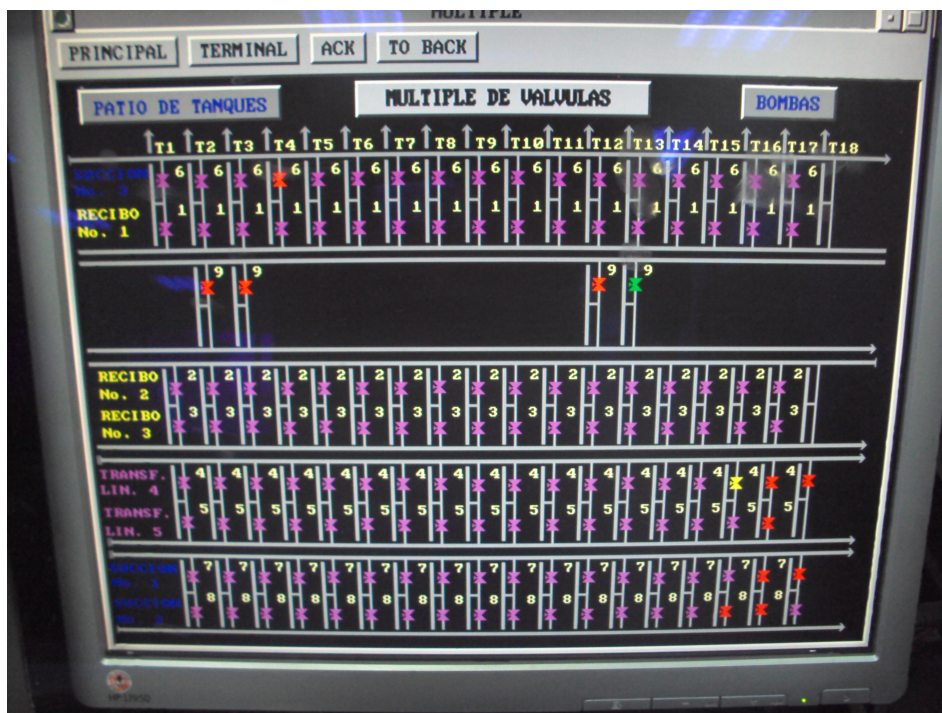


FIGURE 2.6 – Visualisation d’une section du réseau de canalisation du Centre de Stockage de Punta de Palmas avec le logiciel Reallflex.

PDVSA développe actuellement un logiciel SCADA appelé “Guardián del Alba” ou Galba, avec l’intention de diminuer la dépendance par rapport à d’autres entreprises, réduire les coûts correspondant à des licences de logiciel, et uniformiser le software utilisé dans les différents systèmes de la compagnie rendant compatible le flux d’information provenant des systèmes SCADA. Ce logiciel est disponible pour tout les pays faisant partie de l’Alliance Bolivarienne pour les Amériques ou ALBA (Venezuela, Cuba, Bolivie, Équateur, Nicaragua, Dominique, Antigua-et-Barduda), d’où son nom.

Le logiciel galba est développé par le DST (District Socialiste Technologique) de PDVSA à Mérida, Venezuela. Il est en cours d’installation dans le Centre de Stockage de Punta de Palmas, et son utilisation est aussi envisagée dans le Terminal Maritime de Guaraguao.

Les logiciels SCADA gèrent des informations concernant les dispositifs connectés au SCADA, mais généralement l’information concernant la topologie et la structure du réseau de canalisation n’existe que sous forme de vues graphiques. Dans aucun de ces logiciels cette information existe sous forme de tableau ou de matrice. L’information sous forme matricielle est très utile (voir indispensable) pour la modélisation et le traitement des données en vue d’assister automatiquement les opérateurs. Un logiciel d’assistance devra non seulement pouvoir communiquer avec le logiciel SCADA et les opérateurs, mais aussi faciliter la saisie des données concernant la topologie et la structure du réseau.

La compagnie française Thalès offre un logiciel SCADA pour des réseaux de transfert de pétrole avec une fonctionnalité plus étendue, permettant la commande simultanée de plusieurs éléments (vannes, pompes) et intégrant les informations sur la topologie des réseaux. Ces caractéristiques facilitent de manière importante la communication avec un logiciel d’assistance aux opérateurs.

2.5 État de l’art

Dans le domaine de l’optimisation du transport de pétrole et de ses dérivés, les travaux existants se concentrent principalement sur l’ordonnement des lots (batchs) de produit dans des oléoducs [Koenig et al., 1999] [Rejowsky & Pinto, 2002]. Pour cette thèse nous avons fait appel à la bibliographie appartenant à deux domaines d’étude différents : la modélisation des réseaux de canalisation, et le développement d’algorithmes de graphes. La modélisation des réseaux de canalisation est un sujet avec des applications particulières dans les systèmes de canalisation d’eau, les systèmes de drainage, les systèmes d’irrigation, et dans le transport de gaz, pétrole et de ses dérivés. Le développement d’algorithmes de graphes est un sujet avec une très large plage d’applications dans des domaines très divers, tels que les télécommunications, les réseaux sociaux, les interactions entre protéines, les structures moléculaires, le traitement d’images, les circuits électroniques et les systèmes de transport.

2.5.1 Modélisation des réseaux de canalisation

La modélisation des réseaux de canalisation repose sur des modèles mathématiques très liés au concept de graphe. À partir de ces modèles, plusieurs outils ont été développés pour servir d'outils aux ingénieurs dans la conception, la simulation et l'analyse de ces systèmes.

Les graphes sont des modèles mathématiques naturellement adaptés à la représentation de réseau de canalisation, et des réseaux en général [Bondy & Murty, 1982]. La théorie des graphes a été utilisée pour l'analyse du flux et de la pression sur un réseau de canalisation [Gupta & Prasad, 2000] [Salem & Nagy, 2004], l'étude des comportements transitoires [Zecchin et al., 2009], et le développement de méthodes de conception de réseaux [Walters & Smith, 1995] [Afshar, 2008]. Dans le chapitre suivant, les concepts de base de la théorie des graphes seront introduits.

La plupart des travaux d'étude des réseaux de canalisation sont focalisés sur le comportement continu. Les réseaux de Petri, un type particulier de graphe bipartite, ont été utilisés pour modéliser des systèmes avec une dynamique d'événements discrets, permettant l'étude des réseaux de canalisation ayant des états de fonctionnement définis, et des transitions entre ces états considérées instantanées. [Chou & Chang, 2005] ont présenté une méthode de synthèse de contrôleurs basée sur les Réseaux de Petri. [Dejean et al., 2007] utilise ce type de modélisation pour réaliser une analyse de disponibilité d'une raffinerie. Cependant les réseaux de Petri ne sont pas des modèles adaptés à la problématique présentée dans le chapitre précédent. Les contraintes considérées dans la problématique des alignements ne correspondent pas à des relations de dépendance, ou de concurrence, entre les événements discrets (fermeture/ouverture de vannes, démarrage/arrêt des pompes), c'est pourquoi la relation de dépendance implicite dans les règles de franchissement des transitions des réseaux de Petri n'est pas utile.

Les modèles mathématiques les plus adaptés à la problématique des alignements sont les graphes non orientés et les graphes orientés, en particulier les graphes de flot. Ces graphes permettent de modéliser des connexions sous forme de chaînes ou chemins, et de tenir compte des possibles interférences entre ces connexions. Dans la section suivante, des algorithmes du domaine de la recherche opérationnelle utilisant ces modèles seront introduits. Dans le chapitre suivant, ils seront présentés d'une façon détaillée.

Outils de modélisation et simulation

Plusieurs outils logiciels sont actuellement utilisés par les ingénieurs pour analyser les réseaux de canalisation et tester leurs performances avant la mise en œuvre (simulations), réduisant ainsi les coûts entraînés par des tests réels. Dans le cadre de cette thèse nous avons étudié plusieurs de ces logiciels, leur utilité pour notre problématique et le type de modélisation sur lequel ils reposent :

Matlab-Simulink : Simulink est un logiciel de simulation multi-domaine et de modélisation de systèmes dynamiques. Il utilise une approche port-signal (chaque

port soit envoi soit reçoit un signal) dans lequel l'information est transférée d'un bloc (représentant un composant) vers un autre dans une direction fixe. Comme la simulation des réseaux hydrauliques implique un échange d'information dans les deux directions, la modélisation sous Simulink nécessite deux connexions entre chaque paire de composants liés, ce qui peut rendre compliquée la modélisation. Des paquets (packages) et des logiciels basés sur Matlab-Simulink et orientés spécifiquement sur la simulation de systèmes hydrauliques et pneumatiques, facilitent la modélisation des réseaux de canalisation. C'est le cas de produits informatiques comme FLOWMASTER, et les logiciels ACUSCOMP, ACUSYS et PNEUMA de SATE Srl. [Herrán-González et al., 2009] et [Behbahani-Nejad & A.Bagheri, 2010] présentent des bibliothèques pour simuler des réseaux de canalisation de gaz.

AMESim LMS Imagine : AMESim est un outil de simulation permettant la modélisation et l'analyse de systèmes multi-domaines. Les composants des systèmes sont décrits en utilisant des modèles analytiques validés représentant des comportements qui peuvent être de type hydraulique, pneumatique, électrique ou mécanique. Bien qu'utilisé généralement pour répondre à des besoins liés à l'industrie automobile, aérospatiale, offshore et robotique, ce logiciel a été utilisé pendant la première année de cette thèse pour analyser l'impact des coups de bélier sur les canalisations pour le transport de pétrole, à l'aide des modèles représentant des canalisations, des jonctions, des vannes, et des pompes, issus de la bibliothèque de composants hydrauliques. [Lebrun et al., 2009] donnent une description des caractéristiques techniques qui ont été les objectifs principaux du développement de ce logiciel, et quelques exemples d'applications typiques, dans le cas des systèmes hydrauliques. En particulier, l'approche multi-ports (chaque port envoie et reçoit des signaux) utilisée par AMESim est présentée comme très avantageuse dans le cas de systèmes avec transmission de puissance, en contraste avec l'approche port-signal utilisée par d'autres logiciels.

Bentley WaterCAD® : WaterCAD® est un logiciel conçu pour la modélisation des systèmes de distribution d'eau [Bentley WaterCad Inc, 2010]. Il permet des analyses hydrauliques, des analyses de coûts et de consommation d'énergie. Il tient compte de la topologie et des élévations de sites avec une modélisation en 3-D. Il utilise la notion de "segment" dans les réseaux de canalisation, ce qui permet une modélisation sous forme de graphe basée sur les vannes. Nous utiliserons cette notion de segment pour notre modélisation.

RAINCAD : RAINCAD est un logiciel pour la conception de réseaux d'irrigation, permettant le dimensionnement du réseau et de ses composants pour satisfaire des contraintes d'espace, de flux et de pression.

Matgas : Matgas est un projet de simulation des réseaux de oléoducs de gaz en temps réel développé actuellement par PDVSA. Il permet l'analyse d'un réseau tenant compte de sa topologie et des paramètres de pression, température et flux.

Ces outils utilisent différents modèles suivant le type d'analyse envisagé. Une analyse basée sur des variables physiques (pression, flux, température, etc.) va faire appel à des modèles détaillés permettant de prendre en compte des paramètres physiques de chaque composant et des connexions entre ces composants, lesquels

sont nécessaires pour les opérations de calcul réalisées pendant les simulations. Dans ce cas les connexions entre les différents composants ou éléments sont cohérentes avec une représentation sous forme de graphe.

Dans la problématique des alignements nous avons retenu notamment le modèle de graphe du logiciel Bentley WaterCAD dans lequel les réseaux sont segmentés à partir des vannes. Dans le chapitre 4, la modélisation à partir des segments sera présentée, ainsi que son utilité pour la représentation des alignements.

2.5.2 Algorithmes de graphes

La problématique des alignements abordée dans cette thèse est très liée à des sujets classiques de la recherche opérationnelle, en particulier aux algorithmes de graphes. Intuitivement les alignements entre deux éléments du réseau peuvent être vus comme des chemins dans un graphe. La bibliographie a abordé des problèmes liés à la notion de chemin, tels que le problème du plus court chemin, le problème du flot maximum, et le problème du nombre maximum de chemins disjoints. Ces problèmes définissent souvent différents critères de choix des chemins. Nous avons considéré aussi des critères moins évidents, tels que les différents indices de centralité des nœuds dans les graphes.

2.5.2.1 Problème du plus court chemin

Le problème du plus court chemin sera défini de manière rigoureuse dans le chapitre suivant. Il s'agit de trouver un chemin dans un réseau avec des coûts assignés à chaque arc, de façon à minimiser le coût total des arcs utilisés pour connecter deux nœuds du graphe.

Edsger Dijkstra a publié en 1959 un article de référence sur le problème du plus court chemin pour des arcs à coût positif [Dijkstra, 1959]. Le problème du plus court chemin, en considérant la possibilité des arcs à coûts négatifs (sans cycle de coût négatif), est généralement traité en utilisant l'algorithme Bellman-Ford [Bellman, 1958] [Ford & Fulkerson, 1962]. Ces algorithmes utilisent les données du graphe sous forme de liste d'adjacences. Plusieurs travaux ont étendue ses études pour améliorer l'efficacité asymptotique dans des cas particuliers [Cherkassky & A. V. Goldberg, 1996], tandis que d'autres travaux s'adressent à des problèmes similaires.

Eppstein [Eppstein, 1999] s'adresse à la recherche des k plus courts chemins entre deux nœuds. Dans ce cas les algorithmes utilisent des informations (coûts des chemins vers d'autres nœuds, coûts estimés) ayant permis de trouver le plus court chemin pour accélérer la recherche du deuxième plus court chemin, puis du troisième plus court chemin, etc...

D'autres algorithmes permettent la détermination des plus courts chemins entre chaque paire de nœuds du graphe [Lawler, 1976] [Floyd, 1962] et [Johnson, 1977]. Dans ce cas, l'information utilisée pour trouver un plus court chemin entre une paire de nœuds, peut être exploitée pour accélérer la détermination du plus court chemin

d'un autre paire de nœuds du graphe. La plupart de ces algorithmes utilisent les données du graphe sous forme de matrice d'adjacence.

2.5.2.2 Problème du flot maximum

Le problème du flot maximum nécessite un modèle du système sous forme de graphe de flot, avec des capacités affectées à chaque arc, et avec un nœud source et un nœud puits. Dans le chapitre suivant ce problème sera défini de manière rigoureuse.

La première solution pour le problème du flot maximum a été présentée par L. R. Ford et D. R. Fulkerson [Ford & Fulkerson, 1956] utilisant les notions de graphe résiduel et des chemins d'augmentation du flot. Cette solution, présente une méthode générale de résolution du problème. À partir de cette méthode plusieurs algorithmes ont été développés, améliorant l'efficacité algorithmique pour des types de graphes particuliers. Il faut remarquer que la méthode de Ford et Fulkerson fonctionne pour des valeurs entières du flot. Pour des valeurs réelles, des méthodes basées sur la programmation linéaire sont mieux adaptées [Orlin, 1997]. La méthode de Ford-Fulkerson sera présentée en détail dans le chapitre suivant.

L'algorithme présenté par [Edmonds & Karp, 1972] est basé sur la méthode de Ford Fulkerson, et utilise une sous-routine de parcours en largeur (BFS, Breadth-First Search) pour réaliser la recherche des chemins d'augmentation.

Les algorithmes basés sur des méthodes de poussage/re-étiquetage (push-relabel) [Goldberg, 1987] [Goldberg & Tarjan, 1988] sont les plus utilisés actuellement, grâce à des implantations avec des temps de calcul très réduits. Ils sont basés sur la notion de "pré-flots" proposée par [Karmarkar, 1984]. Cette méthode est notamment utilisée pour des problèmes de flots à coût minimum.

L'algorithme de Goldberg et Rao [Goldberg & Rao, 1998], basé sur la notion des flots bloquants [Dinic, 1970], a un temps asymptotique de calcul très réduit.

Le problème du flot maximum à coût minimum rajoute des contraintes supplémentaires au problème du flot maximum. Dans ce cas les arcs sont aussi affectés d'un coût proportionnel à la quantité de flot traversant l'arc. [Liu, 2003] et [Sokkalingam et al., 1996] présentent des algorithmes permettant de résoudre ce problème.

2.5.2.3 Problème du nombre maximum de chemins disjoints

Le problème du nombre maximum de chemins disjoints considère des paires de nœuds à connecter dans un graphe, et cherche à trouver le nombre maximum de chemins disjoints connectant ces nœuds. Dans le chapitre suivant ce problème sera défini de manière rigoureuse.

[Karp, 1975] a montré que ce problème est NP-complet, alors il est improbable de trouver une solution du problème en temps polynomial. Dans ce cas, il est convenable d'utiliser des algorithmes d'approximation [Ausiello et al., 2003] [Vazirani, 2003], lesquels vont permettre de trouver un ensemble de chemins disjoints qui ne sera pas forcément maximum, mais qui sera calculé en temps polynomial. [Kleinberg, 1996]

présente et analyse plusieurs des algorithmes d'approximation pour ce problème. [Kolman & Scheideler, 2001] proposent des algorithmes de résolution en ligne pour ce même problème.

2.5.2.4 Indices de centralité

La notion de capacité opérative proposée dans cette thèse, et plus particulièrement l'analyse de criticité des éléments d'un réseau à partir de la capacité opérative, a une similitude avec les indices de centralité définis pour tenir compte des différents critères d'importance dans un graphe ou dans un réseau. Souvent ces notions de centralité ont été utilisées pour étudier des réseaux sociaux ou des réseaux de communication. Dans ces systèmes l'information est souvent transmise selon le plus court chemin : c'est le cas des réseaux téléphoniques, des réseaux de transport aérien, et de l'internet. [Bavelas, 1948], [Beauchamp, 1966] et [Sabidussi, 1966] utilisent des indices de centralité basés sur la somme des distances minimales d'un nœud vers tous les autres nœuds du graphe.

La centralité d'intermédiation (betwenness centrality) définie par [Freeman, 1977] comptabilise le nombre de plus courts chemins traversant un nœud. Au lieu de mesurer la distance d'un nœud vers les autres nœuds du graphe, cet indice mesure la capacité d'un nœud à contrôler et même à arrêter le flux d'information entre les autres nœuds du réseau. La centralité d'intermédiation d'un nœud est calculée à partir du nombre de plus courts chemins entre toutes paires de nœuds du graphe. Cet indice a été utilisé dans des domaines assez distincts : [Narayanan, 2005] s'adresse à l'étude des interactions protéine-protéine dans des réseaux biologiques. [Guimera et al., 2005] utilise la centralité d'intermédiation pour analyser l'importance des villes dans des réseaux de transport aérien.

2.6 Conclusion

Dans ce chapitre nous avons présenté une problématique concrète dans un domaine de grande importance économique. Il faut remarquer que cette problématique a été formalisée à partir des échanges avec les opérateurs des systèmes de transport de pétrole vénézuéliens. Après un travail de recherche bibliographique, nous n'avons pas trouvé de véritable écho à cette problématique pétrolière. Il s'agit donc d'un sujet d'étude nouveau en terme d'application, ayant toutefois des similitudes avec des problèmes plus théoriques et/ou avec d'autres domaines d'application.

L'absence de référence à des études similaires dans la bibliographie peut s'expliquer notamment par la tradition du secret industriel dans le domaine de l'industrie pétrolière [Mommer, 2010]. Une autre explication possible est le faible besoin d'optimisation des opérations dans une industrie qui a été historiquement très rentable. Cependant, l'augmentation de la demande de pétrole, et l'épuisement des réserves pétrolières dans plusieurs pays producteurs, est en train d'imposer un besoin d'optimisation des opérations de production, raffinage et transport de pétrole et ses

dérivés : des gisements de pétrole extra-lourd (au Venezuela) et de sables bitumineux (au Canada) étant considérées peu rentables auparavant, sont actuellement de plus en plus exploités; de plus en plus de développements actuels ciblent la réutilisation des résidus du raffinage, l'optimisation des opérations de maintenance, l'ordonnancement des lots de produits dans des oléoducs multi-produits, etc. Dans ce contexte, l'optimisation des opérations de transport de pétrole fait partie des projets à prendre en compte dans un futur prochain. L'intérêt porté à ce sujet par la compagnie vénézuélienne PDVSA, dans le cadre de cette thèse, et de la compagnie française Thalès, dans le cadre d'un projet d'études post-gradués (PCP), montre l'importance de cette problématique.

Dans le cas particulier du Venezuela l'optimisation des opérations de transport de pétrole s'inscrit dans un contexte de développement de logiciels de supervision et contrôle (Galba) de la part de PDVSA.

La problématique des alignements n'étant pas abordée de manière spécifique dans la bibliographie, une partie de notre travail consiste à traduire des spécifications techniques et des critères de choix des alignements, basés sur l'expertise des opérateurs des systèmes étudiés, en un langage formel permettant de lier cette problématique à des problèmes classiques de la recherche opérationnelle. Une fois ce lien établi, le développement de nouveaux algorithmes ayant un intérêt du point de vue scientifique (ainsi que du point de vue de l'application) sera abordé.

La modélisation retenue fait appel à des notions de la théorie des graphes qui seront présentées dans le chapitre suivant. Elle va reprendre la notion de segment, utilisée par le logiciel Bentley WaterCAD, explicitant la relation des alignements avec des chaînes dans un graphe non orienté.

Le problème du nombre maximum de chemins disjoints entre des paires de nœuds a des similitudes avec la problématique des alignements. La contrainte d'isolement des alignements dans un réseau de canalisation est analogue à la problématique des chemins ou chaînes disjointes dans un graphe. Néanmoins, le problème du nombre maximum de chemins disjoints entre des paires de nœuds ne tient pas compte des multiples possibilités de choix d'éléments fournisseurs ou d'éléments récepteurs. Il est donc nécessaire de poser le problème du nombre maximum de chemins disjoints de manière à tenir compte de cette particularité.

Des problèmes plus spécifiques sur les alignements, concernant le choix d'un alignement minimisant les interférences, des alignements forcés, ou l'analyse de la criticité des éléments par rapport à leur influence sur les alignements, requièrent du développement de nouveaux algorithmes. Ceux-ci seront basés sur des travaux précédents concernant la recherche du plus court chemin, et du flot maximum.

Cadre théorique

Résumé : Les solutions à la problématique des alignements proposées dans cette thèse se basent sur une modélisation sous forme de graphes et sur l'utilisation d'algorithmes de graphes. Ce chapitre présente les définitions mathématiques issues de la théorie des graphes qui seront utilisées par la suite, ainsi que les bases théoriques des contributions présentées dans le chapitre suivant. En particulier des algorithmes classiques du plus court chemin et du flot maximum seront présentés.

Sommaire

3.1	Modélisation sous forme de graphe	29
3.1.1	Définitions et concepts de base	30
3.1.2	Représentation des graphes	34
3.2	Algorithmes de graphes	37
3.2.1	Problème du plus court chemin	38
3.2.2	Problème du flot maximum	42
3.2.3	Problème du nombre maximum de chemins disjoints	48
3.3	Conclusion	48

3.1 Modélisation sous forme de graphe

Les graphes sont des outils qui permettent de modéliser un système sous forme d'un ensemble d'objets mathématiques avec des liaisons ou des relations binaires entre ses éléments. Cette modélisation permet d'extraire des caractéristiques intrinsèques d'un système qui sont utiles pour résoudre des problèmes particuliers, en faisant abstraction du reste des caractéristiques du système. La résolution de ces problèmes va généralement impliquer la recherche d'un ensemble d'éléments du système satisfaisant des contraintes imposées, ou la détermination de propriétés générales du système dépendant des liens entre les éléments. La problématique introduite dans le premier chapitre correspond à une recherche de routes d'alignement, définies comme une suite d'éléments (canalisations) du système considéré (réseau). Cette problématique relève de la théorie des graphes ce qui justifie ce type de modélisation. Nous allons introduire les définitions et les concepts de base appartenant à la théorie des graphes et qui seront utilisés par la suite, ainsi que les représentations matricielles des graphes.

3.1.1 Définitions et concepts de base

L'ensemble des notions présentées dans cette section est nécessaire pour l'interprétation de nos propositions. Par contre nous laissons le lecteur rechercher de plus amples connaissances dans les références suivantes : [Bondy & Murty, 1982], [Gondran & Minoux, 2009].

3.1.1.1 Graphes orientés

Un graphe orienté G est un couple $G = (V, E)$ constitué :

- d'un ensemble V dont les éléments sont appelés nœuds ou sommets. Nous noterons $|V|$ le nombre de nœuds du graphe. Lorsque plusieurs graphes sont considérés, nous noterons $V[G]$ l'ensemble de nœuds du graphe G .
- d'un ensemble E dont les éléments $a \in E$ sont des couples ordonnés de nœuds appelées arcs. Si $a = (u, v) \in V^2$ est un arc de G , u est l'extrémité initiale de a et v l'extrémité terminale de a . Nous noterons $|E|$ le nombre d'arcs du graphe. Un arc orienté est un arc d'un graphe orienté. Lorsque plusieurs graphes sont considérés, nous noterons $E[G]$ l'ensemble des arcs du graphe G .

Dans les représentations graphiques les nœuds seront représentés par des cercles ou des points, tandis que les arcs seront représentés par des flèches connectant deux nœuds. L'arc $a = (u, v)$ sera représenté par une flèche joignant les cercles (ou les points) u et v (v correspondant à la pointe de la flèche). La figure 3.1 montre la représentation graphique d'un graphe orienté :

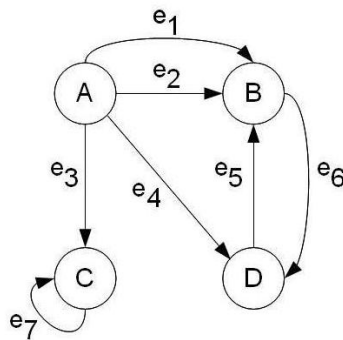


FIGURE 3.1 – Représentation graphique d'un graphe orienté.

Un arc $a = (u, v)$ dont les extrémités coïncident ($u = v$) est appelé boucle.

Un p -graphe est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (u, v) entre deux nœuds quelconques u et v , pris dans cet ordre. En particulier, un 1-graphe est un graphe tel qu'il n'existe jamais plus d'un arc de la forme (i, j) , pour tout nœud $i \in V$ et tout nœud $j \in V$.

Un arc incident du nœud n est un arc de la forme (a, n) , $a \in V$. Un arc sortant du nœud n est un arc de la forme (n, a) , $a \in V$.

Un nœud s est dit successeur du nœud n s'il existe un arc ayant n comme extrémité initiale et s comme extrémité terminale. De même, un nœud p est dit prédécesseur du nœud n s'il existe un arc ayant p comme extrémité initiale et n comme extrémité terminale.

3.1.1.2 Graphes non orientés

Les graphes non orientés sont des graphes dans lesquels les arcs ne sont pas orientés. Ils sont définis de manière similaire aux graphes orientés.

Un graphe non orienté G est un couple $G = (V, E)$ constitué :

- d'un ensemble V dont les éléments sont appelés des nœuds ou des sommets. $|V|$ exprime le nombre de nœuds du graphe.
- d'un ensemble E dont les éléments $a \in E$ sont des couples non ordonnés de nœuds appelés des arêtes, ou arc non orienté, avec $|E|$ le nombre d'arêtes du graphe. Si $a = (u, v) \in V^2$ est une arête de G , u et v sont les extrémités de a .

Dans les représentations graphiques les nœuds seront représentés par des cercles ou des points, tandis que les arêtes seront représentés par des lignes (sans flèche) connectant deux nœuds. La figure 3.2 est une représentation graphique d'un graphe non orienté :

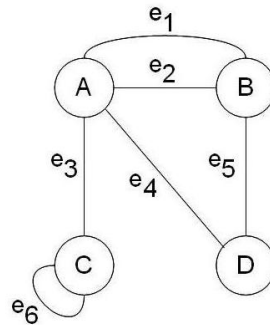


FIGURE 3.2 – Représentation graphique d'un graphe non orienté.

Les graphes non orientés peuvent être aperçus comme un type particulier de graphe orienté, dans lequel pour chaque arc orienté $a = (u, v)$ il existe un arc complémentaire $a' = (v, u)$.

Un multigraphe est un graphe dans lequel il peut exister plusieurs arêtes entre deux nœuds donnés.

Un graphe est dit simple s'il n'a pas de boucle et s'il n'y a jamais plus d'une arête entre deux nœuds quelconques.

3.1.1.3 Principales définitions

Arcs adjacents. Arêtes adjacentes Deux arcs (ou arêtes) sont adjacents s'ils ont au moins une extrémité commune.

Degrés et demi-degrés Le demi-degré extérieur du nœud u , noté $d_G^+(u)$, est le nombre d'arcs (ou d'arêtes) ayant u comme extrémité initiale. Le demi-degré intérieur du nœud u , noté $d_G^-(u)$, est le nombre d'arcs (ou d'arêtes) ayant u comme extrémité terminale. Le degré du nœud u , noté $d_G(u)$, est le nombre d'arcs (ou d'arêtes) ayant u comme extrémité. Pour un nœud u nous avons : $d_G(u) = d_G^+(u) + d_G^-(u)$

Arcs incidents (arêtes incidentes) à $A \subset V$. Cocycles Étant donné un sous-ensemble de nœuds $A \subset V$, nous définissons :

$\omega^+(A)$ = ensemble des arcs ayant une extrémité initiale dans A , et leur extrémité terminale dans $V - A$.

$\omega^-(A)$ = ensemble des arcs ayant leur extrémité terminale dans A , et leur extrémité initiale dans $V - A$.

$\omega(A) = \omega^+(A) + \omega^-(A)$.

L'ensemble d'arcs (ou d'arêtes) $\omega(A)$ est appelé un cocycle du graphe.

Graphes symétriques Un graphe G est dit symétrique si pour toute paire de nœuds (u, v) il existe autant d'arcs de la forme (u, v) que de la forme (v, u) .

Graphes antisymétriques Un 1-graphe $G = (V, E)$ est dit antisymétrique si : $(i, j) \in E \Rightarrow (j, i) \notin E$.

Sous-graphe induit par un sous-ensemble de nœuds Étant donné $A \subset V$, le sous-graphe induit par A est le graphe G_A dont les nœuds sont les éléments de A et dont les arcs sont les arcs de G ayant leur deux extrémités dans A .

Graphe partiel engendré par un sous-ensemble d'arcs Soient un graphe $G = (V, E)$ et $A \subset E$. Le graphe partiel engendré par A est le graphe ayant le même ensemble V de nœuds que G , et dont les arcs sont les arcs de A .

Sous-graphe partiel Étant donné un graphe $G = (V, E)$ et $U \subset V$, $A \subset E$, le sous-graphe partiel engendré par U et A est le graphe partiel G_A engendré par U .

Chaîne Dans un graphe orienté ou non orienté $G = (V, E)$, une chaîne de longueur q est une suite de q arcs ou arêtes de E : $C = e_1, e_2, \dots, e_q$ telle que chaque arc (ou arête) e_n de la suite ($2 \leq n \leq q - 1$) ait une extrémité commune avec l'arc e_{n-1} et l'autre extrémité commune avec l'arc e_{n+1} . L'extrémité i de e_1 non adjacente à e_2 , et l'extrémité j de e_q non adjacente à e_{q-1} sont appelées les extrémités de la chaîne C .

Deux chaînes sont disjointes par les arêtes, si et seulement si aucune arête faisant partie d'une des chaînes, ne fait aussi partie de l'autre.

Nous appelons chaîne élémentaire une chaîne dans laquelle chaque nœud est parcouru une seule fois. Dans une chaîne élémentaire tous les nœuds sont de degré 2 au plus. En général les chaînes seront considérées élémentaires dans cette thèse.

Dans un graphe simple, une chaîne de longueur q est parfaitement définie par la succession des nœuds qu'elle rencontre : u_1, u_2, \dots, u_q .

Deux chaînes sont disjointes par les nœuds, si et seulement si aucun nœud faisant partie d'une des chaînes, ne fait aussi partie de l'autre. Si deux chaînes sont disjointes par les nœuds, elles sont aussi disjointes par les arêtes.

Un cycle est une chaîne dont les extrémités coïncident.

Un cycle élémentaire est une chaîne élémentaire dont les extrémités coïncident. Il ne contient aucun autre cycle.

Chemin Dans un graphe orienté $G = (V, E)$ un chemin de longueur q est une suite de q arcs de E : $P = e_1, e_2, \dots, e_q$, avec $e_n = (u_{n-1}, u_n)$.

Autrement dit l'extrémité initiale de chaque arc en $(2 \leq n \leq q - 1)$ coïncide avec l'extrémité terminale de l'arc e_{n-1} adjacent, et l'extrémité terminale de chaque arc coïncide avec l'extrémité initiale de l'arc e_{n+1} adjacent. Un chemin est alors une chaîne dont tous les arcs sont orientés dans le même sens. Le nœud u_0 est l'extrémité initiale du chemin P , et le nœud u_q est l'extrémité terminale du chemin P .

La notation $u \overset{p}{\rightsquigarrow} v$ désigne un chemin p ayant u comme extrémité initiale et v comme extrémité terminale.

Deux chemins sont disjoints par ses arcs, si et seulement si aucun arc faisant partie d'un des chemins, ne fait aussi partie de l'autre.

Dans le cas d'un 1-graphe un chemin est parfaitement défini par la succession des nœuds qu'il rencontre : u_1, u_2, \dots, u_q .

Deux chemins sont disjoints par ses nœuds, si et seulement si aucun nœuds faisant partie d'un des chemins, ne fait aussi partie de l'autre. Si deux chemins sont disjoints par les nœuds, ils sont aussi disjoints par les arcs.

Nous appelons chemin élémentaire un chemin dans lequel chaque nœud est parcouru une seule fois. Dans un chemin élémentaire tous les nœuds sont de degré 2 au plus. En général, les chemins seront considérés élémentaires dans cette thèse.

Un circuit est un chemin dont les extrémités coïncident. Un circuit élémentaire est un chemin élémentaire dont les extrémités coïncident.

Sous-chemin Un sous-chemin S du chemin $P = e_1, e_2, \dots, e_q$ est un chemin dont la suite d'arcs est contenue dans la suite d'arcs de P .

Graphe connexe Un graphe est dit connexe si, pour tout couple de nœuds u et v , il existe une chaîne joignant u et v . Une composante connexe d'un graphe G est un sous-graphe maximal de G tel que pour toute paire de sommets u et v dans ce sous-graphe, il existe une chaîne de u à v et un chemin de v à u .

Graphe fortement connexe Un graphe orienté est dit fortement connexe si, pour tout couple de nœuds u et v , il existe un chemin d'extrémité initial u et d'extrémité terminale v . Une composante fortement connexe d'un graphe orienté G est un sous-graphe maximal de G tel que pour toute paire de sommets u et v dans ce sous-graphe, il existe un chemin de u à v et un chemin de v à u .

3.1.2 Représentation des graphes

La modélisation sous forme de graphe est très utile lorsqu'elle peut être utilisée pour exécuter des algorithmes, ce qui est en pratique la seule façon de résoudre des problèmes de grande complexité. L'exécution de ces algorithmes nécessite une représentation des graphes compatible avec les structures de base de données et de traitement de données utilisées par les ordinateurs. Cette représentation est possible grâce à l'utilisation des matrices et des listes d'éléments. Ces représentations peuvent être décomposées en deux types : les représentations basées sur les matrices d'incidence et les représentations basées sur les matrices d'adjacence. Chaque type de représentation possède des propriétés de compacité et vitesse de calcul qui seront à prendre en compte lors de l'exécution d'un algorithme de graphes.

3.1.2.1 Représentations basées sur les matrices d'incidence

Les représentations basées sur les matrices d'incidence permettent de représenter des multigraphes (plusieurs arcs ou arêtes entre les mêmes paires de nœuds). Cependant, dans le cas des matrices d'incidence, les boucles ne peuvent pas être représentées.

Matrice d'incidence d'un graphe orienté La matrice d'incidence d'un graphe orienté $G = (V, E)$, ou matrice nœuds-arcs, est une matrice $A = (a_{ij}), i = 1, \dots, N, j = 1, \dots, M$ à coefficients entiers 0, +1, -1, telle que chaque colonne correspond à un arc de G , et chaque ligne à un nœud de G ; si $e = (i, j) \in E$, la colonne e a tous ses termes nuls, sauf :

$$\begin{aligned} a_{ie} &= +1 \\ a_{je} &= -1 \end{aligned}$$

La figure 3.3 représente un graphe et sa matrice d'incidence.

Matrice d'incidence d'un graphe non orienté La matrice d'incidence d'un graphe non orienté $G = (V, E)$, ou matrice nœuds-arêtes, est une matrice $A = (a_{ie}), i = 1, \dots, N, j = 1, \dots, M$ à coefficients entiers 0, +1, telle que chaque colonne correspond à une arête de G , et chaque ligne à un nœud de G ; si $e = (u, v)$, la colonne e a tous ses termes nuls, sauf :

$$\begin{aligned} a_{ie} &= +1 \\ a_{je} &= +1 \end{aligned}$$

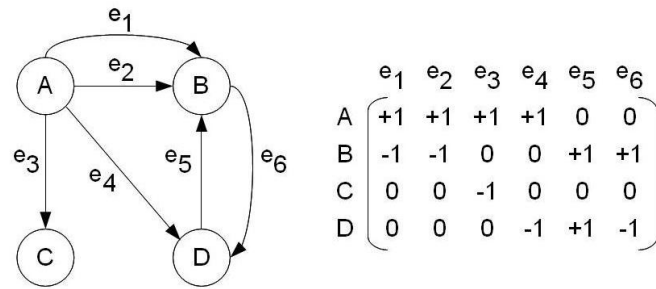


FIGURE 3.3 – Représentation graphique et matrice d'incidence d'un graphe orienté.

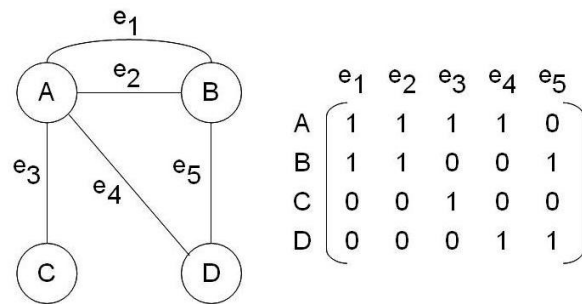


FIGURE 3.4 – Représentation graphique et matrice d'incidence d'un graphe non orienté.

La figure 3.4 donne la représentation graphique et la matrice d'incidence d'un graphe non orienté.

Liste d'arcs ou d'arêtes Une liste d'arcs est une forme condensée de la matrice d'incidence, dans laquelle les termes nuls de la matrice ne sont pas pris en compte. À chaque arc correspond la paire de nœuds qu'il connecte. Cette représentation est très utile lorsque des informations spécifiques doivent être affectées à chaque arc. Elle permet de représenter des boucles, alors tout type de graphe peut être représenté par une liste d'arcs.

La liste suivante représente le graphe de la figure 3.4 :

$$\begin{aligned} e_1 &= (A, B) \\ e_2 &= (A, B) \\ e_3 &= (A, C) \\ e_4 &= (A, D) \\ e_5 &= (B, D) \\ e_6 &= (D, B) \end{aligned}$$

3.1.2.2 Représentations basées sur les matrices d'adjacence

Les représentations basées sur les matrices d'adjacence permettent de représenter des 1-graphes (seulement un arc ou une arête entre les mêmes paires de nœuds) orientés ou non orientés, avec ou sans boucles. Comme chaque arc (ou arête) peut être défini(e) par un couple de nœuds, il n'est pas nécessaire d'étiqueter les arcs. Ces représentations sont très utiles dans le cas d'algorithmes pour lesquels une recherche rapide des nœuds adjacents est souhaitée.

Matrice d'adjacence Soit $G = (V, E)$ un 1-graphe, comportant éventuellement des boucles. La matrice d'adjacence, ou matrice nœuds-nœuds, est une matrice $A = (a_{ij}), i = 1, \dots, N, j = 1, \dots, N$ à coefficients entiers 0 ou +1, telle que chaque colonne correspond à un nœud de G , chaque ligne à un nœud de G ; et :

$$\begin{aligned} a_{ij} &= +1 \text{ si et seulement si : } (i, j) \in E \\ a_{ij} &= 0 \text{ sinon} \end{aligned}$$

La figure 3.5 montre la représentation graphique et la matrice d'adjacence d'un graphe orienté :

Dans le cas non orienté, la matrice d'adjacence d'un graphe simple peut être définie en considérant qu'à chaque arête (i, j) correspondent deux arcs : (i, j) et (j, i) . Dans ce cas, la matrice d'adjacence est symétrique.

Liste d'adjacence Une liste d'adjacence est une forme condensée de la matrice d'adjacence, dans laquelle les termes nuls de la matrice ne sont pas pris en compte. Dans le cas d'un graphe non orienté, à chaque nœud est affectée une liste avec ses

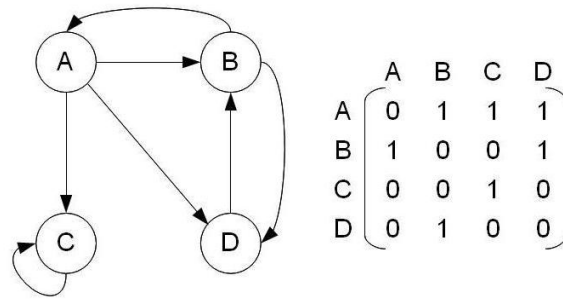


FIGURE 3.5 – Représentation graphique et matrice d'adjacence d'un graphe orienté.

nœuds adjacents.. Dans le cas d'un graphe orienté, à chaque nœud est affectée une liste avec ses nœuds successeurs.

La liste suivante représente le graphe orienté de la figure 3.5 :

A : (*B*, *C*, *D*)
B : (*A*, *D*)
C : (*C*)
D : (*B*)

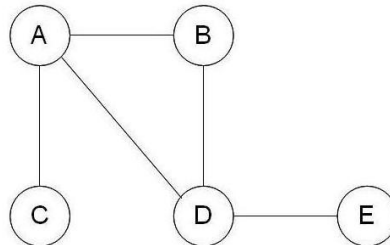


FIGURE 3.6 – Représentation graphique d'un graphe simple.

La liste suivante représente le graphe non orienté de la figure 3.6 :

A : (*B*, *C*, *D*)
B : (*A*, *D*)
C : (*A*)
D : (*A*, *B*, *E*)
E : (*D*)

3.2 Algorithmes de graphes

Les contributions présentées dans cette thèse reposent sur l'exploitation d'algorithmes de graphes classiques. Ces algorithmes sont présentés dans cette section.

Nous reprenons les notations et la représentation en pseudo-code des algorithmes utilisés par [Cormen et al., 2001].

3.2.1 Problème du plus court chemin

Le problème du plus court chemin s'adresse à des graphes dans lesquels des distances sont affectées aux arcs, et dont l'objectif est de trouver un chemin entre deux nœuds minimisant leur distance. Par la suite nous utiliserons le terme de coût plutôt que distance ou longueur d'arcs, lequel est plus général et peut être étendu aux valeurs négatives. Les coûts des arcs peuvent alors être interprétés comme des grandeurs autres que des distances, comme par exemple, des pénalités, du temps, des pertes... Ces grandeurs doivent être cumulables tout au long du chemin. Dans le cas d'un réseau de transport de pétrole, ce coût peut correspondre à différentes caractéristiques ou propriétés, comme les longueurs des canalisations, les pertes de charge, le nombre d'éléments utilisés, la fiabilité des éléments, la capacité opérative... Nous allons présenter le problème du plus court chemin et l'algorithme de Dijkstra, permettant de résoudre ce problème dans le cas des coûts non négatifs.

Soit un graphe $G = (V, E)$ et une fonction de coûts $w : E \rightarrow \mathbb{R}$ qui assigne à chaque arc de graphe un coût à valeur réelle. Le coût d'un chemin $p = (v_0, v_1, \dots, v_k)$ est égal à la somme des coûts de ses arcs :

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

Nous définissons le coût du plus court chemin entre u et v comme :

$$\delta(u, v) = \begin{cases} \min w(p) : u \xrightarrow{p} v & \text{s'il existe un chemin entre } u \text{ et } v \\ \infty & \text{sinon} \end{cases}$$

Un plus court chemin du nœud source u jusqu'au nœud terminal v est défini comme un chemin p avec un coût $w(p) = \delta(u, v)$.

3.2.1.1 Algorithme de Dijkstra

L'algorithme de Dijkstra a été proposé en 1959 [Dijkstra, 1959] et il a de très importantes applications dans divers domaines, tels que le transport, les télécommunications, l'automatique, etc. Il permet de résoudre le problème du plus court chemin dans le cas d'un graphe avec une fonction de coût à valeurs réelles positives ou nulles (pour tout arc $(u, v) \in E$, $w(u, v) \geq 0$). L'algorithme calcule les plus courts chemins de la source jusqu'à chaque nœud atteignable du graphe. Si nous ne nous intéressons qu'au plus court chemin de la source jusqu'à un nœud en particulier, alors il faut arrêter l'algorithme lorsqu'il a fini de calculer le plus court chemin jusqu'à ce nœud.

Typiquement, les algorithmes du plus court chemin se basent sur la propriété

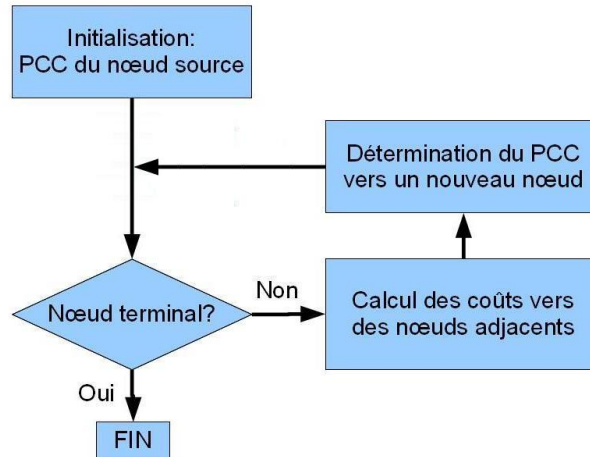


FIGURE 3.7 – Fonctionnement général de l’algorithme de Dijkstra pour le problème du plus court chemin (PCC).

d’inclusion suivante : un plus court chemin entre deux nœuds est composé de plusieurs plus courts chemins.

Soit $G = (V, E)$ un graphe orienté avec une fonction de coûts $w : E \rightarrow \mathbb{R}$, soit $p = (v_0, v_1, \dots, v_k)$ le plus court chemin du nœud v_0 au nœud v_k , et soit $p_{ij} = (v_i, v_{i+1}, \dots, v_j)$ le sous-chemin de p du nœud v_i au nœud v_j . Alors p_{ij} est un plus court chemin de v_i à v_j .

L’algorithme de Dijkstra utilise cette propriété pour déterminer de nouveaux plus courts chemins à partir des plus courts chemins déjà connus. Les nœuds dont le plus court chemin (à partir du nœud source) est déjà connu vont être utilisés pour calculer les coûts des possibles chemins vers des nœuds adjacents dont le plus court chemin est inconnu. À partir de ces coûts (estimés) il est possible de déterminer de plus courts chemins vers de nouveaux nœuds. L’algorithme s’initialise avec le plus court chemin du nœud source vers lui même, lequel a un coût nul. La figure 3.7 montre la logique de fonctionnement de cet algorithme.

Chaque nœud n du graphe est affecté d’un estimé du coût du plus court chemin, appelé $d[n]$, lequel sera une borne supérieure du coût du plus court chemin $\delta[n]$ de la source au nœud n . Ces estimés seront actualisés pendant l’exécution de l’algorithme pour se rapprocher de plus en plus de la valeur $\delta[n]$.

Chaque nœud n du graphe est affecté d’un prédécesseur $\pi[n]$ lequel sera égal soit à un autre nœud du graphe, soit à la valeur NIL (rien, nul). Ces prédécesseurs permettront de déterminer un chemin allant de la source jusqu’à un nœud n du graphe, en remontant les prédécesseurs successifs à partir de n jusqu’au nœud source. Les prédécesseurs des nœuds seront actualisés pendant l’exécution de l’algorithme.

La procédure d’initialisation de l’algorithme (algorithme 1) assigne la valeur NIL aux prédécesseurs $\pi(n)$ des nœuds $n \in V$, des valeurs infinies aux estimés d des nœuds $n \in (V - \{s\})$, et zéro à l’estimé $d[s]$ du nœud source.

Algorithme 1 Initialiser(G, s)

```

1: pour chaque nœud  $n \in V[G]$  faire
2:    $d[n] \leftarrow \infty$ 
3:    $\pi[n] \leftarrow NIL$ 
4:    $d[n] \leftarrow 0$ 

```

L'actualisation des estimés (algorithme 2) est une sous-routine qui se charge de tester si $d[v]$ (l'estimé du plus court chemin au nœud v), peut être amélioré (réduit) en prenant un chemin qui passe par u . Dans ce cas, l'estimé est remplacé par le coût du chemin passant par u , lequel est égal à $d[u] + w(u, v)$, et le prédécesseur de v devient u . Dans le cas contraire, l'estimé $d[v]$ et le prédécesseur $p[v]$ ne sont pas modifiés.

Algorithme 2 Actualiser(u, v, w)

```

1: si  $d[v] > d[u] + w(u, v)$  alors
2:    $d[v] \leftarrow d[u] + w(u, v)$ 
3:    $\pi[v] \leftarrow u$ 

```

L'algorithme (algorithme 3) utilise un ensemble S de nœuds pour lesquels le plus court chemin à partir du nœud source a déjà été déterminé. L'algorithme, de façon itérée, sélectionne un nœud $u \in (V - S)$ avec l'estimé minimum, ajoute u à l'ensemble S , et actualise tous les nœuds sortant de u .

Algorithme 3 Dijkstra(G, w, s)

```

1: Initialiser( $G, s$ )
2:  $S \leftarrow \emptyset$ 
3:  $Q \leftarrow V[G]$ 
4: tant que  $Q \neq \emptyset$  faire
5:    $u \leftarrow \text{Extraire\_min}(Q)$ 
6:    $S \leftarrow S - \{u\}$ 
7:   pour chaque nœud  $v \in \text{Adj}[u]$  faire
8:     Actualiser( $u, v, w$ )

```

La ligne 1 initialise les valeurs des estimés et des prédécesseurs des nœuds, la ligne 2 initialise l'ensemble S comme un ensemble vide et la ligne 3 initialise l'ensemble Q comme égal à l'ensemble des nœuds du graphe $V[G]$. Pendant l'exécution, au début de la boucle "tant que" (while) des lignes 4-8 l'égalité $Q = V - S$ sera toujours vraie. A chaque itération de cette boucle, le nœud u est retiré de l'ensemble Q et ajouté à l'ensemble S . Ce nœud a le plus petit coût estimé de l'ensemble Q . Les lignes 7 et 8 pour chaque arc (u, v) sortant de u actualisent l'estimé $d[v]$ et le prédécesseur $\pi[v]$ si le plus court chemin vers v peut être amélioré en prenant le chemin passant par u .

Comme l'ensemble Q commence avec un nombre $|V|$ d'éléments, et comme à

chaque itération de la boucle “tant que” (while) un élément est extrait de Q , sans ajouter aucun élément, cette boucle est itérée $|V|$ fois. La procédure “Extraire_min” contenue dans la boucle prend un temps $O|V|$, alors cette procédure rajoute un temps $O(|V|^2)$ à l’algorithme. Comme chaque nœud n est ajouté à l’ensemble S une seule fois, chaque arc dans la liste d’adjacence sera examiné une seule fois pendant l’exécution de tout l’algorithme. La procédure d’actualisation est exécutée alors $|E|$ fois. Dans le cas général, le temps asymptotique d’exécution de l’algorithme de Dijkstra est de $O(|V|^2 + |E|) = O(|V|^2)$.

Le pseudo-code précédent ne s’arrête qu’après avoir calculé le plus court chemin à partir de la source vers chaque nœud du graphe. Lorsque nous recherchons le plus court chemin vers un nœud terminal donné, l’algorithme peut s’arrêter avant le calcul de tous les plus court chemins (algorithme 4).

Algorithme 4 Dijkstra2(G, w, s)

```

1: Initialiser( $G, s$ )
2:  $S \leftarrow \emptyset$ 
3:  $Q \leftarrow V[G]$ 
4: tant que  $Q \neq \emptyset$  faire
5:    $u \leftarrow \text{Extraire\_min}(Q)$ 
6:    $S \leftarrow S - \{u\}$ 
7:   si  $u \neq t$  alors
8:     pour chaque nœud  $v \in \text{Adj}[u]$  faire
9:       Actualiser( $u, v, w$ )
  
```

La ligne 7 arrête le calcul des plus courts chemins, lorsque le plus court chemin vers le nœud terminal a été déterminé.

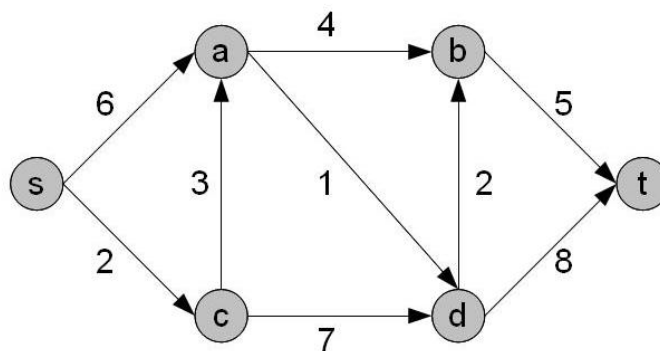


FIGURE 3.8 – Graphe orienté avec un nœud source ‘s’, et un nœud terminal ‘t’. Chaque arc est affecté d’un coût positif.

u	$d[s]$	$d[a]$	$d[b]$	$d[c]$	$d[d]$	$d[t]$
0	∞	∞	∞	∞	∞	∞
s	-	6	∞	2	∞	∞
c	-	5	∞	-	9	∞
a	-	-	9	-	6	∞
d	-	-	8	-	-	14
b	-	-	-	-	-	13

TABLE 3.1 – Valeurs des coûts estimés vers chaque nœud du graphe à chaque itération de l’algorithme de Dijkstra

Exemple La figure 3.8 montre un graphe orienté dans lequel chaque arc est affecté d’un coût positif. L’algorithme de Dijkstra permet de trouver le plus court chemin du nœud source s , au nœud terminal t .

Le tableau 3.1 montre les valeurs des coûts estimés vers chaque nœud du graphe à chaque itération de l’algorithme, ainsi que la valeur de u .

A chaque itération, u devient le nœud dont l’estimé est le plus petit (en gras). Lorsqu’un nœud devient u , son plus court chemin devient connu, alors l’estimé du coût n’est plus actualisé. Le plus court chemin de s vers t peut être reconstruit en remontant les nœuds prédécesseurs. Dans l’exemple, nous avons : $p(t) = b$, $p(b) = d$, $p(d) = a$, $p(a) = c$, $p(c) = s$, ce qui donne le chemin : (s, c, a, d, b, t) . La figure 3.9 montre le plus court chemin entre s et t .

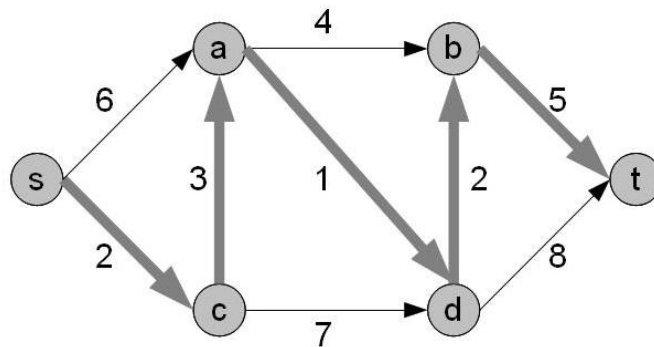


FIGURE 3.9 – Plus court chemin entre les nœuds ‘s’ et ‘t’.

3.2.2 Problème du flot maximum

Les graphes peuvent être utilisés pour modéliser un réseau dans lequel il peut y avoir une circulation de flot. Ce flot va représenter le transfert de biens, de matière, ou d’information, à travers un moyen de circulation. Par exemple, ce flot peut représenter le flux d’un liquide ou d’un gaz à travers des canalisations, du courant électrique à travers des câbles conducteurs, ou le flux de données dans un réseau de

télécommunication. Dans ces réseaux, il existe des éléments considérés comme des sources, puisqu'ils peuvent être vus comme des créateurs de flot, et des éléments considérés comme des puits, puisqu'ils peuvent être vus comme des consommateurs de flot. Il existe souvent des contraintes de capacité de flot dans différentes parties du réseau, alors le flot sera plus ou moins limité selon les routes prises pour circuler des sources aux puits. Le problème du flot maximum consiste à déterminer la quantité maximum de flot qui peut être transférée des sources aux puits. Ce problème a des applications évidentes dans les réseaux de transport et de communication, mais il a aussi des applications moins évidentes dans lesquelles un graphe représente des relations abstraites possibles entre éléments, et le flot indique l'existence de ces relations. Ces abstractions permettent de résoudre des problèmes d'ordonnement, d'affectation, et des problèmes de mathématiques combinatoires.

Dans le cas des réseaux de transport de pétrole, le problème du flot maximum permet non seulement de s'adresser au problème évident de calcul de flux maximum de pétrole, mais aussi de considérer des problèmes d'alignement. Nous allons présenter le problème du flot maximum et l'algorithme de Ford Fulkerson [Ford & Fulkerson, 1956], permettant de résoudre ce problème pour des flots à valeurs entières.

Grphe de flot Un graphe de flot $G = (V, E)$ est un graphe orienté dans lequel chaque arc $(u, v) \in E$ a une capacité positive ou nulle $c(u, v) \geq 0$. Si $(u, v) \in E$, alors $c(u, v) = 0$. Dans ce graphe, deux nœuds spéciaux sont distingués : le nœud source s et le nœud puits t . Nous définissons une fonction de flot comme $f : V \times V \rightarrow \mathbb{Z}$ satisfaisant les propriétés suivantes :

- Contrainte de capacité : $\forall u, v \in V, f(u, v) \leq c(u, v)$
- Antisymétrie : $\forall u, v \in V, f(u, v) = -f(v, u)$
- Conservation du flot : $\forall u \in V - \{s, t\}, \sum_{v \in V} f(u, v) = 0$.

La quantité $f(u, v)$ désigne le flot du nœud u vers le nœud v . Soit $G = (V, E)$ un graphe de flot, et soient f_1 et f_2 deux fonctions de flot de $V \times V$ vers \mathbb{R} . La somme des flots $f_1 + f_2$ est une fonction de $V \times V$ vers \mathbb{Z} définie de la façon suivante : $(f_1 + f_2)(u, v) = f_1(u, v) + f_2(u, v)$

La valeur du flot pour une fonction de flot f est définie comme :

$$|f| = \sum_{v \in V} f(s, v) = \sum_{u \in V} f(u, t)$$

Il s'agit du flot total sortant du nœud source, lequel est égal au flot total rentrant dans le nœud puits.

La figure 3.10 montre un graphe de flot et les valeurs de flot et de capacité de chaque arc (flot/capacité). Par exemple, pour l'arc (c, a) , la valeur du flot est de $f(c, a) = 1$, tandis que la capacité de l'arc est $c(c, a) = 3$. Dans cet exemple la valeur du flot total est de 2 car :

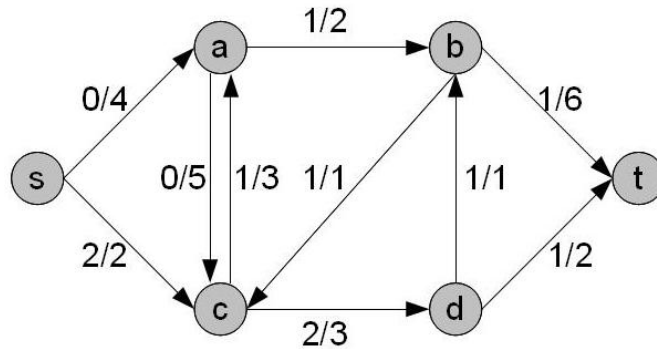


FIGURE 3.10 – Graphe de flot et les valeurs de flot et de capacité de chaque arc (flot/capacité).

$$|f| = \sum_{v \in V} f(s, v) = f(s, a) + f(s, c) = 0 + 2 = 2$$

Le problème du flot maximum est celui de trouver une valeur de flot maximum pour un graphe de flot $G = (V, E)$ donné.

3.2.2.1 Méthode Ford Fulkerson

Plusieurs algorithmes s'adressant au problème du flot maximum utilisent la méthode de Ford Fulkerson proposée en 1956 [Ford & Fulkerson, 1956]. Cette méthode est basée sur les notions de chemin d'augmentation et de graphe résiduel ou graphe d'écart. L'algorithme débute à partir d'une fonction de flot valide (satisfaisant les contraintes de capacité, antisymétrie et conservation du flot) à partir de laquelle des chemins d'augmentation sont recherchés. Un chemin d'augmentation peut être vu intuitivement comme une chaîne allant du nœud source au nœud puits et à travers laquelle il est possible d'augmenter la valeur du flot tout en respectant les contraintes de capacité de chaque arc du chemin et les contraintes de conservation de flot à chaque nœud du chemin. La figure 3.11 montre le fonctionnement général des algorithmes basés sur la méthode de Ford Fulkerson. Dans le cas des fonctions de flot à valeurs entières les algorithmes de type Ford Fulkerson garantissent l'obtention d'une solution.

Graphe résiduel La recherche d'un chemin d'augmentation est une des procédures principales d'un algorithme de type Ford Fulkerson. Cette recherche se base sur les notions de capacité résiduelle et de graphe résiduel. Pour un graphe de flot et une fonction de flot f , la capacité résiduelle d'un arc est définie de la manière suivante : $c_f(u, v) = c(u, v) - f(u, v)$

Cette capacité indique la quantité de flot maximum qui peut être ajouté à l'arc concerné. Il faut remarquer que lorsque le flot $f(u, v)$ est négatif, l'ajout du flot dans

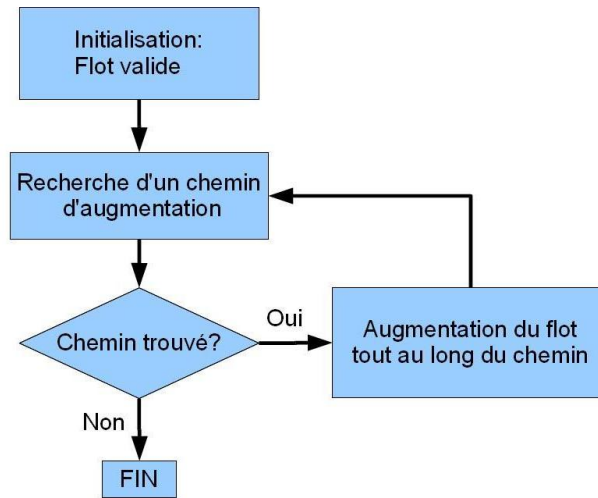


FIGURE 3.11 – Fonctionnement général des algorithmes basés sur la méthode de Ford Fulkerson.

le sens de l'arc (u, v) revient à retirer du flot dans le sens de l'arc (v, u) . Alors la capacité résiduelle $c_f(u, v)$ peut être supérieure à la capacité $c(u, v)$, car elle sera égale à la quantité de flot qui peut être retirée dans le sens de l'arc (v, u) plus la quantité de flot qui peut traverser l'arc (u, v) .

Pour un graphe de flot $G = (V, E)$ avec une fonction de capacité c , le graphe résiduel de G induit par la fonction de flot f est $G_f = (V, E_f)$ avec : $E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$ Les arcs appartenant à E_f sont soit des arcs appartenant à E soit ses arcs opposés. Le graphe résiduel G_f est aussi un graphe de flot avec des capacités données par la fonction c_f .

La figure 3.12 montre le graphe résiduel correspondant au graphe de flot et à la fonction de flot de la figure 3.10. Dans le cas des arcs ayant une valeur de flot nulle, la capacité résiduelle reste égale à la capacité de l'arc :

$$c_f(s, a) = c(s, a) - f(s, a) = 4 - 0 = 4$$

$$c_f(a, s) = c(a, s) - f(a, s) = 0 - 0 = 0$$

Dans le cas des arcs avec une valeur de flot positive et inférieur à leur capacité, la capacité résiduelle diminue dans le sens de l'arc, mais elle augmente dans le sens opposé :

$$c_f(d, t) = c(d, t) - f(d, t) = 2 - 1 = 1$$

$$c_f(t, d) = c(t, d) - f(t, d) = 0 - (-1) = 1$$

Dans le cas des arcs avec une valeur de flot positive et égale à la capacité de l'arc, la capacité résiduelle devient nulle dans le sens de l'arc, mais elle augmente dans le sens opposé :

$$c_f(s, c) = c(s, c) - f(s, c) = 2 - 2 = 0$$

$$c_f(c, s) = c(c, s) - f(c, s) = 0 - (-2) = 2$$

Dans le cas des arcs avec une valeur de flot négative, la capacité résiduelle est supérieure à la capacité de l'arc :

$$c_f(a, c) = c(a, c) - f(a, c) = 5 - (-1) = 6$$

$$c_f(c, a) = c(c, a) - f(c, a) = 3 - 1 = 2$$

L'utilité de la notion de graphe résiduel repose sur la propriété suivante : soit $G = (V, E)$ un graphe de flot avec un nœud source s , et un nœud puits t . Soit f une fonction de flot dans G . Soit G_f le graphe résiduel de G induit par f , et soit f' une fonction de flot dans G_f . Alors la somme des fonctions de flot $f + f'$ est une fonction de flot valide dans G avec $|f + f'| = |f| + |f'|$.

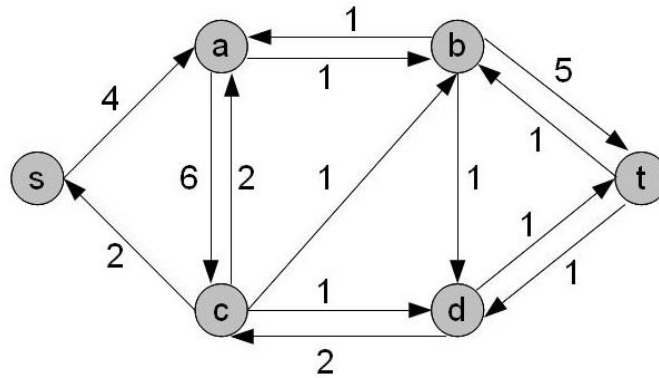


FIGURE 3.12 – Graphe résiduel correspondant au graphe et à la fonction de flot de la figure 3.10.

Chemin d'augmentation Soit $G = (V, E)$ un graphe de flot avec un nœud source s , et un nœud puits t , et soit f une fonction de flot dans G . Un chemin d'augmentation est défini comme un chemin de s vers t dans le graphe résiduel G_f . Chaque arc (u, v) du chemin d'augmentation admet au moins une quantité positive additionnelle de flot de u vers v tout en respectant la contrainte de capacité $c(u, v)$.

La capacité résiduelle minimum des arcs du chemin donne la quantité maximum de flot admissible à chaque arc du chemin et qui respecte la contrainte de conservation de flot. Cette capacité résiduelle minimum est appelée la capacité résiduelle de p . Elle est définie de la manière suivante : $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ est dans } p\}$

Formellement, soit $G = (V, E)$ un graphe de flot, soit f une fonction de flot dans G , et soit p un chemin d'augmentation dans G_f . Soit une fonction $f_p : V \times V \rightarrow \mathbb{Z}$ telle que :

$$f_p(u, v) = \begin{cases} c_f(p) & \text{si } (u, v) \text{ est dans } p \\ -c_f(p) & \text{si } (v, u) \text{ est dans } p \\ 0 & \text{si } (u, v) \text{ ou } (v, u) \text{ n'est pas dans } p \end{cases}$$

Alors f_p est une fonction de flot dans G_f de valeur $|f_p| = c_f(p) > 0$.

De cette propriété nous pouvons déduire le corolaire suivant :

Soit $G = (V, E)$ un graphe de flot, soit f une fonction de flot dans G , et soit p un chemin d'augmentation dans G_f . Soit $f' : V \times V \rightarrow \mathbb{Z}$ avec $f' = f + f_p$. Alors f' est un flot dans G de valeur $|f'| = |f| + |f_p| > |f|$.

Ceci montre que chaque fois qu'un chemin d'augmentation est trouvé, il est possible d'augmenter le flot du graphe. Lorsqu'il n'existe pas de chemin d'augmentation pour une fonction de flot f , alors la valeur de flot de cette fonction $|f|$ est la valeur de flot maximum du graphe de flot.

L'algorithme 5 présente le pseudo-code d'un algorithme Ford Fulkerson.

Algorithme 5 Ford-Fulkerson(G, s, p)

```

1: pour chaque arc  $(u, v) \in E$  faire
2:    $f[u, v] \leftarrow 0$ 
3:    $f[v, u] \leftarrow 0$ 
4: tant que il existe un chemin  $p$  de  $s$  à  $t$  dans le graphe résiduel  $G_f$  faire
5:    $c_f(p) \leftarrow \min c_f(u, v) : (u, v) \text{ est dans } p$ 
6:   pour chaque arc  $(u, v)$  dans  $p$  faire
7:      $f[u, v] \leftarrow f[u, v] + c_f(p)$ 
8:      $f[v, u] \leftarrow -f[u, v]$ 

```

Les lignes 1-3 initialisent la valeur du flot à zéro. Néanmoins une fonction de flot valide déjà connue peut être utilisée pour réduire le temps d'exécution de l'algorithme. La ligne 4 cherche un chemin d'augmentation. Si un chemin est trouvé, la ligne 6 détermine la valeur de la capacité résiduelle du chemin, et les lignes 7-8 vont rajouter du flot égal à cette valeur de capacité résiduelle tout au long du chemin d'augmentation. La boucle 4-8 s'arrête lorsqu'il est impossible de trouver un nouveau chemin d'augmentation. La fonction de flot résultant après l'application de cette méthode a une valeur de flot maximum, mais elle n'est pas unique. Plusieurs fonctions de flot différentes peuvent avoir la même valeur de flot.

Plusieurs algorithmes différents peuvent être utilisés pour trouver des chemins d'augmentation, donnant différents temps d'exécution à l'algorithme. Dans le cas général, considérant des chemins d'augmentation trouvés au hasard, le temps asymptotique de l'algorithme est de $O(|E| \times |f_{max}|)$ avec $|f_{max}|$ étant la valeur de flot maximum du graphe de flot. Ceci parce que, dans le pire des cas, chaque chemin augmente le flot d'une valeur de un. Dans ce cas $|f_{max}|$ recherches de chemins sont nécessaires pour atteindre la valeur de flot maximum, et chaque recherche de chemin prend un temps $O(|E|)$.

L'algorithme donne, en plus de la valeur de flot maximum pour le graphe considéré, une fonction de flot maximum, laquelle n'est pas forcément unique.

Exemple La figure 3.13 montre l'application de la méthode de Ford-Fulkerson sur un graphe de flot avec un nœud source s et un nœud puits p . À gauche : le graphe de flot et les valeurs de flot et capacité de chaque arc (flot/capacité). À droite : le graphe résiduel correspondant, avec les capacités résiduelles de chaque arc. À chaque itération un chemin d'augmentation est trouvé sur le graphe résiduel. Le flot total

est augmenté d'une valeur égale à la capacité résiduelle minimale des arcs du chemin d'augmentation. Pour chaque itération montrée dans la figure nous avons les valeurs de flot total suivantes :

- a) flot total = 0
- capacité du chemin d'augmentation = 2
- b) flot total = 2
- capacité du chemin d'augmentation = 2
- c) flot total = 4
- capacité du chemin d'augmentation = 1
- d) flot total = 5

Lorsqu'il est impossible de trouver un chemin d'augmentation entre le nœud source et le nœud puits, le flot maximum a été atteint.

3.2.3 Problème du nombre maximum de chemins disjoints

Dans le chapitre suivant, la problématique des alignements présentée au chapitre 2 sera formalisée en termes de théorie des graphes. Le problème mathématique résultant de cette formalisation est similaire au problème du nombre maximum de chemins disjoints. Ce problème étant très étudié dans la bibliographie, il nous a semblé important pour la compréhension du lecteur de le définir formellement.

Soit $G = (V, E)$ un graphe avec un ensemble de nœuds V , et un ensemble d'arcs E . Soient T un ensemble de k paires de nœuds (s_i, p_i) avec $(1 \leq i \leq k)$. Le problème consiste à trouver le sous-ensemble maximum de T , tel qu'il existe un ensemble disjoint de chaînes ou chemins disjoints P_1, \dots, P_m , avec P_i connectant le nœud s_i au nœud p_i .

Ce problème peut être divisé en deux problèmes différents dépendant de la prise en compte des chaînes (chemins) disjointes par les arcs, ou par les nœuds.

3.3 Conclusion

Dans ce chapitre nous avons présenté les bases théoriques à partir desquelles ont été développées nos contributions. Dans le chapitre suivant, les termes et méthodologies présentées seront utilisées et supposées connues par le lecteur.

Les concepts, propriétés et les différentes représentations des graphes seront utiles pour comprendre la modélisation des réseaux de canalisation sous forme de graphe non orienté, et de graphe de flot. En particulier, les notions de chaîne, chemin, chemin disjoint, sous-graphe, et les propriétés de connexité serviront à décrire ces modèles et le lien avec la problématique des alignements. Les listes d'adjacence ont été les structures de données choisies pour représenter ces graphes.

Les algorithmes de Dijkstra, pour le problème du plus court chemin, et de Ford Fulkerson, pour le problème du flot maximum, vont faciliter la compréhension des nouveaux algorithmes présentés. En particulier, la notion de coûts estimés pour le premier, et les contraintes de conservation de flot, de capacité, et des flots à valeurs entières pour le deuxième.

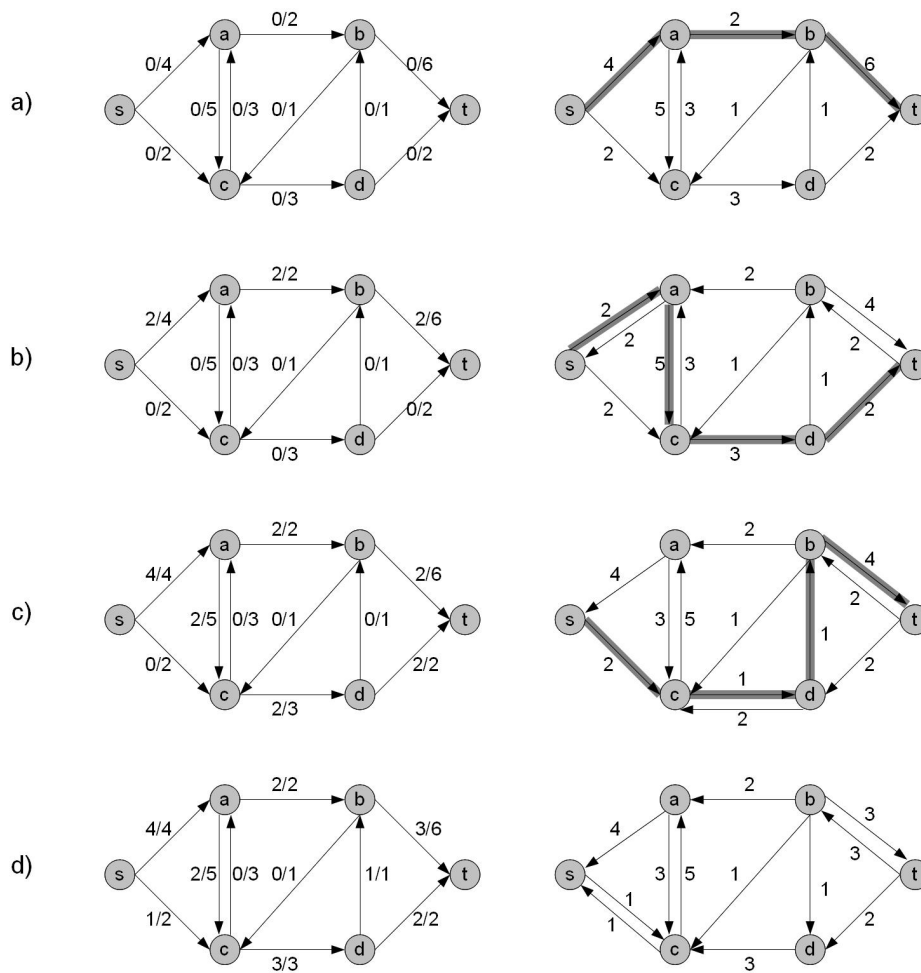


FIGURE 3.13 – Application de la méthode de Ford-Fulkerson à un graphe de flot avec un nœud source 's' et un nœud puits 'p'. A gauche : le graphe de flot et les valeurs de flot et de capacité de chaque arc (flot/capacité). A droite : le graphe résiduel correspondant avec les capacités résiduelles de chaque arc et le chemin d'augmentation utilisé.

Contributions

Résumé : Ce chapitre présente les principales contributions de cette de thèse. Nous présentons un modèle permettant d'intégrer de manière précise les éléments du système pertinents pour la problématique du chapitre 2. Nous proposons une définition de la capacité opérative pour les systèmes et les opérations considérées dans nos travaux, et nous présentons un algorithme permettant de calculer la capacité opérative. C'est, à notre connaissance, la première fois qu'une définition formelle de la capacité opérative est proposée dans le cas des réseaux de transport. À la fin du chapitre, des développements complémentaires sont présentés.

Sommaire

4.1	Modélisation des réseaux de canalisation sous forme de graphe non orienté	51
4.2	Indicateurs de Capacité Opérative	55
4.2.1	Relation d'équivalence entre des éléments du réseau	56
4.2.2	Définition	57
4.3	Algorithme de calcul de la Capacité Opérative	58
4.3.1	Modèle de graphe de flot pour des chaînes disjointes	59
4.3.2	Algorithme du flot maximum	64
4.3.3	Validation	65
4.3.4	Capacité Opérative pour des sous-types d'alignement	66
4.3.5	Ensemble d'indicateurs de Capacité Opérative	70
4.4	Développements complémentaires	72
4.4.1	Analyse de criticité	72
4.4.2	Algorithme de recherche de chemin maximisant la capacité opérative	77
4.4.3	Opérations d'alignements forcés	84
4.5	Conclusion	90

4.1 Modélisation des réseaux de canalisation sous forme de graphe non orienté

L'optimisation des opérations de transport présentées dans le chapitre 2 repose sur un modèle structurel permettant de tenir compte des éléments et des événements pertinents pour la problématique posée. Dans cette section nous présentons

un modèle pour les réseaux de canalisation sous forme de graphe non orienté. Les vannes et les pompes étant les seuls éléments qui peuvent être commandés, dans le réseau considéré. Ces éléments vont être la base de notre modélisation.

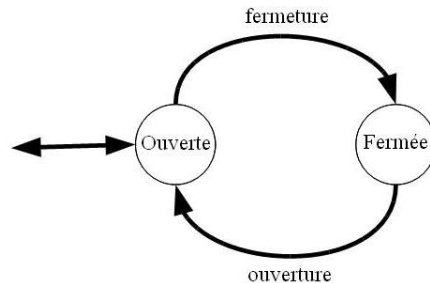


FIGURE 4.1 – Modèle d’automate à états d’une vanne du réseau.

Un modèle est une version simplifiée de la réalité qui conserve convenablement des propriétés particulières de cette réalité. Le modèle de graphe non orienté proposé repose sur les simplifications suivantes :

- **Alignements sans pompe** : dans un premier temps, les alignements avec des pompes ne sont pas considérés. À la fin du chapitre, un algorithme de recherche d’alignement avec pompe sera présenté (alignement dit forcé).
- **Sens du flux de pétrole** : dans les réseaux considérés, le pétrole peut traverser la plupart des canalisations dans les deux sens. Dans le cas des pompes, le pétrole ne peut circuler que dans un sens imposé, cependant les alignements avec des pompes ne sont pas considérés dans un premier temps. D’autres éléments, tels que les oléoducs d’entrée, les oléoducs de sortie, les bras de chargement et les réservoirs desservant une raffinerie, imposent souvent un sens unique de circulation du pétrole. Mais comme ce sont des éléments terminaux, le sens de circulation n’est pas pris en compte dans la modélisation mais sera imposé par les algorithmes de recherche de chemin utilisés. De plus cela permet d’utiliser le même modèle pour tester des situations extraordinaires, comme l’utilisation d’éléments normalement fournisseurs en tant que receveurs, et vice-versa.
- **Dynamique discrète des vannes** : les vannes sont, la plupart du temps, soit ouvertes soit fermées. Les transitions entre ces états, c’est à dire la fermeture et l’ouverture des vannes, peuvent être considérées comme instantanées par rapport à la durée de chaque état. Le pourcentage d’ouverture de la vanne n’est pas considéré, l’état “ouverte” correspond à tout pourcentage d’ouverture de la vanne permettant le flux du pétrole. L’état initial des vannes peut être “ouverte” aussi bien que “fermée”, il dépendra des opérations précédentes à l’instant considéré. En général, l’état initial des vannes n’a pas d’incidence sur la problématique abordée. Les défaillances ne sont pas considérées dans cette description. La figure 4.1 montre un modèle d’automate à états représentant

le comportement discret d'une vanne quelconque du réseau.

Un réseau de canalisation contenant des vannes peut être modélisé sous forme de graphe non orienté en se basant sur l'analogie existant entre une vanne qui connecte deux portions d'un réseau, et un arc qui connecte deux nœuds dans un graphe. Le pétrole pouvant traverser les vannes dans les deux sens, les arcs qui le représentent doivent être de type non orienté (arête). Les vannes étant représentées par des arcs, il faut définir les objets représentés par les nœuds. Dans cette mémoire, ces objets sont appelés des "segments" [Bentley WaterCad Inc, 2010].

Segment : un segment est défini comme une portion du réseau limité par des vannes, ne contenant pas de vannes à l'intérieur.

Les segments sont les plus petites portions du réseau qui peuvent être isolées. La figure 4.2 montre un segment dans un réseau de canalisation. Un segment peut contenir un réservoir, une pompe, un bras de chargement, ou correspondre à un oléoduc qui connecte le réseau à un autre système de transport hors du contrôle des opérateurs. Les segments correspondant à des réservoirs, des bras de chargement ou à des oléoducs, seront modélisés par des nœuds terminaux. Un multiple sera modélisé par des nœuds connectés à plusieurs arcs.

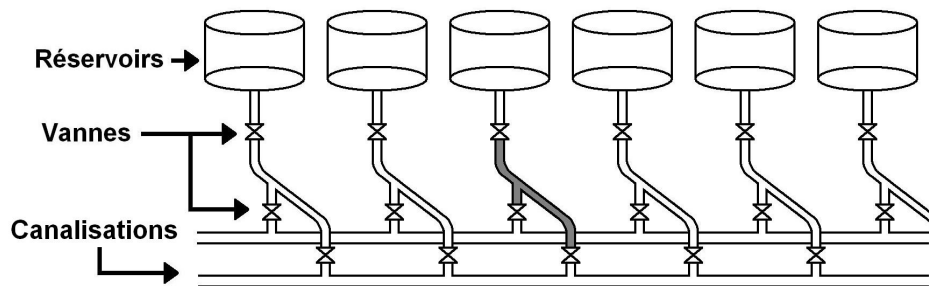


FIGURE 4.2 – En gris, un segment du réseau.

La notion de segment va permettre de modéliser le réseau faisant abstraction des longueurs et diamètres des canalisations, et du placement des différents éléments dans l'espace. Seules les possibles connexions au travers des vannes sont prises en compte.

La construction du modèle de graphe non orienté comprend les étapes suivantes :

- Création des nœuds du graphe : pour chaque segment du réseau ne contenant pas de pompe, un nœud n est rajouté au graphe non orienté.
- Rajout des arêtes au graphe : pour chaque vanne du réseau connectant deux segments du réseau, une arête (u, v) est rajoutée au réseau, u et v étant les nœuds correspondants aux segments connectés par cette vanne.

La figure 4.3 montre le modèle de graphe non orienté construit à partir d'un réseau de canalisation contenant 15 vannes. Les réservoirs (nœuds : a , b et c) et les bras de chargement (nœuds : l , m et n) sont dans des segments qui sont connectés

chacun à une seule vanne. Alors ses nœuds respectifs sont seulement connectés à un arc chacun. Deux lignes de chargement (nœuds : g et h) connectent la zone des réservoirs à la zone des bras de chargement. Les segments supérieurs (nœuds : d , e et f) connectent les réservoirs aux lignes de chargement, et les segments inférieurs (nœuds : i , j et k) connectent les lignes de chargement aux bras de chargement.

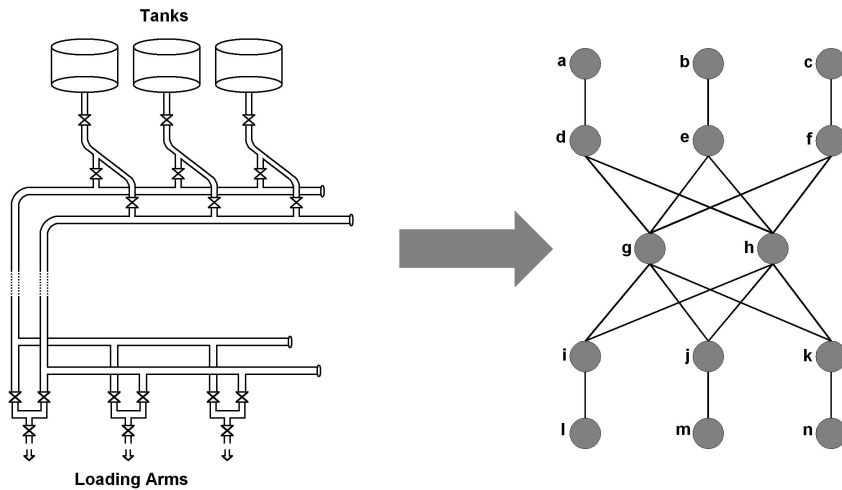


FIGURE 4.3 – Modélisation sous forme de graphe non orienté d’un réseau de canalisation contenant des vannes.

La modélisation sous forme de graphe non orienté permet de traduire de façon simple une route (chaîne, chemin) calculée à partir du graphe dans un ensemble d’instructions compréhensibles. Un alignement entre deux éléments sera représenté par une chaîne élémentaire dans le graphe. Les arcs contenus dans cette chaîne vont correspondre aux vannes qu’il faudra ouvrir, tandis que les arcs adjacents à la chaîne vont correspondre aux vannes qu’il faudra fermer, tout cela avec le but de permettre le transport de pétrole entre les deux éléments. Par exemple, la chaîne (a, d, h, j, m) qui à la figure 4.3 permet l’alignement du premier réservoir avec le deuxième bras de chargement peut s’établir si et seulement si les vannes correspondant aux arcs (a, d) , (d, h) , (h, j) , (j, m) sont ouvertes, et les vannes correspondant aux arcs (d, g) , (e, h) , (f, h) , (h, i) , (h, k) , (g, j) sont fermées.

Dans le chapitre 6 nous allons voir que ce modèle peut être généralisé pour le cas des “réseaux d’interrupteurs”.

Structure de données

Le choix d’une structure de données pour représenter des entités mathématiques est très important pour obtenir un fonctionnement efficace des algorithmes qui traitent ces données. Dans le cas des graphes, les matrices et les listes sont les structures de données les mieux adaptées pour les représenter. Parmi les types de structures pouvant représenter des graphes, présentées dans le chapitre précédent,

nous avons choisi la liste d'adjacence. Ce choix est dû à la recherche rapide des nœuds adjacents que ce type de structure permet. La recherche des nœuds adjacents est une sous-routine essentielle pour des algorithmes de recherche de chemins, lesquels sont la base des algorithmes que nous avons développés. Par contre, pour les graphes peu denses ($|E| < |V|^2$, $|E|$ et $|V|$ étant respectivement le nombre d'arcs et le nombre de nœuds) que nous étudions, une liste d'adjacence est moins compacte que d'autres structures de données comme les liste d'arêtes. Cependant, la nécessité de sous-routines rapides pour des algorithmes avec des temps d'exécution qui peuvent devenir assez grands, et la taille relativement réduite, en termes d'espace mémoire, des réseaux, fait de la liste d'adjacence la représentation la plus adaptée pour nos travaux.

La liste d'adjacence suivante correspond au graphe de la figure 4.3 :

$a : (d)$	$h : (d, e, f, i, j, k)$
$b : (e)$	$i : (g, h, l)$
$c : (f)$	$j : (g, h, m)$
$d : (a, g, h)$	$k : (g, h, n)$
$e : (b, g, h)$	$l : (i)$
$f : (c, g, h)$	$m : (j)$
$g : (d, e, f, i, j, k)$	$n : (k)$

Le nombre d'éléments dans la liste d'adjacence de chaque nœud donne le nombre de segments connectés au segment correspondant, et donne aussi le nombre de vannes environnantes permettant d'isoler ce segment.

4.2 Indicateurs de Capacité Opérative

D'après la problématique présentée dans le chapitre 2, le critère principal de choix d'un alignement est celui qui minimise les possibles interférences avec d'autres opérations, c'est-à-dire, avec d'autres possibles alignements. Ce problème nous amène à définir des indicateurs permettant de mesurer la quantité d'alignements simultanés possibles dans le réseau de canalisation ; quantité que nous avons appelée "Capacité Opérative".

Dans le langage courant, la notion de capacité opérative est comprise comme la capacité d'un système à soutenir des opérations. Dans le système considéré, ces opérations correspondent principalement à des alignements entre un ensemble d'éléments délivrant du pétrole, et un ensemble d'éléments recevant du pétrole. Dans la plupart des cas, chaque alignement est un chemin entre deux éléments du réseau de canalisation, et ce chemin est isolé du reste du réseau. Des alignements entre plus de deux éléments sont moins fréquents. Dans cette définition, nous allons considérer uniquement des alignements entre deux éléments.

4.2.1 Relation d'équivalence entre des éléments du réseau

Très fréquemment, plusieurs éléments terminaux du réseau peuvent accomplir la même fonction, et satisfaire de la même manière les spécifications d'une demande de pétrole. Par exemple, deux réservoirs vides de même capacité peuvent être utilisés pour exactement la même tâche de stockage de pétrole. Deux réservoirs contenant le même type de pétrole peuvent être utilisés pour exactement la même tâche de fourniture de pétrole à un tanker ou à une raffinerie. Deux bras de chargement placés sur des quais permettant d'accueillir des tankers d'une certaine capacité, peuvent être utilisés pour réaliser exactement la même tâche de livraison de pétrole à un tanker. Nous pouvons dire que, pour des spécifications données, deux alignements sont de même type si et seulement si ils satisfont toutes ces spécifications. Alors nous pouvons définir une relation d'équivalence entre les éléments du réseau pouvant accomplir la même tâche : pour un type d'alignement donné, deux éléments sont considérés équivalents si et seulement si ils peuvent être utilisés pour ce type d'alignement dans la même fonction, soit de fourniture soit de réception de pétrole. Deux éléments sont dits de même type si et seulement s'ils sont équivalents pour un type d'alignement donné.

La relation d'équivalence entre des éléments du réseau dépend des spécifications demandées pour un alignement. Il faut remarquer que selon le type d'analyse envisagé par les opérateurs et l'information pertinente disponible concernant les alignements à réaliser, les spécifications peuvent être plus ou moins contraignantes.

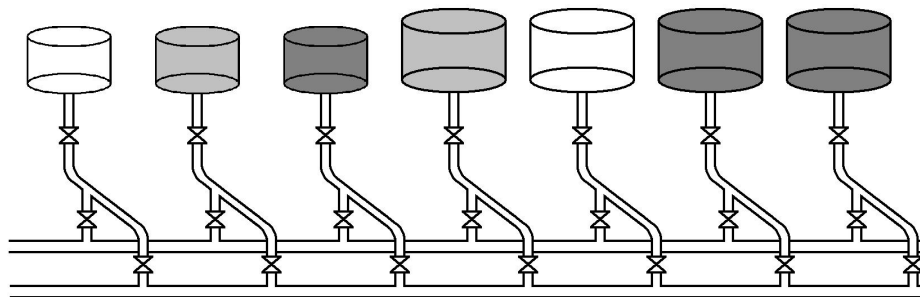


FIGURE 4.4 – Section d'un réseau de transport de pétrole. Les couleurs des réservoirs indiquent leur contenu : blanc : réservoir vide, gris claire : réservoir contenant du pétrole léger, gris foncé : réservoir contenant du pétrole lourd. La taille d'un réservoir indique sa capacité.

La figure 4.4 présente une section d'un réseau de transport de pétrole avec plusieurs réservoirs. Les couleurs des réservoirs indiquent leur contenu. La taille d'un réservoir indique sa capacité. Le tableau 4.1 montre des exemples des relations d'équivalence avec différents degrés de précision pour les réservoirs de la figure 4.4.

Les spécifications vont déterminer la relation d'équivalence entre les différents éléments du réseau. Le degré de description des spécifications peut réduire le nombre

Type d'élément (spécification)	Éléments équivalents dans la figure	Degré de description
Réservoir	Tous les réservoirs.	Très général
Réservoir à grande capacité	4eme, 5eme, 6eme et 7eme réservoir.	Général
Réservoir vide	1er et 5eme réservoir.	Général
Réservoir contenant du pétrole léger	2eme et 4eme réservoir.	Spécifique
Réservoir à grande capacité contenant du pétrole lourd	6eme et 7eme réservoir.	Très spécifique

TABLE 4.1 – Relations d'équivalence entre des réservoirs d'un réseau de canalisation

d'éléments équivalents.

Par exemple, si le contenu des réservoirs et les demandes des tankers sont connus, seulement les réservoirs contenant le même type de pétrole seront considérés équivalents. Dans ce cas, la capacité opérative va indiquer exactement le nombre d'alignements pouvant être établis. Par contre, si le contenu des réservoirs ou les demandes des tankers sont inconnues (c'est le cas des opérations prévues à moyen terme) alors tous les réservoirs peuvent être considérés équivalents. Dans ce cas la capacité opérative donne une limite maximale du nombre d'alignements possibles, ce qui est un estimé de la capacité opérative réelle, et qui peut diminuer lorsque plus d'information devient disponible.

Il se peut aussi que l'opérateur veuille connaître la capacité opérative pour des types d'alignements avec différents degrés de précision (des spécifications plus ou moins contraignantes).

4.2.2 Définition

Soit N un réseau de transport de pétrole. Soit T un type d'alignement, satisfaisant un ensemble S de spécifications, connectant un élément de type F avec un élément de type R . À l'instant t , nous définissons la capacité opérative de N pour T comme le nombre maximal d'alignements simultanés de ce type qui peuvent être établis entre les ensembles d'éléments $A = \{a : a \text{ est de type } F\}$ et $B = \{b : b \text{ est de type } R\}$.

Par exemple, pour des alignements entre réservoirs et quais, une capacité opérative de cinq indique qu'il est possible d'établir au même temps cinq alignements entre un réservoir et un quai dans le réseau considéré.

Dans le cas de la figure 4.3, nous pouvons considérer une demande avec les spécifications suivantes :

- Récepteur : tanker
- Type de pétrole : Leona
- Quantité de pétrole : 250.000 barils

Supposons que tous les réservoirs de la figure 4.3 contiennent le type de pétrole demandé et ont une capacité supérieure ou égale à 250.000 barils. Ces spécifications peuvent être satisfaites par tout alignement connectant un réservoir à un tanker. Alors l'alignement demandé est de type réservoir/bras de chargement. La capacité opérative du réseau pour un type d'alignement réservoir/bras de chargement est égale au nombre maximal d'alignements simultanés entre un réservoir et un bras de chargement. Dans cet exemple simple nous pouvons voir que la capacité opérative pour ce type d'alignement est de 2, car il est impossible de connecter plus de deux réservoirs avec plus de deux bras de chargement sans interférence.

Une définition précise de la capacité opérative, dans le cadre des réseaux de transport de pétrole, est une des principales contributions de ces travaux. Cette définition permettra le calcul de la valeur de la capacité opérative et l'utilisation de cette valeur pour des buts pratiques. Nous proposons un algorithme permettant de calculer la capacité opérative du système pour un opération donnée.

4.3 Algorithme de calcul de la Capacité Opérative

La définition de la Capacité Opérative que nous venons d'établir n'est utile que s'il est possible de calculer les valeurs de capacité opérative correspondant aux opérations prévues dans le réseau considéré. À chaque type d'opération correspond une valeur de capacité opérative différente. Pour un même réseau, les valeurs de capacité opérative pour chaque type d'alignement sont appelées "indicateurs de capacité opérative". L'ensemble de ces indicateurs donne une idée concrète sur la capacité opérative du système. Nous allons présenter un algorithme permettant de calculer les indicateurs de capacité opérative. Cet algorithme considère des alignements connectant deux éléments du système. Dans ces alignements le pétrole est transporté soit par gravité, l'élément délivrant du pétrole étant placé à un niveau plus haut que l'élément recevant du pétrole, soit par la pression produite par des pompes localisées hors du système (et hors du contrôle direct des opérateurs), comme dans le cas des oléoducs. Les segments contenant des pompes ne sont pas considérés, et les nœuds du graphe correspondant à ces segments, s'il existent, doivent être supprimés, avant l'exécution de l'algorithme. Les opérations nécessitant l'exploitation de pompes appartenant au système seront étudiées à la fin du chapitre.

Dans le chapitre précédent, nous avons décrit le problème du nombre maximum de chaînes ou chemins disjoints (nœud-disjoints) entre des paires de nœuds donnés. Ici, ce problème est relaxé, à cause de la relation d'équivalence entre des éléments du système, laquelle peut être vue comme une redondance. Au lieu de rechercher des chaînes entre des paires de nœuds spécifiques du réseau, le problème revient à rechercher des chaînes entre des nœuds quelconques dans deux ensembles de nœuds spécifiques. Chaque chaîne connecte ces deux ensembles. Les chaînes correspondent à des routes d'alignements de même type, ou équivalents, pour des spécifications données. Alors le problème de calcul de la capacité opérative peut être défini de façon mathématique d'une manière similaire au problème du nombre maximum de

chaînes (ou chemins) disjointes. Définissons le “problème du nombre maximum de chaînes disjointes équivalentes” (PMCDE) :

Soit $G = (V, E)$ un graphe avec un ensemble de nœuds V , et un ensemble d’arcs E . Soient A et B deux ensembles de nœuds inclus dans V . Le PMCDE consiste à trouver le nombre maximum de chaînes disjointes par les nœuds, tels que chaque chaîne connecte un nœud appartenant à A avec un nœud appartenant à B . Les chaînes sont disjointes, chacune connectant des paires de nœuds différents.

La capacité opérative définie précédemment correspond alors au nombre maximum de chaînes disjointes équivalentes dans un graphe représentant le réseau de canalisation considéré. Chaque chaîne connecte des ensembles de nœuds correspondant aux éléments considérés comme équivalents pour l’opération prévue.

L’algorithme permettant de calculer le nombre maximum de chaînes disjointes équivalentes est présent dans la bibliographie concernant la recherche de l’ensemble maximal de chemins disjointes par les nœuds entre deux nœuds d’un graphe [Leeuwen, 1998]. Nous allons présenter une version plus générale de cet algorithme afin d’augmenter la versatilité de la modélisation, permettant l’ajout de spécifications particulières à notre problématique. L’algorithme est basé sur des algorithmes de flot maximum, et en particulier sur la méthode de Ford Fulkerson présentée dans le chapitre 3. L’algorithme peut être subdivisé en deux étapes : la première étape consiste à construire un réseau de flot à partir du modèle de graphe non orienté du réseau de canalisation, tandis que la deuxième étape consiste à l’utilisation d’un algorithme de flot maximum classique.

4.3.1 Modèle de graphe de flot pour des chaînes disjointes

Le PMCDE peut être regardé en termes de flot entre un nœud source et un nœud puits. Cependant il existe des différences importantes entre le problème classique de flot maximum et le PMCDE. Le problème du flot maximum s’adresse à des graphes orientés particuliers appelés des graphes de flot, dans lesquels à chaque arc correspond une capacité discrète maximale, et dans lesquels il est possible, a priori, d’avoir des convergences et des divergences de flot dans les nœuds. Le PMCDE, par contre, s’adresse à des graphes non orientés, dans lesquels les capacités de flux de pétrole ne sont pas considérées, et dont les chaînes sont disjointes. En effectuant des transformations successives au graphe non orienté présenté précédemment, il est possible de construire un graphe de flot incorporant les contraintes spécifiques du PMCDE, et dans lequel le flot représente les chaînes connectant les différents éléments du graphe non orienté (et donc les routes des alignements du réseau). Ces transformations permettront l’utilisation d’un algorithme de flot maximum classique pour déterminer le nombre maximum de chaînes disjointes équivalentes recherché.

La figure 4.5 montre les transformations nécessaires pour construire le graphe de flot à partir du modèle de graphe non orienté du réseau. Ces transformations correspondent à la première partie de l’algorithme de calcul de la capacité opérative. L’idée est de construire un graphe de flot, dans lequel le flot correspondra à des chaînes disjointes dans le graphe non orienté. Alors il faut interdire les convergences

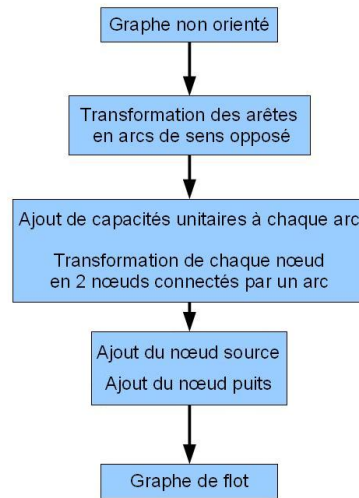


FIGURE 4.5 – Transformations permettant de construire un graphe de flot à partir du modèle de graphe non orienté, tenant compte des contraintes des alignements.

et les divergences de flot dans le graphe, hormis les nœuds source et puits.

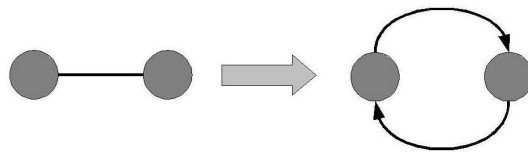


FIGURE 4.6 – Transformation d'un arc non orienté en deux arc orientés opposés.

La première transformation est assez simple, elle consiste à remplacer chaque arc non orienté du graphe par deux arcs orientés de sens opposé (figure 4.6).

La deuxième transformation permet d'interdire des chemins (correspondant à des chaînes dans le graphe non orienté) qui ne sont pas disjoints. Dans le problème que nous abordons, les chemins vont être représentés par le flot traversant le graphe. Les chemins étant disjoints, le flot ne doit pas avoir de divergences ni de convergences dans le graphe, sauf, comme nous verrons, dans le cas du nœud source et du nœud puits. Autrement dit, chaque nœud aura au plus deux arcs adjacents avec des flots différents de zéro. Or, la somme des flots incidents est égale à la somme des flots sortants d'un nœud (conservation du flot), donc chaque nœud aura un seul arc incident et un seul arc sortant avec des flots différents de zéro. Cette contrainte sera incorporée au graphe en deux étapes.

Premièrement, nous allons assigner des capacités unitaires à chaque arc du graphe. Pour les flots à valeurs entières d'un graphe de flot, ces capacités d'une unité interdisent les convergences et les divergences de flot pour tout nœud ayant

seulement un arc incident, et pour tout nœud ayant seulement un arc sortant (figure 4.7). La preuve est trivialement obtenue à partir de la propriété de conservation du flot des graphes de flot.

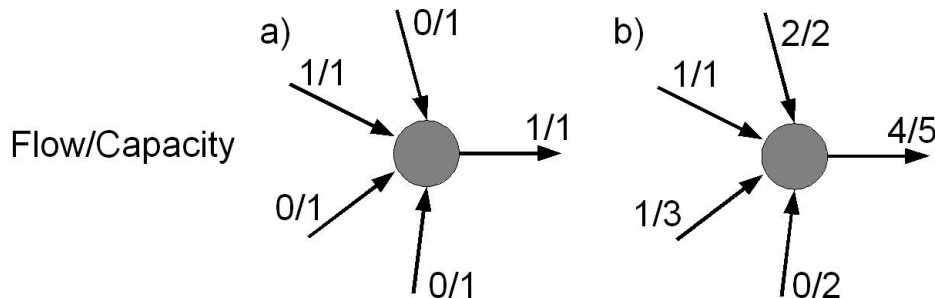


FIGURE 4.7 – Pour des nœuds avec seulement un arc de sortie, l’assignation de capacités unitaires aux arcs interdit les convergences de flot (a), tandis que des capacités supérieures permettent des convergences du flot (b).

Deuxièmement, nous allons remplacer chaque nœud par deux nœuds, le premier, que nous appelons “nœud d’entrée” (écrits avec un exposant ‘-’) sera connecté aux arcs entrant du nœud original, tandis que le deuxième, que nous appelons “nœud de sortie” (écrits avec un exposant ‘+’), sera connecté aux arcs sortant du nœud original. Ces deux nouveaux nœuds seront connectés par un arc (de capacité unitaire) allant du nœud d’entrée vers le nœud de sortie (figure 4.8).

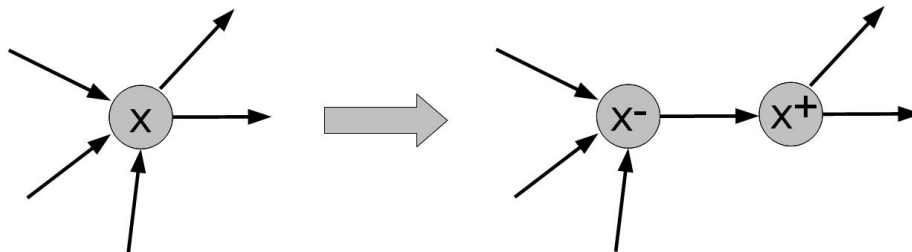


FIGURE 4.8 – Un nœud avec trois arcs incidents et deux arcs sortants est remplacé par deux nœuds : un nœud connecté aux arcs entrants et un autre connecté aux arcs sortants, les deux nœuds connectés entre eux par un arc de capacité unitaire. Cette transformation interdit la convergence du flot.

Avec cette transformation, tous les nœuds du graphe auront soit un seul arc incident, soit un seul arc sortant, et comme nous avons expliqué, cela interdit les convergences et les divergences de flot dans ces nœuds, et, par conséquent, dans tout le graphe. En même temps, toute chaîne dans le modèle de graphe non orienté aura un chemin correspondant dans le graphe résultant de ces transformations, et vice-versa. Le graphe résultant est fortement connexe.

Finalement, un nœud source et un nœud puits doivent être ajoutés pour obtenir un graphe de flot. Ces sont les seuls nœuds qui pourront avoir des divergences et des convergences de flot. Le nœud source sera connecté par des arcs sortants aux nœuds de l'ensemble A , tandis que le nœud puits sera connecté par des arcs entrants aux nœuds de l'ensemble B . Même si dans la plupart des cas, la permutation des ensembles A et B n'affecte pas les résultats de l'algorithme, il est préférable de faire correspondre à l'ensemble A les éléments délivrant du pétrole, et à l'ensemble B les éléments recevant du pétrole. Dans les chapitres suivants nous allons aborder des problèmes dans lesquels cette distinction sera importante. Alors, dans le cas d'une opération de chargement d'un tanker, par exemple, le nœud source sera connecté aux nœuds (représentant des segments avec des) réservoirs, tandis que le nœud puits sera connecté aux nœuds (représentant des segments avec des) bras de chargement.

Le graphe résultant des transformations est un graphe de flot avec une composante fortement connexe, constituée par les nœuds (d'entrée et de sortie) correspondant à des segments du réseau, et excluant les nœuds source et puits (lesquels ne correspondent pas à des segments réels du réseau).

L'algorithme 6 permet de réaliser ces transformations. Il construit un graphe de flot F à partir du graphe non orienté G . Les exposants sont utilisés pour distinguer des nœuds du graphe de flot liés au graphe non orienté original.

Algorithme 6 Graphe_de_flot(G,A,B)

```

1:  $F \leftarrow (\emptyset, \emptyset)$ 
2: pour chaque nœud  $n \in V[G]$  faire
3:    $V[F] \leftarrow V[F] \cup \{n^-\}$ 
4:    $V[F] \leftarrow V[F] \cup \{n^+\}$ 
5:    $E[F] \leftarrow E[F] \cup \{(n^-, n^+)\}$ 
6: pour chaque arc  $(u, v) \in E[G]$  faire
7:    $E[F] \leftarrow E[F] \cup \{(u^+, v^-\}$ 
8:  $V[F] \leftarrow V[F] \cup \{s\}$ 
9:  $V[F] \leftarrow V[F] \cup \{p\}$ 
10: pour chaque nœud  $u \in A$  faire
11:    $E[F] \leftarrow E[F] \cup \{(s, u^-\}$ 
12: pour chaque nœud  $v \in B$  faire
13:    $E[F] \leftarrow E[F] \cup \{(v^+, p)\}$ 
14: retour  $F$ 

```

Pour chaque nœud n de la liste d'adjacence du graphe non orienté G , deux nœuds sont créés dans la nouvelle liste du graphe de flot F : le nœud d'entrée n^- et le nœud de sortie n^+ (lignes 3 et 4). Chaque nœud d'entrée n^- aura un seul nœud dans sa liste de nœuds adjacents : le nœud de sortie correspondant n^+ (ligne 5). Chaque nœud de sortie n^+ aura dans sa liste de nœuds adjacents tous les nœuds d'entrée correspondant à la liste de nœuds adjacents du nœud n du graphe G (lignes 6 et 7).

L'algorithme contient une boucle effectuée un nombre $|V|$ de fois (lignes 1-4), une boucle effectuée $|E|$ fois (5-6), une boucle effectuée $|A|$ fois, et une autre effectuée $|B|$

fois. Alors le temps asymptotique de cette procédure est de $O(|V| + |E| + |A| + |B|)$.

La matrice d'adjacence du graphe de flot construit à partir du graphe de la figure 4.3, dans lequel les éléments délivrant du pétrole sont les réservoirs, tandis que les éléments recevant du pétrole sont les bras de chargement, est la suivante :

$a^- : (a^+)$	$h^+ : (d^-, e^-, f^-, i^-, j^-, k^-)$
$a^+ : (d^-)$	$i^- : (i^+)$
$b^- : (b^+)$	$i^+ : (g^-, h^-, l^-)$
$b^+ : (e^-)$	$j^- : (j^+)$
$c^- : (c^+)$	$j^+ : (g^-, h^-, m^-)$
$c^+ : (f^-)$	$k^- : (k^+)$
$d^- : (d^+)$	$k^+ : (g^-, h^-, n^-)$
$d^+ : (a^-, g^-, h^-)$	$l^- : (l^+)$
$e^- : (e^+)$	$l^+ : (i^-, p^-)$
$e^+ : (b^-, g^-, h^-)$	$m^- : (m^+)$
$f^- : (f^+)$	$m^+ : (j^-, p^-)$
$f^+ : (c^-, g^-, h^-)$	$n^- : (n^+)$
$g^- : (g^+)$	$n^+ : (k^-, p^-)$
$g^+ : (d^-, e^-, f^-, i^-, j^-, k^-)$	$s : (a^-, b^-, c^-)$
$h^- : (h^+)$	$p : --$

La figure 4.9 présente le graphe de flot correspondant au réseau de canalisation de la figure 4.3 :

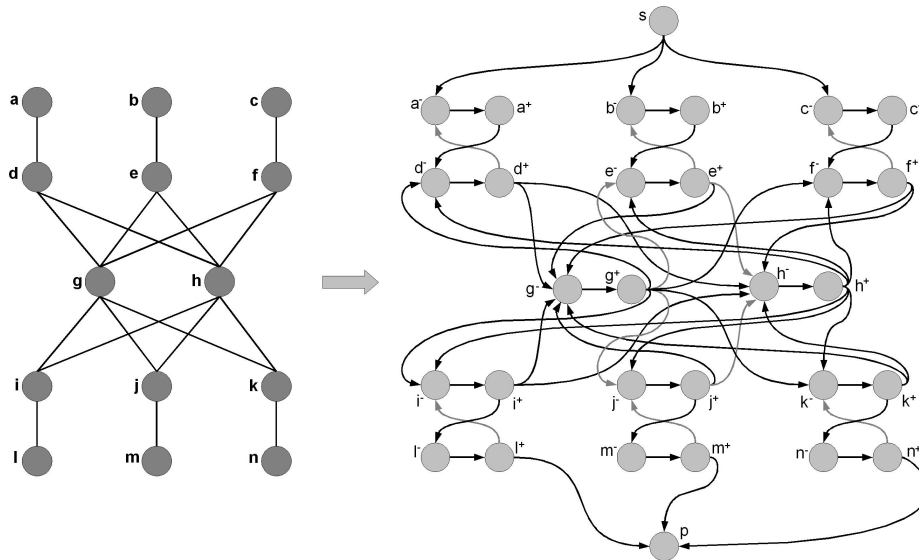


FIGURE 4.9 – Transformation d'un graphe non orienté en un graphe de flot pour des chemins disjoints. Tout les arcs du graphe de flot ont des capacités de un.

Toute chaîne du graphe non orienté a un chemin correspondant dans le

graphe de flot. Par exemple, la chaîne (a, d, g, j, h, f, c) correspond au chemin $(a^-, a^+, d^-, d^+, g^-, g^+, j^-, j^+, h^-, h^+, f^-, f^+, c^-, c^+)$.

4.3.2 Algorithme du flot maximum

Une fois que le graphe de flot a été construit, un algorithme de flot maximum quelconque peut être utilisé pour déterminer le nombre maximum de chemins disjoints équivalents. Nous avons utilisé un algorithme basé sur la méthode de Ford Fulkerson (voir chapitre 3). Cette méthode commence avec un flot valide initial (généralement un flot nul) lequel est incrémenté de façon successive à travers des chemins d'augmentation dans le graphe résiduel. Le flot maximum est obtenu lorsqu'il est impossible de trouver un chemin d'augmentation dans le graphe résiduel. Le temps asymptotique de l'algorithme Ford Fulkerson est de $O(|E| \times |f_{max}|)$, $|f_{max}|$ étant le flot maximum. N'importe quel algorithme de recherche de chemin peut être utilisé pour trouver les chemins d'augmentation. Un algorithme de parcours en profondeur (Depth-first Search) est un bon choix pour des valeurs petites de flot maximum. Pour des valeurs importantes de flot maximum, un algorithme de parcours en largeur pour rechercher des chemins, des méthodes de poussage/re-tiquetage, et des méthodes de flot bloquant, peuvent améliorer la performance de l'algorithme de flot maximum. Dans le cas des réseaux de transport de pétrole considérés, le flot maximum est borné par la taille de l'ensemble d'éléments fournisseurs, et par la taille de l'ensemble d'éléments récepteurs (la valeur du flot correspond au nombre d'alignements). Ces ensembles étant normalement de petite taille (quelques centaines au plus), nous avons choisi un algorithme de parcours en profondeur.

Une caractéristique importante de cet algorithme est le fait de non seulement calculer la valeur de flot maximum du graphe, mais aussi de donner une des possibles fonctions de flot maximum. Cette fonction donne un ensemble de chemins entre les deux ensembles d'éléments à connecter. Ces chemins sont une possible solution pour les opérations de transport considérées.

La fonction de flot maximum trouvée par un algorithme Ford Fulkerson dépend de l'algorithme de recherche des chemins d'augmentation utilisé, et aussi de choix arbitraires lors de la modélisation du réseau et lors du codage de l'algorithme (elle peut dépendre, par exemple, de l'ordre des nœuds dans la liste d'adjacence du graphe). L'existence d'autres fonctions de flot maximum est tout à fait possible, mais l'algorithme Ford Fulkerson n'en détermine qu'une. D'ailleurs, la détermination de toutes les fonctions possibles entraîne probablement des temps de calcul exponentiels, irréalisables dans la pratique pour les réseaux considérés.

La figure 4.10 présente une fonction de flot trouvée en utilisant un algorithme Ford Fulkerson pour le graphe de flot de la figure 4.9. Dans ce cas, un flot maximum de deux indique le nombre maximum d'alignements indépendants et simultanés entre les réservoirs et les bras de chargement. Au plus deux réservoirs peuvent délivrer du pétrole à deux bras de chargement en même temps. Les chemins observés dans la figure peuvent être traduits comme un ensemble de vannes à ouvrir/fermer de façon à établir le nombre maximum d'alignements entre réservoirs et bras de chargement.

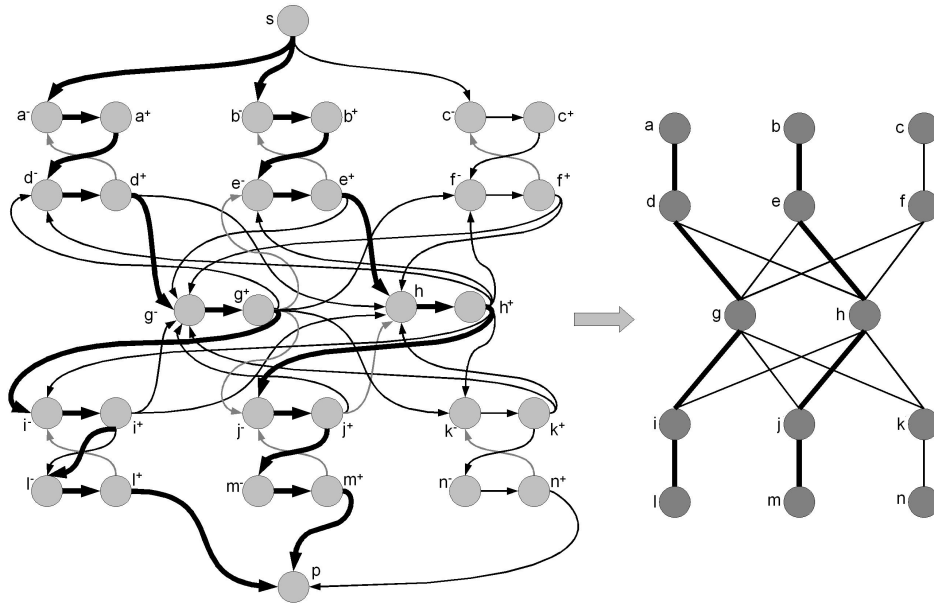


FIGURE 4.10 – Fonction de flot maximale obtenue en utilisant un algorithme Ford Fulkerson dans un graphe de flot, et le graphe non orienté correspondant.

- Chaînes correspondantes dans le modèle de graphe non orienté :

$$\begin{aligned} &(a, d, g, i, l) \\ &(b, e, h, j, m) \end{aligned}$$

- Chemins de flots trouvés par l'algorithme Ford Fulkerson :

$$\begin{aligned} &(s, a^-, a^+, d^-, d^+, g^-, g^+, i^-, i^+, l^-, l^+, p) \\ &(s, b^-, b^+, e^-, e^+, h^-, h^+, j^-, j^+, m^-, m^+, p) \end{aligned}$$

- Vannes à ouvrir : $(a, d), (d, g), (g, i), (i, l); (b, e), (e, h), (h, j), (j, m)$.

Il s'agit des vannes contenues dans les chaînes trouvées.

- Vannes à fermer : $(d, h), (e, g), (f, g), (g, j), (g, k), (h, i), (f, h), (h, k)$.

Il s'agit de toutes les vannes adjacentes aux segments (nœuds) faisant partie des chaînes trouvées, mais n'étant pas contenues dans ces chaînes.

Ces informations peuvent être utilisées directement par les opérateurs des systèmes SCADA pour établir des alignements dans le réseau de transfert.

4.3.3 Validation

La définition de la capacité opérative pour un type d'alignement lie directement sa valeur au nombre maximum de chaînes disjointes équivalentes entre deux ensembles de nœuds du graphe non orienté (PMCDE). Un algorithme permettant de résoudre le PMCDE donne aussi la valeur de capacité opérative recherchée.

Le PMCDE est équivalent au problème du nombre maximum de chemins disjoints (par les nœuds) entre deux nœuds, correspondant aux nœuds source et puits, d'un graphe. Le problème est abordé en effectuant des transformations dans le graphe considéré, et en appliquant un algorithme de flot maximum. Ces transformations vont donner lieu à un graphe de flot avec la propriété particulière d'interdire les convergences/divergences de flot dans tous les nœuds exceptant les nœuds source et puits. En raison de cette propriété, les fonctions de flot vont être composées de chemins disjoints de flot allant de la source au puits. Les capacités des arcs étant unitaires, chaque chemin de flot contribue au plus avec une valeur de un à la valeur totale du flot, donc la valeur du flot donne aussi le nombre de chemins de flot. En particulier, un flot maximum va correspondre au nombre maximum de chemins de flot. Donc, l'algorithme proposé donne bien une solution au PMCDE, et permet le calcul de la valeur de la capacité opérative pour un type d'alignement.

4.3.4 Capacité Opérative pour des sous-types d'alignement

La modélisation sous forme de graphe de flot utilisée pour le calcul de la capacité opérative, interdisant des convergences/divergences de flot, donne la possibilité d'ajouter quelques contraintes additionnelles lors du calcul de la capacité opérative. Ces nouvelles contraintes sont liées à la notion de sous-type d'alignement et vont permettre de trouver des solutions à des problèmes spécifiques d'alignement.

Comme nous l'avons montré, les spécifications demandées pour un alignement peuvent être plus ou moins contraignantes, ce qui définit des types d'alignement avec différents degrés de description. Souvent, un type d'alignement peut être défini à partir d'un autre type d'alignement, en rajoutant des spécifications. Prenons par exemple, les spécifications suivantes :

- Fournisseur : réservoir
- Récepteur : tanker

Ces spécifications définissent un type d'alignement A , entre des réservoirs et des bras de chargement. En rajoutant une autre spécification, un nouveau type d'alignement est créé :

- Fournisseur : réservoir
- Récepteur : tanker
- Type de pétrole : Anaco Wax

Le nouveau type d'alignement B , entre des réservoirs contenant du pétrole de type "Anaco Wax" et des bras de chargement, satisfait aussi les spécifications demandées pour le type d'alignement A . Nous pouvons dire que le type d'alignement B est un sous-type du type d'alignement A .

À partir de ces considérations nous pouvons définir les sous-types d'alignement de la manière suivante.

Soit un type d'alignement A , satisfaisant un ensemble de spécifications S_A , et un type d'alignement B , satisfaisant un ensemble de spécifications S_B . B est un sous-type d'alignement de A , si et seulement si : tout alignement x de type B est aussi un alignement de type A .

Un type d'alignement étant défini à partir d'un ensemble d'éléments fournisseurs et un ensemble d'éléments récepteurs, un sous-type d'alignement peut être défini à partir de la relation entre des ensembles d'éléments fournisseurs, ou récepteurs à connecter.

Soit un type d'alignement A , connectant un élément de type F_A avec un élément de type R_A , et un type d'élément B , connectant un élément de type F_B avec un élément de type R_B . Soient les ensembles d'éléments $X_A = x : x$ est de type F_A , $Y_A = y : y$ est de type R_A , $X_B = x : x$ est de type F_B , $Y_B = y : y$ est de type R_B , B est un sous-type d'alignement de A si et seulement si $X_B \subseteq X_A$ et $Y_B \subseteq Y_A$.

À partir de la notion de sous-type d'alignement, il est possible de découper un ensemble d'éléments de même type en plusieurs sous-ensembles d'éléments de différents sous-types. Par exemple, l'ensemble des réservoirs peut être découpé en sous-ensembles pour chaque type de pétrole contenu par les réservoirs ; l'ensemble des bras de chargement peut être découpé en sous-ensembles pour chaque type de quai...

Lors du calcul d'une fonction de flot pour un type d'alignement, des valeurs maximales peuvent être imposées à chaque sous-ensemble d'éléments pour limiter les nombres d'éléments de chaque sous-ensemble qui peuvent être utilisés. L'exemple suivant montre l'utilité de l'imposition de ces contraintes.

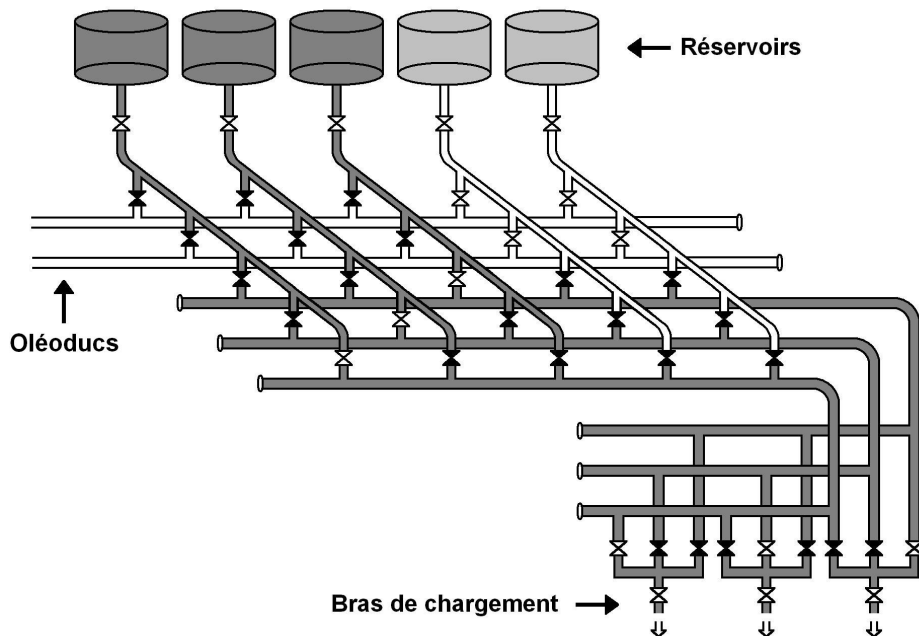


FIGURE 4.11 – Alignements entre réservoirs et bras de chargement dans un réseau de canalisation. Le gris foncé et le gris clair des réservoirs correspondent à différents types de pétrole.

La figure 4.11 montre un réseau dans lequel l'ensemble des réservoirs pleins

peut être découpé en deux sous-ensembles correspondant aux réservoirs contenant deux types différents de pétrole (gris foncé pour l'un et gris clair pour l'autre). La figure montre aussi les alignements entre réservoirs et bras de chargement qui correspondent à la solution de flot maximum trouvée pour les alignements entre l'ensemble des cinq réservoirs et l'ensemble des trois bras de chargement. Nous rappelons que les vannes colorées en noir correspondent à des vannes qui doivent être fermées pour isoler l'alignement du reste du réseau. Les vannes non colorées qui se trouvent dans l'alignement (connectant deux canalisations en gris) doivent être ouvertes pour permettre le transport de pétrole entre les éléments.

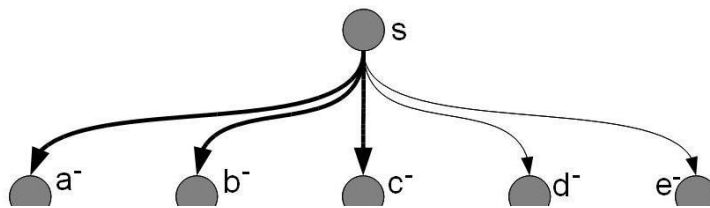


FIGURE 4.12 – Partie du graphe de flot pour des chaînes disjointes construit à partir du réseau de la figure 4.11.

La figure 4.12 présente une section du graphe de flot correspondant au nœud source (s) et aux arcs qui le lient aux nœuds correspondant aux différents réservoirs (a^- , b^- , c^- , d^- , e^-). Dans la méthode de construction du graphe de flot utilisée précédemment le nœud source est connecté directement aux nœuds (d'entrée) correspondant aux différents éléments de l'ensemble A , par des arcs avec des capacités unitaires. Cette modélisation peut donner une solution de flot maximum dans laquelle, par exemple, tous les alignements connectent les réservoirs contenant un seul type de pétrole (représenté par le gris foncé dans la figure 4.11). Cette solution n'est pas unique, alors il peut être utile aux opérateurs de limiter la quantité d'alignements connectant (aux bras de chargement) des réservoirs contenant ce type de pétrole, représenté par le gris foncé, pour explorer la possibilité de connecter des réservoirs contenant d'autres types de pétrole. La figure 4.12 montre une modélisation permettant de tenir compte d'une restriction sur le nombre d'alignements pour certains sous-types d'éléments à connecter.

Cette modélisation implique l'ajout de nœuds intermédiaires entre le nœud source du graphe de flot et les nœuds (d'entrée) correspondant à l'ensemble d'éléments fournisseurs à connecter. Chacun de ces nœuds intermédiaires (s' , s'') va correspondre aux différents sous-types d'éléments faisant partie de l'ensemble d'éléments fournisseurs. Chacun sera connecté aux nœuds (d'entrée) correspondant aux éléments faisant partie du sous-ensemble respectif. La capacité des arcs entre le nœud source et les nœuds intermédiaires va permettre de restreindre le nombre d'éléments de chaque type utilisés pour les alignements. La figure 4.13 montre ce type de modélisation pour le réseau de la figure 4.11, dans lequel les alignements pour des réservoirs contenant le premier type de pétrole (gris foncé) sont limités

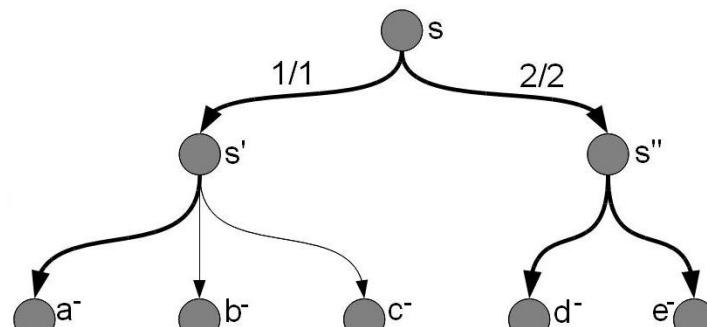


FIGURE 4.13 – Partie du graphe de flot du réseau de la figure 4.11 construit d'après la méthode tenant compte des contraintes additionnelles sur le nombre de sous-types d'alignement.

à un. Ceci permet de savoir s'il est possible de connecter des réservoirs contenant l'autre type de pétrole (gris clair), étant ceux-ci limités à deux. La solution trouvée est montrée figure 4.14.

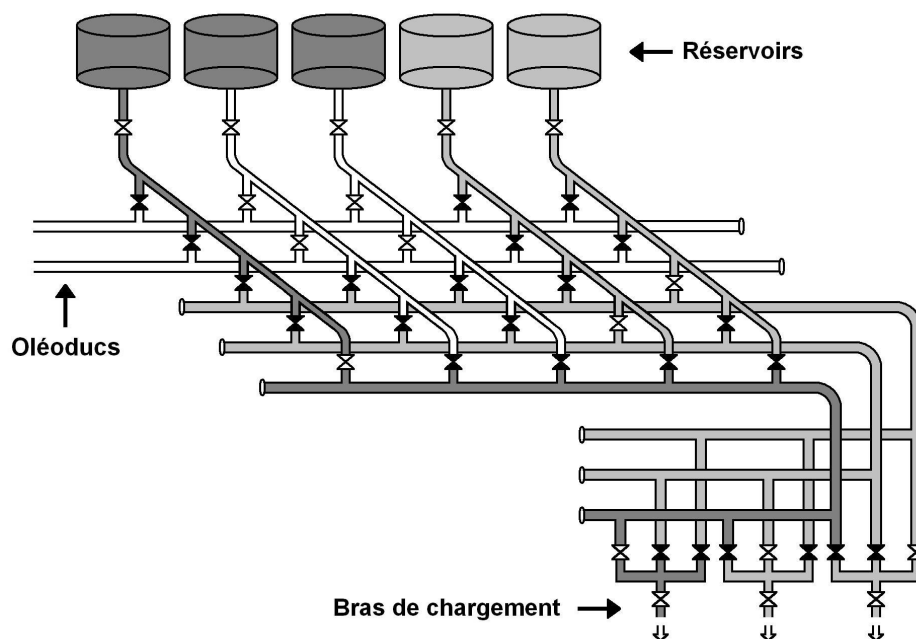


FIGURE 4.14 – Alignements trouvés en utilisant une modélisation qui restreint le nombre de réservoirs contenant le premier type de pétrole (gris foncé) à un et le nombre de réservoirs contenant le deuxième type de pétrole à deux.

Ces restrictions sont possibles de façon analogue pour les éléments récepteurs. Dans ce cas les nœuds intermédiaires vont être entre le nœud puits et les nœuds (de

sortie) correspondants aux différents éléments récepteurs de pétrole.

Les nœuds intermédiaires ne font pas partie de la composant fortement connexe du graphe correspondant aux nœuds des segments du réseau de canalisation.

4.3.5 Ensemble d'indicateurs de Capacité Opérative

Un indicateur de capacité opérative est défini comme la valeur de la capacité opérative pour un type d'opération donnée. À chaque type d'opération correspond un indicateur de capacité opérative. Il est possible alors de regrouper les indicateurs correspondant aux opérations les plus importantes et les présenter aux opérateurs, afin qu'ils puissent prendre des décisions en évaluant les effets de ces décisions sur ces indicateurs.

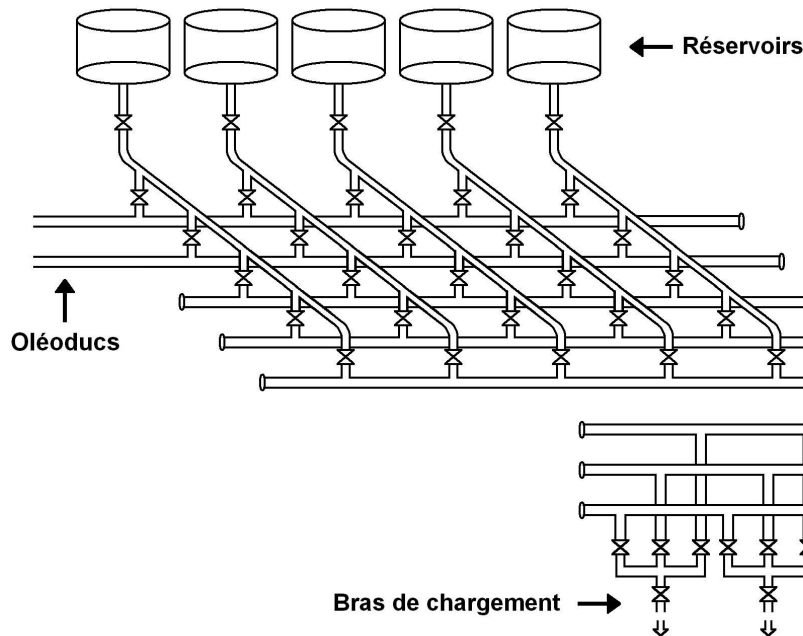


FIGURE 4.15 – Réseau de transport de pétrole.

La figure 4.15 montre un réseau avec 5 réservoirs, 2 oléoducs d'arrivée et 3 bras de chargement. Trois types d'opération sont considérés et à chacun correspond un indicateur de capacité opérative (tableau 4.2).

La figure 4.16 montre un alignement entre un réservoir et un bras de chargement. Cette opération a un effet sur les indicateurs de capacité opérative.

Le tableau 4.4 montre les nouvelles valeurs des indicateurs de capacité opérative.

La figure 4.17 montre une opération de maintenance sur deux vannes du réseau, et le tableau 4.4 les conséquences de cette opération sur les indicateurs de capacité opérative.

Les indicateurs vont permettre de prévoir l'impact du choix d'un alignement ou

Type d'opération	Indicateur de C.O.
Alignements entre oléoducs et réservoirs.	2
Alignements entre réservoirs et bras de chargement.	3
Alignements entre oléoducs et bras de chargement.	2

TABLE 4.2 – Ensemble d'indicateurs de capacité opérative (C.O.).

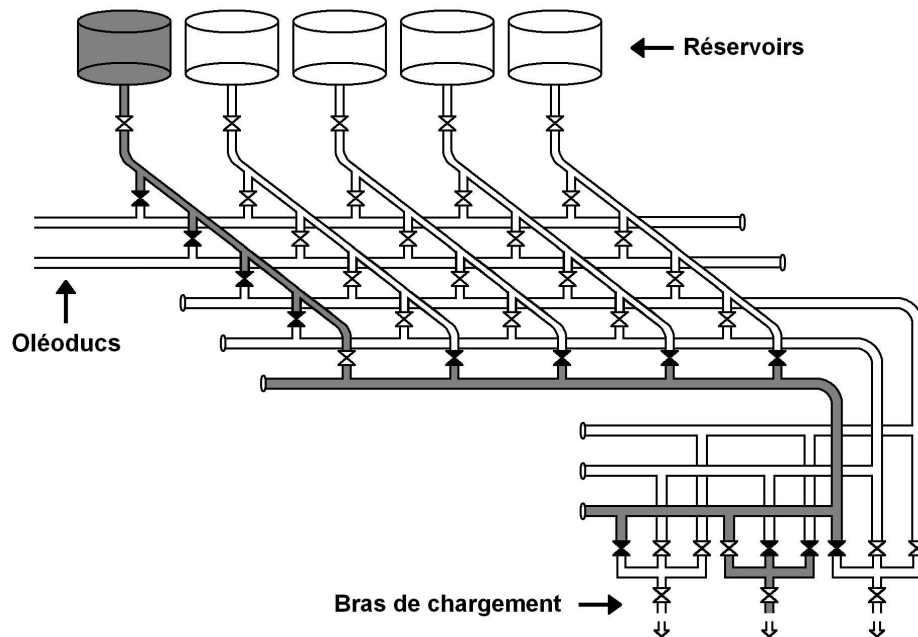


FIGURE 4.16 – Alignement entre un réservoir et un bras de chargement (en gris).

Type d'opération	Indicateur de C.O.
Alignements entre oléoducs et réservoirs.	2
Alignements entre réservoirs et bras de chargement.	2
Alignements entre oléoducs et bras de chargement.	2

TABLE 4.3 – Ensemble d'indicateurs de capacité opérative (C.O.) résultants d'un alignement.

Type d'opération	Indicateur de C.O.
Alignements entre oléoducs et réservoirs.	1
Alignements entre réservoirs et bras de chargement.	2
Alignements entre oléoducs et bras de chargement.	1

TABLE 4.4 – Ensemble d'indicateurs de capacité opérative (C.O.) résultants de deux défaillances.

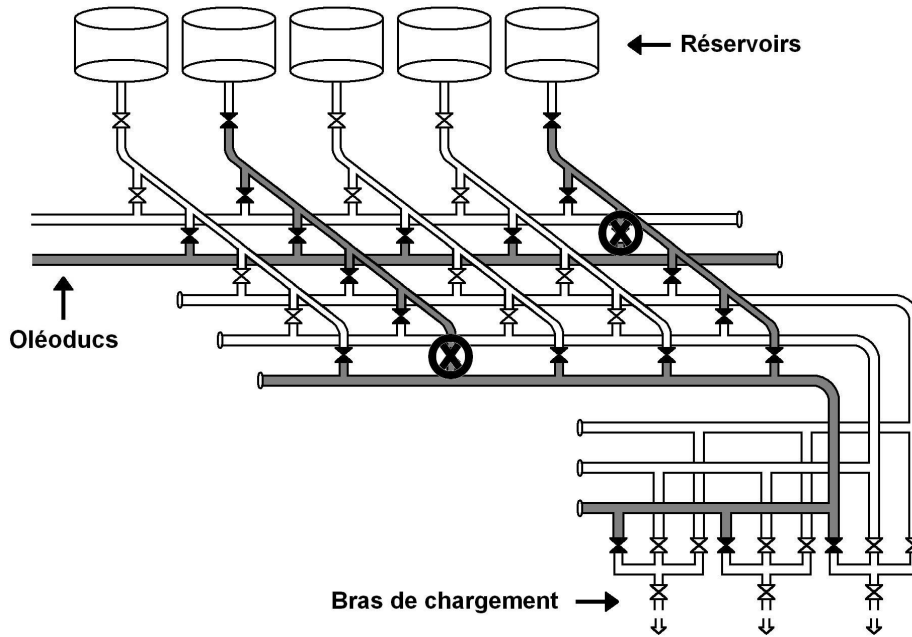


FIGURE 4.17 – Opération de maintenance pour deux vannes du réseau.

des opérations éventuelles de maintenance. Ceci constitue un outil pour la prise de décision à court, moyen et long terme.

4.4 Développements complémentaires

Nous allons présenter deux applications intéressantes basées sur les indicateurs de capacité opérative. La première est la définition des notions de criticité liées à la capacité opérative et l'analyse des éléments critiques du système. La deuxième est un algorithme de recherche de chemin entre deux nœuds et qui maximise la capacité opérative pour un type d'opération donnée. Cet algorithme sera utile pour l'extension de l'algorithme de recherche de chemin pour des alignements forcés.

4.4.1 Analyse de criticité

La capacité opérative d'un système pour un type d'opération donné dépend du nombre de nœuds du graphe et des arcs qui les connectent. Si des nœuds ou des arcs sont supprimés du graphe, ou si leur disposition est modifiée, alors la capacité opérative peut changer. Différentes situations, telles que des opérations d'alignement ou l'occurrence de défaillances, peuvent affecter la capacité opérative du système du fait de la modification du réseau. Ces situations peuvent être groupées en deux ensembles selon les effets qu'elles ont sur le modèle du réseau.

La plupart des opérations ou l'occurrence de défaillances vont produire des res-

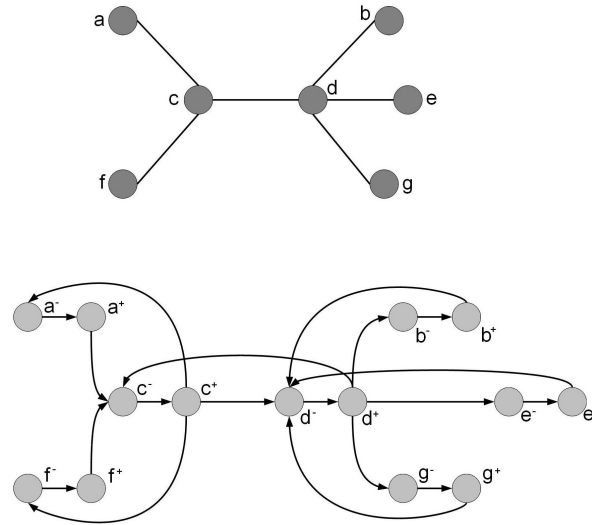


FIGURE 4.18 – Graphe non orienté et graphe de flot correspondants à un réseau de canalisation.

trictions dans l’usage d’une ou plusieurs portions du réseau. L’installation de nouveaux équipements ou de nouvelles canalisations, les opérations de maintenance, et les opérations de réparation provoquées par des fractures ou des fissures dans des canalisations ou par des défaillances mécaniques sur des vannes ou des pompes, sont des exemples de situations dans lesquelles il est nécessaire d’isoler une portion du réseau. L’isolement est assuré en fermant les vannes délimitant la portion du réseau concernée. Un effet équivalent est produit par la défaillance d’une vanne en position fermée, généralement due à une faute électrique ou de communication, empêchant son ouverture. Toutes ces situations impliquent la fermeture d’une ou plusieurs vannes pendant un temps significatif, empêchant la circulation du pétrole à travers des portions du réseau. Elles seront identifiées comme des “situations bloquantes de flux”. Chaque vanne fermée à cause d’une situation bloquante de flux sera appelée : “vanne indisponible fermée”. Une vanne indisponible fermée cause l’isolement de deux segments du réseau, ce qui est modélisé dans le graphe non orienté en supprimant l’arc correspondant. Pour l’arc (a, b) supprimé du graphe non orienté, les arcs (a^+, b^-) et (b^+, a^-) seront supprimés dans le graphe de flot correspondant.

La figure 4.18 présente le graphe non orienté et le graphe de flot d’un réseau de canalisation simple. La figure 4.19 montre les graphes du même réseau dans lequel la vanne connectant les segments c et d est une vanne indisponible fermée.

L’autre type d’événement ayant un impact sur le graphe du système correspond à un type de faute particulier. Dans ce cas, une défaillance dans une vanne en position ouverte empêche sa fermeture. Le flux de pétrole à travers cette vanne ne peut alors pas être bloqué. Ces situations seront identifiées comme des “situations permissives de flux”. Chaque vanne ouverte à cause d’une situation permissive de

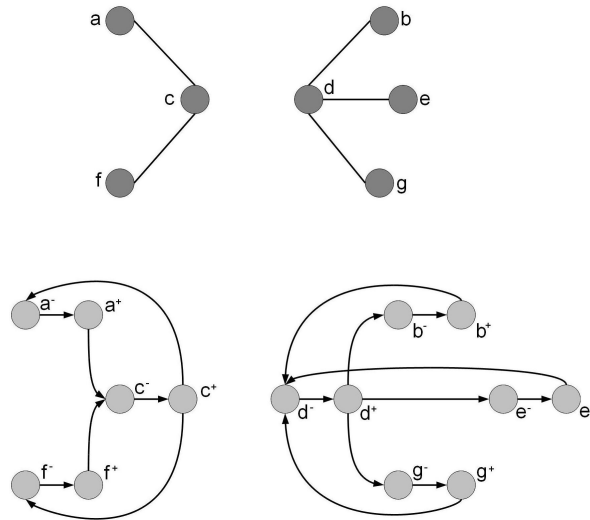


FIGURE 4.19 – Graphes correspondants au cas où la vanne connectant les segments c et d de la figure 4.18 est une vanne indisponible fermée.

flux sera appelée : “vanne indisponible ouverte”. Une vanne indisponible ouverte ne peut pas isoler les segments qu’elle connecte, alors, d’après la définition de segment que nous avons établie, cela implique la fusion des deux segments connectés en un seul segment. Dans le modèle, cette fusion de segments peut être réalisée par la fusion des nœuds correspondants. Pour les nœuds a et b fusionnés dans le graphe non orienté, dans le graphe de flot correspondant seront fusionnés les nœuds a^- et b^- d’une part, et les nœuds a^+ et b^+ d’autre part.

La figure 4.20 montre les graphes du même réseau dans lequel la vanne connectant les segments c et d est une vanne indisponible fermée.

Les situations bloquantes de flux aussi bien que les situations permissives de flux, peuvent réduire la capacité opérative du système pour un type d’opération donné. Les premières peuvent bloquer de possibles chemins entre les deux ensembles d’éléments à connecter. Les deuxièmes peuvent rendre indisponibles des chemins empêchant leur isolement. À partir de ces remarques, nous pouvons définir une notion de criticité pour des éléments du réseau.

Soit E un ensemble d’arcs du graphe non orienté G . E est un ensemble bloquant de flux pour un type d’opération donné si la suppression de tous les arcs $e \in E$ du graphe G diminue la capacité opérative du système pour l’opération concernée. De même, E est un ensemble permissif de flux pour un type d’opération donné si la fusion de chaque paire de nœuds connectés par chaque arc $e \in E$ diminue la capacité opérative du système pour l’opération concernée.

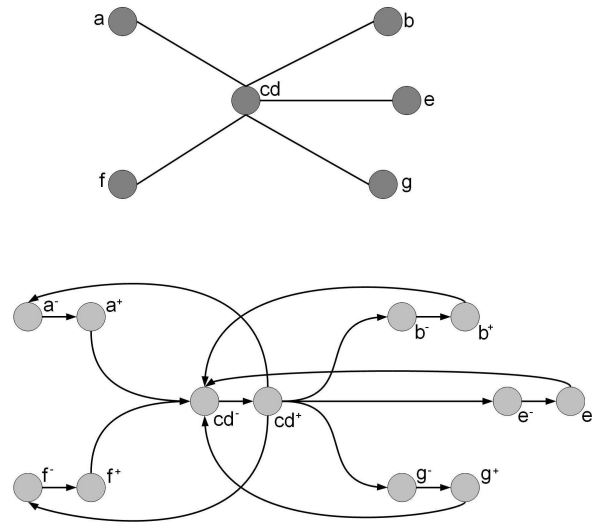


FIGURE 4.20 – Graphes correspondants au cas où la vanne connectant les segments c et d de la figure 4.18 est une vanne indisponible ouverte.

4.4.1.1 Détermination des vannes critiques

La criticité bloquante et la criticité permissive peuvent être étudiées pour des ensembles ne contenant qu'un seul arc. Chacun de ces ensembles correspond à une vanne unique. Un algorithme peut calculer la diminution de la capacité opérative causée par chaque vanne indisponible, pour des situations bloquantes aussi bien que pour des situations permissives, considérant le reste des vannes comme disponibles. Le résultat de cet algorithme est une liste des vannes critiques bloquantes, et une liste des vannes critiques permissives pour un type d'opération donné.

La figure 4.21 montre un réseau dans lequel les vannes critiques bloquantes et permissives ont été déterminées. Dans ce cas le type d'opération considéré est le transport de pétrole à partir des réservoirs vers les bras de chargement, le type de pétrole étant considéré inconnu. La capacité opérative du système pour cette opération est de deux. Nous pouvons constater que comme le nombre de réservoirs est supérieur au nombre de bras de chargement, la fermeture d'une seule vanne des bras de chargement est suffisante pour diminuer la capacité opérative. Moins évident est le fait d'avoir une diminution de la capacité opérative lorsque la vanne du premier réservoir est une vanne indisponible fermée. Cela est dû au fait de ne pas pouvoir connecter le deuxième et le troisième réservoir aux bras de chargement en même temps. Par conséquent, ces trois vannes sont critiques bloquantes. La vanne critique permissive indiquée sur la figure est nécessaire pour isoler l'alignement entre le premier réservoir et le premier bras de chargement, et les possibles alignements entre les autres réservoirs et l'autre bras de chargement. À partir de cet exemple nous pouvons déduire que, pour des réseaux complexes, l'analyse de la criticité des vannes peut devenir très difficile, ce qui rend nécessaire l'utilisation de l'algorithme

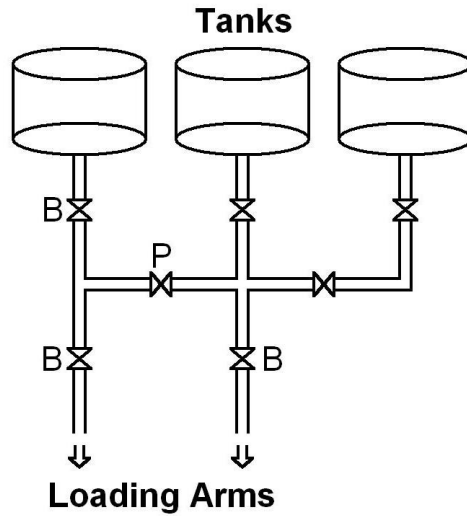


FIGURE 4.21 – Un réseau simple avec trois vannes critiques bloquantes (B) et une vanne critique permissive (P) pour les opérations de transport de pétrole des réservoirs vers les bras de changement.

proposé.

L'information de criticité de vannes peut être importante dans différents contextes de la prise de décision.

Par exemple, la connaissance de la criticité de certains éléments peut mener à incorporer de la redondance au réseau, augmentant ainsi la disponibilité du système. Une vanne redondante peut être ajoutée en parallèle à une vanne critique bloquante pour améliorer la capacité opérative du système en cas de défaillances. De même, une vanne redondante peut être ajoutée en série à une vanne critique permissive.

Des mécanismes de “chien de garde” (watchdog) peuvent être réglés de manière correcte en utilisant ces notions de criticité bloquante et permissive. L'état initial des vannes critiques bloquante étant l'état ouvert, ces vannes s'ouvriraient automatiquement lors de la détection d'une faute de communication et après le dépassement de l'horloge du chien de garde. Cela permettrait d'éviter la diminution de la capacité opérative pour les opérations prioritaires. De même l'état par défaut des vannes critiques permissives étant l'état fermée permettrait aussi d'éviter la diminution de la capacité opérative. Ces mécanismes de chien de garde font actuellement partie du logiciel SCADA développé par PDVSA.

Quelques opérations, telles que les opérations de maintenance, l'installation de nouveaux réservoirs ou canalisations, peuvent être planifiées en tenant compte de la criticité des éléments. Des opérations impliquant la fermeture des vannes critiques bloquantes pourraient être réalisées de façon plus rapide pour augmenter la disponibilité du système.

4.4.1.2 Détermination des segments critiques

La criticité bloquante permet aussi l'analyse de la criticité des segments du réseau, considérés un par un. La criticité d'un segment est définie comme la criticité bloquante d'un ensemble E , tel que les éléments appartenant à E sont les arcs connectés au nœud du segment.

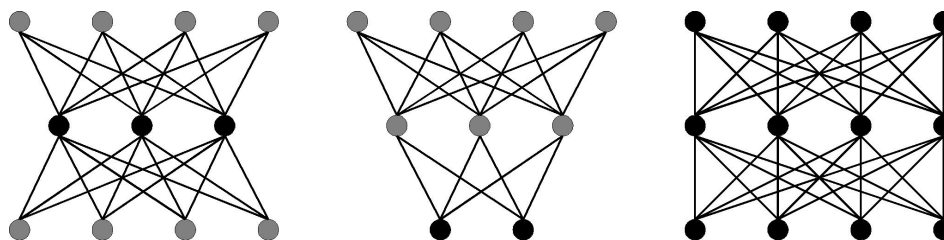


FIGURE 4.22 – Trois graphes correspondant à trois réseaux différents. Les opérations considérées sont les alignements entre les nœuds supérieurs et les nœuds inférieurs. Les nœuds critiques sont colorés en noir.

Pour un type d'opération donné, nous pouvons déterminer la liste des segments critiques du réseau en calculant la diminution de la capacité opérative causée par la suppression de chaque segment. La figure 4.22 montre les segments critiques de trois réseaux différents. Cette information peut être utilisée pour augmenter la disponibilité du système en faisant plus attention aux fautes dans des éléments critiques. La fissure ou rupture d'une canalisation dans un segment critique devrait être traitée prioritairement.

4.4.2 Algorithme de recherche de chemin maximisant la capacité opérative

Il est possible de développer un algorithme de recherche de chemin entre deux nœuds maximisant la capacité opérative. Le chemin recherché évitera le plus possible les interférences avec les alignements entre deux ensembles de nœuds. Ce problème est lié au problème du nombre maximum de chemins disjoints, mais il diffère du fait de considérer la connexion d'une paire de nœuds comme prioritaire, tandis que le reste des paires de nœuds à connecter est considéré inconnu, et le chemin en question ne sera pas modifié lors que les paires de nœuds à connecter deviennent connues.

L'idée est d'utiliser la logique d'un algorithme de recherche du plus court chemin mais dont les coûts des arcs correspondront à la diminution du flot maximum causée par la suppression du nœud ciblé par cet arc (extrémité terminale de l'arc). Il faut remarquer que, contrairement aux problèmes classiques de recherche du plus court chemin, ici les coûts des arcs ne sont pas connus d'avance. En effet, car la diminution du flot maximum ne dépend pas seulement du nœud ciblé, mais aussi du parcours effectué par l'algorithme avant d'examiner ce nœud. Il faudra alors faire un calcul du flot maximum à chaque nœud examiné par l'algorithme. Ce calcul à chaque pas

augmente de façon importante le temps d'exécution de l'algorithme. Pour faire face à cet inconvénient nous avons étudié les conséquences de la suppression d'un nœud du graphe de flot sur la fonction de flot maximum. Ceci permettra de calculer le flot maximum à chaque pas à partir du flot maximum calculé précédemment et ainsi de réduire considérablement les temps de calcul.

La figure 4.23 présente le fonctionnement de l'algorithme maximisant la capacité opérative. Le fonctionnement est très similaire à celui de l'algorithme de Dijkstra pour le plus court chemin. Lors du calcul des coûts estimés vers les nœuds adjacents, une sous-routine calcule le flot maximum résultant de la suppression du nœud adjacent considéré. Le coût de l'arc ciblant le nœud adjacent correspondra à la diminution du flot maximum résultant de cette suppression.

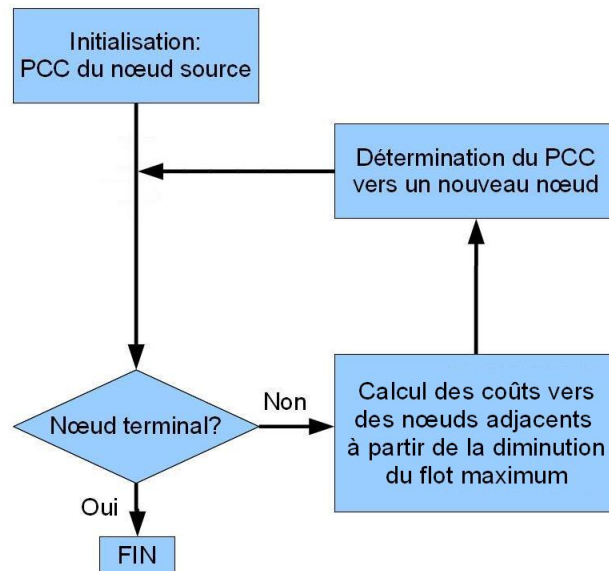


FIGURE 4.23 – Fonctionnement général de l'algorithme de recherche de chemin maximisant la capacité opérative.

Premièrement, nous allons étudier les conséquences de la suppression d'un nœud du graphe sur le flot maximum. Ensuite, à partir de cette étude nous allons développer une sous-routine permettant de calculer le flot maximum induit par la suppression du nœud à partir du flot maximum précédent. Cette sous-routine est incorporée dans un algorithme classique du chemin le plus court dans lequel les coûts correspondent à la diminution du flot produite par la suppression du nœud ciblé. La suppression des nœuds ne pouvant que diminuer le flot maximum, ces coûts seront toujours positifs ou nuls, et donc l'algorithme de Dijkstra est le mieux adapté pour notre problème.

Soit $G = (V, E)$ le graphe de flot correspondant à notre réseau. Chaque arc $(u, v) \in E$ a une capacité de un : $c(u, v) = 1$. Soit s et p les nœuds source et puits

respectivement. Un flot dans G est une fonction à valeurs entières, $f : V \times V \rightarrow Z$, laquelle satisfait les contraintes de capacité, antisymétrie et conservation du flot.

4.4.2.1 Diminution du flot due à la suppression d'un nœud

Considérons un graphe de flot dans lequel l'algorithme de flot maximum a été exécuté. Soit f_m le flot obtenu après cette exécution de l'algorithme.

Les conséquences de la suppression d'un nœud sur le flot peuvent être distinguées en deux cas :

1) Le nœud supprimé r n'est traversé par aucun flot :

$$\forall n \in V, f_m(n, r) = 0$$

Alors le flot du nouveau graphe G' , lequel est le sous-graphe de G induit par $V - \{r\}$, n'est pas affecté ; la conservation du flot n'est pas violée.

2) Le nœud supprimé r est traversé par un flot non nul.

$$\exists n \in V, f_m(n, r) \neq 0$$

Le flot maximum doit être recalculé. La suppression du nœud produit une violation de la contrainte de conservation du flot f_m dans G' . Le flot excédentaire doit être calculé pour le soustraire de la fonction de flot. Dû à la contrainte de non convergence-divergence du flot (introduite lors de la construction du graphe de flot) le flot excédentaire peut être calculé d'une manière très simple, en recherchant un chemin de flot négatif du nœud r jusqu'au nœud source s , et un chemin de flot positif du nœud r jusqu'au nœud puits p . Un chemin de flot est défini comme un chemin composé d'arcs avec un flot non nul. Le flot excédentaire est une fonction assignant des flots de valeur un à chaque arc appartenant aux chemins de flot de s jusqu'à r et de r jusqu'à p . Le flot résultant $f_r = f_m - f_s$ est un flot valide pour le graphe G' (satisfaisant la conservation du flot).

4.4.2.2 Sous-routine du flot excédentaire

La sous-routine de flot excédentaire (algorithme 7) détermine la fonction de flot pour un chemin de flot allant de s jusqu'à p et en passant par n , à partir d'un flot valide du graphe de flot pour des chemins disjoints G .

Les lignes 2 à 4 initialisent la fonction de flot excédentaire à zéro. Les lignes 5 à 11 recherchent un chemin de n jusqu'à s , et assignent un flot de un à chaque arc du chemin trouvé, tandis que les lignes 12 à 16 réalisent une tâche analogue en cherchant un chemin de n jusqu'à p . Si aucun flot ne traverse n , les boucles "tant que" (while) ne sont pas exécutées et la sous-routine retourne une valeur nulle.

L'algorithme de flot maximum utilisé détermine une fonction de flot qui correspond à un flot maximum, mais cette fonction peut contenir des cycles de flot. Dans les lignes 1, 6, 10 et 11 la variable booléenne "CYCLE" est utilisée pour détecter des cycles de flot et ainsi éviter des recherches infinies de chemins.

Algorithme 7 Flot_ex(G, n, f)

```

1: CYCLE ← FAUX
2: pour chaque arc  $(u, v) \in E[G]$  faire
3:    $f_s[u, v] \leftarrow 0$ 
4:    $f_s[v, u] \leftarrow 0$ 
5:  $v \leftarrow n$ 
6: tant que (il existe un arc  $(u, v) \in E$ ) et  $(f[u, v] = 1)$  et  $(\text{CYCLE} = \text{FAUX})$ 
   faire
7:    $f_s[u, v] \leftarrow 1$ 
8:    $f_s[v, u] \leftarrow 1$ 
9:    $v \leftarrow u$ 
10:  si  $v = n$  alors
11:    CYCLE ← VRAI
12:   $u \leftarrow n$ 
13: tant que (il existe un arc  $(u, v) \in E$ ) et  $(f[u, v] = 1)$  et  $(\text{CYCLE} = \text{FAUX})$ 
   faire
14:    $f_s[u, v] \leftarrow 1$ 
15:    $f_s[v, u] \leftarrow 1$ 
16:    $u \leftarrow v$ 
17: retour  $f_s$ 

```

4.4.2.3 Sous-routine de recalcul du flot maximum

La sous-routine du flot excédentaire permet de calculer un flot f_r , valide pour le graphe G' . Ce flot résulte de la déduction du flot excédentaire au flot maximum f_m trouvé pour G . Or, due aux capacités unitaires assignées à chaque arc, la diminution du flot causée par la suppression du nœud a une valeur maximum de un. Néanmoins f_r peut ne pas être un flot maximum de G' , car un chemin d'augmentation dans le graphe résiduel induit par f_r peut bien exister. Soit G_f le graphe résiduel du graphe G induit par la fonction de flot f .

En appliquant une procédure basée sur l'algorithme Ford Fulkerson nous pouvons déterminer f_h , le flot maximum de G' . Soit G_f le graphe résiduel de G induit par f . La procédure est réalisée par l'algorithme 8. Dû aux capacités de un, le flot ne peut augmenter que d'une valeur unitaire (ligne 6).

4.4.2.4 Sous-routine d'actualisation

Les sous-routines du flot excédentaire et de recalcul du flot maximum sont appelées à chaque actualisation des distances estimées vers les nœuds pour déterminer le coût de l'arc concerné. Ce coût correspond à la diminution du flot causée par la suppression du nœud ciblé par cet arc. Il faut remarquer que le même arc peut avoir différents coûts dépendant du parcours suivi par l'algorithme. Cela ne pose pas de problème car les chemins ne sont effacés que si un autre chemin plus court est trouvé.

Algorithme 8 Recalcul_flot_max(G, s, p, f)

```

1: pour chaque arc  $(u, v) \in E[G]$  faire
2:    $f_h[u, v] \leftarrow f[u, v]$ 
3:    $f_h[v, u] \leftarrow f[v, u]$ 
4: si il existe un chemin  $\pi$  de  $s$  jusqu'à  $p$  dans  $G_f$  alors
5:   pour chaque arc  $(u, v) \in \pi$  faire
6:      $f_h[u, v] \leftarrow f_h[u, v] + 1$ 
7:      $f_h[v, u] \leftarrow -f_h[u, v]$ 
8: retour  $f_h$ 

```

La sous-routine d'actualisation (algorithme 9) non seulement détermine les distances les plus courtes estimées vers un nœud en termes de diminution du flot, mais elle détermine aussi la fonction de flot correspondant et le sous-graphe induit par la suppression du nœud concerné.

Algorithme 9 Actualiser(u, v, G_u, f_u, s, p)

```

1:  $f_s \leftarrow \text{Flot\_ex}(G_u, v, f_u)$ 
2: pour chaque arc  $(u, v) \in E[G_u]$  faire
3:    $f_r[u, v] \leftarrow f_u[u, v] - f_s[u, v]$ 
4:  $G_r \leftarrow G_u - \{v\}$ 
5:  $f_r \leftarrow \text{Recalcul\_flot\_max}(G_r, s, p, f_r)$ 
6:  $w[u, v] \leftarrow f_u[s, V[G_u - \{s}]] - f_r[s, V[G_r - \{s}]]$ 
7: si  $d[v] > d[u] + w[u, v]$  alors
8:    $d[v] \leftarrow d[u] + w[u, v]$ 
9:    $\pi[v] \leftarrow u$ 
10:  $f_v \leftarrow f_r$ 
11:  $G_v \leftarrow G_r$ 

```

Les lignes de 10 à 11 enregistrent le flot maximum et le graphe correspondant à la suppression du nœud ciblé. Ils seront utilisés comme le graphe et la fonction de flot de départ au moment du parcours d'un nœud adjacent, devenant G_u et f_u . Cette procédure prend un temps de $O(|E|)$ car la boucle "pour" (for), et la procédure de recalcul du flot sont limitées par le nombre d'arcs du graphe.

4.4.2.5 Sous-routine d'initialisation

L'algorithme de recherche de chemin est initialisé (algorithme 10) avec le calcul du flot maximum du graphe de flot G . Ensuite le flot résultant f_a causé par la suppression du nœud de départ a est calculé en appelant les sous-routines du flot excédentaire et de recalcul du flot maximum.

De même que pour l'algorithme de Dijkstra classique, une valeur infinie est assignée aux distances estimées vers chaque nœud, à exception du nœud de départ a .

Algorithme 10 Initialiser(G, s, p, a)

```

1:  $f_m \leftarrow \text{Ford-Fulkerson}(G, s, p)$ 
2:  $f_s \leftarrow \text{Flot\_ex}(G, a, f_m)$ 
3: pour chaque arc  $(u, v) \in E[G_u]$  faire
4:    $f_a[u, v] \leftarrow f_m[u, v] - f_s[u, v]$ 
5:  $G_a \leftarrow G - \{a\}$ 
6:  $f_a \leftarrow \text{Recalcul\_flot\_max}(G_a, s, p, f_a)$ 
7: pour chaque nœud  $v \in V[G]$  faire
8:    $d[v] \leftarrow \infty$ 
9:    $\pi[v] \leftarrow \text{NIL}$ 
10:  $d[a] \leftarrow 0$ 

```

4.4.2.6 Algorithme complet de recherche de chemin

Cet algorithme (algorithme 11) correspond à l'algorithme du chemin le plus court de Dijkstra mais qui fera appel aux sous-routines d'initialisation et d'actualisation présentées précédemment. Soit a et b les nœuds de départ et d'arrivée, respectivement, du chemin recherché.

Algorithme 11 Chemin_min_int(G, s, p, a, b)

```

1: Initialiser( $G, s, p, a$ )
2:  $Q \leftarrow V[G]$ 
3: tant que  $Q \neq \emptyset$  faire
4:    $u \leftarrow \text{Extraire\_min}(Q)$ 
5:    $S \leftarrow S - \{u\}$ 
6:   si  $u \neq t$  alors
7:     pour chaque nœud  $v \in \text{Adj}[u]$  faire
8:       Actualiser( $u, v, G_u, f_u, s, p$ )

```

L'algorithme de Dijkstra classique prend un temps de $O(|V|^2 + |E|)$, dont le temps $O|E|$ est produit par la procédure d'actualisation, laquelle a un temps asymptotique de $O(1)$ et est itérée $|E|$ fois. Dans l'algorithme proposé, chaque procédure d'actualisation prend un temps $O(|E|)$. Alors cet algorithme a un temps asymptotique de $O(|V|^2 + |E|^2)$.

Le chemin trouvé consiste en un ensemble de nœuds du graphe de flot. Ces nœuds ont une correspondance avec les nœuds du graphe non orienté original, et ces derniers peuvent être utilisés pour déterminer les vannes qu'il faudra ouvrir/fermer pour permettre le transport de pétrole entre les éléments.

4.4.2.7 Validation

L'algorithme de recherche d'un chemin maximisant la capacité opérative est basé sur l'algorithme de Dijkstra du plus court chemin pour des arcs à coûts positifs. À la différence de ce dernier, l'algorithme proposé ne dispose pas des valeurs des coûts

de chaque arc, mais elles doivent être déterminées de façon dynamique, pendant l'exécution de l'algorithme. La logique de fonctionnement de l'algorithme de Dijkstra reste valable, en raison des considérations suivantes :

- Coûts positifs ou nuls : la suppression d'un nœud du graphe ne peut pas augmenter la valeur du flot maximum. Les coûts des arcs étant égaux à la diminution du flot maximum, ces coûts ne peuvent pas être négatifs.
- Les coûts des arcs sont constants : comme chaque nœud est retiré de l'ensemble Q (avec la procédure Extraire_min) une seule fois pendant toute l'exécution de l'algorithme, alors chaque arc est examiné une seule fois pendant toute l'exécution de l'algorithme. Chaque nœud retiré de l'ensemble Q est un nœud dont le plus court chemin à partir de la source est connu. Alors même si le coût d'un arc dépend du parcours de recherche effectué par l'algorithme au moment d'examiner cet arc, ce parcours correspondra toujours à un chemin maximisant le flot. Les valeurs de coûts sont calculées de façon dynamique parce qu'elles ne peuvent pas être calculés sans connaître les coûts des autres arcs, mais ces coûts sont uniques pour une paire ordonnée de nœuds à connecter.

4.4.2.8 Exemple

La figure 4.24 montre un graphe non orienté avec un des possibles ensembles maximaux de chemins disjoints entre les ensembles de nœuds : $A = (a, b, c, d)$ et $B = (j, k, l, m)$. Cette possible solution au PMCDE a été trouvée en exécutant l'algorithme de calcul de la capacité opérative. Les chemins trouvés sont (a, h, j) , (b, f, k) , (c, g, l) , et (d, i, m) .

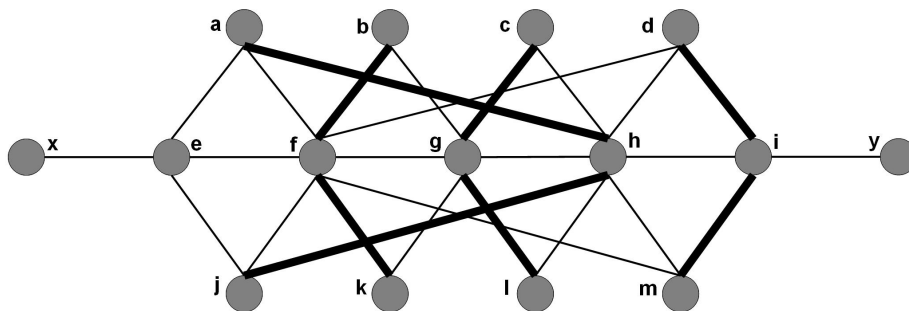


FIGURE 4.24 – Chemins disjoints (lignes épaisses) entre les ensembles : (a, b, c, d) et (j, k, l, m) déterminés avec l'algorithme de calcul de la capacité opérative, dans un graphe non orienté.

Un chemin connectant les nœuds x et y peut interférer avec les possibles chemins entre les ensembles A et B . Par exemple, le chemin (x, e, f, g, h, i, y) bloque complètement les chemins entre les deux ensembles. La figure 4.25 montre un chemin entre x et y qui a été trouvé avec l'algorithme de recherche de chemin maximisant la capacité opérative. Ce chemin interfère le moins possible avec les chemins entre

les ensembles A et B . Nous trouvons (x, e, a, h, i, y) , avec une capacité opérative résultante de deux. Cette valeur de capacité opérative indique qu'il est impossible de trouver plus de deux chemins disjoints entre A et B et n'interférant pas avec un chemin entre x et y . Les chemins (b, f, k) et (c, g, l) correspondent au flot maximum trouvé en appliquant l'algorithme de calcul de la capacité opérative dans le graphe dont nous avons supprimé les nœuds qui font partie du chemin (chaîne) entre x et y .

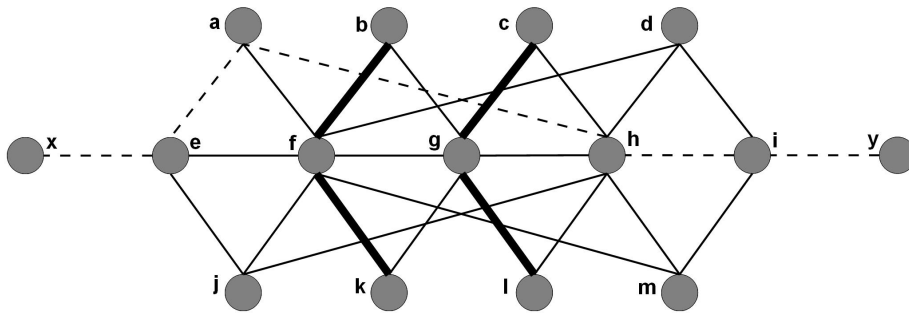


FIGURE 4.25 – Chemin entre x et y (pointillés) déterminé avec l'algorithme de recherche de chemin maximisant la capacité opérative pour les ensembles (a, b, c, d) et (j, k, l, m) . Les lignes épaisses correspondent à de possibles chemins disjoints résultants.

4.4.3 Opérations d'alignements forcés

Les alignements sont définis à partir d'une route connectant des éléments du système. Dans le cadre d'étude retenu ici nous considérons uniquement des alignements entre deux éléments. Les alignements avec des pompes constituent un type particulier d'alignement dans lequel la pression négative entre les éléments à connecter rend nécessaire l'utilisation de pompes pour surmonter cette pression et permettre le transport de pétrole de l'élément fournisseur vers l'élément récepteur. Normalement ces alignements utilisent une seule pompe. Alors les alignements avec des pompes correspondent à des chaînes ou chemins entre deux nœuds et passant obligatoirement par un nœud intermédiaire dans un sens donné. Ce sens doit permettre de transporter le pétrole de l'élément fournisseur vers l'élément récepteur. Dans la bibliographie, [Wu & Ruan, 2006] ont abordé un problème similaire de recherche du plus court chemin avec plusieurs nœuds intermédiaires obligatoires. Nous allons présenter un algorithme permettant de trouver un chemin entre deux nœuds et passant par une pompe spécifique dans le sens indiqué. Pour cela nous allons exploiter l'algorithme de recherche de chemin maximisant la capacité opérative présenté précédemment.

4.4.3.1 Modèle

Dans le cas du PMCDE, les pompes étaient considérées comme des segments interdits, et ces segments n'étaient pas représentés par des nœuds dans le graphe non orienté. Ici, le modèle du réseau doit subir des modifications pour tenir compte de la pompe choisie. En particulier, le modèle va devoir permettre de spécifier le sens de pompage. Ces modifications vont être appliquées au graphe de flot du réseau.

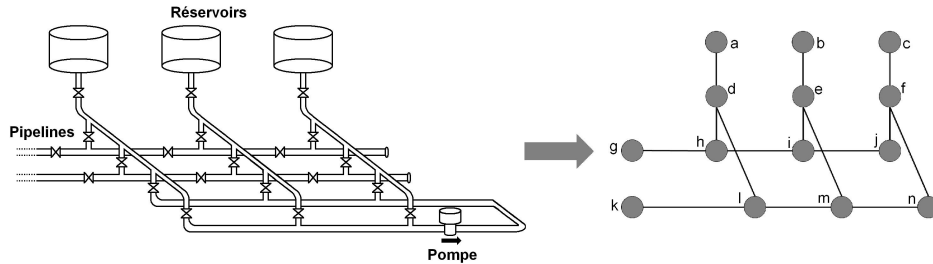


FIGURE 4.26 – Modèle de graphe non orienté (droite) d'un réseau de canalisation (gauche). Le nœud du segment correspondant à la pompe a été supprimé.

La figure 4.26 présente un réseau de canalisation contenant une pompe et le graphe non orienté correspondant. D'après la modélisation présentée précédemment le segment contenant la pompe n'est pas représenté dans le graphe.

Chaque pompe est située dans un segment du réseau. Ces segments sont connectés à d'autres segments au travers de vannes. Une pompe divise le segment dans lequel elle est située en deux parties : une partie en amont du sens de pompage, et une partie en aval du sens de pompage. Ainsi, chaque segment adjacent au segment de la pompe est : soit adjacent à la partie en amont du sens de pompage, soit adjacent à la partie en aval du sens de pompage. Nous allons appeler les premiers, les segments en amont de la pompe, et les deuxièmes, les segments en aval de la pompe. Dans le cas de la figure 4.26, les segments en amont de la pompe correspondent aux nœuds l , m et n du graphe non orienté, tandis que les segments en aval de la pompe, correspondent aux nœuds h , i et j .

Nous considérons un graphe de flot construit à partir du graphe non orienté du réseau, suivant la méthodologie présentée précédemment. Ce graphe de flot ne contient pas de nœud correspondant à des pompes. Pour la pompe choisie deux nœuds seront ajoutés au graphe de flot : un nœud en amont, sur lequel vont arriver des arcs à partir de chaque nœud de sortie correspondant à un segment en amont de la pompe ; et en nœud en aval, duquel vont sortir des arcs vers chaque nœud d'entrée correspondant à un segment en aval de la pompe. Un seul arc orienté reliera le nœud en amont au nœud en aval.

Soit un graphe de flot $F = (V, E)$ construit à partir du graphe non orienté d'un réseau de canalisation. Soient $M \subseteq V$ et $L \subseteq V$ les ensembles de nœuds en amont et en aval d'une pompe du réseau. L'algorithme 12 rajoute les nœuds permettant

de tenir compte de la pompe.

Algorithme 12 Ajout_Pompe(G, s)

```

1:  $H \leftarrow F$ 
2:  $V[H] \leftarrow V[H] \cup \{p^-\}$ 
3:  $V[H] \leftarrow V[H] \cup \{p^+\}$ 
4:  $E[H] \leftarrow E[H] \cup \{(p^-, p^+)\}$ 
5: pour chaque nœud  $x \in M$  faire
6:    $E[H] \leftarrow E[H] \cup \{(x^+, p^-\}$ 
7: pour chaque nœud  $y \in L$  faire
8:    $E[H] \leftarrow E[H] \cup \{(p^+, y^-\}$ 
9: retour H
  
```

Dans le cas de la figure 4.26, la liste d'adjacence suivante correspond au graphe de flot F construit à partir du graphe non orienté G du réseau, avec la procédure Graphe_de_flot(G, \emptyset, \emptyset) :

$a^- : (a^+)$	$h^- : (h^+)$
$a^+ : (d^-)$	$h^+ : (d^-, g^-, i^-)$
$b^- : (b^+)$	$i^- : (i^+)$
$b^+ : (e^-)$	$i^+ : (e^-, h^-, j^-)$
$c^- : (c^+)$	$j^- : (j^+)$
$c^+ : (f^-)$	$j^+ : (f^-, i^-)$
$d^- : (d^+)$	$k^- : (k^+)$
$d^+ : (a^-, h^-, l^-)$	$k^+ : (l^-)$
$e^- : (e^+)$	$l^- : (l^+)$
$e^+ : (b^-, i^-, m^-)$	$l^+ : (d^-, k^-, m^-)$
$f^- : (f^+)$	$m^- : (m^+)$
$f^+ : (c^-, j^-, m^-)$	$m^+ : (e^-, l^-, n^-)$
$g^- : (g^+)$	$n^- : (n^+)$
$g^+ : (h^-)$	$n^+ : (f^-, m^-)$

Pour la pompe du réseau de la figure 4.26, les ensembles des nœuds en amont et en aval sont les suivants :

$$M = (d, e, f)$$

$$L = (d, e, f)$$

L'exécution de la procédure Ajout_pompe(F, M, L) donne un graphe avec la liste d'adjacence suivante :

$a^- : (a^+)$	$h^+ : (d^-, g^-, i^-)$
$a^+ : (d^-)$	$i^- : (i^+)$
$b^- : (b^+)$	$i^+ : (e^-, h^-, j^-)$
$b^+ : (e^-)$	$j^- : (j^+)$
$c^- : (c^+)$	$j^+ : (f^-, i^-)$
$c^+ : (f^-)$	$k^- : (k^+)$
$d^- : (d^+)$	$k^+ : (l^-)$
$d^+ : (a^-, h^-, l^-, p^-)$	$l^- : (l^+)$
$e^- : (e^+)$	$l^+ : (d^-, k^-, m^-)$
$e^+ : (b^-, i^-, m^-, p^-)$	$m^- : (m^+)$
$f^- : (f^+)$	$m^+ : (e^-, l^-, n^-)$
$f^+ : (c^-, j^-, m^-, p^-)$	$n^- : (n^+)$
$g^- : (g^+)$	$n^+ : (f^-, m^-)$
$g^+ : (h^-)$	$p^- : (p^+)$
$h^- : (h^+)$	$p^+ : (d^-, e^-, f^-)$

4.4.3.2 Algorithme de recherche de chemin passant par une pompe

La modélisation qui vient d'être présentée, permet de poser le problème de recherche d'un chemin passant par une pompe spécifique de la façon suivante.

Soit $H = (V, E)$ un graphe de flot avec un ensemble de nœuds V , et un ensemble d'arcs E . Soit un nœud $x \in V$, et un nœud $y \in V$, avec $x \in y$. Soit le nœud $p^- \in V$ et $p^+ \in V$, deux nœuds liés par l'arc $e = (p^-, p^+) \in E$, avec $p^- \neq p^+$, $x \neq p^-$, $x \neq p^+$, $y \neq p^-$, $y \neq p^+$. Le problème consiste à trouver un chemin élémentaire (π_1, e, π_2) , π_1 étant un sous-chemin de x à p^- , et π_2 un sous-chemin de p^+ à y .

Dans ce problème, p^- et p^+ vont correspondre aux nœuds en amont et en aval du segment de la pompe. Le chemin (π_1, e, π_2) étant élémentaire, les sous-chemins π_1 et π_2 sont disjoints.

Comme nous avons vu précédemment, l'algorithme de recherche de chemin maximisant la capacité opérative permet de trouver un chemin entre deux nœuds donnés, qui maximise le nombre de chemins disjoints possibles entre deux ensembles de nœuds. Nous allons utiliser cet algorithme pour trouver une solution au problème posé. Pour cela, nous allons utiliser le modèle de graphe de flot, tenant compte de la pompe considérée.

Dans le problème de recherche d'alignement avec une pompe, nous allons considérer un cas particulier du problème du chemin maximisant la capacité opérative dans lequel les ensembles de nœuds correspondant à cette capacité opérative contiennent un seul élément (nœud) chacun. Soient a et b les nœuds à connecter, et soient S et T les ensembles entre lesquels la capacité opérative doit être maximisée. Le problème de trouver un chemin entre deux nœuds et passant par une pompe dans le sens correct peut être résolu en appliquant l'algorithme de recherche de chemin maximisant la capacité opérative en posant :

$$\begin{aligned}
 a &= x \\
 b &= p^- \\
 S &= \{p^+\} \\
 T &= \{y\}
 \end{aligned}$$

Un nœud source s , avec un arc sortant vers le nœud p^+ (ensemble S) et un nœud puits t , avec un arc provenant du nœud y (ensemble T), doivent être rajoutés pour pouvoir exécuter l'algorithme de recherche de chemin maximisant la capacité opérative : $\text{Chemin_min_int}(H, s, t, x, p^-)$.

Le problème admet une solution si et seulement si il existe un chemin entre a et b , maximisant la capacité opérative entre S et T , avec une capacité opérative résultante non nulle. Le chemin maximisant la capacité opérative va correspondre au sous-chemin π_1 du problème, tandis que le chemin entre S et T correspondant à la fonction de flot résultant va correspondre au sous-chemin π_2 du problème. L'alignement est donc déterminé à partir du chemin (π_1, e, π_2) , avec $e = (p^-, p^+)$.

4.4.3.3 Validation

L'algorithme proposé est une application, à un cas particulier, de l'algorithme de recherche de chemin maximisant la capacité opérative. Dans ce cas, le flot est maximum s'il est égal à un. La maximisation de la capacité opérative va correspondre à la minimisation des interférences entre le sous-chemin connectant l'élément fournisseur à la pompe et le sous-chemin connectant la pompe à l'élément récepteur.

4.4.3.4 Exemple

Dans le réseau de la figure 4.27, considérons le problème de recherche d'un alignement entre le premier réservoir (contenant du pétrole), et le deuxième réservoir (vide). Dans le cas du réseau de flot, il faut trouver un chemin entre a et b passant par le nœud p^- , puis par le nœud p^+ . En appliquant l'algorithme de recherche de chemin maximisant la capacité opérative, le sous-chemin π_1 trouvé est le suivant :

$$(a^-, a^+, d^-, d^+, p^-)$$

Le flot résultant correspondant au sous-chemin π_2 est le suivant :

$$(p^+, e^-, e^+, b^-, b^+)$$

Alors le chemins permettant de connecter les nœuds a et b à travers les nœuds p^- et p^+ est le suivant :

$$(a^-, a^+, d^-, d^+, p^-, p^+, e^-, e^+, b^-, b^+)$$

Le problème de cet exemple simple peut être résolu facilement sans utiliser un algorithme de recherche. Dans l'exemple suivant le modèle du réseau est légèrement modifié suite à une défaillance sur une vanne fermée. Dans ce cas, la solution n'est pas évidente. Dans des réseaux réels le problème peut devenir très complexe, d'où l'utilité de l'algorithme proposé.

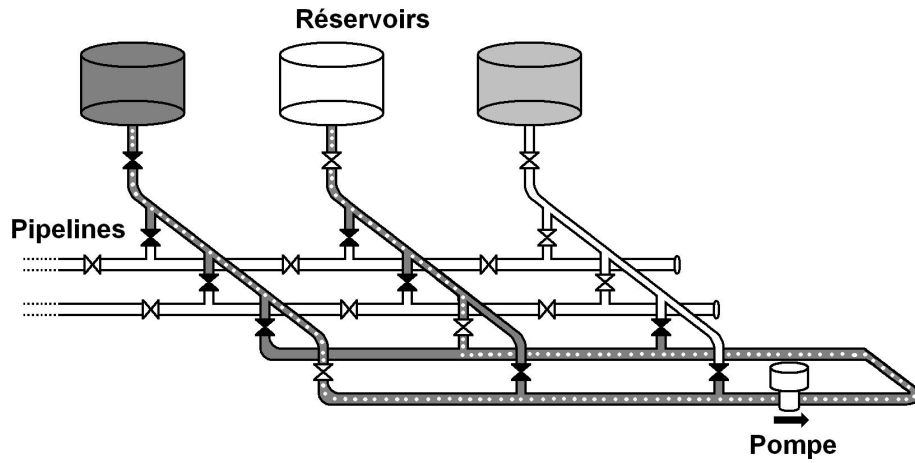


FIGURE 4.27 – Alignement entre le premier (gris foncé) et le deuxième réservoir (blanc) utilisant une pompe. Les vannes en noir sont fermées. Les points blancs montrent le flux de pétrole entre les réservoirs.

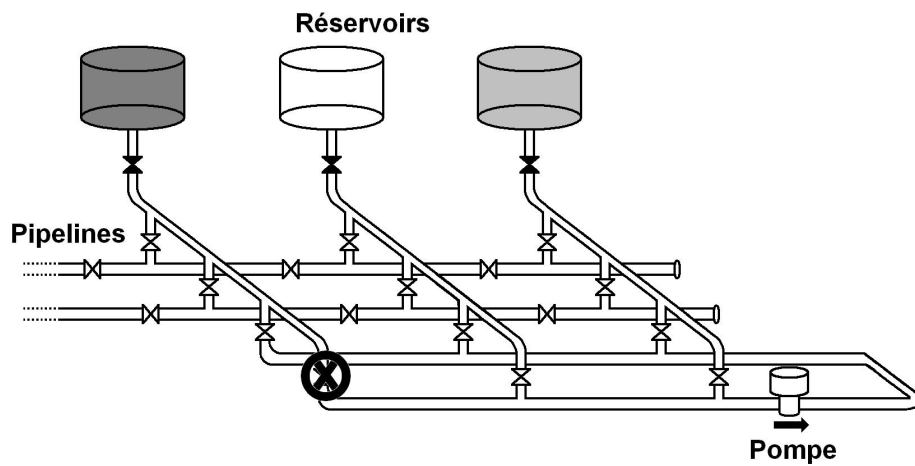


FIGURE 4.28 – Réseau de canalisation. Une vanne fermée présente une défaillance (X).

La figure 4.28 présente le même réseau que les figures 4.26 et 4.27, dans lequel la vanne connectant le segment du nœud d à la partie en amont de la pompe est une vanne fermée bloquante. Cette défaillance va avoir des conséquences sur le modèle de graphe de flot. L'arc connectant le nœud d^+ au nœud p^- est supprimé. La liste d'adjacences du graphe de flot devient alors :

$a^- : (a^+)$	$i^- : (i^+)$
$a^+ : (d^-)$	$i^+ : (e^-, h^-, j^-)$
$b^- : (b^+)$	$j^- : (j^+)$
$b^+ : (e^-, t)$	$j^+ : (f^-, i^-)$
$c^- : (c^+)$	$k^- : (k^+)$
$c^+ : (f^-)$	$k^+ : (l^-)$
$d^- : (d^+)$	$l^- : (l^+)$
$d^+ : (a^-, h^-, l^-, p^-)$	$l^+ : (d^-, k^-, m^-)$
$e^- : (e^+)$	$m^- : (m^+)$
$e^+ : (b^-, i^-, m^-, p^-)$	$m^+ : (e^-, l^-, n^-)$
$f^- : (f^+)$	$n^- : (n^+)$
$f^+ : (c^-, j^-, m^-, p^-)$	$n^+ : (f^-, m^-)$
$g^- : (g^+)$	$p^- : (p^+)$
$g^+ : (h^-)$	$p^+ : (d^-, e^-, f^-)$
$h^- : (h^+)$	$s : (p^+)$
$h^+ : (d^-, g^-, i^-)$	$t : --$

Dans cette liste, les nœuds s et t ont été rajouté, puisque nous considérons le même alignement entre le premier et le deuxième réservoir.

Appliquant l'algorithme de recherche de chemin maximisant la capacité opérative, le sous-chemin π_1 trouvé est le suivant :

$$(a^-, a^+, d^-, d^+, h^-, h^+, i^-, i^+, j^-, j^+, f^-, f^+, p^-)$$

Le flot résultant correspondant au sous-chemin π_2 est le suivant :

$$(p^+, e^-, e^+, b^-, b^+)$$

Alors le chemin permettant de connecter les nœuds a et b à travers les nœuds p^- et p^+ est le suivant :

$$(a^-, a^+, d^-, d^+, h^-, h^+, i^-, i^+, j^-, j^+, f^-, f^+, p^-, p^+, e^-, e^+, b^-, b^+)$$

La figure 4.29 présente l'alignement correspondant à ce chemin.

4.5 Conclusion

Dans ce chapitre nous avons présenté les principales contributions de cette thèse dans le cadre d'opérations de transfert de pétrole. Premièrement nous avons lié la problématique des alignements, présentée dans le chapitre 2, à des problèmes mathématiques, appartenant spécifiquement au domaine de la recherche opérationnelle.

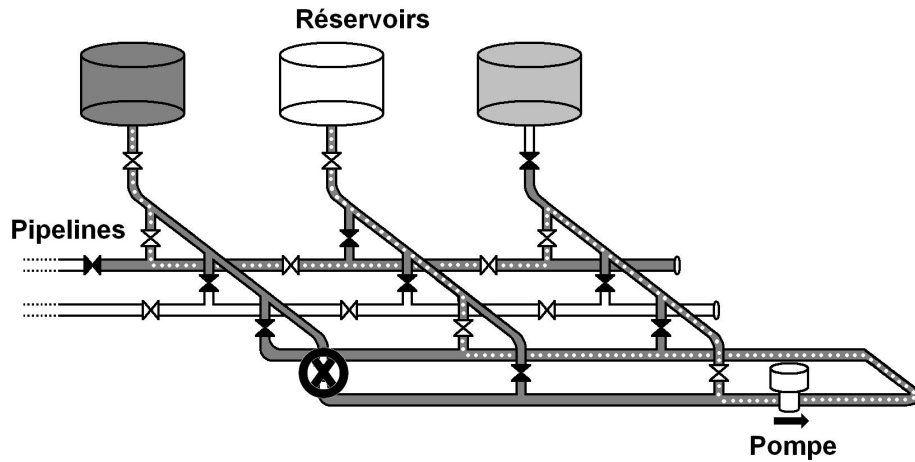


FIGURE 4.29 – Alignement entre le premier (gris foncé) et le deuxième réservoir (blanc) utilisant une pompe. Les vannes en noir sont fermées. Les points blancs montrent le flux de pétrole entre les réservoirs. Une vanne fermée présente une défaillance (X).

Ce lien repose sur une étude des critères de choix utilisés par les opérateurs dans leur activité de supervision, basés sur leur expertise ; permettant d'aboutir à un modèle exploitable reposant sur la théorie des graphes. Le contexte opérationnel a souligné l'importance de répondre à des demandes multiples et simultanées de transfert de pétrole, imposant ainsi comme critère principal un fort degré d'indépendance désigné ici par la capacité opérative. Nous avons ainsi défini formellement la capacité opérative pour pouvoir quantifier cette notion d'interférence entre les alignements.

La définition de la capacité opérative permet de quantifier l'impact des opérations d'alignement ou des défaillances sur le réseau. Un algorithme de calcul de la capacité opérative est présenté. Cet algorithme de base classique a été étendu de façon à disposer d'une modélisation plus aisée de contraintes spécifiques à la problématique des transferts en considérant l'existence de sous-types d'alignement.

Ainsi, en nous inspirant des indices de centralité utilisés pour mesurer l'importance selon différents critères des nœuds dans un graphe, nous avons proposé des notions de criticité, permettant de mesurer l'impact des opérations ou des défaillances sur certains éléments d'un réseau.

Un algorithme de recherche de chemin maximisant la capacité opérative a été proposé. Il utilise une méthode innovante, basée sur l'utilisation d'un algorithme de Dijkstra dans lequel les coûts des arcs sont calculés de façon dynamique et qui dépendent de l'interférence des nœuds ciblés avec le flot considéré. En plus de l'application de cet algorithme à la problématique de recherche d'un alignement interférant le moins possible avec d'autres alignements, cet algorithme permet d'aborder d'autres problèmes d'alignement. Nous présentons une méthode utilisant cet algorithme pour

trouver un alignement forcé.

Ces contributions représentent des développements technologiques très concrets en supervision. D'autre part, la formalisation des problèmes et des algorithmes présentés permettent d'étendre les champs d'application de ces contributions à d'autres problématiques liés au transport, et même à d'autres domaines ayant un rapport avec la théorie des graphes.

Le chapitre suivant s'adresse à la mise en œuvre de ces algorithmes sur le réseau réels, présentés au chapitre 2, et l'analyse des résultats. À la différence de ce chapitre, dans lequel nous avons présenté des contributions générales, le chapitre suivant s'adresse à des problèmes techniques, tenant compte des contraintes imposées par le fonctionnement réel des réseaux de canalisation, de l'entreprise PDVSA et des systèmes SCADA.

Mise en œuvre

Résumé : Les contributions de cette thèse ont été testées sur des données réelles de deux sites industriels de PDVSA : le système de transport de pétrole du terminal maritime de Guaraguao, à Puerto La Cruz, et le centre de stockage de pétrole de Punta de Palmas, à proximité de la ville de Maracaibo. L’objectif à moyen terme de nos travaux est le développement d’un logiciel permettant d’assister les opérateurs des systèmes de transport de pétrole. Les caractéristiques de ce logiciel ont été choisies en considérant les exigences et les intérêts de l’entreprise. Il faudra prendre en compte la législation vénézuélienne, ainsi que les moyens de saisie des données et la compatibilité avec d’autres logiciels déjà utilisés dans l’entreprise ou en cours de développement par l’entreprise. Ici nous montrons les premières étapes de développement de ce logiciel, comprenant la mise en œuvre des algorithmes présentés précédemment, et proposant une architecture de base du logiciel.

Sommaire

5.1	Langage de programmation	93
5.2	Codage de la structure du réseau de canalisation	94
5.3	Codage des algorithmes	101
5.4	Résultats des tests	103
5.4.1	Terminal Maritime de Guaraguao	104
5.4.2	Centre de stockage de Punta de Palmas	106
5.5	Préconisations pour le développement d’un logiciel d’assistance aux opérateurs	107
5.6	Conclusion	110

5.1 Langage de programmation

Parmi les nombreux langages de programmation existants, nous avons choisi le langage Python, créé dans les années 80s par Guido Van Rossum [Python docs, 2008]. Il s’agit d’un langage de programmation de haut niveau qui favorise la programmation impérative structurée et orientée objet. Parmi les caractéristiques de ce langage que nous avons considéré adéquats pour notre projet se trouvent :

- **Licence Libre :** au Venezuela, en décembre de 2004, le décret présidentiel numéro 3390 a conféré au “Software Libre” la priorité dans les systèmes, les projets et les services de l’administration publique. Ainsi, bien qu’une quantité

très importante des sites industriels de PDVSA utilisent des logiciels de licence privative, les logiciels de licence libre sont de plus en plus utilisés et plusieurs projets actuels de l'entreprise ciblent le développement de logiciels de licence libre. La Python Software Foundation License est une licence libre à laquelle sont soumis les logiciels codés avec Python.

- **Intégration dans d'autres langages** : Python n'est pas le langage de conception de la plupart des logiciels utilisés dans l'entreprise. Néanmoins, le code écrit sous Python peut être intégré (embedded) dans d'autres langages de programmation tels que C et C++. Cela permet de créer des applications compatibles avec la plupart des logiciels et en particulier avec le logiciel SCADA développé par PDVSA, Guardián del Alba, lequel est codé avec les langages C++ et Java.
- **Langage de haut niveau** : Python est un langage de haut niveau, avec une gestion automatique de la mémoire et un système de gestion d'exceptions qui facilite la conception, la compréhension et les possibles modifications du code. Cet aspect est important pour la mise en œuvre d'algorithmes complexes et pour le développement futur du logiciel.

5.2 Codage de la structure du réseau de canalisation

Les modèles de graphes présentés dans le chapitre 3, et le choix des listes d'adjacence comme structure de données pour ces graphes, sont pris en compte lors du codage des données du réseau. Pour une mise en œuvre en Python, nous utilisons une classe d'objet, appelée Graphe, pour créer des objets qui vont représenter les réseaux de canalisation :

```
class Graphe:

    def __init__(self):
        self.adj, self.f, self.adj_s = {}, {}, {}
```

En Python, un dictionnaire (dictionary) est une structure de données dans laquelle des "clés" sont utilisées pour accéder à des données. Une clé peut correspondre à un nœud, ou à un arc, permettant d'accéder à une liste de nœuds adjacents, ou à la capacité, le flot ou le poids d'un arc. Cette structure permet de créer des listes d'adjacence pour les graphes correspondants au réseau de canalisation, et des fonctions discrètes pour les fonctions de flot. Les objets Graphe sont composés de trois dictionnaires :

- **adj_s** : correspond à la liste d'adjacence du modèle de graphe non orienté du réseau.
- **adj** : correspond à la liste d'adjacence du modèle de graphe de flot utilisé pour l'algorithme de calcul de la capacité opérative.
- **f** : correspond à la fonction de flot du graphe de flot.

Les listes d'adjacence sont construites à travers des méthodes ajoutant des nœuds et des arêtes. Dans le cas du graphe non orienté, l'ajout d'un nœud (`agr_n`) correspond à l'ajout d'une clé dans le dictionnaire "adjs", et à la création d'une liste de données vide. L'ajout d'une arête (`agr_a`) correspond à ajouter chaque nœud dans la liste de données du nœud adjacent. Les arêtes étant des couples non ordonnés, l'ordre dans lequel les nœuds d'une arête sont saisis n'est pas important. Par contre, il est nécessaire que les nœuds soient ajoutés au dictionnaire avant d'ajouter les arêtes qui les connectent. Il est donc préférable d'ajouter tous les nœuds du graphe avant d'ajouter les arêtes.

```
def agr_n (self, v):
    self.adjs[v]=[]

def agr_a (self, u, v):
    self.adjs[u].append(v)
    self.adjs[v].append(u)
```

Dans le cas du graphe de flot, l'ajout des nœuds (`add_v`) est identique au cas du graphe non orienté. Par contre, l'ajout d'un arc (`add_e`) tient compte de la capacité de l'arc (`w`). S'agissant d'arcs orientés, l'ordre dans lequel les nœuds de l'arc sont saisis est très important. Le premier nœud (`u`) correspond à l'extrémité initiale, tandis que le deuxième (`v`) correspond à l'extrémité terminale. Un couple de données (`v,w`), contenant l'extrémité terminale et la capacité de l'arc, est ajouté à la liste d'adjacence du nœud correspondant à l'extrémité terminale. Un couple de données (`u,0`) contenant l'extrémité initiale, et une capacité nulle, est ajouté à la liste de données du nœud d'arrivée. Cela sera très utile lors de la recherche de nœuds adjacents dans le graphe de flot, et lors du calcul du flot résiduel, pour les algorithmes utilisant le graphe de flot du réseau. L'arc est aussi rajouté au dictionnaire "f", correspondant à la fonction de flot du graphe de flot. La valeur du flot d'un arc est nulle par défaut. De la même façon que pour le graphe non orienté, les nœuds doivent être ajoutés avant les arcs qui les connectent.

```
def add_v (self, v):
    self.adj[v]=[]

def add_e (self, u,v,w):
    self.adj[u].append((v,w))
    self.adj[v].append((u,0))
    self.f[(u,v)] = self.f[(v,u)] = 0
```

Avec ces méthodes il est possible de stocker les données d'un graphe non orienté, ou d'un graphe de flot, sous forme de liste d'adjacence. Pour cela il faut créer un objet de type graphe, puis ajouter les nœuds et les arcs (arêtes) avec les méthodes présentées.

Saisies des données du réseau

Actuellement les systèmes de transport de pétrole de PDVSA opèrent avec divers logiciels, lesquels souvent sont incompatibles. Les affichages, les diagrammes et les synoptiques disponibles pour saisir les données des réseaux ont chacun une logique et une nomenclature différente. La saisie des données est une étape fastidieuse qui accapare l'opérateur et pour laquelle il doit prêter une attention bien particulière. La mise en œuvre de notre proposition impose la résolution de certains de ces points délicats pour deux types de réseaux très différents du point de vue de la complexité.

Réseaux simples : Punta de Palmas

Le réseau de canalisation de Punta de Palmas est le cas le plus simple. Ses éléments caractéristiques peuvent être saisis manuellement à partir des écrans d'affichage et des synoptiques disponibles, ou même en se rendant sur le site, car les canalisations sont clairement visibles.

La figure 5.1 montre une section d'une des affichages du réseau de Punta de Palmas.

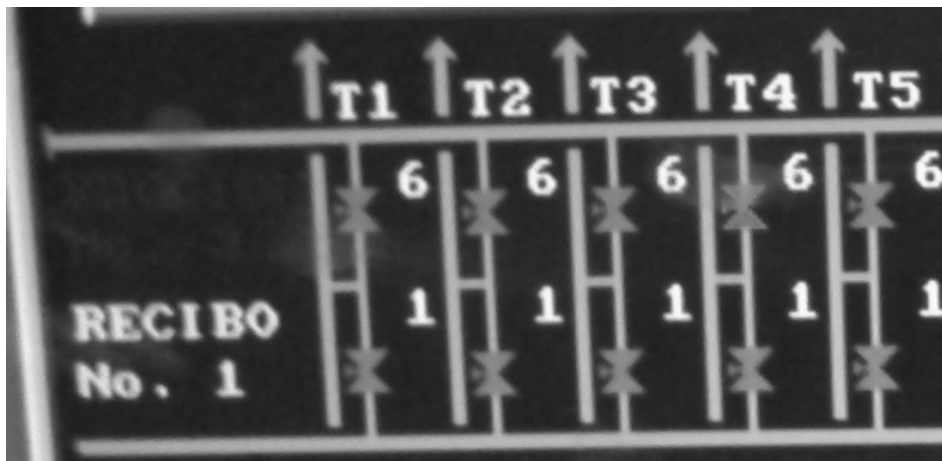


FIGURE 5.1 – Section du multiple principal du réseau de Punta de Palmas.

Cette section montre les connexions entre les lignes de succion 3 (Succion No. 3 - étiquette difficilement visible sur l'image) et de réception 3 (Recibo No. 1); avec plusieurs réservoirs (T1, T2, T3, T4, T5). Chaque ligne est connectée à travers une vanne aux segments des réservoirs. Les données sont saisies avec le code suivant :

```
Section = Graphe()
```

```
Section.agr_n('Succion_3')
```

```
Section.agr_n('Recibo_1')
```

```
Section.agr_n('T_1')
```

```
Section.agr_n('T_2')
```

```
Section.agr_n('T_3')
Section.agr_n('T_4')
Section.agr_n('T_5')

Section.agr_a('Succion_3','T_1')
Section.agr_a('Succion_3','T_2')
Section.agr_a('Succion_3','T_3')
Section.agr_a('Succion_3','T_4')
Section.agr_a('Succion_3','T_5')

Section.agr_a('Recibo_1','T_1')
Section.agr_a('Recibo_1','T_2')
Section.agr_a('Recibo_1','T_3')
Section.agr_a('Recibo_1','T_4')
Section.agr_a('Recibo_1','T_5')
```

Nous avons ajouté des méthodes dans la classe “Graphe” qui facilitent la saisie des données.

Par exemple, la méthode multiple permet d’ajouter automatiquement les nœuds et les arcs d’un multiple comme celui de Punta de Palmas :

```
def multiple (self, lignes, reservoirs):
    for l in lignes:
        self.agr_n(l)
    for r in reservoirs:
        self.agr_n(r)
    for l in lignes:
        for r in reservoirs:
            self.agr_a(l,r)
```

Réseaux complexes : Guaraguao

Dans des réseaux plus complexes, nous avons trouvé que la probabilité d’erreur dans la saisie manuelle des données pose un problème important. Le réseau de canalisation étant très complexe, le logiciel de supervision ne présente aux opérateurs que des sections du réseau dans chaque affichage (display). L’opérateur visualise dans chaque section les éléments à surveiller/commander, tels que des réservoirs, des vannes et des pompes, lesquels sont étiquetés (label). Des liens entre les affichages permettent aux opérateurs d’accéder aux affichages des sections adjacentes. La figure 5.2 présente une affichage du réseau de Guaraguao dans lequel il est possible de repérer les étiquettes des vannes, des pompes (en bas), et des canalisations d’entrée/sortie (à droite). Les mains à gauche et à droite sont des liens vers d’autres affichages.

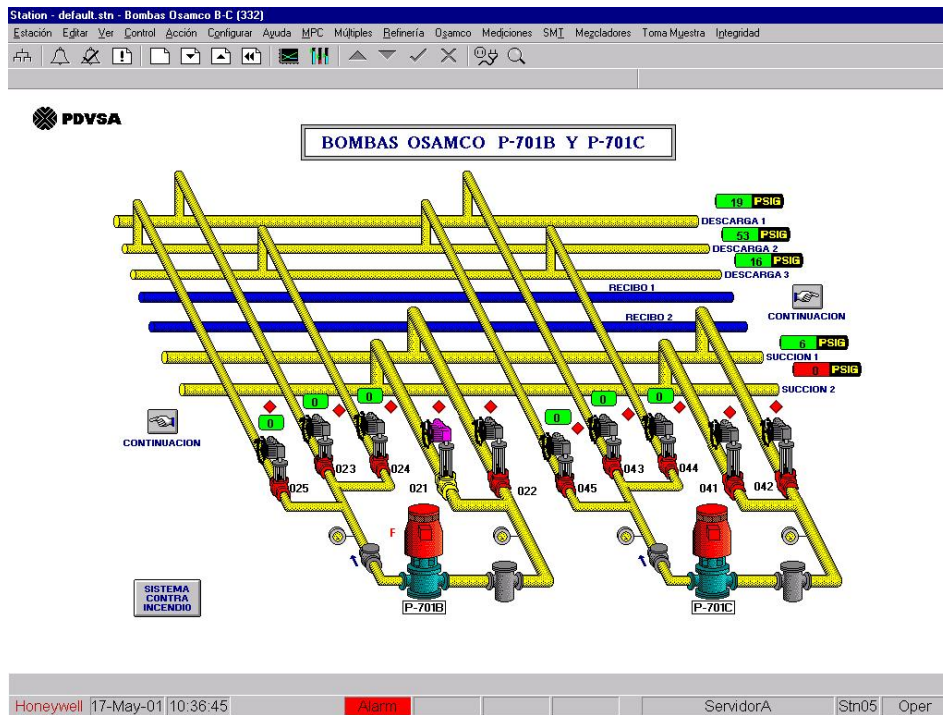


FIGURE 5.2 – affichage d'une section du réseau de canalisation de Guaraguao.

Avec le système actuel, un même segment peut apparaître dans plusieurs visualisations différentes, plusieurs segments peuvent apparaître avec la même étiquette, et il existe une grande quantité de segments sans aucune étiquette permettant de les identifier. Du fait de ces inconvénients, nous avons développé une méthode de saisie automatique basée sur les éléments ayant une étiquette d'identification. Ces éléments correspondent aux vannes, réservoirs, pompes, bras de chargement, et canalisations de connexion (appelées d'entrée/sortie) vers d'autres affichages. À partir de la saisie de ces éléments, les nœuds et les arcs du graphe sont générés automatiquement. Nous avons trouvé que cette méthode permet de réduire les erreurs de saisie.

Dans le type de réseau considéré, chaque segment peut être identifié sans ambiguïté par les éléments étiquetés auxquels il est directement connecté. En effet, le seul cas dans lequel ce type d'identification pourrait poser problème est un réseau composé de deux segments, ce qui n'est pas réaliste. La figure 5.3 montre un exemple de ce cas théorique : avec la nomenclature proposée, les deux segments sont identifiés avec le même ensemble de vannes (101,102,103).

Dans les réseaux réels, les étiquettes des éléments adjacents à un segment permettent d'identifier ce segment sans ambiguïté. La méthode de saisie présentée permet à l'utilisateur de saisir les informations de la topologie du réseau d'une façon simple et méthodique, bien que pénible. L'utilisateur saisit tout d'abord l'ensemble des étiquettes des éléments (étiquetés) présentes dans un affichage (correspondant à

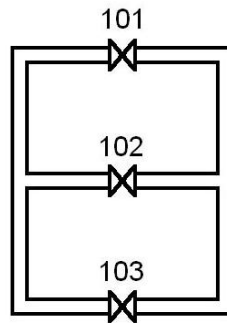


FIGURE 5.3 – Réseau théorique avec seulement deux segments connectés aux mêmes éléments étiquetés.

une section du réseau). Parmi ces éléments, il peut exister des éléments terminaux, chacun connecté à d'autres éléments par une seule extrémité. C'est le cas des réservoirs, des bras de chargement, et des canalisations d'entrée/sortie. Il existe des éléments intermédiaires, chacun connecté à d'autres éléments par ses deux extrémités. C'est le cas des vannes et des pompes. Pour chaque élément terminal, l'utilisateur va saisir les étiquettes des éléments adjacents. Pour chaque élément intermédiaire, l'utilisateur va saisir dans un premier temps les étiquettes des éléments adjacents à une des extrémités, et dans un deuxième temps les étiquettes des éléments adjacents à l'autre extrémité. Il y a ainsi redondance de certaines saisies, puisque chaque connexion entre vannes sera prise en compte plusieurs fois.

La démarche suivante permet de réaliser cette saisie et de stocker ces données pour une section du réseau correspondant à un affichage :

1. Saisir la quantité totale d'éléments étiquetés de l'affichage.
2. Saisir l'étiquette de chaque élément (étiqueté).
3. Pour chaque élément :
 - (a) Saisir la quantité d'éléments étiquetés adjacents à un côté de l'élément.
 - (b) Saisir les étiquette des éléments adjacents à un côté de l'élément.
 - (c) Saisir la quantité d'éléments étiquetés adjacents à l'autre côté de l'élément.
 - (d) Saisir le label des éléments étiquetés adjacents à l'autre côté de l'élément.

La méthode "saisir_reseau" de la classe Graphe demande ces données à l'utilisateur et crée un dictionnaire contenant ces informations. Pour chaque élément étiqueté, le dictionnaire donne soit une liste d'adjacence, s'il s'agit d'un élément terminal, soit deux listes d'adjacence, s'il s'agit d'un élément intermédiaire.

La figure 5.4 montre un réseau très simple dans lequel la méthode "saisir_reseau" est appliquée.

La table 5.1 montre les données saisies pour chaque élément étiqueté. Pour chaque élément, le nombre d'éléments étiquetés adjacents de chaque côté (A et

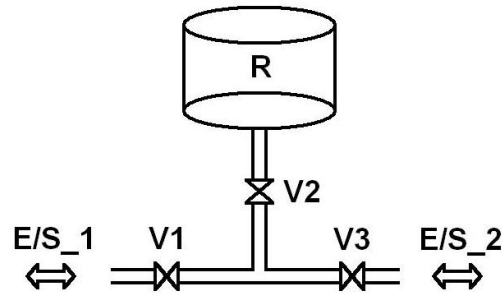


FIGURE 5.4 – Réseau simple avec 6 éléments étiquetés : les vannes V1, V2 et V3, le réservoir R, et les canalisations d’entrée/sortie E/S_1 et E/S_2.

Élément	N° A	Étiquettes A	N° B	Étiquettes B
E/S_1	0	-	1	V1
V1	1	E/S_1	2	V2,V3
R	0	-	1	V2
V2	1	R	2	V1,V3
V3	2	V1,V2	1	E/S_2
E/S_2	1	V3	0	-

TABLE 5.1 – Saisie des étiquettes d’un réseau de canalisation.

B) est saisi, puis les étiquettes de ces éléments pour chaque côté (A et B).

La méthode retourne le dictionnaire suivant : 'E/S_2' : [['V3']], 'E/S_1' : [['V1']], 'V3' : [['V1', 'V2'], ['E/S_2']], 'V1' : [['E/S_1'], ['V2', 'V3']], 'V2' : [['R'], ['V1', 'V3']], 'R' : [['V2']]

À partir de cette liste, la méthode “nommer” permet de créer une étiquette pour chaque segment. Nous avons choisi de nommer les segments en concaténant les étiquettes des éléments étiquetés correspondant à ce segment, dans l’ordre alphabétique/numérique, et séparées par un tiré. Par exemple, le segment correspondant à l’ensemble (vanne1, vanne5, reservoir4) est identifié par l’étiquette “vanne1-vanne5-reservoir4”. La méthode retourne un dictionnaire avec les étiquettes de chaque segment, et les éléments adjacents à chaque segment. Pour l’exemple de la figure 5.4, cette méthode retourne le dictionnaire suivant : ('E/S_1-V1' : ['E/S_1', 'V1'], 'V1-V2-V3' : ['V1', 'V2', 'V3'], 'R-V2' : ['R', 'V2'], 'E/S_2-V3' : ['E/S_2', 'V3'])

La méthode “creer_graphe” va construire les nœuds du graphe correspondant à chaque affichage à partir des étiquettes de chaque segment, et va rajouter des arêtes de connexion entre ces nœuds à partir des listes d’adjacence des éléments étiquetés. Les segments correspondant à des pompes ne sont pas rajoutés au modèle sous forme de nœuds (il faudra les rajouter en cas d’utilisation de l’algorithme de recherche de chemin passant par une pompe). Pour l’exemple de la figure 5.4, cette méthode retourne un graphe avec le dictionnaire “adjs” suivant : ('E/S_1-V1' : ['V1-

V2-V3'], 'E/S_2-V3' : ['V1-V2-V3'], 'R' : ['V1-V2-V3'], 'V1-V2-V3' : ['E/S_1-V1', 'E/S_2-V3', 'R'])

La saisie est faite en créant un graphe pour chaque affichage, et puis regroupant ces graphes en un seul et connectant les éléments d'entrée/sortie des affichages. Les canalisations d'entrée/sortie connectant deux affichages correspondent à un même segment, et souvent ils ont une même étiquette. Alors le regroupement de ces affichages doit se faire en fusionnant ces segments. La méthode "fusion_segments" permet de réaliser cette tâche.

Le graphe résultant, correspondant au réseau de Guaraguao, contient des milliers de nœuds et d'arêtes.

5.3 Codage des algorithmes

Les algorithmes proposés dans le chapitre précédent ont été codés sous forme de méthodes de la classe "Graphe". Ces méthodes utilisent les données stockées dans les dictionnaires d'un objet de classe graphe, et retournent des résultats bruts sous forme de listes d'éléments, souvent difficiles à comprendre. Des méthodes complémentaires sont utilisées pour traiter ces données et faciliter leur compréhension. Concrètement, quatre algorithmes ont été testés :

- Algorithme de calcul de la capacité opérative
- Algorithme de détermination d'éléments critiques
- Algorithme de recherche de chemin maximisant la capacité opérative
- Algorithme de recherche de chemin passant par une pompe

Ces algorithmes fonctionnent à partir d'une modélisation du réseau sous forme de graphe de flot. La construction du graphe de flot est réalisée par plusieurs méthodes de la classe "Graphe".

Construction du graphe de flot

La méthode "graphe_de_flot", construit la partie fortement connexe du graphe de flot du réseau à partir du graphe non orienté. Suite à la saisie (manuelle ou automatique) des données du réseau, les données du graphe non orienté se trouvent dans le dictionnaire "adjs" de l'objet de classe "Graphe" correspondant. En exécutant la méthode "graphe_de_flot", la partie fortement connexe (excluant les nœuds source et puits) est automatiquement construite et stockée dans le dictionnaire "adj". Le même objet contient alors le dictionnaire correspondant au graphe non orienté et le dictionnaire correspondant au graphe de flot. La méthode "sources" permet de rajouter un nœud source connecté à chaque nœud des éléments fournisseurs. La méthode "puits" rajoute un nœud puits connecté à chaque nœud des éléments receveurs. Ces deux méthodes déterminent le type d'alignement considéré pour le calcul de la capacité opérative. Avec les nœuds source et puits le graphe de flot est complet. Le code suivant montre un exemple de code créant le graphe de flot pour un objet G de classe Graphe correspondant au réseau de canalisation de Punta de Palmas.


```
G.graphe_de_flot ()
G.sources('s', ['Recibo_No_1', 'Recibo_No_2', 'Recibo_No_3'])
G.puits('p', ['T1', 'T2', 'T3', 'T4', 'T5', 'T6', 'T7', 'T8', 'T9'])
```

Dans ce cas, les lignes “Recibo_No_1”, “Recibo_No_2” et “Recibo_No_3” correspondent aux éléments fournisseurs, et les 9 réservoirs (T1, T2, ...) correspondent aux éléments receveurs.

Plusieurs nœuds source et puits peuvent être rajoutés au graphe de flot pour tenir compte de chaque type d’alignement. Lors de l’exécution des algorithmes il faudra spécifier les paires source/puits désirées.

Un graphe de flot tenant compte des sous-types d’alignement peut être construit utilisant les méthodes de construction basiques “add_v” et “add_e”. Le code suivant correspond à un graphe de flot tenant compte de deux sous-types d’éléments receveurs.

```
G.graphe_de_flot ()
G.sources('s', ['Recibo_No_1', 'Recibo_No_2', 'Recibo_No_3'])
G.puits('p1', ['T1', 'T2', 'T3', 'T4', 'T5', 'T6'])
G.puits('p2', ['T7', 'T8', 'T9'])
G.add_v('p')
G.add_e('p', 'p1', 2)
G.add_v('p', 'p2', 1)
```

Le premier sous-type est limité à deux alignements, tandis que le deuxième type est limité à un alignement.

Algorithme de calcul de la capacité opérative

Une fois le graphe de flot créé, il suffit d’exécuter la méthode “flot_max” pour obtenir la valeur du flot maximum, laquelle correspond à la valeur de la capacité opérative pour le type d’alignement considéré. Cette méthode modifie les valeurs de la fonction de flot stockées dans le dictionnaire “f”. Ces valeurs représentent un des possibles ensembles maximaux d’alignements. Cependant, la méthode “flot_max” retourne uniquement une valeur entière correspondant à la capacité opérative, et ne retourne pas les chemins correspondants à l’ensemble maximal d’alignements. La méthode “determiner_chemins” permet de déterminer ces chemins à partir de la fonction de flot du graphe et de présenter ces chemins à l’utilisateur sous forme de listes de nœuds.

Algorithme de détermination d’éléments critiques

L’analyse de criticité présentée tient compte de deux types d’éléments : des segments et des vannes. Dans le cas des vannes, la criticité peut être de type bloquante, ou de type permissive. La méthode “arcs_critiques” détermine et retourne, sous forme de liste, les arêtes du graphe non orienté correspondant à des vannes critiques bloquantes. La méthode “arcs_critiques_fusion” détermine et retourne sous

forme de liste, les arêtes du graphe non orienté correspondant à des vannes critiques permissives. Dans le cas des segments, la criticité est uniquement de type bloquante. La méthode “nœuds_critiques” détermine et retourne, sous forme de liste, les nœuds du graphe non orienté correspondant à des segments critiques.

Algorithme de recherche de chemin maximisant la capacité opérative

La méthode “chemin_min_int” permet de calculer le chemin maximisant la capacité opérative pour un type d’alignement donné. Ce chemin minimise les inférences avec les chemins disjoints entre deux ensembles de nœuds, correspondant aux nœuds source et puits choisis. La méthode retourne le chemin sous forme de liste de nœuds.

Algorithme de recherche de chemin passant par une pompe

La méthode “chemin_pompe” détermine un chemin entre deux nœuds du graphe, passant par un arc intermédiaire obligatoire. Avant l’exécution de cette méthode l’utilisateur doit rajouter les nœuds correspondant à la pompe choisie, avec la méthode “add_pompe”, pour laquelle il doit saisir les ensemble M et L de nœuds en amont et en aval de la pompe.

Présentation des résultats

Les alignements déterminés par les méthodes présentées sont retournés sous forme de liste de nœuds. Cependant, l’information pertinente pour les opérateurs correspond aux vannes qu’il faudra ouvrir et fermer pour établir ces alignements. La méthode “determiner_vannes_chemin” retourne la liste des vannes faisant partie de la route d’un alignement, lesquelles seront ouvertes pour établir l’alignement. La méthode “determiner_vannes_isolement” retourne la liste des vannes entourant la route d’un alignement, lesquelles seront fermées pour établir l’alignement.

5.4 Résultats des tests

L’exécution des algorithmes à travers les méthodes de la classe "Graphe" a été réalisée pour les deux réseaux de transport de pétrole étudiés. Nous avons étudié des situations habituelles de transport de pétrole, ainsi que des situations extraordinaires se traduisant par l’indisponibilité de plusieurs éléments du réseau. Les résultats étant souvent de longues listes d’éléments des réseaux, nous allons présenter quelques exemples et remarques pertinentes sur le temps de calcul, et la cohérence des résultats. Nous avons utilisé un ordinateur de 1,6 GHz et 1Go de mémoire RAM. Le code utilisé est disponible en annexe (A).

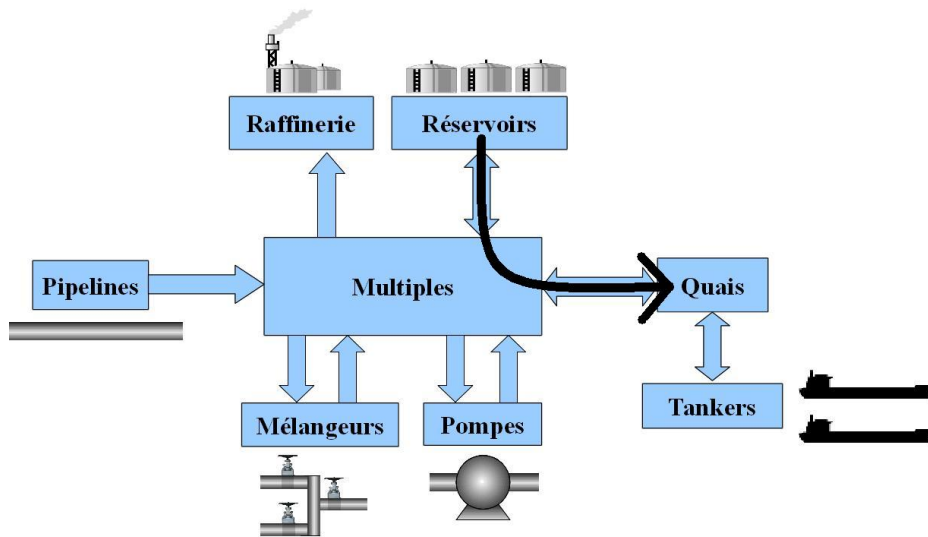


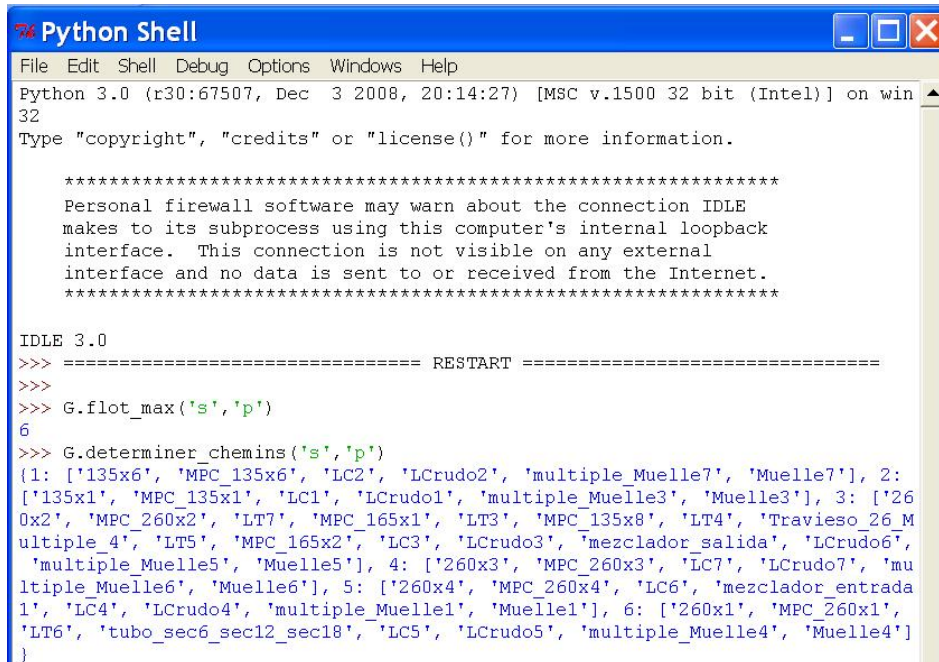
FIGURE 5.5 – Le calcul de la capacité opérative correspond à des alignements entre des réservoirs et les bras de chargement des quais.

5.4.1 Terminal Maritime de Guaraguao

Pour faciliter la compréhension des résultats, nous avons travaillé avec un graphe “G” correspondant à une partie du réseau de canalisation de Guaraguao. La complexité de ce réseau peut être appréhendée considérant qu’il est représenté par plus de 100 affichages du même type que celui de la figure 5.2 Ici les étiquettes des segments ne correspondent pas à la méthodologie de saisie automatique présentée précédemment, mais nous utilisons des étiquettes plus simples créées de façon manuelle. Les figures présentent les résultats de l’exécution des méthodes pour le graphe G utilisant l’interpréteur de Python. La plupart correspondent à des alignements connectant des réservoirs à des bras de chargement (figure 5.5).

La figure 5.6 montre les résultats de l’exécution des méthodes “flot_max” et “determiner_chemin” pour le graphe “G” et des alignements entre des réservoirs du “Multiple Principal de Chargement” (Multiple Principal de Carga) et les bras de chargement des quais. Chacune de ces deux méthodes a pris un temps d’exécution de moins d’une seconde. La fonction de ces réservoirs étant principalement le chargement des tankers, les routes des alignements trouvés sont assez courtes. La capacité opérative (6) est limitée par la quantité de bras de chargement de pétrole.

La figure 5.7 montre les résultats de l’exécution des méthodes “nœuds_critiques”, “arcs_critiques” et “arcs_critiques_fusion” pour le graphe “G” et des alignements entre des réservoirs du “Multiple Principal de Chargement” (Multiple Principal de Carga) et les bras de chargement des quais. L’exécution de chaque méthode a pris respectivement 23, 10 et 90 secondes. Les 12 nœuds critiques et les 6 arcs critiques bloquants trouvés correspondent aux segments contenant les bras de chargement et les segments adjacents. Les réservoirs étant plus nombreux que les bras de charge-



```

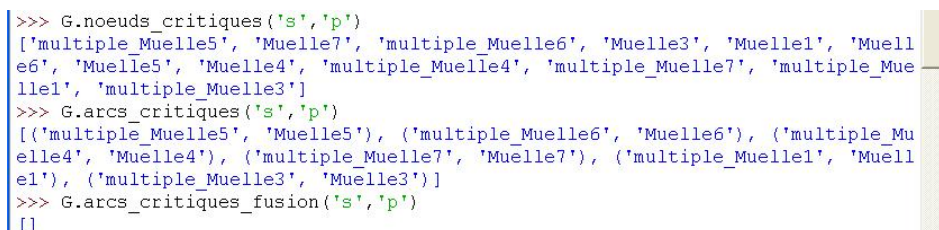
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.0 (r30:67507, Dec 3 2008, 20:14:27) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 3.0
>>> ===== RESTART =====
>>>
>>> G.flot_max('s', 'p')
6
>>> G.determiner_chemins('s', 'p')
{1: ['135x6', 'MPC_135x6', 'LC2', 'LCrudo2', 'multiple_Muelle7', 'Muelle7'], 2:
['135x1', 'MPC_135x1', 'LC1', 'LCrudo1', 'multiple_Muelle3', 'Muelle3'], 3: ['26
0x2', 'MPC_260x2', 'LT7', 'MPC_165x1', 'LT3', 'MPC_135x8', 'LT4', 'Travieso_26_M
ultiple_4', 'LT5', 'MPC_165x2', 'LC3', 'LCrudo3', 'mezclador_salida', 'LCrudo6',
'multiple_Muelle5', 'Muelle5'], 4: ['260x3', 'MPC_260x3', 'LC7', 'LCrudo7', 'mu
ltiple_Muelle6', 'Muelle6'], 5: ['260x4', 'MPC_260x4', 'LC6', 'mezclador_entrada
1', 'LC4', 'LCrudo4', 'multiple_Muelle1', 'Muelle1'], 6: ['260x1', 'MPC_260x1',
'LT6', 'tubo_sec6_sec12_sec18', 'LC5', 'LCrudo5', 'multiple_Muelle4', 'Muelle4']
}

```

FIGURE 5.6 – Résultats de l’exécution des méthodes “flot_max” et “determi-
ner_chemins” pour le graphe “G”.



```

>>> G.noeuds_critiques('s', 'p')
['multiple_Muelle5', 'Muelle7', 'multiple_Muelle6', 'Muelle3', 'Muelle1', 'Muell
e6', 'Muelle5', 'Muelle4', 'multiple_Muelle4', 'multiple_Muelle7', 'multiple_Mue
lle1', 'multiple_Muelle3']
>>> G.arcs_critiques('s', 'p')
[(('multiple_Muelle5', 'Muelle5'), ('multiple_Muelle6', 'Muelle6')), ('multiple_Mu
elle4', 'Muelle4'), ('multiple_Muelle7', 'Muelle7'), ('multiple_Muelle1', 'Muell
e1'), ('multiple_Muelle3', 'Muelle3')]
>>> G.arcs_critiques_fusion('s', 'p')
[]

```

FIGURE 5.7 – Résultats de l’exécution des méthodes “nœuds_critiques”,
“arcs_critiques” et “arcs_critiques_fusion” pour le graphe “G”.

ment, ils peuvent être remplacés par d'autres réservoirs pour une même capacité opérative. Par contre un bras de chargement défaillant diminue le nombre d'alignements possibles, d'où la criticité de ces segments. Aucun arc critique permissif n'a été trouvé.

```
>>> G.chemin_min_int('s','p','Anaco_Multiple_4','Muelle6')
['Anaco_Multiple_4', 'LT2', 'MPC_260x4', 'LC1', 'LCrudol', 'multiple_Muelle6', 'Muelle6']
```

FIGURE 5.8 – Résultats de l'exécution des méthodes “chemin_min_int” pour le graphe “G”.

La figure 5.8 montre les résultats de l'exécution des méthodes “chemin_min_int” pour le graphe “G” et des alignements entre des réservoirs du “Multiple Principal de Chargement” (Multiple Prinical de Carga) et les bras de chargement des quais. Le temps d'exécution est de 24 secondes.

```
>>> G.chemin_pompe('135x3','135x5','amont','aval')
[(('135x3', '135x3*', 1), ('135x3*', 'MPC_135x3', 1), ('MPC_135x3', 'MPC_135x3*', 1), ('MPC_135x3*', 'LT8', 1), ('LT8', 'LT8*', 1), ('LT8*', 'Merey_Multiple_4', 1), ('Merey_Multiple_4', 'Merey_Multiple_4*', 1), ('Merey_Multiple_4*', 'tubo_multiple3', 1), ('tubo_multiple3', 'tubo_multiple3*', 1), ('tubo_multiple3*', 'Succion1_97x1/7', 1), ('Succion1_97x1/7', 'Succion1_97x1/7*', 1), ('Succion1_97x1/7*', 'Succion1_97x1/7_Bombas_1,6,7', 1), ('Succion1_97x1/7_Bombas_1,6,7', 'amont', 1), ('amont', 'aval', 1), [('aval', 'tubo_multiple5', 1), ('tubo_multiple5', 'tubo_multiple5*', 1), ('tubo_multiple5*', 'LT2', 1), ('LT2', 'LT2*', 1), ('LT2*', 'MPC_135x5', 1), ('MPC_135x5', 'MPC_135x5*', 1), ('MPC_135x5*', '135x5', 1)]]
>>> |
```

FIGURE 5.9 – Résultats de l'exécution des méthodes “chemin_pompe” pour le graphe “G”.

La figure 5.9 montre les résultats de l'exécution des méthodes “chemin_pompe” pour le graphe “G”. Le temps d'exécution est de 30 secondes.

5.4.2 Centre de stockage de Punta de Palmas

L'objet P de la classe “Graphe” correspond au réseau de canalisation de Punta de Palmas. La figure 5.11 présente les résultats de l'exécution des méthodes pour le graphe P utilisant l'interpréteur de Python. La plupart correspondent à des alignements entre les oléoducs d'entrée et les réservoirs (figure 5.10).

Chacune des méthodes “flot_max” et “determiner_chemins” a pris un temps d'exécution de moins d'une seconde. Le nombre d'alignements (3) est limité par la quantité d'oléoducs d'entrée. Pour la méthode “nœuds_critiques” le temps d'exécution est de 2 secondes. Les 3 nœuds critiques trouvés correspondent aux oléoducs d'entrée. Pour les méthodes “arcs_critiques” et “arcs_critiques_fusion” les temps d'exécution sont respectivement de moins de une seconde, et de 9 secondes. Aucun arc critique (bloquant ou permissif) n'a été trouvé. Par conséquent, une défaillance sur une seule vanne quelconque du réseau ne diminue pas sa capacité opérative pour

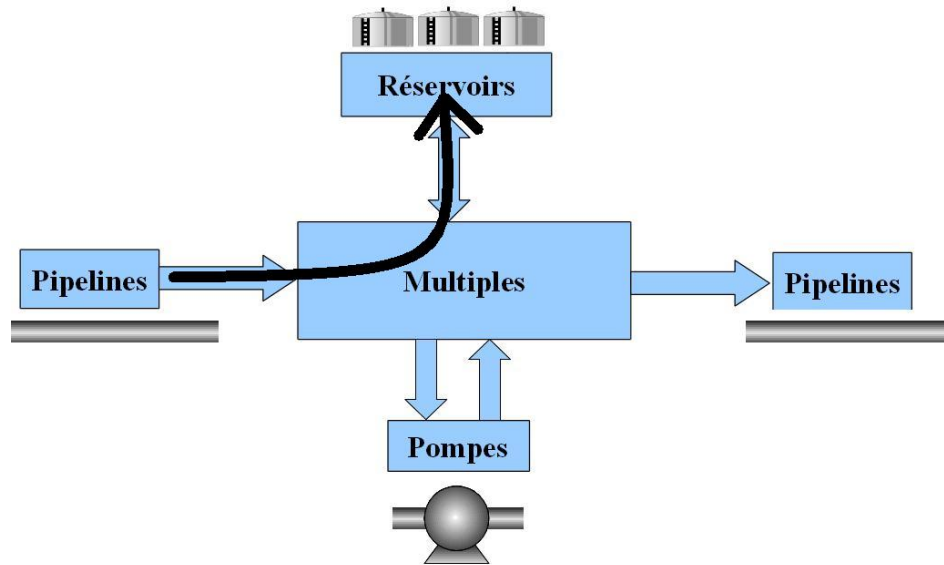


FIGURE 5.10 – Le calcul de la capacité opérative correspond à des alignements entre les oléoducs d'entrée et des réservoirs.

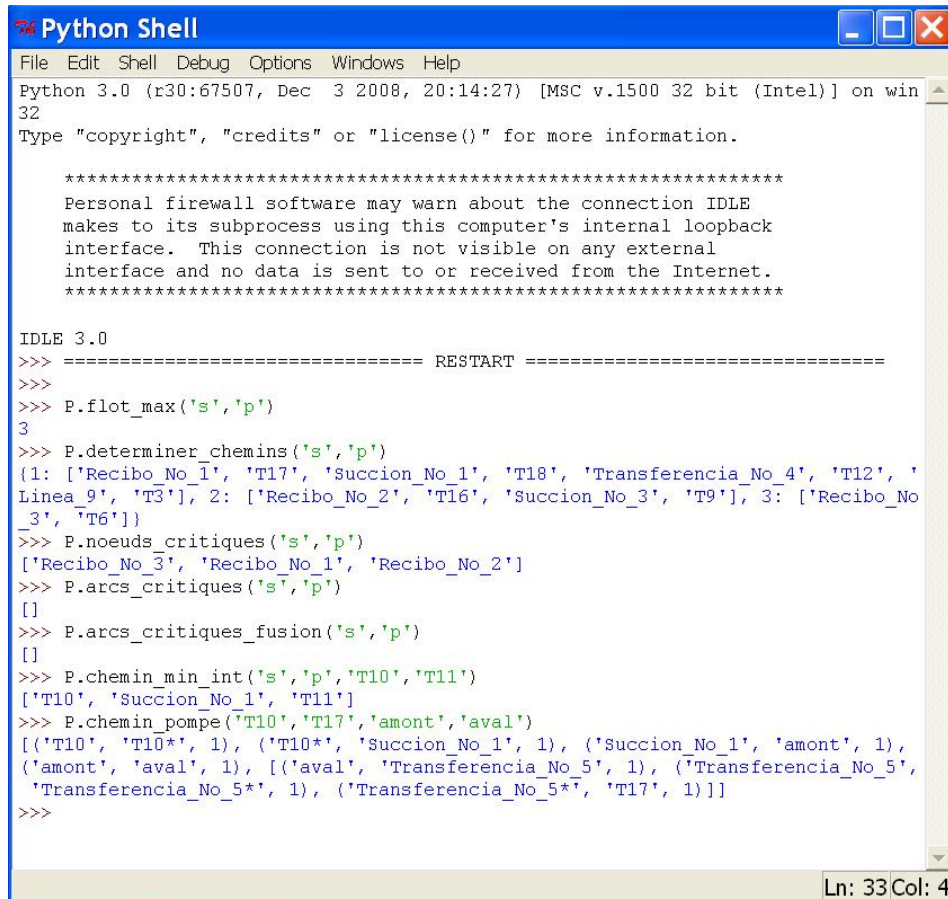
ce type d'alignement. Les méthodes "chemin_min_int" et "chemin_pompe" ont pris un temps d'exécution de 1 seconde.

5.5 Préconisations pour le développement d'un logiciel d'assistance aux opérateurs

L'objectif à moyen terme de nos travaux, est le développement d'un logiciel permettant d'assister les opérateurs de supervision des systèmes de transport de pétrole dans la prise de décisions. Dans un premier temps, nous avons envisagé l'implantation du logiciel sous forme d'un module du logiciel SCADA "galba" développé par PDVSA. Néanmoins, à cause de la variété de logiciels SCADA utilisés par l'entreprise, et des recommandations de développeurs du logiciel galba, le développement d'un logiciel indépendant, qui puisse communiquer, si nécessaire, avec le logiciel SCADA, nous a semblé plus convenable.

Les mises en œuvre des algorithmes en Python constituent le squelette de ce logiciel. Il devra être complété avec des modules complémentaires. L'architecture proposée serait constituée des modules suivants :

- Interface Homme Machine
- Unité de calcul
- Base de données
- Historique
- Communication SCADA



```

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.0 (r30:67507, Dec 3 2008, 20:14:27) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 3.0
>>> ===== RESTART =====
>>>
>>> P.flot_max('s','p')
3
>>> P.determiner_chemins('s','p')
{1: ['Recibo_No_1', 'T17', 'Succion_No_1', 'T18', 'Transferencia_No_4', 'T12', '
Linea_9', 'T3'], 2: ['Recibo_No_2', 'T16', 'Succion_No_3', 'T9'], 3: ['Recibo_No
_3', 'T6']}
>>> P.noeuds_critiques('s','p')
['Recibo_No_3', 'Recibo_No_1', 'Recibo_No_2']
>>> P.arcs_critiques('s','p')
[]
>>> P.arcs_critiques_fusion('s','p')
[]
>>> P.chemin_min_int('s','p','T10','T11')
['T10', 'Succion_No_1', 'T11']
>>> P.chemin_pompe('T10','T17','amont','aval')
[('T10', 'T10*', 1), ('T10*', 'Succion_No_1', 1), ('Succion_No_1', 'amont', 1),
('amont', 'aval', 1), [('aval', 'Transferencia_No_5', 1), ('Transferencia_No_5',
'Transferencia_No_5*', 1), ('Transferencia_No_5*', 'T17', 1)]]
>>>
Ln: 33 Col: 4

```

FIGURE 5.11 – Résultats de l’exécution des méthodes “flot_max”, “determiner_chemins”, “nœuds_critiques”, “arcs_critiques”, “arcs_critiques_fusion”, “arcs_critiques_fusion” et “chemin_pompe” pour le graphe “P”.

Interface Homme Machine : Une interface plus conviviale pour les opérateurs, permettant de saisir et de recevoir de l'information de manière simple et fiable est un module importante à rajouter. Ce module devra permettre une saisie des données du réseau sans connaissance particulière d'un langage de programmation. Il pourrait définir les types d'alignements à partir des spécifications des demandes de pétrole et de l'information sur l'état courant du système. En plus des chemins, et des listes de vannes à ouvrir/fermer, le logiciel devrait présenter les alignements dans les affichages.

Unité de Calcul : Ce module exploite l'information provenant de l'utilisateur, la base de données sur le réseau de canalisation, et l'information provenant du SCADA pour réaliser les calculs demandés. Sa fonction est l'exécution des algorithmes basés sur la notion de capacité opérative.

Base de données : Contient l'information permanente du réseau de canalisation, comme la topologie (graphe non orienté) les étiquettes de chaque élément, et des informations complémentaires permettant d'établir types et sous-types d'éléments, comme par exemple, la capacité des réservoirs, son élévation, le type de pétrole délivré par les oléoducs, etc.

Historique : Un module de sauvegarde des résultats évite un retard inutile causé par des temps de calcul non négligeables. Un historique des résultats calculés pour des situations spécifiques permet d'utiliser ces résultats si une de ces situations a lieu à nouveau.

Communication avec le logiciel SCADA : Les indisponibilités causées par des alignements, des opérations de maintenance ou des défaillances sont utiles pour calculer la capacité opérative résultante de ces indisponibilités. D'autre part, les alignements calculés par les algorithmes proposés pourraient conduire à des commandes permettant l'établissement automatique des alignements avec l'autorisation de l'opérateur. Ces tâches peuvent être accomplies à travers l'échange d'information entre le logiciel d'assistance et le logiciel SCADA. Comme nous avons expliqué précédemment, les systèmes de transport étudiés utilisent des logiciels différents. Actuellement, la plupart des logiciels utilisés ne tiennent pas compte des informations pertinentes pour notre problématique. Ces informations incluent :

- Vannes disponibles/indisponibles : les logiciels SCADA utilisés tiennent compte de l'état des dispositifs, comme l'ouverture des vannes ou le niveau des réservoirs mais non de leur disponibilité. Pour les logiciels d'assistance il est important de connaître cette disponibilité, en particulier celle des vannes établissant des alignements. Les vannes déclarées indisponibles modifient les modèles de graphes utilisés lors des calculs, ce qui peut affecter les résultats. Cette information pourrait, soit provenir du logiciel SCADA, soit être gérée par le logiciel d'assistance.
- Regroupement des commandes : dans les logiciels SCADA utilisés chaque élément doit être commandé séparément. Le regroupement des commandes permettraient d'automatiser l'établissement des alignements. Pour une liste de vannes à ouvrir/fermer, les logiciels d'assistance pourraient, à travers le logiciel SCADA, commander l'ouverture/fermeture de ces vannes de façon auto-

matique.

5.6 Conclusion

Bien qu'un algorithme soit validé par sa logique de fonctionnement, il est très important de tester sa mise en œuvre dans des cas de complexité comparable aux applications réelles. Cela permet de connaître les temps d'exécution et de déterminer la faisabilité de son application. Dans le cas des algorithmes proposés, les exemples présentés ainsi que les tests réalisés avec les données réels des réseaux de canalisation ont permis de vérifier leur faisabilité. Les temps d'exécution de l'ordre de quelques minutes, dans les pires des cas, sont suffisamment réduits pour des décisions devant être prises avec des heures d'anticipation. Dans plusieurs cas, une étude plus approfondie des temps d'exécution peut probablement conduire à l'amélioration des algorithmes dans ce sens.

Les mises en œuvre n'ont pas encore été utilisées par les opérateurs du système. Même si nous n'avons pas de réels retours de leur part, nos contributions sont basées en grand partie sur leurs indications et semblent prometteuses.

Le code développé est un point de départ très important pour le développement du logiciel d'assistance aux opérateurs de supervision. Nous avons proposé une architecture modulaire assez simple pour le développement futur de ce logiciel, tenant compte de sa probable interaction avec des logiciels SCADA.

Conclusion : Bilan, Extensions et Perspectives

Sommaire

6.1	Bilan	111
6.2	Extensions	112
6.3	Perspectives	115

6.1 Bilan

La croissance de la demande d'énergie et l'épuisement des gisements de pétrole dans le monde impulsent le développement de méthodes d'optimisation des activités liées à la production et à la consommation du pétrole et de ses dérivés. Le transport de pétrole est une de ces activités. Ces travaux abordent l'optimisation des opérations de transport dans des réseaux de canalisation, gérant plusieurs types de pétrole et interconnectant des pipelines avec des réservoirs, des tankers, ou des raffineries. D'un point de vue applicatif, cette étude a concerné le système de transport de pétrole du Terminal Maritime de Guaraguao, au Venezuela.

L'objectif final de notre contribution est l'assistance aux activités de supervision. Il a ainsi été nécessaire, dans un premier temps, de nous familiariser et de recueillir les informations portant sur ces activités et leurs usages. Les opérations de transfert de pétrole ont été caractérisées en termes d'alignements dont les plus notoires ont été retenus. Il s'en dégage que les besoins identifiés sont de maximiser le nombre d'opérations simultanées; de fait le concept de capacité opérative a été défini. La détermination d'indicateurs de capacité opérative permet aux opérateurs de prendre des décisions de manière à optimiser les opérations d'alignement. Ici, l'optimisation consiste à minimiser les interférences entre les possibles routes des alignements.

La problématique de recherche relève de la théorie des graphes. Une étude bibliographique préliminaire a confirmé le type de modèles et de concepts mathématiques exploitables. Le réseau de canalisation est ainsi modélisé sous forme de graphe non orienté, établissant une équivalence entre les routes des alignements et le concept de chaîne ou chemin dans un graphe. La capacité opérative a été identifiée comme une propriété permettant de mesurer le nombre maximum d'alignements de même type, ce qui correspond au nombre maximum de chemins disjoints entre deux ensembles

de nœuds. Un algorithme de calcul de la capacité opérative, basé sur un modèle de graphe de flot tenant compte des contraintes d'isolement des alignements, a été développé. Cet algorithme fournit également un possible ensemble maximal d'alignements. Pour chaque type d'alignement, une valeur de capacité opérative peut être calculée. L'ensemble de ces valeurs, appelés indicateurs de capacité opérative, permet de prévoir l'impact d'une opération ou les conséquences de l'occurrence de défaillances sur le réseau de canalisation.

A partir du concept de capacité opérative, plusieurs outils d'aide à la décision complémentaires sont proposés :

- La criticité d'un élément du système a été définie pour la diminution de la capacité opérative résultante de son indisponibilité. Des algorithmes permettant de déterminer les nœuds et les arcs critiques du réseau ont été développés.
- Un algorithme de recherche de chemin minimisant les interférences avec d'autres alignements a été proposé. Il repose sur la logique d'un algorithme de Dijkstra du plus court chemin, calculant de façon dynamique les coûts des arcs à partir des valeurs de capacité opérative.
- Un algorithme de recherche d'alignement forcé obligatoirement par une pompe dans un sens donné est proposé.

Les algorithmes proposés ont été implantés en Python et testés en utilisant les données de deux réseaux réels : celui du Terminal Maritime de Guaraguao, système complexe qui a été notre cas d'étude, et celui du Centre de Stockage de Punta de Palmas, un réseau plus simple ayant une problématique similaire. Les résultats sont satisfaisants et encouragent le développement industriel d'un logiciel d'assistance basé sur les algorithmes proposés. A ce titre, nous proposons une architecture modulaire de ce logiciel, en soulignant l'importance d'une communication avec le système de supervision et contrôle (SCADA) du réseau de canalisation.

6.2 Extensions

Au cours de la réalisation de cette thèse, plusieurs problèmes et développements possibles ont été exclus de nos travaux. Ils n'étaient pas considérés prioritaires ou nécessitaient une nouvelle étude approfondie du sujet. D'autre part, des analogies entre l'applicatif pétrole et d'autres domaines se sont présentées ; elles évoquent le caractère générique des problèmes abordés dans la théorie des graphes. Nous allons présenter ces possibles extensions avec quelques précautions d'usage.

Amélioration des temps de calcul des procédures Les algorithmes présentés dans cette thèse ont été développés en ciblant un temps de calcul adéquat pour les opérations d'alignement, lesquelles sont particulièrement lentes par rapport aux opérations appartenant à d'autres domaines (électronique, télécommunications). Même si nous avons conçu des procédures avec le but spécifique de diminuer les temps de calcul (comme par exemple la sous-routine du flot excédentaire (chapitre 4) dans la recherche de chemin maximisant la capacité opérative), nous n'avons pas recher-

ché à améliorer les temps de calcul pour des procédures moins critiques. Cependant il est fort probable que les temps de calcul puissent être réduits. Comme présenté dans le chapitre 2, de nombreux algorithmes améliorent les temps de calcul pour les versions de l'algorithme de Dijkstra et de Ford Fulkerson utilisées dans cette thèse. D'autre part, le langage de haut niveau Python, utilisé pour la mise en œuvre de ces algorithmes, s'il facilite énormément le codage des procédures, il peut être moins efficient que des programmes codés sous des langages de bas niveau comme C ou Fortran. La diminution du temps de calcul permettrait l'utilisation des algorithmes dans des réseaux de très grande complexité (réseau national intégrant plusieurs réseaux de canalisation et les oléoducs qui les interconnectent), et des extensions dans des domaines nécessitant une réponse plus rapide (réseaux électriques, télécommunications).

Alignements entre plus de deux éléments La problématique considérée dans cette thèse se limite au cas le plus fréquent, celui des alignements entre deux éléments. Cependant, il est tout à fait concevable de considérer plusieurs éléments devant être connectés par un même alignement. En restant toujours dans l'applicatif pétrole, par exemple : plusieurs réservoirs peuvent être remplis en même temps par un oléoduc, deux réservoirs contenant le même type de pétrole peuvent être utilisés en même temps pour augmenter le débit de pétrole vers un tanker. Ces alignements correspondent à des sous-graphes connexes sans cycles, appelés arbres, du graphe du réseau. Le calcul de la capacité opérative tenant compte de ces alignements semble un problème très complexe, dans lequel les algorithmes d'approximation sont nécessaires.

Alignements minimisant les risques Les conséquences des défaillances sur les alignements et les modèles de graphes ont été analysés dans le chapitre 4. Il est possible de tenir compte des risques liés aux défaillances dans le choix des alignements. Cette approche a été abordée par Karla Quintero dans ses travaux de master [Quintero, 2010]. A partir du modèle de graphe de flot présenté dans cette thèse, des coûts ont été assignés aux arcs du réseau, associant ce coût à la probabilité de défaillance d'un composant technologique. Un algorithme de flot maximum à coût minimum a été implanté, permettant de déterminer un ensemble maximal d'alignements minimisant les risques.

Alignements privilégiant des routes habituelles Très souvent, les opérateurs utilisent les mêmes routes lors de l'établissement des alignements. Les alignements déterminés à partir de l'algorithme de calcul de la capacité opérative sont donc utiles surtout dans le cas de défaillances et de situations rares où les opérateurs sont contraints d'utiliser des routes alternatives. Néanmoins, des routes semblables aux habituelles permettent aux opérateurs de mieux gérer le système et de rétablir les routes habituelles plus facilement. Il est possible de minimiser les déviations des routes par rapport aux routes habituelles en utilisant un algorithme de flot maximum

à coût minimum. Il suffit d'affecter des coûts nuls aux arcs faisant partie des routes habituelles, et des coûts unitaires au reste.

Généralisation du modèle de graphe non orienté : Réseaux à interrupteurs

Le modèle de graphe non orienté pour les réseaux de canalisation étudiés est basé sur l'analogie existante entre une vanne connectant deux parties d'un réseau, et une arête connectant deux nœuds dans un graphe. Dans un sens plus général, les vannes peuvent être vues comme des interrupteurs, contrôlant le passage de flux entre deux parties du réseau. Alors, le modèle peut être généralisé pour tout réseau d'interrupteurs de flux. Tout réseau contenant des interrupteurs simples, connectant deux portions du réseau, peut donc être modélisé de la même manière. Dans le cas des réseaux électriques, le circuit tally (figure 6.1) [Anceau, 2007], permettant le comptage du nombre des bits à "1" dans une séquence de bits, est un exemple de tels réseaux.

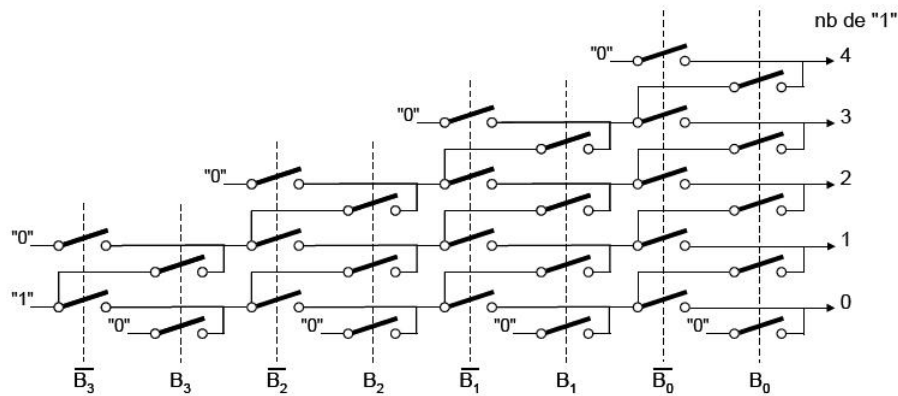


FIGURE 6.1 – Circuit tally permettant le comptage du nombre des bits à "1" dans une séquence de bits.

Algorithme de recherche de chemin avec un nœud intermédiaire obligatoire

Lors du développement de l'algorithme de recherche d'alignement passant par une pompe, nous avons recherché dans la bibliographie le cas similaire de recherche de chemin avec un nœud intermédiaire obligatoire, sans tenir compte du sens de pompage. [Wu & Ruan, 2006] propose une solution au problème du plus court chemin avec plusieurs nœuds intermédiaires, mais sans garantir l'optimalité de la solution. Ce problème peut être résolu pour des graphes non orientés, de manière simple, à partir du modèle de graphe de flot présenté dans cette thèse (chapitre 4).

6.3 Perspectives

Cette thèse prétend être le point de départ d'un projet à moyen terme de développement d'une plateforme d'assistance à la supervision des activités de transport de pétrole. Nous espérons que ce projet puisse avoir une continuation dans un futur proche. Dans ce sens, les intérêts manifestés par le DST de PDVSA, la division de Security Systems de Thalès, et le Laboratoire Ampère, permettent de prévoir un futur prometteur à ce projet. L'implantation du logiciel SCADA galba dans le système de transport du Terminal Maritime de Guaraguao ou dans d'autres systèmes de complexité comparable sera un pas important vers l'implantation et l'utilisation de ce type de techniques d'optimisation.

Mise en œuvre sous Python

Le code utilisé dans le chapitre 5 peut être téléchargé dans le site : <https://sourceforge.net/projects/opi/files/>

Le fichier “these.py” contient le code des algorithmes ainsi que les données des graphes “G” et “P” correspondant respectivement au réseau du Terminal Maritime de Guaraugao et au réseau du Centre de Stockage de Punta de Palmas.

Bibliographie

- [Afshar, 2008] Afshar, M. H. (2008). *Canadian Journal of Civil Engineering* 35, 129139. 23
- [Anceau, 2007] Anceau, F. (2007). *Conception des circuits VLSI*. Dunod. 114
- [Ausiello et al., 2003] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A. & Protasi, M. (2003). *Complexity and Approximation*. Springer. 26
- [Badiru & Cheung, 2002] Badiru, A. B. & Cheung, J. (2002). *Fuzzy Engineering Expert Systems with Neural Network Applications*. Wiley-Interscience. 4
- [Bavelas, 1948] Bavelas, A. (1948). *Applied Anthropology* 7, 16–30. 27
- [Beauchamp, 1966] Beauchamp, M. (1966). *Behavioral Science* 10, 161–163. 27
- [Behbahani-Nejad & A.Bagheri, 2010] Behbahani-Nejad, M. & A.Bagheri (2010). *Journal of Petroleum Science and Engineering* 70, 256–265. 24
- [Bellman, 1958] Bellman, R. (1958). *Quarterly of Applied Mathematics* 16, 87–90. 25
- [Bentley WaterCad Inc, 2010] Bentley WaterCad Inc (2010). Online : <http://docs.bentley.com/en/HMWaterCAD/>. Segmentation. 24, 53
- [Bondy & Murty, 1982] Bondy, J. & Murty, U. (1982). *Graph Theory with Applications*. 5ème édition, Elsevier Science Ltd. 23, 30
- [Cherkassky & A. V. Goldberg, 1996] Cherkassky, B. V. & A. V. Goldberg, T. R. (1996). *Mathematical Programming* 73, 129–174. 25
- [Chou & Chang, 2005] Chou, H. & Chang, C. (2005). *Industrial and Engineering Chemistry Research* 44, 114–123. 23
- [Cormen et al., 2001] Cormen, T., Stein, C., Rivest, R. & Leiserson, C. (2001). *Introduction to Algorithms*. 2ème édition, McGraw-Hill Higher Education. 38
- [Dejean et al., 2007] Dejean, J., Averbuch, D., Maurel, P., Gainville, M., Guet, S. & Johnsen, O. (2007). In *Proceedings of the European Safety and Reliability Conference*. 23
- [Dijkstra, 1959] Dijkstra, E. W. (1959). *Numerical Mathematics* 1, 395412. 25, 38
- [Dinic, 1970] Dinic, E. A. (1970). *Soviet Mathematics Doklady* 11, 1277–1280. 26
- [Edmonds & Karp, 1972] Edmonds, J. & Karp, R. M. (1972). *Journal of the ACM* 19, 248–264. 26
- [Eppstein, 1999] Eppstein, D. (1999). *SIAM Journal on Computing* 28, 652–673. 25
- [Floyd, 1962] Floyd, R. (1962). *Communications of the ACM* 5, 345. 25
- [Ford & Fulkerson, 1956] Ford, L. & Fulkerson, D. (1956). *Canadian Journal of Mathematics* 8, 399–404. 26, 43, 44

- [Ford & Fulkerson, 1962] Ford, L. & Fulkerson, D. (1962). Flow in networks. 5ème edition, Princeton University Press. 25
- [Freeman, 1977] Freeman, L. (1977). *Sociometry* 40, 3541. 27
- [Goldberg & Tarjan, 1988] Goldberg, A. & Tarjan, R. E. (1988). *Journal of the ACM* 35, 921–940. 26
- [Goldberg, 1987] Goldberg, A. V. (1987). Efficient Graph Algorithms for Sequential and Parallel Computers. PhD thesis, Department of Electrical Engineering and Computer Science, MIT. 26
- [Goldberg & Rao, 1998] Goldberg, A. V. & Rao, S. (1998). *Journal of the ACM* 45, 783–797. 26
- [Gondran & Minoux, 2009] Gondran, M. & Minoux, M. (2009). Graphes et algorithmes. 4ème edition, Lavoisier. 30
- [Guimera et al., 2005] Guimera, R., Mossa, S., Turttschi, A. & Amaral, L. (2005). In Proceedings of the National Academy of Sciences. USA. 27
- [Gupta & Prasad, 2000] Gupta, R. & Prasad, T. D. (2000). *Journal of Hydraulic Engineering* 126, 56–62. 23
- [Herrán-González et al., 2009] Herrán-González, A., Cruz, J. D. L., Andrés-Toro, B. D. & Risco-Martín, J. (2009). *Applied Mathematical Modelling* 33, 1584–1600. 24
- [Johnson, 1977] Johnson, D. (1977). *Journal of Computer and System Sciences* 9, 256–278. 25
- [Karmarkar, 1984] Karmarkar, N. (1984). *Combinatorica* 4, 373–395. 26
- [Karp, 1975] Karp, R. (1975). *Networks* 5, 45–68. 26
- [Kleinberg, 1996] Kleinberg, J. (1996). Approximation Algorithms for Disjoint Paths Problems. PhD thesis, Department of Electrical Engineering and Computer Science, MIT. 26
- [Koenig et al., 1999] Koenig, S., Youngberg, E. D. & Wright, D. L. (1999). *PSIG*. 22
- [Kolman & Scheideler, 2001] Kolman, P. & Scheideler, C. (2001). In Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures pp. 38–37,. 27
- [Lawler, 1976] Lawler, E. (1976). *Combinatorial Optimization : Network and Matroids*. Holt, Rinehart, and Winston. 25
- [Lebrun et al., 2009] Lebrun, M., Vasiliu, D. & Vasiliu, N. (2009). *SIC* 18, 111–118. 24
- [Leeuwen, 1998] Leeuwen, J. V. (1998). Handbook of theoretical Computer Science. Volume A : Algorithms and Complexity. The MIT Press. 59
- [Liu, 2003] Liu, J. (2003). 26
- [Mommer, 2010] Mommer, B. (2010). *La Cuestión Petrolera*. 3ème edition, Fondo Editorial Darío Ramírez. 27

- [Narayanan, 2005] Narayanan, S. (2005). Master's thesis Virginia Polytechnic Institute and State University. 27
- [Orlin, 1997] Orlin, J. B. (1997). *Mathematical Programming* 78, 109–129. 26
- [Python docs, 2008] Python docs (2008). Python v3.0 documentation. Python Software Foundation. 93
- [Quintero, 2010] Quintero, K. (2010). Master's thesis INSA de Lyon. 113
- [Rejowsky & Pinto, 2002] Rejowsky, R. J. & Pinto, J. M. (2002). *Brazilian Journal of Chemical Engineering* 19, 467–474. 22
- [Rojas et al., 2008] Rojas, J., González, J., Boutleux, E. & Niel, E. (2008). In *Proceedings of the XIII Latin-American Congress on Automatic Control*. 5
- [Rojas et al., 2009] Rojas, J., González, J., Boutleux, E. & Niel, E. (2009). In *Proceedings of the European Control Conference*. 6
- [Sabidussi, 1966] Sabidussi, G. (1966). *Psychometrika* 31, 581–603. 27
- [Salem & Nagy, 2004] Salem, A. A. S. & Nagy, H. M. (2004). *Brazilian Journal of Chemical Engineering* 43, 671–683. 23
- [Sokkalingam et al., 1996] Sokkalingam, P. T., Ahuja, R. K. & Orlin, J. B. (1996). *Networks* 36, 53–63. 26
- [Trench, 2001] Trench, C. J. (2001). Memorandum Allegro Energy Group. 8
- [Vazirani, 2003] Vazirani, V. V. (2003). *Approximation Algorithms*. Springer. 26
- [Walters & Smith, 1995] Walters, G. A. & Smith, D. K. (1995). *Engineering Optimization* 24, 261–268. 23
- [Wu & Ruan, 2006] Wu, W. & Ruan, Q. (2006). In *Proceedings of the 8th International Conference on Signal Processing* vol. 4,. 84, 114
- [Zecchin et al., 2009] Zecchin, A. C., Simpson, A. R., Lambert, M. F., White, L. B. & Vitkovsky, J. P. (2009). *Journal of Engineering Mechanics* 135, 538–547. 23

Glossaire

Alignement : ensemble de canalisations connectant au moins deux éléments du réseau et permettant le transport de pétrole entre ces éléments. Un alignement est établi lorsque les vannes connectant les canalisations faisant partie de l'alignement sont ouvertes et les vannes adjacentes à ces canalisations sont fermées.

Canalisation : tube destiné au transport à distance ou à la distribution d'un liquide.

DST : District Socialiste Technologique. Centre de développement de logiciels et de solutions technologiques, appartenant à PDVSA.

Multiple : portion du réseau vers laquelle convergent de nombreuses canalisations. Grâce à l'ouverture et à la fermeture de vannes, différentes connections peuvent être établies entre les canalisations. Les multiples permettent de regrouper de nombreuses vannes dans un endroit localisé, facilitant l'ouverture et la fermeture manuelle des vannes, ainsi que des opérations de maintenance, réparation et remplacement. Ces multiples offrent de la redondance.

Oléoduc : Conduite servant au transport à distance de certains hydrocarbures liquides.

PDVSA : Pétroles du Venezuela S.A. Société de l'État de la République Bolivarienne du Venezuela chargée de l'exploration, production, traitement, transport et commercialisation des hydrocarbures.

PMCDE : problème du nombre maximum de chaînes disjointes équivalentes.

Route : ensemble de canalisations d'un alignement connectant deux éléments du système. Une route est analogue à une chaîne ou à un chemin dans un graphe.

SCADA : acronyme de l'anglais Supervisory Control and Data Acquisition (contrôle par supervision et acquisition de données). C'est un système de supervision et de contrôle centralisé, permettant aux opérateurs de surveiller en temps réel des procédés industriels, au travers de capteurs, et d'agir sur ces procédés à travers d'actionneurs placés à distance.