This project is funded by the European Union

Grant Agreement Number:
657263

Action acronym:
GATEWAY

Action full title:
Developing a Pilot Case aimed at establishing a European infrastructure project for CO_2 transport

Type of action:

H2020-LCE-19-2014-2015

Starting date of the action: 2015-05-01
Duration: 24 months

D4.2
Pilot Case Scenarios

Due delivery date: 30 September 2016
Actual delivery date:

Organization name of lead participant for this deliverable:
University of Leeds

Project funded by the European Commission within Horizon2020		
Dissemination Level		
PU	Public	
CO	Confidential , only for members of the consortium (including the Commission Services)	X

Deliverable number:	D4.2
Deliverable title:	Pilot Case Scenarios
Work package:	WP 4 WP Pilot Case Scenarios
Lead participant:	ULEEDS

Author(s)		
Name	**Organisation**	**E-mail**
Tim Cockerill (Corresponding Author)	ULEEDS	t.cockerill@leeds.ac.uk
Maria de los Reyes Gonzalez Ferreras	ULEEDS	mn15mdlr@leeds.ac.uk
Dr. Andrew Pimm	ULEEDS	A.J.Pimm@leeds.ac.uk
Dr. Giuseppe Colantuono	ULEEDS	G.Colantuono@leeds.ac.uk

Abstract

The GATEWAY project has proposed a number of candidate pilot cases for initial development of a carbon dioxide pipeline network in Europe. A comprehensive impartial selection process was then carried out to choose the pilot case best suited for future development, as described in deliverable “D4.1: Pilot Case Definitions”.

Inevitably the level of detail in which it is sensible to describe the pilot cases is limited. Equally, simplifying assumptions must be made with respect to the nature of the pipeline network connecting sources of carbon dioxide, in order to keep the decision making process tractable. The work described in this report represents a series of studies that explore how different approaches to constructing pipeline networks in three case study areas might produce contrasting outcomes and benefits. In order that a wide range of possibilities may be rapidly considered, (semi-) automated, optimising approaches to modelling the design of potential CO_2 pipeline networks have been employed, and the key elements of these approaches are summarized. The case study areas selected are comparable to, but not identical to, the GATEWAY Pilot Case areas, and located in Germany, the Netherlands and UK.

The generated results allow inter-comparison of four contrasting network design strategies in the three case study areas with respect to both the network topologies and their economic performance. Results indicate that, in general, networks that rely on a greater degree of co-ordination and co-operation between the connecting sources (e.g. through pipeline sharing and oversizing) offer more attractive economics. The benefits of such co-operation over alternative network approaches were found to increase with the alignment of the sources collaborating in the network and to a lesser extent, the distances to be covered. These conclusions support the selection of the Netherlands Pilot case for further development within GATEWAY.

This project is funded by the European Union

TABLE OF CONTENTS

Page

1 INTRODUCTION

1.1 Background to the GATEWAY project

The stated objectives of GATEWAY, quoting Eickhoff et al. (2016), are as follows:

a) to define an initiative, referred to as the Pilot Case, providing a model for establishing a European carbon dioxide infrastructure project, targeting a gateway transferring carbon dioxide from source to sink. The gateway is intended to form the first leg of a cross-border network, allowing multiple sources and multiple sinks.

b) to make profound assessments of the substantial funding needs and available resources.

c) to solicit strong actions by the partners involved (member states of the EU and other countries).

The idea is to develop a comprehensive model case which, intentionally, will pave the ground for CCS deployment in Europe. It will result from the examination of, and agreement on, technical, commercial, judicial and societal issues pertaining to a future CO_2 transport infrastructure. The Pilot Case derived on this basis, will emphasise a gateway for CO_2 transport in the North Sea Basin.

Necessary and important elements are the possible arrangements for a super-national legal entity responsible for the planning, construction, commissioning, operations, future extension, and eventually the decommissioning of the infrastructure. Additional to innovation and technological refinements, a detailed understanding is required of the legal and statutory framework, ownership arrangements, commercial aspects including synchronised funding from multiple sources, and the sharing of risk and liability.

1.2 Aims of this report

The GATEWAY project has proposed a number of candidate Pilot Cases for initial development of a carbon dioxide pipeline network. A comprehensive impartial selection process was then carried out to choose the pilot case best suited for future development. The candidate cases, and the selection process are thoroughly described in the report by Eickhoff et al. (2016) for the relevant deliverable "D4.1: Pilot Case Definitions".

Inevitably the level of detail in which it is sensible to describe most of the Pilot Cases is limited. Equally, simplifying assumptions must be made with respect to the nature of the pipeline network connecting sources of carbon dioxide, in order to keep the decision making process tractable.

The work reported in this deliverable represents a series of studies that explore how different approaches to constructing pipeline networks in some parts of the Pilot Case areas might produce contrasting outcomes and benefits. In order that a wide range of possibilities may be rapidly considered, (semi-)automated, optimising approaches to modelling potential CO_2 pipeline networks have been employed, and the key elements of these approaches are summarized.

The report sets out the key features of alternative onshore pipeline development scenarios/strategis in areas of the UK, Germany and the Netherlands broadly coincident with those within the candidate Pilot Cases. The geographical correspondence is not exact as one of the purposes is to explore the impact of differing network extents. To try to avoid confusion, the areas considered in this report are referred to as “case studies” to distinguish them from the similar but more tightly specified candidate GATEWAY Pilot Cases. The analysis has also been constrained to onshore networks, as the range of possible optimal network topologies and routing constraints is much more complex than offshore.

The results for each of the three areas are subsequently compared, and this allows us to shed light on questions such as:

- What are benefits of taking a cooperative, government stimulated approach to carbon dioxide pipeline network development (as implied by the GATEWAY project), compared to less cooperative approaches relying on initiatives by individual stakeholders?

- How do carbon dioxide transportation costs vary with different network development approaches in different locations?

- How do costs vary with network extent and the volume of carbon dioxide transported?

In the original work programme, it was envisioned that this deliverable would also consider “Implementing actions pursuant to the PCI prerequisites (Project of Common Interest)”. Due to changes in the scheduling of the work and the consortium, it was no longer sensible to include this element within this deliverable. Instead this is dealt with in the report associated with deliverable 4.1.

2 OBJECTIVES AND STRUCTURE

2.1 Key objectives

The specific objectives of this report are to:

(1) Briefly review key studies exploring the benefits of co-operative approaches to CCS pipeline network development,

(2) Identify algorithms that could be deployed for automated carbon dioxide pipeline network design, and implement those algorithms within computational tools such that they can be practically applied,

(3) Gather data representing sets of possible carbon dioxide sources within the selected case areas, along with key environment information that might impact on pipeline routing,

(4) Apply the algorithms formulated in objective (2) to explore and evaluate alternative approaches to onshore carbon dioxide pipeline network development within parts of the case study areas,

(5) Analyse the following with respect to possible carbon dioxide pipeline networks within the considered case study areas:

- Possible network connection topologies for the identified carbon dioxide sources
- Expected volumes of carbon dioxide transport for the network topologies
- Financial resources and costs associated with the construction and operation of the alternative network topologies, extending to alternative development time histories.

(6) Draw conclusions with respect to the feasibility and benefits of alternative approaches to pipeline network development within the case study areas.

2.2 Report structure and relationship to objectives

The immediately following Chapter 3 begins by describing the geographical areas considered by the analysis, relating them to the GATEWAY Pilot Case areas. Selected key data used for each area is summarized, thereby fulfilling objective (3).

Chapter 4 describes the analytical procedures and tools that were developed for the work, including a brief review of the relevant literature. Objectives (1) and (2) are satisfied in this Chapter.

The results of applying the developed tools to each of the case study areas are set out in Chapter 5, with some immediate discussion. This completes objective (4), and begins to tackle objective (5). Chapter 6 compares the results for the case study areas, and in doing so completes objective (5) and objective (6). A final conclusion chapter summarises the key findings.

3 CASE STUDY AREAS

3.1 Selected areas and relationship to the GATEWAY pilot cases

Three case study areas have been analysed within the UK, Germany and the Netherlands. The areas have varying degrees of correspondence with the Pilot Case areas as the intention of this work was to both (i) understand how different approaches to pipeline network development might produce different outcomes, as well as (ii) provide some insight into the expansion potential of initial 'seed' networks. To achieve the first intention, it was important that a range of contrasting topologies, sources, flow rates and geographies was explored and some liberties have been taken with reality to ensure this. In all cases only the onshore pipeline network has been studied, up to an assumed connection point with an offshore network to storage reservoirs in the North Sea.

The UK case study area is essentially identical to the UK component of GATEWAY candidate pilot Case (A), but with a larger number of diverse sources and an increased annual flow rate. This represents a further stage of development from the Pilot Case, but assuming the network within the local area rather than expanding further across the country. Given the high concentration of carbon dioxide sources in the local area this is in fact the most likely next step.

The Germany (DE) case study area is intended to explore the impact of a wide ranging expansion across the whole of the country stimulated by the GATEWAY Pilot Case (B). Again there is a significantly increased flow rate over that envisioned by the Pilot Case. This case study area represents several stages of development after the Pilot Case. As with the UK case, the next stages of development after the Pilot Case are more likely to be among the concentration of sources within the local area. However the results of studying these earlier local stages with the modeling tools would be similar to that for the UK, which focused on the effects of local expansion. Hence it is more useful to take a more forward looking, if rather unrealistic, case study here to allow the effects of long distance, geographically diverse expansion to be explored. As the pipeline model is very time consuming to run with large numbers of sources, the selected sources are not the same as those in the pilot case, having been chosen to emphasise geographical diversity.

The Netherlands (NL) case study area is very broadly analogous to the onshore portions of GATEWAY Pilot Cases (C) and (D).The development potential of both these Pilot Cases has been relatively well explored in other projects, and there was no value in replicating this work in the current report. Instead, the primary intention of this case study was to explore the characteristics of networks with relatively high carbon dioxide flow rates, linking sources with that are relatively well aligned. Again a reduced number of sources is considered thanks to the run-time of the models.

3.2 UK Area

The UK case study area investigates a network between seven CO_2 sources located in the Teesside area of the East of England. The data for the sources and the offshore connection point is provided in Table 3.1, and Figure 3.1 shows their location on a map. This case aimed to investigate pipeline networks covering relatively short distances (less than 10km) and carrying a low flow rate (11.5 $MtCO_2$/year), representing an initial stage of expansion from GATEWAY Pilot Case (A). Carbon dioxide flow rates were taken from a database of sources developed for a previous pipeline networks project (Lone et al., 2010).

Table 3.1: Sources and offshore connection point for the UK case study area.

Reference	Longitude	Latitude	Annual CO_2 to network (tCO_2/yr)	Description
Source 1	-1.11108	54.58375	1,420,000	Petrochemicals
Source 2	-1.13794	54.58888	6,298,090	Steelworks
Source 3	-1.17914	54.60593	573,000	Petrochemicals
Source 4	-1.17794	54.61741	2,637,671	Cogeneration site
Source 5	-1.1186	54.58941	869,831	Power plant
Source 6	-1.17837	54.60871	508,000	Chemical plant
Source 7	-1.13005	54.6193	499,144	Power plant
Connection	-1.1084	54.627	Offshore connection point	

Figure 3.1: Overview map of the UK case study area showing sources and offshore connection point.

3.3 Germany (DE) Area

The Germany case explored the characteristics of networks covering long distances (>200km) with six sources widely spaced sources emitting producing a relatively high flow rate of 55.2MtCO_2/year. The CO_2/year emissions for the sources were given realistic but arbitrary values, to explore the effects of having widely different plant capacities on the network. These values do not represent the real emissions from the power plants therefore. Table 3.2 lists the sources and offshore connection point, which are shown as a map in Figure 3.2.

Reference	Longitude	Latitude	Annual CO_2 to network (tCO_2/yr)	Location
Source 1	9.364748	48.71706	12,700,000	Altbach Power Plant
Source 2	7.620569	51.63671	7,470,000	Bergkamen Power Plant
Source 3	13.24453	52.53457	60,000,000	Berlin-Reutter/West Plant
Source 4	7.23705	49.36348	7,730,000	Bexbach Power Plant
Source 5	8.951658	50.08827	20,000,000	Staudinger Power Plant
Source 6	10.09379	52.31504	7470,000	Hohenhameln Power Plant
Connection	7.775462	53.70908	Offshore connection point	

Table 3.2: Sources and offshore connection point for the Germany case study area.

3.4 Netherlands (NL) Area

The Netherlands case study area produced networks for the six sources listed in Table 3.3, which in contrast to the other two cases, are relatively well aligned as can be seen in Figure 3.3. This case aimed to investigate networks over medium distances (between 10km and 200km) for a medium total flow of 32.3 MtCO_2/year. The location of the sources was useful to test the sensitivity of the angle and cluster optimisers. The location of the sources was estimated from online searches and finding the plants on Google Maps.

The CO_2 emissions were calculated by taking the rated electrical power output of each plant and multiplying it by 0.75 (an estimated load factor), and representative values for the CO_2 emissions per unit of energy (450g of CO_2 per kWh for gas and 850g of CO_2 per kWh for coal).

Table 3.3: Sources and offshore connection point for the Netherlands case study area.

Reference	Longitude	Latitude	Annual CO_2 to network (tCO_2/yr)	Description
Source 1	53.435590	6.875031	8,935,200	Eemshaven
Source 2	51.702757	4.842304	6,952,703	Amer
Source 3	51.960827	4.093722	5,975,415	Maasvlakte MPP3
Source 4	51.958734	4.027208	5,819,049	Maasvlakte MV1 & 2
Source 5	52.567500	5.549722	2,601,720	Maxima
Source 6	51.149616	5.907777	5,617,350	Claus
Connection	52.323545	4.492497	Offshore connection point	

This project is funded by the European Union

Figure 3.2: Overview map of the Germany case study area showing sources and offshore connection point.

Figure 3.3: Overview map of the Netherlands case study area showing sources and offshore connection point.

4 METHOD

4.1 Brief review of selected pipeline network studies

4.1.1 Pipeline network approaches

The function of a pipeline network is to directly transport CO_2 emissions from sources (power plants, factories) to reservoirs. The source-to-reservoir path depends on the requirements of the project. Optimum path algorithms can be formulated for shortest distance; however, minimum cost is generally the dominant objective. The optimised path depends on source capture rates, storage locations, the number of reservoirs and their capacity, and geographical constraints (Middleton & Bielicki, 2009a).

Three path algorithms to link sources with reservoirs are common in the literature: the Ring main method, the Direct method and the Distributed method. The Ring main method connects each source to at least two other nearby sources, and at least two sources are connected to the reservoir. This method incurs the largest costs but has the greatest flexibility; if one pipe fails the rest of the system is not compromised. The Ring main method is more suited for gas distribution rather than CO_2 reception (Lone, 2009).

The Direct method connects each source to a reservoir without intermediate points. The Distributed method creates a tree like structure where sources are connected to the main trunk, which ends at the reservoir. Kuby et al. (2011) have demonstrated that the distributed network produces the lowest CCS costs, transportation costs and total pipeline length. While other configurations are possible, the analysis reported in this project opted to implement direct and distributed networks for a multiple source and single reservoir network.

Having selected the type of network, obstacle avoidance should be included since building pipelines is highly dependent on geography. Middleton et al. (2012) created the program SimCCS, which calculates the optimum path by dividing the terrain into 1km x 1km squares and assigning a weight to each one. Weighting is based on topography, crossings, ownership, land use, right of way, and population. By overlapping the maps, a construction cost surface is produced and the optimum path can be calculated.

4.1.2 Studies in the USA

The first large CO_2 pipeline (352km) was the Canyon Reef Carries (Texas) built in 1970 (Doctor et al., 2005). Since then, thousands of km of pipelines have been built for enhanced oil recovery (EOR) but none for permanent storage. Studies have been conducted for potential CCS pipelines for carbon dioxide storage but none have been constructed to date. This section summarises three studies for the USA, for which all costs are based on gas lines or EOR lines.

Middleton & Bielicki (2009b) used the program SimCCS to calculate the optimum path between the 37 largest CO_2 sources in California (22 power plants, 10 oil refineries and 5 cement manufacturers) and 14 reservoirs. The model was implemented for an increasing number of sources. The study found that for the final network (50Mt/year), source costs were US$35–55/tCO2, reservoir costs were US$1.50–5.50/t$CO_2$ and transportation costs were less than US$1/tCO2.

Another study by Middleton et al. (2012) used SimCCS, to investigate networks in the USA Midwest for 8 coal power plants (emitting approximately 102MtCO_2/yr) and 15 reservoirs. The study proved that pipeline transport benefited from full use of pipeline capacity and from economies of scale (bigger pipes were more economical provided they were fully utilised). The total CCS cost for 100MtCO_2/year was US$49.77/t$CO_2$, of which capture costs were US$38.44/t$CO_2$, US$8.9/tCO_2 for storage, and approximately US$2.43/t$CO_2$ for transportation. For 50MtCO_2/yr the total cost was about US$45.0/t$CO_2$ which is within the range obtained for the California study.

A study by Johnson & Ogden (2011) aimed to model connecting 86 sources to 62 reservoirs across 8 states in the USA southwest by 2050 to capture 362.8MtCO_2/yr. The total estimated cost for capturing 300MtCO_2/yr in 2050 was around US$45/t$CO_2$. The same cost was obtained for capturing 50MtCO_2/yr in 2016. In their paper, the price in 2050 was expected to be low thanks to economic incentives provided during the beginning of the program. They also assumed that by 2050 almost all power plants would have been retrofitted for CCS. The price range for 50MtCO_2/year agreed with the studies by Middleton & Bielicki (2009b) and Middleton et al. (2012).

4.1.3 European studies

Singh & Haines (2014) conducted a study of the current state of CCS networks and found that European projects were all at a conceptual stage, and that construction was unlikely to begin before 2020. This section reviews paper studies for three areas in Europe: the UK, the Iberian Peninsula (with Morocco), and northern Europe.

Two power plants, the Don Valley Power Project (920 MW) and the White Rose CCS Project (450 MW), and one underwater reservoir in the North Sea were analysed by Luo et al. (2014). In the study, an engineering analysis calculated pipeline diameter and an economic analysis found the lowest path cost. For the optimum case, the levelised capital cost for the pipelines was estimated to be 8.1 €/tCO_2 and the levelised energy and utilities cost was 7.62 €/tCO_2.

Berghout et al. (2015) designed a CCS network for Spain, Portugal and Morocco. Their model evaluated the possibility of creating country-shared networks and the benefits and difficulties entailed in such a venture. The study focused on a qualitative analysis rather than cost calculations. The barriers found for the project were administrative (time, complexity, lack of experience), political (borders, differences in policies and laws, lack of economic incentives) and geographical (mountains, land ownership).

Wildenborg et al. (2009) analysed the economic benefits of CCS in Europe. They created a non-optimised map to approximate the investment costs for each country involved. Germany and the UK were identified as the top emitters, while Ireland and Spain had the highest transport and storage costs (18€/tCO_2 and 13€/tCO_2 respectively). A cost-driven study was performed by Kjärstad et al. (2011) for Belgium, the Czech Republic, Germany, the Netherlands, Poland and the Slovak Republic. Together, the countries counted 640 power plants and produced ~500Mt of CO2 in 2007. The calculated transportation costs varied from 4.06€/tCO_2 for the Netherlands to 1.00€/tCO_2 for the Czech Republic.

4.2 Pipeline network design strategies investigated

4.2.1 Introduction

To facilitate investigation of alternative pipeline network design strategies, a series of MATLAB codes (the “pipeline network design model”) have been developed to produce networks for arbitrary case study areas when following each strategy. This section describes the network design strategies that were implemented within the codes. The subsequent Section 4.3 briefly discusses the main computational algorithms developed to perform the network design.

4.2.2 Direct routing

The direct route strategy, as its name suggests, connects each source directly to the reservoir via a straight pipeline. Each source is independent from the others and there is no coordination between them. Figure 4.1 shows how the sources are connected to the reservoir.

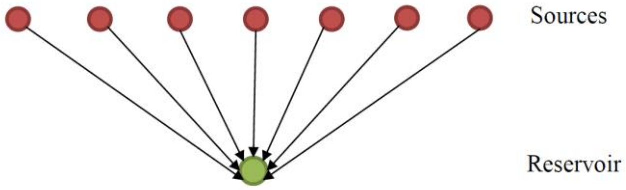


Figure 4.1: Direct routing strategy.

4.2.3 Angle based route optimiser

The angle optimiser works in two stages. In stage one, the closest pair of sources are joined at a meeting point, then the next two closest sources are joined and so forth until all sources are joined. For an odd number of sources, the last source is connected to the closest meeting point. In stage two, the two closest meeting points are joined, forming a new meeting point and the original meeting points are removed from the list of points to be joined. The step is repeated until all meeting points are joined. The last meeting point is connected to the reservoir. Figure 4.2 illustrates how the optimiser would connect an odd number of sources. The location of each meeting was determined using a polar coordinate optimisation method described by González Ferreras (2016).

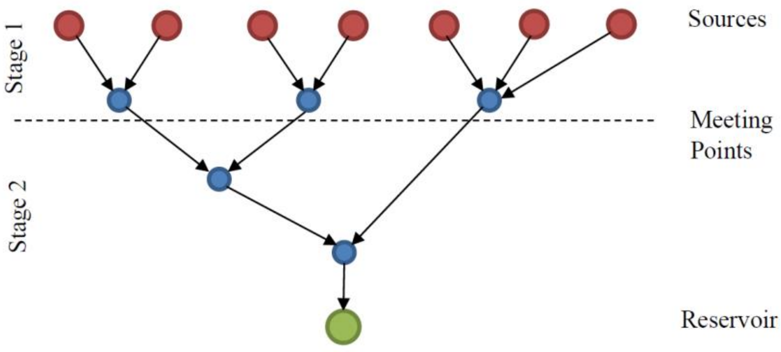


Figure 4.2: Angle base route optimiser stategy.

4.2.4 Source cluster based route optimizer

The cluster optimiser creates the first meeting points by clustering the sources. The MATLAB function *kmeans* was used to create clusters of points for the sources. First, the number of clusters is selected by the user. Next, *kmeans* is used to create the first meeting points (stage one) and each source is connected to its closest meeting point unless the source is closer to the reservoir, in which case, the source is directly connected to the reservoir. Finally, the angle optimiser method is used to connect the stage two meeting points until the reservoir is reached. Figure 4.3 shows the application of the cluster optimiser for seven sources and the selection of two clusters. The connections vary depending on the number of clusters chosen by the user.

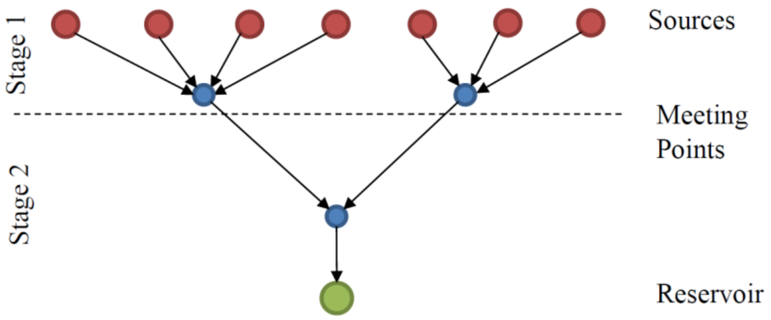


Figure 4.3: Cluster based route optimiser strategy.

4.2.5 Highly co-operative network development

The highly co-operative design strategy aims to model a situation in which pipelines would be built with a high degree of planning and co-ordination between the sources involved with respect to infrastructure sharing. As such it attempts to replicate the underlying philosophy of the GATEWAY project. Pipeline paths are based on those produced by the angle and cluster optimizers, with scope for manual editing. As there is manual input required for this design strategy it does not represent a true mathematical optimum, but has been included nonetheless to provide insight to the potential benefits of a co-ordinated, stimulated approach to network development.

4.3 Pipeline network design model

4.3.1 Introduction

This section describes the main algorithms underlying the pipeline network design model implemented in MATLAB. Only summary information is provided here, while full details can be found in a recently completed University of Leeds Thesis (González Ferreras, 2016), on which this section draws heavily.

4.3.2 Model overview

The flow chart in Figure 4.4 illustrates the essential features of the pipeline network design model. Initially the case study area specific data is read in. The first stage in the calculation process is to establish the optimum pipeline meeting points using the methodologies described in section 4.2. As the cost depends on both the pipeline lengths and the diameters, the path optimization function is run iteratively with a cost evaluation function to establish the least cost topology. Subsequently an obstacle avoidance function optionally operates to produce the final pipeline routes. Finally the outlet and booster conditions are calculated and the derived routes are plotted.

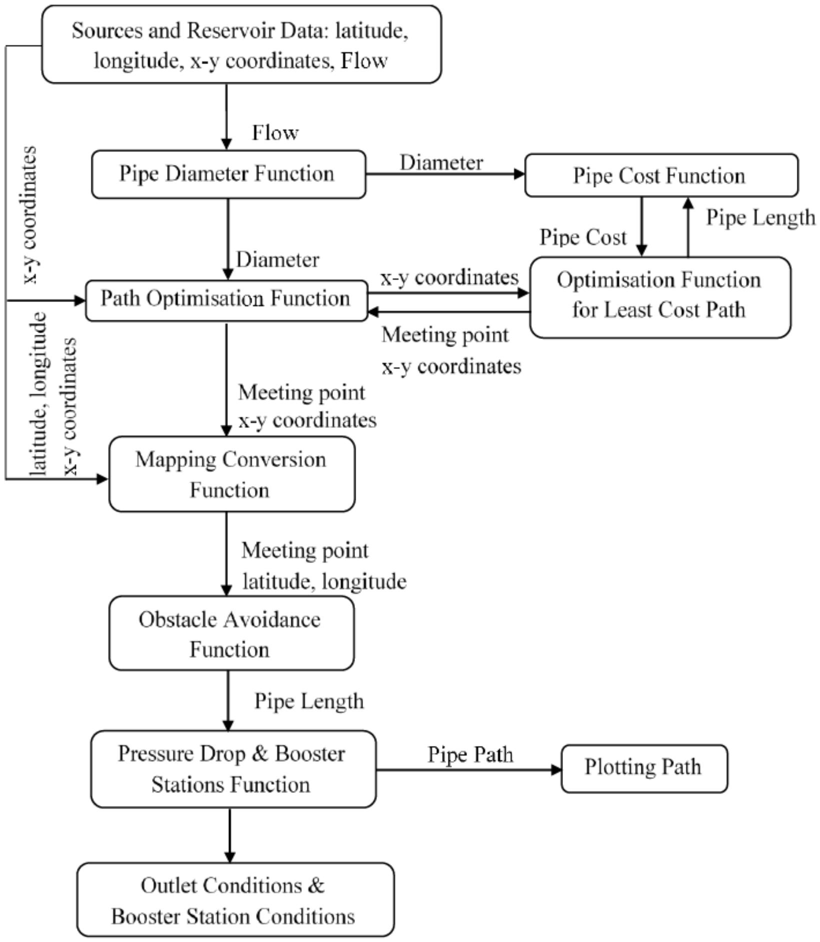


Figure 4.4: Model overview flow chart.

As the flow chart makes clear a number of co-ordinate transformation are required at several stages of the calculation, with the optimization elements of the model working in an x-y plane and the final design elements operating in a lattitute-longitude system. The details of these are relatively lengthy and tedious, so are not discussed further in this report. The principles underlying the main technical calculations carried out by the model are outlined in the following subsections.

4.3.3 Engineering pipeline design

Transporting CO_2 at high pressures over long distances results in significant pressure and hence booster stations can be required to raise the pressure. Modelling the pipe diameter, pressure drop, and the resulting number of booster stations is essential to calculate the total costs of the pipeline system. This section reviews the key fluid dynamic principles that are implemented in the model.

4.3.3.1 Pipeline relationships

In addition to length, pipeline diameter is a key factor in determining costs. Vandeginste and Piessens (2008) compared five pipeline diameter equations of varying complexity and accuracy, demonstrating that different results were obtained according to the number of factors accounted for. In optimisation studies, such as that presented here, there is a premium on simplicity as every calculation may potentially be repeated many thousands of times, and hence an equation quoted by the IEA (2005) was considered sufficiently accurate for our purposes here:

$$D = \sqrt{\left(\frac{4Q_m}{v\pi\rho}\right)} \tag{4.1}$$

Use of this equation clearly requires knowledge of the mass flow rate Q_m, the fluid density ρ, and the fluid velocity v. The mass flow rate is simply determined as the sum of the relevant capture carbon dioxide emissions from the sources in the relevant parts of the network, and assumed to be constant over time.

The density and viscosity of CO_2 are difficult to calculate under the conditions typical for pipeline transport, so use was made of the freely available CoolProp (2016) database to estimate these. The density is of course a function of the local fluid temperature and pressure. For the former it is assumed that flow is isothermal, adopting the annual average ground temperature in the region analysed. The Darcy-Weisbach equation

$$\Delta p = L f_D \frac{\rho}{2} \frac{v^2}{D} \tag{4.2}$$

is used to calculate pressure drop as the carbon dioxide flows through the pipeline. Turbulent flow is assumed in all cases, with the Swamee-Jain equation used to calculate the friction factor

$$f_D = \frac{1.325}{\left[\ln\left(\frac{e}{3.7D}\right) + \left(\frac{5.74}{Re^{0.9}}\right)\right]^2} \tag{4.3}$$

taking the roughness height e to be 0.045mm, which is a common value in the literature for carbon steel.

4.3.3.2 Implementing the pipeline equations

The pressure in the pipeline reduces as the carbon dioxide flow along it. At the same time, density and velocity also change as pressure changes along the length. To account for these changes in conditions along the length, each pipe run is divided into stations at 1 km intervals, with the pressure and other parameters evaluated at each station. Figure 4.5 illustrates the arrangement.

Using the equations in Section 4.3.3.1 the pressure, density and velocity at the end of each segment (i) are calculated and used as the initial conditions for the following segment (i+1). If the pressure in the pipe drops below a minimum pressure a booster station is inserted and the pressure raised to a predefined value, with the temperature, density and other properties also being reset. The procedure is repeated until the end of the pipe is reached and the outlet conditions were calculated.

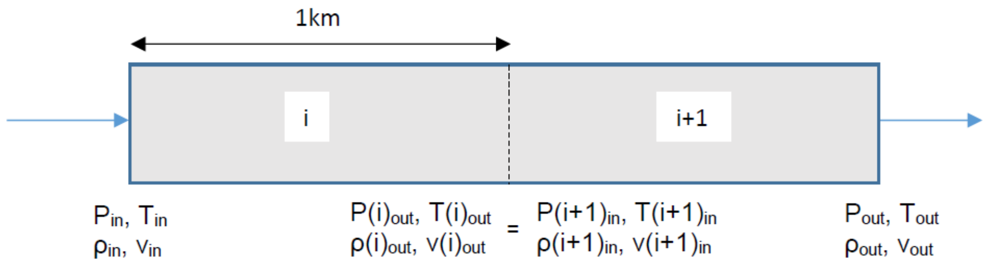


Figure 4.5: Discretisation of the pipeline.

4.3.4 Pipeline conditions and assumptions

Table 4.1 summaries the key values assumed for the pipeline design calculations. The following discussion explains the origins of these values.

Table 4.1: Key values assumed for the pipeline design calculations.

Parameter	Value
Initial Pressure	12 MPa
Minimum Pressure	8.5 MPa
Temperature	15 ºC
Velocity	2 m/s
Diameter Safety Factor	1.03
Pipe roughness	0.045 mm
Wall thickness	10 mm

The literature makes clear that supercritical conditions are required for the efficient transport of carbon dioxide by pipeline. However, because pipes would be buried underground, maintaining a temperature above 31ºC in Northern Europe would require additional insulation and/or temperature booster stations. In the UK for example, ground temperatures 1m below the surface range between 5ºC and 16ºC (NERC, 2011). To avoid additional costs, it was assumed therefore that carbon dioxide would remain within this temperature range, and hence could be modelled as a high pressure liquid (subcooled).

Kaufmann (2008) states that the minimum pressure for CO_2 transport in the dense phase (high pressure liquid) in pipes is 8MPa. An IEA (2010) report agrees with this value and states that temperature is allowed to drop to the dense phase as long as the pressure remains high (+8MPa). Vanderginste & Piessens (2008) used 8-11 MPa for their analysis while Chandel et al., (2011) used 10-13MPa, and Kaufmann (2008) used 8.5-13 MPa. For the work reported in this deliverable the minimum pressure was set to 8.5 MPa, to avoid a two-phase CO_2 state arising. The maximum pressure, in other words the pressure at which carbon dioxide will be delivered to the network from sources, and the repressurisation pressure in booster stations, was set to 12MPa in accordance with the literature.

The initial fluid velocity was also set based on recommended values in literature. Vanderginste and Piessens (2008) used values between 1-2 m/s, and an IEA (2010) report recommended a maximum velocity of 3 m/s. An additional sensitivity study carried out by one of the authors of this report (see González Ferreras (2016)) indicated that an initial velocity of 2 m/s would minimize overall costs with respect to balancing pipeline diameters with the number of booster stations required. The choice for the inlet velocity therefore was 2m/s.

Other factors that can impact the pipe conditions and the diameter calculations are impurities in the flow and corrosion effects. Kaufmann (2008) found CO_2 impurities (e.g. N_2, H_2, and NO_2) increase pressure in pipes, and impurities cause a decrease in density (Serpa et al., 2011). Corrosion due to water content in the CO_2 flow is a problem in pipes however, this report assumes all sources are equipped with a dehydration unit such that water concentration is below the maximum permitted levels. While impurities and corrosion have an effect on pipeline conditions and diameter, it was not practical to account for them with the very limited resources available for this workpackage, and they were treated as negligible in all calculations. A safety factor of 1.03 on diameter calculations was included.

Wall thicknesses for CO_2 pipes are typically within the 5.2-27 mm range, and for pressures 9.8-4.5MPa wall thickness range 10-13mm (IEA, 2013). A wide range of approaches are reported in the literature to calculate the minimum thickness as a function of the pressure and the yield strength of the pipe material. However the impact on overall costs of small changes in wall thickness is relatively minor, and again in view of the limited resources available for the project, for simplicity a constant value of 10mm wall thickness was assumed.

4.3.5 Cost relationships

The study used the cost equations of the IEA 2013 CO_2 Pipeline Infrastructure Report, which are based on an analysis of USA based existing oil and gas pipelines. Table 4.2 summarizes the values and relationships used, where D is the diameter in inches and L the length in miles. Values were converted into Pounds Sterling using the exchange rates in Appendix A1.The total capital cost of each pipe network is calculated as the sum of all the costs for each pipe in the network, plus the additional capital costs and booster station costs.

Table 4.2: Pipeline cost estimating relationships used in the model.

Cost	**Estimating formula (US$)**
Materials (Carbon steel)	$70,350+2.01*L*(330.5*D^2+686.7*D+26,960)$
Labour	$371,850+2.01*L*(343.2*D^2+2,074*D+170,013)$
Miscellaneous	$147,250+1.55*L*(8,471*D+7,234)$
Right of Way	$51,200+1.28*L*(577*D+29,788)$
Additional costs	
Surge Tank	1,244,724
Pipeline Control System	111,907

Booster stations costs were assumed to be a constant value (per station), taken from a CCS pipeline analysis by the IEA (2005). Once converted to Pounds Sterling, the costs per station equated to approximately £5.5M per station. The number of booster stations required was determined by performing the pressure drop analysis described in section 4.3.3.

4.3.6 Obstacle avoidance

Realistic pipeline routes will need to avoid crossing certain areas of land for safety reasons (e.g. built up areas areas), environmental reasons (e.g. conservation areas) or for ease of construction (e.g. lakes, large rivers). The A* (or A star) algorithm (e.g. Delling et al. (2009)) has been used to implement obstacle avoidance within the model. The algorithm considers all possible paths from a starting point (source or meeting point) to a target (meeting point or offshore connection point) and calculates the least cost path. The MATLAB code for A* was obtained from MathWorks file exchange and modified to suit this project.

In principle the model code can be readily configured to ensure pipeline routes avoid any form of obstacle that is geo-spatially described in vector form within an ESRI Shapefile. For this study only built up areas and internal waters were included, again constrained by the very limited resources available to this work package. Appropriate spatial descriptors were obtained from the VMAP level 0 data provided by the US National Imagery and Mapping Agency (NIMA, 1998). Some limitations of the A* algorithm became apparent on testing. In particular, the algorithm had difficulty in finding routes across very long obstacles. As a result the spatial data was edited to remove some very long European rivers (notably the Volga, the Rheine, the Seine, and the Loire)

An illustration of the obstacle avoidance technique for two arbitrary locations is shown in Figure 4.6. The red lines show the populated areas, the blue lines the internal waters, and the green line is the path calculated between the two specified points.

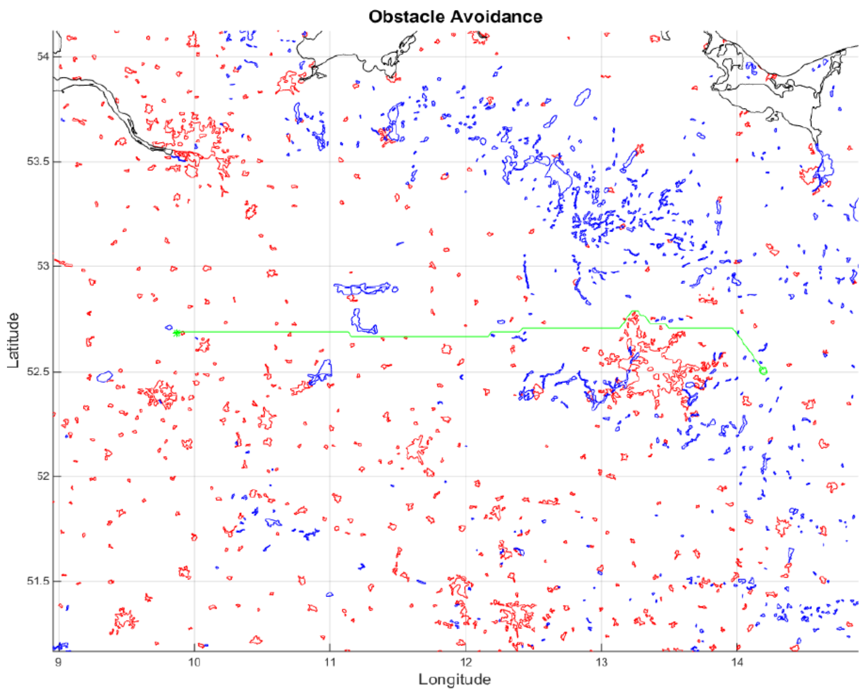


Figure 4.6: Illustration of the obstacle avoidance technique.

The A* algorithm operates by creating a map of cells to identify the location of the obstacles and determine the optimum path. The size of the cells can be reduced to increase accuracy, albeit at the expense of computational time. A grid independence study was performed to check the accuracy as a function of the calculation time required, based on the direct connection strategy for the Germany case study area. Table 4.3 shows the performance of the algorithm.

Using cell divisions of 2,000, 3,000 and 4,000 produced similar paths for all the sources (plots are shown in Appendix A2). A cell division of 1,000 had the lowest computational time but was unable to avoid small obstacles; whereas 5,000 cells avoided all obstacles but took seven hours to compute. A cell division of 2,000 was selected to reduce computational time when obtaining results. Additionally, there was only a maximum difference of 1.3% in total pipe distance between all the cell sizes, and a 1% difference between the 2,000 and 3,000 cell sizes.

Table 4.3: Obstacle avoidance grid independence test results.

No. of cells	Run Time (minutes)	Total Pipeline length (km)
1,000	9.4	2,845
2,000	17.5	2,838
3,000	67.4	2,809
4,000	209.3	2,806
5,000	467.1	2,809

4.4 Implementation

The pipeline network design model has been implemented as a number of MATLAB codes. For efficiency in development and operation extensive use has been made of supporting MATLAB toolboxes, notably the mapping tool box which deal with much of the reading/writing, processing and plotting of the spatial data. Appendix A4 provides a sample listing of the main code for the cluster network optimiser.

The obstacle avoidance function described in section 4.3.6 is implemented as an optional feature. In practice, the obstacle avoidance function was found inconvenient to use, thanks to (i) very long run times and (ii) that there were certain configurations of obstacles (e.g. very long rivers) that could cause it to fail. As a result it is frequently preferable to omit it from the analysis, particularly when analyzing a new case study for the first time. In many cases including obstacle avoidance makes only a relatively small difference to the overall results i.e. the best performing network design strategy with obstacle avoidance is likely to be the same as that without it. The results in the next section of this report all describe output produced without use of the obstacle avoidance. Data for equivalent networks accounting for obstacle avoidance is available in González Ferreras (2016).

Cost calculations are all carried out in GB Pounds Sterling, with data having been pre-converted into that currency using the exchange rates detailed in Appendix A1.

5 ANALYSIS

5.1 Introduction

This Chapter presents the outcomes of applying the different network design strategies to the case study areas. The overall costs of each network are presented in Chapter 6, where they can be more conveniently compared and analysed. For the reasons discussed previously, and in particular because it has only a marginal impact on the results, only routes without obstacle avoidance are presented here. Full details of the results produced when obstacle avoidance is taken into account are available in González Ferreras (2016), with some representative route maps also included in Appendix A3.

5.2 UK Area

Figure 5.1 shows the pipeline networks produced for the UK case study area. Note that thanks to the short distances involved in this case study, no booster stations were required. The direct route optimiser connected each source to the reservoir via a straight pipeline, as expected. The angle optimiser first connected sources 3 and 6 at M1, then 1 and 5 at M2, then 4 and 7 at M3, and finally source 2 to its nearest meeting point, M2. Next, it connected the meeting points M1 and M3 at M4, then it joined meeting points M2 and M4 at M4, and finally M4 was connected to the reservoir.

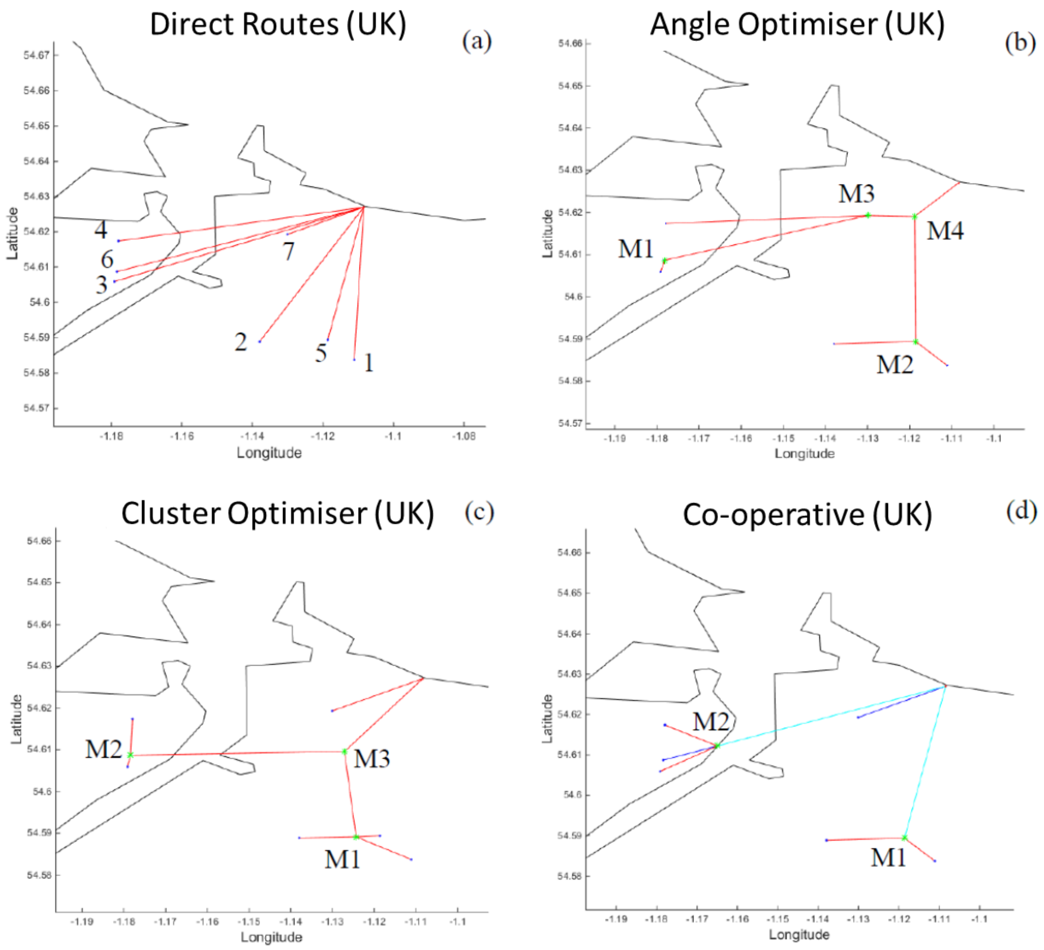


Figure 5.1: Generated networks for the UK case study area.

The cluster optimiser was set to create two clusters. The first cluster (M1) joined sources 1, 2 and 5, and the second cluster (M2) joined sources 3, 4 and 6. Source 7 was connected directly to the reservoir because the distance to the reservoir was shorter than the distance to either meeting point. Then, M1 and M2 were joined at M3, which was connected to the reservoir. For the trivial optimisers the meeting points selected were M1 located at source 5, which joined sources 1, 2 and 5, and M2 located at (54.61221, -1.16498), which joined sources 3, 4 and 6.

For the highly co-operative strategy, sources 5, 6 and 7 were directly connected to the reservoir (dark blue lines Figure 5.1 (d)). Sources 1 and 2 were connected to M1, and sources 3 and 4 to M2 (red lines). Finally M1 and M2 were connected to the reservoir (light blue lines).

By way of illustration, details of the network produced by the angle optimiser is showcased in Table 5.1. Similar tables were produced for all cases and network strategies, but are not included in this report in order to avoid making it overly long. For full details reference can be made to González Ferreras (2016), which also provides details of the required pressure booster stations for each network.

Table 5.1: Breakdown of the angle optimiser network for the UK case study area.

Network segment	**Length (km)**	**Flow (tCO_2/hour)**	**Diameter (m)**
Pipe 1	0.3160	58.870	0.110
Pipe 2	0.0172	52.192	0.104
Pipe 3	0.7953	145.890	0.174
Pipe 4	0.0031	89.366	0.136
Pipe 5	3.0969	270.994	0.237
Pipe 6	0.0047	51.282	0.103
Pipe 7	1.2502	647.064	0.366
Pipe 8	3.3172	758.126	0.396
Pipe 9	0.0000	322.276	0.258
Pipe 10	3.2991	235.257	0.220
Pipe 11	0.7216	1080.401	0.473
Pipe 12	1.1053	1315.658	0.521
Total	13.9017		

5.3 German Area

Calculated networks for the German case study are shown in Figure 5.2. The direct optimiser connected each source directly to the reservoir, with required booster stations denoted by green dots as shown in sub-figure (a). The angle optimiser joined sources 4 and 5 at M1, sources 2 and 6 at M2, and sources 1 and 3 at M3. Next, meeting points M2 and M3 were connected at M4, then M1 and M4 were connected at M4, and finally M4 reached the reservoir.

The cluster optimiser created two clusters. Cluster M1 joined sources 1, 4 and 5, and cluster M2 joined sources 2, 3 and 6. The two meeting points M1 and M2 were connected at M3 which went to the reservoir. The highly co-operative strategy (Figure 5.2(d)) joined pipelines from sources 1 and 4 at source 5, from sources 6 and 3 at point M2.

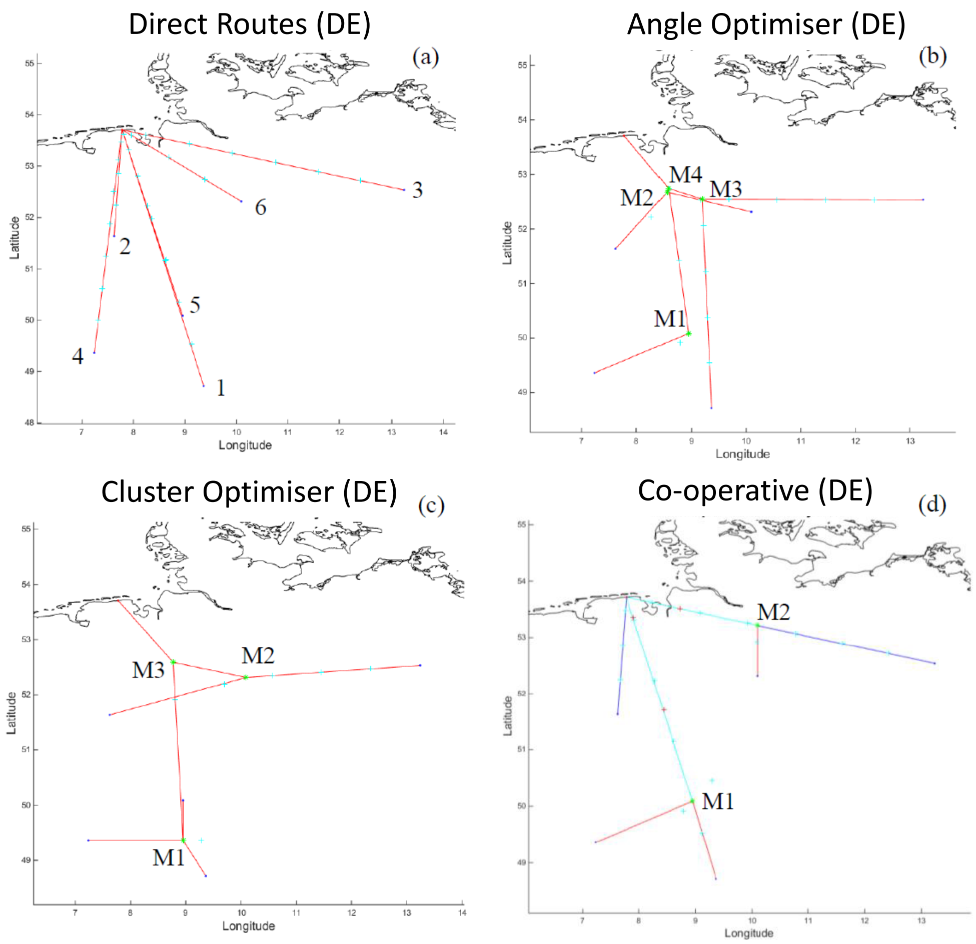


Figure 5.2: Generated networks for the German case study area.

5.4 Netherlands Area

The results for the Netherlands case study are shown in Figure 5.3. The direct connections are as expected, while the angle optimiser joined sources 3 and 4 at M1, sources 2 and 6 at M2 and sources 1 and 5 at M3. It then combined meeting points M1 and M2 at M4 (whose coordinates at very close to M1); next it joined M4 and M3 at M5, and finally M5 was connected to the reservoir. Green dots again denote pressure booster stations.

The cluster optimiser created two clusters, M1 joined sources 1, 5 and 6, and M2 joined 2, 3 and 4. Then, meeting points M1 and M2 were joined at M3, which connected to the reservoir. For the co-operative strategy, source 6 was connected to 2 at meeting point M1, source 4 to 3 at M2, and source 1 to 5 at M3. Sources 2, 3 and 5 were directly connected to the reservoir.

In some cases for this particular study, lengths of pipeline appear to run across water. While this is of course unrealistic, it does not have a significant impact on the results as there is currently no cost penalty applied to underwater pipelines. A similar problem also exists when the obstacle avoidance routines are activated (see Appendix A3). Further work is needed to investigate this, but resource limitations are likely to prohibit this within the current project.

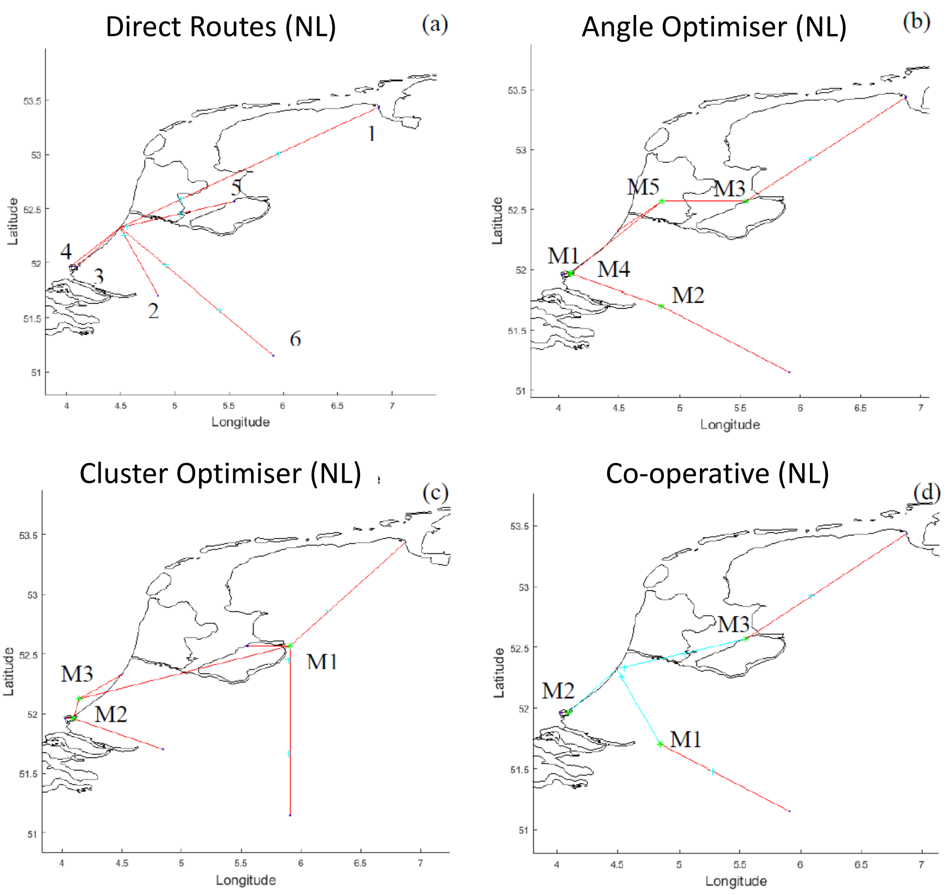


Figure 5.3: Generated networks for the Netherlands case study area.

6 COMPARISON & DISCUSSION

6.1 How do network design strategies impact costs?

Table 6.1 summarises the key outputs from the analysis, grouped by case study area. A range of economic criteria are presented, with the most useful being the capital cost, representing the total cost of constructing the network infrastructure, and the levelised cost, representing the effective cost of transporting a tonne of carbon dioxide from source to the offshore connection point. All monetary values are in 2013 Euros.

Table 6.1: Comparison of network design strategies in the three case study areas

Study Area	Network Design Type	Length (km)	Capital Cost (k€)	Levelised Cost (€/tCO_2)	Cost per km (€)
UK	Direct Connection	30	14,775	1.28	492,388
	Angle Optimiser	14	11,357	0.99	816,986
	Cluster Optimiser	13	11,782	1.02	901,022
	Co-operative design	15	11,607	1.01	772,635
DE	Direct Connection	2,300	1,649,930	29.88	717,445
	Angle Optimiser	1,562	1,369,237	24.80	876,622
	Cluster Optimiser	1,279	1,359,640	24.61	1,063,382
	Co-operative design	1,432	1,292,245	23.40	902,526
NL	Direct Connection	615	360,805	11.17	586,549
	Angle Optimiser	460	354,944	10.99	771,731
	Cluster Optimiser	543	386,172	11.95	711,460
	Co-operative design	430	171,024	5.30	397,453

Unsurprisingly perhaps, Table 6.1 makes clear that the direct connection approach is never going to be economically attractive. In the UK and DE cases it is the most expensive option with respect to both the key economic criteria. More surprisingly it is only the second most expensive network strategy in the NL case, presumably due to some peculiarities of the geometry that have disadvantaged the cluster optimizer. The direct connection is always the cheapest per km of pipeline, as it requires the smallest diameter pipelines, but this is more than offset by the long network lengths required.

The two automatically optimizing network approaches both perform fairly well in all case study areas. Again in the UK and DE cases, the resulting networks are considerably cheaper, and shorter than relying direct connections. For the NL case only one beats the direct connection approach. The evidence suggests that the two automated network optimisers, taken together will reliably beat simplistic network design approaches.

Turning to the highly co-operative approach to network design, this offers the best outcome in two of the case study areas and is marginally the second best in the third. The potential of this approach, representing the GATEWAY philosophy, is examined further in the next section.

6.2 To what extent does taking a cooperative approach provide benefits?

Table 6.2 compares the cost of CO_2 transport using a co-operative strategy of the type proposed in the GATEWAY PCI with the cheapest alternative strategy identified using the various optimised network design methodologies described in Chapter 4. In both the of the Germany and Netherlands case study areas, the co-operative approach offers lower overall CO_2 transport costs than any of the automatically optimized networks.

Table 6.2: Evaluation of benefits of a co-operative approach.

Case Study Area	Levelised CO_2 Transport Cost (€/te)		Percent cost reduction from co-operative approach
	Co-operative strategy	Cheapest other strategy	
UK	1.01	0.989	-2.2%
Germany	29.6	31.1	4.68%
Netherlands	11.2	41.4	41.3%

The cost reduction for the Netherlands case is quite remarkable, with an improvement of more than 40% over the best automatically optimized case. This scale of benefit arises because the Netherlands sites are relatively well aligned geographically meaning the scope for cost saving through carefully scheduled network development is considerable. In the German case study areas, there is still a worthwhile saving of close to 5%, but the size is less impressive due to the geographical diversity of the sources. As a result there is much less scope for cost saving through co-operatively designed and installed shared infrastructure.

For the UK study area, the co-operatively designed network in fact shows a small increase in costs (represented by the negative decrease) compared to the optimally designed networks. This should be interpreted with care, as it is a reflection of the limitations of the analysis tools and the short distances involved. Firstly, as previously mentioned, the co-operative network strategy does not represent a true mathematical optimum, unlike the other strategies with which it is compared. There is therefore likely to be scope for further improving the co-operative strategy in this case, which would yield an overall saving. Secondly the relatively short distances in this case mean there are only marginal differences between any of the considered strategies, as can be seen in Table 6.1.

6.3 Relationships between key network parameters

One strength of the automated and semi-automatic design tools of the type used in this study is that they allow relatively rapid evaluation of a wide range of possible pipeline networks. To try to provide some insight into how to design an economically effective network the relationships between certain key overall network parameters in the three case study areas has been investigated. Costs in this section are reported in 2013 GB Pounds sterling.

The results described in this section include some modified network design approaches not included in the results of Chapter 5. Also included are some variations in the case study area specification with respect to the numbers of carbon dioxide sources incorporated in the network compared to the description in Chapter 3. Again these have not previously been discussed as they are not directly relevant to the current objectives. The outputs from these additional analysis cases help to shed light on some key sensitives and hence have been left in the results reported in this section. As a result,the number of data points in some of the graphs presented is greater than the number of results in Table 6.1

6.3.1 Mass flow rate and levelised costs

As shown in Figure 6.1 for the cluster based optimiser, there seems to be a trend that higher mass flows result in lower levelised costs. However, plots for other network design approaches did not support this trend as strongly.

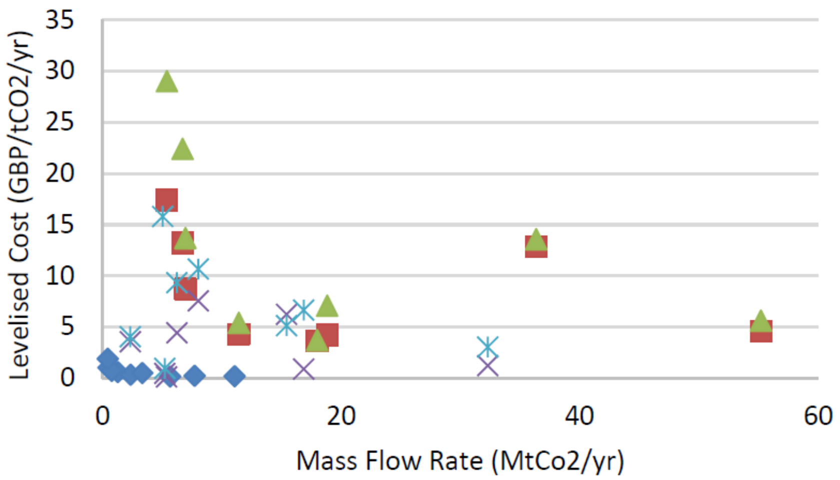


Figure 6.1: Relationship between total network mass flow rate and levelised costs. The differing symbols represent different case study areas and optimisation approaches.

6.3.2 Pipeline length and levelised cost

Comparing all the network design approaches, it was found that longer pipelines resulted in higher levelised costs, as is shown in Figure 6.2 for the direct connection approach. The other design approaches showed very similar relationships. This result is hardly surprising because building longer pipelines necessarily results in higher costs, but it is also clear that pipeline length is not the only factor.

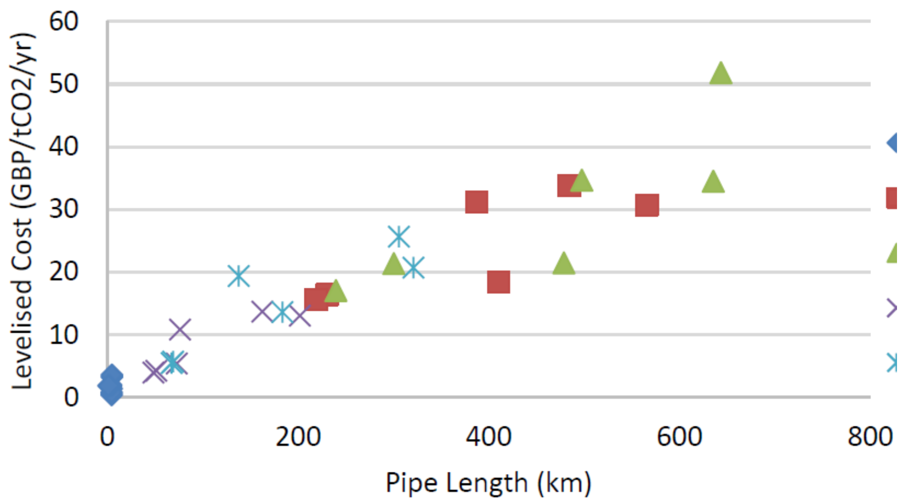


Figure 6.2: Relationship between pipeline length and levelised costs. The differing symbols represent different case study areas and optimisation approaches.

6.3.3 Total mass flow rate and unit cost

In all cases and with all design approaches, the unit cost decreased as the mass flow increased. As an illustration, Figure 6.3 shows the relationship for the direct connection design strategy.

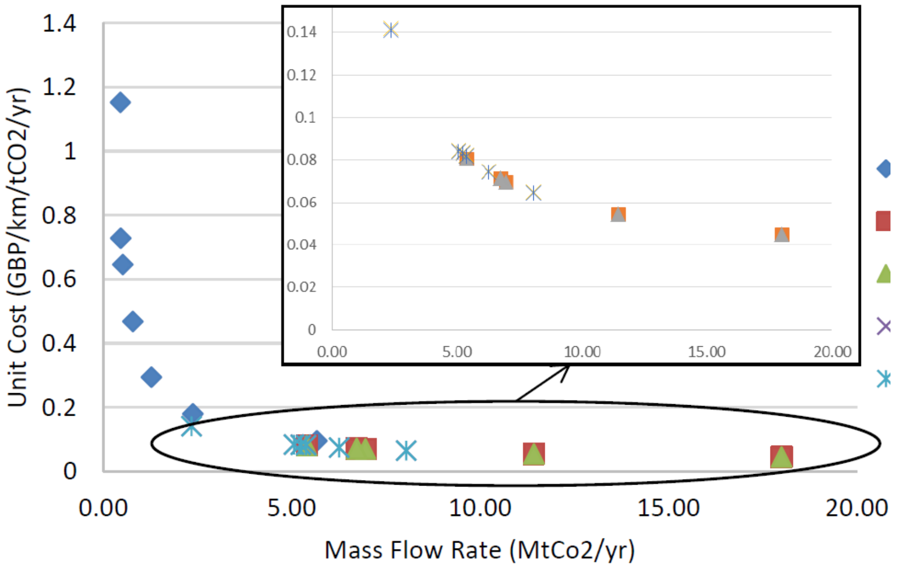


Figure 6.3: Relationship between mass flow rate and unit costs. The differing symbols represent different case study areas and optimisation approaches.

6.4 Brief evaluation of results

To enhance confidence in the network design tools and their results, comparisons were drawn between the generated data and that contained in similar studies found in the literature, as a means of validation. Inevitably, given the limited quantity of similar work, comparison must be made with some studies that have very different assumptions, and investigate pipelines in different locations. At best therefore, it is only possible to hope for a qualitative correspondence, and that similar broad trends will emerge. As in Section 6.3 a slightly wider range of results is included in graphical outputs than discussed elsewhere in the report.

Figure 6.4 shows that the mass flow rate and levelised cost relationship for the UK case follows a similar trend to the USA cases examined by McCoy & Rubin (2008) and the IPCC (2005). The trend was not as clear for the Germany and Netherlands cases, whose levelised costs rose over £30/tCO_2 for some pipes. An increase in levelised cost due to an increase in pipe length was also shown by McCoy & Rubin; the levelised costs for the Germany and Netherlands increased about twice as quickly as their model.

Kjärstd et al. (2011) investigated CCS pipelines in Europe using similar CO_2 characteristics to the ones in this study. Their German case of 18 sources emitting 3,472MtCO_2/year, resulted in a 2,306km network with a levelised cost of 2.73€/tCO_2. The German case in this study had 6 sources emitting 55MtCO_2, producing an average 2,211km network and a levelised cost of €33.90/tCO_2 The levelised cost here is significantly higher, but as Figure 6.1 showed we established a strong decreasing trend of levelised cost with mass flow rate. Extending the trend of Figure 6.1 indicates that the results are not incompatible.

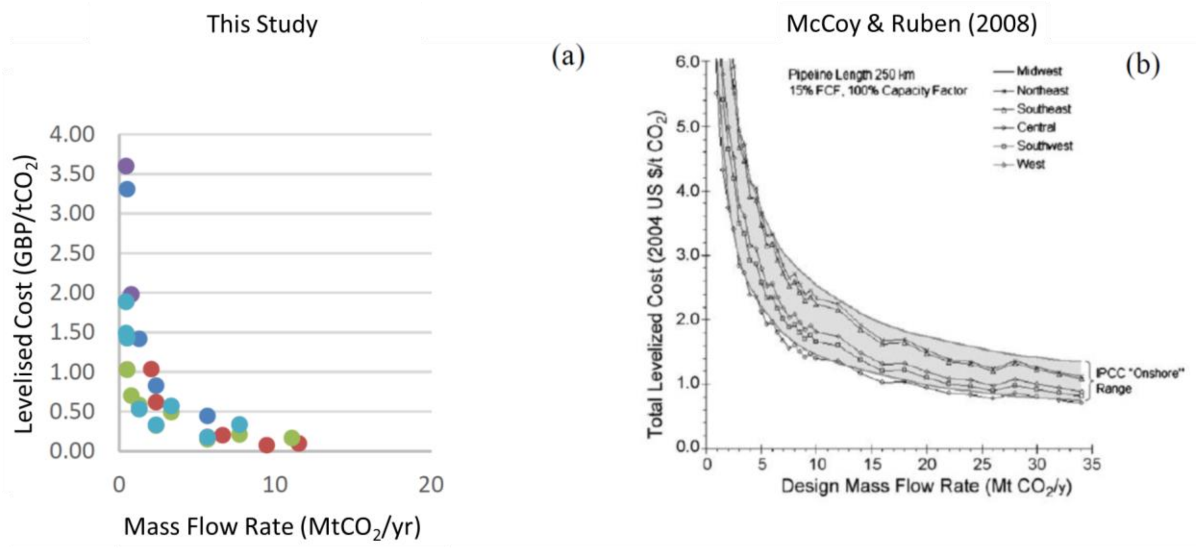


Figure 6.4: Comparison of selected results of this study with McCoy & Ruben (2008). The differing symbols represent different case study areas and optimisation approaches

6.5 What conclusions can be drawn for the GATEWAY project?

The key conclusions from the work reported for the GATEWAY project are as follows.

Firstly, point to point pipeline network strategies, where each source has a dedicated link to the offshore network connection node is always among the most expensive network strategy. There is always some benefit from attempting to co-ordinate network development in some rational way, and hence the premise of the GATEWAY project is valid.

Secondly, in two of the three case study areas considered, the co-operative development strategy offered the network with the most appealing economics. For the UK case the co-operative strategy produced a network that was marginally more expensive than the cheapest network, which was that produced by the cluster optimizing design approach. This result can be attributed to the unusually short distances in the UK case, and the fact that the co-operative network does not represent a true mathematical optimum. It is likely therefore that a more refined algorithm would produce a result in accordance with those for the NL and DE case study areas. Certainly we can conclude the co-operative strategy will produce a network with an economic performance, that is either the best, or very close to it, of all the design algorithms considered in all the case study areas. As the co-operative strategy represents the ethos of the GATEWAY initiative, developing carbon dioxide pipelines in this way will help to optimize the economic performance of CCS in Europe.

Thirdly, co-operative strategies offer the most benefit where sources are geographically aligned, and to a lesser extent, where there are long distances to be covered by pipelines. The likely benefit reduces markedly where sources are less geographically aligned and where distances are short (say of the order of 15km). In fact where distances are short, there is little to choose between any of the strategies considered.

Given the above, the Netherlands case study area appears to have most to gain from adopting a co-operative strategy to CCS network development, and thus this report supports the conclusions of Deliverable 4.1

7 CONCLUSION

7.1 Evaluation of pipeline network development strategies

The work described in this report has explored the application of contrasting approaches to the initial stages of onshore CCS carbon dioxide pipeline networks in three European case study areas. The areas selected are in the UK, Germany and the Netherlands, are comparable to, but not identical to, the candidate Pilot Case areas of the GATEWAY project.

Four development strategies, detailed in Table 7.1 were implemented in spatially explicit MATLAB codes, partially drawing on previous work by some of the authors in this area (Lone et al., 2010). Two automated approaches to identifying economically optimal networks were implemented, as the problem is computationally challenging. The fourth strategy in the table, the "Co-operative strategy" is intended to represent the development approach advocated by the GATEWAY project.

Table 7.1: Summary of pipeline development strategies.

Strategy Name	Brief Description
Direct connection	Carbon dioxide sources are directly connected to a single offshore connection point using one dedicated pipeline per source
Cluster optimiser	An automated approach to identifying economically optimal network topologies that does not explicitly emphasise collaboration in developing shared pipelines.
Angle optimiser	An alternative automated approach to identifying economically optimal network topologies that does not explicitly emphasise collaboration in developing shared pipelines.
Co-operative strategy	A pipeline network approach that focusses on a high degree of collaboration between sources in building shared pipelines. This strategy is intended to represent the development strategy implicitly embodied by the GATEWAY project.

Taken across the three case study areas, the direct connection approach consistently provided the poorest results with the resulting networks being either the most expensive or second most expensive by a number of measures. With one exception the two automated optimisers produced network topologies that were significantly cheaper than the direct connection approach.

In two of the three cases the co-operative strategy resulted in the most economically attractive network. The exception was the UK, were the result was marginally more expensive than that produced by the angle optimizer. This result can be attributed to the unusually short distances in the UK case, and the fact that the co-operative network does not represent a true mathematical optimum. It is likely therefore that a more refined algorithm representing the co-operative strategy would consistently produce the most attractive network.

Hence we can conclude that co-operative strategy consistently produces a network with an economic performance, that is either the best, or very close to it, of all the design algorithms considered in all the case study areas. As the co-operative strategy represents the ethos of the GATEWAY initiative, developing carbon dioxide pipelines in this way will help to optimize the economic performance of CCS in Europe.

7.2 Benefit of a co-operative pipeline development strategy in each case study area

Table 7.2 summarises the cost reduction offered by adopting the co-operative development strategy in each of the case studies, compared to the next cheapest strategy. For the UK case, there is a negative reduction, indicating a small increase. As set out in the previous section, this result arises because of limitations in the implementation of the algorithms and the unusually short distances involved

Table 7.2: Economic benefits of a co-operative strategy.

Case Study Area	Percent cost reduction over next cheapest from adopting co-operative approach
UK	-2.2%
Germany	4.68%
Netherlands	41.3%

Based on these results, co-operative strategies offer the most benefit where CO_2 sources are geographically aligned, and to a lesser extent, where there are long distances to be covered by pipelines. The likely benefit reduces markedly where sources are less geographically aligned and where distances are short. In fact where distances are short, there is little to choose between any of the strategies considered.

REFERENCES

Berghout, N., Cabal, H., Gouveia, J.P., van den Broek, M., & Faaij, A., (2015). *Method for identifying drivers, barriers and synergies related to the deployment of a CO_2 pipeline network: A case study of the Iberian Peninsula and Morocco*. International Journal of Greenhouse Gas Control, Volume 41, pp. 82-106.

Chandel, M., Pratson, L., & Williams, E. (2011). *Potential economies of scale in CO_2 transport through use of a trunk pipeline*. Energy Conversion and Management Volume 51 (2011) pp. 2825-2834.

CoolProp, (2016). *CoolProp Online*. Available at: http://www.coolprop.org/online/index.html [Last Accessed 25 August 2016].

Delling, D., Sanders, P., Schultes, D., & Wagner, D (2009). *Engineering Route Planning Algorithms*, in Algorithmics of Large and Complex Networks. Lecture Notes in Computer Science, vol. 5525, pp. 117-139. Springer ISBN 978-3-642-02093-3.

Doctor, R., Palmer, A., Coleman, D., Davison, J., Hendriks, C., Kaardstad, O., & Ozaki, M., (2005). *Carbon Dioxide Capture and Storage*. Chapter 4. Intergovernmental Panel on Climate Change.

Eickhoff C, Hanstock D, Brunner L, Mikunda T, Neele F (2016); GATEWAY Project Deliverable D4.1: *Pilot Case Definition*; 30th September 2016.

González Ferreras M (2016). *Optimal Modelling of CCS Pipelines*; Masters Thesis, University of Leeds.

IEA Greenhouse Gas R&D Programme (2005). *Building the cost curves for CO_2 Storage: North America*. International Energy Agency Report no. 2005/3.

IEA GHG (2010). *Development of a global CO_2 pipeline infrastructure*. International Energy Agency.

IEA (2013). CO_2 Pipeline Infrastructure Report, IEA GHG Report 2013/18. Available at: http://ieaghg.org/docs/General_Docs/Reports/2013-18.pdf.

Johnson, N., & Ogden, J., (2011). *Detailed spatial modelling of carbon capture and storage (CCS) infrastructure deployment in the southwestern United States*. Energy Procedia, Volume 4, pp.2691-2699.

Kaufmann, I., (2008). *Carbon dioxide transport in pipelines – under special consideration of safety-related aspects*. Pipeline Technology Conference. ILF Beratende Ingenieure GmbH, Germany.

Kjärstad, J., Ramdani, R., Gomez, P., Rootzén, J., & Johnsson, F. (2011). *Establishing an integrated CCS transport infrastructure in northern Europe: challenges and possibilities*. Energy Procedia, Volume 4. pp. 2417-2424.

Kuby, M., Middleton, R. & Bielicki, J., (2011). *Analysis of cost savings from networking pipelines in CCS infrastructure systems*. Energy Procedia, Volume 4. pp. 2808-2815.

Lone, S. (2009) *Power generation to CO_2 disposal network analysis – Development of UK CO_2 transportation infrastructure*. Masters Thesis, Imperial College London.

Lone S, Cockerill TT & Macchietto S (2010). *Techno-economics of a phased approach to developing a UK carbon dioxide pipeline network*; Journal of Pipeline Engineering, Vol. 9, No. 4**,** pp. 223-235.

Luo, X., Wang, M., Oko, E., & Okezue, C., (2014). *Simulation-based techno-economic evaluation for optimal design on CO_2 transport pipeline network*. Applied Energy, Volume 132, pp.610-620.

McCoy, S. & Rubin, E., (2008). *An engineering-economic model of pipeline transport of CO_2 with application to carbon capture and storage*. International Journal of Greenhouse Gas Control, Volume 2, pp. 219.229.

Middleton, R.S., & Bielicki, J.M. (2009a). *A comprehensive Carbon Capture and Storage Infrastructure Model*. Energy Procedia, Volume 1, pp. 1611-1616.

Middleton, R., & Bielicki, J., (2009b). *A scalable infrastructure model for carbon capture and storage: SimCCS*. Energy Policy, Volume 37, pp. 1052-1060.

Middleton, R.S. Kuby, M.J., & Bielicki, J.M., (2012). *Generating candidate networks for optimisation: The CO_2 capture and storage optimization problem*. Computer Environment and Urban Systems, Volume 36, pp.18-29.

Natural Environment Research Council (NERC) (2011). *Temperature and Thermal Properties (Basic)*. British Geological Survey. BGS Report No: GR_999999/1.

NIMA (1998); VMAP level 0 data provided by the US National Imagery and Mapping Agency (1998). Available at: http://earth-info.nga.mil/publications/vmap0.html [last accessed 8 October 2016].

Serpa, J., Morberr, J., & Tzimas, A. (2011). *Technical and Economic Characteristics of a CO_2 Transmission Pipeline Infrastructure.* European Commission, European Union. ISBN 978-92-79-19425-2.

Singh, P., & Haines, M., (2014). *A Review of Existing Carbon Capture and Storage Cluster Projects and Future Opportunities*. Energy Procedia, Volume 63, pp. 7247-7260.

Vandeginste, V., & Piessens, K., (2008). *Pipeline design for a least-cost router application for CO_2 transport in the CO_2 sequestration cycle*. International Journal of Greenhouse Gas Control, Volume 2

Wildenborg, T., Coussy, P., Doukelis, A., Ekström, C., Georgiou, G., Gkountanis, S., Kramers, L., van der Kuip, M., Lindeberg, E., Nordbø, Ø., Serbutoviez, S., & Simonsson, D., (2009). *Scenario for large-scale implementation of CCS in Europe*. Energy Procedia, Volume 1 pp.

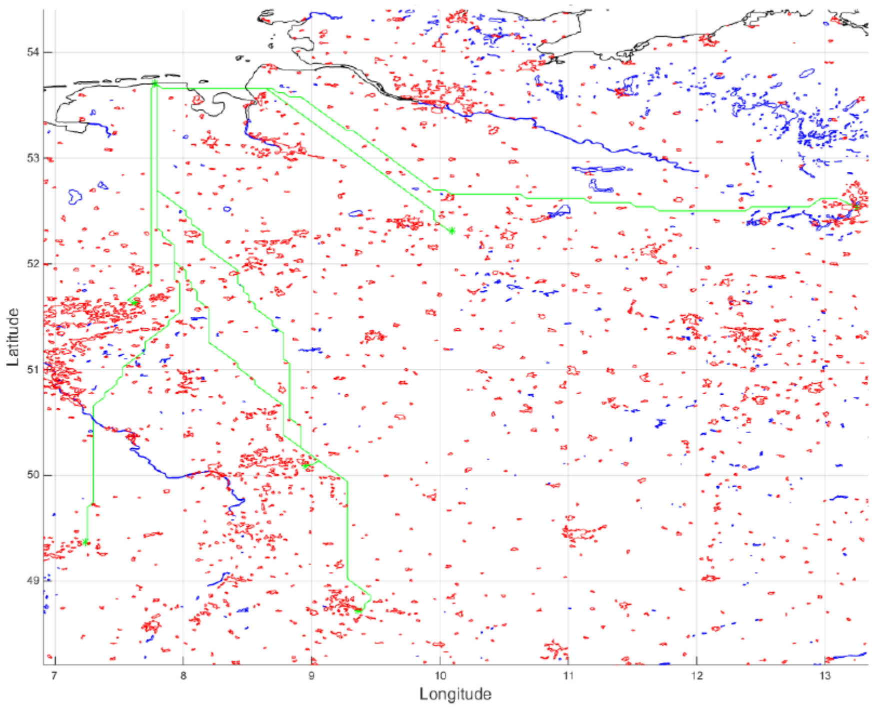
4265-4272.

APPENDIX 1: EXCHANGE RATES

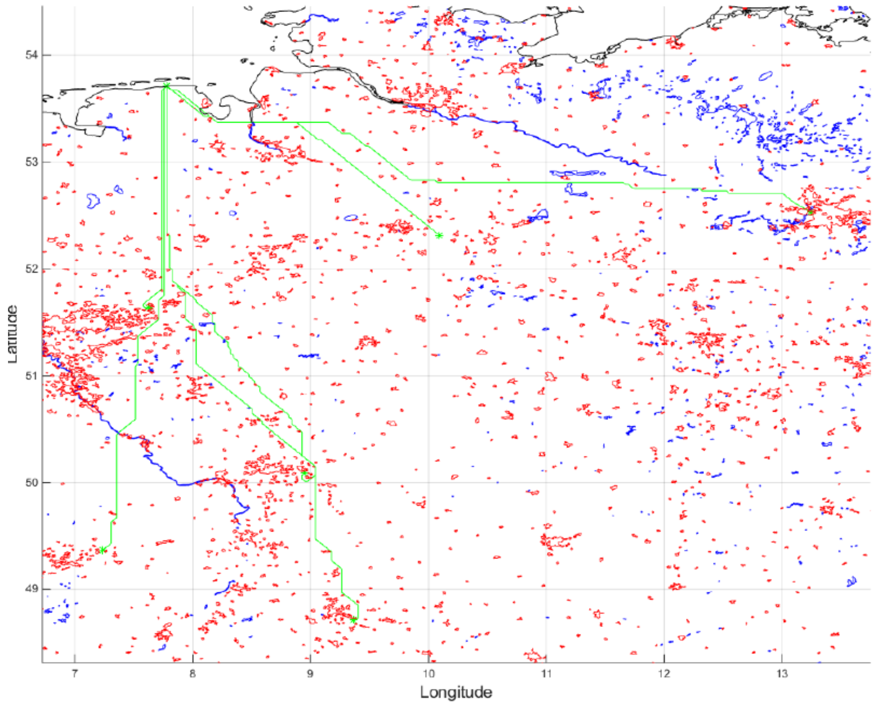
Currency	Rate
British Pounds Sterling	1
Euro	1.136
US Dollar	1.41

APPENDIX 2: OBSTACLE AVOIDANCE GRID INDEPENDENCE TEST PLOTS

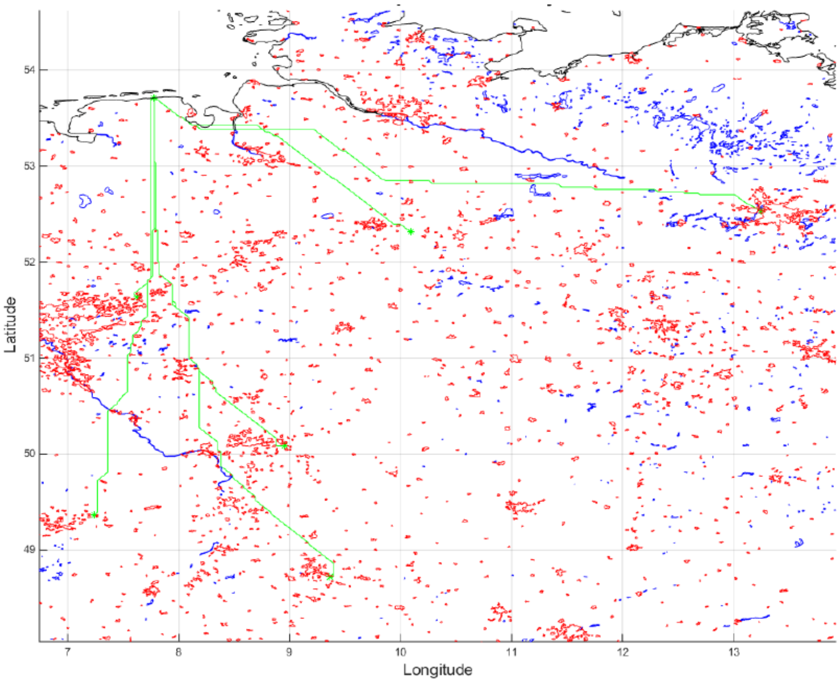
A2.1 Pathways with 1,000 cells



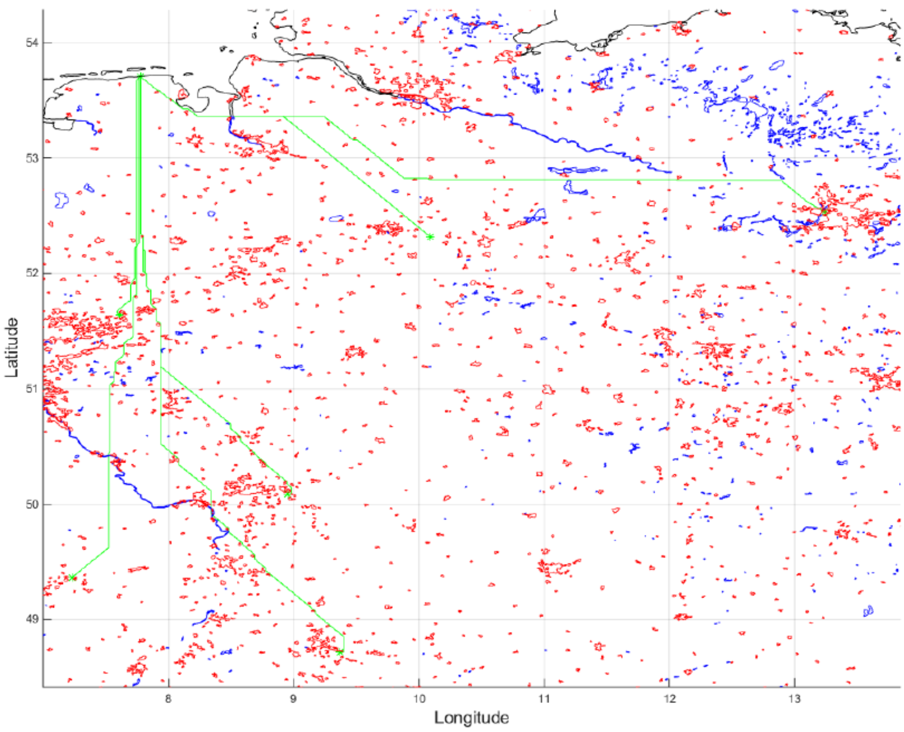
A2.2 Pathways with 2,000 cells



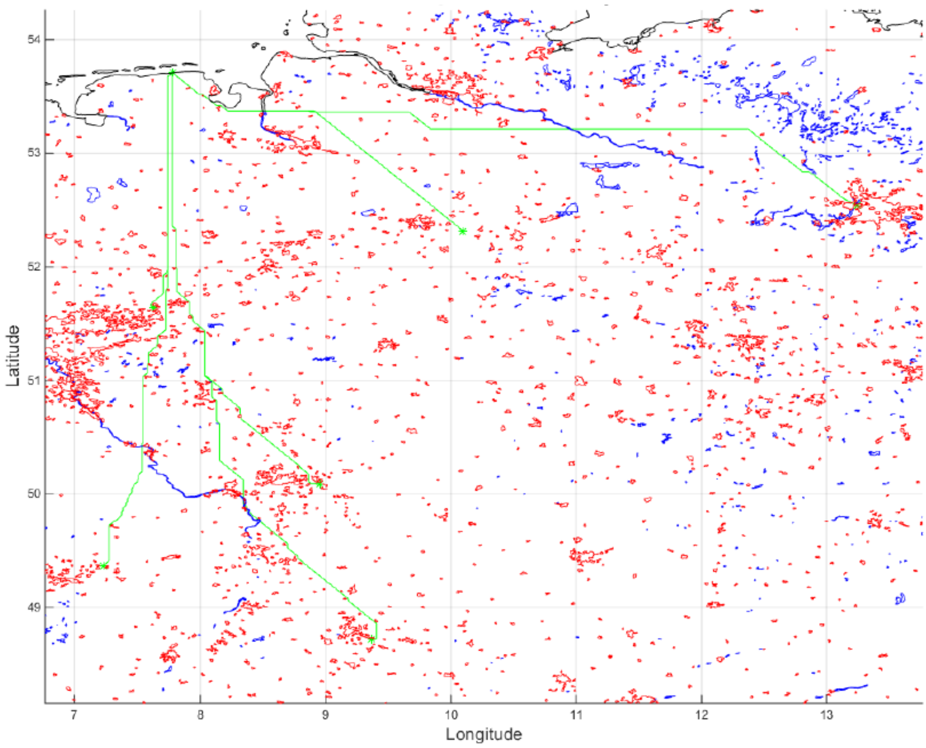
A2.3 Pathways with 3,000 cells



A2.4 Pathways with 4,000 cells



A2.5 Pathways with 5,000 cells



APPENDIX 3: SAMPLE ROUTES USING OBSTACLE AVOIDANCE

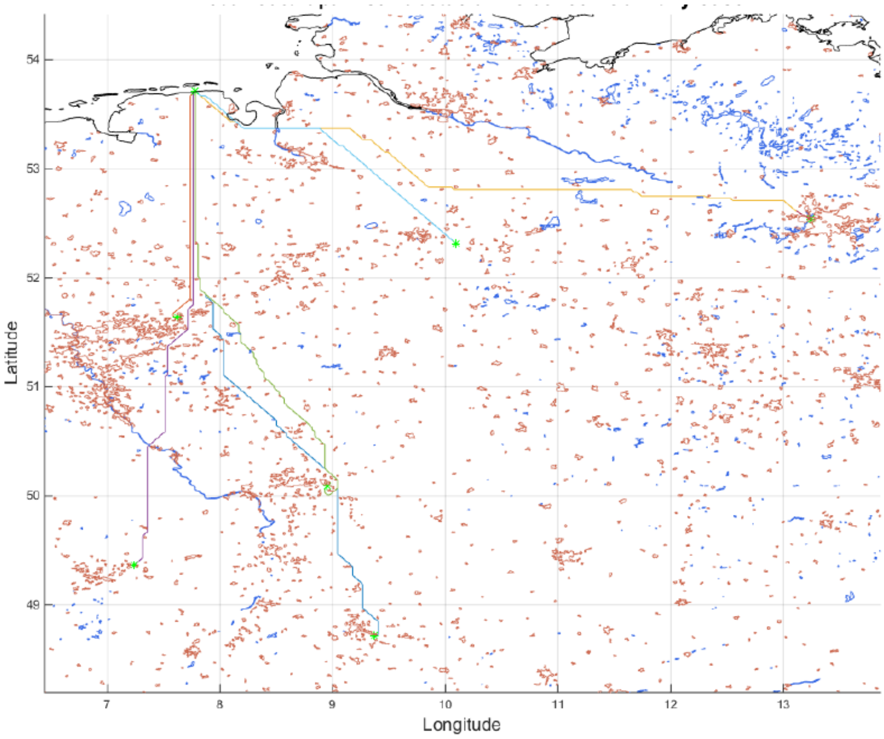
A3.1 Introduction

This Appendix provides network route maps generates when obstacle avoidance was switched on in the pipeline network model. Obstacle avoidance was not incorporated in any UK case study. The short distances involved meant that there were negligible difference between cases with and without obstacle avoidance.

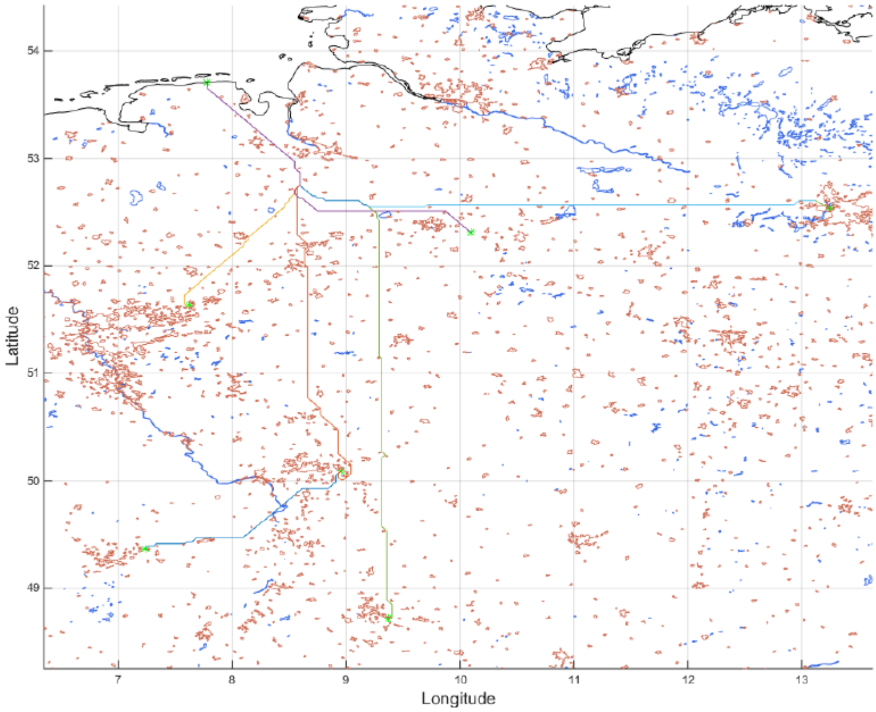
In the figures that follow, areas outlines in red represent built up locations, while those in blue represent significant areas of internal waters. The pipelines (shown by other coloured lines) have been routed to avoid these obstacles. The black lines represent the coastline.

A3.2 Germany

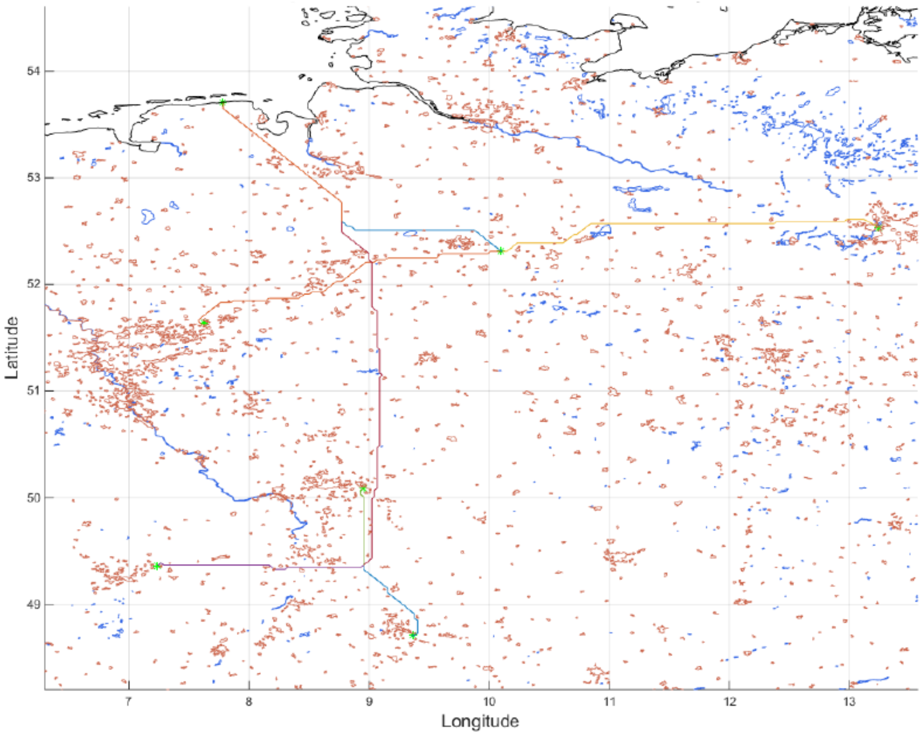
A3.2.1 Direct routes with obstacle avoidance



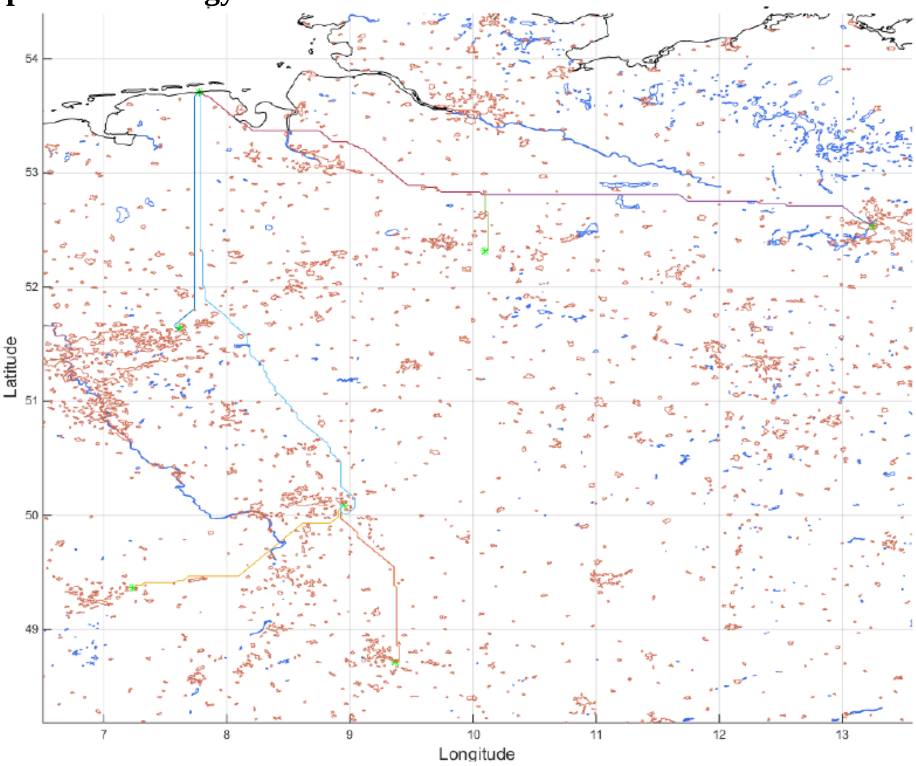
A3.2.2 Angle optimiser with obstacle avoidance



A3.2.3 Cluster optimiser with obstacle avoidance

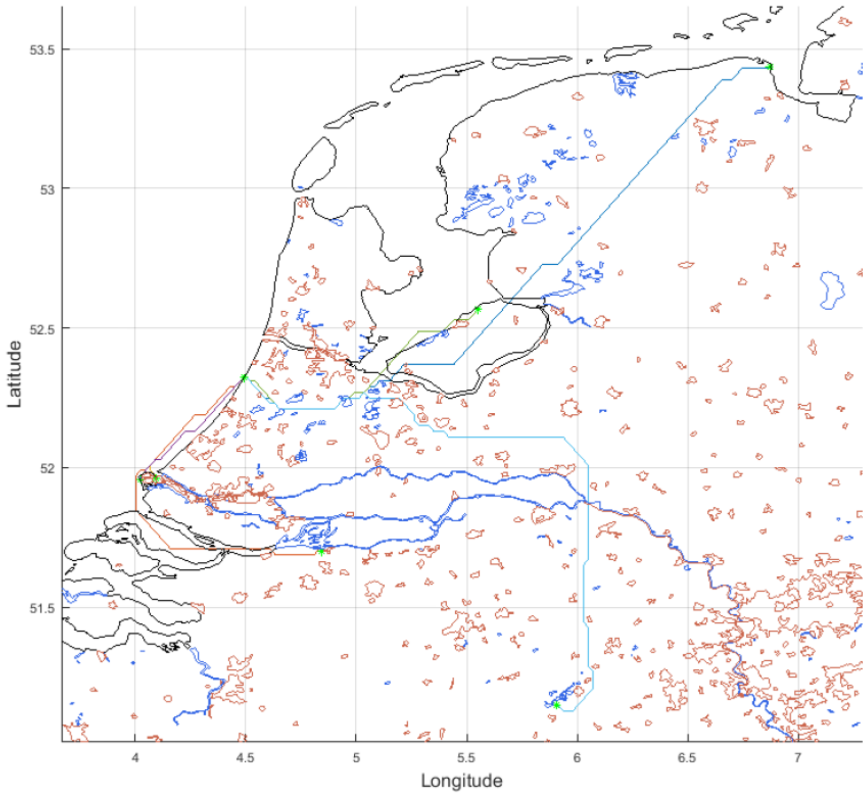


A3.2.4 Co-operative strategy with obstacle avoidance

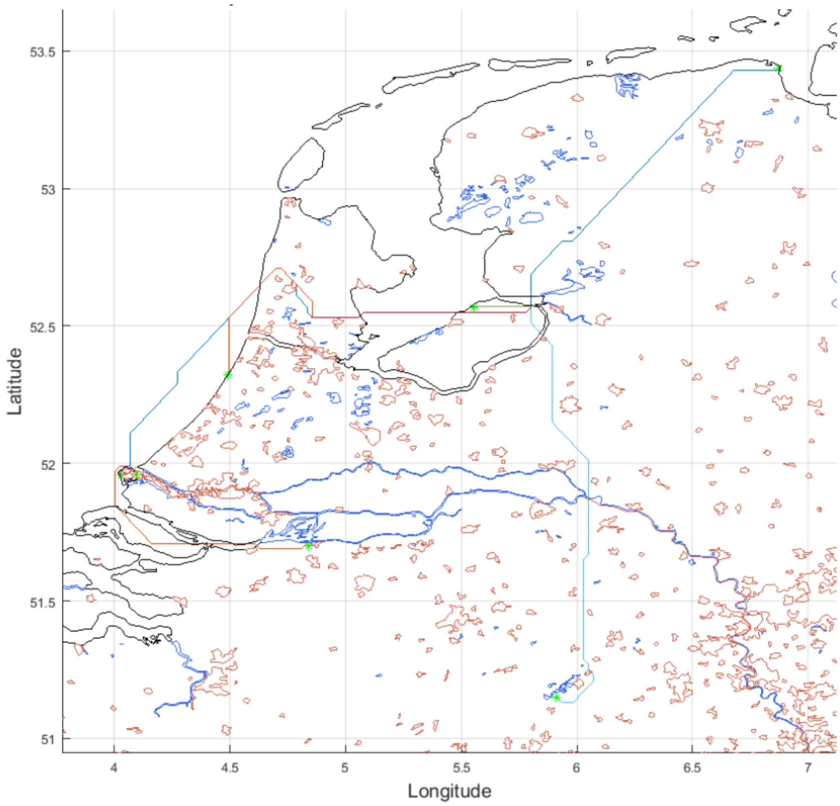


A3.3 Netherlands

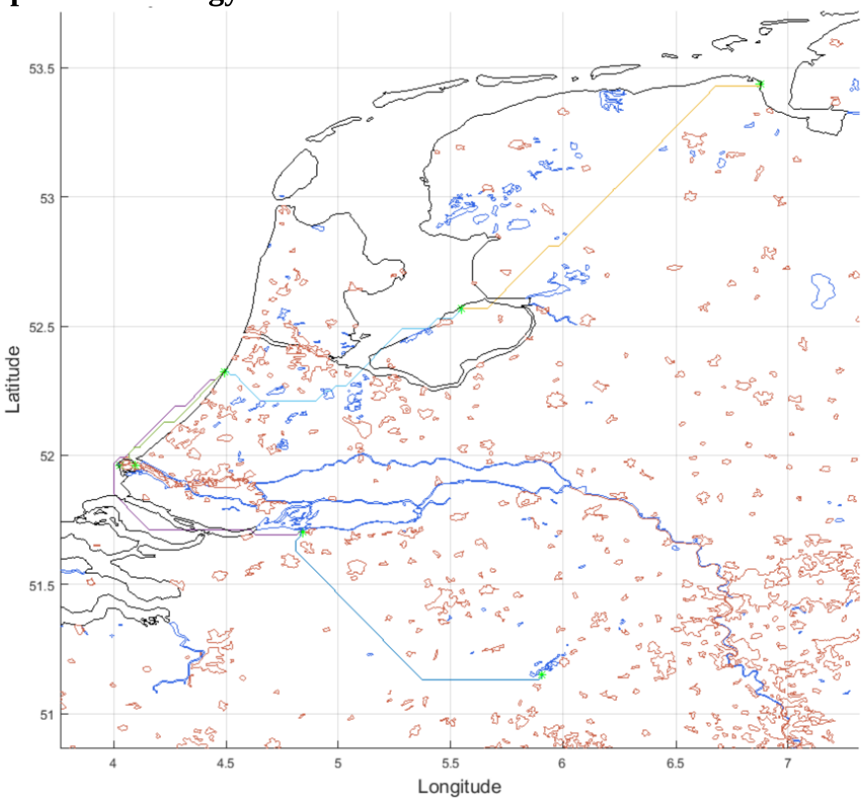
A3.3.1 Direct routes with obstacle avoidance



A3.3.2 Cluster optimiser with obstacle avoidance



A3.3.3 Co-operative strategy with obstacle avoidance



APPENDIX 4: EXAMPLE MATLAB CODE

CODE FOR CLUSTER OPTIMISER

```
% MECH5825 Cluster optimiser with Obstacle avoidance Germany case
% Created 26 July 2016 by Reyes Gonzalez Ferreras
% calculates the optimum path by joining the sources at meeting
points
% using kmeans if the distance to them is smaller than the distance
to the
% reservoir. Then uses the angle optimiser to join the meeting
points.
% Also applies A path optimiser obstacle avoider to the pipes.
%----------------------------------------------------------------------
---

tic
% Set image of Europe
figure
grid on;
hold on;

% Read the data information for latitude and longitude (in decimal)
delimiterIn = '\t';
data = importdata('German_sources.txt',delimiterIn);
% Assign latitude and longitude to a matrix
latitude = data.data(1:end,2);
longitude = data.data(1:end,1);
CO_flow_annual = data.data(1:end,3); %tCo2/year
sources_latlong =  [latitude(1:end-1),longitude(1:end-1)];

% Plot the sources and reservoir
sources_and_res = [latitude(1:end)';longitude(1:end)'];
for i= 1:length(latitude)
plot(sources_and_res(2,i),sources_and_res(1,i),'g*');
end

% Calculate the distances between the points and the reservoir, last
point
% is the reservoir
dist_to_res = zeros(length(latitude)-1,1);
dist_vert_to_res = zeros(length(latitude)-1,1);
dist_hor_to_res = zeros(length(latitude)-1,1);
for i = 1:length(latitude)-1
    dist_to_res(i,1) = pos2dist(latitude(i,1),longitude(i,1),...
                    latitude(end,1),longitude(end,1),2);
    dist_vert_to_res(i,1) = pos2dist(latitude(i,1),longitude(i,1),...
                    latitude(end,1),longitude(i,1),2); % vert
dist to R
    dist_hor_to_res(i,1) = pos2dist(latitude(i,1),longitude(i,1),...
                    latitude(i,1),longitude(end,1),2); % hor dist
to R
end

% Find the cluster points
num_clusters = 2; % Number of clusters wanted; decided a priori
opts = statset('Display','final');
[cluster_to_join,mpoint_lat_long] =
kmeans(sources_latlong,num_clusters,...
                    'Distance','cityblock','Replicates',3,...
                    'Options',opts);
```

```
S_count_for_cluster = zeros(num_clusters,length(latitude)-1);

% Calculate the distance between the clusters and the reservoir
dist_vert_C_to_res = zeros(length(mpoint_lat_long),1);
dist_hor_C_to_res = zeros(length(mpoint_lat_long),1);

for i = 1:length(mpoint_lat_long)
    dist_vert_C_to_res(i,1) = pos2dist(mpoint_lat_long(i,1),...
        mpoint_lat_long(i,2),latitude(end,1),mpoint_lat_long(i,2),2);
    dist_hor_C_to_res(i,1) = pos2dist(mpoint_lat_long(i,1),...

mpoint_lat_long(i,2),mpoint_lat_long(i,1),longitude(end,1),2);
end

% Calculate the distance between each point and its closest cluster
dist_S_to_cluster = zeros(length(latitude)-1,1);
for i = 1:length(latitude)-1
dist_S_to_cluster(i,1) = pos2dist(latitude(i,1),longitude(i,1),...

mpoint_lat_long(cluster_to_join(i,1),1),...

mpoint_lat_long(cluster_to_join(i,1),2),2);

end

% Setting the flow per hour of each pipe and calculating the diameter
CO_flow_hour_S_to_m = CO_flow_annual(1:end-1)*0.9/(365*24); %tCO2/h
pipe_diam_S_to_m = pipe_diam_func(CO_flow_hour_S_to_m);

% point 3 has to be moved because it is surrounded by obstacles.
Remember
% to add the direct distance from the original location at the end
point3_new_lat = 52.61;
point3_new_long = 13.12;
point3_orin_lat = latitude(3,1);
point3_orin_long = longitude(3,1);
point3dist = pos2dist(point3_new_lat,point3_new_long,...
                      latitude(3,1),longitude(3,1),2);
latitude(3,1) = point3_new_lat;
longitude(3,1) = point3_new_long;

% Set matrices to be filled
pipe_length_S_to_m = zeros(length(latitude)-1,1);
length_obs_S_to_m = zeros(length(latitude)-1,1);
pipe_cost_S_to_m = zeros(length(latitude)-1,1);
outlet_conds_S_to_m = zeros(length(latitude)-1,3);
booster_stations_S_to_m = zeros(length(latitude)-1,5);
booster_lat_S_to_m = zeros(length(latitude)-1,10);
booster_long_S_to_m = zeros(length(latitude)-1,10);

% Join the sources to the cluster points or the reservoir
for i = 1:length(latitude)-1
% if the distance to the cluster is smaller, join to cluster point
if dist_S_to_cluster(i,1) < dist_to_res(i,1)
    pipe_length_S_to_m(i,1) = dist_S_to_cluster(i,1);
    pipe_cost_S_to_m(i,1) = pipe__cap_cost_func...
                    (pipe_diam_S_to_m(i,1),pipe_length_S_to_m(i,1));

    % Use the Astar algorithm to avoid obstacles from each source to
its
```

```
    % meeting point.
    Optimum_path_latlong_l = obstacle_avoider_func(...
                             latitude(i,1),longitude(i,1),...
mpoint_lat_long(cluster_to_join(i,1),1),...
mpoint_lat_long(cluster_to_join(i,1),2));
    plot(Optimum_path_latlong_l(:,1),Optimum_path_latlong_l(:,2));

    if length(Optimum_path_latlong_l)>2
        distances_obsl = zeros(length(Optimum_path_latlong_l),1);
        for j = 1:length(Optimum_path_latlong_l)-1
        distances_obsl(j,1) =
pos2dist(Optimum_path_latlong_l(j,1),...
        Optimum_path_latlong_l(j,2),Optimum_path_latlong_l(j+1,1),...
        Optimum_path_latlong_l(j+1,2),2);
        length_obs_S_to_m(i,1) = sum(distances_obsl);
        % Add point 3 distance if needed
        if latitude(i,1) == point3_new_lat
        length_obs_S_to_m(i,1)= length_obs_S_to_m(i,1)+point3dist;
        end
        end
    else
       length_obs_S_to_m(i,1)=0;
    end
    Optimum_path_latlong_l = [];

    % Calculate the pipe's outlet conditions and booster stations
    if length_obs_S_to_m(i,1) > 1
        [outlet_conds_S_to_m(i,:),booster_stations_S_to_m(i,:)]=...
        pipe_pressure_drop_func(length_obs_S_to_m(i,1),...
        CO_flow_hour_S_to_m(i,1));
    else
        outlet_conds_S_to_m(i,:) = 0;
        booster_stations_S_to_m(i,:) = 0;
    end

    % Track which sources are joined to which pipe
    S_count_for_cluster(cluster_to_join(i,1),i) = i;

else % if the dist to the reservoir is shorter than the distance to
cluster
    pipe_length_S_to_m(i,1) = dist_to_res(i,1);
        pipe_cost_S_to_m(i,1) = pipe__cap_cost_func...
(pipe_diam_S_to_m(i,1),pipe_length_S_to_m(i,1));

    % Use the Astar algorithm to avoid obstacles from each source to
its
    % meeting point.
    Optimum_path_latlong_l = obstacle_avoider_func(...
                             latitude(i,1),longitude(i,1),...
mpoint_lat_long(cluster_to_join(i,1),1),...
mpoint_lat_long(cluster_to_join(i,1),2));
    plot(Optimum_path_latlong_l(:,1),Optimum_path_latlong_l(:,2));

    if length(Optimum_path_latlong_l)>2
        distances_obsl = zeros(length(Optimum_path_latlong_l),1);
```

```
        for j = 1:length(Optimum_path_latlong_1)-1
        distances_obs1(j,1) =
pos2dist(Optimum_path_latlong_1(j,1),...
        Optimum_path_latlong_1(j,2),Optimum_path_latlong_1(j+1,1),...
        Optimum_path_latlong_1(j+1,2),2);
        length_obs_S_to_m(i,1) = sum(distances_obs1);
        % Add point 3 distance if needed
        if latitude(i,1) == point3_new_lat
        length_obs_S_to_m(i,1)= length_obs_S_to_m(i,1)+point3dist;
        end
        end
    else
       length_obs_S_to_m(i,1)=0;
    end
    Optimum_path_latlong_1 = [];

    % Calculate the pipe's outlet conditions and booster stations
    if length_obs_S_to_m(i,1) > 1
        [outlet_conds_S_to_m(i,:),booster_stations_S_to_m(i,:)]=...
        pipe_pressure_drop_func(length_obs_S_to_m(i,1),...
        CO_flow_hour_S_to_m(i,1));
    else
        outlet_conds_S_to_m(i,:) = 0;
        booster_stations_S_to_m(i,:) = 0;
    end

end
end
%-------------------------------------------------------

% M POINT ANALYSIS - Angle optimiser
% Save the distance between meeting points and their latitude and
longitude
m1_xcoord = dist_hor_C_to_res;
m1_ycoord = dist_vert_C_to_res;
m_point_lat_temp = mpoint_lat_long(:,1);
m_point_long_temp = mpoint_lat_long(:,2);

% Calculate the flow and pipe diameter of the meeting points
m_CO_flow_hour = zeros(num_clusters,1);
for i = 1:num_clusters
m_CO_flow_hour(i,1) = sum(CO_flow_hour_S_to_m(...
                            S_count_for_cluster(i,:)>0,1));

end
m_CO_flow_hour_save = m_CO_flow_hour;
m_pipe_diam = pipe_diam_func(m_CO_flow_hour);

% Setting the matrices for the initial meeting points
CO_flow_hour_m2point = zeros(num_clusters-1,1);
pipe_diam_m2point = zeros(num_clusters-1,1);
m2_point_xcoord = zeros(num_clusters-1,1);
m2_point_ycoord = zeros(num_clusters-1,1);
m_closest_row = zeros(num_clusters-1,1);
m_closest_col = zeros(num_clusters-1,1);
save_m1_point_lat = zeros(num_clusters-1,1);
save_m1_point_long = zeros(num_clusters-1,1);
CO_flow_mpoint_save = zeros(num_clusters-1,1);
pipe_diam_mpoint_save = zeros(num_clusters-1,1);
length_m_to_m = zeros(2*num_clusters-1,1);
```

```
length_obs_m_to_m = zeros(2*num_clusters-1,1);
pipe_cost_m_to_m = zeros(2*num_clusters-1,1);
outlet_conds_m_to_m = zeros(2*num_clusters-1,3);
booster_stations_m_to_m = zeros(2*num_clusters-1,5);
% Matrices to save the location of the booster stations
booster_lat_m_to_m = zeros(2*num_clusters-1,10);
booster_long_m_to_m = zeros(2*num_clusters-1,10);

% Find all the meeting points between the meeting points
for w = 1:num_clusters-1
if num_clusters-1 > 1

    % Distance between meeting points
        dist_bet_mpoints = zeros(length(ml_xcoord)-
1,length(ml_xcoord)-1);
    for i = 1:length(ml_xcoord)
        for j = 1:length(ml_xcoord)
            dist_bet_mpoints(i,j) = sqrt(((ml_xcoord(j,1)-...
                                    ml_xcoord(i,1))^2)+...
                                    ((ml_ycoord(j,1)-
ml_ycoord(i,1))^2));
        end
    end

    % Find shortest distance between 2 points
    m_closest_distances = zeros(1,min(size(dist_bet_mpoints)));
    m_sort_closest_points = sort(dist_bet_mpoints); % min is on 1st
row
    m_sort_closest_points(m_sort_closest_points==0) = inf;
    m_closest_dist = min(m_sort_closest_points(2,1:end));
    [m_closest_row(w,1),m_closest_col(w,1)] = ...

find(m_closest_dist==dist_bet_mpoints,1);

    % Calculating the cost and diameter of the pipe from the meeting
    % point to the Reservoir
    CO_flow_hour_m2point(w,1) =
m_CO_flow_hour(m_closest_col(w,1),1)+...
                m_CO_flow_hour(m_closest_row(w,1),1); % from mid
point to R
    pipe_diam_m2point(w,1) =
pipe_diam_func(CO_flow_hour_m2point(w,1));
    CO_flow_mpoint_save(2*w-1,1) =
m_CO_flow_hour(m_closest_col(w,1),1);
    CO_flow_mpoint_save(2*w,1) =
m_CO_flow_hour(m_closest_row(w,1),1);
    pipe_diam_mpoint_save(2*w-1,1) = pipe_diam_func(...
                                    CO_flow_mpoint_save(2*w-1,1));
    pipe_diam_mpoint_save(2*w,1) = pipe_diam_func(...
                                CO_flow_mpoint_save(2*w,1));

    % Distances and costs for optimisation function
two_point_costfunc
    x1 = (ml_xcoord(m_closest_col(w,1)));
    y1 = (ml_ycoord(m_closest_col(w,1)));
    x2 = (ml_xcoord(m_closest_row(w,1)));
    y2 = (ml_ycoord(m_closest_row(w,1)));
    pipe_diam1 = m_pipe_diam(m_closest_col(w));
    pipe_diam2 = m_pipe_diam(m_closest_row(w));
    pipe_diam3 = pipe_diam_m2point(w,1);
```

```
    [m2_point_xcoord(w,1),m2_point_ycoord(w,1),...
        length_m_to_m(2*w-1:2*w),pipe_cost_m_to_m(2*w-1:2*w)] = ...

meeting_point_func(x1,y1,x2,y2,pipe_diam1,pipe_diam2,pipe_diam3);

    % transform point to lat-long and plot the lines
    [m1_point_lat,m1_point_long] = km_to_lat_long(...
        latitude(end,1),longitude(end,1),...
        m_point_lat_temp(m_closest_col(w),1),...
        m_point_long_temp(m_closest_col(w,1),1),...
        m_point_lat_temp(m_closest_row(w),1),...
        m_point_long_temp(m_closest_row(w,1),1),x1,y1,x2,y2,...
        m2_point_xcoord(w,1),m2_point_ycoord(w,1));

    %-----------------------------------

    % Use the Astar algorithm to avoid obstacles from each source to
its
    % meeting point. 2 pipes eare built each time
    Optimum_path_latlong_1 = obstacle_avoider_func(...

m_point_lat_temp(m_closest_col(w),1),...

m_point_long_temp(m_closest_col(w,1),1),...
                                m1_point_lat,m1_point_long);
    Optimum_path_latlong_2 = obstacle_avoider_func(...

m_point_lat_temp(m_closest_row(w),1),...

m_point_long_temp(m_closest_row(w,1),1),...
                                m1_point_lat,m1_point_long);
    plot(Optimum_path_latlong_1(:,1),Optimum_path_latlong_1(:,2));
    plot(Optimum_path_latlong_2(:,1),Optimum_path_latlong_2(:,2));

    if length(Optimum_path_latlong_1)>2
        distances_obs1 = zeros(length(Optimum_path_latlong_1),1);
        for j = 1:length(Optimum_path_latlong_1)-1
        distances_obs1(j,1) =
pos2dist(Optimum_path_latlong_1(j,1),...

Optimum_path_latlong_1(j,2),Optimum_path_latlong_1(j+1,1),...
            Optimum_path_latlong_1(j+1,2),2);
            length_obs_m_to_m(2*w-1,1) = sum(distances_obs1);
        end
    else
        length_obs_m_to_m(2*w-1,1)=0;
    end

    if length(Optimum_path_latlong_2)>2
        distances_obs2 = zeros(length(Optimum_path_latlong_2),1);
        for j = 1:length(Optimum_path_latlong_2)-1
        distances_obs2(j,1) =
pos2dist(Optimum_path_latlong_2(j,1),...

Optimum_path_latlong_2(j,2),Optimum_path_latlong_2(j+1,1),...
            Optimum_path_latlong_2(j+1,2),2);
            length_obs_m_to_m(2*w,1) = sum(distances_obs2);
        end
    else
        length_obs_m_to_m(2*w,1)=0;
```

```
    end
    Optimum_path_latlong_1 = [];
    Optimum_path_latlong_2 = [];
    %-----------------------------------

    % Calculate the pipe outlet conditions and booster stations
    % pipe 1
    if length_obs_m_to_m(2*w-1) > 1
        [outlet_conds_m_to_m(2*w-1,:),booster_stations_m_to_m(2*w-
1,:)]=...
        pipe_pressure_drop_func(length_obs_m_to_m(2*w-1),...
        m_CO_flow_hour(m_closest_col(w,1),1));
    else
        outlet_conds_m_to_m(2*w-1,:) = 0;
        booster_stations_m_to_m(2*w-1,:) = 0;
    end
    % pipe 2
    if length_obs_m_to_m(2*w) > 1
        [outlet_conds_m_to_m(2*w,:),booster_stations_m_to_m(2*w,:)]
= ...
        pipe_pressure_drop_func(length_obs_m_to_m(2*w),...
        m_CO_flow_hour(m_closest_row(w,1),1));
    else
        outlet_conds_m_to_m(2*w,:) = 0;
        booster_stations_m_to_m(2*w,:) = 0;
    end

    %-----------------------------------

    % Eliminate joined points from list (rows and columns)
    m1_xcoord(m_closest_col(w,1)) = [];
    m1_xcoord(m_closest_row(w,1)-1) = [];
    m1_ycoord(m_closest_col(w,1)) = [];
    m1_ycoord(m_closest_row(w,1)-1) = [];
    m_point_lat_temp(m_closest_col(w,1)) = [];
    m_point_lat_temp(m_closest_row(w,1)-1) = [];
    m_point_long_temp(m_closest_col(w,1)) = [];
    m_point_long_temp(m_closest_row(w,1)-1) = [];

    % Add new point to remaining points and calculate new distances
    m1_xcoord(end+1,1) = m2_point_xcoord(w,1);
    m1_ycoord(end+1,1) = m2_point_ycoord(w,1);
    m_point_lat_temp(end+1,1) = m1_point_lat;
    m_point_long_temp(end+1,1) = m1_point_long;
    save_m1_point_lat(w,1) = m1_point_lat;
    save_m1_point_long(w,1) = m1_point_long;

    % Update flow and diameter for new meeting point and eliminate
flows
    % and diameters used
    m_CO_flow_hour(end+1,1) =
m_CO_flow_hour(m_closest_col(w,1),1)+...
                m_CO_flow_hour(m_closest_row(w,1),1); % from mid
point to R
    m_pipe_diam(end+1,1) = pipe_diam_func(m_CO_flow_hour(end,1));
    m_CO_flow_hour(m_closest_col(w,1)) = [];
    m_CO_flow_hour(m_closest_row(w,1)-1) = [];
    m_pipe_diam(m_closest_col(w,1)) = [];
    m_pipe_diam(m_closest_row(w,1)-1) = [];
```

```
elseif length(m_CO_flow_hour) > 1 %final meeting point

    % Calculating the cost of the pipe from the meeting point to the
Res
    CO_flow_hour_m2point(w,1) = m_CO_flow_hour(1,1)+...
                m_CO_flow_hour(2,1); % from mid point to res
    pipe_diam_m2point(w,1) =
pipe_diam_func(CO_flow_hour_m2point(w,1));
    CO_flow_mpoint_save(2*w-1,1) = m_CO_flow_hour(1,1);
    CO_flow_mpoint_save(2*w,1) = m_CO_flow_hour(2,1);
    pipe_diam_mpoint_save(2*w-1,1) = pipe_diam_func(...
                                CO_flow_mpoint_save(2*w-1,1));
    pipe_diam_mpoint_save(2*w,1) = pipe_diam_func(...
                               CO_flow_mpoint_save(2*w,1));

    % Distances and costs for function two_point_costfunc
    x1 = (m1_xcoord(1,1));
    y1 = (m1_ycoord(1,1));
    x2 = (m1_xcoord(2,1));
    y2 = (m1_ycoord(2,1));
    pipe_diam1 = m_pipe_diam(1);
    pipe_diam2 = m_pipe_diam(2);
    pipe_diam3 = pipe_diam_m2point(w,1);
    [m2_point_xcoord(w,1),m2_point_ycoord(w,1),...
        length_m_to_m(2*w-1:2*w),pipe_cost_m_to_m(2*w-1:2*w)] = ...

meeting_point_func(x1,y1,x2,y2,pipe_diam1,pipe_diam2,pipe_diam3);

    % Calculate the lat-long from source to meeting point
    [m2_point_lat,m2_point_long] = km_to_lat_long(...
        latitude(end,1),longitude(end,1),...
        m_point_lat_temp(1,1),...
        m_point_long_temp(1,1),...
        m_point_lat_temp(2,1),...
        m_point_long_temp(2,1),x1,y1,x2,y2,...
        m2_point_xcoord(w,1),m2_point_ycoord(w,1));

    %-----------------------------------
    % Use the Astar algorithm to avoid obstacles from each source to
its
    % meeting point. 2 pipes eare built each time
    Optimum_path_latlong_1 = obstacle_avoider_func(...
                            m_point_lat_temp(1,1),...
                            m_point_long_temp(1,1),...
                            m2_point_lat,m2_point_long);
    Optimum_path_latlong_2 = obstacle_avoider_func(...
                            m_point_lat_temp(2,1),...
                            m_point_long_temp(2,1),...
                            m2_point_lat,m2_point_long);
    plot(Optimum_path_latlong_1(:,1),Optimum_path_latlong_1(:,2));
    plot(Optimum_path_latlong_2(:,1),Optimum_path_latlong_2(:,2));

    if length(Optimum_path_latlong_1)>2
        distances_obs1 = zeros(length(Optimum_path_latlong_1),1);
        for j = 1:length(Optimum_path_latlong_1)-1
        distances_obs1(j,1) =
pos2dist(Optimum_path_latlong_1(j,1),...

Optimum_path_latlong_1(j,2),Optimum_path_latlong_1(j+1,1),...
            Optimum_path_latlong_1(j+1,2),2);
```

```
            length_obs_m_to_m(2*w-1,1) = sum(distances_obs1);
        end
    else
        length_obs_m_to_m(2*w-1,1)=0;
    end

    if length(Optimum_path_latlong_2)>2
        distances_obs2 = zeros(length(Optimum_path_latlong_2),1);
        for j = 1:length(Optimum_path_latlong_2)-1
        distances_obs2(j,1) =
pos2dist(Optimum_path_latlong_2(j,1),...

Optimum_path_latlong_2(j,2),Optimum_path_latlong_2(j+1,1),...
            Optimum_path_latlong_2(j+1,2),2);
            length_obs_m_to_m(2*w,1) = sum(distances_obs2);
        end
    else
        length_obs_m_to_m(2*w,1)=0;
    end

    Optimum_path_latlong_1 = [];
    Optimum_path_latlong_2 = [];
    %-----------------------------------

    % Calculate the pipe outlet conditions and booster stations
    % pipe 1
    if length_obs_m_to_m(2*w-1) > 1
        [outlet_conds_m_to_m(2*w-1,:),booster_stations_m_to_m(2*w-
1,:)]=...
        pipe_pressure_drop_func(length_obs_m_to_m(2*w-1),...
                                m_CO_flow_hour(1,1));
    else
        outlet_conds_m_to_m(2*w-1,:) = 0;
        booster_stations_m_to_m(2*w-1,:) = 0;
    end
    % pipe 2
    if length_obs_m_to_m(2*w) > 1
        [outlet_conds_m_to_m(2*w,:),booster_stations_m_to_m(2*w,:)]
= ...
        pipe_pressure_drop_func(length_obs_m_to_m(2*w),...
                                m_CO_flow_hour(2,1));
    else
        outlet_conds_m_to_m(2*w,:) = 0;
        booster_stations_m_to_m(2*w,:) = 0;
    end

%-----------------------------------
% Final Pipe
    % Calculate the final pipe's length and cost
    length_m_to_m(end,1) = sqrt(((m2_point_xcoord(end,1))^2)+...
                                ((m2_point_ycoord(end,1))^2)); % Res
at 0,0
    pipe_cost_m_to_m(end,1) = pipe__cap_cost_func(...

pipe_diam_m2point(w,1),length_m_to_m(end,1));

    % Use the Astar algorithm to avoid obstacles from each source to
its
    % meeting point. 2 pipes eare built each time
    Optimum_path_latlong_1 = obstacle_avoider_func(...
```

```
                            m2_point_lat,...
                            m2_point_long,...
                            latitude(end,1),longitude(end,1));
    plot(Optimum_path_latlong_1(:,1),Optimum_path_latlong_1(:,2));

    if length(Optimum_path_latlong_1)>2
        distances_obs1 = zeros(length(Optimum_path_latlong_1),1);
        for j = 1:length(Optimum_path_latlong_1)-1
            distances_obs1(j,1) =
pos2dist(Optimum_path_latlong_1(j,1),...

Optimum_path_latlong_1(j,2),Optimum_path_latlong_1(j+1,1),...
            Optimum_path_latlong_1(j+1,2),2);
            length_obs_m_to_m(end,1) = sum(distances_obs1);
        end
    else
        length_obs_m_to_m(end,1)=0;
    end

    % Calculate the final pipe outlet conditions and booster stations
    if length_obs_m_to_m(end,1) > 1
        [outlet_conds_m_to_m(end,:),booster_stations_m_to_m(end,:)]
= ...
        pipe_pressure_drop_func(length_obs_m_to_m(end,1),...
        CO_flow_hour_m2point(end,1));
    else
        outlet_conds_m_to_m(end,:) = 0;
        booster_stations_m_to_m(end,:) = 0;
    end

end
end
%----------------------------------------------------------------------
---

% Gather all the information

% Putting together all the meeting points
m_points_xcoords = [dist_hor_C_to_res;m2_point_xcoord];
m_points_ycoords = [dist_vert_C_to_res;m2_point_ycoord];

% eliminate all 0 lats and longs
save_m1_point_lat(save_m1_point_lat==0) = [];
save_m1_point_long(save_m1_point_long==0) = [];

% Put together all latitudes and longitudes
m_points_lat = [mpoint_lat_long(:,1);save_m1_point_lat;m2_point_lat];
m_points_long =
[mpoint_lat_long(:,2);save_m1_point_long;m2_point_long];

% Put together all pipeline lengths, flows, costs and outlet
conditions
pipeline_lengths = [pipe_length_S_to_m;length_m_to_m];
pipeline_real_lengths = [length_obs_S_to_m;length_obs_m_to_m];
pipeline_flows = [CO_flow_hour_S_to_m;CO_flow_mpoint_save;...
                  CO_flow_hour_m2point(end,1)];
pipeline_costs = [pipe_cost_S_to_m;pipe_cost_m_to_m];
pipeline_diams = pipe_diam_func(pipeline_flows);
pipeline_real_costs = zeros(length(pipeline_diams),1);
for i = 1:length(pipeline_diams)
```

```
pipeline_real_costs(i,1) =
pipe__cap_cost_func(pipeline_diams(i,1),...

pipeline_real_lengths(i,1));
end
pipeline_outlet_cond = [outlet_conds_S_to_m;outlet_conds_m_to_m];

% Booster station data for all pipes
booster_stations_data =
[booster_stations_S_to_m;booster_stations_m_to_m];

%-------------------------------------------------------------------
---

% plot the line from point 3 to its new start
plot([point3_orin_long,point3_new_long],[point3_orin_lat,point3_new_l
at]);

% Plot the coastine
coastline = shaperead('vmap0_coastline.shp');
for i = 1:length(coastline)
xcoords_obs_temp(1,1:length(coastline(i,1).X)) = coastline(i,1).X;
ycoords_obs_temp(1,1:length(coastline(i,1).Y)) = coastline(i,1).Y;
plot(xcoords_obs_temp(1,:),ycoords_obs_temp(1,:),'k');
end

% Set location of WATER Obstacles
rivers = shaperead('vmap0_internalwaters.shp');
for i = 1:length(rivers)
xcoords_obs_temp(1,1:length(rivers(i,1).X)) = rivers(i,1).X;
ycoords_obs_temp(1,1:length(rivers(i,1).Y)) = rivers(i,1).Y;
plot(xcoords_obs_temp(1,:),ycoords_obs_temp(1,:),'Color',[0.2 0.4
0.9]);
end

% Set location of Population Obstacles
populations = shaperead('vmap0_populated.shp');
for i = 1:length(populations)
xcoords_obs_temp(1,1:length(populations(i,1).X)) =
populations(i,1).X;
ycoords_obs_temp(1,1:length(populations(i,1).Y)) =
populations(i,1).Y;
plot(xcoords_obs_temp(1,:),ycoords_obs_temp(1,:),'Color',[0.8 0.4
0.3]);
end

% Title and axis for the plot
title('Cluster Optimiser with Obstacle Avoidance - Germany Case',...
        'FontSize',16);
xlabel('Longitude','FontSize',14);
ylabel('Latitude','FontSize',14);
toc
```