NTNU – Trondheim
Norwegian University of
Science and Technology

Computational User Support Tool for Decision Making Processes

Caroline Bauge Gulliksen

Master of Energy and Environmental Engineering
Submission date: June 2013
Supervisor: Carlos Alberto Dorao, EPT

Norwegian University of Science and Technology
Department of Energy and Process Engineering

NTNU

Norwegian University
of Science and Technology

Department of Energy
and Process Engineering

EPT-M-2013-44

MASTER THESIS

for

Stud.techn Caroline Bauge Gulliksen
Spring 2013

Computational user support tool for decision making process

Datastyrt brukerstøtteverktøy for beslutningsprosesser

Background and objective

Selecting a particular engineering solution from a set of possibilities is not a simple task. Several factors can affect the view of the decision maker regarding which is the optimal solution. However it is difficult for the decision maker to control the complexity of the process. For this reason tools like the Decission Support Problem Techniques have been developed for helping designers. This technique provides the support and a rationale for using human judgment in design synthesis. Multiple objectives that are quantified using analysis-based "hard" and insight-based "soft" information can be modeled in the DSPs.

The main objective of this work will consist in developing a decision maker support system based on the Decision Support Problem Technique for generic engineering problems. The tool will be tested with selected cases.

The following tasks are to be considered:

1 Literature review on DSP
2 Design of the computational tools, and identification of critical algorithms
3 Prototyping of the system on suitable software
4 Testing of the system with selected cases

-- " --

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

☒ Work to be done in lab (Water power lab, Fluids engineering lab, Thermal engineering lab)
☐ Field work

Department of Energy and Process Engineering, 14. January 2013

Olav Bolland
Department Head

Carlos A. Dorao
Academic Supervisor

Research Advisor: Luis Castillo

Abstract

A computational tool to perform calculations for the Decision-Support Problem (DSP) Technique is developed to aid decision-makers in evaluating technologies suited for design problems. The DSP Technique is presented in a step-by-step manner and an object-oriented analysis is performed to convert the steps of the DSP Technique to a computer system. The object-oriented analysis uses Unified Modeling Language (UML) to identify and visualize the system structure. From the object-oriented analysis, the DSP computational tool is developed in C++. A prototype user interface is created in Objective-C as an application for iPhone on the iOS platform to guide the user through the steps of the DSP Technique.

The program is tested in two selected cases. The first case is the selection of a cryogenic separation process for CO_2 removal from a natural gas field initially containing 30 mol% CO_2. The Controlled Freeze Zone process is selected as the most suited process after performing the DSP Technique to evaluate the available solutions. In the second case, the DSP Technique is performed to evaluate solutions for a liquefaction process for an offshore production vessel for LNG (Liquefied Natural Gas) production. The results from the computational tool does not provide a clear solution for the most suited liquefaction process, due to small differences in merits between the feasible alternatives.

From testing the computational tool for the selected cases, it is concluded that, by letting a computer perform calculations for the DSP Technique, the method becomes more efficient and less time consuming. The program performs as expected and provides the user with correct results from the input given.

Sammendrag

Et brukerstøtteverktøy for å utføre beregningene i DSP (Decision-Support Problem) metoden er utviklet for å støtte beslutningstaker i evaluering av teknologier egnet til designprosjekter. DSP metoden blir gjennomgått steg for steg og en objektsorientert analyse er utført for å omforme stegene i DSP metoden til et dataprogram. Den objektsorienterte analysen tar i bruk UML (Unified Modeling Language) for å identifisere og visualisere strukturen til programmet. Brukerstøtteverktøyet for DSP metoden blir så utviklet i programmeringsspråket C++. Et prototype brukergrensesnitt er utviklet til programmet for å veilede brukeren gjennom stegene i DSP metoden. Brukergrensesnittet er utviklet i programmeringsspråket Objektiv-C som en applikasjon for iPhone.

Programmet er testet for utvelgelse av en kryogenisk separasjonsprosess for CO_2 rensing av naturgass som inneholder 30 mol% CO_2. Prosessen Controlled Freeze Zone blir valgt ut som det beste alternativet etter bruk av DSP programmet som er utviklet. Programmet er også testet for å finne den beste kjøleprosessen for LNG (Liquefied Natural Gas) produksjon på en FPSO (Floating Production Storage and Offloading). Det fremkommer ingen klare resultater for den best egnede kjøleprosessen på grunn av små forskjeller i resultatene mellom alternativene.

Fra testing av programmet konkluderes det med, at ved å la et dataprogram gjennomføre beregningene i DSP metoden, blir metoden mer effektiv og tar kortere tid å gjennomføre. Programmet utfører beregningene som forventet og gir brukeren korrekte resultater fra den informasjon som er gitt til programmet.

Acknowledgements

I would like to express my gratitude to my supervisor during this master thesis, Professor Carlos Dorao at NTNU. He has inspired me to learn new things and has always been a good support. I would also like to thank Luis Castillo for support with the Decision-Support Problem Technique.

Contents

List of Figures

List of Tables

1 Introduction

Selecting an engineering solution for a particular project means choosing the most acceptable alternative from multiple possible solutions. This selection process can be both complex and time consuming. By standardizing a method for evaluating and comparing different solutions, the decision-making process can be made more efficient, thereby minimizing the resources put into this phase. The Decision-Support Problem (DSP) Technique has been developed so that structured decision-making can be carried out. The DSP Technique makes use of both objective, science-based information and human judgement to take advantage of engineers' experiences in their respective fields [16]. In the DSP method, all available technologies for a specified design problem are evaluated and compared with each other with respect to a set of criteria that are defined by the decision-maker. The DSP Technique can be categorized into three areas of decision-making processes. These are Selection DSP, Compromise DSP and Coupled/Hierarchial DSP. The Selection DSP is used when alternatives are compared based on a set of criteria to yield a preferable solution. The Compromise DSP is used when feasible alternatives are improved through modification to satisfy constraints of a design problem. The Coupled or Hierarchical DSP is used for a combination of the Selection DSP and the Compromise DSP [20].

In this thesis, a computational tool for the Selection DSP is developed. The Selection DSP method is based on a model of a real-life system. The model is created as a set of criteria reflecting a real case, in which technologies are evaluated and compared. By making a realistic model of the design project, an ultimate and unique solution is rarely found, but several sufficient alternatives can be identified for the design case. Engineering projects are multidisciplinary and complex, and optimal solutions are seldom found [16]. The DSP Selection method offers engineers to find acceptable solutions based on selected criteria of a design project. The main goal of the Selection DSP method is to compare and accept technologies that are ”good enough” for a particular case, and reject those that are not. By evaluating the available solutions with respect to these criteria, the DSP method helps engineers justify the most acceptable solution.

The DSP method provides a more efficient method for early phase decision-making in an engineering project by creating a method for structured evaluation and comparison of available technologies[16]. The method transforms human experience and judgment to quantitative information that can be compared, and used together with objective, science-based information. In this way the decision-maker can uncover the most acceptable technologies. Calculations must be performed to normalize both the soft information and the objective information so that they can be processed together and integrated. These calculations are time consuming and provide an uncertainty factor related to human error. A computational tool for calculating and integrating the information may increase the efficiency and reliability of the DSP method. The decision-maker is responsible for providing information regarding the evaluation of the technologies and criteria, while a computer is responsible for calculations providing merits of technologies to find the best suited technology for a given engineering project.

A basic computational program has been developed at the Department of Mechanical Engineering at the University of Houston, Texas as a part of their graduate program [16]. The computational tool is called DSIDES (Decision Support In the Design of Engineering Systems), and is used to solve calculations needed in the DSP Selection method. DSIDES has successfully been used in the conceptual design of aircraft tires, airplanes, ships and composite materials [19].

1.1 Goal and outline

The main goal of this thesis is to create a computational tool for the Decision-Support Problem (DSP) Technique. The computational tool will perform calculations used in the DSP method to identify the best suited technologies for the conceptual design of an engineering project. By developing a generic computational tool, the DSP method may be less time consuming to perform. Human errors in calculations may also be minimized. The DSP method will be presented in a step-by-step manner and an object-oriented analysis is performed to identify the structure of the computer program. The computer program is prototyped using the C++ programming language. The program receives input from a user and performs the necessary calculations to compare technologies with respect to the given criteria, and provides the user with results from the calculations. A user interface is developed to guide the decision-maker through the DSP process. The user interface is made as an application for the iPhone and is written in Objective-C.

The system is tested with two case studies to report the performance of the computational tool. The first case study consists of choosing the best technology for removal of CO_2 from a highly sour natural gas stream at an offshore location by use of cryogenic separation processes.

The second case study is a selection of a liquefaction process for LNG production on a FPSO (Floating Production Storage and Offloading).

1.2 Scope

The DSP Technique uses both objective, science-based information and soft, experience based information to support decision-making. The quality of the information gathered by the decision-maker is important for the accuracy of the results in the DSP. The decision-maker's knowledge of the design problem is crucial to provide a good model for a real-life system, where relevant criteria are identified. The computational tool in this thesis handles the Selection DSP method, which can be divided into a Preliminary Selection DSP and a Selection DSP. The Preliminary Selection DSP is used in early phase conceptual design where soft information is available. The Selection DSP is used in later stages of a engineering design problem where both soft and objective information may be available. The computational tool does not provide support for the Compromise DSP and the Coupled/Hierarchial DSP.

1.3 Structure of thesis

Chapter 2 provides a detailed description of the Decision-Support Problem (DSP) Technique in a step-by-step manner. An object-oriented analysis of the DSP is carried out in Chapter 3 to identify the structure of the computational tool. In Chapter 4, a prototype of the program is developed in the C++ programming language. Chapter 5 presents a user interface created for the computational tool. The program is tested with two selected case studies in Chapter 6. The conclusion of the thesis is presented in Chapter 7 with suggestions for further work. In the Appendices, detailed data of the programming code for the DSP program are given. Detailed results from the case studies are also presented.

2 The Decision-Support Problem (DSP) Technique

The Decision-Support Problem (DSP) Technique is a method for evaluating a set of possible solutions for a specified design problem to find the best available technology. The DSP Technique has been developed on the basis that information not only comes from scientific principles, but from human judgment and experience as well [16]. The technique also reflects that for complex, multi disciplinary problems, there is not necessarily only one optimal solution, but rather several ”good enough” solutions. The DSP Technique provides help to identify these solutions and is also used to rank the feasible alternatives to find the best one [16].

The DSP method is categorized into three types of decisions. These are Selection, Compromise and Coupled/Hierarchial. This thesis will focus on the Selection DSP. The Selection DSP can be divided into a Preliminary Selection DSP and a Selection DSP. The Preliminary Selection DSP is used to determine the alternatives most-likely to succeed for a specified design problem. The Preliminary Selection DSP uses soft information to rate the performance of the identified, possible solutions and can be useful in early stages of a project where available objective information is limited. The most-likely to succeed alternatives that are identified in the Preliminary Selection DSP, are further evaluated in a Selection DSP, where the result yields a preference-based ranking of the feasible alternatives [20]. The Selection DSP uses both soft and objective information and is often relevant when more information about the design problem and alternatives is identified.

The DSP method is described in detail to determine the critical steps of the decision-making process.

2.1 The Preliminary Selection DSP

The Preliminary Selection DSP makes use of soft, experience-based information to obtain a sample of most-likely to succeed alternatives in a design problem. A set of concepts are chosen from all available technologies. The relevant criteria for the design problem are identified and weighted with respect to their importance. The decision-maker must gather information about how concepts influence and perform with given criteria. Based on gathered information, concepts are compared on basis of identified criteria, with respect to a chosen concept, denoted as datum. This is used to rank and score the concepts to obtain a small set of feasible alternatives that will be relevant for decision-maker in finding the best alternative for the design. The process of the Preliminary Selection DSP is given as follows [20]:

Given A set of concepts

Identify The criteria influencing the decision and the relative importance of each criterion

Capture Soft, experience-based information about the concepts with respect to the selected criteria

Rank The concepts are ranked based on the criteria and their relative importance

The Preliminary Selection DSP can be set up in a step-by-step manner. The steps are summarized in Table 1 and then presented in detail [13].

Table 1: Steps in the Preliminary Selection DSP

Step 1	Describe the concepts and provide acronyms
Step 2	Describe each generalized criterion, provide acronyms and weighting constants
Step 3	Choose a datum with which all other concepts will be compared
Step 4	Compare the concepts
Step 5	Evaluate the merit function for each concept within each generalized criterion
Step 6	Include interactions between generalized criteria
Step 7	Post-solution sensitivity analysis: determine the most-likely to succeed concepts

Step 1: Describe the concepts and provide acronyms.
All available concepts are identified and described in detail. Main characteristics, advantages and disadvantages are listed in the concept description. The concepts are given acronyms for simplification.

Step 2: Describe each generalized criterion, provide acronyms and weighting constants for the specified criteria.
The criteria that are relevant for the design problem are identified and described. For simplification, the criteria are given acronyms and weighting constants to compensate for their relative importance. The relative importance of the different criteria is based on user experience, and may affect the quality of the Preliminary Selection DSP. Different scenarios of the criteria's relative importance can be created to reveal the consequences of the weighting constants.

Step 3: Choose an initial datum.
One concept is chosen as the initial datum to which the other concepts are compared, as done in step 4. This process is repeated using all concepts as datums, so there is no specific rule as to which concept is chosen as the initial datum. It may be the concept that is perceived to be one of the most-likely to succeed concepts, or it may be the most controversial concept.

Step 4: Compare the concepts.
The concepts are compared to the datum with respect to the generalized criteria. The concepts are given scores as to how they perform compared to the datum. The scoring system is given in Table 2 [20].

Table 2: Evaluation method in Preliminary Selection DSP

0	The criteria of the datum is given the value 0. The concepts that score the same as the datum are given the same value.
+1	The concepts that are better than the datum on the specific criteria are given the value +1.
-1	The concepts that perform worse than the datum on the selected criteria are given the value -1.

The scoring of concepts is performed in turn using all concepts as datums.

Step 5: Evaluate the merit function.

After scoring all the concepts with respect to the criteria, the scores are normalized so that they can be used to calculate the merit function. The scores are normalized using Equation 1 [20]:

$$R_{ij} = \frac{A_i - A_j^{min}}{A_j^{max} - A_j^{min}} \tag{1}$$

R_{ij} is the normalized score, A_i is the score of alternative i, A_j^{min} and A_j^{max} are the lowest and highest value of the scores for each criterion, respectively.

The merit function is then obtained by multiplying the normalized scores from each criterion for a concept with their relative importance. The merit function will be used to compare the different concepts in the Preliminary Selection DSP. High values will indicate preference, and the concepts with the highest merit function will be the most-likely to succeed. The equation for calculating the merit function is given below [20]:

$$MF_i = \sum_{1}^{j} I_{ij} \cdot R_{ij} \tag{2}$$

MF_i is the merit function of concept i, I_{ij} is the relative importance of criterion j in concept i, and R_{ij} is the normalized score of criterion j in concept i.

Step 6: Include interactions between generalized criteria.

The merit functions of the concepts are calculated using all concepts as datums. This produces a number of merit functions for each concept, and these are combined in an overall merit function for each concept. The overall merit function will provide the final rating in the Preliminary Selection DSP. High values indicate preference, and the top concepts will serve as feasible alternatives for the Selection DSP.

Step 7: Post-solution sensitivity analysis.

To ensure the quality of the Preliminary Selection DSP, a post-solution sensitivity analysis should be included. Different scenarios of the relative importance of the criteria can be added to see changes made to the merit functions. This can help to get a better view of how the relative importance influences the results.

2.2 The Selection DSP

In the Selection DSP, the feasible alternatives found in the Preliminary Selection DSP are ranked in order of preference. The result of the Selection DSP will give an indication of how much one alternative is preferred over the other and the decision-maker can choose the best alternative for the design problem. In this step in the design phase, more objective, science-based information is often available and the Selection DSP uses both objective and experience-based information to rank the alternatives. The alternatives are found in the Preliminary Selection DSP and the main attributes that are important for the design problem are identified. The attributes are ranked with respect to their importance and the alternatives are compared and rated based on the identified attributes. Based on this rating and the attributes' relative importance, the final ranking of the alternatives and order of preference is found. The process is given as follows [20]:

Given A set of feasible alternatives

Identify The main attributes that influence the design problem and the relative importance of each attribute

Rate The alternatives are rated with respect to the attributes and their relative importance

Rank The alternatives are ranked by order of preference

The steps in the Selection DSP are summarized in Table 3 [20].

Table 3: Steps in the Selection DSP

Step 1	Describe each alternative and provide acronyms
Step 2	Describe each attribute, provide acronyms and specify their relative importance
Step 3	Specify scales, rate the alternatives with respect to each attribute
Step 4	Normalize ratings
Step 5	Evaluate the merit function for each alternative
Step 6	Post-solution sensitivity analysis

Step 1: Describe the alternatives and provide acronyms
The feasible, most-likely to succeed alternatives from the Preliminary Selection DSP must be described in detail, including main characteristics, advantages and disadvantages. Acronyms are given to the alternatives for simplification.

Step 2: Describe each attribute, provide acronyms and specify the relative importance of each attribute.
The relevant attributes for the design problem are listed, given acronyms and described. The relative importance of the attributes are found by ranking the attributes by their importance. The least important attribute is given the lowest value 1, and the most important attribute is given the highest value, which will correspond to the number of selected attributes. The relative importance of each attribute is then calculated by dividing the rank of importance with the sum of ranks for all attributes, as shown in Equation 3 [20]:

$$I_i = \frac{m_i}{\sum_i^m m_i} \tag{3}$$

I_i is the relative importance of attribute i , m_i is the rank given to attribute i and m is the number of attributes. High scores will indicate increased importance.

Step 3: Specify scales and rate the alternatives with respect to each attribute.

The attributes represent different aspects of the design problem, and the available information can vary from hard, science-based information to human judgment and experience. Different scales are therefore implemented to get consistent results throughout the evaluation. The types of scales that can be used are listed in Table 4 [20].

Table 4: Types of scales used in the Selection DSP

Ratio	The ratio scale is used when physical quantities are available, for example length, mass, power consumption etc. The ratio scale is based on objective, science-based information. It is important to specify the upper and lower bound of the scale independent of the alternatives.
Interval	The interval scale is used when soft information is available. It is used to transform experience-based information into a numerical interval scale. The boundaries of the scale should be determined with respect to all available technologies, not the feasible alternatives of the Selection DSP. When an appropriate interval is set, the alternatives are rated.
Ordinal	The ordinal scale is used when words are used to describe an attribute.
Composite	The composite scale can resemble the interval scale. It is used to model the collective preference related to a number of sub-attributes.

Care must be taken when computing the scales to ensure the quality of the Selection DSP. The boundaries of the scales are based on knowledge and experience, and justification is important for validation. It is important that the boundaries are computed based on all available technologies to get a realistic picture of the performance of the feasible alternatives in the Selection DSP.

Step 4: Normalize ratings.

In step 3, the alternatives are rated with respect to each attribute. The alternatives are rated in different scales and the ratings have to be normalized to be able to compare and rank them. Both high and low ratings in the different scales may indicate preference, and this has to be accounted for. To normalize the ratings where low values indicate preference, Equation 4 is used and Equation 5 is used when high values of ratings indicate preference [20]:

$$R_{ij} = 1 - \frac{A_{ij} - A_j^{min}}{A_j^{max} - A_j^{min}} \tag{4}$$

$$R_{ij} = \frac{A_{ij} - A_j^{min}}{A_j^{max} - A_j^{min}} \tag{5}$$

R_{ij} is the normalized rating, A_{ij} is the rating of alternative i with respect to attribute j, A^{min} and A^{max} are the minimum and maximum value of alternative rating A_{ij}, respectively.

The normalized ratings will range from 0 to 1, with high values indicating preference.

Step 5: Evaluate the merit function of each attribute.
The normalized scores of the attributes are multiplied with their relative importance to obtain the merit function. The merit function yields the final ranking of the alternatives. A high value of the merit function will indicate preference. The equation is given below [20].

$$MF_i = \sum_{j=1}^{n} I_j \cdot R_{ij} \tag{6}$$

MF_i is the merit function for alternative i, I_j is the relative importance of attribute j and R_{ij} is the normalized ranking of attributes.

Step 6: Post-solution sensitivity analysis.
A post-solution sensitivity analysis should be included to validate the ranking of the alternatives. The attributes that seem to contribute to either a very low or high merit function could be changed to see how it will affect the outcome of the Selection DSP. If small changes to the merit functions results in different rankings, care should be taken when making a final decision of technology. A step that can be made are to add more attributes to better differentiate between the feasible alternatives and go through the Selection DSP to see how this will affect the outcome.

2.3 Using the DSP method for decision-making

The following section is included to show the features of the computational tool. An example is presented to illustrate the steps and calculations in the DSP method. The case is to select a cryogenic separation process for removal of CO_2 from a natural gas field containing 30 mol% CO_2 at an offshore location. Detailed explanations of the technologies and criteria used in this example are given in section 6.2, where the DSP method is performed on the same case with aid from the computational tool developed. Section 6.2 also contains an introduction to the case study and discussions of results.

2.3.1 Preliminary Selection DSP

Step 1: Describe the concepts and provide acronyms.
The concepts are given acronyms for simplification.

CryoCell - CRYO

Sprex - SPREX

Ryan/Holmes - RH

Twister - TWIST

Controlled Freeze Zone - CFZ

Amine process - AMINE

Step 2: Describe each generalized criterion, and provide acronyms and weighting constants for the specific criteria.
The case study specifies a natural gas containing 30 mol% CO_2 that is to be processed on an offshore platform. Many aspects are important for decision-making and the following criteria has been developed for the case study:

Complexity - COMPLEX

Performance - PERFORM

Power consumption - POWER

Safety - SAFE

Maturity - MATUR

The relative importance, or weighting, of each criterion is based on experience, and will thereby contribute to the accuracy of the Preliminary Selection DSP. Scenarios with different weighting of relative importance are listed in Table 5.

Table 5: Scenarios of different weighting of the relative importance

Generalized Criterion	Scenarios				
	1	2	3	4	5
COMPLEX	0.2	0.2	0.1	0.1	0.4
POWER	0.2	0.1	0.2	0.4	0.2
SAFE	0.2	0.4	0.2	0.2	0.2
PERFORM	0.2	0.2	0.4	0.2	0.1
MATUR	0.2	0.1	0.1	0.1	0.1

Step 3: Choose a datum with which all other concepts will be compared.
The amine process will be chosen as the initial datum. This is the most accepted technology, and it is assumed to be one of the most likely to succeed technologies.

Step 4: Compare the concepts.
The concepts are compared with respect to the chosen datum and the attributes. The scores of the concepts in the Preliminary Selection DSP are shown in Table 6. This process is repeated using all the concepts as datum.
The scores are based on soft information, and it is therefore important to justify the view-point behind the evaluation.

Step 5: Evaluate the merit function for each concept within each generalized criterion.
The scores are normalized using Equation 1. The normalized scores are shown in Table 6. The merit functions are calculated by Equation 2. The relative importance of the criteria for the different scenarios is given in Table 5. The merit functions of the concepts when using the Amine process as datum are shown in Table 7.

Table 6: Scores in Preliminary Selection. Amine as datum

Criteria	Concepts					
	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX	1	-1	1	1	1	0
Normalized score	1	0	1	1	1	0.5
PERFORM	-1	0	0	-1	0	0
Normalized score	0	1	1	0	1	1
POWER	1	1	1	1	1	0
Normalized score	1	1	1	1	1	0
SAFE	-1	-1	-1	-1	-1	0
Normalized score	0	0	0	0	0	1
MATUR	-1	-1	0	-1	-1	0
Normalized score	0	0	1	0	0	1

Table 7: Merit function for all technologies in different scenarios using Amine process as datum

Scenario	Concepts					
	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.400	0.400	0.800	0.400	0.600	0.700
2	0.300	0.300	0.600	0.300	0.500	0.800
3	0.300	0.600	0.800	0.300	0.700	0.750
4	0.500	0.600	0.800	0.500	0.700	0.550
5	+.600	0.300	0.800	0.600	0.700	0.600

Step 6: Include interactions between generalized criteria.
The overall merit functions are calculated from the merit functions obtained by using six datums, and the result is shown in Table 8.

Step 7: Post-solution analysis, determine the most likely to succeed concepts.
Different scenarios are defined to take into account the variation of the relative importance. By comparing the results with different weighting, the sensitivity of the generalized criteria can be analyzed. A plot to demonstrate the ranking in the different scenarios is shown in Figure 1.
From Table 8 the concepts with the highest merit functions for all scenarios are the Amine process, the Ryan/Holmes process and the Controlled Freeze Zone process. These are chosen as the most-likely to succeed alternatives and will be evaluated further in the Selection DSP.

Table 8: Merit function of Preliminary Selection DSP

Scenario	Concepts					
	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.350	0.267	0.617	0.400	0.550	0.617
2	0.242	0.242	0.450	0.300	0.442	0.717
3	0.242	0.458	0.683	0.300	0.625	0.708
4	0.375	0.308	0.633	0.500	0.525	0.508
5	0.442	0.183	0.508	0.600	0.525	0.433

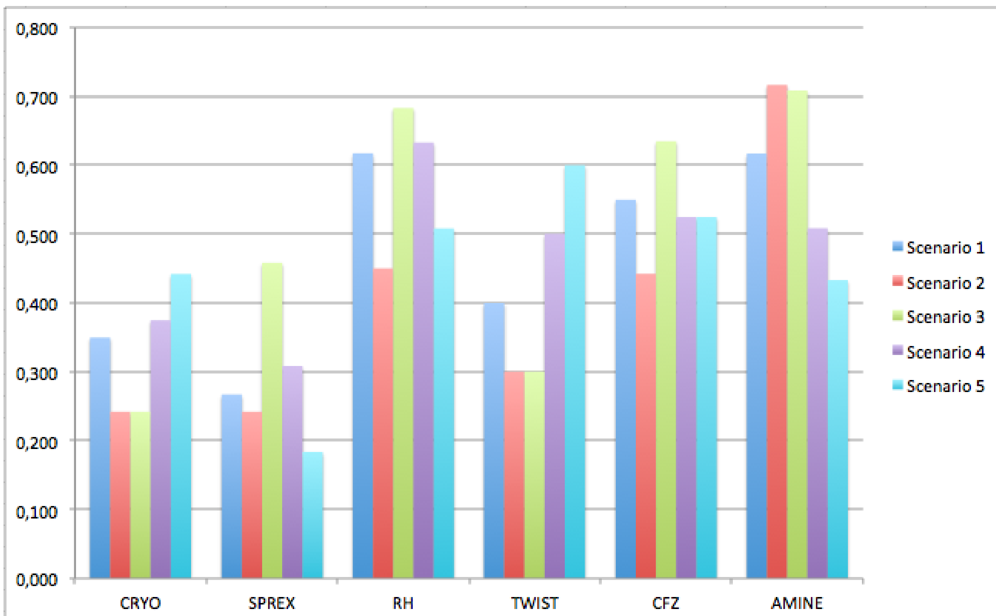


Figure 1: Plot representing scores in the Preliminary Selection DSP

2.3.2 Selection DSP

From the Preliminary Selection DSP, it was found that the most likely to succeed alternatives were the Amine process, the Controlled Freeze Zone technology and the Ryan/Holmes process. These are evaluated in the Selection DSP to determine the best technology for the sour gas containing 30 mol% CO_2.

Step 1: Describe the alternatives and provide acronyms.
The acronyms assigned to the alternatives will be:

Controlled Freeze Zone - CFZ

Amine process - AMINE

Ryan/Holmes process - RH

Step 2: Describe each attribute, specify the relative importance of the attributes and provide acronyms.
The following attributes are defined for the Selection DSP:

Environmental - ENVIRON

Compactness - COMPACT

Number of equipment - EQUIP

Power Consumption - POWER

Operability - OPER

Sensibility to motion - SENS

The relative importance of the attributes is shown in Table 9. The relative importance is calculated using Equation 3. The most important attribute is given the highest number and the least important attribute is given the lowest number.

Table 9: Relative importance of the Attributes

Attributes	Rank of importance	Relative importance
ENVIRON	1	0.048
COMPACT	6	0.286
EQUIP	3	0.143
POWER	5	0.238
OPER	2	0.095
SENS	4	0.190

Step 3: Specify scales and rate the alternatives with respect to each attribute.
The type of scaling used is identified with the description of the attributes. The interval scale is used for the Environmental, the Compactness, the Power Consumption, the Operability and the Sensibility to motion attributes. The interval scale is used when there is no hard

(science-based) information available. The Ratio interval is used for the Number of Equipment attribute. From schematics, the number of equipment for the specific alternative can be found as a physical number. The upper bound will be the highest number of equipment for CO_2 removal processes and the lower bound will be the minimum number of equipment necessary. The attributes are ranked in Table 10. The scale type and the upper and lower bounds are also given.

Table 10: Alternative rating and attribute scales

Alternatives	Attributes					
	ENVIRON	COMPACT	EQUIP	POWER	OPER	SENS
AMINE	6	7	7	9	6	4
RH	5	6	6	4	6	4
CFZ	4	4	5	6	4	2
Attribute summary:						
Type	Interval	Interval	Ration	Interval	Interval	Interval
Preference	Low	Low	Low	Low	Low	Low
Upper Bound	8	9	14	10	7	5
Lower Bound	0	1	1	1	0	0

Step 4: Normalize ratings.

The attributes are normalized using Equation 4. The normalized attributes are listed in Table 11.

Table 11: Normalized Alternative Rating

Alternatives	Attributes					
	ENVIRON	COMPACT	EQUIP	POWER	OPER	SENS
AMINE	0.250	0.250	0.538	0.111	0.143	0.200
RH	0.375	0.375	0.615	0.667	0.143	0.200
CFZ	0.500	0.625	0.692	0.444	0.429	0.600

Step 5: Evaluate the merit function for each alternative.

The merit functions are calculated using Equation 6. The normalized attribute ratings from Table 11 are multiplied by the relative importance of the attributes given in Table 9. The final ranking is presented in Table 12.

Table 12: Merit functions and final rating of the alternatives

Alternatives	Merit function	Difference (%)	Rank
AMINE	0.238	58	3
RH	0.423	25	2
CFZ	0.562	-	1

Step 6: Post-solution sensitivity analysis.

The Controlled Freeze Zone and the Ryan/Holmes process have the highest merit functions. A sensitivity analysis is performed to investigate the consequences by changing the relative importance of the attributes. The Ryan/Holmes process has the highest score for the power consumption attribute. This attribute is adjusted with a 20% increase in importance. The merit functions are then calculated and the result is investigated. The Controlled Freeze Zone has the highest merit function for the compactness attribute. The relative importance of this attribute is decreased by 20% and the merit functions are re-calculated. The merit functions are also calculated for a 20% increase in relative importance of the power consumption together with a 20% decrease of the compactness attribute. The results of these changes are seen in Table 13 and Figure 2.

Table 13: Merit functions with different relative importances

Alternative	20% Decrease Compact	20% Increase Power	20% Decrease Compact and 20% Increase Power	No Change
AMINE	0.224	0.244	0.229	0.328
RH	0.402	0.455	0.434	0.423
CFZ	0.526	0.583	0.548	0.562

It is seen that the changes in relative importance for the attributes will not affect the outcome of the Selection DSP, and the Controlled Freeze Zone process is the most suited alternative for the case study.

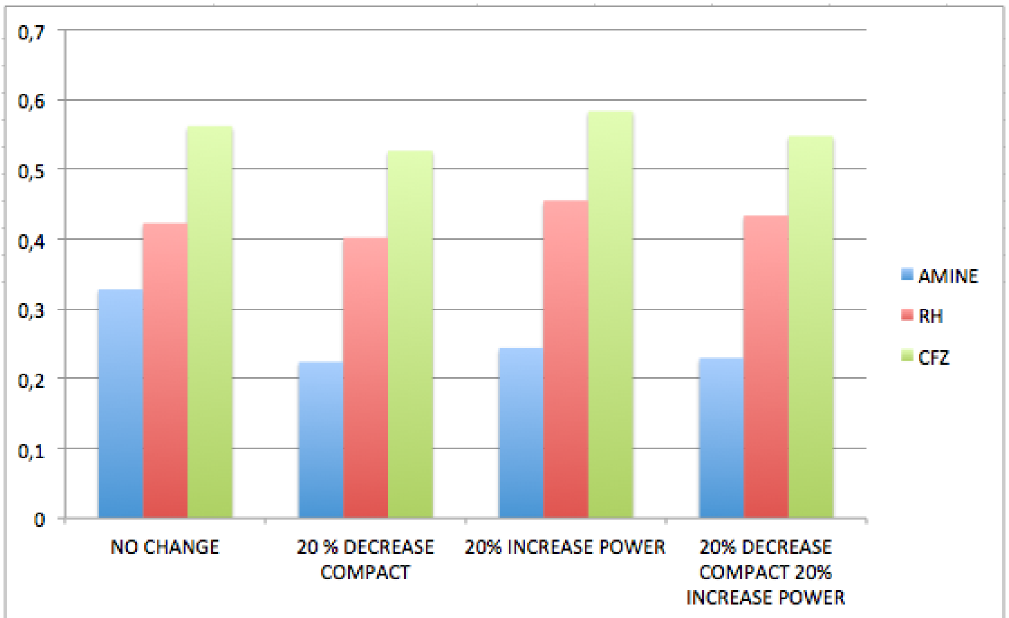


Figure 2: Post-solution sensitivity analysis for different values of relative importance

3 Object-oriented analysis and design of the Decision-Support Problem Technique

An object-oriented analysis is performed to transform the DSP method into a computational tool. The object-oriented analysis will describe how the program is structured through use cases, a sequence diagram and a class diagram. The object-oriented analysis does not include any programming code, and is performed prior to prototyping the method in any software. The analysis should be language independent and show an object-oriented structure that can be extended to any programming language [22]. Object-oriented analysis and design is an important tool for developing a new program. The analysis and design methods identify how the system will interact and communicate. By putting effort into the analysis and design, the process of writing code and organizing the program becomes a more manageable task [22]. The object-oriented analysis and design is performed using the Unified Modeling Language (UML). UML is a visual way of representing information and structure in object-oriented programming and a method for representing data types and their communication with each other [22]. The process of the DSP Technique has been described in a step-by-step manner. The object-oriented analysis and design will convert these steps into a system structure that can be further converted into programming code.

3.1 Objects

An object or type can illustrate a class or anything else with a unique identity. An object encapsulates certain attributes and operations. The attributes relate to the object's properties, and the operations denote what the object can do [22]. In the object-oriented analysis there is only need to define and deal with objects. What the objects are associated with becomes important during the design process [22]. By use of UML, objects can be illustrated as shown in Figure 3 [22]. The top bar includes the object's name, and the attributes and operations are listed below.

Objects communicate with each other through messages. Messages are illustrated in UML as linked lines between communicating objects. Objects communicate with each other to obtain or access data encapsulated within objects. Arrows help show which direction the messages are sent.

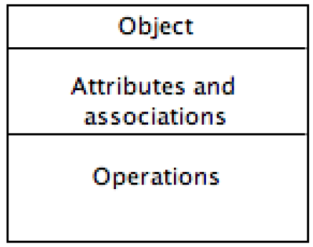


Figure 3: Detailed object structure in UML.

3.2 Use cases

Use cases are a popular technique for identifying objects and describing how the system interacts and communicates with its surroundings [22]. The use cases describe what is supposed to happen when a user interacts with the system, and identifies operations and actions needed to be performed to get a desired result. An action in a use case includes all operations, activities or tasks that need to be performed. Actions are visualized as ellipses in UML use cases [22]. Figure 4 and 5 show two use cases for the Preliminary Selection DSP and the Selection DSP, respectively.

The use case for the Preliminary Selection DSP consists of an actor, shown in Figure 4 as a pin man. An actor represents what interacts with the system. Here, the actor represents the user interface for the program. The actor supplies input for the Preliminary Selection DSP. The input is sent to the program and the given concepts and criteria are stored. From the input, the program also needs to store the relative importance of the criteria, and the scores of the concepts with respect to the datum and criteria. Normalized scores and merit functions of the concepts are calculated and stored. Results from the Preliminary Selection DSP are sent to the user, where the user then can evaluate the results of the merit functions to determine the most-likely to succeed concepts that will be further evaluated in the Selection DSP.

The use case for the Selection DSP in Figure 5 shows that the user sends input to the program. The input will consist of information about the alternatives and attributes, as well as the rankings of the attributes and the rating of alternatives with respect to the attributes. The program stores the information about the alternatives and attributes, as well as the rankings and ratings from the input data. The program then calculates the normalized ranks of the relative importance and the normalized ratings of the alternatives. The merit functions

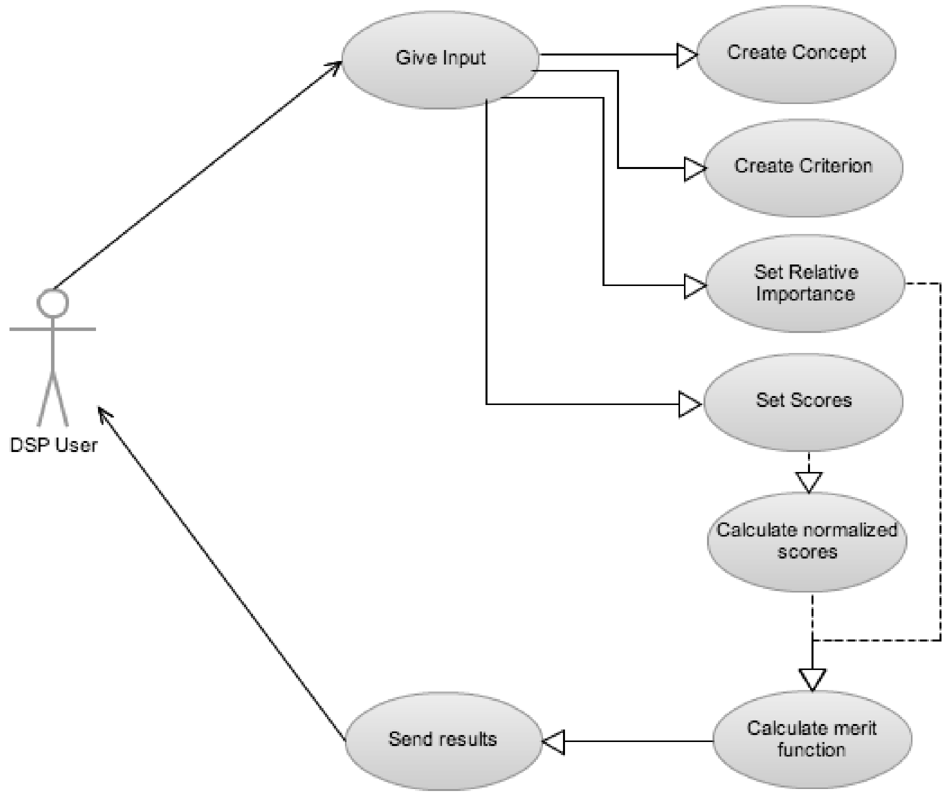


Figure 4: Preliminary Selection DSP use case

of the alternatives are then calculated and sent to the user as results. The user can evaluate the results to determine the best suited alternative for the specified case.

From the use cases for the Preliminary Selection DSP and the Selection DSP, the objects in the system have been identified to be objects for the concepts, the criteria, the alternatives and the attributes. These objects contain information about the names of the given concepts and alternatives, as well as names that are descriptive for the identified criteria and attributes. The concepts and alternatives describe available technologies, and the criteria and attributes describe certain criteria in which the technologies are evaluated. The concepts and alternatives will function as instances of object Concept, and criteria and attributes are instances of object Criterion. The calculations, scores and relative importances in the Preliminary Selection DSP are contained in object PreDSP and the ratings, calculations and rankings in the Selection DSP are encapsulated in object SelDSP.

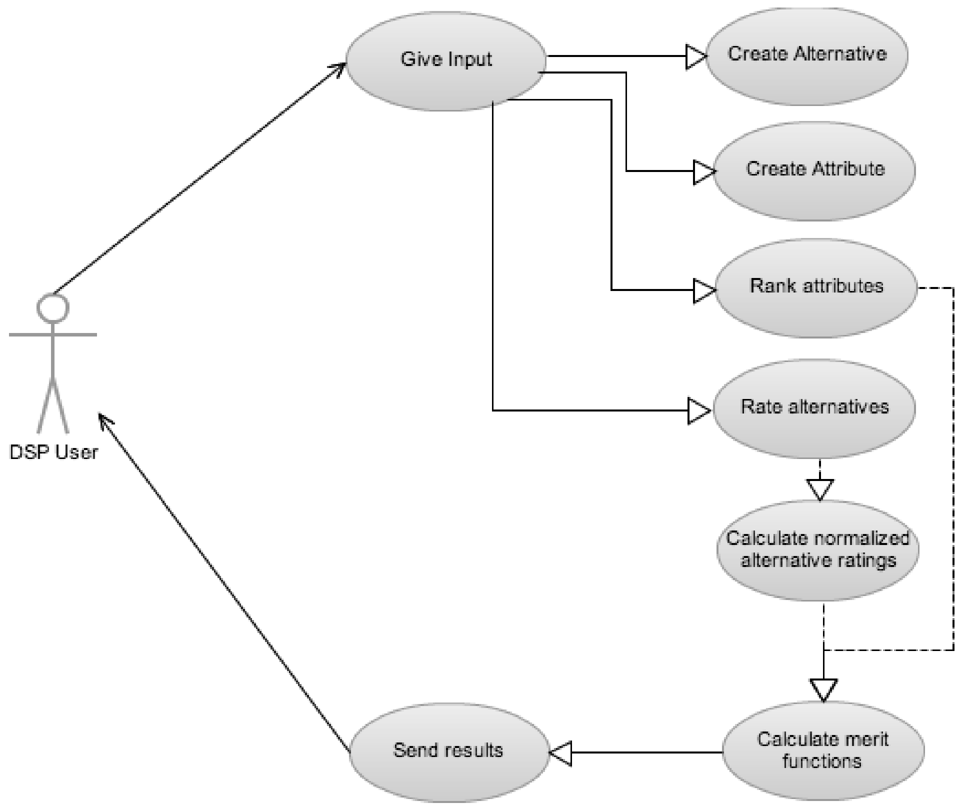


Figure 5: Selection DSP use case

3.3 Sequence diagram

The use cases in Figure 4 and 5 show the actions performed in the DSP method. From the use cases, the appropriate objects are identified and the actions are assigned to the specified objects. A more descriptive visualization is necessary to specify the communication between the identified objects. This can be done by use of a sequence diagram. Sequence diagrams are used to visualize use cases and show the sequence of events in the program structure [22]. Sequence diagrams describe messages between objects and actions that are performed. Vertical lines in the sequence diagram show instances of the type or object listed and the horizontal lines show the communication between objects. The sequence of events are also shown vertically. The order of events in a sequence diagram starts from the top and moves downwards [22]. From the use cases in Figure 4 and 5, a sequence diagram for the program structure of the DSP is shown in Figure 6. The object DSP User denotes the actor of the system. The actor represents the user and the first event of the DSP program is that the user provides input for the Preliminary Selection DSP as a text file. The Input text file/Output text file object represents the text file with information to the program and the text file with results from the calculations. The files are illustrated as part of the same object in the sequence diagram. The object PreDSP reads the content of the input text file and creates the appropriate instances of the objects Concept and Criterion. The PreDSP object performs the calculations of the Preliminary Selection DSP and stores all information. The next event of the sequence diagram shows that object PreDSP writes the results to a text file. The DSP User receives the results from the output text file and can evaluate the results. When the user has identified the feasible alternatives for the Selection DSP, the DSP User provides the system with the Selection DSP input. This is sent to the program as a text file. The object SelDSP reads the input text file created for the Selection DSP, and creates appropriate instances of the objects Concept and Criterion. The SelDSP object performs all the necessary calculations and stores the information. The results are written to a text file. The user is then presented with the results from the Selection DSP and can evaluate the results to choose the best suited alternative for the design case.

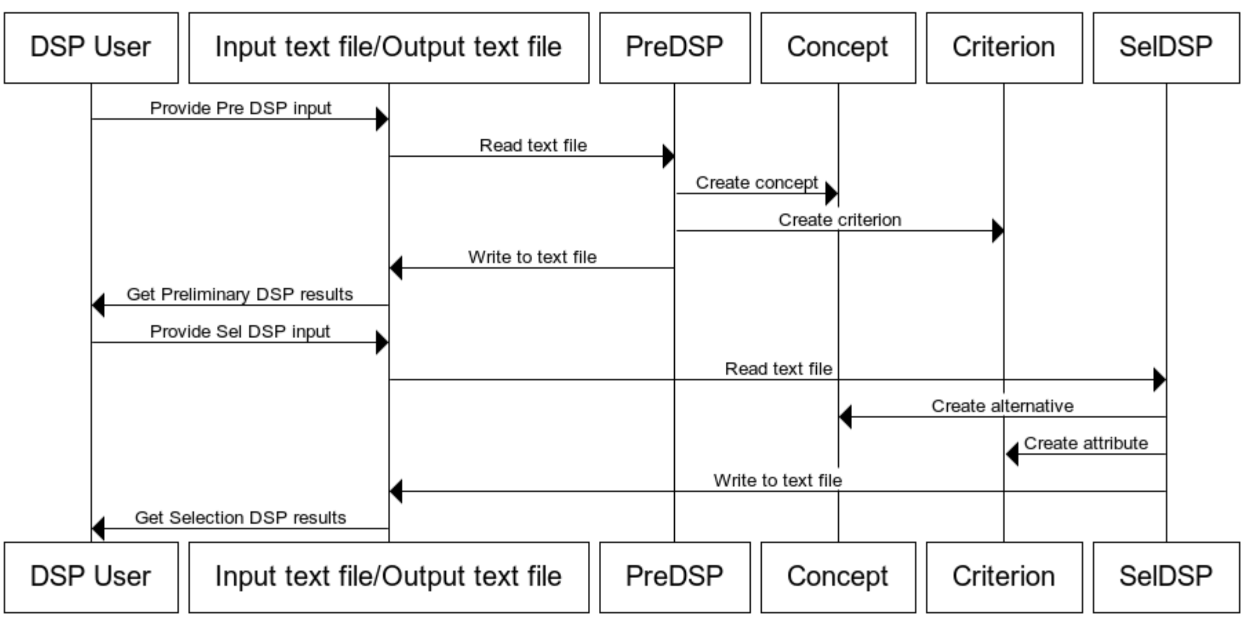


Figure 6: Sequence Diagram

3.4 Class diagram

From the use cases in Figures 4 and 5 and the sequence diagram in Figure 6, the objects and the necessary interactions between the objects are identified. A class diagram for the objects in the DSP program is created to show the attributes and the operations of each object in the DSP program. The class diagram is shown in Figure 7.

The class PreDSP relates to the Preliminary Selection DSP and the class SelDSP relates to the Selection DSP of the DSP method. They are connected to the classes Concept and Criterion, which denote technologies evaluated in the design case and the appropriate criteria or attributes identified with the design case, respectively. All attributes and operations are shown in the class diagram. This is a useful representation of the system for the development of the software and the DSP method is broken down into small operations, which makes the software coding a more managable task. The attributes of the classes in the class diagram is first listed with names. The attribute type is given after the colon mark. The operations are presented with names, and arguments in parenthesis. The return type is given after the colon mark.

A software can now be developed for the DSP method as the structure of the program and all relevant objects and their attributes and operations are identified.

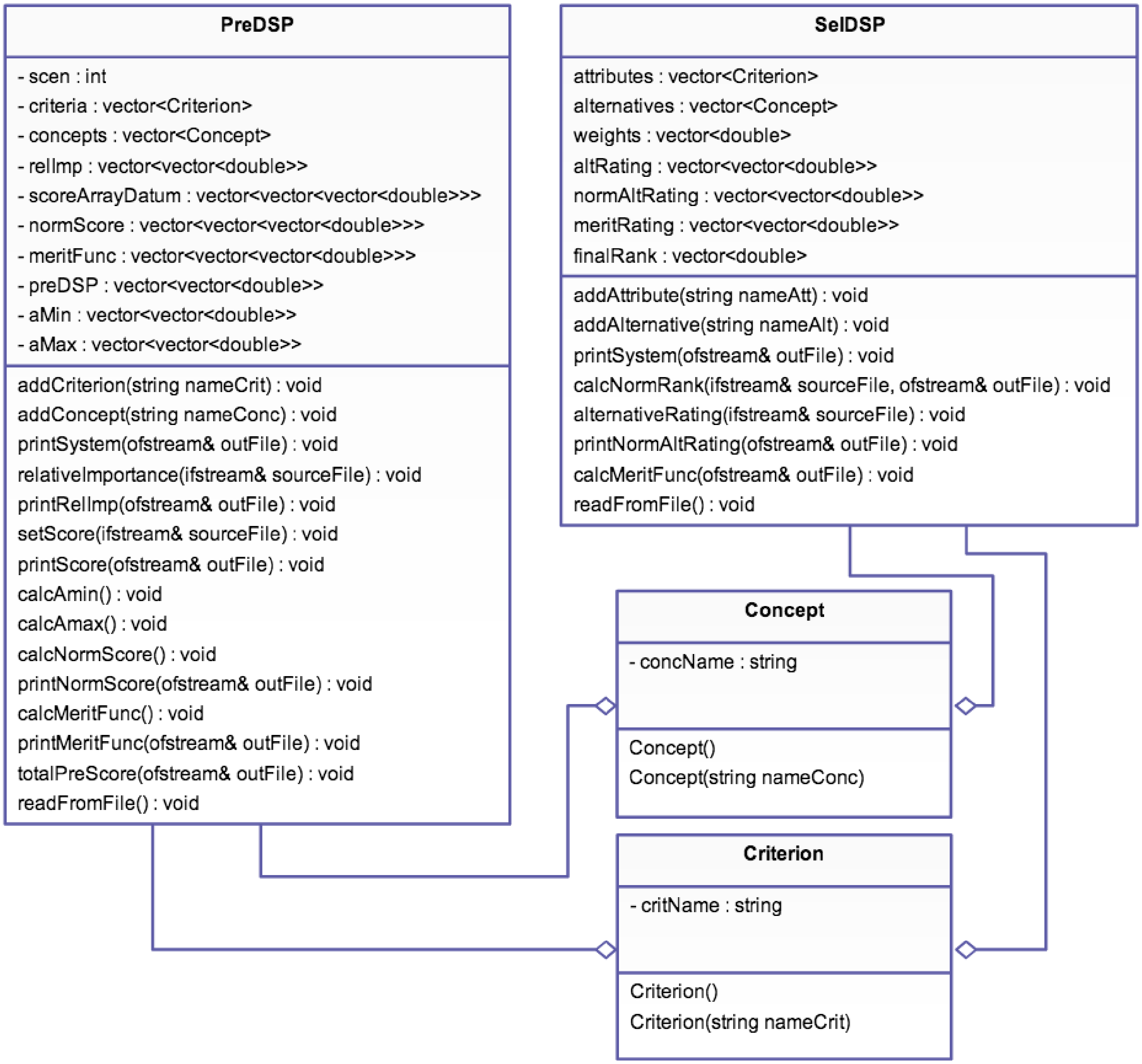


Figure 7: Class Diagram

4 Software prototyping

The DSP computational tool is developed in C++. C++ is an object-oriented language developed from the C programming language and is widely used for software development [29]. The main features of the computational tool for the DSP Technique will be described and explained. The most critical algorithms performed in the Preliminary Selection DSP and the Selection DSP are presented by showing snippets of code with explanations. The Preliminary Selection DSP section will show operations from the PreDSP class and the Selection DSP section shows operations from the SelDSP class.

4.1 Accessing data input

The program receives input from the user in form of a text file. The results from the Preliminary Selection DSP and the Selection DSP is written to a text file for the user to examine. Reading from and writing to files is defined in the <fstream> library in C++ . To read from a file, the program must connect that file to an object of the class ifstream. To write to a file, the program connects the file to an object of the class ofstream [29]. By making the program communicate with the user via reading and writing to text files, the user is distanced from the program. A user interface can then function as a bridge between the user and the program. The user communicates with an interface, where the input data is given. The interface writes all input data to a text file in the correct order, and the program reads from this text file and computes the calculations. This is done both for the Preliminary Selection DSP and the Selection DSP. The user can then examine the results from the Preliminary Selection DSP before moving on to the Selection DSP. The final results from the Selection DSP should be thoroughly examined. If there are marginal differences in the merit functions in the Selection DSP, a post-solution sensitivity analysis should be carried out.

4.2 Storing data

The objects defined from the use cases and sequence diagram are Concept, Criterion, PreDSP and SelDSP. These objects are specified as classes in the program. A class is a structure with member data and member functions [29]. The classes Concept and Criterion include the names of the concept and criterion, respectively. The names are stored in the classes as strings. The string library in C++ is used to store multiple characters. There is no need to specify the size of the string because the necessary space is automatically created for each string [29]. The acronym will be used to store the names of the concepts and criteria. The classes PreDSP and SelDSP contain attributes or member data to store data input from the user and the results from the calculations performed. The data is stored in vectors. Vectors are arrays that can change size while the program is running [29]. A base type of a vector must be specified, and the vector will be a collection of values of its base type. A vector is declared with its base type as follows [29]:

```
vector<Base_Type> nameOfVector;
```

In the DSP program, one-dimensional vectors are used to store instances of technologies and criteria, and multidimensional vectors are used to store relative importances, scores and rankings in the PreDSP class and SelDSP class. Multidimensional vectors are vectors inside vectors. To declare a two-dimensional vector, a vector type is used as the base type as follows:

```
vector<vector<Base_Type>> nameOf2dVector;
```

To add elements to a vector, the member function push_back is used. This function will add the desired element at the end of the vector.

```
nameOfVector.push_back(element);
```

A multidimensional vector is initialized with a loop that goes through the first vector and adds a second vector to each element as follows:

```
vector<vector<Base_Type>> nameOf2dVector;
for (int i = 0; i < nameOf2dVector.size(); i++ ) {
        vector<Base_Type> v;
        nameOf2DVector.push_back(v);
        for (int j = 0; j < v.size(); j++) {
                nameOf2dVector[i].push_back(element);
        }
}
```

The member function .size() returns the number of elements in the vector. To create a three-dimensional vector, a vector type is added as the base type of the second vector. To access elements in vectors, square brackets are used after the name of the vector created. Vector elements start their indexing with 0 and continue to the size of the vector. The member function .at() can also be used to access vector elements. The index that is to be accessed is placed inside the parenthesis and the member function will return an error if the specified element is not in the range of the vector.

```
nameOfVector[i];          // Access element i of vector
                          // nameOfVector
nameOfVector.at(i)        // Does the same as the
                          // bracket method, but will
                          // return an error if i is
                          // not an element
                          // of nameOfVector
```

4.3 Preliminary Selection DSP

User input is accessed by the class PreDSP. The PreDSP class creates instances of the Concept class for the concepts and Criterion class for the criteria. The user specifies the relative importance of the criteria and can add several scenarios of the relative importances. These are stored in a two-dimensional vector in the PreDSP class. The user scores the concepts on how well they perform on each criterion with respect to the chosen datum. This is done using all the concepts as datums. The scores are stored in a three-dimensional vector. The outer vector contains all the concepts, which in turn function as datum. This vector stores a two-dimensional vector with values of the concepts on how they score on the criteria with respect to the datum. The same setup is used for the function that calculates the normalized scores of the concepts. The normalized scores are found by use of Equation 1 in section 2.1. The minimum and maximum score values for each criterion are found and stored separately in a two-dimensional vector. These are needed to find the normalized scores. The merit functions are calculated by use of Equation 2 in section 2.1. The code for calculating the merit functions are shown below:

```
void calcMeritFunc() {
    double sum;
    for (int i = 0; i < concepts.size(); i++) {
        vector<vector<double>> mFunction;
        meritFunc.push_back(mFunction);
```

```
            for (int j = 0; j < scen; j++) {
                vector<double> row;
                meritFunc[i].push_back(row);
                sum = 0;
                for (int k = 0; k < concepts.size(); k++) {
                    sum = 0;
                    for (int l = 0; l < criteria.size(); l++) {
                        sum += (normScore[i][l][k]
                            * relImp[j][l]);
                    }
                    meritFunc[i][j].push_back(sum);
                }
            }
        }
}
```

Calculations of merit functions are done by multiplying the normalized scores with the relative importance of each criterion for all scenarios. The normalized scores are a collection of two-dimensional arrays for each concept acting as datum. The function multiplies the normalized score of each criterion with the relative importance of the criterion and these are added over all criteria for each concept. The result is a three-dimensional vector that contains the merit functions of the concepts for all scenarios, using all concepts as datums. A setup of the configuration is shown in Table 14. All concepts are used as datums and for each concept as datum, a two-dimensional vector shows the merit functions of all concepts for every scenario.

Table 14: Representation of merit functions

Concept no. 1 is datum

Scenario	Concepts		
	Concept 1	...	Concept m
Scenario 1	merit function [1][1]	...	merit function [1][m]
Scenario 2	merit function [2][1]	...	merit function [2][m]
.	.	...	.
Scenario n	.	...	merit function [n][m]

The final result is found by taking the overall value of every set of merit functions where each concept is used as datum. The function for calculating the overall merit functions is given below:

```
void totalPreScore(ofstream& outFile) {
    double sum;
    for (int i = 0; i < scen; i++) {
        for (int j = 0; j < concepts.size(); j++) {
            vector<double> row;
            preDSP.push_back(row);
            for (int k = 0; k < concepts.size(); k++) {
                sum += meritFunc[k][i][j];
                }
            sum = sum / concepts.size();
            preDSP[i].push_back(sum);
        }
    }
}
```

The result from the Preliminary Selection DSP is given to the user in form of the overall merit functions. This is a set of merit functions of the concepts for each scenario specified by the user. The user must then evaluate the results in a post-solution sensitivity analysis where the top rated concepts for every scenario is identified. The concepts that distinguish from the others in terms of higher overall merit functions are chosen as feasible alternatives and used in the Selection DSP. If there are no concepts that stand out from the merit functions, the Preliminary Selection DSP may be performed again with additional criteria or scenarios to identify the most-likely to succeed alternatives. A text file of all the results from the calculations performed in the Preliminary Selection DSP is available for more details.

4.4 Selection DSP

Calculations of the Selection DSP are done by the class SelDSP. The class receives input from the user regarding the feasible alternatives and the attributes they are to be compared to. The user ranks the attributes in order of importance, with the least important attribute given the value 1 and the most important attribute given the value reflecting the number of attributes given. The rankings are normalized by use of Equation 3 in section 2.2. The scales and preference of the scales are then identified and the alternatives are rated on these scales for each attribute. The program normalizes the ratings so that they can be compared. The normalizing is done by use of Equation 4 or Equation 5 in section 2.2. The preference of the scales determines which equation is used. The algorithm for normalizing the ratings is shown below:

```
void alternativeRating(ifstream& sourceFile) {
    double input, lowBound, highBound;
    string pref;
    for (int i = 0; i < attributes.size(); i++) {
        vector<double> row;
        normAltRating.push_back(row);
        sourceFile >> highBound;
        sourceFile >> lowBound;
        sourceFile >> pref;
        for (int j = 0; j < alternatives.size(); j++) {
            sourceFile >> input;
            if (pref == "L") {
                pref = l;
            }
            if (pref == "H" ) {
                pref = h;
            }
            if (pref == "l") {
                normAltRating[i].push_back(1 - (input -
                    lowBound) / (highBound - lowBound));
            }
            else if (pref == "h") {
                normAltRating[i].push_back(
                    (input - lowBound) /
                    (highBound - lowBound));
```

```
            }
        }
    }
}
```

The user specifies if high values or low values of the ratings indicate preference by typing "h" if high values are preferred or "l" if low values indicate preference. The program recognizes the input and uses the appropriate equation. The normalized scores are stored in a two-dimensional vector that specifies the attributes and the normalized ratings of each alternative.

The normalized ratings are multiplied with the normalized rankings of the attributes in a function that calculates the merit functions. The merit functions for the alternatives are then found by adding the merit functions for the alternatives over all attributes. The algorithm for the calculation of the merit functions is given below:

```
void calcMeritFunc(ofstream& outFile) {
    for (int i = 0; i < attributes.size(); i++) {
        vector<double> row;
        meritRating.push_back(row);
        for (int j = 0; j < alternatives.size(); j++) {
            meritRating[i].push_back(normAltRating[i][j]
                * weights[i]);
        }
    }
    double sum;
    for (int i = 0; i < alternatives.size(); i++) {
        sum = 0;
        for (int j = 0; j < attributes.size(); j++) {
            sum += meritRating[j][i];
        }
        finalRank.push_back(sum);
    }
}
```

The result is a vector named finalRank that stores the rankings of the alternatives. The alternative with the highest merit function is calculated to be the most suitable alternative for the specified case, and is presented to the user for evaluation. The user will examine the results and determine whether one alternative stands out as the most suited alternative, or if there is too little margin between the feasible alternatives to make a decision. The analysis may then be performed again with additional attributes to further distinguish the alternatives. A detailed text file of the input and results of the Selection DSP is available for information.

4.5 Validation of computational tool

The DSP computational tool created, is able to perform calculation for the DSP method with an unknown number of technologies and criteria. Calculations are performed accessing the indexes of each vector. Information is systematically added to vectors so that the calculations performed with technologies and criteria are matched in order, and the correct numbers are used. The object-oriented analysis performed in Chapter 3 provides means to divide the problem statement of creating a computational tool into smaller problems, which are more easily solved. The use cases reveals the necessary actions in the DSP method that have to be captured in the computational tool. The sequence diagram provides a visual structure of how different objects communicate, and the order of communication. Finally, a class diagram shows the attributes and operations that need to be created to perform the necessary actions of the objects to provide desired results. By performing an object-oriented analysis, prototyping the software becomes a more manageable task, with objects and operations already identified. The DSP Technique becomes more efficient with aid of the program developed to perform calculations.

5 User Interface

A user interface is created for the DSP program to guide the user through the steps of the DSP Technique. The user interface is created in the iOS platform for iPhone and written in Objective-C. Objective-C is an object-oriented programming language, and the default language of all Apple products. Objective-C is an extension of C, which is a procedural language [15]. An iPhone application may not be the optimal platform for a user interface for the DSP computational program, but it is chosen due to its simplicity and the work load put into developing the interface. The optimal interface may be desktop based, but developing an interface is time consuming, and therefore the interface iPhone application is developed as a prototype to be used as a guide for the user to perform the DSP.

In the application, the DSP program is included in C++ files. The application is written in Objective-C, and the two languages, though similar, cannot be mixed together without making some alterations. Mixing the two languages is done by converting the Objective-C class that will make a function call to a C++ class to an Objective-C++ class. This is done by changing the name of the class from a .m file to a .mm file. When this is done, the Objective-C++ class can import the relevant C++ class and make function calls to this class.

The user provides information about the technologies and criteria for a specific case. An overview of the different views in the application are shown in Figure 8. Here, all the tab views in the menu bar are shown. The first view is seen at the bottom left of the black tab bar. The next views continue along the tab bar to the right. Under the tab bar ”More”, the rest of the views are shown in the window, starting from the top and moving downwards. The tab bar images show which part of the DSP the views belong to. PreDSP indicate that the view is part of the Preliminary Selection DSP and SelDSP indicates that the view belongs to the Selection DSP part of the program. The two bottom table view cells in Figure 8 gives the option of examining a detailed result of Preliminary Selection and the Selection DSP, where all results of the calculations can be viewed in a text file.

A storyboard of the views in the application are shown in Figure 9 and Figure 10 for the Preliminary Selection DSP part and the Selection DSP part, respectively. The details of the user interface is presented for the Preliminary Selection part and the Selection part of the DSP program. In Figures 9 and 10, some technologies and criteria are added to show how the application works. The number of technologies and criteria that can be added are not

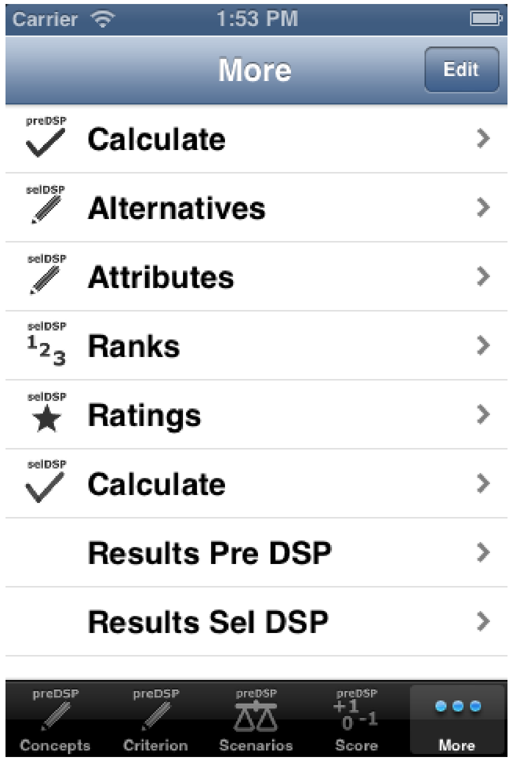
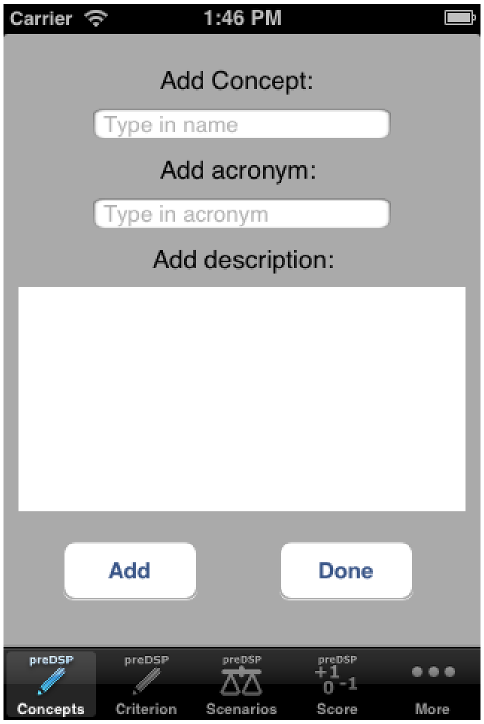


Figure 8: Overview of the views in the application

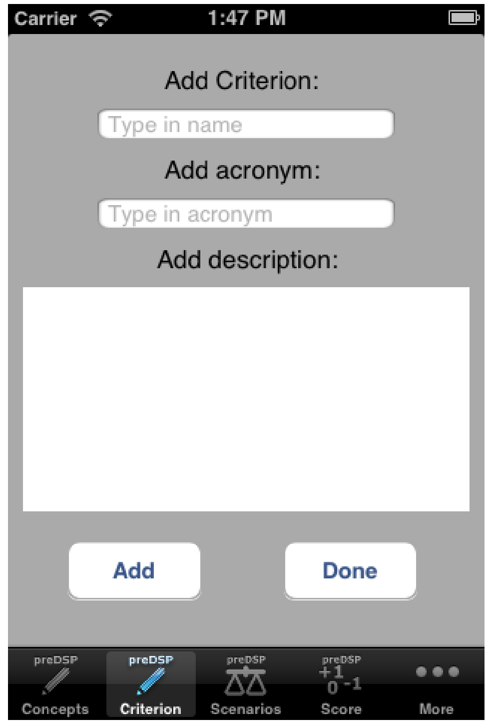
limited to any number, and the numbers shown in the figures are only for illustration.

5.1 Preliminary Selection DSP

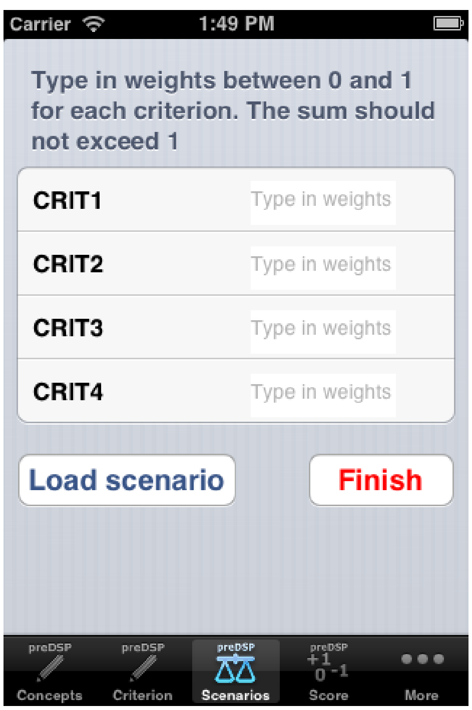
A storyboard view of the Preliminary Selection DSP is given in Figure 9. The first view lets the user add information about the concepts for the Preliminary Selection DSP. The first view is given in Figure 9a. The user adds the name, acronym and a description of the concept to the respective fields. The interface writes the name and description to a text file for documentation. The acronym is stored separately to be used in the scoring of the concepts. Figure 9b shows the view for adding criteria to the Preliminary Selection DSP. The user provides information about the name, acronym and description of each generalized criterion, and the interface writes the name and description to a text file for documentation, while the acronym is used for the Preliminary Selection part. The criteria are given relative importances in the view given in Figure 9c. The user may add several scenarios of relative importances to each criteria. After each scenario of relative importances for the criteria is



(a) Add concepts



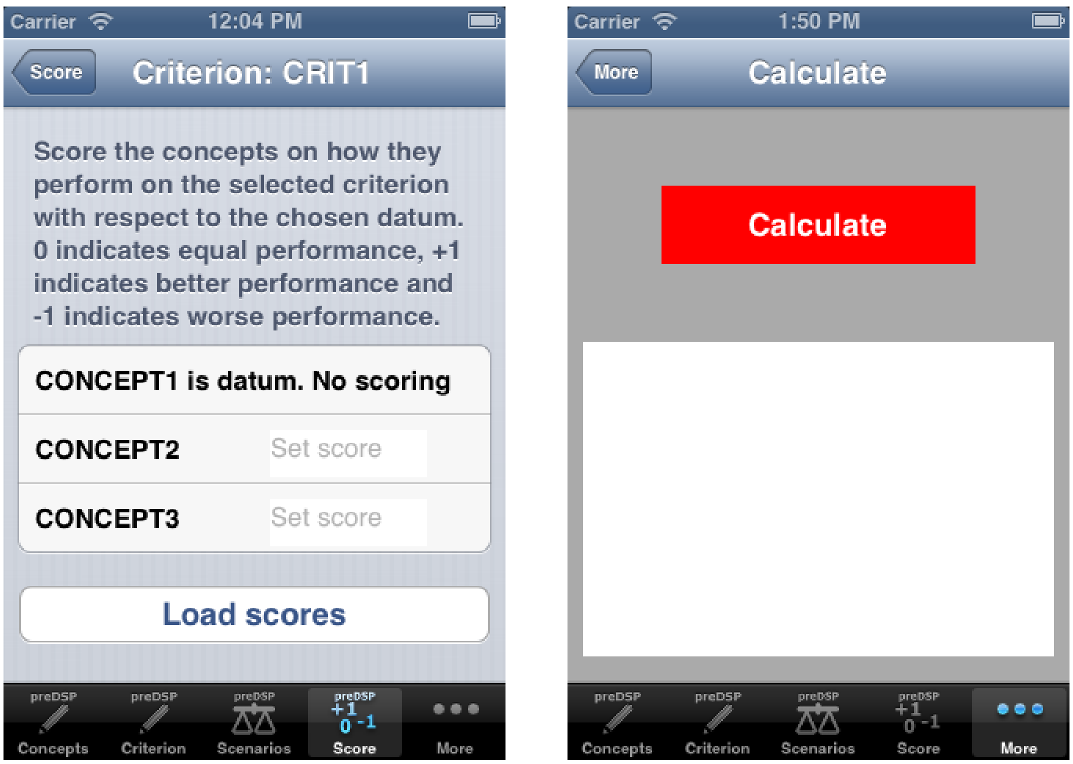
(b) Add criteria



(c) Add relative importance of criteria

(d) Score concepts

Figure 9: Screenshots from Preliminary Selection DSP application



(e) Score concepts for selected criterion (f) Calculate view

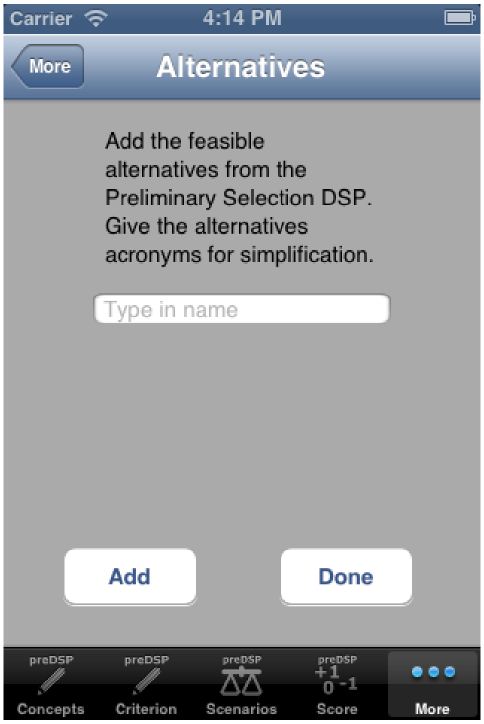
Figure 9: Screenshots from Preliminary Selection DSP application

added, the button ”Load scenario” is pushed and when done, the ”Finish” button is pushed before continuing to the next view. The view in Figure 9d shows the scoring process of the Preliminary Selection DSP. The concepts added will in turn function as the datum, and the user selects the criteria under each concept to continue to the scoring of the concepts for the selected criteria and datum. By selecting a criteria, the user is directed to the view in Figure 9e where the scores for the concepts are given for the selected criterion, with respect to the selected datum. The calculation view in Figure 9f lets the user press a Calculate button, which prompts the DSP program to perform calculations and prints the merit functions as result from the Preliminary Selection DSP.

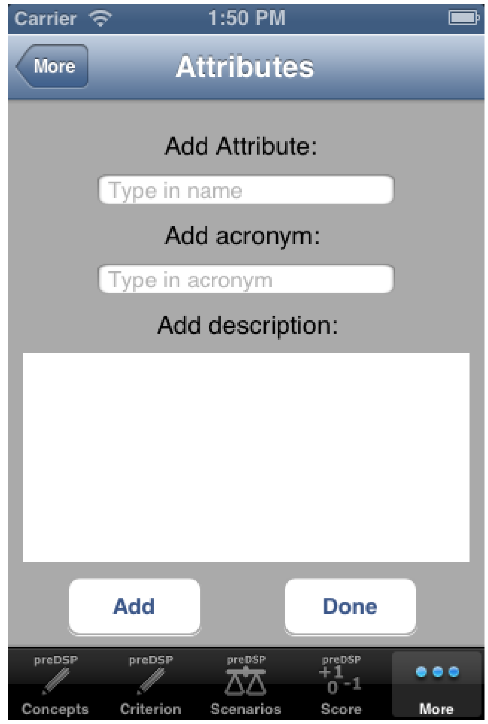
The user input from the interface is written to a text file in the correct order. The text file is read by the PreDSP class when the Calculate button is pushed, and the PreDSP class performs calculations of the Preliminary Selection DSP. The result is written to a text file, which can be viewed in the application.

5.2 Selection DSP

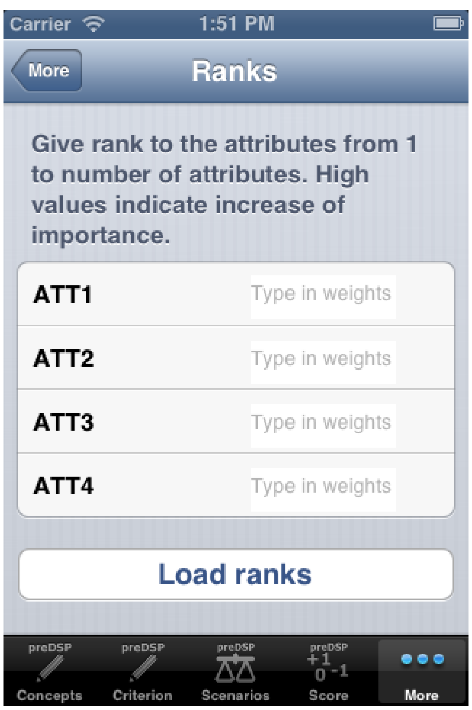
A summary of the views in the Selection DSP part of the DSP is given in Figure 10. The user continues to the Selection DSP after receiving results from the Preliminary Selection DSP. The first step is to add the feasible alternatives from the Preliminary Selection, which is done as indicated from Figure 10a. The user only provides the acronym because the name and description are already added in the Preliminary Selection. The attributes are added in the view given in Figure 10b. The user provides the name, an acronym and a description to the respective fields, and the interface writes the name and description to a separate text file for documentation. The acronyms are stored and used throughout the Selection DSP. In the view in Figure 10c, the user ranks the importance of the attributes. The button ”Load ranks” is pushed when all the attributes are given a rank. The next view is shown in Figure 10d. The view displays a list of the attributes. The user will select each attribute in turn and is then directed to the view in Figure 10e. Here, the bounds of the selected attribute is given and the user indicates the preference of the rates given to the alternatives. The alternatives are then rated on how they perform on the selected attribute. When scales are provided to all the attributes and the alternatives are rated on all attributes, the user continues to the Calculate view shown in Figure 10f. The user pushes the calculate button that sends a message to the DSP program to start calculations. The result of the Selection DSP is



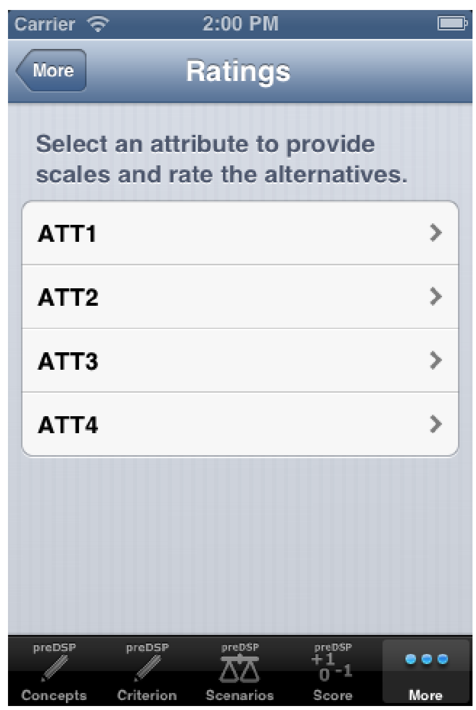
(a) Add alternatives



(b) Add attributes

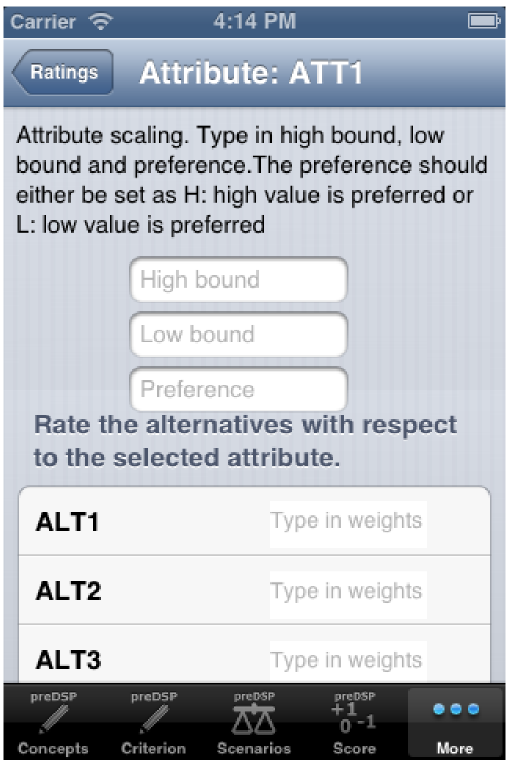


(c) Rank importance of attributes

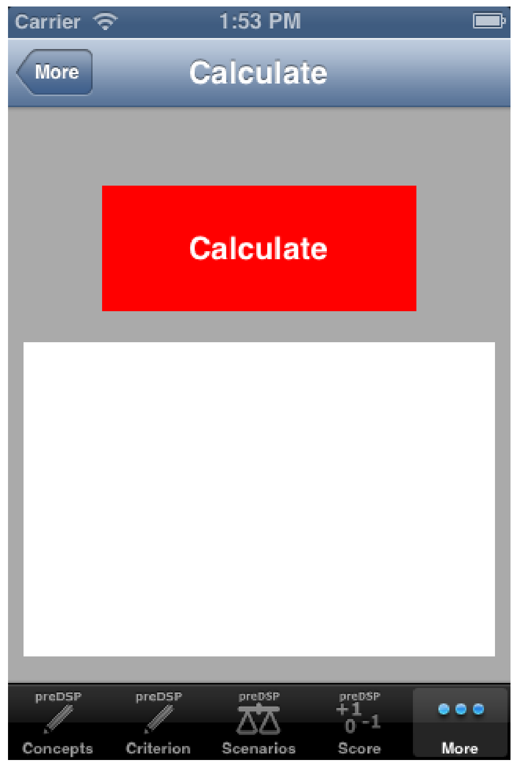


(d) Select attribute for rating

Figure 10: Screenshots from Selection DSP application



(e) Provide scales and rate alternatives



(f) Calculate view

Figure 10: Screenshots from Selection DSP application

displayed in the text view below the button.

The user input from the interface is written to a text file which is read by the SelDSP class when the calculate button is pushed. SelDSP performs the calculations of the Selection DSP and writes the results to a text file which can be viewed in the application.

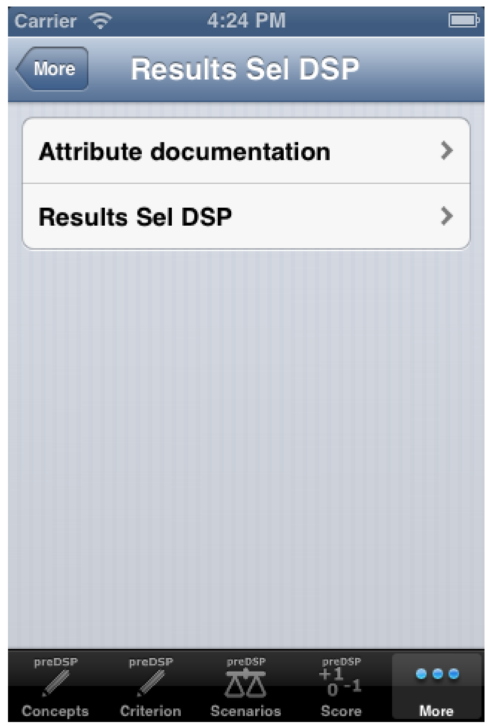
A demonstration of the program is video taped and is available from: `http://www.youtube.com/watch?v=w2qt-r2XkAM&feature=youtu.be`

5.3 Documentation

When the user defines the concepts and criteria of a special case, additional information about the name and description of the given concept or criterion is also specified. The user provides the name, the acronym and a detailed description for each concept and criterion. The acronyms are used in the DSP to simplify the data handling, and the name and description is written to a separate text file for documentation. The documentation can be viewed in the application. Documentation of technologies added, criteria added and the results from the calculations from the Preliminary Selection DSP are accessed from the view in Figure 11a. Documentation of attributes and results from calculations from the Selection DSP can be accessed from the view in Figure 11b. These views are found under the Results Pre DSP and Results Sel DSP tabs in Figure 8.

It is important to justify and document the evaluation of the concepts when giving scores. In the prototype of the DSP interface, the user has to document the reasoning behind the evaluations separately. This text file is not a part of the DSP program, and must be created by the user separately. Documentation for justification of the scores are important to ensure the quality of the DSP method. The validity of the DSP depends on the quality of the information gathered from the concepts. Decision-maker must have good knowledge of the design problem and show good judgment when using subjective, experience-based information to compare technologies.

(a) Documentation from Preliminary Selection DSP



(b) Documentation from Selection DSP

Figure 11: Documentation features for DSP application

6 Case studies

The DSP program is tested with two selected case studies. An introduction is given as motivation for the selected cases, and the DSP method is carried out for the two cases with aid from the DSP program developed in this thesis.

6.1 Introduction to the case studies

Roughly 40% of the world's natural gas have not been produced and a large part of these reserves pose technical and economic challenges [18]. It is estimated that roughly 20-33% of the world's natural gas resources can be classified as sour, with a Carbon Dioxide (CO_2) and Hydrogen Sulfide (H_2S) content ranging from 20-40% and even as high as 70% [11]. CO_2/H_2S are impurities in the natural gas that need to be removed for the gas to be commercially acceptable. CO_2 lowers the heating value of the gas if present and is not desirable to release to the atmosphere due to environmental concerns [10]. H_2S is a highly toxic gas for humans and must be removed for the gas to be sold to consumers [9]. Both CO_2 and H_2S are referred to as acid components because they are corrosive when present with free water. The specification for H_2S content in the sales gas is set to be 4 ppmv [9]. The specification for CO_2 varies with respect to the end product. For pipeline gas the specifications usually range from 2-2.5mol% CO_2 [9]. If the end product is Liquefied Natural Gas (LNG) the specifications become more stringent because CO_2 freezes at higher temperatures than that of the natural gas. This can cause blockage and damage equipment. The CO_2 content in LNG should not exceed about 50 ppmv [9]. Highly sour natural gas reserves pose a technical and economic challenge for producing. When the acid content in the gas increases, the conventional removal process becomes uneconomic and the sour gas reserves are not produced [18]. This is due to large amount of solvents needed to remove the acid gas, large costs, and an increase in energy needed to regenerate the solvents [11]. Several new technologies are being developed to make production of sour gas field economic. The DSP program is used to evaluate these technologies to find the best suited process for removal of acid gas in a highly sour gas field.

Other non-conventional gas reserves include stranded gas locaded at offshore locations [30]. Stranded gas reserves raise the challenges of processing and transporting the gas to the consumer, but with the world's increasing energy demand, motivation for producing these reserves increase [30]. Pipelines become an uneconomic transportation method when the gas is to be transported over long distances and liquefying the natural gas for transportation by ship becomes a better alternative [9]. Floating Liquefied Natural Gas (FLNG) is a possible solution to overcome the economic challenges of producing stranded gas. LNG-FPSO (Floating Production Storage and Offloading) is a floating production vessel that liquefies the natural gas. A LNG-FPSO vessel does pose some technical challenges compared to onshore LNG plant. Space is limited offshore and and the refrigeration cycles must therefore be compact and due to the motion at sea, all processes must resist movement from the ocean [30]. The DSP method is carried out to find the best suitable refrigeration cycle for an LNG-FPSO.

6.2 Cryogenic separation for removal of CO_2 in natural gas streams

The DSP program is in this case used to evaluate the most suited cryogenic separation processes for removal of CO_2 from a natural gas reservoir that contains 30 mol% CO_2. This is a high concentration of acid gas and presents a technical and economic challenge for processing. The processing take place on an offshore platform and the goal is to obtain a gas quality with a 2.5 mol% maximum allowance of CO_2.

Different technologies are available for removal of acid gas from natural gas streams. These include [31]:

- Absorption by solvents
- Adsorption on solids
- Membranes
- Cryogenic separation

Absorption is the most common method for treating sour gas, but in reservoirs with a high concentration of acid gas, the amine process becomes uneconomic. Adsorption processes remove acid gas components by adsorption onto porous material. The adsorption processes are limited to handle CO_2 content less than 2% due to the large quantity of solid adsorbents needed [31]. Membranes remove acid contaminants by permeation of selected components through a membrane. It is more compact than the absorption and adsorption processes, and removal of CO_2 and H_2S is done without phase change [31].

This case study will evaluate the cryogenic separation of CO_2 from natural gas. The natural gas is separated from the acid components by taking advantage of their high relative volatility [10]. Relative volatility relates to the difference in vapor pressures of the components in the natural gas. By decreasing the temperature, the idea behind cryogenic separation is to condense the CO_2 and H_2S at temperatures where the methane still is in a vapor phase. The acid components can then be separated from the methane, and the gas product will mainly be methane. The liquid product will mainly consist of CO_2 and H_2S.

The main challenge for cryogenic separation is solidification of CO_2 from vapor phase at temperatures and pressures required to separate the acid gas from the natural gas. When the CO_2 content in the natural gas exceeds about 5%, some of the carbon dioxide solidifies from vapor phase [28]. This occurs when the natural gas is cooled below the triple point of CO_2, at about -56.6 °C, where CO_2 may exist in gas phase, liquid phase and solid phase [31]. If conventional separation equipment is used, the solidification can cause blockage and will damage equipment. Different technologies are therefore developed to handle the solidification of CO_2. The methods used for cryogenic separation can be categorized into [31]:

- **Gas-Liquid phase separation.** The sour gas is maintained above the triple point of CO_2 at about -56.6 °C so solidification of CO_2 does not occur.
- **Gas-solid phase separation.** The CO_2 is separated from methane below the triple point of CO_2 and CO_2 will solidify from vapor phase.

The DSP method is used to evaluate and compare the available cryogenic separation processes to identify the best suited process for the removal of CO_2 from a highly sour natural gas field located offshore. A Preliminary Selection DSP is first performed to identify the most-likely the succeed technologies, and a Selection DSP is performed to rank the feasible alternatives to identify the best suited alternative.

6.2.1 Preliminary Selection DSP

In the Preliminary Selection DSP, all concepts are identified and given acronyms to simplify the data. The technologies are described and the main advantages and disadvantages are listed. The concepts included in the DSP are given below. These concepts are added to the program for evaluation.

- **CryoCell (CRYO):** The CryoCell is developed by Cool Energy Ltd. and a pilot plant has been installed in Australia at the ARC Energy's Xyris gas field as a joint venture with Shell Global Solutions [7]. The unit consists of a cryogenic contact vessel. The feed gas is dehydrated before entering the contact vessel to ensure that no hydrates are formed due to the low temperature in the vessel . The gas entering the contact vessel is pre-cooled and further cooled from cold reflux entering the contact vessel from the top side [8]. The CryoCell vessel is a specially constructed contact vessel that allows solidification of CO_2. The CO_2 solidifies inside the contact tower and descends to the bottom of the tower. Heat is provided to melt the solid CO_2 so that is mixes with cold liquid and is transported from the column. The methane rich gas exits the column at the top. The CryoCell can handle different gas compositions, but with a CO_2 content of more than 20 mol%, a CryoCell fractionation process is included upstream from the separator to remove excess CO_2 [7]. The CryoCell may not be able to obtain pipeline quality of the gas without further processing [7]. Advantages are that the CryoCell handles solidification of CO_2 and no dehydration of the sweet gas is necessary. The liquid CO_2 is at a high pressure, which reduces the energy needed to compress the CO_2 for futher handling. Disadvantages are that if the CO_2 content exceeds 20% or if the feed gas contains heavier hydrocarbons, extra processing equipment is needed for optimal operation. In addition, the Cryocell is not able to obtain pipeline quality of the gas without furhter processing.

- **Sprex (SPREX):** The Sprex process is developed by Total and IFP (Institute Français du Pétrole). A pilot plant has been built an the onshore Lacq plant in France [18]. The Sprex process is mainly for handling the removal of H_2S from natural gas streams. A Sprex process for bulk removal of CO_2 is under development, but has not yet been demonstrated. This process is called SprexCO2 [5]. The Sprex method removes bulk amounts of CO_2 and H_2S and is used upstream from a conventional amine unit [18]. The gas entering the Sprex removal unit must be dehydrated to prevent hydrate formation. The gas in the Sprex unit is cooled to -65 °C [18]. By placing the Sprex unit upstream from a conventional amine unit, the size of the amine unit is decreased and the amount of amine solvent is also decreased, which will decrease the investment costs and the energy consumption of the amine unit [18]. Advantages with the Sprex unit is that it is able to handle sour gas with over 20% H_2S [18]. It takes up less space than a conventional amine process and the liquid product is at a high pressure, which will reduce the energy needed for re-compression. Disadvantages are that the Sprex will primarily remove H_2S until the SprexCO2 is in operation. The process needs an amine unit to obtain pipeline quality, and dehydration of the sweet gas is therefore necessary downstream from the process.

- **Twister (TWIST):** The Twister process is developed by Shell. Shell and the Beacon Group launched the company Twister B.V. in 2000 [17]. Twister technology is currently focused on gas dehydration and NGL recovery, but it can also be used for bulk removal of CO_2 and H_2S [27]. The technology is demonstrated at the Petronas/Sarawak B11 platform operated by Shell, where the Twister dehydrates the natural gas [17]. The feed gas enters a supersonic separator where static vanes are used to give the gas swirl. The gas is expanded to supersonic velocities through a Laval nozzle and pressure is transformed to kinetic energy. This results in a temperature drop where water and heavier hydrocarbons condenses [1]. The liquids are then separated in a cyclonic separator and the gas enters a diffusor section where pressure is regained [17]. The lean gas can be further treated using a conventional amine unit to obtain pipeline quality. Advantages are that the Twister is a compact cyclonic separator. There is minimum maintenance required, which make this technology attractive on offshore locations. Disadvantages are that the Twister does not obtain pipeline quality of the natural gas.

- **Ryan/Holmes (RH):** The Ryan/Holmes process is developed at Koch Process Systems. The process is used at the Chevron Buckeye CO_2 plant in New Mexico, USA. The plant was built in 1998, and separates a high-content CO_2 gas stream produced from the Chevron Vacuum Field [21]. The Ryan/Holmes process uses an additive to remove acid gas at a low temperature without crystallizing the CO_2. The additive works as a solid-preventing agent, which is added to the feed gas prior to where freeze-out may occur [10]. C_3-C_6 alkanes are typically used as the solid-preventing agent because they are usually found in the feed gas and can be recycled in the process [10]. The feed gas is cooled before it enters a distillation column. The overhead product is pure methane and the bottom product consists of heavier hydrocarbons and CO_2. The bottom products can be further separated to a CO_2 fraction, ethane plus fraction and recycled agent. An improved Ryan/Holmes process has been developed, which lets the process operate at a temperature above the triple point of CO_2 to prevent solidification [26]. This reduces the energy needed to cool the gas. It is done by increasing the amount of additive in the upper part of the column [26]. The Ryan/Holmes process can handle large concentrations of acid gas, and with the improved technology, the process operates at a higher temperature, which reduces the energy needed for refrigeration. Disadvantages are that the compactness of this process suffers from the need of additives, regeneration and multiple columns depending on the amount of CO_2 in the feed gas.

- **Controlled Freeze Zone (CFZ):** The Controlled Freeze Zone process is developed by Exxon. The technology has been demonstrated in a pilot plant and the first commercialized plant, Shute Creek natural gas processing facility, started operation in 2010-2011 [11]. The Controlled Freeze Zone process utilizes a specially constructed distillation tower to control the solidification of CO_2 at cryogenic temperatures [11]. The tower is made up of three parts; the Top Conventional Distillation, the Bottom Conventional Distillation and the Controlled Freeze Zone. The overhead product of the distillation tower is methane and light components, such as nitrogen and hydrogen if present in the feed gas stream. The bottom product will be a rich stream of liquid CO_2. The top and bottom distillation columns operate as conventional distillation columns, where gas is cooled from liquid descending and the heavier components in the gas are condensed. The liquid is heated by the rising gas, freeing light components that evaporates and rises with the gas. The middle section is the Controlled Freeze section. This section handles solidification of CO_2. The solid CO_2 accumulates at the bottom of the CFZ section and is heated to melt the CO_2. The melted CO_2 is directed to the bottom distillation column for methane recovery. Cold liquid from the top distillation column is spray contacted with the CO_2 rich gas. The CO_2 is then solidified in the CFZ section, which is constructed as an open area so the solid formation does not damage or block equipment [11]. Advantages are that the CFZ is compact and the removal of acid gas is done in a single column. There is no need for dehydration of the sweet gas, and liquid CO_2 is removed at high pressures. The process is able to handle large concentrations of acid gas [24]. The CFZ process can obtain pipeline quality of the gas [28] . Disadvantages are that the energy demand is increased due to the need to cool the gas to low temperatures.

- **Amine process (AMINE):** Acid gas removal from natural gas streams by means of alkanolamines is the most common technology used today [12]. Amines are classified as chemical solvents and will effectively remove CO_2 [12]. The amines are dissolved in water, so the feed gas going through the process will be saturated. The saturated feed gas leaving the amine process will then need dehydration before further processing to prevent corrosion problems [12]. The amine process consists of a contactor where lean amine removes CO_2 from the feed gas. The overhead product is sweet gas. Rich amine exits the contactor at the bottom and is sent for recycling in a stripping column. This is done at high temperatures and low pressures [12]. The CO_2 product is gaseous and at low pressure. Advantages are that the amine process is mature and will obtain pipeline quality of the natural gas. Disadvantages are that when the sour content increases, the amine process becomes expensive due to energy demanding regeneration of the amine [12]. The CO_2 is at low pressures, so energy is needed to re-compress the gas for further handling. Dehydration of the sweet gas is also necessary downstream of an amine unit to prevent corrosion. The amine process is not classified as a cryogenic separation process, but is included in the case study because it is the most used method for removal of acid gas today.

The next step of the Preliminary Selection DSP is to identify the criteria for the Preliminary Selection DSP. The case study specifies an offshore location and the sales gas should have a maximum CO_2 content of 2.5 mol%, which is a common specification for pipeline gas. The generalized criteria for the Preliminary Selection DSP are listed below.

- **Complexity (COMPLEX):** For offshore locations, weight and size are important factors to consider. Weight of equipment is linked to the cost of the platform or production vessel and heavy equipment will lead to increased cost. Complexity will also increase with excessive piping and number of equipment needed for the specific technology.

- **Performance (PERFORM):** The quality of the processed gas is important to consider because of the regulations and specifications for sales gas. If the technology considered is not able to meet these specifications, further processing is necessary.

- **Power consumption (POWER):** High power consumption will lead to increased operational costs and will decrease efficiency of the technology. The power consumption is related to regeneration of additives or amines, cooling of the gas and re-compression of the acid components produced.

- **Safety (SAFE):** Safety is an important parameter for any production. This parameter is linked to hazardous and flammable chemicals, such as additives, amines and refrigerants.

- **Maturity (MATUR):** It is important that the technologies evaluated are proven to work. Experience will justify installation of a specific process.

The criteria are added to the the DSP program and their relative importance are given in Figure 12. Five scenarios of the relative importances are used for the post-solution sensitivity analysis. In the first scenario, all criteria are equally weighted. The most important criteria in the second scenario is the safety criteria. In the third scenario, performance is the most important attribute, and in the fourth scenario, the power consumption is deemed as the most important factor. For the fifth scenario it is the complexity of the processes that is the most important criterion.

```
Relative Importance:
---------------------
            Scenario
              1    2    3    4    5
 COMPLEX:   0.2  0.2  0.1  0.1  0.4
 PERFORM:   0.2  0.2  0.4  0.2  0.1
   POWER:   0.2  0.1  0.2  0.4  0.2
    SAFE:   0.2  0.4  0.2  0.2  0.2
   MATUR:   0.2  0.1  0.1  0.1  0.1
----------------------------------------
   Sum:      1    1    1    1    1
----------------------------------------
```

Figure 12: Scenarios for relative importances of criteria for Preliminary Selection DSP: CO2 removal

Scores are given to the concepts for all criteria with respect to the the datum. All concepts are used as datums. The scores can be viewed in Appendix B. From the scores and relative importances of the criteria, the merit functions for the concepts are calculated. The results from the Preliminary Selection DSP is given in Figure 13.

The overall merit functions are presented for all five scenarios added in the Preliminary Selection DSP. The Ryan/Holmes process, the Controlled Freeze Zone process and the Amine process score well over all scenarios. The Amine process is in the top three ratings in four out of five scenarios and the Ryan/Holmes process and the CFZ process are in the top three

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.35	0.267	0.617	0.4	0.55	0.617
2	0.242	0.242	0.45	0.3	0.442	0.717
3	0.242	0.458	0.683	0.3	0.625	0.708
4	0.375	0.308	0.633	0.5	0.525	0.508
5	0.442	0.183	0.508	0.6	0.525	0.433

Figure 13: Merit functions for Preliminary Selection DSP: CO2 removal

ratings in all five scenarios. The Amine process has the highest merit functions in scenario 2 and 3, where safety and performance are the parameters of highest importance, respectively. The Ryan/Holmes process and the Amine process have the highest merit functions in scenario 1 where all criteria are of equal importance. In scenario 4, power consumption has the highest weighting and the Ryan/Holmes process receives the highest merit function, followed by the CFZ process and the Twister process. The Twister process has the highest merit function in scenario 5, where the complexity of the process is the most important criterion. The CryoCell process and the Sprex process does not perform as well throughout the scenarios and have lower merit functions compared with the other technologies. The Ryan/Holmes process and the Controlled Freeze Zone process have the overall highest merit functions throughout all five scenarios, and is included as the most-likely to succeed technologies that is further evaluated in the Selection DSP. The Amine process receives an high merit functions in four out of five scenarios and is also included as one of the feasible alternatives in the Selection DSP.

The alternatives that will be evaluated in the Selection DSP are the Ryan/Holmes process, the Controlled Freeze Zone process and the Amine process.

6.2.2 Selection DSP

The merit functions of the concepts in the Preliminary Selection DSP reveals that the most feasible technologies for all scenarios are the Controlled Freeze Zone process, the Ryan/Holmes process and the Amine process. These technologies scores as the top three concepts in four out of five scenarios. The CFZ process and the Ryan/Holmes process have one of the three highest merit functions in all five scenarios. From the Preliminary Selection

DSP, the CFZ process, the Ryan/Holmes process and the Amine process are chosen to act as the most-likely to succeed alternatives in the Selection DSP. These are added to the program as alternatives.

The attributes of the Selection DSP are defined to be:

- **Environmental (ENVIRON):** Environmental concerns relate to the need for hazardous additives or refrigerants in the processes. Interval scale. Range of rating is 0-8. Low values are preferred.

- **Compactness (COMPACT):** The compactness of the technology is important for offshore locations because space and weight are factors that will increase cost. The layout of the process is considered. Interval scale. Range of rating is 1-9. Low values are preferred.

- **Number of equipment (EQUIP):** This attribute is related to the reliability and maintainability of the technology. A large number of units will decrease the reliability of the process and increase the maintenance necessary. The equipment includes dehydration units, heat exchangers, distillation columns, regeneration/recovery units and compressors or pumps. Ratio scale. Range of rating is 1-14. Low values are preferred.

- **Power consumption (POWER):** This attribute is related to the energy needed for the process. Cooling, heating, regeneration and compression are included as sources for energy. Interval scale. Range of rating is 1-10. Low values are preferred.

- **Operability (OPER):** This attribute is related to how complex the process is, and the need for monitoring and staffing to operate the process. It is also related to the stability of the process. Interval scale. Range of rating 0-7. Low values are preferred.

- **Sensibility to motion (SENS):** This attribute relates to how motion from waves, wind etc. will affect the performance of the technology. Interval scale. Range of scale 0-5. Low values are preferred.

The attributes are ranked according to their importance. Compactness is ranked as the most important attribute, while Environmental concerns are viewed as the least important attribute for this case. Power consumption and sensibility to motions are also ranked high. The power consumption should ideally be as low as possible to increase the efficiency of the processes and decrease the cost related to operation of the processes. The case study is specified to be at an offshore location and sensibility to motion should be low to ensure that the technologies performs as expected to remove CO_2 from the natural gas stream. The attribute ranking is shown in Table 15.

The type of scales and the range of the scales used are identified with the list of attributes for the Selection DSP. This input is given to the program and the alternatives are rated on the attribute scales. A summary of the alternative ratings and the attribute scales are given in Table 15. This information is given to the program, which calculates the normalized rating values and the merit functions for the alternatives.

Table 15: Summary of alternative rating and attribute scales and rankings: CO2 removal

Alternatives	Attributes					
	ENVIRON	COMPACT	EQUIP	POWER	OPER	SENS
AMINE	6	7	7	9	6	4
RH	5	6	6	4	6	4
CFZ	4	4	5	6	4	2
Attribute summary:						
Relative importance	1	6	3	5	2	4
Type	Interval	Interval	Ration	Interval	Interval	Interval
Preference	Low	Low	Low	Low	Low	Low
Upper Bound	8	9	14	10	7	5
Lower Bound	0	1	1	1	0	0

The final ranking is shown in Figure 14. The Controlled Freeze Zone process has the highest merit function. It is 25% higher than the merit function of the Ryan/Holmes process and 58% higher than the merit function of the Amine process. The CFZ process has a clear advantage over the other feasible alternatives and is chosen as the best suited alternative for the process for removal of 30mol% CO_2 for the offshore production platform.

6.3 Liquefaction processes for a Liquefied Natural Gas (LNG) project

Liquefied Natural Gas (LNG) is used to transport natural gas [9]. If the natural gas is to be transported over a long distance, it becomes more economic to liquefy the gas and transport

```
---------------
Final ranking:
---------------
 AMINE: 0.238
    RH: 0.423
   CFZ: 0.562
```

Figure 14: Merit functions for Selection DSP: CO2 removal

the LNG by ship, instead of transporting gas by pipeline. LNG takes up less space than natural gas. $1m^3$ of LNG corresponds to approximately $600Sm^3$ of natural gas [9]. About 33% of the world's natural gas reserves are found offshore [13]. The natural gas in these fields must be processed and brought to shore, and this poses technical and economic challenges. If stranded gas is to be produced, an alternative is to liquefy the natural gas on an offshore production vessel for transportation [30]. A LNG-FPSO is a floating production vessel for LNG production, and space is therefore limited. A LNG-FPSO will be similar to an onshore peak-shaving LNG plant in size [30]. The peak-shaving plants usually have a liquefaction capacity of about 200 tons/day [9]. This is equivalent to a medium scale LNG plant.

The technologies available for the medium scale LNG plants can be categorized into two main groups. They are [30]:

- Mixed refrigerant (MR) technologies
- Expansion-based technologies

The mixed refrigerant processes use a mixture of refrigerants to closely match the composition curve of the natural gas to efficiently liquefy the natural gas. The natural gas is liquefied at a gliding temperature, and the mixed refrigerant follows this curve as closely as possible [9]. This is done by using several refrigerants with different evaporating temperatures so that the mixture will vaporize at a gliding temperature as well [9]. The expansion-based technologies liquefy the natural gas by compression and work-expansion of a gas stream. The refrigerant is compressed and then expanded to low temperatures, which is used to liquefy the natural gas [6]. The refrigerant used in expander processes are in a gaseous phase throughout the refrigeration cycle [30].

A Preliminary Selection DSP is carried out for the available technologies for an LNG-FPSO to uncover the most-likely to succeed processes. A Selection DSP will then rank the feasible alternatives to find the best suited alternative for a LNG-FPSO.

6.3.1 Preliminary Selection DSP

The available technologies are described in detail. The main process characteristics, advantages and disadvantages are listed.

- **Single Mixed Refrigerant (PRICO):** The Prico process is a simple liquefaction process. Mixed refrigerant is compressed to about 30 bar and is partially condensed in heat exchanger by air or seawater to approximately 12°C [9]. The partially condensed refrigerant is then sent to the main heat exchanger that may be a plate fin heat exchanger. In the main heat exchanger, natural gas enters parallel to the refrigerant. The high pressure refrigerant and natural gas are cooled to about -155°C and liquefied in the heat exchanger [9]. The refrigerant leaving the hot side of the heat exchanger is sent through a Joule-Thomson valve where the pressure is decreased from 30 bar to 5 bar. This produces a temperature drop of about 0.5K [9]. The refrigerant then enters the cold side of the main heat exchanger and exits the exchanger at about 6.5°C [9]. To optimize the process, the refrigerant pressures, flow rate and composition can be adjusted. A high refrigerant circulation rate is needed to liquefy the natural gas and this contribute to a relative high power consumption [9]. The Prico process requires a specific power consumption of approximately 403 kWh/ton LNG produced [14]. The Prico process is developed by Black & Veatch and is a relevant liquefaction process for peak-shave or medium scale LNG plants [9]. The PRICO process has several industrial references, and 25% of the peak-shaving LNG plants use the PRICO process in the United States [30].

- **Linde Basic Single Flow (LBSF):** The Linde Basic Single Flow LNG process consists of a plate-fin heat exchanger which liquefies the natural gas to LNG [3]. A single mixed refrigerant cycle is used to liquefy the natural gas. The mixed refrigerant is compressed in a turbo compressor and partially condensed by an air or seawater heat exchanger [3]. The partially condensed refrigerant then enters a separation vessel. The vapor is sent to the main heat exchanger where it enters in parallel with the natural gas and is expanded through a Joule-Thomson valve. It then re-enters the main heat exchanger counter-current to the natural gas and is further re-compressed in the turbo compressor. The liquid refrigerant exits the separator at the bottom and is sent to the main heat exchanger where is enters parallel to the natural gas. It is further throttled in a Joule-Thomson valve and redirected to the main heat exchanger counter-current to the natural gas before it is returned for re-compression [3]. By use of a separator in the refrigeration cycle, the energy efficiency of the process is increased and power consumption is reduced [13]. The Linde Basic Single Flow is developed by Linde and is used at the Kollsnes LNG plant [3].

- **(AP-M^{TM}) Process (AP-M):** The AP-M process is developed by Air Products & Chemicals [30]. The process uses a coil wound heat exchanger to liquefy the natural gas [30]. A single mixed refrigerant is used as the working medium and the refrigerant is evaporated at two different pressure stages [30]. The mixed refrigerant is compressed and partially condensed in a heat exchanger before it enters a separator, and vapor and liquid are separated. The liquid and vapor enters the coil wound heat exchanger separately in parallel to the natural gas. As the vapor is condensed in the main heat exchangers, it directed to a second separator where liquid and vapor are separated and redirected to the main heat exchanger. The dual pressure process makes the AP-M process energy efficient, by allowing a reduction in size of the heat exchanger and compressor [30]. Air Products & Chemicals has experience in developing liquefaction technologies, but the AP-M process has not been used in any LNG plants previously [30].

- **Linde Multistage Mixed Refrigerant (LiMUM):** Linde has developed the LiMUM process, which consist of a coil wound heat exchanger where the natural gas is first pre-cooled, liquefied and then sub-cooled in three different stages by a single mixed refrigerant [30]. A medium pressure separator is used for the pre-cooling of the natural gas at the bottom of the coil wound heat exchanger. A high pressure separator is used to liquefy the natural gas and a low pressure separator is used to sub-cool the natural gas at the top of the coil wound heat exchanger [3]. The refrigerant is compressed in a two-stage compressor with intercooling, and aftercooling with air or seawater [3], which allows the process to have a high capacity of 2.5 MTPA (Million Tonnes per Annum) [30]. The LiMUM process has several industrial references. The process is used at the Shan Shan LNG plant in China, and at a LNG plant in Kwinana, Australia [3].

- **Kryopak SCMR (K-SCMR)** This process is a single mixed refrigerant process developed by Kryopak and is designed for peak-shaving LNG plants [4]. The energy consumption of the Kryopak SCMR is reported to be approximately 300 kWh/ton LNG produced [4]. A single brazed aluminium plate fin heat exchanger functions as the main heat exchanger in the process and the compressors used are electric motor driven centrifugal compressors [4]. The Kryopak SCMR process is a simple process with high operational reliability. The typical startup time after shutdown ranges from three to six hours [4].

- **Pre-cooling + SMR (K-PCMR):** The Kryopak PCMR consists of a pre-cooling stage where the mixed refrigerant is partially condensed by ammonia or propane and separated before entering a plate-fin heat exchanger [30]. The vapor from the mixed refrigerant is fully condensed in the main heat exchanger before it enters an expansion valve where the temperature drops. This cold stream is used to cool the mixed refrigerants and the natural gas before it is mixed with the mixed refrigerant liquids and re-compressed. The Kryopak PCMR allows for a specific power consumption of approximately 312 kWh/ton LNG produced [14]. It has a capacity of less than 0.1 MTPA [30].

- **Single Nitrogen Expander (SNExp):** The single cycle nitrogen expander is a simple expander process where all refrigerant is expanded to low temperatures to liquefy the natural gas. Nitrogen is used as the refrigerant [25]. The nitrogen is compressed and cooled in an intercooler. Next, it is compressed and cooled in an aftercooler. The high pressure nitrogen is sent to the main heat exchanger parallel to the natural gas. The high pressure nitrogen is then expanded to low pressures and temperatures. The cold, low pressure nitrogen is directed to the main heat exchanger in a counter-current flow from the natural gas where the nitrogen liquefies the natural gas. The nitrogen refrigerant is in a gaseous phase throughout the cycle and this will reduce the process' efficiency. Since all the refrigerant is expanded to the lowest temperature to liquefy the natural gas, the temperature difference between the hot side and the cold side is large, which decreases the efficiency and the compressor work is high. The small-scale LNG plant Snurrevarden in Norway use the Single nitrogen expander [25]. The capacity of the LNG plant is 0.02 MTPA and the energy consumption needed to liquefy the natural gas is about 780 kWh/ton LNG produced [25]. Advantages with the expander process is that startup and shutdown processes are simple and rapid. The expander process is simple with a low equipment number, which increases its reliability. The process is also light weight and compact and safety is increased due to nonflammable refrigerant. It is also insensitive to motion, which is an important parameter for offshore production. Disadvantages are that the power consumption suffers from the single expansion of the refrigerant and the efficiency is reduced [30].

- **Dual Nitrogen Expander (DNExp):** The Dual Nitrogen Expander cycle uses two expanders to increase the efficiency of the expander process [25]. The expanders operate over two temperature levels, which allows for closer matching of the temperature difference between the hot side and the cold side in the heat exchanger. In the dual expander process, only the required amount of refrigerant is expanded to the lowest temperatures to sub-cool the natural gas [25]. The remaining of the refrigerant is expanded at warmer temperatures [25]. The power consumption of a dual nitrogen expander process used at the Kollsnes II LNG plant in Norway is reported to be about 510 kWh/ton LNG [25]. This is a significant reduction of energy needed compared to the single nitrogen expander process.

- **Open Expander Processes (K-EXP):** The Kryopak EXP process uses cold gas flashed from the liquid product as refrigerant to liquefy the natural gas [2]. The gas is isentropically expanded in a semi-closed loop. By using the product gas as the refrigerant, no mechanical refrigerant is necessary, which simplifies the process and decreases the space needed [2]. The refrigerant is in a gaseous phase throughout the cycle, which reduce the equipment necessary in the process. The need for premixing refrigerant is eliminated in the Kryopak EXP process and the process can therefore handle variations and fluctuations in the gas composition and flow rate [2]. A typical K-EXP process has a production capacity of 125 tons LNG/day with a total refrigeration energy consumption of 2550 kW [2]. This is equivalent of a specific power consumption of about 490 kWh/ton LNG.

A summary of the available technologies are given in Table 16 [30]:

A LNG-FPSO is an offshore LNG production vessel. Offshore locations pose technical challenges that a LNG-FPSO must overcome for safe and economic production. For an FPSO, space is limited to the vessel. Safety is also an important parameter when evacuation possibilities are limited. Use of flammable process components and flaring of gas decrease the

Table 16: Summary of liquefaction technologies

Technology	Process	Advantages	Disadvantages
Mixed Refrigerant	PRICO LBSF AP-M LiMUM K-SCMR K-PCMR	- High efficiency - Flexibility to changes in feed gas composition	- Flammable refrigerant - Increased space - High equipment number - Complex operation - Sensitive to motion
Expansion-based	SNExp DNExp K-EXP	- Nonflammable refrigerant - Compact - Simple, low equipment number - Insensitive to motion - Easy operation	- Higher refrigerant flowrate - Lower efficiency

safety, and must be taken into account. Motion sensibility is a factor that must be included offshore. The process performance of the technologies should not suffer from motions in the ocean. The technologies should be easy to operate, reliable and easy to maintain [30]. The following criteria have been identified to incorporate the important factors of a LNG-FPSO:

- **Power consumption (POWER):** Power consumption is important to reduce the operational costs of the LNG-FPSO. Increased power consumption decreases the efficiency of the process, and the power consumption should therefore be as low as possible. Specific power consumption is given in kWh/ton LNG.

- **Complexity (COMPLEX):** The complexity of the process is a measure for the size and weight of the process. This should be as low as possible for an offshore production unit where space is limited. Weight is also directly affecting cost. This criterion is also related to the maintainability of the process.

- **Sensibility to motion (SENS):** This criterion relates to how motions from the ocean will affect the performance of the technologies.

- **Safety (SAFE):** Safety is a very important factor at offshore locations where evacuation possibilities are limited. Use of flammable components affect the safety of the production vessel.

- **Reliability (REL):** This criterion relates to the operability of the technology, as well as how the technology is affected by shutdowns and startups. This criterion can also be related to the maturity and industrial experience of the processes.

Five scenarios of the relative importances of the criteria are used to find the most-likely to succeed technologies. The relative importances are shown in Figure 15.

Relative Importance:

	Scenario 1	2	3	4	5
POWER:	0.2	0.3	0.1	0.4	0.2
COMPLEX:	0.2	0.1	0.2	0.2	0.3
SENS:	0.2	0.2	0.2	0.1	0.2
SAFE:	0.2	0.3	0.3	0.2	0.1
REL:	0.2	0.1	0.2	0.1	0.2
Sum:	1	1	1	1	1

Figure 15: Relative Importance of criteria in Preliminary Selection DSP: LNG-FPSO

In the first scenario, all criteria are assigned equal importance. In the second scenario, the criteria power consumption and safety are the most important, while reliability and complexity are deemed least important. Safety is the most important criterion in the third scenario and power consumption is the least important criterion in this scenario. In the fourth scenario, power consumption is the most important criterion and sensibility to motion and reliability are given the lowest weights. In the last scenario, complexity is the most important criterion. In this scenario, safety is the least important criterion. The criteria are weighted differently in five scenarios to evaluate the merit functions. It may not be clear which criterion is the most important, and by creating multiple scenarios, the merit functions of the concepts become more nuanced and decision-maker may reveal the most-likely to succeed technologies with more confidence.

The identified concepts are scored on how well they perform on the criteria included with respect to each concept acting as the datum in turn. The result from the Preliminary Selection DSP is calculated in the DSP program and shown in Figure 16.

From the overall merit functions in Figure 16, the K-EXP, SNExp and DNExp score well for all scenarios. The K-EXP process has the top merit function in all five scenarios. The

----
Overall merit functions:

Scenario	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
1	0.478	0.256	0.378	0.144	0.244	0.311	0.5	0.456	0.6
2	0.317	0.239	0.322	0.217	0.322	0.356	0.539	0.55	0.644
3	0.439	0.2	0.333	0.0722	0.144	0.211	0.594	0.533	0.667
4	0.456	0.367	0.4	0.289	0.444	0.511	0.4	0.389	0.556
5	0.578	0.328	0.4	0.144	0.267	0.367	0.461	0.378	0.583

Figure 16: Results from Preliminary Selection DSP: LNG-FPSO

SNExp process is in the top three merit functions in four out of five scenarios and the DNExp process is in the top three merit functions in two out of five scenarios. The PRICO is also placed in the top three merit functions in two out of five scenarios. In scenario 2, power consumption and safety are the most important criteria. In this scenario, the K-EXP has the highest merit function. The K-EXP process does not use refrigerants to liquefy the natural gas, but rather a portion of the product liquefied natural gas. The expander technologies are less efficient than the mixed refrigerant processes, but they are more safe to use. Safety is the most important criterion in scenario 3 and in this scenario the K-EXP process has the highest merit function, followed by the DNExp process and the SNExp process. The SNExp and the DNExp process use nitrogen as refrigerant to liquefy the natural gas. This is a more safe option than the mixed refrigerants, which consists of hydrocarbons that are flammable. The K-EXP process uses flashed gas from the liquid product to liquefy the natural gas, which also minimizes the hydrocarbon amount in the process. In scenario 5, complexity is the most important criterion, and the K-EXP, PRICO and SNExp processes have the highest merit functions. These are simple processes with a low equipment count. The K-EXP has the highest merit function in scenario 4, where power consumption is the most important criterion. The K-EXP does not have the lowest power consumption of the evaluated technologies, but in this scenario, complexity and safety are weighted higher than reliability and the K-EXP process performs well on these criteria. The LBSF, AP-M, LiMUM, K-SCMR and K-PCMR processes does not perform as well as the expander technologies and the PRICO process. They use mixed refrigerants to liquefy the natural gas, which can pose a safety hazard at an offshore location. They are also more complex with higher equipment count, which may lead to higher maintenance of the processes. The mixed refrigerant processes are sensible to motion as well, which is an important factor for offshore usage. The mixed refrigerant processes liquefy the natural gas in both liquid and gas phases, and this could lead to maldistribution of liquids that could affect the performance of the processes.

The most-likely to succeed alternatives that are chosen from the Preliminary Selection DSP are the K-EXP process, the SNExp process, the DNExp process and the PRICO process. These will be further evaluated in the Selection DSP.

6.3.2 Selection DSP

The feasible alternatives in the Selection DSP are chosen to be the K-EXP process, the DNExp process, the SNExp process and the PRICO process. These have the overall highest merit functions from the Preliminary Selection DSP. The feasible technologies have been described in the Preliminary Selection DSP section. The relevant attributes for the Selection DSP are identified to be:

- **Power Consumption (POWER):** Power consumption is an important attribute to ensure high efficiency of the process and minimize operational costs. The power consumption of the liquefaction technologies are usually given as specific power consumption in kWh/ton LNG. Low values of power consumption are preferred. Ratio scale. Range of scale is from 200-1000 kWh/ton LNG.

- **Equipment number (EQUIP):** The number of equipment relates to the need for maintenance of the process and the failure rate of the process. Low equipment count in the LNG process can reduce the maintenance and decrease the chance of failure in the liquefaction process. The main equipment of the LNG process include compressors, turbines and heat exchangers. Low values are preferred. Ratio scale. Range of scale is 2-8.

- **LNG production (PRODS):** This attribute is related to the production capacity for the liquefaction process. Interval scale. Range of scale is 1-10. High values are preferred.

- **Space requirement (SPACE):** For offshore production, space is a limiting factor. High space requirement can also mean that a larger vessel is required. This will lead to higher costs. Layout of the process is therefore important to reduce the space necessary. Low values are preferred. Interval scale. Range of scale is 1-10.

- **Complexity (COMPLEX):** In this attribute, the start up and shutdown complexity of the process is evaluated. It is important that downtime and start up time is minimized to reduce loss in production. Interval scale. Range of scale is 1-10. Low values are preferred.

- **Experience (EXPER):** It is important that the processes are proven to work and that they have operational experience. Interval scale. High values are preferred. Range of scale is 1-6.

- **Safety (SAFE):** Safety is a very important attribute for offshore production. This attribute relates to the need of flammable refrigerant to liquefy the natural gas. This can pose a safety hazard if an explosion should occur due to the difficulty in evacuation. Interval scale. Range of scale is 1-6. High values are preferred.

A summary of the information provided to the DSP program in the Selection DSP is given in Table 17.

From Table 17, the attribute with the highest rank of importance is safety. Safety is an important attribute, especially for offshore locations, where evacuation possibilities are limited and available space is minimized. Power consumption is also ranked with high importance. Low power consumption for a liquefaction process is important to minimize operational costs and ensure a high efficiency of the process. The LNG production capacity is the attribute with the lowest rank of importance. A LNG-FPSO is similar to a medium-scale LNG plant with a typical production capacity of 200 tons/day. This is not the highest production capacity available for LNG plants, with onshore base-load plants able to produce more LNG, but these large-scale plants are not suitable for offshore production due to the large area necessary. Space requirements are also an important attribute and is ranked as the third most important attribute. Space is a limiting factor on a LNG-FPSO because the liquefaction process is installed on a floating production vessel and space must be made for other modules as well. The attribute Equipment number is ranked as the fourth most important attribute followed by the Complexity of the process. The equipment number of the process

Table 17: Summary of alternative rating and attribute scales and rankings: LNG-FPSO

Alternatives	Attributes						
	POWER	EQUIP	PRODS	SPACE	COMPLEX	EXPER	SAFE
SNExp	800	5	2	2	4	3	4
DNExp	600	6	8	3	5	2	4
K-EXP	500	6	4	3	5	2	4
PRICO	400	3	8	5	6	5	1
Attribute summary: Relative importance	6	4	1	5	3	2	7
Type	Ratio	Ratio	Interval	Interval	Interval	Interval	Interval
Preference	Low	Low	High	Low	Low	High	High
Upper Bound	1000	8	10	10	10	6	6
Lower Bound	200	2	1	1	1	1	1

relates to the maintenance and failure of the process, which can be costly and complex at an offshore location. Equipment number should therefore be as low as possible. The complexity of the process relates to the start up and shutdown processes of the liquefaction process. The start up time after a shutdown should be as short as possible to avoid long pauses in production. The operational experience of the liquefaction processes does not receive a high rank of importance. All alternatives have been commercially demonstrated at onshore locations and are proven technology.

From the ratings in Table 17, the DSP program calculates the normalized ratings of the alternatives. The normalized ratings are shown in Figure 17.

The SNExp process has the lowest rating and a reported power consumption of 780 kWh/ton LNG. This is the highest power consumption of the technologies. For the equipment number attribute, the DNExp and K-EXP have the lowest normalized ratings. These processes have a higher equipment count than the SNExp and PRICO processes. The PRICO process has the highest rating for this attribute and consists only of a main heat exchanger, a seawater or air heat exchanger and a compressor. One of the advantages of the expander processes for liquefying natural gas, is their simplicity for small liquefaction plants. The expander processes have an advantage over the mixed refrigerant technologies in that the refrigerant used in expander technologies usually is nitrogen in a gas phase. The mixed refrigerant tech-

Normalized attribute rating:

Attributes:	SNExp	DNExp	K-EXP	PRICO
POWER:	0.25	0.5	0.625	0.75
EQUIP:	0.5	0.333	0.333	0.833
PRODS:	0.111	0.778	0.333	0.778
SPACE:	0.889	0.778	0.778	0.556
COMPLEX:	0.667	0.556	0.556	0.444
EXPER:	0.4	0.2	0.2	0.8
SAFE:	0.6	0.6	0.6	0

Figure 17: Normalized Ratings from Selection DSP: LNG-FPSO

nologies are more complex in start up due to the need of mixing the refrigerants [25]. The SNExp, DNExp and K-EXP processes therefore receive a higher normalized rating for the complexity attribute than the PRICO process. The complexity of the expander processes will however increase as the production capacity increases. The expander processes need a higher circulation of refrigerants and combined with the higher power consumption, the maximum train size of an expander process should not exceed that of a production of 1 MTPA [25]. The expander processes are given high rating for safety because they do not use flammable refrigerants to liquefy the natural gas. The PRICO process is a mixed refrigerant process, which makes it less safe because the mixed refrigerants consists of flammable hydrocarbons. The PRICO process is the process with the most experience. The expander technologies are not as much used, and they receive a lower normalized rating than the PRICO process.

From the normalized ratings and the relative importances of the attributes, the merit functions of the alternatives are calculated. The results from the Selection DSP are shown in Figure 18.

The K-EXP process receives the highest ranking of the feasible alternatives in the Selection DSP with a 2.5% higher merit function than that of the DNExp process, which has the second highest merit function. The SNExp process has a 3.3% lower merit function than the K-EXP and the PRICO process has a 8% lower merit function. From the merit functions, the K-EXP process seems like the best choice, but there is very little difference between the

```
---------------
Final ranking:
---------------
 SNExp: 0.538
 DNExp: 0.545
 K-EXP: 0.556
 PRICO: 0.512
```

Figure 18: Merit function from Selection DSP: LNG-FPSO

merit functions of the alternatives. Only 2.5% difference separates the K-EXP process and the DNExp process, and this is not enough to make a decision of the best suited alternative. A post-solution sensitivity analysis is carried out to see if a change in the relative importance of the attributes will affect the results. Information from the normalized ratings and the normalized relative importances are used for the post-solution sensitivity analysis. From the normalized ratings in Figure 17, the K-EXP process has a higher normalized rating for the power consumption than the DNExp process. The relative importance of the power consumption attribute is decreased with 20% to see the result is affected. The DNExp process has a higher normalized rating for the production capacity attribute than the K-EXP, and the relative importance of the production attribute is increased with 20%. The merit functions are recalculated and the results are shown in Table 18

Table 18: Post-solution sensitivity analysis: LNG-FPSO

Alternative	20% Decrease POWER	20% Increase PRODS	20% Decrease POWER 20% Increase PRODS
SNExp	0.527	0.538	0.528
DNExp	0.524	0.551	0.529
K-EXP	0.529	0.559	0.532
PRICO	0.479	0.517	0.485

The K-EXP process has the highest merit function when the relative importance of the power consumption attribute decreases with 20%, but the second highest merit function now belongs to the SNExp process. When the relative importance of the LNG production attribute increases with 20% the difference in merit functions between K-EXP and DNExp becomes 1.4% and the merit function of the K-EXP process remains the highest. The K-EXP process continues to have the highest merit function when the power consumption decreases with 20% and the LNG production attribute increases with 20%. The difference between K-EXP and DNExp now becomes 0.45% and the order of the ranking does not change. The post-solution sensitivity analysis reveals that the merit functions of the alternatives are too close to make a final decision. The margin between the alternatives are small and the most suited technology for the LNG-FPSO cannot be identified without adding more attributes to further differentiate the alternatives.

6.4 Use of computational tool to perform DSP Technique

The DSP method is versatile because it can be used for a wide range of areas. The DSP method could be used in all engineering fields where a set of criteria for an engineering problem can be used to accept or reject available solutions. The DSP method may not be limited to engineering applications. Contract evaluation could also be an area in which the DSP method could prove useful. The DSP method has previously also been used with success in aircraft design and ship design among others.

The validity of the DSP method strongly depends on the quality of the information gathered for technologies evaluated and the decision-maker's knowledge of the engineering problem. The results from the DSP will not be better than the information used. Sufficient documentation of all choices made in the DSP is crucial to validate the method and justify results. The documentation should include information and reasoning behind relative importances, scores given to the concepts and ratings of the alternatives. Scales for attributes in the Selection DSP should also be documented in detail and care must be taken to ensure that the range of attribute scales include all technologies available, and not only feasible alternatives included in the Selection DSP. Documentation and justification behind the DSP is also important if there are several people involved in the decision-making process.

The DSP computational tool reduces the time needed to perform calculations necessary in the DSP method, and by use of the computational tool, the risk of human error regarding calculations is reduced as well. The time spent performing the DSP method for the selected case studies was significantly reduced by letting a computer perform all calculations.

The case of selecting a suitable cryogenic separation process for removal of CO_2 in a gas field containing 30 mol% CO_2 is compared to a similar case performed in the report "Evaluation of processes for removal of CO_2/H_2S from natural gas streams" [23]. The results obtained in the report supports the results given from the computational tool. This validates the results provided from the calculations in the computational tool, and validates the goal of the program to be able to make correct calculations from the input given.

7 Conclusion

In this thesis, a computational tool for the Decision-Support Problem Technique is developed to aid a decision-maker in calculations required in the DSP method. An object-oriented analysis provides means to create a structured system of objects with attributes and operations to make the software coding a more manageable task. The computational tool is developed in C++ and results from the program are written to a text file for the user to evaluate. A prototype user interface is developed as an application for iPhone. The computational tool is tested in two case studies where the first case is the selection of a cryogenic separation process for removal of CO_2 in a natural gas field containing 30mol% CO_2. The second case study is the selection of a liquefaction process for offshore LNG production. The first case study yields that the Controlled Freeze Zone process is the best suited process for the problem statement. The second case study does not provide any clear results of the best suited process. Additional attributes for the Selection DSP should be added to differentiate the liquefaction processes further.

Testing of the computational tool validates that the program gives correct results from calculations, and provides support to the hypothesis that a computational tool can aid a decision-maker in calculations in the DSP. By letting a computer perform calculations, the DSP method becomes more efficient and less time consuming. The decision-maker is then responsible for providing information to the program and evaluating the results of the DSP method.

7.1 Further work

For further work, the development of a computational tool for the Compromise DSP is recommended. The Coupled DSP method is used when a design problem is a combination of either Selection - Selection, Compromise - Compromise or Selection - Compromise. A computational tool for these types of design problems could also be developed.

The computational tools of the Selection DSP, Compromise DSP and Coupled DSP could be combined in an extensive DSP program that includes means for sufficient documentation of engineering problems and choices behind the evaluation of technologies. A user interface should allow the user to document all choices made in the DSP process to justify and validate results.

References

[1] Factsheet 1. How does Twister work? Available from: http://twisterbv.com/PDF/resources/Twister_-_How_Does_It_Work.pdf. Accessed 28.05.2013.

[2] Kryopak PCMR and EXP LNG Process. Max Universal Inc. Available from: http://www.max-universal.com/KryopakPCMR_eng.html. Accessed 20.05.2013.

[3] LNG Technology. Available from: Linde Engineering. http://www.linde-engineering.com/internet.global.lindeengineering.global/en/images/LNG_1_1_e_10_150dpi19_4577.pdf. Accessed 28.05.2013.

[4] Max Univeral's Proprietary LNG Process Max SCMR LNG process. Max Universal Inc. Available from: http://www.max-universal.com/scmr.html. Accessed 20.05.2013.

[5] Sour Gas. A History of Expertise. Available from: http://www.total.com/MEDIAS/MEDIAS_INFOS/239/EN/sour-gas-2007.pdf. Accessed 28.05.2013.

[6] Terry R. Tomlinson Adrian J. Finn, Grant L. Johnson. LNG Technology for offshore and mid-scale plants. 79th Annual GPA Convention, 2000.

[7] Nimalan Gnanendran Allan Hart. Cryogenic CO2 Capture in Natural Gas, 2009.

[8] R. Amin and T. Kennaird A. Jackson. The Cryocell: An Advanced Gas-Sweetening Technology. International Petroleum Conference in Doha, Qatar, 2005.

[9] Jostein Pettersen Olav Bolland Arne Fredheim, Even Solbraa. *TEP 4185 Industrial Process and Energy Technology.* 2011.

[10] James M. Ryan Arthur S. Holmes. Cryogenic Distillative Separation of Acid Gases from Methane. Koch Process Systems, Inc., 1979.

[11] P.S. Northrop C.J. Mart B.T. Kelley, J.A. Valencia. Controlled Freeze Zone for developing sour gas reserves, 2011.

[12] John M. Campbell. *Gas Conditioning and Processing: Gas Treating and Sulfur Recovery.* John M. Campbell & Company, 1998.

[13] Luis Castillo. Decision Support Problems (DSPs): Review and application in a Liquefied Natural Gas (LNG) Project.

[14] Erica C. Carvalho Christian D. T. Begazo and José R. Simões-Moreira. Small-scale LNG plant technologies. In *Hydrocarbon World 2007*. Touch Briefings, 2007.

[15] Robert Clair. *Learning Objective-C 2.0 : A hands-on guide to Objective-C for Mac and iOS developers*. Addison-Wesley Professional, 2 edition, 2012.

[16] F. Mistree D. Muster. The Decision-Support Problem Technique in Engineering Design. *International Journal of Applied Engineering*, 4:23–33, 1988.

[17] Salim Sibani Dr. Fred T. Okimoto and Michael Lander. Twister Supersonic Gas Conditioning Process. Society of Petroleum Engineers in the 9th Abu Dhabi International Petroleum Exhibition and Conference, 2000.

[18] C. Streicher F. Lallemand, F. Lecomte. Highly Sour Gas Processing: H2S Bulk Removal With the Sprex Process. In *International Petroleum Technology Conference, I.P.T. Conference*. 2005.

[19] Bert Bras Farrokh Mistree, Owen F. Hughes. *The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm*, volume 150, pages 247–286. Amer Inst of Aeronautics, 1993.

[20] Luke Stonis Farrokh Mistree, Kemper Lewis. Selection in the Conceptual Design of Aircraft, 1994.

[21] Mark Garner. Chevron Buckeye CO2 Plant Treating of Naural Gas using the Ryan/Holmes Separation Process, 2008. In Industrial Technology, University of Texas.

[22] Ian Graham. *Object-Oriented Methods*. Pearson Education Limited, 3 edition, 2001.

[23] Caroline Bauge Gulliksen. Evaluation of processes for removal of CO2/H2S from natural gas streams, 2012.

[24] Robert D. Denton Jaime A. Valencia. Method and Apparatus for Separating Carbon Dioxide and other Acid Gases from Methane by the use of Distillation and a Controlled Freezing Zone. Exxon Production Research Company, Houston, Texas, 1983.

[25] Bengt Olav Neeraas Knut Arild Maråk. Comparison Of Expander Processes For Natural Gas Liquefaction. Statoil Research Center, Trondheim, Norway.

[26] John V. O'Brien. Distillative Separation of Methane and Carbon Dioxide. Koch Process Systems, Inc., 1981.

[27] Hugh D. Epsom Peter Schinkelshoek. Supersonic Gas Conditioning - Commercialisation of Twister Technology. 87th Annual Convention in Grapevine, Texas, USA, 2008.

[28] E.R. Thomas R.C. Haut, R.D. Denton. Development and Application of the Controlled-Freeze-Zone Process, 1989.

[29] Walter Savitch. *Absolute C++*. Pearson Education International, 2010.

[30] Rocío Díez Sivia Pérez. Opportunities of monetising natural gas reserves using small to medium scale LNG technologies. IGU 24th World Gas Conference, REPSOL, 2009.

[31] G.C.Y. Watson B.F. Graham J. Boxall J.C. Diniz da Costa E.F. May T.E. Rufford, S. Smart. The removal of CO2 and N2 from natural gas: A review of conventional and emerging process technologies. Journal of Petroleum Science and Engineering, 2012.

A DSP method software code in C++

```
#ifndef __PreliminaryDSPinterface__Criterion__
#define __PreliminaryDSPinterface__Criterion__

#include <iostream>
#include <string>
using namespace std;

/** Class Criterion stores criteria for the
 Preliminary Selection DSP and attributes for
 Selection DSP.
 Code language: C++
 Part of: DSP program*/

class Criterion {
public:
    string critName;

    Criterion() {
        cout << "Criterium::constructor";
    }

    Criterion(string nameCrit) {
        critName = nameCrit;
    }
};
#endif /* defined(__PreliminaryDSPinterface__Criterion__) */
```

```
#ifndef __PreliminaryDSPinterface__Concept__
#define __PreliminaryDSPinterface__Concept__

#include <iostream>
#include <string>
using namespace std;

/** Class concept contains concepts for the
 Preliminary Selection DSP and alternatives for
 the Selection DSP.
 Code language: C++
 Part of: DSP program*/
class Concept {
public:
    string concName;

    Concept() {
        cout << "Concept::constructor";
    }

    Concept(string nameConc) {
        concName = nameConc;
    }
};

#endif /* defined(__PreliminaryDSPinterface__Concept__) */
```

```
#include "PreDSP.h"
#include <vector>
#include "Concept.h"
#include "Criterion.h"
```

```
#include <fstream>
#include <iomanip>

/**
 Class PreDSP performs all calculation for the
 Preliminary Selection DSP. This class also create
 appropriate instances of classes Concept and Criterion.
 Code language: C++
 Part of: DSP program
 */

class PreDSP {
public:
    int scen;

    vector<Criterion> criteria;
    vector<Concept> concepts;

    vector<vector<double>> relImp;
    vector<vector<vector<double>>> scoreArrayDatum;
    vector<vector<vector<double>>> normScore;
    vector<vector<vector<double>>> meritFunc;
    vector<vector<double>> preDSP;

    vector<vector<double>> aMin;
    vector<vector<double>> aMax;


    void addCriterium(string nameCrit) {
        Criterion cr(nameCrit);
        criteria.push_back(cr);
    }
```

```
void addConcept(string nameConc) {
    Concept co(nameConc);
    concepts.push_back(co);
}

/** printSystem function print a summary
 of the concepts and criteria added for the case. */

void printSystem(ofstream& outFile) {
    outFile << "------------------------------" << endl;
    outFile << "Preliminary Selection DSP" << endl;
    outFile << "------------------------------" << endl;
    for (int i = 0; i < concepts.size(); i++) {
        outFile << concepts.at(i).concName <<endl;
        for ( int j = 0; j < criteria.size(); j++) {
            outFile << "              "
            << criteria.at(j).critName << endl;
        }
    }
}

/**
 relativeImportance reads from text tile
 and creates a two-dimensional vector to
 store the scenarios of relative importances.
 */

void relativeImportance(ifstream& sourceFile) {
    double weight;
    sourceFile >> scen;
    for (int i = 0; i < scen; i++) {
```

```
        vector<double> row;
        relImp.push_back(row);
        for (int j = 0; j < criteria.size(); j++ ) {
            sourceFile >> weight;
            relImp[i].push_back(weight);
        }
    }
}

/** printRelImp writes the relative importances
 to the output text file */

void printRelImp(ofstream& outFile) {
    outFile << "------------------------------------------" << endl;
    outFile << "Relative Importance: " << endl;
    outFile << "------------------------------------------" << endl;
    outFile << setw(20) << "Scenario" << endl;
    for ( int i = 0; i < scen; i++) {
        if ( i == 0) {
            outFile << setw(15) << i+1;
        }
        else {
            outFile << setw(5) << i+1;
        }
    }
    outFile << endl;

    for (int j = 0; j < criteria.size(); j++) {
        outFile << setw(8) <<
        criteria.at(j).critName << ": ";
        for (int i = 0; i < scen; i++) {
            outFile << setw(5) << relImp[i][j];
```

```
        }
        outFile << endl;
    }

    double sum;
    outFile << "------------------------------" << endl;
    outFile << setw(8) << "Sum: ";
    for ( int i = 0; i < scen; i++) {
        sum = 0;
        for (int j = 0; j < criteria.size(); j++) {
            sum += relImp[i][j];
        }
        outFile << setw(6) << sum;
    }
    outFile << endl;
    outFile << endl;
}

/** setScore reads from text file and stores
 the scores in a three-dimensional vector
 scoreArrayDatum. */

void setScore(ifstream& sourcFile) {

    double score;
    for (int i = 0; i < concepts.size(); i ++) {
        vector<vector<double>> scoreArray;
        scoreArrayDatum.push_back(scoreArray);

        for (int j = 0; j < criteria.size(); j++) {
            vector<double> row;
            scoreArrayDatum[i].push_back(row);
```

```
            for (int k = 0; k < concepts.size(); k++) {
                if (concepts.at(i).concName ==
                    concepts.at(k).concName) {
                    scoreArrayDatum[i][j].push_back(0);
                }
                else {
                    sourcFile >> score;
                    scoreArrayDatum[i][j].push_back(score);
                }
            }
        }
    }
}

/** printScore writes the three-dimensional
 vector scoreArrayDatum to the output text file. */

void printScore(ofstream& outFile) {

    outFile << "----------------------------------------" << endl;
    outFile << setw(15) << "Criteria: "
    << setw(15) << "Concepts: " << endl;

    for( int i = 0; i < concepts.size(); i++) {
        outFile << "Scores for " << concepts.at(i).concName
        << " as datum: " << endl;
        for (int conc = 0; conc < concepts.size(); conc++) {
            if (conc == 0) {
                outFile << setw(17) << concepts.at(conc).concName;
            }
            else {
                outFile << setw(7) << concepts.at(conc).concName;
```

```
            }
        }
        outFile << endl;
        outFile << endl;
        for (int j = 0; j < criteria.size(); j++) {
            outFile << setw(10) << criteria.at(j).critName << ": ";
            for (int k = 0; k < concepts.size(); k++) {
                outFile << setw(6) << scoreArrayDatum[i][j][k];
            }
            outFile << endl;
        }
        outFile << endl;
    }
}

/** calcAmin finds the lowest score value for each criterion
 that is used to normalize scores. Stores the value in a
 two-dimensional vector: a min values for all datums. */

void calcAmin() {

    double min;
    for (int i = 0; i < concepts.size(); i++) {
        vector<double> row;
        aMin.push_back(row);
        for (int j = 0; j < criteria.size(); j++) {
            min = 1;
            for (int k = 0; k < concepts.size(); k++) {
                if (scoreArrayDatum[i][j][k] < min) {
                    min = scoreArrayDatum[i][j][k];
                }
            }
```

```
            aMin[i].push_back(min);
        }
    }
}
/** calcAmax finds the highest score value for each criterion
 that is used to normalize scores. Stores the value in a
 two-dimensional vector: a max values for all datums. */

void calcAmax() {

    double max;
    for (int i = 0; i < concepts.size(); i++) {
        vector<double> row;
        aMax.push_back(row);
        for (int j = 0; j < criteria.size(); j++) {
            max = -1;
            for (int k = 0; k < concepts.size(); k++) {
                if (scoreArrayDatum[i][j][k] > max) {
                    max = scoreArrayDatum[i][j][k];
                }
            }
            aMax[i].push_back(max);
        }
    }
}

/** calcNormscore calculates the normalized scores for
 all datums using the scoreArrayDatum, aMin and aMax.
 Normalized scores are stored in three-dimensional vector normScore.*/

void calcNormScore() {
```

```
    for ( int i = 0; i < concepts.size(); i++) {
        vector<vector<double>> norm2dScore;
        normScore.push_back(norm2dScore);
        for (int j = 0; j < criteria.size(); j++) {
            vector<double> row;
            normScore[i].push_back(row);
            for (int k = 0; k < concepts.size(); k++) {
                normScore[i][j].push_back(
                            (scoreArrayDatum[i][j][k] - aMin[i][j])
                            / (aMax[i][j] - aMin[i][j]));
            }
        }
    }
}

/** printNormScore writes the normalized scores for
 all datums to output text file.*/

void printNormScore(ofstream& outFile) {
    outFile << endl;
    outFile << endl;
    outFile << "--------------------" << endl;
    outFile << "Normalized scores: " << endl;
    outFile << "--------------------" << endl;
    outFile << endl;
    for (int i = 0; i < concepts.size(); i++) {
        outFile << "Normalized score with "
        << concepts.at(i).concName << " as datum. " << endl << endl;
        for(int conc = 0; conc < concepts.size(); conc++) {
            if (conc == 0) {
                outFile << setw(20) << concepts.at(conc).concName;
            }
```

```
            else {
                outFile << setw(8) << concepts.at(conc).concName;
            }
        }
        outFile << endl << endl;
        for (int j = 0; j < criteria.size(); j++) {
            outFile << setw(10) << criteria.at(j).critName << ": ";
            for (int k = 0; k < concepts.size(); k++) {
                outFile << setw(8) << setprecision(3) <<
                normScore[i][j][k];
            }
            outFile << endl;
            outFile << endl;
        }
    }
}

/** calcMeritFunc calculates the merit functions of all
 concepts for every set of datums. Merit functions are
 stored in a three-dimensional*/

void calcMeritFunc() {

    double sum;
    for (int i = 0; i < concepts.size(); i++) {
        vector<vector<double>> mFunction;
        meritFunc.push_back(mFunction);
        for (int j = 0; j < scen; j++) {
            vector<double> row;
            meritFunc[i].push_back(row);
            sum = 0;
            for (int k = 0; k < concepts.size(); k++) {
```

```
                sum = 0;
                for (int l = 0; l < criteria.size(); l++) {
                    sum += (normScore[i][l][k] * relImp[j][l]);

                }
                meritFunc[i][j].push_back(sum);
            }
        }
    }
}

/** printMeritFunc writes the merit functions of all
 concepts for every set of datums to the output text file.*/

void printMeritFunc(ofstream& outFile) {

    outFile << endl;
    outFile << endl;
    outFile << "-----------------------------------------" << endl;
    outFile << "Merit functions: " << endl;
    outFile << "-----------------------------------------" << endl;
    outFile << endl;
    outFile << endl;

    for (int i = 0; i < concepts.size(); i++) {
        outFile << "Merit function for all technologies using "
        << concepts.at(i).concName << " as datum." << endl << endl;
        for (int conc = 0; conc < concepts.size(); conc++) {
            if (conc == 0) {
                outFile << "Scenario" << setw(5) <<
                concepts.at(conc).concName;
            }
```

```
            else {
                outFile << setw(8) << concepts.at(conc).concName;
            }
        }
        outFile << endl << endl;
        for (int j = 0; j < scen; j++) {

            outFile << setw(5) << j + 1;

            for (int k = 0; k < concepts.size(); k++) {
                outFile << setw(8) << setprecision(3) <<
                meritFunc[i][j][k];
            }
            outFile << endl;
            outFile << endl;
        }
    }
}

/** totalPreScore calculates the overall merit functions
 for every concepts. Overall merit functions are stored
 as a two-dimensional array preDSP.*/

void totalPreScore(ofstream& outFile) {
    double sum;
    for (int i = 0; i < scen; i++) {
        for (int j = 0; j < concepts.size(); j++) {
            vector<double> row;
            preDSP.push_back(row);
            sum = 0;
            for (int k = 0; k < concepts.size(); k++) {
                sum += meritFunc[k][i][j];
```

```
        }
        sum = sum / concepts.size();
        preDSP[i].push_back(sum);
    }

}
outFile << endl;
outFile << endl;
outFile << "----------------------------------------------" << endl;
outFile << "Overall merit functions: " << endl;
outFile << "----------------------------------------------" << endl;
outFile << endl;
for (int i = 0; i < concepts.size(); i++) {
    if (i == 0) {
        outFile << "Scenario" << setw(10) <<
        concepts.at(i).concName;
    }
    else {
        outFile << setw(8) << concepts.at(i).concName;
    }
}
outFile << endl;
outFile << endl;
for (int i = 0; i < scen; i++) {
    outFile << setw(10) << i + 1;
    for (int j = 0; j < concepts.size(); j++) {
        outFile << setw(12) << setprecision(3) <<
        preDSP[i][j];
    }
    outFile << endl << endl;
}
```

```
}

/** readFromFile reads text file and calls on the
 functions in PreDSP to perform the Preliminary Selection
 DSP. The function also writes the result to a text file.*/

void readFromFile() {
    ifstream inStream;
    inStream.open("/Users/carolinegulliksen/Library/Application Support/il
    if (inStream.fail()) {
        cout << "Opening of input file failed.\n";
        exit(1);
    }

    ofstream outputStream;
    ofstream outStream;
    outStream.open("/Users/carolinegulliksen/Dropbox/PreliminaryDSPinterf
    if (outStream.fail()) {
        cout << "Opening of output file failed.\n";
        exit(1);
    }

    outputStream.open("/Users/carolinegulliksen/Dropbox/PreliminaryDSPint
    if (outputStream.fail()) {
        cout << "Opening of summary file failed.\n";
        exit(1);
    }

    int numConcept, numCrit;
    string concept, crit;

    inStream >> numConcept;
```

```
    for (int i = 0; i < numConcept; i++) {
        inStream >> concept;
        addConcept(concept);
    }

    inStream >> numCrit;

    for (int i = 0; i < numCrit; i++) {
        inStream >> crit;
        addCriterium(crit);
    }

    printSystem(outStream);
    relativeImportance(inStream);
    printRelImp(outStream);
    setScore(inStream);
    printScore(outStream);
    calcAmin();
    calcAmax();
    calcNormScore();
    printNormScore(outStream);
    calcMeritFunc();
    printMeritFunc(outStream);
    totalPreScore(outStream);

    totalPreScore(outputStream);

    inStream.close();
    outStream.close();
    outputStream.close();
}
```

```
};



#include "SelDSP.h"
#include "PreDSP.h"
#include <vector>
#include "Concept.h"
#include "Criterion.h"
#include <fstream>
#include <iomanip>

/** Class SelDSP performs all calculations in
  Selection DSP and creates appropriate instances of
  classes Concept and Criterion for the alternatives
  and attributes, respectively.
  Code language: C++
  Part of: DSP program*/


class SelDSP {
public:
    vector<Criterion> attributes;
    vector<Concept> alternatives;

    vector<double> weights;

    vector<vector<double>> attRating;
    vector<vector<double>> normAltRating;
    vector<vector<double>> meritRating;

    vector<double> finalRank;
    vector<double> sortedFinal;
```

```
/** addAttribute creates instances of class
 Criterion for the attributes.*/

void addAttribute(string nameAtt) {
    Criterion at(nameAtt);
    attributes.push_back(at);
}

/** addAlternative creates instances of class
 Concept for the alternatives.*/

void addAlternative(string nameAlt) {
    Concept al(nameAlt);
    alternatives.push_back(al);
}

/** printSystem writes a summary of the alternatives
 and attribute added for the case.*/

void printSystem(ofstream& outFile) {
    outFile << endl;
    outFile << endl;
    outFile << "----------------" << endl;
    outFile << "Selection DSP " << endl;
    outFile << "----------------" << endl;
    for (int i = 0; i < alternatives.size(); i++) {
        outFile << alternatives.at(i).concName << endl;
        for (int j = 0; j < attributes.size(); j++) {
            outFile << "            " <<
            attributes.at(j).critName << endl;
        }
```

```
    }
}

/** calcNormRank reads the attribute rank from a text file
 and calculates the normalized ranks for the attributes.
 Results are stored in a vector weights.*/

void calcNormRank(ifstream& sourceFile, ofstream& outFile) {
    outFile << endl;
    outFile << endl;

    outFile << "-----------------------------------------------------------------"
    << endl;
    outFile << "Calculating normalized rank of the relative importance:"
    << endl;
    outFile << "-----------------------------------------------------------------"
    << endl;
    outFile << endl;

    double rank;
    double sum = 0;
    for (int i = 0; i < attributes.size(); i++) {
        sum += (i+1);
    }
    for (int i = 0; i < attributes.size(); i++) {
        sourceFile >> rank;
        weights.push_back(rank/sum);
    }

    for (int i = 0; i < attributes.size(); i++) {
        outFile << setw(9) << attributes.at(i).critName << ": "
        << setw(4) << setprecision(3) << weights.at(i) << endl;
```

```
    }
}

/** alternativeRating calculates the normalized alternative rating
 from text file input. The ratings are stored in a two-dimensional
 vector normAltRating.*/

void alternativeRating(ifstream& sourceFile) {

    double input, lowBound, highBound;
    string pref;

    for (int i = 0; i < attributes.size(); i++) {
        vector<double> row;
        normAltRating.push_back(row);
        sourceFile >> highBound;
        sourceFile >> lowBound;
        sourceFile >> pref;
        for (int j = 0; j < alternatives.size(); j++) {
            sourceFile >> input;
            if (pref == "L") {
                pref = "l";
            }
            if (pref == "H") {
                pref = "h";
            }
            if (pref == "l") {
                normAltRating[i].push_back(1 - (input - lowBound)
                                           / (highBound - lowBound));
            }
            else if (pref == "h") {
                normAltRating[i].push_back((input - lowBound)
```

```
                                        / (highBound - lowBound));
            }
        }
    }
}

/** printNormAltRating writes the normalized alternative ratings to
 output text file.*/

void printNormAltRating(ofstream& outFile) {
    outFile << endl;
    outFile << endl;

    outFile << "————————————————————————————" << endl;
    outFile << "Normalized attribute rating: " << endl;
    outFile << "————————————————————————————" << endl;

    outFile << setw(15) << "Attributes: " << setw(30) <<
    "Alternatives: " << endl;
    outFile << endl;
    for (int i = 0; i < alternatives.size(); i++) {
        if (i == 0) {
            outFile << setw(26) << alternatives.at(i).concName;
        }
        else {
            outFile << setw(15) << alternatives.at(i).concName;
        }
    }
    outFile << endl;
    for (int i = 0; i < attributes.size(); i++) {
        outFile << setw(10) << attributes.at(i).critName << ": ";
        for (int j = 0; j < alternatives.size(); j++) {
```

```
            outFile << setw(15) << setprecision(3) <<
            normAltRating[i][j];
        }
        outFile << endl;
        outFile << endl;
    }
}

/** calcMeritFunc calculates the merit ratings from the normalized
 alternative ratings and the  normalized ranks of the attributes.
 Results are stored in a two-dimensional vector meritRating. finalRank
 is also calculated and final result is written to output text file.*/

void calcMeritFunc(ofstream& outFile) {
    for (int i = 0; i < attributes.size(); i++) {
        vector<double> row;
        meritRating.push_back(row);
        for (int j = 0; j < alternatives.size(); j++) {
            meritRating[i].push_back(normAltRating[i][j] * weights[i]);
        }
    }

    double sum;
    for (int i = 0; i < alternatives.size(); i++) {
        sum = 0;
        for (int j = 0; j < attributes.size(); j++) {
            sum += meritRating[j][i];
        }
        finalRank.push_back(sum);
    }

    outFile << endl;
```

```
    outFile << endl;

    outFile << "-------------------" << endl;
    outFile << "Final ranking: " << endl;
    outFile << "-------------------" << endl;

    for (int i = 0; i < alternatives.size(); i++) {
        outFile << setw(6) << alternatives.at(i).concName
        << ": " << setprecision(3) << finalRank[i] << endl;
    }
}

/** readFromFile reads the Selection DSP input from text file and
 calls the functions of the SelDSP class to perform the Selection
 DSP. The results are written to a output text file for evaluation.*/

void readFromFile() {

    ifstream inStream;
    ofstream outStream;
    ofstream outStreamSum;

    inStream.open("/Users/carolinegulliksen/Library/Application Support/il
    if (inStream.fail()) {
        cout << "Opening Selection DSP input file failed.\n";
        exit(1);
    }

    outStream.open("/Users/carolinegulliksen/Dropbox/PreliminaryDSPinterf
    if (outStream.fail()) {
        cout << "Opening Selection DSP output file failed.\n";
        exit(1);
```

```
    }

    outStreamSum.open("/Users/carolinegulliksen/Dropbox/PreliminaryDSPint
    if (outStreamSum.fail()) {
        cout << "Opening Selection DSP output summary file failed.\n";
        exit(1);
    }

    int numAlt, numAtt;
    string alternative, attribute;

    inStream >> numAlt;
    for (int i = 0; i < numAlt; i++) {
        inStream >> alternative;
        addAlternative(alternative);
    }

    inStream >> numAtt;
    for (int i = 0; i < numAtt; i++) {
        inStream >> attribute;
        addAttribute(attribute);
    }

    printSystem(outStream);
    calcNormRank(inStream, outStream);
    alternativeRating(inStream);
    printNormAltRating(outStream);
    calcMeritFunc(outStream);
    calcMeritFunc(outStreamSum);

}
```

```
};
```

B CO_2 removal case: Detailed results from Preliminary Selection DSP

```
-------------------------
Preliminary Selection DSP
-------------------------
CRYO
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
SPREX
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
RH
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
TWIST
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
CFZ
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
AMINE
              COMPLEX
              PERFORM
              POWER
              SAFE
              MATUR
---------------------
Relative Importance:
---------------------
            Scenario
              1    2    3    4    5
 COMPLEX:   0.2  0.2  0.1  0.1  0.4
 PERFORM:   0.2  0.2  0.4  0.2  0.1
   POWER:   0.2  0.1  0.2  0.4  0.2
    SAFE:   0.2  0.4  0.2  0.2  0.2
   MATUR:   0.2  0.1  0.1  0.1  0.1
---------------------------------------
   Sum:      1    1    1    1    1
---------------------------------------
     Criteria:      Concepts:
Scores for CRYO as datum:
             CRYO  SPREX     RH  TWIST    CFZ  AMINE

   COMPLEX:      0    -1    -1     1     0    -1
   PERFORM:      0     1     1     0     1     1
     POWER:      0    -1     1     1     0    -1
      SAFE:      0     0     0     0     0     1
     MATUR:      0    -1     1    -1     1     1
```

Scores for SPREX as datum:

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	0	1	1	1	−1
PERFORM:	−1	0	0	−1	0	0
POWER:	1	0	1	1	1	−1
SAFE:	0	0	0	0	0	1
MATUR:	1	0	1	0	1	1

Scores for RH as datum:

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	−1	0	1	1	−1
PERFORM:	−1	0	0	−1	0	0
POWER:	1	−1	0	1	−1	−1
SAFE:	0	0	0	0	0	1
MATUR:	−1	−1	0	−1	−1	1

Scores for TWIST as datum:

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	−1	−1	−1	0	−1	−1
PERFORM:	0	1	1	0	1	1
POWER:	−1	−1	−1	0	−1	−1
SAFE:	0	0	0	0	0	1
MATUR:	1	0	1	0	1	1

Scores for CFZ as datum:

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	0	−1	−1	1	0	−1
PERFORM:	−1	0	0	−1	0	0
POWER:	0	−1	1	1	0	−1
SAFE:	0	0	0	0	0	1
MATUR:	−1	−1	1	−1	0	1

Scores for AMINE as datum:

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	−1	1	1	1	0
PERFORM:	−1	0	0	−1	0	0
POWER:	1	1	1	1	1	0
SAFE:	−1	−1	−1	−1	−1	0
MATUR:	−1	−1	0	−1	−1	0

Normalized scores:

Normalized score with CRYO as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	0.5	0	0	1	0.5	0
PERFORM:	0	1	1	0	1	1
POWER:	0.5	0	1	1	0.5	0
SAFE:	0	0	0	0	0	1
MATUR:	0.5	0	1	0	1	1

Normalized score with SPREX as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	0.5	1	1	1	0
PERFORM:	0	1	1	0	1	1
POWER:	1	0.5	1	1	1	0
SAFE:	0	0	0	0	0	1
MATUR:	1	0	1	0	1	1

Normalized score with RH as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	0	0.5	1	1	0
PERFORM:	0	1	1	0	1	1
POWER:	1	0	0.5	1	0	0
SAFE:	0	0	0	0	0	1
MATUR:	0	0	0.5	0	0	1

Normalized score with TWIST as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	0	0	0	1	0	0
PERFORM:	0	1	1	0	1	1
POWER:	0	0	0	1	0	0
SAFE:	0	0	0	0	0	1
MATUR:	1	0	1	0	1	1

Normalized score with CFZ as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	0.5	0	0	1	0.5	0
PERFORM:	0	1	1	0	1	1
POWER:	0.5	0	1	1	0.5	0
SAFE:	0	0	0	0	0	1
MATUR:	0	0	1	0	0.5	1

Normalized score with AMINE as datum.

	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COMPLEX:	1	0	1	1	1	0.5
PERFORM:	0	1	1	0	1	1
POWER:	1	1	1	1	1	0
SAFE:	0	0	0	0	0	1
MATUR:	0	0	1	0	0	1

Merit functions:

Merit function for all technologies using CRYO as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.3	0.2	0.6	0.4	0.6	0.6
2	0.2	0.2	0.4	0.3	0.45	0.7
3	0.2	0.4	0.7	0.3	0.65	0.7
4	0.3	0.2	0.7	0.5	0.55	0.5

5	0.35	0.1	0.4	0.6	0.5	0.4

Merit function for all technologies using SPREX as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.6	0.4	0.8	0.4	0.8	0.6
2	0.4	0.35	0.6	0.3	0.6	0.7
3	0.4	0.55	0.8	0.3	0.8	0.7
4	0.6	0.45	0.8	0.5	0.8	0.5
5	0.7	0.4	0.8	0.6	0.8	0.4

Merit function for all technologies using RH as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.4	0.2	0.5	0.4	0.4	0.6
2	0.3	0.2	0.4	0.3	0.4	0.7
3	0.3	0.4	0.6	0.3	0.5	0.7
4	0.5	0.2	0.5	0.5	0.3	0.5
5	0.6	0.1	0.45	0.6	0.5	0.4

Merit function for all technologies using TWIST as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.2	0.2	0.4	0.4	0.4	0.6
2	0.1	0.2	0.3	0.3	0.3	0.7
3	0.1	0.4	0.5	0.3	0.5	0.7
4	0.1	0.2	0.3	0.5	0.3	0.5
5	0.1	0.1	0.2	0.6	0.2	0.4

Merit function for all technologies using CFZ as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.2	0.2	0.6	0.4	0.5	0.6
2	0.15	0.2	0.4	0.3	0.4	0.7
3	0.15	0.4	0.7	0.3	0.6	0.7
4	0.25	0.2	0.7	0.5	0.5	0.5
5	0.3	0.1	0.4	0.6	0.45	0.4

Merit function for all technologies using AMINE as datum.

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.4	0.4	0.8	0.4	0.6	0.7
2	0.3	0.3	0.6	0.3	0.5	0.8
3	0.3	0.6	0.8	0.3	0.7	0.75
4	0.5	0.6	0.8	0.5	0.7	0.55
5	0.6	0.3	0.8	0.6	0.7	0.6

```
-------------------------
Overall merit functions:
-------------------------
```

Scenario	CRYO	SPREX	RH	TWIST	CFZ	AMINE
1	0.35	0.267	0.617	0.4	0.55	0.617
2	0.242	0.242	0.45	0.3	0.442	0.717
3	0.242	0.458	0.683	0.3	0.625	0.708
4	0.375	0.308	0.633	0.5	0.525	0.508
5	0.442	0.183	0.508	0.6	0.525	0.433

Examples of justification of scoring with the Amine process as datum:

Criteria	Concepts					
	CRYO	SPREX	RH	TWIST	CFZ	AMINE
COM-PLEX	The Cryocell needs dehydration of feed gas. The Cryocell consists of one distillation column, but if the CO2 content is more than 20%, more equipment is needed.	The process includes a distillation column for bulk removal of acid gas, and dehydration of feed gas before entering the Sprex process. From there, the gas enters an amine unit.	The Ryan/Holmes process consists of a distillation tower and a regeneration separator. Since the process operates under high pressures the need for pumping is less than for an amine process.	The Twister consists of a cyclonic separator, and with no rotating equipment this becomes a simple process	The CFZ consists of one distillation column and does not need any additives so the complexity is reduced because there is no excessive piping and little equipment	Chosen datum, ”0” is assigned
PER-FORM	Not able to obtain pipeline quality	Not able to obtain pipeline quality	RH is able to obtain pipeline quality	Twister is not able to obtain pipeline quality	CFZ is able to obtain pipeline quality.	Chosen datum
POWER	All concepts requires less energy than the amine process					Datum
SAFE	These use flammable refrigerants to cool the gas					Datum
MATUR	Less experience than amine process		A RH plant has been in operation for several years and is proven to work	Less experience than amine process		Chosen datum

C CO_2 removal case: Detailed results from Selection DSP

```
--------------
Selection DSP
--------------
AMINE
            ENVIRON
            COMPACT
            EQUIP
            POWER
            OPER
            SENS
RH
            ENVIRON
            COMPACT
            EQUIP
            POWER
            OPER
            SENS
CFZ
            ENVIRON
            COMPACT
            EQUIP
            POWER
            OPER
            SENS

-------------------------------------------------------
Calculating normalized rank of the relative importance:
-------------------------------------------------------

  ENVIRON: 0.0476
  COMPACT: 0.286
    EQUIP: 0.143
    POWER: 0.238
     OPER: 0.0952
     SENS: 0.19

-----------------------------
Normalized attribute rating:
-----------------------------
   Attributes:                Alternatives:

                  AMINE            RH           CFZ
   ENVIRON:         0.25          0.375           0.5

   COMPACT:         0.25          0.375         0.625

     EQUIP:        0.538          0.615         0.692

     POWER:        0.111          0.667         0.444

      OPER:        0.143          0.143         0.429

      SENS:          0.2            0.2           0.6

---------------
Final ranking:
---------------
 AMINE: 0.238
    RH: 0.423
   CFZ: 0.562
```

Examples on scales:

Rating	Attribute: ENVIRON
0	No emmisions
1	
2	
3	Emissions can lead to global warming
4	
5	
6	Emissions are toxic to humans
7	
8	

Rating	Attribute: COMPACT
1	Very compact
2	
3	
4	Acceptable
5	
6	
7	Not acceptable, takes too much space
8	
9	

Rating	Attribute: EQUIP
1	units
2	units
3	units
4	units
5	units
6	units
7	units
8	units
9	units
10	units
11	units
12	units
13	units
14	units

Rating	Attribute: POWER
0	Very good, low power consumption
1	
2	
3	Normal power consumption
4	
5	
6	Over normal power consumption
7	
8	
9	Unacceptable, high power consumption
10	

Rating	Attribute: OPER
0	No monitoring
1	
2	Low monitoring
3	
4	Acceptable monitoring
5	
6	Excessive monitoring
7	

Rating	Attribute: SENS
0	Low sensibility to motion
1	
2	Some sensiblitiy, acceptable
3	
4	Very sensible to motion
5	

D Liquefaction process for LNG-FPSO case: Detailed results from Preliminary Selection DSP

```
-------------------------
Preliminary Selection DSP
-------------------------
PRICO
              POWER
              COMPLEX
              SENS
              SAFE
              REL
LBSF
              POWER
              COMPLEX
              SENS
              SAFE
              REL
AP-M
              POWER
              COMPLEX
              SENS
              SAFE
              REL
LiMUM
              POWER
              COMPLEX
              SENS
              SAFE
              REL
K-SCMR
              POWER
              COMPLEX
              SENS
              SAFE
              REL
K-PCMR
              POWER
              COMPLEX
              SENS
              SAFE
              REL
SNExp
              POWER
              COMPLEX
              SENS
              SAFE
              REL
DNExp
              POWER
              COMPLEX
              SENS
              SAFE
              REL
K-EXP
              POWER
              COMPLEX
              SENS
              SAFE
              REL
---------------------
Relative Importance:
---------------------
            Scenario
              1    2    3    4    5
```

POWER:	0.2	0.3	0.1	0.4	0.2
COMPLEX:	0.2	0.1	0.2	0.2	0.3
SENS:	0.2	0.2	0.2	0.1	0.2
SAFE:	0.2	0.3	0.3	0.2	0.1
REL:	0.2	0.1	0.2	0.1	0.2
Sum:	1	1	1	1	1

Criteria: Concepts:

Scores for PRICO as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	0	1	1	1	1	1	–1	–1	–1
COMPLEX:	0	–1	–1	–1	–1	–1	–1	–1	–1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	0	–1	0	–1	–1	–1	–1	–1	–1

Scores for LBSF as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	–1	0	0	1	1	1	–1	–1	–1
COMPLEX:	1	0	–1	–1	–1	0	0	–1	0
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Scores for AP–M as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	–1	0	0	1	1	1	–1	–1	–1
COMPLEX:	1	1	0	–1	0	1	1	–1	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	0	–1	0	–1	–1	–1	–1	–1	–1

Scores for LiMUM as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	–1	–1	–1	0	1	1	–1	–1	–1
COMPLEX:	1	1	1	0	1	1	1	1	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Scores for K–SCMR as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	–1	–1	–1	–1	0	0	–1	–1	–1
COMPLEX:	1	1	0	–1	0	1	1	–1	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Scores for K–PCMR as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	–1	–1	–1	–1	0	0	–1	–1	–1
COMPLEX:	1	0	–1	–1	–1	0	–1	–1	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Scores for SNExp as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	1	1	1	1	1	1	0	1	1
COMPLEX:	1	0	–1	–1	–1	–1	0	–1	1
SENS:	–1	–1	–1	–1	–1	–1	0	0	0
SAFE:	–1	–1	–1	–1	–1	–1	0	0	0
REL:	1	0	1	0	0	0	0	0	0

Scores for DNExp as datum:

	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
POWER:	1	1	1	1	1	1	–1	0	1

COMPLEX:	1	1	1	−1	−1	1	1	0	1
SENS:	−1	−1	−1	−1	−1	−1	0	0	0
SAFE:	−1	−1	−1	−1	−1	−1	0	0	0
REL:	1	0	1	0	0	0	0	0	0

Scores for K−EXP as datum:

	PRICO	LBSF	AP−M	LiMUM	K−SCMR	K−PCMR	SNExp	DNExp	K−EXP
POWER:	1	1	1	1	1	1	−1	−1	0
COMPLEX:	1	1	−1	−1	−1	−1	−1	−1	0
SENS:	−1	−1	−1	−1	−1	−1	0	0	0
SAFE:	−1	−1	1	−1	−1	−1	0	0	0
REL:	1	0	0	0	0	0	0	0	0

Normalized scores:

Normalized score with PRICO as datum.

	PRICO	LBSF	AP−M	LiMUM	K−SCMR	K−PCMR	SNExp	DNExp	K−EXP
POWER:	0.5	1	1	1	1	1	0	0	0
COMPLEX:	1	0	0	0	0	0	0	0	0
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with LBSF as datum.

	PRICO	LBSF	AP−M	LiMUM	K−SCMR	K−PCMR	SNExp	DNExp	K−EXP
POWER:	0	0.5	0.5	1	1	1	0	0	0
COMPLEX:	1	0.5	0	0	0	0.5	0.5	0	0.5
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with AP−M as datum.

	PRICO	LBSF	AP−M	LiMUM	K−SCMR	K−PCMR	SNExp	DNExp	K−EXP
POWER:	0	0.5	0.5	1	1	1	0	0	0
COMPLEX:	1	1	0.5	0	0.5	1	1	0	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with LiMUM as datum.

	PRICO	LBSF	AP−M	LiMUM	K−SCMR	K−PCMR	SNExp	DNExp	K−EXP
POWER:	0	0	0	0.5	1	1	0	0	0
COMPLEX:	1	1	1	0	1	1	1	1	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with K-SCMR as datum.

	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
POWER:	0	0	0	0	1	1	0	0	0
COMPLEX:	1	1	0.5	0	0.5	1	1	0	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with K-PCMR as datum.

	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
POWER:	0	0	0	0	1	1	0	0	0
COMPLEX:	1	0.5	0	0	0	0.5	0	0	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with SNExp as datum.

	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
POWER:	1	1	1	1	1	1	0	1	1
COMPLEX:	1	0.5	0	0	0	0	0.5	0	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with DNExp as datum.

	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
POWER:	1	1	1	1	1	1	0	0.5	1
COMPLEX:	1	1	1	0	0	1	1	0.5	1
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	0	0	0	0	1	1	1
REL:	1	0	1	0	0	0	0	0	0

Normalized score with K-EXP as datum.

	PRICO	LBSF	AP-M	LiMUM	K-SCMR	K-PCMR	SNExp	DNExp	K-EXP
POWER:	1	1	1	1	1	1	0	0	0.5
COMPLEX:	1	1	0	0	0	0	0	0	0.5
SENS:	0	0	0	0	0	0	1	1	1
SAFE:	0	0	1	0	0	0	0.5	0.5	0.5
REL:	1	0	0	0	0	0	0	0	0

```
-----------------
Merit functions:
-----------------
```

Merit function for all technologies using PRICO as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.5	0.2	0.4	0.2	0.2	0.2	0.4	0.4	0.4
2	0.35	0.3	0.4	0.3	0.3	0.3	0.5	0.5	0.5
3	0.45	0.1	0.3	0.1	0.1	0.1	0.5	0.5	0.5
4	0.5	0.4	0.5	0.4	0.4	0.4	0.3	0.3	0.3
5	0.6	0.2	0.4	0.2	0.2	0.2	0.3	0.3	0.3

Merit function for all technologies using LBSF as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.4	0.2	0.3	0.2	0.2	0.3	0.5	0.4	0.5
2	0.2	0.2	0.25	0.3	0.3	0.35	0.55	0.5	0.55
3	0.4	0.15	0.25	0.1	0.1	0.2	0.6	0.5	0.6
4	0.3	0.3	0.3	0.4	0.4	0.5	0.4	0.3	0.4
5	0.5	0.25	0.3	0.2	0.2	0.35	0.45	0.3	0.45

Merit function for all technologies using AP–M as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.4	0.3	0.4	0.2	0.3	0.4	0.6	0.4	0.6
2	0.2	0.25	0.3	0.3	0.35	0.4	0.6	0.5	0.6
3	0.4	0.25	0.35	0.1	0.2	0.3	0.7	0.5	0.7
4	0.3	0.4	0.4	0.4	0.5	0.6	0.5	0.3	0.5
5	0.5	0.4	0.45	0.2	0.35	0.5	0.6	0.3	0.6

Merit function for all technologies using LiMUM as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.4	0.2	0.4	0.1	0.4	0.4	0.6	0.6	0.6
2	0.2	0.1	0.2	0.15	0.4	0.4	0.6	0.6	0.6
3	0.4	0.2	0.4	0.05	0.3	0.3	0.7	0.7	0.7
4	0.3	0.2	0.3	0.2	0.6	0.6	0.5	0.5	0.5
5	0.5	0.3	0.5	0.1	0.5	0.5	0.6	0.6	0.6

Merit function for all technologies using K–SCMR as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.4	0.2	0.3	0	0.3	0.4	0.6	0.4	0.6
2	0.2	0.1	0.15	0	0.35	0.4	0.6	0.5	0.6
3	0.4	0.2	0.3	0	0.2	0.3	0.7	0.5	0.7
4	0.3	0.2	0.2	0	0.5	0.6	0.5	0.3	0.5
5	0.5	0.3	0.35	0	0.35	0.5	0.6	0.3	0.6

Merit function for all technologies using K–PCMR as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.4	0.1	0.2	0	0.2	0.3	0.4	0.4	0.6

2	0.2	0.05	0.1	0	0.3	0.35	0.5	0.5	0.6
3	0.4	0.1	0.2	0	0.1	0.2	0.5	0.5	0.7
4	0.3	0.1	0.1	0	0.4	0.5	0.3	0.3	0.5
5	0.5	0.15	0.2	0	0.2	0.35	0.3	0.3	0.6

Merit function for all technologies using SNExp as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.6	0.3	0.4	0.2	0.2	0.2	0.5	0.6	0.8
2	0.5	0.35	0.4	0.3	0.3	0.3	0.55	0.8	0.9
3	0.5	0.2	0.3	0.1	0.1	0.1	0.6	0.6	0.8
4	0.7	0.5	0.5	0.4	0.4	0.4	0.4	0.7	0.9
5	0.7	0.35	0.4	0.2	0.2	0.2	0.45	0.5	0.8

Merit function for all technologies using DNExp as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.6	0.4	0.6	0.2	0.2	0.4	0.6	0.6	0.8
2	0.5	0.4	0.5	0.3	0.3	0.4	0.6	0.7	0.9
3	0.5	0.3	0.5	0.1	0.1	0.3	0.7	0.65	0.8
4	0.7	0.6	0.7	0.4	0.4	0.6	0.5	0.6	0.9
5	0.7	0.5	0.7	0.2	0.2	0.5	0.6	0.55	0.8

Merit function for all technologies using K–EXP as datum.

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.6	0.4	0.4	0.2	0.2	0.2	0.3	0.3	0.5
2	0.5	0.4	0.6	0.3	0.3	0.3	0.35	0.35	0.55
3	0.5	0.3	0.4	0.1	0.1	0.1	0.35	0.35	0.5
4	0.7	0.6	0.6	0.4	0.4	0.4	0.2	0.2	0.5
5	0.7	0.5	0.3	0.2	0.2	0.2	0.25	0.25	0.5

Overall merit functions:

Scenario	PRICO	LBSF	AP–M	LiMUM	K–SCMR	K–PCMR	SNExp	DNExp	K–EXP
1	0.478	0.256	0.378	0.144	0.244	0.311	0.5	0.456	0.6
2	0.317	0.239	0.322	0.217	0.322	0.356	0.539	0.55	0.644
3	0.439	0.2	0.333	0.0722	0.144	0.211	0.594	0.533	0.667
4	0.456	0.367	0.4	0.289	0.444	0.511	0.4	0.389	0.556
5	0.578	0.328	0.4	0.144	0.267	0.367	0.461	0.378	0.583

E Liquefaction process for LNG-FPSO case: Detailed results from Selection DSP

```
--------------
Selection DSP
--------------
SNExp
            POWER
            EQUIP
            PRODS
            SPACE
            COMPLEX
            EXPER
            SAFE
DNExp
            POWER
            EQUIP
            PRODS
            SPACE
            COMPLEX
            EXPER
            SAFE
K-EXP
            POWER
            EQUIP
            PRODS
            SPACE
            COMPLEX
            EXPER
            SAFE
PRICO
            POWER
            EQUIP
            PRODS
            SPACE
            COMPLEX
            EXPER
            SAFE
-------------------------------------------------------
Calculating normalized rank of the relative importance:
-------------------------------------------------------

    POWER: 0.214
    EQUIP: 0.143
    PRODS: 0.0357
    SPACE: 0.179
  COMPLEX: 0.107
    EXPER: 0.0714
     SAFE: 0.25
-----------------------------
Normalized attribute rating:
-----------------------------
   Attributes:                Alternatives:

                      SNExp          DNExp          K-EXP          PRICO
     POWER:            0.25            0.5          0.625           0.75

     EQUIP:             0.5          0.333          0.333          0.833

     PRODS:           0.111          0.778          0.333          0.778

     SPACE:           0.889          0.778          0.778          0.556

   COMPLEX:           0.667          0.556          0.556          0.444
```

```
    EXPER:            0.4            0.2            0.2            0.8

     SAFE:            0.6            0.6            0.6              0

---------------
Final ranking:
---------------
 SNExp: 0.538
 DNExp: 0.545
 K-EXP: 0.556
 PRICO: 0.512
```