Minimizing the oil spill damage by optimizing the locations of the cleaning vessels

Daniela Hanea

A thesis submitted to the Delft University of Technology
in conformity with the requirements for the degree
Master of Science

Delft, The Netherlands

Chairperson of Graduate Committee:	Prof. dr. ir. Arnold W. Heemink
Graduate Committee:	Prof. dr. ir. Arnold W. Heemink
	Prof. dr. Roger M. Cooke
	Dr. Peter Wilders
	Ir. dr. Joost Beckers

Table of Contents

Acknowledgements

The author wishes to express sincere appreciation to the Delft University of Technology, where she had the opportunity to work with Arnold Heemink, Roger Cooke, Peter Wilders, Dorota Kurowicka and Thomas Mazzuchi. Thanks to Joost Beckers from RIKZ for proposing the subject for this thesis, providing data and helpful explanations of the issues.

Last but not least, the author would like to acknowledge her husband for his permanent support.

List of Figures

Chapter 1

Introduction

North Sea

The North Sea is one of the busiest seas on the planet. Around $420,000$ vessels pass this area each year, not including fishing vessels. Many of these vessels carry products that pose a threat to the environment. In particular, oil and chemicals can cause serious pollution if spilled at sea. On the NCP (Netherlands Continental Plat, Dutch part of the Continental Shelf) an average of 24 incidents occur annually. These incidents involve collisions between ships, oil platforms, ship grounding, fires and lost cargo. Moreover, many pollutants are accidentally or deliberately spilled from ships during 'normal' operation. These spills, although small, account for the majority of the total pollution.

The Ministry of Transport, Public Works and Water Management (Verkeer en Waterstraat) is the most important department responsible for oil pollution response within the Netherlands. The North Sea is covered by Rijkwaterstraat (RWS), directie Noordzee (DNZ), which is responsible for the cleanup operations of oil, chemicals and lost cargo on the NCP.

In this study, response at sea is based on mathematical recovery. Chemical dispersants are often used as well, however they are prohibited in Dutch waters. Also, the usage of mechanical dispersion (by propellers) is not often used in Netherlands.

RWS operate a number of oil combating craft and have further crafts on contract for immediate adaptation to oil recovery vessels in the event of an emergency. This includes 5 commercial sand dredgers. A number of other vessels are also available to support recovery operation and vessels boom. The target recovery capacity of these combined units is $30,000m^3$ of oil in three days. These vessels have their basis primarily at Scheveningen and also at Zeeland and at Rotterdam.

Outlines of the thesis

This thesis is part of Rataplan project (*Responsive action* to *tanker* vessel incidents *planning*), whose aim is to optimize the responsive resources (vessels, planes and special equipment) against spill of the oil or chemicals. An optimal planning meets the demand both in terms of response time and capacity.

The goal of the thesis is to find a model for the optimization of capacity and positioning of the available cleaning vessels along the Dutch coast. The model should at least take into account the characteristics of the vessels (speed, cleaning capacity, preparation time), pollution contingency maps, ecological and economical damage, which increase in time if the oil spill is not removed and is directly related to the size of the oil spill.

Firstly, Chapter 1 presents some of the international conventions and agreements regarding the fight against pollution in North Sea. The possible methods that can be used to clean the sea in case of pollution are briefly presented. Chapter 2 contains the description of data about the vessels, harbors and incidents. The model that we developed to reach the goals of the research is presented in Chapter 3. In this chapter, after preliminary discussions about the way in which the vessels would undertake action in an emergency situation, mathematical formulas for the damage produced by an oil spill are derived. The last part of the chapter pays attention to the complexity of the optimization problem and to the difficulty of solving it with classical methods. In Chapter 4, the main tool to solve the optimization problem formulated in the previous chapter is presented,namely the genetic algorithm. Basic notions from the domain of genetic algorithm are given. In the last part of the chapter, a very simple example is given, to show how the genetic algorithm works. Chapter 5 presents the solutions of the optimization problem obtained using genetic algorithms, remarks regarding these solutions and comparison of the results with the existent position of vessels along the Dutch coast. The conclusions of the analysis and recommendations for future research are presented in Chapter 6. Appendix A contains the MatLab codes used to obtain the results from Chapter 5.

Chapter 2

Regulations and methods for the recovery of the floating oil

The North Sea ecosystem suffers from various forms of contamination and pressure resulting from human activities. Protecting and constantly improving the quality of the ecosystem is important and a number of cooperation and consultation mechanisms are applied in this area.

In this chapter, the main international regulations and conventions regarding the pollution in the North Sea are enumerated. The usual methods and equipments used in the recovery of the floating oil are briefly presented in the last section of the chapter.

2.1 Conventions and regulations

When you look at the sea chart, you realize that nothing can be done without real international cooperation.

One first agreement regarding the pollution in the North Sea was signed at Bonn, in June 1969, by Belgium, Denmark, France, Germany, Netherlands, Norway, Sweden, UK and Northern Ireland. The objective of this agreement was to ensure cooperation between the coastal States in providing manpower, supplies, equipment and scientific advice at short notice to deal with discharges of oil or other noxious or hazardous substances in the North Sea. The convention also divided the area of the North Sea into national zones; for Netherlands, the zone was limited by lines joining the following points: $(51°32'N, 3°18'E)$, $(51°32'N, 2°6'E)$, $(52°30'N, 3°10'E)$, $(54°N, 2°40'E)$, $(54°N, 5°30'E)$, $(53°34'N, 6°38'E)$.

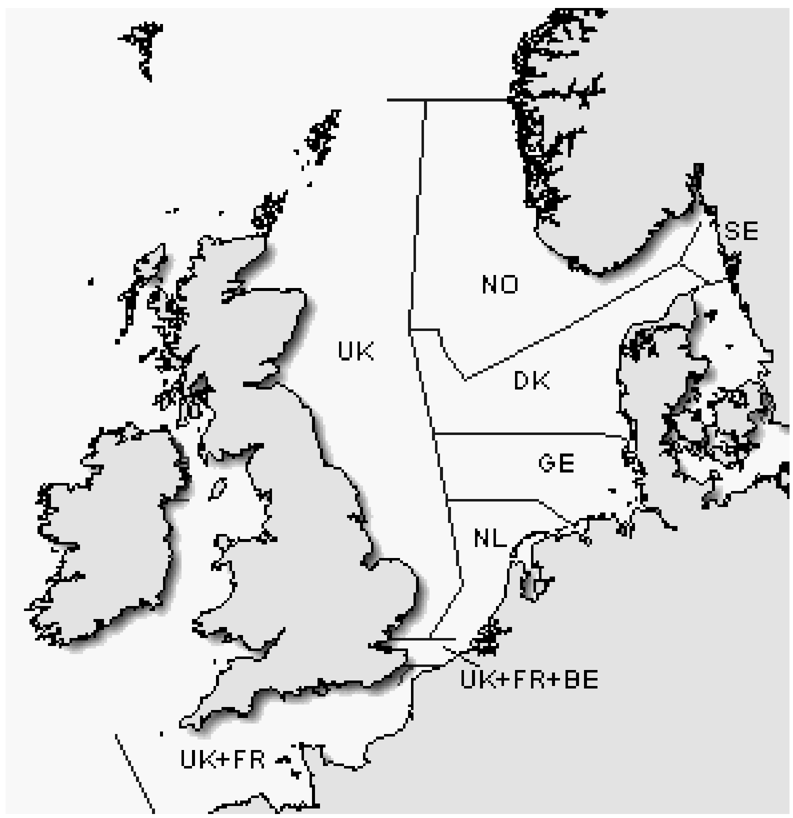


Figure 2.1: North Sea map

The agreement was replaced by the *Agreement for Cooperation in Dealing with the Pollution on the North Sea by Oil and Other Harmful Substances* (Bonn, 13 September 1983), which entered into force on 1 September 1989. This new agreement has mainly the same contents as the previous one, but a few more articles regarding the financial aspects of an international cleanup operation in the North Sea are added.

Another international instrument that stabilizes the norms and presents the research progress in the area of protection against pollution is the system of *International Conferences on the Protection of the North Sea* (Bremen, 1984; London, 1987; The Hague, 1990; Esbjerg, 1995; Norway, 2002), which constitute political forum which adopt far-reaching politically-based commitments. They provide a political framework for a broad and comprehensive assessment of the measures needed to protect the North Sea. Apart from the agreements on action to protect the North Sea, the North Sea Conferences have also played an important role in influencing environmental management decisions in a much wider context.

2.2 Guidelines and recommendations

There is a variety of available resources to fight against pollution at sea. In this paragraph, part of them (as floating booms, skimmers) are briefly described. The resources are adapted to recover oil on the open sea, in coastal waters and from water's edge.

In the event of oil pollution, the use of dispersing chemicals may also be considered. This can be done if it provides that the chemical treatment brings about an extra overall reduction in the adverse effects of the pollution on the marine environment, in comparison with the natural dispersion process and other methods of fighting pollution. The institution that is dealing with the pollution and the response at sea has to authorize the use and spreading of chemicals in the marine environment to combat pollution. This authorization depends on the area where the pollutant is located, on the type of oil and also on the period of year. In the Netherlands, the use of the dispersing chemicals is not yet allowed.

The oil boom is a floating barrier which is used in cleaning up oil on the surface of the water. The booms can be used: to contain oil (containment booming); to collect oil; as a barricade to exclude oil (exclusion booming) from a certain area; to absorb oil; and to deflect oil (diversion booming).

Containment booming is the process of preventing the spread of oil spill by confining the oil to the area in which it has been discharged.

Booms can also be used to divert pollutant to areas where cleanup operation can be conducted and to protect specific areas such as entrance to harbors and rivers or environmentally or economically sensitive areas.

The use of booms to concentrate the floating oil prior to its recovery by specialized skimmers is often seen as the first stage of an effective response to an oil spill.

The most important characteristic of a boom is its behavior in relation with water movement. It should be flexible conform to water motion, but in the same time, sufficiently rigid to retain as much oil as possible. It should tolerate inexpert handling, since trained personnel are not always available. Towing booms at sea, for example in U or J configurations, is a difficult task requiring specialized vessels.

Unfortunately, this approach presents a number of disadvantages. Even if the booms are operational within few hours, in rough seas and low viscosity oil case, the spill can be scattered over many square kilometers. And it will not be feasible for booms to encounter more than a fraction of a widely spread slick. This is the reason why the limitations that poor weather and rough seas impose on operations at sea are seldom fully appreciated.

Because of the difficulties of operation multi-ship towed boom systems, specialized ships have been build which incorporate sweeping arms, skimming devices and on board oil storage. In addition a pump or vacuum device is necessary to transfer recovered oil and water to store. A skimmer is a mechanical recovery device designed to remove the oil from the water's surface without causing major alterations in the physical or chemical properties. Because skimmers float on the water surface, they experience many of the operational difficulties which apply to booms, particularly those posed by wind, waves and currents. Even moderate wave motion greatly reduces the effectiveness of most skimmer designs. In calm waters more satisfactory performance can be achieved.

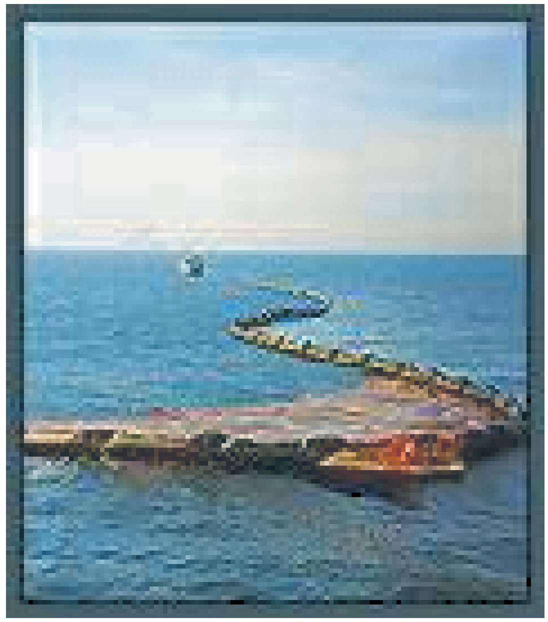


Figure 2.2: Boom

It is important to have adequate temporary oil storage facilities available, otherwise this becomes a bottleneck to successful oil recovery. Temporary storage needs to be easy to handle and easy to empty so that it can be used repeatedly. Suitable units include barges and portable tanks which can be set up on available vessels.

Many factors should be considered when selecting skimmers. The intended use and expected operational conditions should first be identified before criteria such as size, robustness and case of operation, handling and maintenance can be weighted up. The most important factors to consider are the viscosity and adhesive properties over time. Only sometimes the properties of the oil are known; that is why it is preferable to select units which can deal with a range of oils. It is also important to recognize the difficulties posed by floating debris, both natural (sea weeds, sea grasses, trees and branches) and man made (plastic, glass, timber).

Because of the various constraints imposed on skimmers in the field, their design capacities are rarely realized. Experience from numerous spills has consistently shown that recovery rates reported under test conditions cannot be sustained during a spill and so it is important not to have unrealistic expectations about what can be achieved.

Chapter 3

Description of data

The analysis in this thesis is based on the data provided by RWS-DNZ. The information is about the incident probabilities and the size of the oil spill, about the harbors and the available cleaning vessels. The data is briefly presented in the next sections.

3.1 Incident data

The Dutch part of the Continental Shelf is covered by a grid with cells that have the size approximatively 8×8 km. Each cell is characterized by its geographical coordinates (latitude and longitude).

The Samson program was used to compute the probabilities of incidents with vessels and the outflow of oil based on incident statistics from Lloyd's, a large commercial insurance company, which has also insurances against oil pollution from ships. Samson is a Safety Assessment Model for Shipping and Offshore North Sea, which has three models: shipping model (where and how many accidents take place, how much spill is there and which factors are influencing this); offshore mining model (calamities on drilling of oil); and oil spreading effect model (describes the transport of oil as a consequence of a calamity spill on the North Sea).

The data that we use in this thesis comes from the shipping model. This model computes the probability of pollution in a cell starting from the traffic intensity for that cell, taking account of the probability of collision, the probability that the tanker hull is damaged given a collision, the density function for the volume of oil in a tanker.

The oil spills are divided in eight categories and for each category, separately, the probability to have a pollution per year with a volume of oil from that category is given. The categories and the mean volume oil in each of them are presented in table (3.1). The data shows that the probability of large spills is small. Very small spills (the first two categories) are also less frequent because only serious accidents are considered in the calculations. The most frequently spills are those from categories four and five, so with the volume of oil between 750 and 10000 m^3.

	Category	Frequency (per year)	Mean spill size (m^3)
1	$0.01 < V < 20$	0.001653	12.3
2	$20 < V < 150$	0.003442	76.2
3	$150 < V < 750$	0.015155	424.1
4	$750 < V < 3000$	0.024028	1736.3
5	$3000 < V < 10000$	0.024399	5917.5
6	$10000 < V < 30000$	0.016209	16962.5
7	$30000 < V < 100000$	0.011702	57920.7
8	$100000 < V < 999999$	0.006835	186605.0

Table 3.1: Spill categories

3.2 Harbors

There are six harbors along the Dutch Coast where the cleaning vessels may be located. For these ports, the geographical coordinates are known (see table (3.2)).

Figure (3.1) shows how these harbors are scattered along the Dutch Coast. Each harbor can lodge any of the available vessels. Also, there is no limit on the number

Name	Longitude (N)	Latitude (E)
Scheveningen (1)	$52°6'7''$	$4°15'58''$
Terschelling (2)	$53°21'50''$	$5°13'18''$
Rotterdam (3)	$51°53'41''$	$4°5'54''$
IJmuiden (4)	$52°27'41''$	$4°35'27''$
Terneuzen (5)	$51°20'11''$	$3°48'58''$
Den Helder (6)	$52°57'39''$	$4°46'53''$

Table 3.2: Port coordinates

of vessels that can be located in one port.

3.3 Vessels

There are 14 vessels that can participate in an emergency situation at the North Sea. Four of them (Arca, Terschelling, Rijndelta and Hein) are owned by RWS or under contract. In an emergency situation, RWS can count on them any time. There are six more vessels (Cornelia, Gateway, Gepotes, HAM 316, Lesse, Waker) that are not under contract, but can be asked to help in case of an incident. The last four vessels (Albatros, Adelaar, Zandexpress 3, Zanderexpress 4) have an unclear status; in a future research it would be interesting to introduce them into analysis.

Each vessel is characterized by speed, cleaning capacity (km^2/hr), response time (during the weekend or during the rest of the week). Another important characteristic of vessels is the location of sweeping arms. If the arms are on board, then the ship can start sailing to the polluted area right away. If the arms are in other harbor, the ship needs to sail to this harbor, to instal the arms and then to go to the polluted area. For these vessels, the sailing time between their harbor and the harbor where the arms are located has to be taken into consideration.

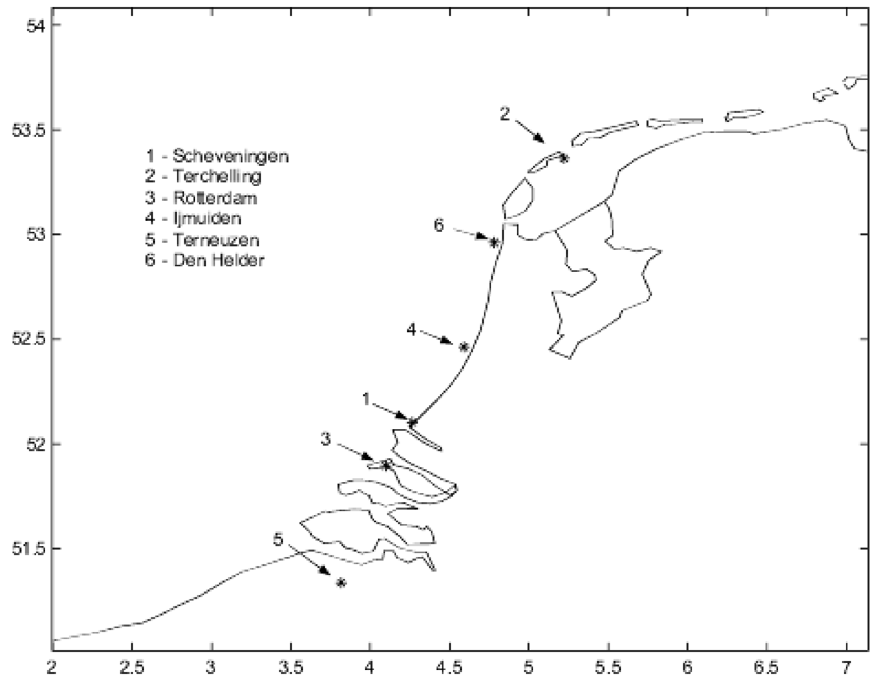


Figure 3.1: Harbors

All these data are presented in Table (3.3).

The present location of vessels can be used to compare the result of the study with the actual situation.

Ship	**Actual location**	**Speed** (kn/hour)	**Clean. cap.** ($km^2/hour$)	**Mob. time** ($hours$)	**Arms location**
Arca	1	14	0.12	1	on board
Terschelling	2	12	0.06	1	2
Rijndelta	3	14	0.12	1	on board
Hein	3	13	0.12	4	3
Cornelia	4	14	0.12	12	3
Gateway	4	14	0.12	12	3
Geopotes	3	14	0.12	10	3
HAM 316	3	14	0.12	10	3
Lesse	5	12	0.06	8	5
Waker	6	13	0.06	1	on board

Table 3.3: Vessel characteristics

All the data that we have described will be used in this study. It will be proved that the data serve well the goals of the project. An analysis of each of the eight categories of oil spills will be done, taking into consideration either the first group of vessels (those vessels that are owned by RWS and can work independently), or the first two groups of vessels (adding the vessels that are not under contract, but can help in case of emergency).

The results of this analysis will be presented in chapter 5, after a model for our problem and a method to solve it has been established.

Chapter 4

Model for the optimization problem

To quantify the effect produced by an oil spills is a very important issue that has to be clarified in order to optimize the location of the cleaning vessels. It is known that there are two types of damages: ecological damage and economical damage.

Perhaps the most dramatic symbol of the consequences of an oil spill is an oiled seabird. Diving birds are especially vulnerable. Some marine mammals may also be affected. The impact of spilled oil on the nearshore marine waters and shallow fishing banks is obvious and often severe.

In addition to ecological concerns, coastal regions can suffer economically from damage done by oil spill to recreation areas, harbors and vessels, commercial shellfish grounds. During the summer months, beaches along the coasts of most maritime countries are crowded with people on weekend outings and vacations. Thus, there is a big interest in protecting beaches from spill or to clean them up quickly.

It is very difficult to put together the ecosystem loss and the economical loss. Economical effects can be transformed more or less in money, but for the bird and animal lives it is a very hard work to do this. If one is interested only in the economical damage, the amount of money lost from the closure of the beaches or harbors and the money used to clean an oil spill are a good measure for the damage.

But, if the interest is to combine both types of damage as in the most of the cases, it is very difficult to consider money as a unit for the damage and other measures have to be found.

We have to take into account that the wildlife oiled and killed are function of area swept by surface oil and vulnerability. Mortality of fish, shellfish and their eggs and larvae are function of concentration and time exposure. Moreover, the economical damage is also a function of time and volume of oil.

All these reasons and the lack of more detailed information regarding the ecosystem and the costs of the economical negative effects of a pollution lead to a definition directed related to the time of exposure and the surface of the oil spill.

In this thesis, we define the damage caused by an oil spill as the total surface that is covered with oil, integrated over time. With this definition of the damage, the effectiveness of the cleanup operation is optimum when the oil is cleaned in as short time as possible.

Using this definition, in the next sections of the chapter, a mathematical formula will be derived for the damage.

First, some preliminary remarks are presented. The first paragraph contains all the questions that arise in the the beginning phase of the work. The assumptions that we make are also included in this paragraph. The next section establishes the notations and gives the general description of the model. In section three, mathematical formulas for damage are derived. The last paragraph summarizes the problem that we are dealing with and gives an idea about the methods that can be used to solve it.

4.1 Preliminary remarks and assumptions

This work is directed to minimize the damage produced by oil pollution in the Dutch part of the North Sea and to find the optimum location of the available cleaning vessels. The damage was chosen as the measure of the effectiveness of the cleaning operation. There are other possibilities, as maximize the volume of the oil that is cleaned in an operation or minimize the cost. For both these possibilities there are other aspects that have to be included in the model (type of oil, the cost of the ecological damage and of the economical damage) and more detailed data are needed.

The first question that arises is, given a certain position of vessels, which vessel(s) should be used to clean up a certain polluted area? One idea is to send that vessel for which, at the end of cleaning operation, the damage would be minimum. Therefore, to compute the damage in a cell, one has to consider each vessel and to see what would be the damage if that vessel is used; then the value of the damage in that certain cell is equal to the minimum value that has been found.

It is easy to see that if more vessels are used, the cleaning operation time is less, so the damage caused by the pollutant is less. This is the case in almost all pollution accidents. So, an intervention with more vessels in an emergency case is more realistic.

In this case, one more question arises: how many vessels must be used and which of them? It seems that the minimum time for a cleaning operation (so, probably, the minimum damage) is obtained if all vessels are used. This implies on one hand a large cleaning cost and on the other hand, no available vessel for other possible oil spills close in time to the previous one. But in the case of very serious accidents (large quantity of oil in a vulnerable area), the cleaning costs are not important if there is a considerable reduction in the damage, or, if, for example, the cleaning operation is finished in an acceptable interval of time.

Hence, several cases may be taken into consideration:

- an intervention with only one vessel (that one which gives minimum damage at the end of cleaning operation);
- an intervention with all vessels;
- an intervention with a combination of vessels (that combination that gives minimum damage at the end of cleaning operation).

The last case will be considered in the rest of the study as the basic case.

It has to be mentioned that in the case with more vessels, the total capacity of the system is the sum of cleaning capacities of the involved vessels. No other additional tools, sych as booms between two vessels, are used to increase the cleaning capacity.

There is one more aspect that has to be clarified. If there are more vessels involved in a cleaning operation, how they should undertake action? Should they wait for each other, or as soon as one arrives at the polluted area, it starts the cleaning? From the calculation point of view, the first situation is more convenient (as we will see in the next sections), but from the realistic point of view, the second situation should be considered. For our study, we will assume that the vessels have to wait for the slowest one and to start the cleanup operation in the same time.

There are few more assumptions that we have to mention. For this study it is assumed that there is no weather influence; the wind, the waves and the currents don't influence the movement of the oil spill. Also, bad weather conditions do not delay the moment when the cleaning up operation starts.

4.2 Notations and definitions. General presentation of the model

Before starting to build the model, some notations have to be considered:

- n - the number of vessels;
- m - the number of harbors (ports);
- N - the number of cells;
- $v \in \{1, 2, \ldots, m\}^n$ - the position vector, where $v(j)$ is the port in which the vessel j is located;
- p_c - the probability to have pollution in cell c.

Using these notations and the previous assumptions, the goal is to find that (or those) location vector(s) that minimizes the expected damage, or, mathematically, to find:

$$argmin_v \sum_{c=1}^{N} p_c D(c, v) \tag{4.2.1}$$

where $D(c, v)$ is the damage in cell c if the vector location is v (which will be explained in more details in the next section).

Hence, first of all, a way to compute the damage $D(c, v)$ as a function of the cell c and the position vector v has to be found. As it was said before, all possible combinations of vessels will be considered and the damage will be calculated for each of these combinations (one way to do this will be presented in the next sections). Then, the damage $D(c, v)$ will be equal to the minimum value which has been found.

For example, if the total number of vessels is three, the decision has to be taken between any of the three vessels, any of the combinations of two vessels (three more cases) and the case in which all vessels are used; so, in total, there are seven cases.

It has to be noted that for a system with n vessels, given a location vector, the number of cases that have to be analyzed for each cell is:

$$\binom{\mathrm{n}}{1} + \binom{\mathrm{n}}{2} + \ldots + \binom{\mathrm{n}}{\mathrm{n}} = 2^n - 1$$

In the next section, a formula for damage function $D(c, v)$ is derived.

4.3 Damage

At the first side, the damage seems to depend on time and volume of pollutant. The longer a quantity of oil exists in the water, the larger the damage; moreover, larger a quantity of oil, larger the loss. The way in which the damage is related to the time and volume of pollutant is via the area of the oil spill. A larger area of oil spill produces more bird deaths, for example.

The formula which relates the damage and the area of the oil spill that we have chosen is:

$$D(t) = \int_0^t A(s)ds \tag{4.3.1}$$

where $D(t)$ is the damage up to time t and $A(s)$ is the area of the oil spill at moment s.

We have to remark that in case that no cleanup action is undertaken (so the area of the oil spill will never decrease), when t goes to infinity, the damage will grow to infinity, which is an unrealistic case. Hence, this formula is not true for a very long time t.

Hence, the first step is to consider a model for the area of the oil spill of a certain fixed volume V. A popular model for this is Mackay's model (Canadian Journal of Chemical Engineering, Vol. 51, p.434, 1980).

$$\frac{dA}{dt} = K_1 \cdot V^{4/3} \cdot \frac{1}{A} \tag{4.3.2}$$

where:

- A - area (m^2);
- V - volume (m^3);
- K_1 - constant $= 150\ (sec^{-1}) = 5.4 \cdot 10^5\ (hour^{-1})$.

Since at moment $t = 0$ it is supposed that there is no oil spill, the first idea is to consider the initial condition $A(0) = 0$. But, if we look at the equation (4.3.2), it doesn't allow the area of the oil spill to be equal to zero. This inconvenience can be eliminated by assuming that the area of the oil spill approach zero when t goes to zero, or:

$$\lim_{t \longrightarrow 0} A(t) = 0$$

With this condition, the solution of the differential equation (4.3.2) is:

$$A(t) = \sqrt{2K_1} \cdot V^{2/3}\sqrt{t}\ ,\ t \geq 0 \tag{4.3.3}$$

We have to remark that the volume of oil V is considered constant. Of course that a more realistic approach would be to consider that the release is not instantaneous, so the volume is not constant, but it increases with a certain rate (for example $V(t) = r{\cdot}t$, where t is the time and r is the rate at which the oil comes into the water). Hence, this approach has to take account of the source of pollution and this needs more data. Moreover, it is very difficult to find the rate r, for example, in a pollution produced by the collision of two vessels, or by the grounding of a vessel.

We have to remark that this formula holds when no cleaning up operation is involved. After a cleaning vessel reaches the polluted area, the cleaning capacity $q(km^2/hour)$ of that vessel has to be taken into consideration. There are two ways to do this.

First, we can include the cleaning capacity into the differential equation, as follows:

$$\frac{dA}{dt} = K_1 \cdot V^{4/3} \cdot \frac{1}{A} - q\ ,\ \text{for } t \geq T_{ar} \tag{4.3.4}$$

where T_{ar} is the arrival time of the cleaning vessel. Solving this equation with the initial condition $A(T_{ar}) = A_0$ known ($A_0 = \sqrt{2K_1} \cdot V^{2/3}\sqrt{T_{ar}}$), we obtain:

$$A(T_{ar}) - A(t) - \frac{K_1 V^{4/3}}{q} \ln\left|\frac{qA(t) - K_1 V^{4/3}}{qA(T_{ar}) - K_1 V^{4/3}}\right| = q(t - T_{ar}) \tag{4.3.5}$$

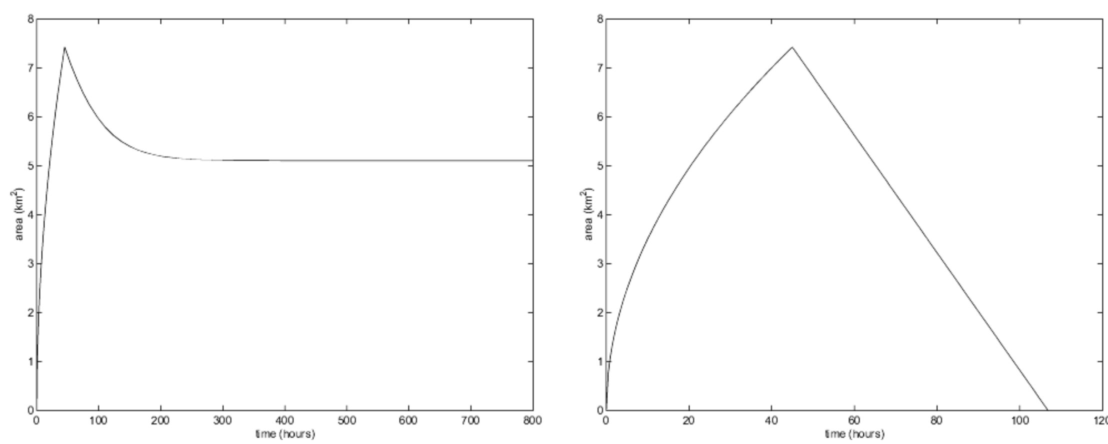
A new remark has to be made. In this case, the formula for the area $A(t)$ holds the same condition $A(t) \neq 0$. More then this, if we look at the graph of the curve $A(t)$ for a very large interval (even unrealistic) of time: $[0\ ,\ 800]$ hours, we can see that the function $A(t)$ doesn't approach the value zero. This fact has a big disadvantage because it seams to be no end for the cleaning operation, so a stopping time, T_{stop}, usually derived from the condition $A(T_{stop}) = 0$, cannot be computed. Even if we assume that the cleaning action stops when the area of the oil spill is below a certain level, let's say ε, with $\varepsilon > 0$, the stopping time cannot be computed. As we can see in figure (4.3,a), this ε should be quite large, which doesn't seem to be a realistic assumption.

Another way is to assume that the area of the oil spill increases until the vessel arrives, stops growing and starts to decrease due to the intervention of the cleaning vessels. This is a realistic approach since the dispersion of the oil spill stops after a certain time. Hence, in this case the equation for the area of the oil spill would be:

$$A(t) = \sqrt{2K_1} \cdot V^{2/3}\sqrt{T_{ar}} - q(t - T_{ar}) \text{ , for } t \geq T_{ar} \tag{4.3.6}$$

It is clear that in the second case, for $t \geq T_{ar}$, the area is linear decreasing in time. Moreover, the stopping time T_{stop} can be computed from the condition $A(T_{stop}) = 0$ and the time t has to be less then the stopping time T_{stop} for the area $A(t)$ having positive values.

If the condition $A(T_{stop}) = 0$ for the end of the cleaning operation seems to be too drastic, we can replace it by $A(T_{stop}) = \varepsilon$, with $\varepsilon > 0$ small enough.



(a) (b)

Figure 4.1: Area of the oil spill (with formula (4.3.5) (a), with formula (4.3.6) (b))

Even both equations (4.3.5) and (4.3.6) are valid, the way in which the plots of the two curves $A(t)$ look (see figure (4.3)) and the condition for the end of the cleaning operation are very good reasons to take the decision to use equation (4.3.6) for the rest of this study. Moreover, in this equation it is assumed that the area of the oil spill increases only until the arrival moment of the cleaning vessel. This is close to the reality since it is known that a spill of oil has a finite spreading time (very probably not exactly the arrival time of the vessel, but for sure not until the "end" of the cleaning operation).

Thus, for the rest of the analysis, the area of the oil spill is considered to be modelled by:

$$A(t) = \begin{cases} \sqrt{2K_1}V^{2/3}\sqrt{t} & \text{for } 0 \leq t \leq T_{ar} \\ \sqrt{2K_1}V^{2/3}\sqrt{T_{ar}} - q(t - T_{ar}) & \text{for } T_{ar} \leq t \leq T_{stop} \end{cases} \tag{4.3.7}$$

with:

$$A(T_{stop}) = 0 \Leftrightarrow T_{stop} = T_{ar} + \frac{1}{q}\sqrt{2K_1} \cdot V^{2/3}\sqrt{T_{ar}} \tag{4.3.8}$$

Going back to the formula of the damage, we are interested in the damage at the end of the cleaning operation, so at moment T_{stop}; hence, we have to compute:

$$\begin{aligned} D(T_{stop}) &= \int_0^{T_{stop}} A(s)ds = \\ &= \int_0^{T_{ar}} A(s)ds + \int_{T_{ar}}^{T_{stop}} A(s)ds = \end{aligned} \tag{4.3.9}$$

which gives the result:

$$D(T_{stop}) = \sqrt{2K_1} \cdot V^{2/3} T_{stop}\sqrt{T_{ar}} - \frac{1}{3}\sqrt{2K_1} \cdot V^{2/3} T_{ar}\sqrt{T_{ar}} - \frac{1}{2}q(T_{stop} - T_{ar})^2 \tag{4.3.10}$$

If we use the formula (4.3.8) for the stopping time T_{stop}, the result would be:

$$D(T_{stop}) = \frac{2}{3}\sqrt{2K_1} \cdot T_{ar}\sqrt{T_{ar}} + \frac{1}{q}K_1 V^{4/3} T_{ar} \tag{4.3.11}$$

It is important to remark that the computation of damage $D(T_{stop})$ has to be done for each cell c, and even the cell c doesn't appear directly in the formula for damage, it influences the result through the arrival time T_{ar}, which is equal to the ratio of the distance between the cell c and the location of the vessel and the speed of the vessel.

One more remark is that, for the calculation of damage, we have assumed that only one vessel is used in the cleaning operation. Indeed, these relations are true if only one vessel is used. For a case with more vessels, if we assume that they wait for the slowest vessel to start the cleaning, the formulas are valid as well. Now T_{ar} (the arrival time) is the time when the last vessel arrives at the polluted area and the cleaning capacity q is the sum of all cleaning capacities of vessels involved in cleanup operation. In the more realistic case, in which each vessel starts to clean just after it arrives at the oil spill location, the formula will change.

To give an idea how the formula would change, let us consider first the following example:

Let us suppose that we want to compute the damage in cell c if two vessels are used, namely vessel i and j, which are located in ports $v(i)$ and $v(j)$. For this simple example, let us assume that the vessels are characterized only by speed and cleaning capacity (so, we don't take into consideration mobilization time and the sailing time to the arms' location). If we denote by $vess$ the matrix of vessel characteristics, then $vess$ will be a matrix with n lines (one for each vessel) and two columns (one for each characteristic: speed and cleaning capacity) and:

- $vess(i, 1)$ is the speed of vessel i;
- $vess(i, 2)$ is the cleaning capacity of vessel i.

Also, we denote by $dist$ the matrix of distances between cells and ports; so $dist$ is a matrix with N lines and m columns and $dist(c, k) =$ the distance between cell c and port k.

Coming back to our simple example, we have to check which vessel arrives first and when; for this, we have to calculate the arrival times:

$$t_i = \frac{dist(c, v(i))}{vess(i,1)}$$

and

$$t_j = \frac{dist(c, v(j))}{vess(j,1)}$$

and let assume that $t_i \leq t_j$ (vessel i arrives first). Then:

$$A(t) = \begin{cases} \sqrt{2K_1} \cdot V^{2/3}\sqrt{t} & \text{for } 0 \leq t < t_i \\ \sqrt{2K_1} \cdot V^{2/3}\sqrt{t_i} - q_i(t - t_i) & \text{for } t_i \leq t < t_j \\ \sqrt{2K_1} \cdot V^{2/3}\sqrt{t_i} - q_i(t - t_i) - q_j(t - t_j) & \text{for } t_j \leq t \leq T_{stop} \end{cases} \tag{4.3.12}$$

where:

$q_i \quad = vess(i,2)$ is the cleaning capacity of vessel i;

$q_j \quad = vess(j,2)$ is the cleaning capacity of vessel j.

and the stopping time is obtained from the condition:

$$\begin{aligned} & A(T_{stop}) = 0 \\ \Leftrightarrow \quad & \sqrt{2K_1} \cdot V^{2/3}\sqrt{t_i} - q_i(T_{stop} - t_i) - q_j(T_{stop} - t_j) = 0 \\ \Leftrightarrow \quad & T_{stop} = \frac{1}{q_i + q_j}\left(\sqrt{2K_1} \cdot V^{2/3}\sqrt{t_i} + q_i t_i + q_j t_j\right) \end{aligned} \tag{4.3.13}$$

It can be seen that in equation (4.3.12), we keep the assumption that the oil spreading stops at the moment when the first vessel arrives at the polluted area.

Now, the damage at the end of cleaning operation is:

$$\begin{aligned} & D(T_{stop}) = \int_0^{T_{stop}} A(s)ds = \\ = \quad & \int_0^{t_i} A(s)ds + \int_{t_i}^{t_j} A(s)ds + \int_{t_j}^{T_{stop}} A(s)ds = \\ = \quad & \sqrt{2K_1} \cdot V^{2/3}T_{stop}\sqrt{t_i} - \frac{1}{3}\sqrt{2K_1} \cdot V^{2/3}t_i\sqrt{t_i} - \frac{1}{2}q_i(T_{stop} - t_i)^2 - \frac{1}{2}q_j(T_{stop} - t_j)^2 \end{aligned}$$

It is easy to see that the formula (4.3.10) is a particular case of this formula, for $q_i = q$, $t_i = T_{ar}$, $q_j = 0$ and $t_j = 0$.

If we want to compute the damage at the end of the cleaning operation if all vessels are used, we obtain:

$$D(T_{stop}) = \sqrt{2K_1} \cdot V^{2/3} T_{stop} \sqrt{t_{min}} - \frac{1}{3}\sqrt{2K_1} \cdot V^{2/3} t_{min} \sqrt{t_{min}} - \frac{1}{2}\sum_{i=1}^{n} q_i (T_{stop} - t_i)^2 \tag{4.3.14}$$

where:

$t_{min} =$ the minimum arrival time (the time when the first vessel arrives)

and $T_{stop} = \frac{1}{\sum_{i=1}^{n} q_i}\left(\sqrt{2K_1} \cdot V^{2/3}\sqrt{t_{min}} + \sum_{i=1}^{n} q_i t_i\right)$

4.4 Solving the optimization problem

We have to remind that our problem is to find the optimum location vector v which makes the expected damage to be a minimum. In the previous section we found a way to compute the damage produced by an oil spill in a certain cell c, given a location vector v. We saw that this value of the damage is also a minimum value, namely the minimum of all possible values of the damage, depending what combinations of vessels is used.

Since we are dealing with an optimization problem, the first step is to make a review of the methods that can be used to solve this problem.

One classical way to solve an optimization problem is to use the gradient method. There are two reasons to exclude this method: first, the set of the feasible solutions is a discrete one and secondly, the objective function does not have the property to be differentiable.

Another way to solve an optimization problem, which can be applied even for discrete set of feasible solutions, is to use the linear programming. This method has also to be excluded, since the objective function is not linear. Moreover, it is like a 'black box', i.e. there is an input, there are some operations and processes (minimum of all combinations of vessels that can be used) and there is an output.

The very large size m^n of the set of the feasible solutions of this problem makes the enumeration of all values of the objective function impossible.

These are the reasons that lead to a search method which permits the exploration of a large part of the set of the feasible solutions. In the next chapter, a heuristic method, called genetic algorithms, will be described and then applied for our problem.

Chapter 5

Genetic Algorithms

5.1 Introduction

Genetic algorithms (GAs) are search techniques based on the mechanism of natural selection and natural genetics. Differing from the conventional search techniques, GA's start with an initial set of random solutions, called *population*. Each individual in the population is called a *chromosome*, representing a possible solution to the problem. A chromosome is a string of symbols, usually, but not necessarily, a binary string. The chromosomes evolve through successive iterations, called *generations*. During each generation, the chromosomes are evaluated using a measure of fitness.

To create the next generation, new chromosomes, called *offspring*, are formed by either merging two chromosomes from current generation, using crossover operator, or modifying a chromosome using mutation operator. A new generation is found by selecting, according to the fitness values, some of the parents and offspring and rejecting other so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected.

After several generations, the algorithm converges to the best chromosome, which hopefully represents the optimum or suboptimum solution to the problem.

The most common method to provide an initial population is to randomly generate solutions. However, since the genetic algorithms can iteratively improve existing solutions (i.e. solutions from other heuristic methods), the beginning population can be seeded with potentially good solutions, with the remainder of the population being randomly generated solutions.

The genetic algorithms moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. Another termination strategy involves population convergence criteria. In general, the genetic algorithms will force much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than some specified level, the algorithm can be terminated. The algorithms can be also terminated due to a lack of improvement in the best solution over a specified number of generations.

Evaluation functions of many forms can be used in a genetic algorithm, subject to the minimal requirement that the function can map the population into a partially ordered set; the evaluation function (or fitness function) is independent of the genetic algorithm. In general, for an unconstrained optimization problem, the fitness function is the same as the objective function (maybe with some restrictions). If there are constraints, these can be included in the fitness function as well.

Let $P(t)$ and $C(t)$ be parents and offspring in the current generation, respectively; the general structure of the genetic algorithm is described as follows:

$t \leftarrow 0$
initialize $P(t)$
evaluate $P(t)$
while (not termination condition) do

recombine $P(t)$ to yield $C(t)$
evaluate $C(t)$
select $P(t+1)$ from $P(t)$ and $C(t)$
$t \leftarrow t+1$

end

Recombination typically involves crossover and mutation to yield offspring. In fact, there are only two kinds of operations in genetic algorithms:

1. Genetic operations: crossover and mutation.

2. Evaluation operation: selection.

Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining both chromosome features. A simple way to achieve crossover would be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point. This methods works well with the binary string representation. The performance of genetic algorithms depends, to a great extend , on the choice of the crossover operator used.

The *crossover rate* (denoted $probXover$) is defined as the ratio of the number of offspring produced in each generation to the population size (usually denoted pop_size). This ratio controls the expected number $probXover \times pop\_size$ of chromosomes to undergo the crossover operation. A higher crossover rate allows exploration of more of the solution space and reduces the chance of setting to a local optimum; but if this rate is too high, it results in a waste of a lot of computation time in exploring unpromising regions of the solution space.

Mutation is a background operator which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes. In genetic algorithms, mutation serves the crucial role of either (a) replacing the genes lost from the population during the selection process so that they can be tried in a new context or (b) providing the genes that were not present in the initial population.

The *mutation rate* (denoted $probMutate$) is defined as the percentage of the total number of genes in the population that are changed. The mutation rate controls the rate at which new genes are introduced into the population for trial. If it is too low, many genes that would have been useful are never tried out; but if it is too high, there will be much random perturbation, the offspring will start losing their resemblance to the parents, and the algorithm will lose the ability to learn from the history of the search.

For both crossover rate and mutation rate, as well as for other parameters that are used in genetic algorithms and that will be introduced later there is no optimum value, depending on the problem for which the algorithms are applied. Several choices have to be tested and based on the behavior of the genetic algorithms (especially their convergence), the best combination of values for all parameters has to be chosen.

Genetic algorithms differ from conventional optimization and search procedures (for example, the gradient method) in several fundamental ways:

1. GAs work with a (binary) coding of the solution set, not the solutions themselves.
2. GAs search from a population of solutions, not a single solution.
3. GAs use payoff information (fitness function), not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

Generally, the algorithm for solving optimization problems is a sequence of computational steps which asymptotically converge to an optimal solution. Most classical optimization methods generate a deterministic sequence of computation based on the gradient or higher-order derivatives of the objective function. The methods are applied to a single point in the search space. The point is then improved along the deepest descending/ascending direction gradually through iterations. This point-to-point approach has the danger of falling in a local optimum. Genetic algorithms perform a multiple directional search by maintaining a population of potential solutions. The population-to-population approach attempts to make the search escape from local optimum. The population undergoes a simulated evolution. At each generation the relatively good solutions are reproduced, while the relatively bad solutions die out. Genetic algorithms use probabilistic transition rules to select individuals to be reproduced and individuals to die so as to guide their search towards regions of search space with likely improvement.

Genetic algorithms have received considerable attention regarding their potential as a novel optimization technique. There are three major advantages when applying genetic algorithms to optimization problems:

1. GAs do not have much restrictions about the optimization problems. Due to their evolutionary nature, GAs will search for solutions without taking into account how the objective function works precisely inside. GAs can handle any kind of objective function (even those that work as a "black box") and any kind of constraints (i.e. linear or nonlinear) defined on discrete, continuous or mixed search spaces.

2. The ergodicity of evolution operators makes genetic algorithms very effective at performing a global search. The traditional approaches perform a local search using a convergent step wise procedure, in which the values of nearby points are compared and moves are undertaken towards better optimal points. Global optima are guaranteed to be found only if the problem possesses certain convexity properties that essentially guarantee that a local optimum is a global optimum.

How to encode a solution of the problem into a chromosome is a key issue for the genetic algorithms. In Holland's work, encoding is carried out using binary strings. For many GA applications, especially for the problems from industrial engineering, the simple GA was difficult to apply directly because the binary string is not a natural coding. Various non string encoding techniques have been created for particular problems - for example, real number coding for constrained optimization problems and integer coding for combinatorial optimization problems. Choosing an appropriate representation of candidate solutions to the problem at hand is the foundation for applying genetic algorithms to solve real world problems. For each application case, it is necessary to perform analysis carefully to ensure an appropriate representation of solutions together with meaningful and problem-specific genetic operators.

The selection of individuals to produce successive generations plays an extremely important role in a genetic algorithm. A probabilistic selection is performed based upon the individual's fitness such that the better individuals have an increased chance of being selected. An individual in the population can be selected more than once with all individuals in the population having a chance of being selected to reproduce into the next generation. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament elitist models and ranking methods.

A common selection approach assigns a probability of selection, p_j, to each individual j, based upon its fitness value. A series of random numbers is generated and compared against the cumulative probability, $C_j = \sum_{i=1}^{j} p_i$, of the population. The appropriate individual, j, is selected and copied into the new population if $C_{j-i} < U(0,1) \leq C_j$, where $U(0,1)$ is a random variable uniformly distributed on the interval $(0,1)$. Various methods exist to assign probabilities to individuals: roulette wheel, linear ranking and geometric ranking.

Roulette wheel, developed by Holland, was the first selection model. The probability p_j for each individual is defined by:

$$p_j = P[\text{individual } j \text{ is chosen}] = \frac{f_j}{\sum_{i=1}^{pop\_size} f_i} \tag{5.1.1}$$

where f_i equals the fitness of individual i. It is easy to see that the individuals with larger fitness values have a larger probability of selection. Hence, the use of roulette wheel selection limits the genetic algorithm to maximization since the evaluation function must map the solution to a fully ordered set of values on R^+. Extensions, such as windowing and scaling, have been proposed to allow for minimization and negativity.

Ranking methods only require the evaluation function to map the solutions to a partially ordered set, thus allowing for minimization and negativity. Ranking methods assign p_j based on the rank of solution j when all solutions are sorted, where $rank = 1$ means the best individual and $rank = pop\_size$ the worst.

There are many methods to assign a number of offspring based on ranking. One possibility is to take a user defined parameter q and define a linear function, e.g.,

$$prob(rank) = q - (rank - 1)r \tag{5.1.2}$$

where the parameter r can be obtain from the condition $\sum_{i=1}^{pop\_size} prob(i) = 1$, or a nonlinear function, e.g.,

$$prob(rank) = q(1 - q)^{rank-1} \tag{5.1.3}$$

Both functions return the probability of an individual ranked in position $rank$ to be selected in a single selection.

Both schemes allow the user to influence the selective pressure of the algorithm. In the case of the linear function, the requirement:

$$\sum_{i=1}^{pop\_size} prob(i) = 1 \tag{5.1.4}$$

implies that:

$$q = \frac{r(pop\_size - 1)}{2} + \frac{1}{pop\_size} \tag{5.1.5}$$

or, equivalently:

$$r = \frac{2(q \cdot pop\_size - 1)}{pop\_size(pop\_size - 1)} \tag{5.1.6}$$

If $r = 0$ (and consequently $q = 1/pop\_size$) there is no pressure at all: all individuals have the same probability of selection. On the other hand, if $q-(pop\_size-1)r = 0$, then:

$$r = \frac{2}{pop\_size(pop\_size - 1)}\ , \text{ and } q = \frac{2}{pop\_size} \tag{5.1.7}$$

provides the maximum selective pressure. In other words, if a linear function is selected to provide the ranked individuals, a single parameter q, which varies between $1/pop\_size$ and $2/pop\_size$ can control the selective pressure of the algorithm. For example, if $pop\_size = 100$ and $q = 0.015$, then $r = q/(pop\_size - 1) = 0.00015151515$ and $prob(1) = 0.015$, $prob(2) = 0.0148484848$, ..., $prob(100) = 0.000000000000000000051$.

For the nonlinear function, the parameter $q \in (0, 1)$ does not depend on the population size; larger values of q imply stronger selective pressure of the algorithm. For example, if $q = 0.1$ and $pop\_size = 100$, then $prob(1) = 0.1$, $prob(2) = 0.09$, $prob(3) = 0.081$, ..., $prob(100) = 0.000003$. Note that:

$$\sum_{i=1}^{pop\_size} prob(i) = \sum_{i=1}^{pop\_size} q(1-q)^{i-1} = 1 - (1-q)^{pop\_size} \tag{5.1.8}$$

To obtain the equality with one, it is sufficient to define:

$$prob(rank) = q'q(1-q)^{rank-1} \tag{5.1.9}$$

where:

$$q' = \frac{1}{1-(1-q)^{pop\_size}} \tag{5.1.10}$$

The parameter q is one of the parameter of the genetic algorithms for which there is no optimum known value. The choice for the value of q depends on the problem. In chapter 6 we will present few experiments made for different values of q, in order to find the best value for our problem. This type of experiments will be made for the other parameters of the genetic algorithms as well.

Tournament selection, like ranking methods, only requires the evaluation function to map solutions to a partially ordered set, however, it does not assign probabilities. Tournament selection works by selecting j individuals randomly with replacement, form the population, and inserts the best of the j into the new population. This procedure is repeated until the population size is reached.

5.2 How genetic algorithms work; example

Let us apply the whole process of genetic algorithms and to sea how they work for the problem which we are dealing with. The implementation of these algorithms is made in Matlab and the codes are in Appendix.

From the beginning we have to specify that the example is for a set of input data created by ourself.

As we said before, the way in which a feasible solution is encoded is very important. Usually, a solution is represented by a binary string. This is the most used way to encode a solution. For this application, we tried first to use binary string. This had the disadvantage of increasing running time. Moreover, one property that we would like our algorithms to have is to be applicable for any other input data (as for any number of vessels, any number of ports, any number of cells). The binary approach met some difficulties when we wanted to modify the number of harbors.

The integer encoding used here does not have this problem. Moreover, the time used to transform the binary string into the integer solution and vice versa was decreased as well, and taking account that this problem is very time consuming, any reduction can be seen as a success.

Thus, let us denote by $v \in \{1, 2, \ldots, m\}^n$ the position vector, where $v(i) =$ the port in which vessel i is located. So, each solution is represented by a string of integer numbers from the set $\{1, 2, \ldots m\}$. The length of a chromosome is n genes, where each bit corresponds to the position of a vessel; we will denote the objective (the expected damage) by f, where f has the formula:

$$f(v) = \sum_{c=1}^{N} p_c D(c, v)$$

as in (4.2.1).

This seems to be a reasonable way to represent the solution of the optimization problem. More than this, it is easy to work with this representation.

Let us take the case of the spill with volume V between 10000 and 30000 m^3 (so the sixth category), with a mean spill size $V = 16962.5$. The probability to have one spill from this category in cell c, with $c \in \{1, 2, \ldots, 1097\}$ is denoted by $p(c)$.

For this example, the choice of the parameters (pop_size, max_gen, $probXover$, $probMutate$, q) is not based on a careful analysis; more values of these parameters and a try to find the optimum will be presented in the next chapter.

We consider also the case with 10 vessels, which are characterized by the data given in Table (2.3). So, $n = 10$ vessels, $m = 6$ harbors. To optimize the function f (4.2.1) using genetic algorithms, we use a population with $pop\_size = 20$ chromosomes. All n genes in each chromosome are generated randomly. For each gene, a random integer number from the set $\{1, 2, \ldots, m\}$ has to be generated.

Assume that after the initialization procedure, we get the following population:

$$
\begin{array}{rl}
v_1 = & (6 \quad 5 \quad 5 \quad 3 \quad 1 \quad 1 \quad 1 \quad 5 \quad 6 \quad 1) \\
v_2 = & (6 \quad 4 \quad 6 \quad 5 \quad 1 \quad 2 \quad 2 \quad 2 \quad 3 \quad 5) \\
v_3 = & (5 \quad 2 \quad 1 \quad 1 \quad 5 \quad 3 \quad 2 \quad 2 \quad 1 \quad 6) \\
v_4 = & (3 \quad 1 \quad 1 \quad 6 \quad 1 \quad 1 \quad 6 \quad 2 \quad 5 \quad 5) \\
v_5 = & (5 \quad 4 \quad 1 \quad 2 \quad 6 \quad 2 \quad 1 \quad 5 \quad 4 \quad 6) \\
v_6 = & (4 \quad 6 \quad 1 \quad 5 \quad 3 \quad 6 \quad 1 \quad 6 \quad 2 \quad 4) \\
v_7 = & (2 \quad 5 \quad 6 \quad 4 \quad 5 \quad 3 \quad 3 \quad 4 \quad 5 \quad 5) \\
v_8 = & (6 \quad 6 \quad 1 \quad 1 \quad 6 \quad 4 \quad 5 \quad 4 \quad 1 \quad 3) \\
v_9 = & (1 \quad 2 \quad 4 \quad 1 \quad 1 \quad 1 \quad 6 \quad 6 \quad 6 \quad 6) \\
v_{10} = & (2 \quad 2 \quad 2 \quad 2 \quad 1 \quad 1 \quad 4 \quad 5 \quad 2 \quad 3) \\
v_{11} = & (1 \quad 4 \quad 2 \quad 6 \quad 2 \quad 1 \quad 1 \quad 5 \quad 1 \quad 1) \\
v_{12} = & (3 \quad 1 \quad 5 \quad 6 \quad 1 \quad 6 \quad 2 \quad 5 \quad 4 \quad 5) \\
v_{13} = & (2 \quad 2 \quad 4 \quad 2 \quad 3 \quad 6 \quad 2 \quad 5 \quad 1 \quad 6) \\
v_{14} = & (6 \quad 2 \quad 6 \quad 6 \quad 1 \quad 2 \quad 1 \quad 6 \quad 3 \quad 1) \\
v_{15} = & (6 \quad 4 \quad 2 \quad 5 \quad 5 \quad 6 \quad 3 \quad 3 \quad 2 \quad 3) \\
v_{16} = & (2 \quad 1 \quad 5 \quad 1 \quad 6 \quad 5 \quad 4 \quad 5 \quad 1 \quad 1) \\
v_{17} = & (1 \quad 2 \quad 6 \quad 4 \quad 1 \quad 6 \quad 1 \quad 4 \quad 1 \quad 3) \\
v_{18} = & (1 \quad 5 \quad 4 \quad 3 \quad 4 \quad 3 \quad 5 \quad 2 \quad 3 \quad 6)
\end{array}
$$

$$v_{19} = (4 \quad 2 \quad 6 \quad 2 \quad 1 \quad 3 \quad 6 \quad 4 \quad 5 \quad 3)$$
$$v_{20} = (5 \quad 2 \quad 5 \quad 2 \quad 2 \quad 2 \quad 5 \quad 3 \quad 6 \quad 2)$$

In the evaluation phase, we calculate the fitness function value f. We get:

$$\begin{aligned}
f(v_1) &= 0.3041 & f(v_2) &= 0.2798 \\
f(v_3) &= 0.3175 & f(v_4) &= 0.2125 \\
f(v_5) &= 0.2897 & f(v_6) &= 0.2879 \\
f(v_7) &= 0.3220 & f(v_8) &= 0.2970 \\
f(v_9) &= 0.3036 & f(v_{10}) &= 0.3076 \\
f(v_{11}) &= 0.2106 & f(v_{12}) &= 0.2125 \\
f(v_{13}) &= 0.3847 & f(v_{14}) &= 0.3218 \\
f(v_{15}) &= 0.2725 & f(v_{16}) &= 0.2764 \\
f(v_{17}) &= 0.2018 & f(v_{18}) &= 0.2730 \\
f(v_{19}) &= 0.2760 & f(v_{20}) &= 0.3933
\end{aligned}$$

It is clear that chromosome v_{11} is the strongest one and the chromosome v_{20} is one of the weakest chromosomes.

Now, we are ready to apply the recombination operator crossover to the individuals in the new population. As mentioned earlier, one of the parameters of the genetic system is probability of crossover $probXover$. This probability gives us the expected number $probXover \cdot pop\_size$ of chromosomes which undergo the crossover operation. We proceed in the following way:

- for each pair of chromosomes, we generate a random number r from the range range $[0, 1]$;
- if $r < probXover$, select given pair of chromosomes for crossover.

Now we mate selected chromosomes randomly: for each pair of coupled chromosomes, we generate a random integer number pos from range $[1, n]$. The number pos indicates the position of the crossing point. Two chromosomes:

$$\begin{matrix} (b_1 & b_2 & \dots & b_{pos} & b_{pos+1} & \dots & b_n) \\ (c_1 & c_2 & \dots & c_{pos} & c_{pos+1} & \dots & c_n) \end{matrix}$$

are replaced by a pair of their offspring:

$$\begin{matrix} (b_1 & b_2 & \dots & b_{pos} & c_{pos+1} & \dots & c_n) \\ (c_1 & c_2 & \dots & c_{pos} & b_{pos+1} & \dots & b_n) \end{matrix}$$

For our example, let the probability of crossover be $probXover = 0.75$, so we expect that (on average) 75% of chromosomes (i.e., 15 out of 20) undergo crossover. We proceed in the following way: for pair of chromosomes in the population, we generate a random number r from the range $[0, 1]$; if $r < 0.75 = probXover$, we select the given chromosome for crossover.

Let us assume that sequence of random number is:

$$\begin{matrix} 0.6521 & 0.3241 & 0.8011 & 0.5356 & 0.5570 \\ 0.7412 & 0.9436 & 0.2105 & 0.7029 & 0.2636 \end{matrix}$$

This means that the chromosomes v_1, v_2, v_3, v_4, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{15}, v_{16}, v_{17}, v_{18}, v_{19}, v_{20} were selected for crossover.

Now,we mate selected chromosomes randomly: for each of these pairs, we generate a random integer number pos form the range $[0, 9]$; the number pos indicates the position of the crossing point. The first pair of chromosomes is:

$$\begin{matrix} v_1 = & (6 & 5 & 5 & 3 & 1 & 1 & 1 & 5 & 6 & 1) \\ v_2 = & (6 & 4 & 6 & 5 & 1 & 2 & 2 & 2 & 3 & 5) \end{matrix}$$

and the random number is $pos = 7$. These chromosomes are cut after the 7 bit and replaced by a pair of their offspring:

$$\begin{matrix} v'_1 = & (6 & 5 & 5 & 3 & 1 & 1 & 1 & 2 & 3 & 5) \\ v'_2 = & (6 & 4 & 6 & 5 & 1 & 2 & 2 & 5 & 6 & 1) \end{matrix}$$

The second pair of chromosomes is:

$$\begin{aligned} v_3 &= (5\ \ 2\ \ 1\ \ 1\ \ 5\ \ 3\ \ 2\ \ 2\ \ 1\ \ 6) \\ v_4 &= (3\ \ 1\ \ 1\ \ 6\ \ 1\ \ 1\ \ 6\ \ 2\ \ 5\ \ 5) \end{aligned}$$

and the random number is $pos = 9$. The chromosomes are replaced by a pair of their offspring:

$$\begin{aligned} v'_3 &= (5\ \ 2\ \ 1\ \ 1\ \ 5\ \ 3\ \ 2\ \ 2\ \ 1\ \ 5) \\ v'_4 &= (3\ \ 1\ \ 1\ \ 6\ \ 1\ \ 1\ \ 6\ \ 2\ \ 5\ \ 6) \end{aligned}$$

The same procedure is applied for all pairs of chromosomes. The respective cut points are: $pos = 7$, $pos = 9$, $pos = 6$, $pos = 5$, $pos = 8$, $pos = 8$, $pos = 1$, $pos = 5$. The current version of the population is:

$$\begin{aligned} v'_1 &= (6\ \ 5\ \ 5\ \ 3\ \ 1\ \ 1\ \ 1\ \ 2\ \ 3\ \ 5) \\ v'_2 &= (6\ \ 4\ \ 6\ \ 5\ \ 1\ \ 2\ \ 2\ \ 5\ \ 6\ \ 1) \\ v'_3 &= (5\ \ 2\ \ 1\ \ 1\ \ 5\ \ 3\ \ 2\ \ 2\ \ 1\ \ 5) \\ v'_4 &= (3\ \ 1\ \ 1\ \ 6\ \ 1\ \ 1\ \ 6\ \ 2\ \ 5\ \ 6) \\ v'_5 &= (5\ \ 4\ \ 1\ \ 2\ \ 6\ \ 2\ \ 1\ \ 5\ \ 4\ \ 6) \\ v'_6 &= (4\ \ 6\ \ 1\ \ 5\ \ 3\ \ 6\ \ 1\ \ 6\ \ 2\ \ 4) \\ v'_7 &= (2\ \ 5\ \ 6\ \ 4\ \ 5\ \ 3\ \ 5\ \ 4\ \ 1\ \ 3) \\ v'_8 &= (6\ \ 6\ \ 1\ \ 1\ \ 6\ \ 4\ \ 3\ \ 4\ \ 5\ \ 5) \\ v'_9 &= (1\ \ 2\ \ 4\ \ 1\ \ 1\ \ 1\ \ 4\ \ 5\ \ 2\ \ 3) \\ v'_{10} &= (2\ \ 2\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 6\ \ 6\ \ 6) \\ v'_{11} &= (1\ \ 4\ \ 2\ \ 6\ \ 2\ \ 1\ \ 1\ \ 5\ \ 4\ \ 5) \\ v'_{12} &= (3\ \ 1\ \ 5\ \ 6\ \ 1\ \ 6\ \ 2\ \ 5\ \ 1\ \ 1) \\ v'_{13} &= (2\ \ 2\ \ 4\ \ 2\ \ 3\ \ 6\ \ 2\ \ 5\ \ 1\ \ 6) \\ v'_{14} &= (6\ \ 2\ \ 6\ \ 6\ \ 1\ \ 2\ \ 1\ \ 6\ \ 3\ \ 1) \\ v'_{15} &= (6\ \ 4\ \ 2\ \ 5\ \ 5\ \ 6\ \ 3\ \ 3\ \ 1\ \ 1) \\ v'_{16} &= (2\ \ 1\ \ 5\ \ 1\ \ 6\ \ 5\ \ 4\ \ 5\ \ 2\ \ 3) \\ v'_{17} &= (1\ \ 5\ \ 4\ \ 3\ \ 4\ \ 3\ \ 5\ \ 2\ \ 3\ \ 6) \\ v'_{18} &= (1\ \ 2\ \ 6\ \ 4\ \ 1\ \ 6\ \ 1\ \ 4\ \ 1\ \ 3) \end{aligned}$$

$$v'_{19} = \quad (4 \quad 2 \quad 6 \quad 2 \quad 1 \quad 2 \quad 5 \quad 3 \quad 6 \quad 2)$$
$$v'_{20} = \quad (5 \quad 2 \quad 5 \quad 2 \quad 2 \quad 3 \quad 6 \quad 4 \quad 5 \quad 3)$$

The next operator, mutation, is performed on a bit-by-bit basis. Another parameter of the genetic system, probability of mutation $probMutate$, gives us the expected number of mutated bits, $probMutate \cdot n \cdot pop\_size$. Every bit (in all chromosomes in the whole population) has an equal chance to undergo mutation, i.e., to change from one value to one of the other $m - 1$ values. So, we proceed in the following way:

- Generate a random number r from the range $[0, 1]$.
- If $r < probMutate$, mutate the bit, i.e. generate an integer random number rr from the range $[1, n]$ which indicates the new value of that bit.

Let consider the probability of mutation be $probMutate = 0.1$, so we expect that (on average) 10% of bits would undergo mutation. There are $n \cdot pop\_size = 10 \cdot 20 = 200$ bits in the whole population; we expect (on average) 20 mutations per generation. Every bit has an equal chance to be mutated, so, for every bit in the population, we generate a random number r from the range $[0, 1]$; if $r < 0.05$, we mutate the bit. This means that we have to generate 200 random numbers. In a sample run, 23 of these numbers were smaller than 0.1. The following table lists the random numbers and translates the bit position into chromosome number and the bit number within the chromosome:

Random number	Bit position	Chromosome number	Bit number within chromosome
0.0383	1	1	1
0.0653	19	2	9
0.0309	43	5	3
0.0874	46	5	6
0.0492	49	5	9
0.0958	51	6	1
0.0663	53	6	3
0.0959	62	7	2
0.0646	66	7	6
0.0680	70	7	10
0.0482	76	8	6
0.0385	88	9	8
0.0903	105	11	5
0.0047	106	11	6
0.0430	122	13	2
0.0431	147	15	7
0.0939	148	15	8
0.0664	152	16	2
0.0967	157	16	7
0.0076	175	18	5
0.0031	179	18	9
0.0582	185	19	5
0.0639	188	19	8

This means that 23 chromosomes are affected by the mutation operator. For each of these bits we have to generate a random number rr, to find the new value of the bit. The random numbers are: 2, 3, 4, 6, 4, 4, 3, 4, 4, 5, 3, 4, 5, 4, 3, 3, 4, 3, 1, 4, 5, 4 and, respectively, 1. The final population is listed below; the mutation bits are typed in boldface.

$$
\begin{array}{rllllllllllll}
v''_1 = & (\mathbf{2} & 5 & 5 & 3 & 1 & 1 & 1 & 2 & 3 & 5) & \\
v''_2 = & (6 & 4 & 6 & 5 & 1 & 2 & 2 & 5 & \mathbf{3} & 1) & \\
v''_3 = & (5 & 2 & 1 & 1 & 5 & 3 & 2 & 2 & 1 & 5) & = v'_3 \\
v''_4 = & (3 & 1 & 1 & 6 & 1 & 1 & 6 & 2 & 5 & 6) & = v'_4 \\
v''_5 = & (5 & 4 & \mathbf{4} & 2 & \mathbf{6} & 6 & 1 & 5 & \mathbf{4} & 6) & \\
v''_6 = & (\mathbf{4} & 6 & \mathbf{3} & 5 & 3 & 6 & 1 & 6 & 2 & 4) & \\
v''_7 = & (2 & \mathbf{4} & 6 & 4 & 5 & \mathbf{4} & 5 & 4 & 1 & \mathbf{5}) & \\
v''_8 = & (6 & 6 & 1 & 1 & 6 & \mathbf{3} & 3 & 4 & 5 & 5) & \\
v''_9 = & (1 & 2 & 4 & 1 & 1 & 1 & 4 & \mathbf{4} & 2 & 3) & \\
v''_{10} = & (2 & 2 & 2 & 2 & 1 & 1 & 6 & 6 & 6 & 6) & = v'_{10} \\
v''_{11} = & (1 & 4 & 2 & 6 & \mathbf{5} & \mathbf{4} & 1 & 5 & 4 & 5) & \\
v''_{12} = & (3 & 1 & 5 & 6 & 1 & 6 & 2 & 5 & 1 & 1) & = v'_{12} \\
v''_{13} = & (2 & \mathbf{3} & 4 & 2 & 3 & 6 & 2 & 5 & 1 & 6) & \\
v''_{14} = & (6 & 2 & 6 & 6 & 1 & 2 & 1 & 6 & 3 & 1) & = v'_{14} \\
v''_{15} = & (6 & 4 & 2 & 5 & 5 & 6 & \mathbf{3} & \mathbf{4} & 1 & 1) & \\
v''_{16} = & (2 & \mathbf{3} & 5 & 1 & 6 & 5 & \mathbf{1} & 5 & 2 & 3) & \\
v''_{17} = & (1 & 5 & 4 & 3 & 4 & 3 & 5 & 2 & 3 & 6) & = v'_{17} \\
v''_{18} = & (1 & 2 & 6 & 4 & \mathbf{4} & 6 & 1 & 4 & \mathbf{5} & 3) & \\
v''_{19} = & (4 & 2 & 6 & 2 & \mathbf{4} & 2 & 5 & \mathbf{1} & 6 & 2) & \\
v''_{20} = & (5 & 2 & 5 & 2 & 2 & 3 & 6 & 4 & 5 & 3) & = v'_{20}
\end{array}
$$

Now, we have to select from this population, a new population, using the method of ranking. In this example, we use the normalized geometric ranking method (5.1.9). We construct the selection procedure as follows:

- construct a matrix with two columns, in which the first column contains indices of the chromosomes and the second column contains the rank of each chromosome (where 1 is the best one);

- using formula (5.1.9), calculate the probability to select the individual in position *rank*:

$$p_{rank} = prob(rank) = q'q(1-q)^{rank-1}$$

- calculate the cumulative probability q_i for each chromosome v_i, $i = 1, 2, \ldots pop\_size$:

$$Q_i = \sum_{j=1}^{i} p_j$$

The selection process consists in repeating *pop_size* times the following steps; each time we select a single chromosome for the new population:

- generate a random number r from the rank $[0, 1]$;
- if $r < Q_1$, then select the first chromosome (v_1); otherwise, select the ith chromosome v_i, ($2 \leq i \leq pop\_size$), such that $Q_{i-1} < r \leq Q_i$.

Applying these rules in our example, the probabilities of selection p_i for each chromosome v_i are:

$$\begin{array}{llll}
p_1 & = 0.01912 & p_2 & = 0.03623 \\
p_3 & = 0.04678 & p_4 & = 0.08865 \\
p_5 & = 0.03188 & p_6 & = 02806 \\
p_7 & = 0.01481 & p_8 & = 0.02469 \\
p_9 & = 0.10074 & p_{10} & = 0.01303 \\
p_{11} & = 0.07801 & p_{12} & = 0.13009 \\
p_{13} & = 0.02173 & p_{14} & = 0.01683 \\
p_{15} & = 0.05316 & p_{16} & = 0.06865 \\
p_{17} & = 0.06041 & p_{18} & = 0.11448 \\
p_{19} & = 0.01147 & p_{20} & = 0.04117
\end{array}$$

The cumulative probabilities q_i for each chromosome v_i are:

$$
\begin{array}{llll}
Q_1 & = 0.01912 & Q_2 & = 0.05535 \\
Q_3 & = 0.10213 & Q_4 & = 0.19079 \\
Q_5 & = 0.22267 & Q_6 & = 0.25073 \\
Q_7 & = 0.26553 & Q_8 & = 0.29022 \\
Q_9 & = 0.39097 & Q_{10} & = 0.40399 \\
Q_{11} & = 0.48201 & Q_{12} & = 0.61210 \\
Q_{13} & = 0.63383 & Q_{14} & = 0.65065 \\
Q_{15} & = 0.70382 & Q_{16} & = 0.77247 \\
Q_{17} & = 0.83288 & Q_{18} & = 0.94736 \\
Q_{19} & = 0.95883 & Q_{20} & = 1.0000
\end{array}
$$

Now, we are ready to generate a sequence of 20 random numbers; each time we select a single chromosome for a new population. Let us assume that a (random) sequence of 20 numbers from the range $[0, 1]$ is:

0.23995	0.24172	0.30540	0.33229	0.41099
0.42479	0.46376	0.46877	0.47675	0.64722
0.66016	0.69859	0.70593	0.71720	0.82933
0.86520	0.88142	0.92279	0.95429	0.97060

The first number is $r = 0.23995$ is greater than $Q_5 = 0.10506$ and smaller than $Q_6 = 0.20580$, meaning that the chromosome v_6 is selected for the new population; the second number $r = 0.24172$ is also greater than $Q_5 = 0.28382$ and smaller than $Q_6 = 0.39830$, meaning the chromosome v_6 is selected again for the new population.

Finally, the new population consists of the following chromosomes:

$$
\begin{array}{ll}
v_1 = & (4 \quad 6 \quad 3 \quad 5 \quad 3 \quad 6 \quad 1 \quad 6 \quad 2 \quad 4) \\
v_2 = & (4 \quad 6 \quad 3 \quad 5 \quad 3 \quad 6 \quad 1 \quad 6 \quad 2 \quad 4) \\
v_3 = & (1 \quad 2 \quad 4 \quad 1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 2 \quad 3) \\
v_4 = & (1 \quad 2 \quad 4 \quad 1 \quad 1 \quad 1 \quad 4 \quad 4 \quad 2 \quad 3)
\end{array}
$$

$$
\begin{aligned}
v_5 &= (1\ \ 4\ \ 2\ \ 6\ \ 5\ \ 4\ \ 1\ \ 5\ \ 4\ \ 5) \\
v_6 &= (1\ \ 4\ \ 2\ \ 6\ \ 5\ \ 4\ \ 1\ \ 5\ \ 4\ \ 5) \\
v_7 &= (1\ \ 4\ \ 2\ \ 6\ \ 5\ \ 4\ \ 1\ \ 5\ \ 4\ \ 5) \\
v_8 &= (1\ \ 4\ \ 2\ \ 6\ \ 5\ \ 4\ \ 1\ \ 5\ \ 4\ \ 5) \\
v_9 &= (1\ \ 4\ \ 2\ \ 6\ \ 5\ \ 4\ \ 1\ \ 5\ \ 4\ \ 5) \\
v_{10} &= (6\ \ 2\ \ 6\ \ 6\ \ 1\ \ 2\ \ 1\ \ 6\ \ 3\ \ 1) \\
v_{11} &= (6\ \ 4\ \ 2\ \ 5\ \ 5\ \ 6\ \ 3\ \ 4\ \ 1\ \ 1) \\
v_{12} &= (6\ \ 4\ \ 2\ \ 5\ \ 5\ \ 6\ \ 3\ \ 4\ \ 1\ \ 1) \\
v_{13} &= (2\ \ 3\ \ 5\ \ 1\ \ 6\ \ 5\ \ 1\ \ 5\ \ 2\ \ 3) \\
v_{14} &= (2\ \ 3\ \ 5\ \ 1\ \ 6\ \ 5\ \ 1\ \ 5\ \ 2\ \ 3) \\
v_{15} &= (1\ \ 5\ \ 4\ \ 3\ \ 4\ \ 3\ \ 5\ \ 2\ \ 3\ \ 6) \\
v_{16} &= (1\ \ 2\ \ 6\ \ 4\ \ 4\ \ 6\ \ 1\ \ 4\ \ 5\ \ 3) \\
v_{17} &= (1\ \ 2\ \ 6\ \ 4\ \ 4\ \ 6\ \ 1\ \ 4\ \ 5\ \ 3) \\
v_{18} &= (1\ \ 2\ \ 6\ \ 4\ \ 4\ \ 6\ \ 1\ \ 4\ \ 5\ \ 3) \\
v_{19} &= (4\ \ 2\ \ 6\ \ 2\ \ 4\ \ 2\ \ 5\ \ 1\ \ 6\ \ 2) \\
v_{20} &= (5\ \ 2\ \ 5\ \ 2\ \ 2\ \ 3\ \ 6\ \ 4\ \ 5\ \ 3)
\end{aligned}
$$

We have just completed one iteration (i.e., one generation) of the while loop in the genetic procedure. It is interesting to examine the results of the evaluation process of the new population. During the evaluation phase, we calculate the fitness function (expected damage) for each chromosome. We get:

$$
\begin{aligned}
f(v_1) &= 0.2879 & f(v_2) &= 0.2879 \\
f(v_3) &= 0.2061 & f(v_4) &= 0.2061 \\
f(v_5) &= 0.2528 & f(v_6) &= 0.2528 \\
f(v_7) &= 0.2528 & f(v_8) &= 0.2528 \\
f(v_9) &= 0.2528 & f(v_{10}) &= 0.3218 \\
f(v_{11}) &= 0.2748 & f(v_{12}) &= 0.2748 \\
f(v_{13}) &= 0.2633 & f(v_{14}) &= 0.2633 \\
f(v_{15}) &= 0.2730 & f(v_{16}) &= 0.2018
\end{aligned}
$$

$$f(v_{17}) = 0.2018 \quad f(v_{18}) = 0.2018$$
$$f(v_{19}) = 0.3693 \quad f(v_{20}) = 0.2834$$

Note that the average fitness of the population is 0.2591 , which is lower than the average fitness of the previous population, 0.2867.

Now we are ready to apply the genetic operators,to evaluate the next resulting population and to run the selection process again, etc. After 15 generations, the population is:

$$\begin{aligned}
v_1 &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_2 &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_3 &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_4 &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_5 &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_6 &= (3\ \ 5\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_7 &= (3\ \ 5\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_8 &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_9 &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{10} &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{11} &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{12} &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{13} &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{14} &= (3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{15} &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3)\\
v_{16} &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_{17} &= (3\ \ 3\ \ 2\ \ 2\ \ 5\ \ 1\ \ 1\ \ 2\ \ 5\ \ 3)\\
v_{18} &= (3\ \ 3\ \ 2\ \ 5\ \ 5\ \ 3\ \ 6\ \ 4\ \ 4\ \ 3)\\
v_{19} &= (3\ \ 3\ \ 2\ \ 2\ \ 2\ \ 3\ \ 6\ \ 2\ \ 5\ \ 3)\\
v_{20} &= (3\ \ 3\ \ 2\ \ 6\ \ 5\ \ 6\ \ 6\ \ 3\ \ 2\ \ 3)
\end{aligned}$$

and the fitness values are all equal to 0.1993 for v_6 and v_7 and 0.1862 for the rest of chromosomes.

It may happen in a more complex problem, that in earlier generations the fitness values of some chromosomes are better than the value of the best chromosome in the last generation. This is due to the stochastic nature of sampling.

It is relatively easy to keep track of the best individual in the evolution process. It is customary (in genetic algorithm implementation) to store "the best ever" individual at a separate location; in that way, the algorithm would report the best value found during the whole process (as opposed to the best value in the final population).

The optimum location vectors (all give the same value for the objective function) are:

$$\begin{pmatrix} 3 & 3 & 2 & 6 & 5 & 6 & 5 & 6 & 5 & 3 \end{pmatrix}$$
$$\begin{pmatrix} 3 & 3 & 2 & 6 & 1 & 6 & 2 & 4 & 6 & 3 \end{pmatrix}$$
$$\begin{pmatrix} 3 & 2 & 2 & 2 & 5 & 1 & 6 & 4 & 4 & 3 \end{pmatrix}$$

Figures 5.2 shows the evolution of the genetic algorithms for this simple example.

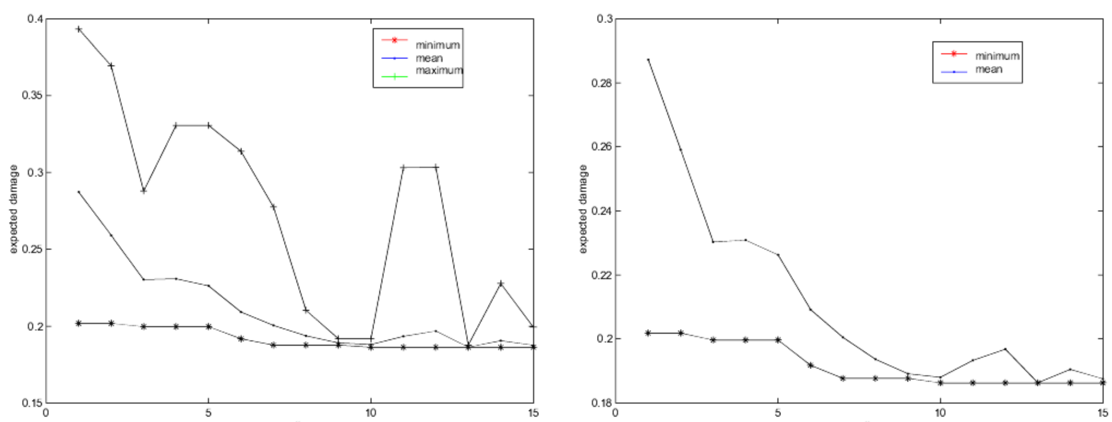


Figure 5.1: Simple example

As it can be seen, it is possible to obtain more optimum solutions. For this practical problem, all the optimum solutions have to be presented. The final choice for one of them depends on the specialists, the people that will transform in practice the results of this study; it is probable that other criteria need to be used as well. Moreover, most of the time, the optimum solutions have many similarities; thus, some of the cleaning vessels have to be located in certain harbors and for the rest of the vessels, any of the possibilities gives the same result.

As an example, for the optimum solutions obtained before, it can be seen that the fixed vessels are 1, 2, 3, 4, 7, 8 and 10, that have to be located in harbors 5, 5, 2, 6, 3, 3 and respectively 5. Vessels 5 can be located either in port 1 or 2; vessel 6 can be located in port 4, 5 or 6 and vessel 9 can be located in harbor 2 or 5.

Situations like this will be found as well in the next section; possible suggestions to deal with such cases are given as well.

Chapter 6

Results and discussions

In this chapter, the optimum solutions of the optimization problem are presented. In the first section, we analyze how the parameters of the genetic algorithms influence the results. The computations are made for the first category of oil spills and for the four vessels owned by RWS. The results for the all eight spill categories are presented in the second chapter, together with interpretations of these results. For the computations from this section, it has been assumed that if more vessels are used in a cleanup operation, they have to wait for the slowest one and to start to clean the water in the same time.

6.1 Sensitivity analysis of the genetic algorithm parameters

There are many parameters used in genetic algorithms that have values given by user: mutation rate ($probMutate$), crossover rate ($probXover$), population size (pop_size), maximum number of generations (max_gen), probability of selection for the best individual in a population (q). There is no known optimum value for any of these parameters. They can be chosen only after a few experiments, for each problem separately.

In this section we will analyze each of these parameters. The computations were made for the first category of oil spills and only for four vessels. This choice was made because the spills from this category have a smaller frequency and for only four vessels the running time is much less.

Mutation rate ($probMutate$)

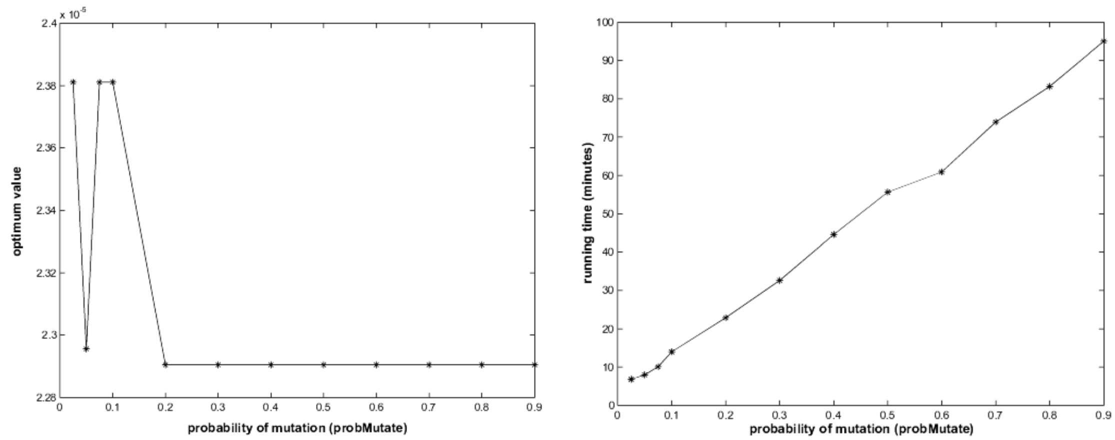
As we said in Chapter 5, the mutation rate is defined as the percentage of the total number of genes in the population that are changed by the mutation operator. It influences the expected number of new genes that are introduced into the population. If it is too low, many genes that would be useful are never tried out and the algorithm doesn't explore too much from the search space; if it is too high, there will be much random perturbations, the offsprings will start loosing their resemblance to parents and the algorithm will lose the ability to learn from the history of the search. Moreover, for our particular problem, the running time will increase considerably and this is a very important aspect that we have to take into consideration.

In general, if n is the number of cleaning vessels that have to be located (so an individual has n genes), the expected number of genes that undergo the mutation is: $probMutate \times n \times pop\_size$. If we would like on average to change one gene in each individual from a population, we need that the expected number of mutations to be equal to the population size; hence, for the probability of mutation we obtain: $probMutate = 1/n$.

Thus, in order to have on average one mutation per each individual from a population, we should choose $probMutate = 0.25$ in case that only four vessels are introduced in the study and $probMutate = 0.1$ for ten vessels.

Here we present the optimum values and the running time for more values of the probability of mutation, varying between 0.025 and 0.9. We gave to the rest of the parameters the following values: $probXover = 0.75$, $pop\_size = 20$, $max\_gen = 15$, $q = 0.1$.

It can be seen in figure (6.1) that for values of the probability of mutation larger



(a) (b)

Figure 6.1: Sensitivity in probability of mutation

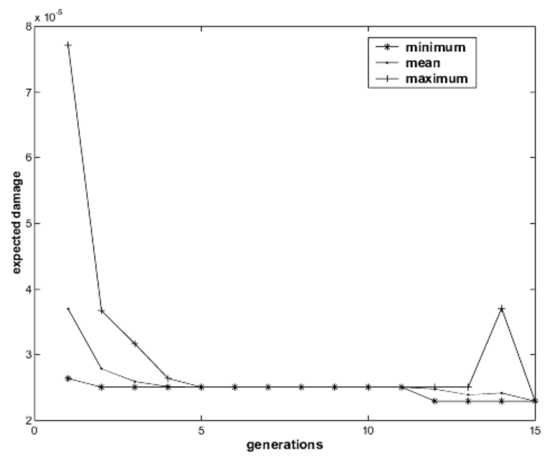
then 0.2, all computations give the same optimum value. The running time is linear increasing as function of probability of mutation.

Figure (6.2) shows how the genetic algorithms work for two different values of the probability of mutation. For a smaller value (a), the genetic algorithms don't vary too much, but for a higher value of the probability of mutation, there are big perturbations.

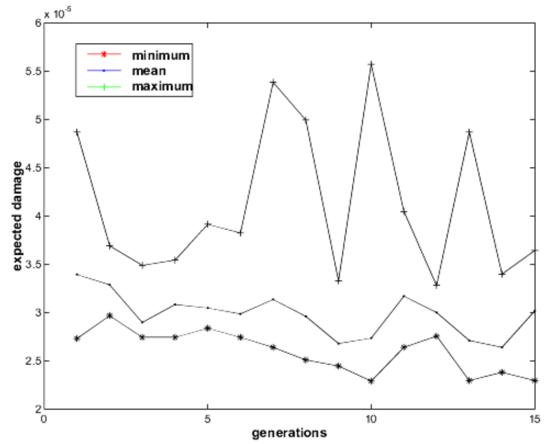
For the rest of the study we will use for the probability of mutation the value 0.2 if the analysis is made only for four vessels and the value 0.1 is ten vessels are considered. Hence, for these experiments we have $probMutate = 0.2$.

Crossover rate ($probXover$)

The crossover rate (denoted by $probXover$) represents the percentage of pairs that undergo the crossover operator. A higher probability of crossover allows exploration of more of the feasible solution set and reduces the chance of setting to local optimum; if the rate is too high, it results in a large computation time.



(a) $probMutate = 0.05$



(b) $probMutate = 0.7$

Figure 6.2: Examples for two values of the probability of mutation

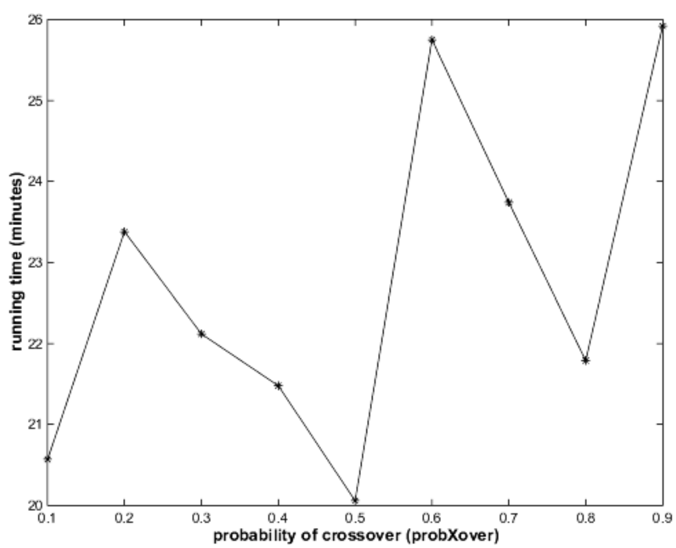
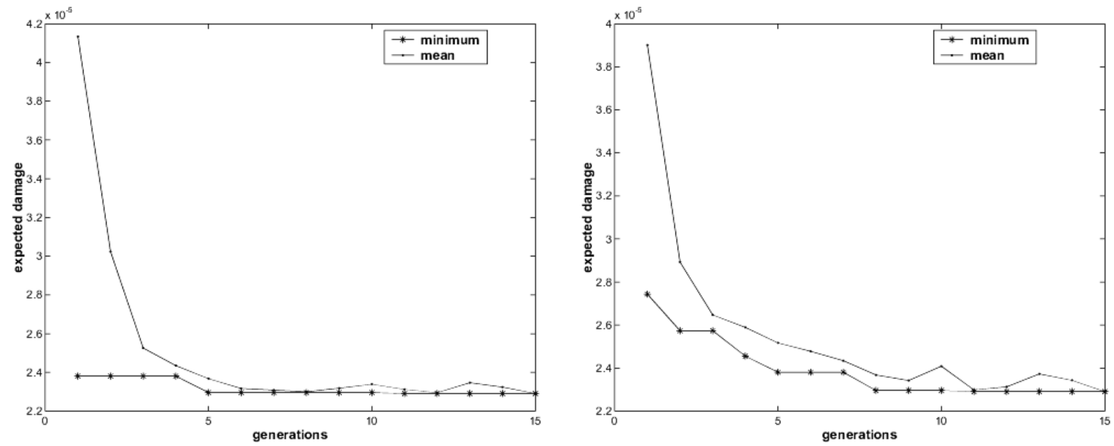


Figure 6.3: Sensitivity in probability of crossover



(a) $probXover = 0.2$ (b) $probXover = 0.7$

Figure 6.4: Examples for two values of the probability of crossover

In order to analyze how the crossover rate influences the results of the genetic algorithms and the running time, we made computations for the values of the probability of crossover in the interval $[0.1; 0.9]$. For the rest of the genetic algorithm parameters, we took the values: $probMutate = 0.2$, $pop\_size = 20$, $max\_gen = 15$ and $q = 0.1$. For the optimum value there is no variation; all the time there is the same value, namely $2.2905 * 10^{-5}$. The running time is not increasing, in contrast with what we have expected.

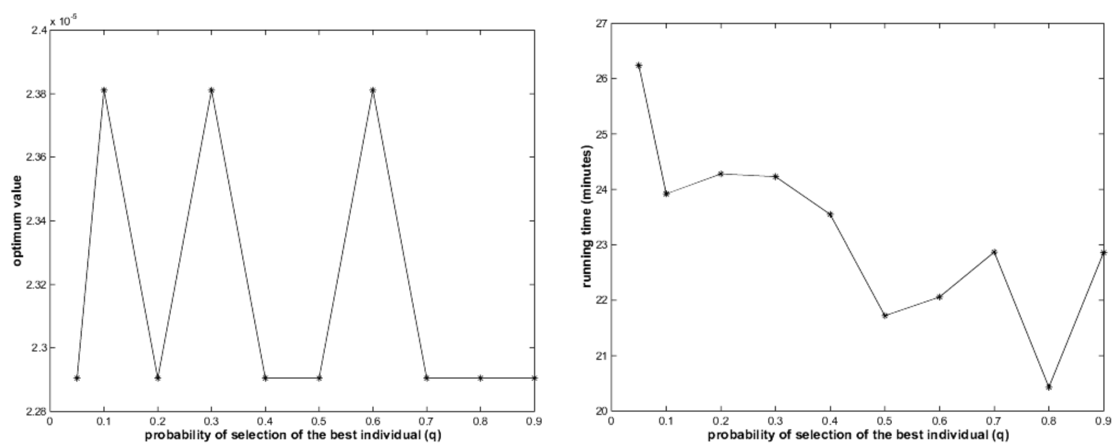
We have to say that after the selection operation, if one chromosomes is chosen more times, its copies are introduced in the new population each after other. Taking into account that the crossover operator is applied to two consecutive chromosomes, many of the pairs, even if they undergo the crossover operator, do not bring new chromosomes, so the expected damage (the fitness function) doesn't have to be computed again. This explain why the crossover probability doesn't influence too much the running time. Anyway, to increase the chance to introduce new chromosomes, we should give a large value to the crossover rate.

The figure (6.3) shows how the running time varies as a function of the crossover probability. The next figure presents how the genetic algorithms work for two different values of the crossover probability.

For the rest of the study we chose $probXover = 0.75$, based on the previous explanations.

Probability of selection of the best individual in a population (q)

In the selection phase of a genetic algorithm, a very important role is played by the probability of selection of the best individual in a population, which is controlled by the parameter q. Large values of q imply stronger pressure of the algorithm, so more copies of the best individual from a population will by selected for the next generation. A smaller value provides a larger variety of chromosomes in a population.



(a) (b)

Figure 6.5: Sensitivity in probability of selection of the best individual

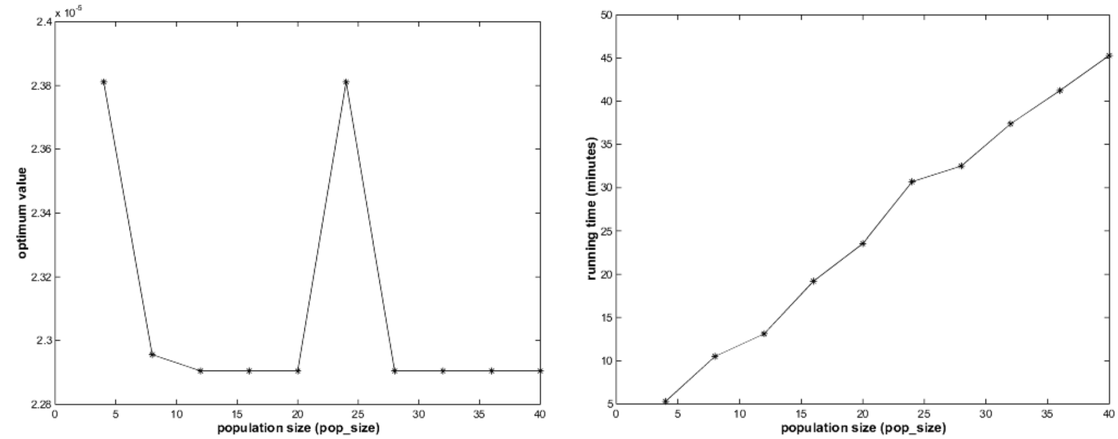
A few experiments are made for values of q in the interval [0.05 0.9]. From figure (6.5) it can be seen that the value of q doesn't influence the optimum solution found by the genetic algorithm; the variation in the optimum value that is found by the genetic algorithms appears due to their stochastic character. For the running time, larger values of q allow the appearance of more copies of the most fitted chromosomes, and during the crossover operator there are not too many new chromosomes (so there is no need to compute the expected damage, i.e. the fitness function for too many new individuals). This explain the fact that the running time seams to be decreasing as function of q.

Taking account that the crossover operator is applied to two consecutive chromosomes, we should avoid to have too many copies of the same individual in one population, such that we can search more from the set of feasible solutions. Therefore, we chose for the rest of the study the value 0.1. Hence, the probability of selection of individuals in a population will be: $prob(1) = 0.1138$, $prob(2) = 0.1025$, $prob(3) = 0.0922$, $\ldots$, $prob(20) = 0.0154$.

Population size (pop_size) and maximum number of generations (max_gen)

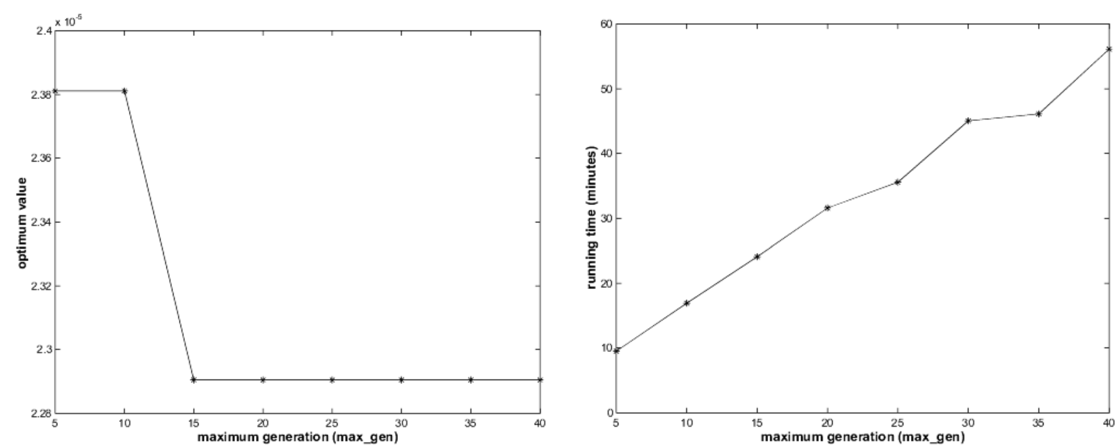
It is very difficult to say what is the best number of chromosomes that a population needs to have. Of course, a larger population permits a larger exploration of the search space. But if the genetic algorithms are very time consuming, this doesn't allow us to work with a large population. Figure (6.6) shows how the optimum value and the running time vary with the population size.

After experiments made for the ten vessels as well, we choose the population size to be equal to 20 for our further studies.



(a) (b)

Figure 6.6: Sensitivity in population size



(a) (b)

Figure 6.7: Sensitivity in maximum number of generations

Maximum number of generations is again a difficult choice and the running time is the criterium used to set the value for the parameter *max_gen*.

We have to remark that the running time doesn't look to be too large (at most one hour), because these computations are made only for the first group of vessels, so only for four cleaning vessels. Thus, to compute the expected damage given a location vector and a pollution in a certain cell, there are only $2^4 - 1 = 15$ possibilities to use cleaning vessels, so 15 cases to analyze for each individual into the population. For ten cleaning vessels, the number of cases that have to be analyzed for each individual is $2^{10} - 1$ and the running time will increase to $25 - 29$ hours, even more, as we will see in the next section. For our problem we choose the maximum number of generation equal to 15.

6.2 Results

In this section we will present the results and we will try to interpret them. The computations are made for each category separately and for two groups of vessels (first group includes four vessels, and the second group includes six more vessels).

First category of oil spills

The first category of oil spills includes the spills that have the volume less then 20 m^3. Their frequency is 0.001653 per year. The mean spill size is 12 m^3; this value we will use in our computations.

The computations for the first group of vessels ended up with a solution which brings a reduction of 25% of the damage. Figure (6.8) shows the convergence of the genetic algorithms.

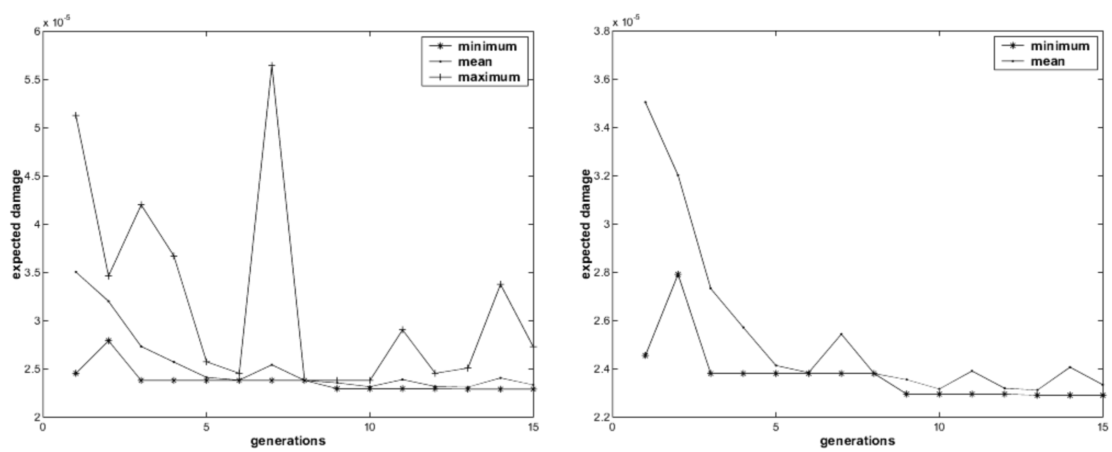
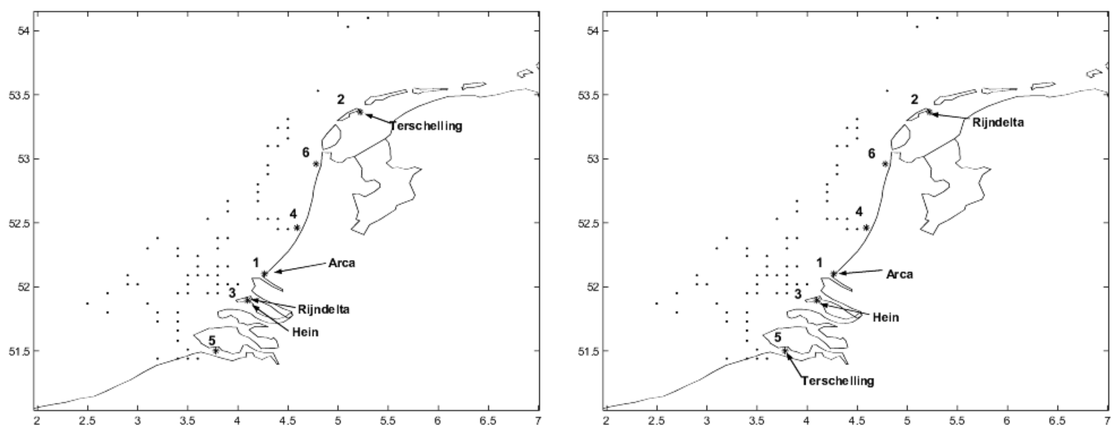


Figure 6.8: Convergence of the genetic algorithms for first category of oil spills, first group of vessels

The location vector is $v = (1\ 5\ 2\ 2)$. The solution implies that Arca has to be located in Scheveningen (as it is now), Terschelling in Terneuzen, Rijndelta and Hein in Terschelling. The actual location and the optimum location of the cleaning vessels are presented in figure (6.9).



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (1\ 5\ 2\ 3)$

Figure 6.9: Location of the cleaning vessels from the first group, for the first category of oil spills

We have to remark that the location of vessel Hein is not important, since if we change its location, we obtain the same value for the expected damage. So, all the location vectors with the form: $v = (1\ 5\ 2\ *)$ give the same value of the expected damage, and other criterium has to be used to decide the location of vessel four. Also, for vessels 1 and 3 the location of the arms is on board, but for vessel 2, the location is in port 2 (Terschelling); more then this, in the optimum location vector, it seems that the best location for vessel 2 is in port 5 (Terneuzen), so at the other end of the coast. We have to remind that a vessel which hasn't the arms on board has to sail first to the arms location and then to leave for the polluted area. So, the vessel 2 has to sail almost 10 hours to the arm location. Hence, it would be interesting to see what would happen if we locate the vessel 2 in the port where its arms are; this means that we would like to compute the expected damage if the location vector is (1 2 2 2). The result is $2.2952 \cdot 10^{-5}$, which is still better then the expected damage for the actual location of the cleaning vessels ($3.0450 \cdot 10^{-5}$), but worse than the optimum value

$(2.2905 \cdot 10^{-5})$.

For the second group of vessels, the optimum value of the expected damage is $2.115 \cdot 10^{-5}$ and this is obtained for the location vectors:

$$\begin{pmatrix} 3 & 5 & 2 & 4 & 3 & 6 & 3 & 5 & 4 & 4 \end{pmatrix}$$
$$\begin{pmatrix} 3 & 5 & 2 & 4 & 3 & 6 & 1 & 5 & 4 & 4 \end{pmatrix}$$
$$\begin{pmatrix} 3 & 5 & 2 & 4 & 3 & 4 & 3 & 5 & 4 & 4 \end{pmatrix}$$

The result shows that only vessel 7 (Geopotes) remains (eventually) at the same location as present, but the others vessels have to be moved. The reduction in the value of the expected damage would be 28%. The actual and the optimum locations are presented in figure (6.11)

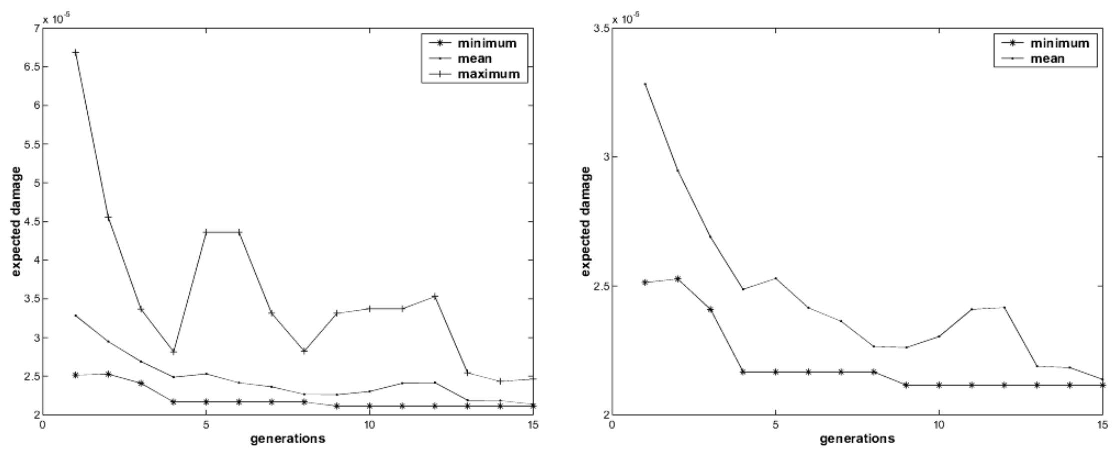
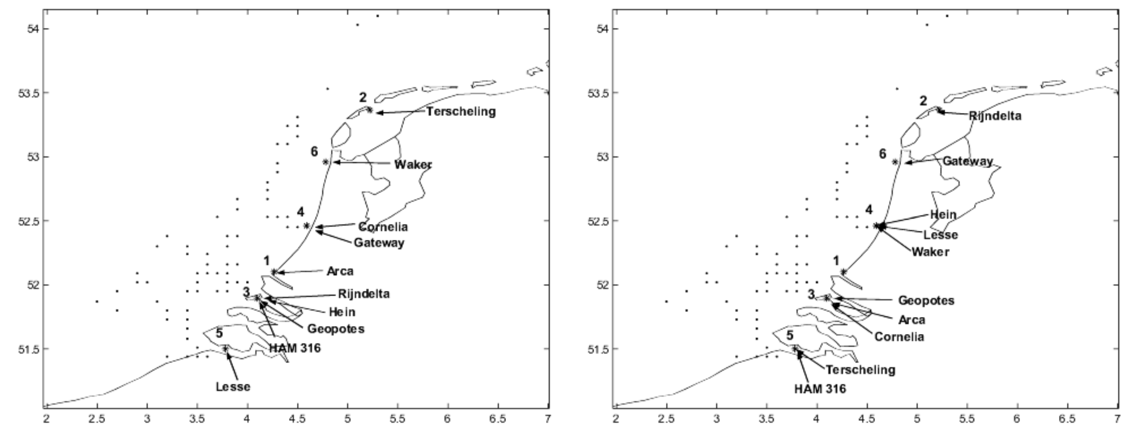


Figure 6.10: Convergence of the genetic algorithms for first category of oil spills, second group of vessels

It is difficult to interpret the optimum location found, but several remarks can be made, especially if we look at the arm locations.

The first vessel seems to be very effective, since it is one of the fastest vessels, has a large cleaning capacity, the mobilization time is only one hour and, more than this, the arms are located on board. This can explain why it is located in port 3



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (3\ 5\ 2\ 4\ 3\ 6\ 3\ 5\ 4\ 4)$

Figure 6.11: Location of the cleaning vessels from the second group, for the first category of oil spills

(Rotterdam), where the largest probability of pollution seems to be. It would be interesting to see what will happen if we locate this vessel in port 1 (Scheveningen), where it is in its actual location. If we change only vessel 1, three possible location vectors appear:

$$\begin{aligned} v_1 &= (1 \quad 5 \quad 2 \quad 4 \quad 3 \quad 6 \quad 3 \quad 5 \quad 4 \quad 4) \\ v_2 &= (1 \quad 5 \quad 2 \quad 4 \quad 3 \quad 6 \quad 1 \quad 5 \quad 4 \quad 4) \\ v_3 &= (1 \quad 5 \quad 2 \quad 4 \quad 3 \quad 4 \quad 3 \quad 5 \quad 4 \quad 4) \end{aligned}$$

and the value of the expected damage is equal to $2.2213 \cdot 10^{-5}$, which increases the value with 5% comparing with the optimum value. Hence it seems to be important that the vessel 1 to be located in port 1.

For vessels 2 and 3 (Terschelling and Rijndelta) we have to remark that their optimum locations in this group of vessels coincides with their optimum locations in case that only four vessels are considered. Again vessel 2 has to sail to the other end of the coast, which delays with 10 hours the starting moment of the cleanup operation. This makes us to check what happens if we move this vessel in the port where the arms are located; in this way, we obtain three now location vectors:

$$
\begin{aligned}
v_4 &= (3 \quad 2 \quad 2 \quad 4 \quad 3 \quad 6 \quad 3 \quad 5 \quad 4 \quad 4) \\
v_5 &= (3 \quad 2 \quad 2 \quad 4 \quad 3 \quad 6 \quad 1 \quad 5 \quad 4 \quad 4) \\
v_6 &= (3 \quad 2 \quad 2 \quad 4 \quad 3 \quad 4 \quad 3 \quad 5 \quad 4 \quad 4)
\end{aligned}
$$

which give the same value of the expected damage: $2.4620 \cdot 10^{-5}$.

Vessels 4, 5, 6 and 7 are all located close or in the port where there arms are. Comparing with the mobilization time which is 10 or 12 hours, the sailing time to the arm location which is at most 3 hours is not too much. Taking into account that in this model we assume that the vessels have to wait for the slowest one to start the cleanup operation, the large time before these vessels leave for the polluted area can explain why for the vessel number 2, although it has its arms at one end of the coast, its optimum location seems to be at the other end of the coast. We have to remark also that in the three optimum location vectors that we have found there are eight vessels that have the same location and only two (vessels 6 and 7) have different locations. Thus it would be good to check what it is the value of the expected damage if we locate these vessels where their arms are, namely in port 3 (Rotterdam); in this case, at least the cost of the sailing to the arms location would be saved. Hence, for the location vector $v = (3\ 5\ 2\ 4\ 3\ 3\ 3\ 5\ 4\ 4)$ we obtain the value of the expected damage $2.1150 \cdot 10^{-5}$, which is equal to the optimum value. Thus, this location vector can be also considered optimum.

For vessel 8 (HAM 316), the optimum location is again not very far from the port where its arms are (only 2.5 sailing hours). If we move the vessel in port Rotterdam to save this time and also the cost of the sailing, the expected damage would be equal to the optimum value.

The same situation holds for vessel 9 (Lesse), with a larger time (5 hours) to reach the port where its arms are located. An eventual move of this vessel in port Terneuzen brings no change in the value of the expected damage.

One problem that we have to solve is to find a way to combine more optimum locations vectors and to combine also the results for the two groups of vessels. For the first case, it can be seen that the optimum vectors found have many similarities and only few vessels appear with more possible optimum locations. It is very probable that some other considerations have to be taken into account to decide the location for this kind of vessels. One possibilities is to locate the vessels where their arms are, if this doesn't influence the value of the expected damage.

In order to combine the results for the two groups of vessels, we can consider for the first four vessels the optimum location found for the first group of vessels and for the other six vessels, the optimum location found for the second group of vessels. Hence, if the two optimum location vectors are (1 5 2 2) and respectively (3 5 2 4 3 6 3 5 4 4), for example, it would be interesting to check how the value of the expected damage would change if the location vector is (1 5 2 2 3 6 3 5 4 4) and to compare this value with the actual value of the expected damage and with the optimum value found. The results shows that this new location vectors increases the expected damage with 5% comparing with the optimum value, but is still with 25% better than the actual value.

The second category of oil spills

In this category are included the spills that have the volume between 20 and 150 m^3; the average volume of oil is 76.2 m^3. The frequency of spills from this category is 0.003442 per year, which is larger than the frequency of spills from the first category.

The actual location of the four vessels which are owned by RKS gives an expected damage with the value $2.3984 \cdot 10^{-4}$. The genetic algorithms applied for this category of oil spills find an optimum value of the expected damage equal to $1.891 \cdot 10^{-4}$, which brings a reduction with 20%. The optimum locations found by the genetic algorithms are:

$$v_1 = (3 \quad 5 \quad 2 \quad 5)$$
$$v_2 = (3 \quad 5 \quad 2 \quad 1)$$

The convergence of the genetic algorithms is presented in figure (6.12).

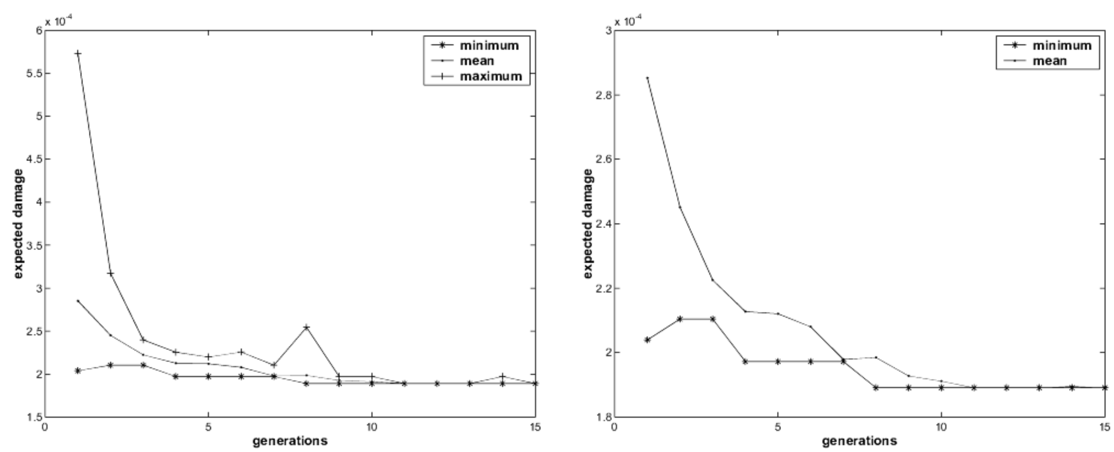
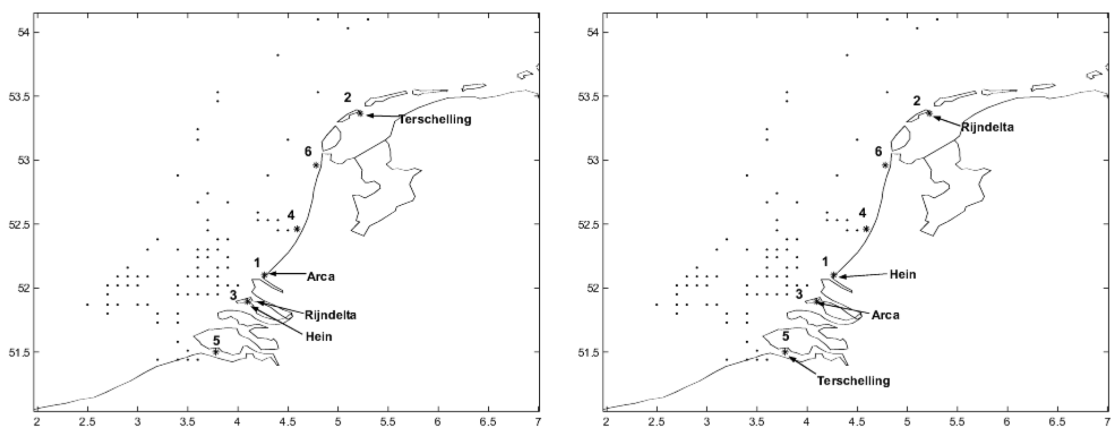


Figure 6.12: Convergence of the genetic algorithms for the second category of oil spills, first group of vessels

The same value of the expected damage is obtained if we move the fourth vessel in any of the six ports. Hence, again, the optimum location vector for this category of oil spills has the form: $(3\ 5\ 2\ *)$.

We have to remark that comparing with the results obtained for the first category, only the first vessel (Arca) has another optimum location. If for the previous category, the optimum location was at Scheveningen, for this category, the optimum location is at Rotterdam. Moreover, if we compute the value of the expected damage with the first vessel located in port 1 (as for the previous category), we obtain a larger value ($1.9042 \cdot 10^{-4}$, so 10% larger).

Analyzing the possibilities of locations derived from the optimum location vector, namely moving the vessels in the port where there arms are located, we do not obtain any other location which gives a better value of the expected damage.



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 5\ 2\ 1)$

Figure 6.13: Location of the cleaning vessels from the first group, for the second category of oil spills

For the second group of vessels, the optimum value of the expected damage ($1.8590 \cdot 10^{-4}$) brings an improvement with 23% comparing with the value of the expected damage for the actual locations of the cleaning vessels ($2.3986 \cdot 10^{-4}$). The optimum solutions are:

$$(4\quad 5\quad 2\quad 5\quad 6\quad 1\quad 6\quad 3\quad 6\quad 3)$$
$$(4\quad 5\quad 2\quad 5\quad 6\quad 1\quad 6\quad 3\quad 4\quad 3)$$

which differ only by the location of vessel 9 (Lesse).

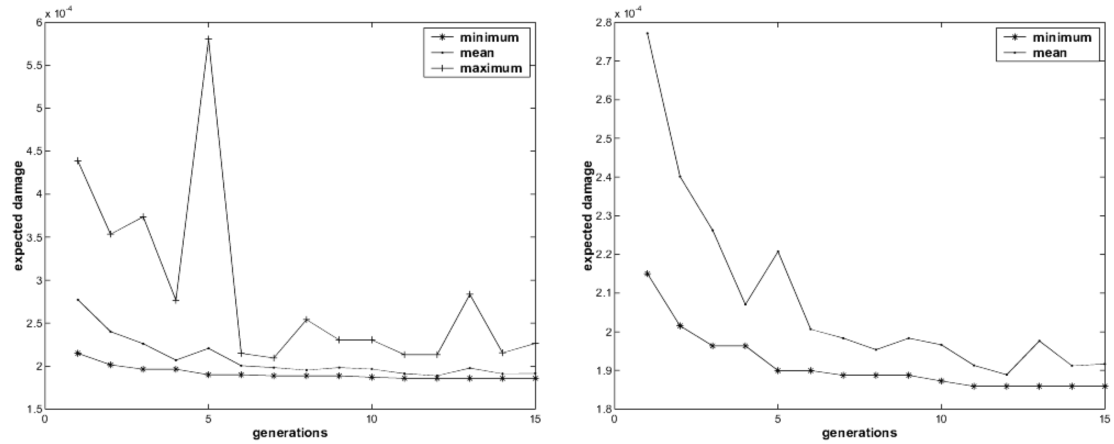


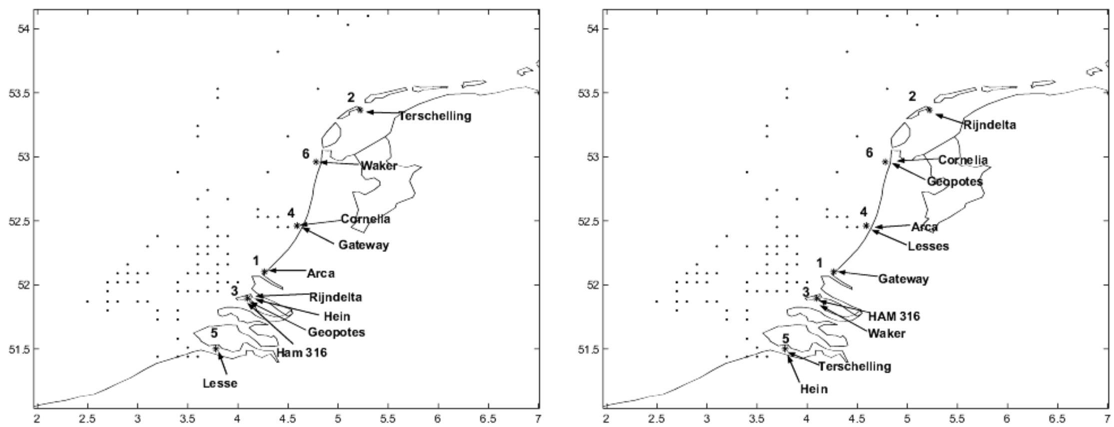
Figure 6.14: Convergence of the genetic algorithms for the second category of oil spills, the second group of vessels

Comparing with the actual location (1 2 3 3 4 4 3 3 5 6), we observe that only vessel 8 has the same location and all the other vessels have to be moved in other ports.

Vessel 1 (Arca) has the optimum location in port 4 (IJmuiden), which is very close to the actual location (Scheveningen), only 1.76 sailing hours. But, if we locate Arca in Scheveningen, hence the location vector would be (1 5 2 5 6 1 6 3 6 3) the expected damage is $1.9092 \cdot 10^{-4}$.

For vessels 2 and 3 (Terschelling and Rijndelta), their optimum locations are the same with their optimum locations found for the first group of vessels, and moreover, with those for the previous category of oil spills. An eventual change of the location of vessel 2 from port 5 (Terneuzen) to port 2 (Terschelling) produces a change also of the expected damage, to $2.1169 \cdot 10^{-4}$.

Vessel 4 (Hein) has the optimum location in port 5 (Terneuzen), which is not too far (only 2.7 sailing hours) from the port where its arms are located. If we change its location to the port of its arms, there is no change in the value of the expected



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (4\ 5\ 2\ 5\ 6\ 1\ 6\ 3\ 6\ 3)$

Figure 6.15: Location of the cleaning vessels from the second group, for the second category of oil spills

damage; hence, a new optimum location vector is found, namely: (4 5 2 3 6 1 6 3 6 3)

For the vessel 5 (Cornelia) there is no considerable change from the actual location (from IJmuiden to Den Helder), only 2 sailing hours far. But if we keep the actual location for this vessel, the same value of the expected damage is found, namely the optimum value. Also, if we locate this vessel in the port where its arms are (Rotterdam), the value of the expected damage in equal to the optimum value. So, for this vessel there are three possibilities to locate it such that the expected damage has the optimum value.

Comparing with the actual location, vessel 6 (Gateway) has an optimum location closer to the arms location. But, still, if we keep it in the actual location, no change in the value of the expected damage would be found. The same situation is for vessels 7 (Geopotes), 8 (HAM 316) and 9 (Lesse).

Vessel 10 (Waker) is very slowly and this can explain the fact that its optimum location is in an area where the spills are more probably to appear close to the coast, as in the area of Rotterdam.

In order to combine the optimum solutions formed for the two groups of vessels, we use the same method as for the previous category of oil spills: we keep the first four vessels in the optimum positions found for the first group and we locate the rest of the vessels in ports found for the second group of vessels. In this way, we obtain the location vector: (3 5 2 5 6 1 6 3 6 3), for which the expected damage is $1.9663 \cdot 10^{-4}$ (6% larger than the optimum value).

The third category of oil spills

In the third category of oil spills are included those spills that have the volume of oil between 150 and 750 m^3, with an average size of 424.1 m^3. They are more frequently in the North Sea (0.015155 per year) comparing with the spills from the first two categories.

The optimum value of the expected damage in case that only first four vessels are considered is $3.0469 \cdot 10^{-3}$ and it is obtained for the (optimum) location vectors: (3 5 2 3) and (3 5 2 6). This brings an improvement with 20% compared with the value of the expected damage for the actual location of the cleaning vessels ($3.8 \cdot 10^{-3}$).

We have to remark that the optimum locations for the first three vessels coincide with their optimum locations for the second category of oil spills. Also, in these two optimum vectors, only the position of the last vessel is different; moreover, wherever the last vessel is located, the value of the expected damage is the same and it is equal to the optimum value. We have to remind that the same situation has been met for the previous two categories of the oil spills.

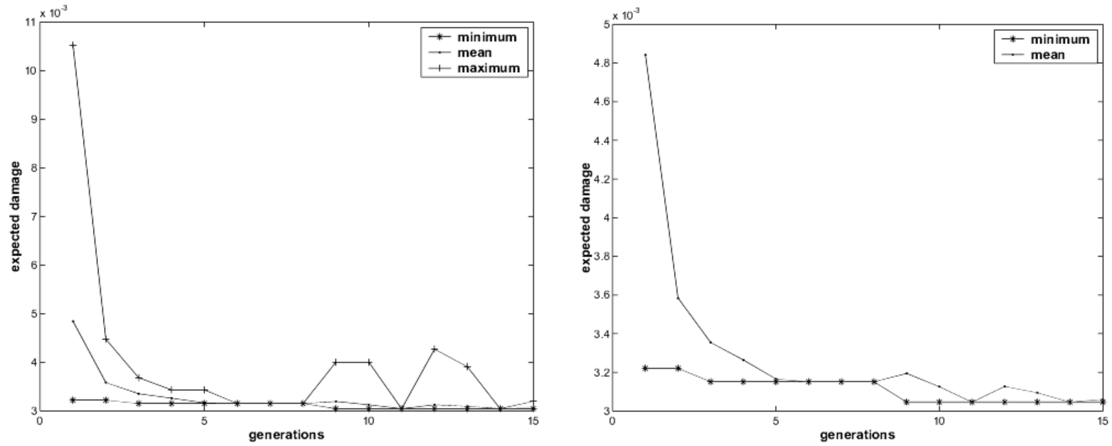
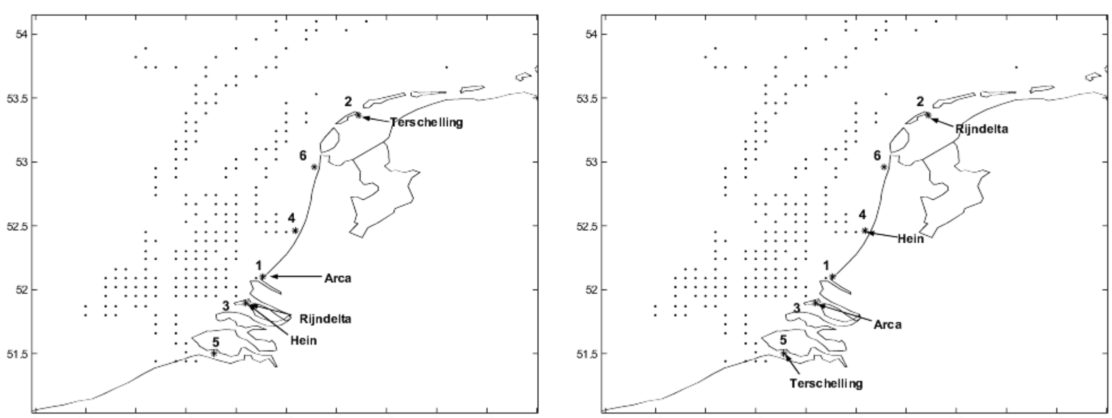


Figure 6.16: Convergence of the genetic algorithms for the third category of oil spills, first group of vessels



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 5\ 2\ 6)$

Figure 6.17: Location of the cleaning vessels from the first group, for the third category of oil spills

For the second group of vessels, the optimum value found using genetic algorithms is $3.352 \cdot 10^{-3}$, which is with 17% better then the actual value of the expected damage. For the optimum solutions, there are six locations vectors. They have in common only few components, namely the locations of the first three vessels, of the eight vessel and of the last vessel.

$$\begin{pmatrix}5 & 3 & 2 & 3 & 1 & 6 & 1 & 2 & 4 & 4\end{pmatrix}$$
$$\begin{pmatrix}5 & 3 & 2 & 3 & 1 & 2 & 2 & 2 & 4 & 4\end{pmatrix}$$
$$\begin{pmatrix}5 & 3 & 2 & 3 & 4 & 2 & 2 & 2 & 4 & 4\end{pmatrix}$$
$$\begin{pmatrix}5 & 3 & 2 & 3 & 4 & 2 & 2 & 2 & 5 & 4\end{pmatrix}$$
$$\begin{pmatrix}5 & 3 & 2 & 3 & 5 & 2 & 2 & 2 & 5 & 4\end{pmatrix}$$
$$\begin{pmatrix}5 & 3 & 2 & 6 & 6 & 2 & 2 & 2 & 1 & 4\end{pmatrix}$$

For the rest of the vessels it is quite difficult to analyze their optimum locations, but it can be seen that for the fourth vessel the predominant location is port 3, for the sixth and seventh vessels, the predominant location is port 2. For the vessels 5 and 9, the optimum location can be any of the locations 1, 4 and 5.

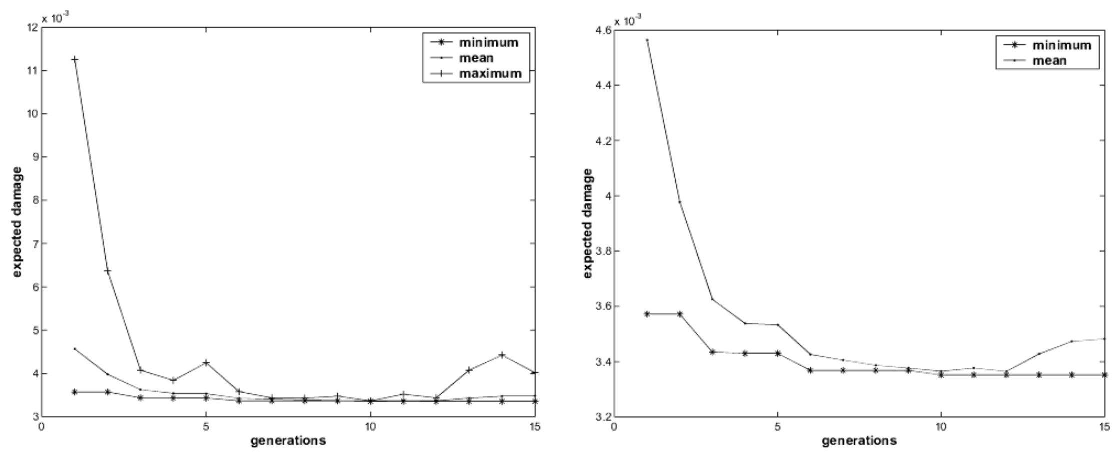
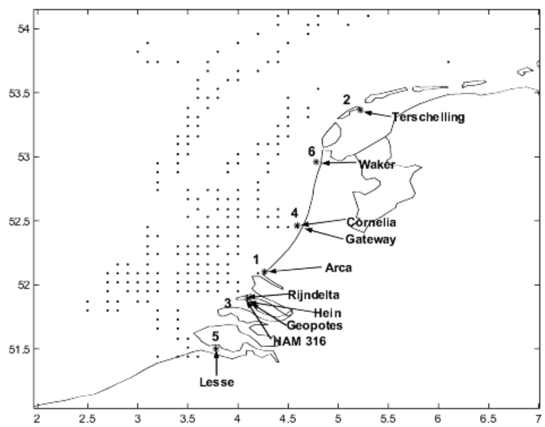
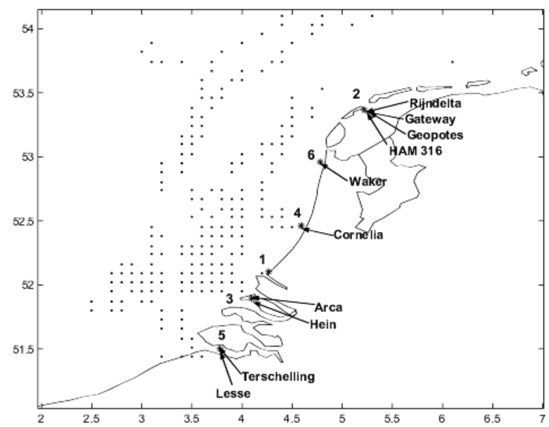


Figure 6.18: Convergence of the genetic algorithms for the third category of oil spills, second group of vessels



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$



(b) optimum location:
$v = (3\ 5\ 2\ 3\ 4\ 2\ 2\ 2\ 5\ 4)$

Figure 6.19: Location of the cleaning vessels from the second group, for the third category of oil spills

Taking account of the actual location of the cleaning vessels, we can combine all the optimum solutions in a single location vector, namely: (5 3 2 3 4 2 2 2 5 4) for which the expected damage has the value $3.352 \cdot 10^{-3}$, equal to the optimum value.

If we would like to combine the optimum location vectors for the two groups of vessels as we did for the previous categories of oil spills, we obtain the vector (3 5 2 3 4 2 2 2 5 4), for which the value of the expected damage is $3.241 \cdot 10^{-3}$, even smaller than the value found by the genetic algorithms. Hence, this is an example for which the genetic algorithms falls in a local optimum.

The fourth category of oil spills

The spills form the fourth category ($750\ m^3 < V < 3000\ m^3$) are more frequently (0.024028 per year) than the spills from the first three categories. The mean size of a spill from this category is 1736.3 m^3.

The optimum location vector for the first group of vessels found by the genetic algorithms are:

$$\begin{pmatrix}3 & 5 & 2 & 2\end{pmatrix}$$
$$\begin{pmatrix}3 & 5 & 2 & 3\end{pmatrix}$$
$$\begin{pmatrix}3 & 5 & 2 & 1\end{pmatrix}$$

with an optimum value of the expected damage 0.0146, with 18% better than the actual value of the expected damage (0.0178).

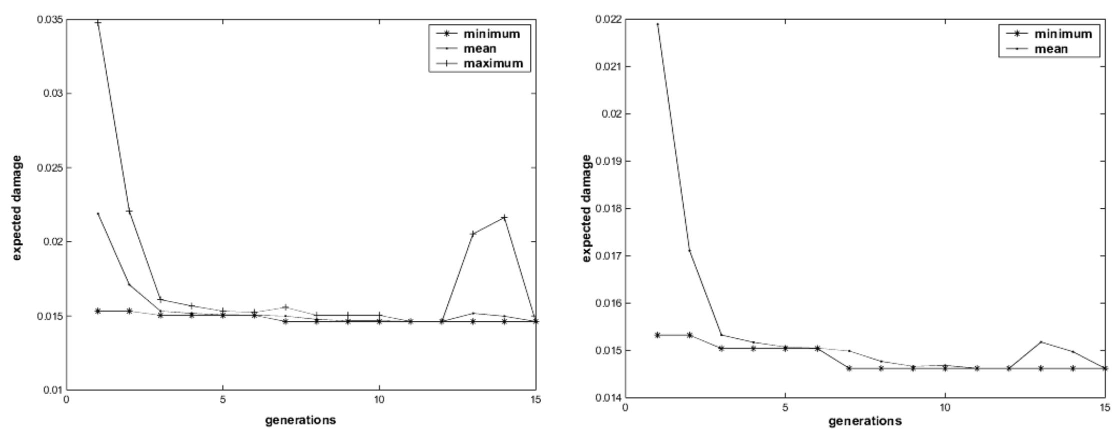
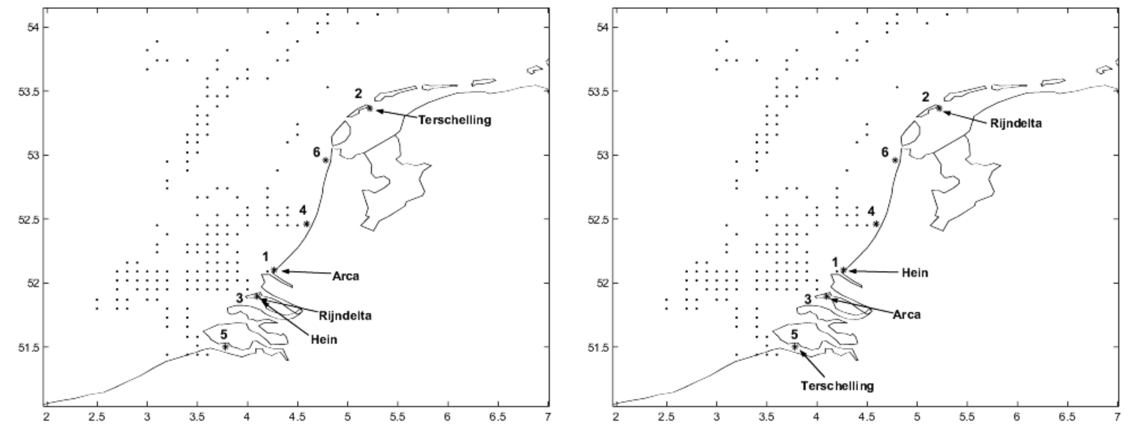


Figure 6.20: Convergence of the genetic algorithms for the fourth category of oil spills, first group of vessels

It is interesting to remark that the optimum vectors are very similar to the optimum solutions found for the previous three categories. Moreover, if we try to move the fourth vessel (Hein) in any of the six harbors, we obtain the same value for the expected damage. Hence, again, for the first group of vessels, the optimum location vectors has the form: (3 5 2 $*$).



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 5\ 2\ 1)$

Figure 6.21: Location of the cleaning vessels from the first group, for the fourth category of oil spills

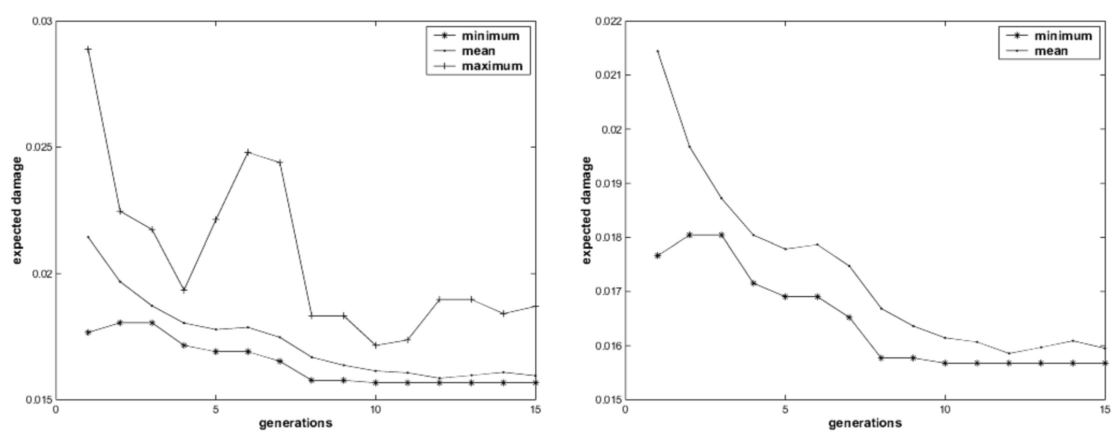


Figure 6.22: Convergence of the genetic algorithms for the fourth category of oil spills, second group of vessels

The first vessel (Arca) is not far from its actual location; more then this, it is located in an area where the frequency of the oil spills is large. Since Arca is one of the most effective vessels, it seems to be a reasonable optimum location. If we try to move it in port 1 (Scheveninghen), where it is located now, we obtain the value of the expected damage equal to 0.0152, with 4% larger than the optimum value.

For vessel 2 (Terschelling), the optimum position is again in an area where the spills appear more frequently, but, at the same time, very far from the harbor where its arms are located. We can try to move it to other places, but other location doesn't bring any improvement.

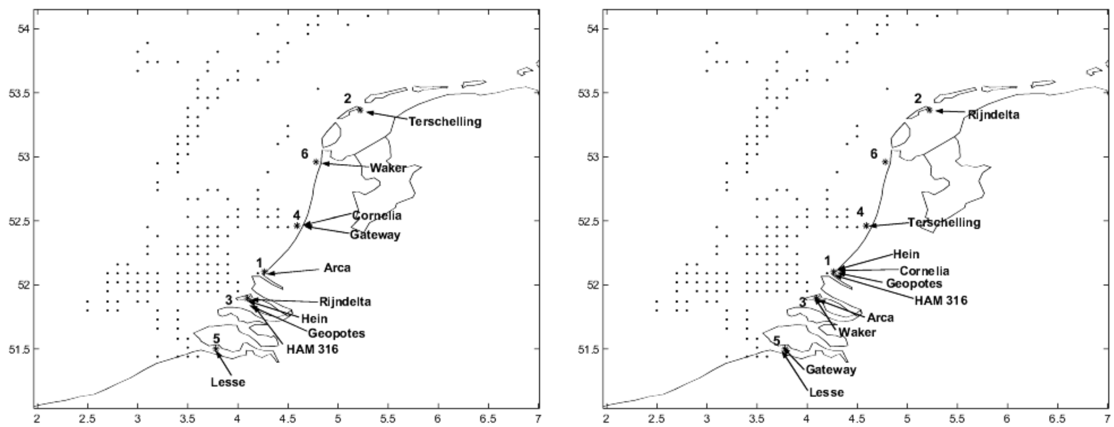
For the second group of vessels, the genetic algorithms end up with four optimum solutions:

$$\begin{array}{c}(3 \quad 4 \quad 2 \quad 6 \quad 1 \quad 2 \quad 1 \quad 4 \quad 5 \quad 3)\\(3 \quad 4 \quad 2 \quad 6 \quad 1 \quad 2 \quad 2 \quad 4 \quad 5 \quad 3)\\(3 \quad 4 \quad 2 \quad 1 \quad 5 \quad 5 \quad 2 \quad 1 \quad 1 \quad 3)\\(3 \quad 4 \quad 2 \quad 1 \quad 5 \quad 4 \quad 2 \quad 1 \quad 1 \quad 3)\end{array}$$

for which the (optimum) value of the expected damage is 0.015678, 27% better than its value for the actual location of the cleaning vessels (0.021237).

The four optimum solutions have few common elements. The locations of the first three vessels and of the last vessel are the same. More then this, the location of the first and the third vessels coincide with their optimum locations obtain for the first group of vessels. If we try to locate the second vessel at port 5, as its optimum location was for the first group, the expected damage is the same.

If we try to combine all these four optimum solutions, we have to take into account where the arms are located and if there are more possibilities for the optimum location, we have to choose that one which is closer to the arms location. In this way, we obtain the location vector (3 4 2 1 1 5 1 1 5 3) for which the expected damage has the value 0.0156, equal to the optimum value.



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (3\ 4\ 2\ 1\ 1\ 5\ 1\ 1\ 5\ 3)$

Figure 6.23: Location of the cleaning vessels from the second group, for the fourth category of oil spills

The fifth category of oil spills

In this category are included the spills that have a volume between 3000 and 10000 m^3, which are more frequently (0.024399 per year). The average spill size is equal to 5917.5 m^3.

For the first group of vessels, the optimum solutions bring a reduction with 15% comparing with the actual value of the expected damage. The optimum solutions are very similar to the optimum solutions found for the previous categories of oil spills. They are: (3 5 2 2) and (3 5 2 4), with a general form (3 5 2 $*$).

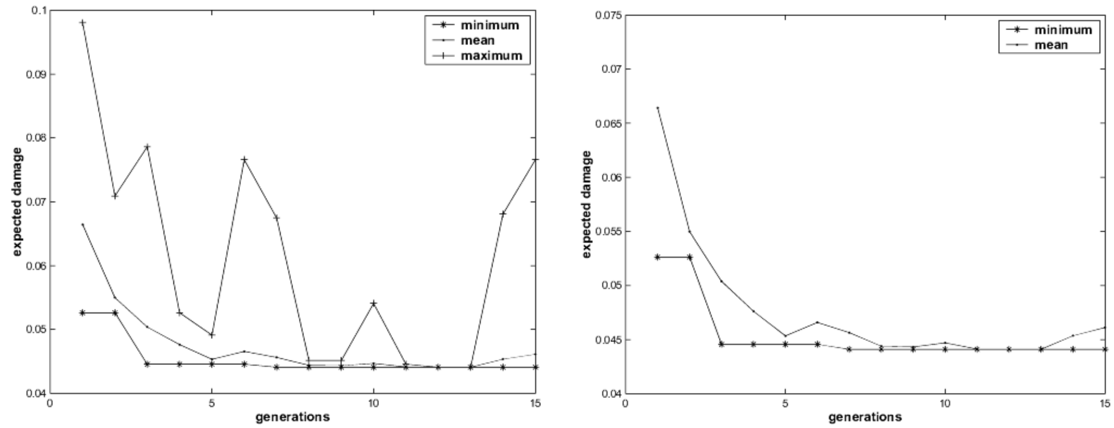
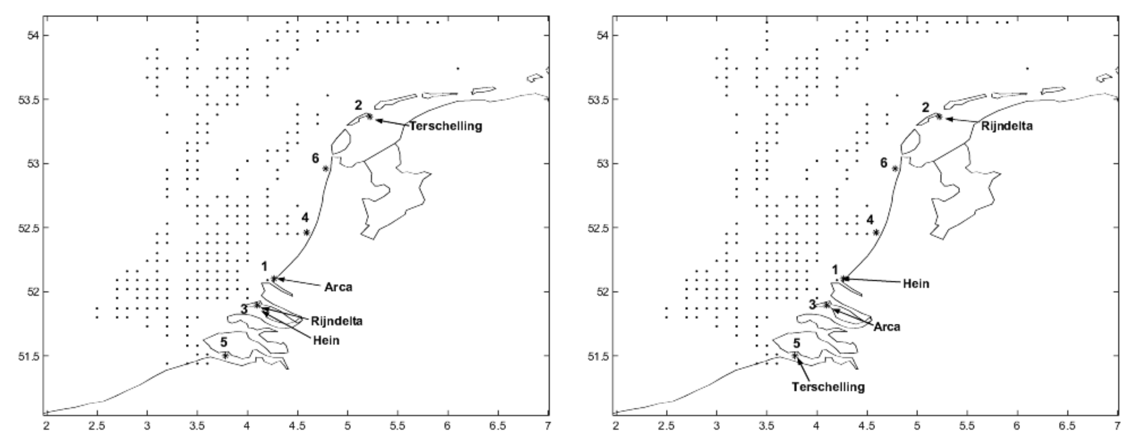


Figure 6.24: Convergence of the genetic algorithms for the fifth category of oil spills, first group of vessels



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 5\ 2\ 1)$

Figure 6.25: Location of the cleaning vessels from the first group, for the fifth category of oil spills

For the second group of vessels, the genetic algorithms end up with four optimum solutions:

$$\begin{array}{c} (3 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 3 \quad 3) \\ (3 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 6 \quad 3) \\ (3 \quad 2 \quad 2 \quad 1 \quad 5 \quad 1 \quad 1 \quad 2 \quad 3 \quad 3) \\ (3 \quad 2 \quad 2 \quad 1 \quad 5 \quad 1 \quad 1 \quad 6 \quad 6 \quad 3) \end{array}$$

for which the (optimum) value of the expected damage is 0.043389, 37% better than its actual value (0.068089). It is easy to observe that the four optimum solutions have many similarities; for all of them the locations of the first four vessels, of vessel 6 and 7 and of the last vessel are the same. For each of the rest of the vessels, there are only two possibilities to be located in order to obtain the optimum expected damage.

If we take account of the location of the arms for choosing between these optimum possibilities, we find the location vector (3 2 2 1 1 1 1 6 3 3), for which the value of the expected damage is 0.043388, equal to the optimum value.

It is easy to remark that spills from this category appear in north of the Dutch part of the Continental Shelf. This is in accordance with the optimum location vectors, in which port 2 (Terschelling) appears more times then for the previous categories. Also, port 5 (Terneuzen) doesn't appear too much, because there are only few spills close to it and these can be cleaned by the vessels from port 1 (Scheveningen) and 3 (Rotterdam). Anyway, most of the vessels are concentrated around ports Rotterdam and Scheveningen.

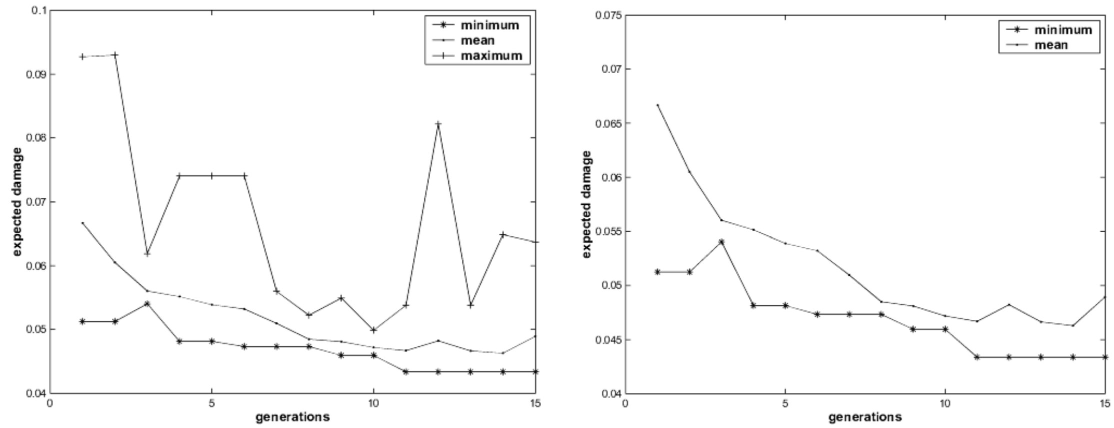
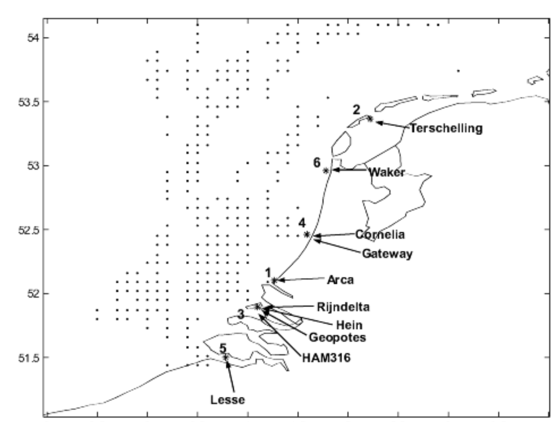
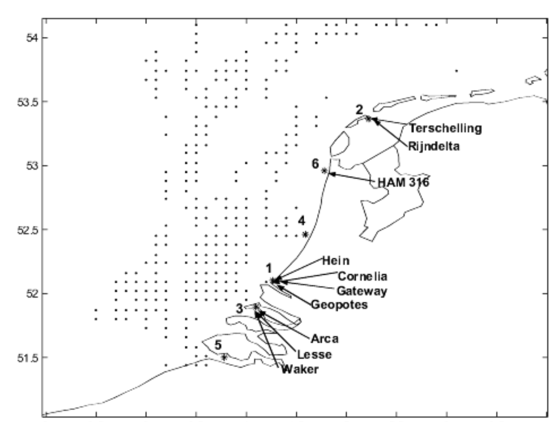


Figure 6.26: Convergence of the genetic algorithms for the fifth category of oil spills, second group of vessels



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$



(b) optimum location:
$v = (3\ 2\ 2\ 1\ 1\ 1\ 1\ 6\ 3\ 3)$

Figure 6.27: Location of the cleaning vessels from the second group, for the fifth category of oil spills

The sixth category of oil spills

For the spills from the sixth category, the volume is between 10000 and 30000 m^3, with a mean spill size of 1696.5 m^3. They are less frequently than the spills from the previous category (0.016209 m^3 per year).

The optimum solution for the first group of vessels reduces with 18% the value of the expected damage, compared with its actual value. The solutions found by the genetic algorithms are (3 3 2 4) and (3 3 2 2); the location of the fourth vessel in any other port doesn't change the optimum value of the expected damage (0.0702).

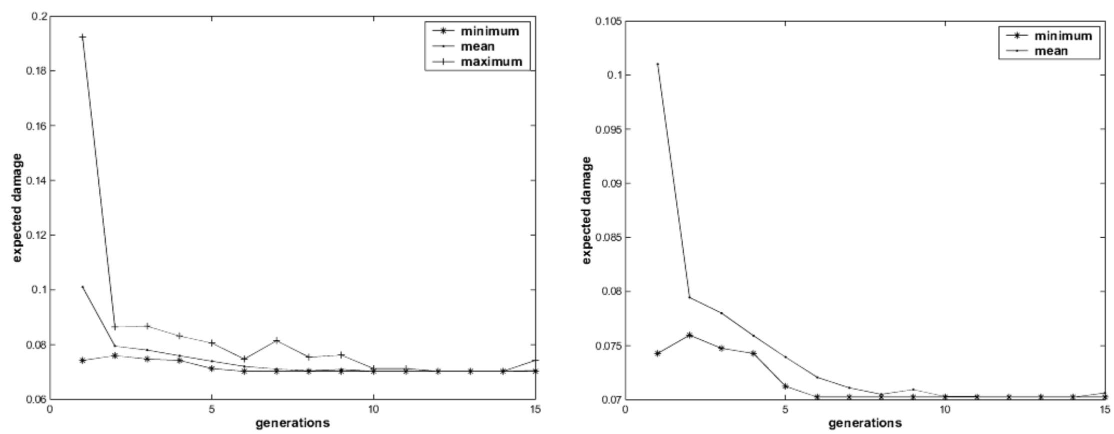
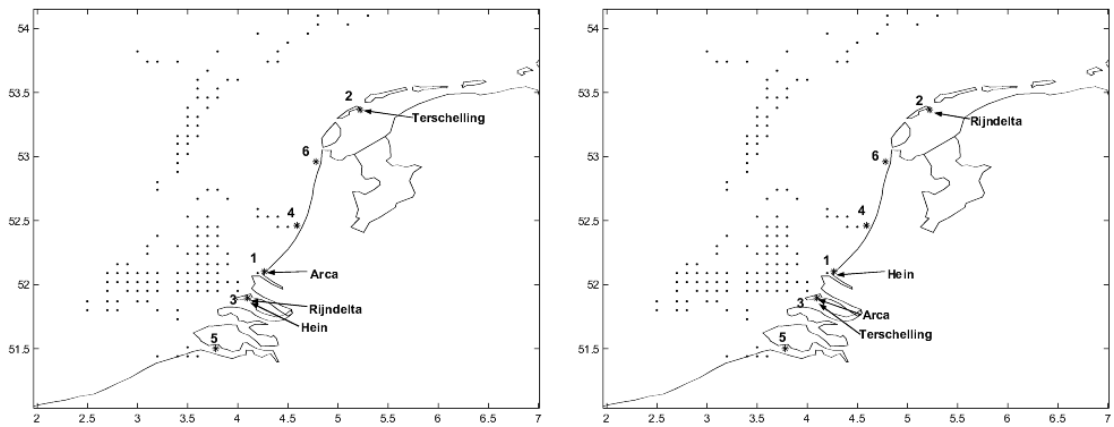


Figure 6.28: Convergence of the genetic algorithms for the sixth category of oil spills, first group of vessels

The first and the third vessels (Arca and Rijndelta) have the same optimum location as for the previous category. The second vessel is optimum located in harbor 3 (Rotterdam) and not in harbor 5 (Terneuzen), as before. As we can be seen in figure (6.29), most of the spills from this category appear in the area of harbors Rotterdam and Scheveningen, and this explain the fact that the optimum location is also in port Rotterdam for two out of four vessels.

For the second group of vessels the predominant locations in the six optimum solutions are also 1 and 3. The optimum value of the expected damage is 0.069722



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 3\ 2\ 1)$

Figure 6.29: Location of the cleaning vessels from the first group, for the sixth category of oil spills

(40% better than its actual value) and the optimum location vectors are:

$$\begin{array}{l}(1\quad 3\quad 2\quad 5\quad 4\quad 2\quad 4\quad 6\quad 1\quad 1)\\(1\quad 3\quad 2\quad 5\quad 4\quad 2\quad 3\quad 6\quad 1\quad 1)\\(1\quad 3\quad 2\quad 5\quad 4\quad 4\quad 3\quad 2\quad 1\quad 1)\\(1\quad 3\quad 2\quad 5\quad 3\quad 2\quad 3\quad 2\quad 1\quad 1)\\(1\quad 3\quad 2\quad 1\quad 4\quad 4\quad 3\quad 2\quad 3\quad 1)\\(1\quad 3\quad 2\quad 1\quad 3\quad 4\quad 3\quad 5\quad 3\quad 1)\end{array}$$

For these six vectors there are only few similarities: only the first three vessels and the last vessel have the same location in all six vectors.

If we try to change vessel 1 (Arca) from port 1 to port 3, where its optimum location was found for the first group of vessels, an increment with 1% in the value of the expected damage appears.

Vessel 4 is located in port 1 (Scheveningen) or port 5 (Terneuzen), so in an area where the frequency of the oil spills is larger. If we move it in port 3 (Rotterdam), which is in the same area, the value of the expected damage in not changed.

The optimum location of vessel 5 (Cornelia) is either port 3 or port 4, that, in fact, are the ports in which its arms are located, respectively, its actual location is.

The sixth vessel (Gateway) has the optimum location in port 2 or in port 4, but if we have to choose between them, we would choose port 4, because it is closer to the harbors where the arms of this vessels are located.

For the seventh vessel (Geopotes), the predominant optimum location is port 3, which coincides with both the actual location and the arms location for this vessel.

For vessel 8 (HAM 316) there are three possibilities for the optimum location: two of them (port 2 and port 6) in the north part of the coast and port 5 in the south part. But, the fact that its arms are located in Rotterdam explains why port 5 (Terneuzen) is considered an optimum location. If we try to move this vessel in Rotterdam, the value of the expected damage doesn't change.

Vessel 9 (Lesse) is located close to its arms location and also in the area with the most of the spills from this categories. For this vessel, we have to choose between ports 1 and 3.

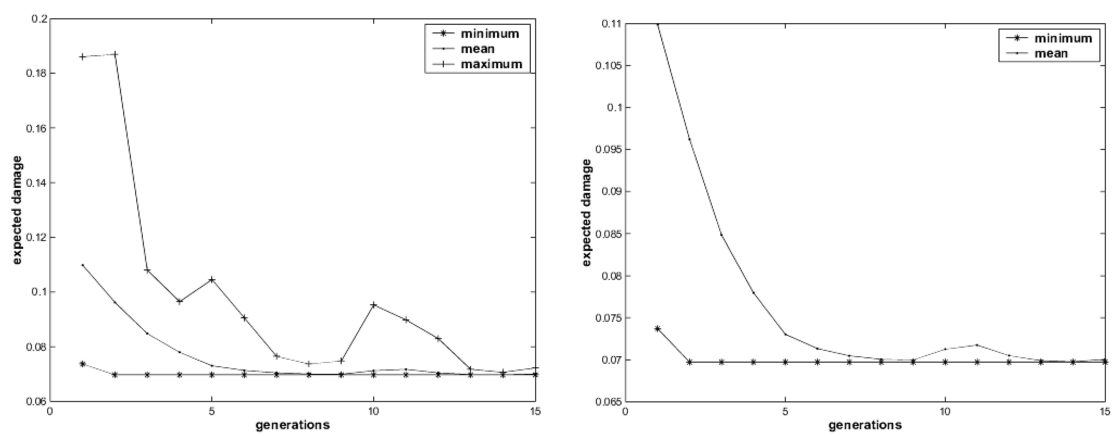
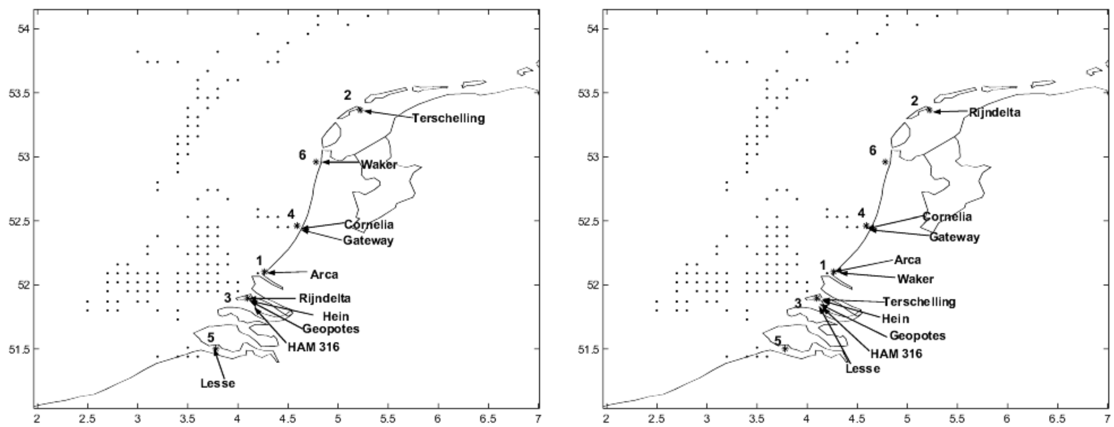


Figure 6.30: Convergence of the genetic algorithms for the sixth category of oil spills, second group of vessels



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (1\ 3\ 2\ 3\ 4\ 4\ 3\ 3\ 3\ 1)$

Figure 6.31: Location of the cleaning vessels from the second group, for the sixth category of oil spills

Trying to combine all six optimum location vectors, and taking account on the comments that we have made, we obtain the location vector (1 3 2 3 4 4 3 3 3 1), for which the expected damage is equal to the optimum value.

The seventh category of oil spills

The spills from this category have the volume between 30000 and 100000 m^3, with an average spill size equal to 57920.7 m^3. They appear more frequently in the area of ports Rotterdam and Scheveningen, and only few in the north part of the coast. The optimum solutions found for the first group of vessels is in accordance with this remark. For the three location vectors, (3 3 2 6), (3 3 2 1) and (3 3 2 2), the first two vessels are located in this area. They improve with 21% the value of the expected damage (from 0.25361, the actual value, to 0.2024, the optimum value).

There are several characteristics of these optimum solutions that we comment upon. First of all, the optimum locations of the first three vessels coincide with their optimum locations found for the previous categories. Moreover, wherever the fourth vessel is located, the value of the expected damage doesn't change.

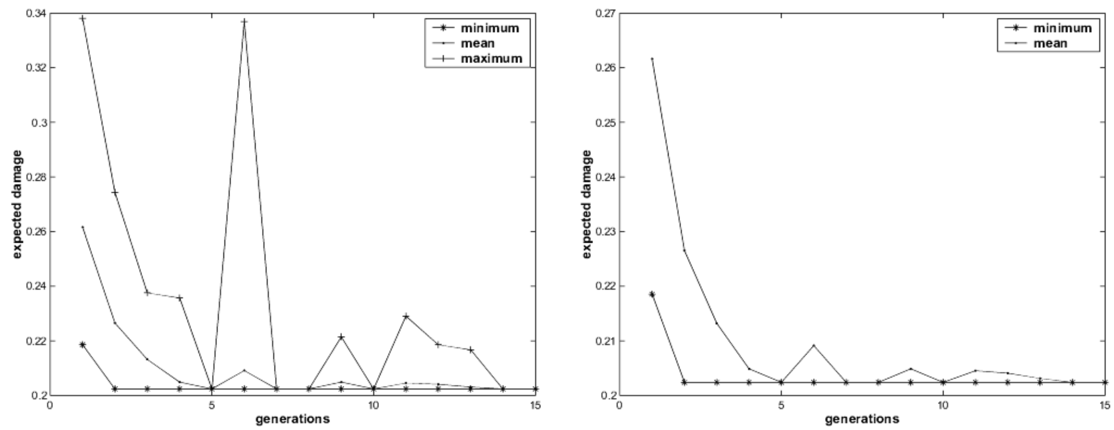


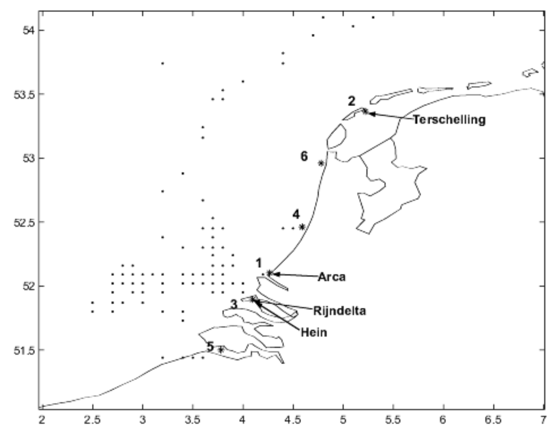
Figure 6.32: Convergence of the genetic algorithms for the seventh category of oil spills, first group of vessels

If we look, also, at the optimum location vector found for the second group of vessels, (3 3 2 4 3 1 3 2 3 3), we observe that the positions of the first four vessels are the same with their positions for the first group of vessels.

We remark, also, that the optimum location for the vessel four coincides with its actual location, vessels 5 and 7 are located in the ports in which their arms are. For vessel 4, the optimum location (port 1) is closer to the arms location (port 3) than the port in which it is now.

Vessel 8 (HAM 316) is moved in port 2 (Terschelling), very far from its arms. If we change its location in port 3, we obtain an expected damage equal to the optimum value.

Vessel 9 (Lesse) is moved from port 5, which coincides also with the port where its arms are located, to port 3. If we try to keep it in port 5 (Terneuzen), the value of the expected damage will not change.



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 3\ 2\ 1)$

Figure 6.33: Location of the cleaning vessels from the first group, for the seventh category of oil spills

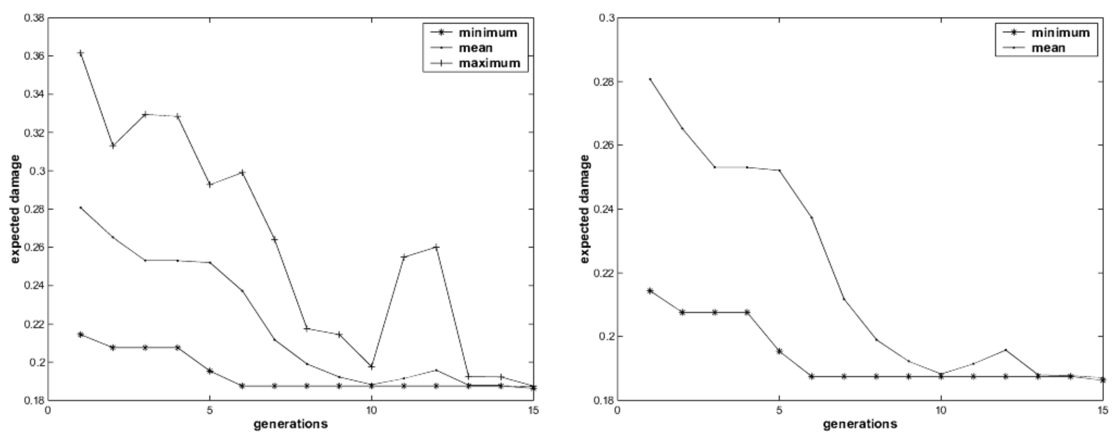
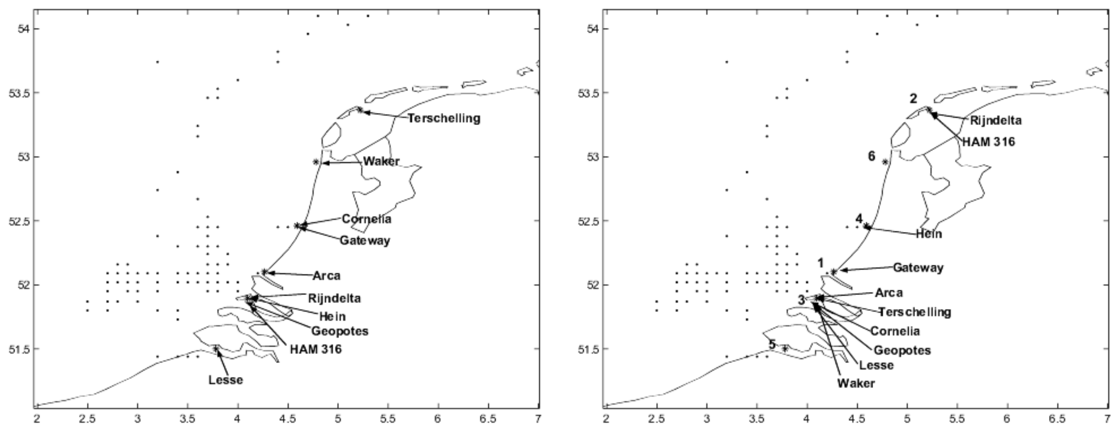


Figure 6.34: Convergence of the genetic algorithms for the seventh category of oil spills, second group of vessels



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (3\ 3\ 2\ 4\ 3\ 1\ 3\ 2\ 3\ 3)$

Figure 6.35: Location of the cleaning vessels from the second group, for the seventh category of oil spills

The eighth category of oil spills

The last of the categories of oil spills includes the spills that have the volume larger than 100000 m^3, with an average spill size equal to 186605 m^3. The spills from this category appear with a frequency of 0.006835 per year. We have to remark that this frequency is larger than the frequency of the spills from the first two categories, hence than the spills that have a volume less than 150 m^3, but this can be explained by the fact that many of the small spills are not registered.

The optimum location for the first four vessels are: (3 3 2 1), (3 3 2 2), (3 3 2 5), for which the value of the expected damage is equal to 0.5407, 20% better than its value for the actual location (0.6747). Again we have to remark that the general form for the optimum location vector is (3 3 2 $*$) and this coincides with the form of the optimum vectors found for categories 6 and 7 of oil spills, but differs from those found for the rest of the categories by the position of the second vessel. If we try to move this vessel to obtain the general form (3 5 2 $*$), the value of the expected damage increases with 10%.

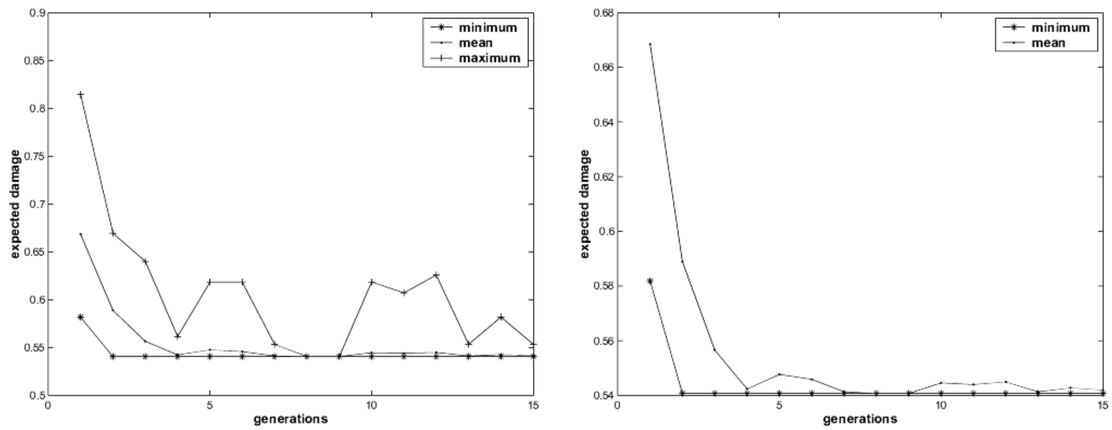


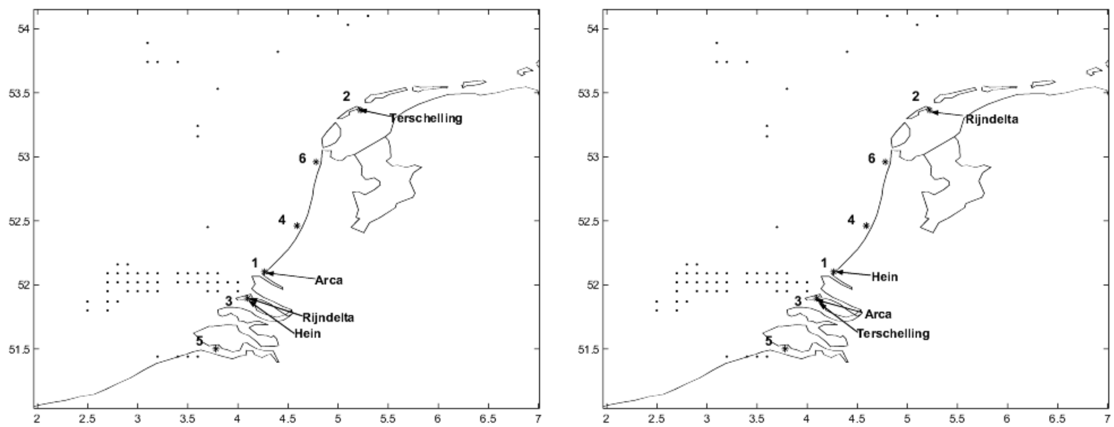
Figure 6.36: Convergence of the genetic algorithms for the eighth category of oil spills, first group of vessels

For the second group of vessels there are nine optimum solutions found by the genetic algorithms. They are:

$$\begin{array}{c}
(3\ \ 3\ \ 2\ \ 3\ \ 4\ \ 3\ \ 3\ \ 6\ \ 4\ \ 3)\\
(3\ \ 3\ \ 3\ \ 3\ \ 4\ \ 3\ \ 3\ \ 6\ \ 4\ \ 3)\\
(3\ \ 3\ \ 3\ \ 3\ \ 4\ \ 3\ \ 3\ \ 6\ \ 6\ \ 3)\\
(3\ \ 3\ \ 3\ \ 4\ \ 4\ \ 3\ \ 4\ \ 6\ \ 5\ \ 3)\\
(3\ \ 3\ \ 3\ \ 1\ \ 4\ \ 3\ \ 4\ \ 6\ \ 5\ \ 3)\\
(3\ \ 3\ \ 3\ \ 1\ \ 6\ \ 3\ \ 4\ \ 1\ \ 5\ \ 3)\\
(3\ \ 3\ \ 3\ \ 4\ \ 6\ \ 3\ \ 4\ \ 1\ \ 5\ \ 3)\\
(3\ \ 3\ \ 3\ \ 4\ \ 6\ \ 3\ \ 4\ \ 5\ \ 5\ \ 3)\\
(3\ \ 3\ \ 3\ \ 4\ \ 6\ \ 3\ \ 4\ \ 3\ \ 3\ \ 3)
\end{array}$$

and the value of the expected damage for them is 0.45331, 35% better than the actual value of the expected damage (0.69127).

In all these optimum vectors we can find few similarities. The first two vessels, the sixth vessel and the last one have the same location in all nine vectors; more then



(a) actual location: $v = (1\ 2\ 3\ 3)$ (b) optimum location: $v = (3\ 3\ 2\ 1)$

Figure 6.37: Location of the cleaning vessels from the first group, for the eighth category of oil spills

this, all these vessels are located in port 3 (Rotterdam). For the other vessels, there are more possibilities to locate them in an optimum way.

First of all it is interesting to remark that only the first optimum vectors (3 3 2 3 4 3 3 6 4 3) includes in it the optimum location found for the first group of vessels (3 3 2 $*$). Hence, if we would choose one of these vectors, probably the first one will be the most convenient.

We can see in figure (6.37) that the majority of the spills from this category appear in the area of ports Rotterdam an Scheveningen, and only few in the north part of the coast. This explains the fact that the predominant location in these optimum vectors is port 3 (Rotterdam).

For vessel 4 (Hein), the optimum location can be any of the ports 1, 3 or 4, but since its actual location is port 3, probably this would be a good choice. For vessel 5 (Cornelia), for an optimum value of the expected damage we can choose between the harbors 4 and 6, but since it is located at this moment in port 4 and, moreover, port 4 is closer to its arms location, this is what we would choose.

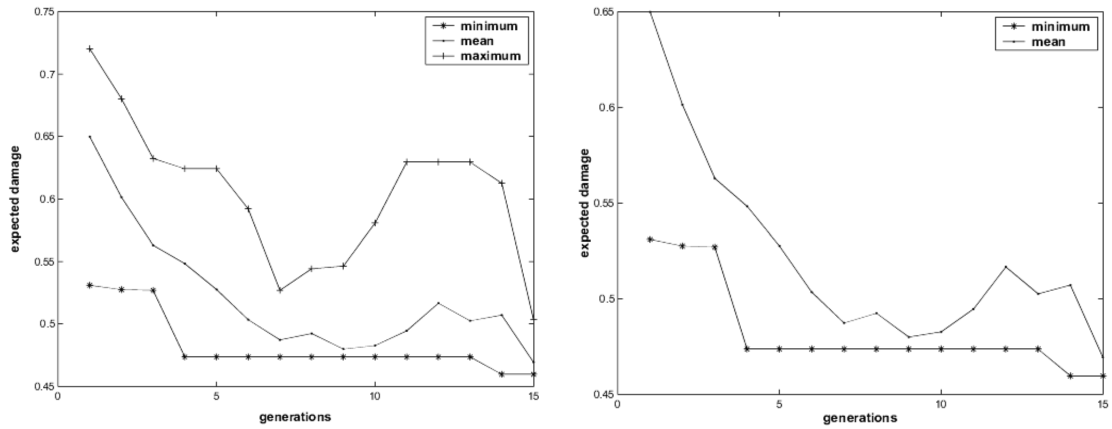
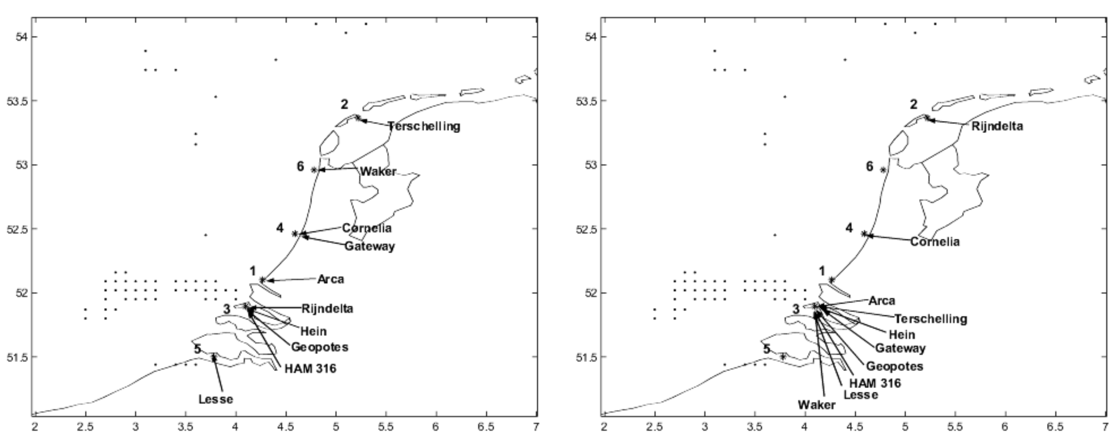


Figure 6.38: Convergence of the genetic algorithms for the eighth category of oil spills, second group of vessels



(a) actual location:
$v = (1\ 2\ 3\ 3\ 4\ 4\ 3\ 3\ 5\ 6)$

(b) optimum location:
$v = (3\ 3\ 2\ 3\ 4\ 3\ 3\ 3\ 3\ 3)$

Figure 6.39: Location of the cleaning vessels from the second group, for the eighth category of oil spills

Vessel 6 (Gateway) is located at this moment in port 4, but its optimum location is port 3, where its arms are located as well. If we want to see what happens if we keep this vessel in its actual location, we have to compute the expected damage for the vector: (3 3 2 3 4 4 3 6 4 3); the result is equal to the optimum value of the expected damage.

The actual location and the arms location for vessel 7 (Geopotes) are in port 3 (Rotterdam), which appears also in possible optimum locations for this vessel. The same situation is for vessel 8 (HAM 316) and vessel 9 (Lesse).

Taking account of these remarks, if we try to combine all the nine optimum vectors found, we obtain a location vector (3 3 2 3 4 3 3 3 3 3) for which the expected damage has a value equal to the optimum value.

6.3 Conclusions

All the optimum vectors that we have found for each category of oil spills bring reduction of the value of the expected damage, if we compare with the damage corresponding with the actual location. But, each of them is good for one category of oil spills. Of course, it would be better if we can find only one vector which is as good as possible for all spills, from all categories. This seems to be possible if more data about the spills is available, namely, the average size of a spill in each cell and the probability to have a spill in that cell, but not divided by categories. Since we do not have this kind of data, we will try to find a location vector which is better than the actual one for all categories of oil spills.

For vessels that are owned by RIKZ, the optimum vector has one of the forms: $(1\ 5\ 2\ *)$, $(3\ 5\ 2\ *)$ or $(3\ 3\ 2\ *)$. It is clear that the location of vessel Hein is not important if only these four vessels are taken into consideration.

For all ten vessels is difficult to find only one location vector which is optimal for all eight categories. Probably all vessels will be used for large spills and this leads us to consider the location vectors obtained for the categories 6, 7 and 8. Even considering only these three categories is difficult to find a single location vector which reduces the actual value of the expected damage.

Taking account of the remarks that we have made for each category of oil spills separately, we end up with three location vectors for the last three categories. They are:

$$(3\quad 3\quad 2\quad 3\quad 4\quad 4\quad 3\quad 3\quad 3\quad 1)$$
$$(3\quad 3\quad 2\quad 4\quad 3\quad 1\quad 3\quad 2\quad 3\quad 3)$$
$$(3\quad 3\quad 2\quad 3\quad 4\quad 3\quad 3\quad 3\quad 3\quad 3)$$

If we look at the similarities that these vectors have, we can conclude that a good vector for the large spills have to have the form: $(3\ 3\ 2 * * * 3 * 3 *)$. For the rest of the vessels there are only two possibilities for each of them. The final decision, which for sure has to take into consideration many other aspects, needs to choose between these possibilities.

Chapter 7

Conclusions

The response at sea in case of oil pollution is a topic for which many studies are made and is, at the same time, a very difficult topis. For a good analysis there are several things that have to be very well defined.

First of all, the damage produced by an oil spill at sea is a very important item that has to be clarified when we start to optimize the response in case of an emergency situation. It has to be defined very clearly and then to be quantified using data. Most of the studies include both economical and ecological aspects of the damage. For these kind of studies, a common unit for the damage has to be found. In general, money is chosen; this implies a very 'sensitive' measure of the ecological loss.

Another item that has to be taken into consideration are the contingency maps. The zones from a maritime area are not equal vulnerable. It is known that a pollution in an area close to a beach or to a harbor produces more economical damage than one which appears far from the coast.

The oil transport has to be included in this kind of studies, taking account of the weather conditions and of the maritime currents. It is known that there are differences between the behavior of the spills from different types of oil. Hence, for a good analysis, it is important also to take account of the type of oil.

All these things make a study of this kind complex and difficult. That is why, for this thesis, a more simple analysis was carried out.

Basic assumptions in our simplified analysis are that we have introduced a damage function which depends upon the area of pollution only, not trying to classify all economical consequences in detail and that we have assumed that the vessels wait and start cleaning at the arrival time of the slowest one. Another important assumption is that in our model we have not taken into account the transport of the pollution.

For several categories of oil spills and two strategies of exploring the cleaning vessels, we have computed optimum location vectors, presenting the optimum harbors for the location of the cleaning vessels. For this purpose we have used a genetic algorithm which has been evaluated with respect to its performance and its sensitivity to change in some of the steering parameters. The genetic algorithm has been found to perform satisfactory. The optimum location vectors found differ a little between the studied categories of oil spills which impedes us drawing a straightforward definite conclusion. However, clear trends have been signaled leading to an average reduction of 25% of the total damage if we compare it with the damage resulting from a cleaning operation based upon the present location of vessels. Thus it is considerable and therefore it is recommended to extend this study in order to enhance its realism.

There are still possibilities to improve the analysis from this thesis, if we have more detailed data. We have to remaind that the optimum location vectors have been found for each of the eight categories separately. Of course that we would like to have an optimum location for all categories. For this, we need other incidents data, namely, for each cell, the probability of pollution (but not separately for each category) and, what is more important, an average size of a spill in each cell. What we had was an average size for each category and we considered that the spill has this size in each cell.

A sensitivity map can be included also in this model. Weights can be given for damage for each cell, such that the more sensitive cells can have a larger influence on the total expected damage.

The way in which the vessel work in a cleanup operation can be improved also, such that they do not have to wait for the slowest one. Also, limitations regarding the vessels that are involved in a cleanup operation, taking account the number, or the arrival time to the polluted area, or the cleaning capacity, can be included.

All these thinks can make our study to be more realistic. But, even without them, it is important to notice that the optimum solutions found using genetic algorithms are better then the actual location vectors. For the case in which there are more optimum solutions, other considerations can help to decide which of them can be chosen.

Once more, this study proves that the pollution at sea is a taft subject which demands much work and energy to be clarified.

Appendix

The expected damage $D(c, v)$

```
% the damage in cell c, if the location vector is v...
      % and we use only one vessel

function  dam1=D1A(c,v) global V; global DistLocPort; global data;

global Vess; global K1; global vess; global port; global n;

clear T; tempVess=vess; for i=1:size(vess,1)
    if vess(i,4)==0
        tempVess(i,4)=v(i);
    end;
end;
% initialization
i1=1;use=[i1];

% the arrival time
T=tempVess(use,3)+dist(port(v(use),:),port(tempVess(use,4),:))/...
tempVess(use,1)+DistLocPort(c,tempVess(use,4))/tempVess(use,1);
T_ar=max(T); qq=tempVess(i,2);
dam1=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;

for i1=2:n
    use=[i1];
    T=tempVess(use,3)+dist(port(v(use),:),port(tempVess(use,4),:))/...
    tempVess(use,1)+DistLocPort(c,tempVess(use,4))/tempVess(use,1);
    T_ar=max(T);
    qq=tempVess(i,2);
    poss_dam=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;
```

```
    if dam1>poss_dam
        dam1=poss_dam;
    end;
end;

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

% the damage function in cell c if the location vector is v
          % and we use only 2 vessels

function dam2=D2A(c,v)

global Vess; global K1; global vess; global port; global n;

clear T; tempVess=vess;
for i=1:size(vess,1)
    if vess(i,4)==0
        tempVess(i,4)=v(i);
    end;
end;
% initialization
i1=1; i2=2; use=[i1 i2];
T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';

T_ar=max(T); q=sum(vess(use(:),2));
dam2=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/q;

comb2=nchoosek(1:n,2); for i1=1:size(comb2,1)
  use=comb2(i1,:);
  T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
  tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';
  T_ar=max(T);
  q=sum(vess(use(:),2));
  poss_dam=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/q;
  if dam2>poss_dam
     dam2=poss_dam;
  end;
end;
```

```
%-----------------------------------------------------------------

% the damage in cell c if the location vector is v
        % and we use 3 vessels

function dam3=D3A(c,v)

global Vess; global K1; global vess; global port; global n;

clear T; tempVess=vess; for i=1:size(vess,1)
    if vess(i,4)==0
        tempVess(i,4)=v(i);
    end;
end;
% initialization
i1=1; i2=2; i3=3; use=[i1 i2 i3];
T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';
T_ar=max(T); qq=sum(vess(use(:),2));
dam3=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;
comb3=nchoosek(1:n,3); for i1=1:size(comb3,1)
  use=comb3(i1,:);
  T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
  tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';
  T_ar=max(T);
  qq=sum(vess(use(:),2));
  poss_dam=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;
  if dam3>poss_dam
     dam3=poss_dam;
  end;
end;

%--------------------------------------------------------------------
...

% the damage in cell c if the location vector is v and we use 9 vessels

function dam9=D9A(c,v)

global Vess; global K1; global vess; global port; global n;
```

```
clear T; tempVess=vess;
for i=1:size(vess,1)
    if vess(i,4)==0
        tempVess(i,4)=v(i);
    end;
end;
% initialization
i1=1;i2=2;i3=3;i4=4;i5=5;i6=6;i7=7;i8=8;i9=9;

use=[i1 i2 i3 i4 i5i6 i7 i8 i9];
T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';
T_ar=max(T); qq=sum(vess(use(:),2));
dam9=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;
comb9=nchoosek(1:n,9); for i1=1:size(comb9,1)
  use=comb9(i1,:);
  T=tempVess(use,3)'+dist(port(v(use),:),port(tempVess(use,4),:))/...
  tempVess(use,1)+DistLocPort(c,tempVess(use,4))./tempVess(use,1)';
  T_ar=max(T);
  qq=sum(vess(use(:),2));
  poss_dam=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;
  if dam9>poss_dam
     dam9=poss_dam;
  end;
end;

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

% the damage in cell c if the location vector is v and we use 10 vessels

function dam10=D10(c,v,dist)

global data; global Vess; global K1; global vess; global port;

% initialization
i1=1;i2=2;i3=3;i4=4;i5=5;i6=6;i7=7;i8=8;i9=9;i10=10;

use=[i1 i2 i3 i4 i5 i6 i7 i8 i9 i10];
T=vess(use,3)'+dist(c,v(use))./vess(use);
```

```
qq=sum(vess(use(:),2));
T_ar=max(T);
dam10=(2/3)*sqrt(2*K1)*V^(2/3)*T_ar^(3/2)+K1*V^(4/3)*T_ar/qq;

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

% damage in cell c if the location vector is v

function dam=D(c,v)

global group; global arm; global data;
if arm==0 % the arms location is not considered
    if group==1
        rez=[D1(c,v);D2(c,v);D3(c,v);D4(c,v)];
    else
        rez=[D1(c,v);D2(c,v);D3(c,v);D4(c,v);...
                D5(c,v);D6(c,v);D7(c,v);D8(c,v);...
                D9(c,v);D10(c,v)];
    end;
else % the arms location is considered
   if group==1
        rez=[D1A(c,v);D2A(c,v);D3A(c,v);D4A(c,v)];
    else
        rez=[D1A(c,v);D2A(c,v);D3A(c,v);D4A(c,v);...
                D5A(c,v);D6A(c,v);D7A(c,v);D8A(c,v);...
                D9A(c,v);D10A(c,v)];
    end;
end;
dam=min(rez);

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

% expected damage (the evaluation function) if the location vector is v

function I=expect_damage(v)

global p; global data;
I=0;
for c=1:length(p)
    if p(c)~=0
```

```
        I=I+p(c)*D(c,v);
    end;
end;
```

The initial population

```
% initial generation (population)

function pop=initialize(pop_size);

global data;global m; global n;
pop=zeros(pop_size,n);
% generate a pop_size X n matrix with integer numbers from [1,m]
pop=unidrnd(m,pop_size,n);
for i=1:pop_size
    pop(i,n+1)=expect_damage(pop(i,:));
end;
```

The crossover operation

```
% crossover
function [c1,c2,rr]=simpleXover(p1,p2)
numVar=size(p1,2)-1; % get the number of variables
% generate randomly the cutting point
cPoint=unidrnd(numVar-1);
c1=[p1(1:cPoint) p2(cPoint+1:numVar)];
c2=[p2(1:cPoint) p1(cPoint+1:numVar)];
```

The mutation operation

```
% mutation
function [parent, r]=simpleMutate(parent,gen)

global m;
r=unidrnd(m);% generate the new value of the gene
parent(gen)=r;
```

Selection

```
% selection
function newPop=selection(oldPop,pop_size,q,gen)

global data;
ee=size(oldPop,2); % nr. colowmns
nn=size(oldPop,1); % nr. rows
newPop=zeros(nn,ee); % allocate space for return population
fitt=zeros(nn,1); % allocate space for probability of selection
x=zeros(nn,2); % sorted list of rank and index
x(:,1)=[1:nn]'; % to know what element it was
[y x(:,2)]=sort(oldPop(:,ee));

% nonlinear function
r=q/(1-(1-q)^nn);
fitt(x(:,2))=r*(1-q).^(x(:,1)-1);% nonlinear function

fit=cumsum(fitt);
rNums=sort(rand(nn,1)); % generate nn sorted random numbers
fitIn=1; newIn=1;
while newIn<=nn
    if (rNums(newIn)<fit(fitIn))
        newPop(newIn,:)=oldPop(fitIn,:);
        newIn=newIn+1;
    else
        fitIn=fitIn+1;
    end;
end;
```

The genetic algorithm (the main program)

```
clc;clear; time=cputime; input_data;
fid=fopen('Solution.txt','a');
fprintf(fid,'Category of oil spill%2.0f\n',categ);
fprintf(fid,'Group of vessels:%2.0f\n',group);
fprintf(fid,'Population size:%3.0f\n',pop_size);
fprintf(fid,'Number of generations:%3.0f\n',max_gen);
fprintf(fid,'\n');

% supply a population P0 of pop_size individuals and ...
```

```
     %respective function values
startPop=initialize(pop_size);
xZomeLength=size(startPop,2);% length of xZome=numVar+fitness
numVar=xZomeLength-1; % number of variables
endPop=zeros(pop_size,xZomeLength); % a secondary population matrix
c1=zeros(1,xZomeLength); c2=zeros(1,xZomeLength);

[oval oindex]=min(startPop(:,xZomeLength)); % best value
opt_position=startPop(oindex,(1:numVar)); % optimum solution
gen=0; % current generation

while gen<max_gen
    gen=gen+1;
    endPop=startPop;
    [bval,bindex]=min(startPop(:,xZomeLength));% keep the best solution
    best=startPop(bindex,(1:numVar));
    traceInfo(gen,1)=gen;
    traceInfo(gen,2)=min(startPop(:,xZomeLength));
    traceInfo(gen,3)=mean(startPop(:,xZomeLength));
    traceInfo(gen,4)=max(startPop(:,xZomeLength));
    if oval>bval % if we have a new best solution
        oval=bval; % update the best val
        opt_position=best; % update the best solution
    elseif oval==bval
        index=0;
        for j=1:size(opt_position,1)
            if best==opt_position(j,:)
                index=1;
            end;
        end;
        if index==0
            opt_position=[opt_position;best];
            opt_position
        end;
    end;

   % crossover
    for kk=1:(pop_size/2)
       rr=rand;
       if rr<probXover
```

```
        a=endPop(2*kk-1,:);
        b=endPop(2*kk,:);
        [c1 c2 rr]=simpleXover(a,b);
        if c1(1:numVar)==a(1:numVar)
            c1(xZomeLength)=a(xZomeLength);
        elseif c1(1:numVar)==b(1:numVar)
            c1(xZomeLength)=b(xZomeLength);
        else c1(xZomeLength)=expect_damage(c1(1:numVar));
        end;

        if c2(1:numVar)==a(1:numVar)
            c2(xZomeLength)=a(xZomeLength);
        elseif c2(1:numVar)==b(1:numVar)
            c2(xZomeLength)=b(xZomeLength);
        else c2(xZomeLength)=expect_damage(c2(1:numVar));
        end;

        endPop(2*kk-1,:)=c1;
        endPop(2*kk,:)=c2;
        end;
end;

% mutation
randMutate=[];bitNum=[];chromNum=[];bitPos=[];
for i=1:pop_size*numVar
    rr=rand;
    if rr<probMutate
        randMutate=[randMutate rr];
        bitPos=[bitPos i];
        gn=mod(i,numVar);
        if gn==0
            gn=numVar;
        end;
        bitNum=[bitNum gn];
        chrom=(i-gn)/numVar+1;
        chromNum=[chromNum chrom];
        parent=endPop(chrom,:);
        [c1 aleat]=simpleMutate(parent,gn);
        % make sure we created a new solution before evaluating
        if c1(1:numVar)==endPop(chrom,(1:numVar))
```

```
                c1(xZomeLength)=endPop(chrom,xZomeLength);
            else  c1(xZomeLength)=expect_damage(c1(1:numVar));
            end;
            endPop(chrom,:)=c1;
            end;
    end;

    % selection
    startPop =selection(endPop,pop_size,q,gen); % for rank selection
   [bval,bindex]=min(startPop(:,xZomeLength));% keep the best solution
   best=startPop(bindex,(1:numVar));
   if oval>bval % if we have a new best solution
       oval=bval; % update the best val
       opt_position=best; % update the best solution
   elseif oval==bval
       index=0;
       for j=1:size(opt_position,1)
           if best==opt_position(j,:)
               index=1;
           end;
       end;
       if index==0
           opt_position=[opt_position;best];
           opt_position
       end;
   end;
end;
traceInfo(gen,1)=gen;
traceInfo(gen,2)=min(startPop(:,xZomeLength));
traceInfo(gen,3)=mean(startPop(:,xZomeLength));
traceInfo(gen,4)=max(startPop(:,xZomeLength));
x=opt_position;xx=oval;
time=cputime-time;time=time/60; %(minutes)
fprintf(fid,'Optimum value is:%5.4f\n',xx);
fprintf(fid,'\n'); fprintf(fid,'Optimum solution is:\n'); if
group==1
    for i=1:size(x,1)
        fprintf(fid,'%2.0f  %2.0f  %2.0f  %2.0f\n',x(i,:));
    end;
else
```

```
 for i=1:size(x,1)
fprintf(fid,'%2.0f  %2.0f  %2.0f  %2.0f  %2.0f  %2.0f  %2.0f ...
             %2.0f  %2.0f  %2.0f\n',x(i,:));
 end;
end;
fprintf(fid,'Running time is (in minutes):%6.2f\n',time);
fprintf(fid,'\n');
fclose(fid);

figure(1)
plot(traceInfo(:,1),traceInfo(:,2),'r*-'); hold on;
plot(traceInfo(:,1),traceInfo(:,3),'b.-'); hold on;
plot(traceInfo(:,1),traceInfo(:,4),'g+-'); hold off;
figure(2)
plot(traceInfo(:,1),traceInfo(:,2),'r*-'); hold on;
plot(traceInfo(:,1),traceInfo(:,3),'b.-'); hold on;
```

Bibliography

[1] W. Banzhaf, *Genetic algorithms: an introduction on the automatic evolution of computer programs and its applications*, San Francisco: Morgan Kaufmann, 1998

[2] R. Cerf, “Asymtotic convergence of genetic algorithms”, *Advanced Applied Probabilities*, **30**, pp 521-550, 1998

[3] R. Cerf “Genetic algorithms and Freidlin-Wentzell theory” Proceedings of the 10th International Conference on Applied Stochasitc Models and Data Analysis (ASMDA), Compiegne, France, Juin 2001

[4] O. Francois “Global optimization with exploration/selection algorithms and simulated annealing”, *The Annals of Applied Probability*, **12**, pp 248-271, 2002

[5] M. Gen and R. Cheng *Genetic algorithms and engineering design*, New York: Wiley, 1997

[6] D. Goldberg *Genetic algorithms in search, optimization and maschine learning*, Addison-Wesley, 1989

[7] J.H. Holland *Adaptation in natural and artificial systems*, The University of Michigan Press, 1975

[8] C. Houck, J. Joines, M. Kay, A genetic algorithm for function optimization: a Matlab implementation, http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/, last access: August 1, 2003

[9] Y. Rabinovich and A. Wigderson, “Techinques for bounding the convergence rate of genetic algorithms”, *Random structures algorithms*, **2**, pp 111-138, 1999