THE UNIVERSITY *of* LIVERPOOL

ONTOLOGY-BASED ARTIFICIAL INTELLIGENCE APPROACHES FOR THE ASSET MANAGEMENT OF POWER SUBSTATIONS

Thesis submitted in accordance with the
requirements of the University of Liverpool
for the degree of Doctor of Philosophy

in

Electrical Engineering and Electronics

by

Long YAN, B.Sc. (Eng.)

March 2016

ONTOLOGY-BASED ARTIFICIAL INTELLIGENCE APPROACHES FOR THE ASSET MANAGEMENT OF POWER SUBSTATIONS

by

Long YAN

To my families

Acknowledgements

First and foremost, I would like to give my heartfelt thanks to my supervisors, Dr. Y. Goulermas (academic year from 2013 to 2016) and Prof. W.H. Tang (academic year from 2011 to 2013), not only for their invaluable support, stimulating discussions and intellectual guidance on the research areas of document clustering; document search engine and power transformer fault diagnosis, respectively, but also for providing such a great opportunity to enhance my capability both on an academic and personal level. They have made a great number of contributions to this research, of which i am truly appreciative.

I am deeply grateful to Prof. Q.H. Wu, for his kind guidance with his knowledge on power systems. The research skill, writing skill and presenting skill he has taught me will benefit me throughout my life.

I also want to offer my regards to all of my friends in the Electrical Engineering and Electronics Department, especially Miss. J. Zhu, Dr. C.H. Wei and Dr. C. Li, for their great support and valuable comments on both of my study and daily life. Moreover, my thanks are offered to the Department of Electrical Engineering and Electronics at the University of Liverpool, for providing the research facilities that made it possible for me to carry out this research.

Last but not least, I am greatly indebted to my parents, for their understanding, encouragement, patience and love throughout the period of my postgraduate life.

Abstract

This thesis focuses on presenting three Artificial Intelligence approaches to the asset management (AM) of power substations, including an improved document ranking method in ontology-based document search engine (ODSE) using evidential reasoning (ER), consensus clustering (CC) for the ontology-based modified document repository, and the development of a novel ontology-based power transformer fault diagnosis (PTFD) system using Bayesian networks (BNs), respectively. The first two approaches, which are carried out on a power substation-related document repository (PSD), belong to the intangible AM of power substations, and they are also dedicated to the research area of information retrieval (IR). The third approach concerns one of the physical assets in power substations, i.e., the power transformer, and it is an extension of the conventional ontology-based PTFD system with the ability of uncertainty reasoning.

The first part of this thesis is devoted to introducing a novel approach to document ranking in an ODSE using ER. A domain ontology model, i.e., substation ontology (SONT), which is used for query expansion (QE), is proposed. A multiple attribute decision making (MADM) tree model is applied to organise expanded query terms. Then, an ER algorithm, which is based on the Dempster-Shafer (DS) theory, is implemented for evidence combination in the MADM tree model. The proposed approach is discussed in a generic framework for document ranking, which is evaluated using document queries in the domain of power substations. Results show that the proposed approach provides an enhanced solution to the document ranking, and the accuracy of the ODSE searches, which is evaluated by precision and recall methodology, has been improved significantly with ER embedded, compared to three existing document search engines.

The second part of the thesis is a logical continuation of study in the aspect of document search engines, as clustering methods aim to organise a document repository into a set of meaningful clusters automatically, which provides an efficient way for power engineers to browse and navigate the required documents. Three recently proposed CC algorithms, i.e., non-negative matrix factorisation-based CC algorithm (NNMF-CC), weighted partition via kernel method-based CC algorithm (WPK-CC), and information theory-based CC algorithm (INT-CC), in which WPK-CC is the most advanced one according to a set of simulation studies on selected datasets, are demonstrated in this thesis. In addition, ontology is utilised in this research to handle the document repository representation, in which a SONT-based vector space model (VSM), is firstly proposed based on Wordnet-based VSM. Moreover, a genetic algorithm (GA) is employed to solve the objective function of the best-performing CC algorithm, i.e., WPK-CC. The impacts of different mechanisms of the genetic operators in GA are also analysed. Meanwhile, this approach not only verifies that the three crucial factors in the clustering, i.e., document repository representation, selection of clustering algorithm and method of solving the consensus function, are capable of improving the document clustering result significantly, but also seeks out the comprehensive settings for the PSD clustering and contributes to the AM of power substations.

The third part of this thesis is dedicated to exploring a formalised framework that can perform uncertainty reasoning in ontology. A probabilistic diagnosis system, which provides quantified confidence support if uncertainties occur and adopts rules with interlinked relationships to form a PTFD-related knowledge base, is developed for the power transformer. The framework provides a set of structural translation rules to convert the web ontology language (OWL) into a BN directed acyclic graph. Meanwhile, algorithms of knowledge integration are employed to modify an existing BN with newly obtained probabilistic knowledge rather than rebuilding a new BN. Finally, the proposed ontology-based BN, which supplements OWL with additional abilities for representing and reasoning with uncertainty, is demonstrated by a small-scale PTFD system for an illustration purpose.

Contents

List of Figures **xi**

List of Tables **xiv**

List of Abbreviations **xvi**

1 Introduction **1**

1.1 Background of Asset Management for Power Systems 1

1.2 Asset Management Aspects concerned in this Thesis 3

1.2.1 Document searching for a power substation-related document repository 3

1.2.2 Cluster analysis of a power substation-related document repository 4

1.2.3 Power transformer fault diagnosis system 5

1.3 Brief Reviews of related Research Areas 6

1.3.1 Document searching in information retrieval 6

1.3.2 Document clustering in information retrieval 7

1.3.3 Knowledge representation in power transformer fault diagnosis 9

1.4 Motivation and Objectives 10

1.4.1 Drawbacks of traditional ontology-based document search engines 10

1.4.2 Bottlenecks of current document clustering methods 11

1.4.3 Limitations of conventional ontology-based power transformer fault diagnosis systems 12

1.4.4 Objectives of this research 12

1.5 Thesis Outline 14

1.6 Major Contributions of this Research 15

1.7 Publications 17

2 Background Knowledge and Literature Review **19**
2.1 Ontology . . . 20
2.1.1 The Semantic Web . . . 21
2.1.2 Ontology languages . . . 23
2.1.3 Description logics . . . 25
2.1.4 A generic process of building domain ontology models . . . 26
2.2 Information Retrival . . . 27
2.2.1 Historical literature review of information retrieval . . . 27
2.2.2 Mathematical models in information retrieval . . . 30
2.3 Optimisation Techniques concerned in this Thesis . . . 34
2.3.1 Simulated annealing algorithm . . . 36
2.3.2 Genetic algorithms . . . 38
2.4 Bayesian Networks . . . 46
2.4.1 Basics of Bayesian networks . . . 46
2.4.2 Bayesian networks . . . 47
2.4.3 Conditional probability tables . . . 48
2.4.4 Probabilistic inference . . . 48
2.5 Summary . . . 51

3 Ontology-based Document Search Engine Using Evidential Reasoning **52**
3.1 Introduction . . . 53
3.1.1 Query expansion by an ontology model of power substations 53
3.1.2 Using the multiple attribute decision making tree model to present expanded queries . . . 57
3.2 Evidential Reasoning for Document Ranking . . . 60
3.2.1 Brief introduction of the evidential reasoning . . . 60
3.2.2 Dempster-Shaper combination rules for evidence combination . . . 60
3.2.3 Integrating basic probability assignments with evidential reasoning algorithm . . . 61
3.2.4 Methodology for assigning weights to the attributes . . . 63
3.2.5 Document ranking by Dempster-Shafer combination rules . 64
3.3 Document Search Engines Designed in this Thesis . . . 65
3.4 Simulation Studies . . . 73
3.4.1 Implementation configuration . . . 73
3.4.2 An illustration for the purposed document search engine . . 74
3.4.3 Performance evaluation method . . . 78
3.4.4 Results and discussion . . . 80
3.5 Summary . . . 84

4 Performance Evaluation of Clustering Algorithms **85**
4.1 Introduction . . . 86
4.1.1 Common applications of clustering . . . 87

4.1.2 Document clustering . . . 88
4.1.3 Clustering algorithms . . . 88
4.1.4 Limitations of single clustering algorithms . . . 91
4.1.5 Consensus clustering . . . 91
4.2 Consensus Clustering Algorithms . . . 94
4.2.1 Non-negative matrix factorisation-based consensus clustering algorithm . . . 95
4.2.2 Weighted partition via kernel-based consensus clustering algorithm . . . 102
4.2.3 Information Theory-based consensus clustering algorithm . 106
4.3 Validation Methods . . . 108
4.3.1 Internal validation . . . 109
4.3.2 External validation . . . 110
4.4 Simulatition Studies of the Clustering Algorithms . . . 111
4.4.1 Data collections . . . 111
4.4.2 Implementation configuration . . . 112
4.4.3 Simulation results . . . 112
4.5 Summary . . . 118

5 Consensus Clustering for Ontology-embedded Document Repository of Power Substations using Kernel-based Method 120
5.1 Introduction . . . 121
5.2 Document Repository of Power Substations with Ontology-based Vector Space Model and Term Mutual Information . . . 122
5.3 Simulation Studies . . . 127
5.3.1 The impact of ontology model for document representation . 127
5.3.2 The improvement of GA-embeded kernel-based consensus clustering algorithm . . . 130
5.3.3 The performance evaluation of genetic operators for proposed consensus clustering algorithm . . . 134
5.4 Summary . . . 142

6 Ontology-based Bayesian Networks for Power Transformer Fault Diagnosis 143
6.1 Introduction . . . 143
6.2 Knowledge Integration . . . 145
6.2.1 Related works . . . 146
6.2.2 Iterative proportional fitting procedure . . . 147
6.2.3 Conditional-iterative proportional fitting procedure . . . 148
6.2.4 Extended-iterative proportional fitting procedure . . . 151
6.2.5 Decomposed-iterative proportional fitting procedure . . . 154
6.3 Ontology-based Bayesian Networks . . . 157
6.3.1 Structural translation . . . 157

6.3.2 Conditional probability table construction 158
6.3.3 Representation of probabilistic knowledge in OWL 161
6.3.4 Framework Implementation 161
6.4 A Simple Illustration for Power Transformer Fault Diagnosis using Ontology-based Bayesian Networks 164
6.4.1 Building the initial ontology-based BN 164
6.4.2 Modification by new probabilistic knowledge and belief updates 166
6.5 Summary 174

7 Conclusions **175**
7.1 Summary 175
7.2 Limitations of the Present Study and Suggestions for Future Work . 178

References **182**

List of Figures

1.1 Asset management aspects addressed in this thesis 2

2.1 Common applications of ontology techniques 20
2.2 An example of an RDF/XML graph 22
2.3 Ontology defined for a specific domain 23
2.4 A generic process of developing SONT and ontology-based BNs . . 28
2.5 The general components of an IR model 30
2.6 Development of modern IR 30
2.7 VSM model 34
2.8 The working process of the SA 37
2.9 The working process of the GA 40
2.10 An illustration of RW 41
2.11 An illustration of LR 42
2.12 A five-node BN 47
2.13 BNs approaches for PTFD 50

3.1 Classes and hierarchies of SONT defined in Protégé editor 54
3.2 The class "Action" and its hierarchies 54
3.3 Individuals defined for "Fault diagnosis" 55
3.4 A QE process with SONT 56
3.5 A general structure of a MADM tree model 58
3.6 A tree model used for the combination of multiple relevance scores regarding to the synonyms (including the plural forms) and hyponyms of the query terms 59
3.7 The mechanism of SE_1 66
3.8 The mechanism of SE_2 67
3.9 The mechanism of SE_3 68
3.10 The mechanism of SE_4 69
3.11 An index server generation process with Lucene 70
3.12 Posting list 73
3.13 Average precision-recall curves with 10 unique-keyword queries . . 81
3.14 Average precision-recall curves with 10 combined-keyword queries 82

3.15 Average precision-recall curves with all 20 queries 82

4.1 Clustering results of different cluster numbers, i.e., 2, 3 and 4, respectively 86
4.2 The generic document clustering procedure 88
4.3 A twenty-object dataset clustered by three types of clustering algorithm 90
4.4 A nine-objects dataset with three underlying classes v.s. labelled clustering result 91
4.5 Three clustering results by different runs of k-means 92
4.6 The multiplicative update rules for NNMF 97
4.7 Data in the original space and the equivalent feature space 103
4.8 The experiment working process 113
4.9 Purity of each clustering algorithm for different datasets 117
4.10 F-measure of each clustering algorithm for different datasets 117

5.1 The flowchart of each simulation study, and the WPKGA in the case study 2 only concerns the *italic*terms 128
5.2 The comparison among the average purity of WPKGA of the entire population, the maximum purity of WPKGA in the population, and the purity of WPKSA 133
5.3 WPKGA for the MPSD CC with different *PopSize* 135
5.4 WPKGA for the MPSD CC with different selection mechanisms, and *PopSize* is 100, where "X" represents the generation at convergence for each mechanism; "Y" shows the average purity; "L" & "U" denote the lower and upper SD 137
5.5 WPKGA for the MPSD CC with different crossover rates, when *PopSize* is 100, and ELR selection applies 138
5.6 WPKGA for the MPSD CC with different crossover mechanisms, when *PopSize* is 100; ELR selection applies; and P_c is 0.7. The numeric values on the "X" axis represent the probability of binominal crossover 139
5.7 WPKGA for the MPSD CC with different mutation rates, when *PopSize* is 100; ELR selection; and crossover rate is 0.7 with a binominal crossover probability of 0.2 140
5.8 WPKGA for the MPSD CC with different mutation mechanisms, when *PopSize* is 100; ELR selection; crossover rate is 0.7 with a binominal crossover probability of 0.2; mutation rate is 0.15; and the range of the adaptive mutation rate is $[0.05, 0.2]$ 141

6.1 A four-node BN with its CPTs 148
6.2 K-L divergence of $I(Q_k(X)||Q_0(X))$ using IPFP 149
6.3 Resulted BN with constraint set R_1 using IPFP 150
6.4 Resulted BN with constraint set R_2 using IPFP 151

6.5 K-L divergence of $I(Q_k(X)||Q_0(X))$ using E-IPFP 154
6.6 Resulted BN with constraint set R_2 using E-IPFP 156
6.7 L-node owl:complementOf (LNC) and its CPT 158
6.8 L-node for owl:disjointWith (LND) and its CPT 158
6.9 L-node for owl:equivalentClass (LNE) and its CPT 159
6.10 L-node for owl:unionOf relation (LNU) and its CPT 159
6.11 L-node for owl:intersectionOf (LNI) and its CPT 160
6.12 Properties of class "MarginalProb" and "ConditionalProb" 161
6.13 The flowchart of the ontology-based BN for PTFD 163
6.14 Translated BN for the union relation of the insulation fault example 164
6.15 Translated BN with the initial belief bars and CPTs ($\%$) of C-nodes for the insulation fault example 167
6.16 Modified BN by constraint set R_3 168
6.17 Modified BN by constraint set R_3, when "insulation fault" is true . . 169
6.18 Modified BN by constraint set R_3, when "insulation fault" and "water occurs" are true 169
6.19 Modified BN by constraint set R_4 171
6.20 Modified BN by constraint set R_4, when "insulation fault" is true . . 172
6.21 Modified BN by constraint set R_4, when "insulation fault" and "water occurs" are true 172
6.22 Modified BN by constraint set R_4, when "fault in oil" is true 173
6.23 Modified BN by constraint set R_4, when "water in oil" is true 173

List of Tables

2.1 Advantages and disadvantages of OWL sub-languages 24
2.2 OWL constructors for classes versus DLs concepts 25
2.3 OWL class relationships versus DLs inclusions 26

3.1 Grade in the AHP 63
3.2 Term dictionary 72
3.3 Statistical information of the PSD in the simulation studies 74
3.4 Queries employed for the simulation studies 75
3.5 Two documents used for ranking 76
3.6 Relevance scores in the factor level 76
3.7 Normalised relevance scores in the factor level 76
3.8 Relevance scores in the attribute level 77
3.9 Evaluation metrics 79
3.10 Average precisions of four search engines on recall levels of 10% and 20% 83

4.1 Two principle steps of consensus clustering 92
4.2 The SA for solving the objective function of WPK-CC 106
4.3 Three sample datasets from UCI machine learning repository 111
4.4 Four document repositories 111
4.5 The experiment parameter settings of each algorithm 112
4.6 Validated results for the Iris dataset 114
4.7 Validated results for the Wine dataset 114
4.8 Validated results for the Glass dataset 114
4.9 Validated results for the bbcsport document repository 116
4.10 Validated results for the bbc document repository 116
4.11 Validated results for the TDT2-6 document repository 116
4.12 Validated results for the PSD 116

5.1 The comparison between the term-based VSM and the SONT-based VSM 125
5.2 Purity analysis of each clustering algorithm on the PSD and the MPSD 130
5.3 Termination status of the SA and the GA related algorithms 132

5.4 Results of the PSD CC using WPKSA and WPKGA 134
5.5 Purity and convergence of WPKGA with different *PopSize* on the MPSD 135

6.1 The original JPD ($Q_0(X)$), JPD of the first fitting step ($Q_1(X)$)) and the final JPD (($Q^*(X)$)) of the four-node BN 149
6.2 Resulted JPD $Q^*(X)$ with constraint set R_2 using IPFP 152
6.3 Extracted $P_1(X_1, X_4)$ from resulted JPD 152
6.4 Resulted JPD with constraint set R_2 using E-IPFP 155

List of Abbreviations

AHP	Analytic Hierarchy Process
AM	Asset Management
BNs	Bayesian Networks
CC	Consensus Clustering
CI	Connectivity Internal validation
C-IPFP	Conditional IPFP
CPT	Conditional Probability Table
DI	Dunn index Internal validation
D-IPFP	Decomposed-IPFP
DLs	Description Logics
DS	Dempster Shafer
E-IPFP	Extended-IPFP
ELR	Elitism combined with LR
ER	Evidential Reasoning
ERW	Elitism combined with RW
GAs	Genetic Algorithms
INT-CC	Information Theory based CC
IPFP	Iterative Proportional Fitting Procedure
IR	Information Retrieval
JPD	Joint Probability Distribution
LNC	L-nodes for owl:complementOf
LND	L-nodes for owl:disjointWith
LNE	L-nodes for owl:equivalentClass
LNI	L-nodes for owl:intersectionOf
LNU	L-nodes for owl:unionOf
LR	Baker's Linear Ranking selection
MADM	Multiple Attribute Decision Making

MPSD	Modified PSD
NNMF	Non-Negative Matrix Factorisation
NNMF-CC	NNMF based CC
NRW	Non-Randomness-based Weighting model
ODSE	Ontology-based Document Search Engine
OWL	Web Ontology Language
PSD	Power Substation-related Document depository
PTFD	Power Transformer Fault Diagnosis
PVIs	Property Validity Indexes
QE	Query Expansion
RW	Roulette Wheel selection
SA	Simulated Annealing
SD	Standard Deviation
SI	Silhouette width Internal validation
SONT	Substation ONTology
tf-idf	term frequency-inverse document frequency method
TREC	Text REtrieval Conference
VI	Variance Internal validation
VSM	Vector Space Model
WPK-CC	Weighted Partition via Kernel based CC
WPKGA	GA embedded WPK based CC
WPKSA	SA embedded WPK based CC

Chapter 1

Introduction

1.1 Background of Asset Management for Power Systems

Asset Management (AM) normally concerns finance, economy, engineering, etc, with the goal of providing the relevant level of service to the physical assets in the most cost-effective manner [1] [2]. In general, there are three main types of asset, i.e., intangible assets, physical assets, and financial assets [3]. In detail, intangible assets consist of operating licences, knowledge, skills gained from experience, and other factors. Physical assets contain apparatuses, instruments, equipment, etc. Financial assets are normally considered to include financial instruments, equity accounted investments, etc. A completed AM system of an organisation must meet the requirements considering these three parts, in order to improve the values of the organisation [4].

With the increasing pressures from both industrial growth and capital expenditure, power systems are under considerable strains and suffering from the difficulties of conflicting objectives related to asset organisation and utilisation [2]. To solve these issues, a large number of systerm operators in the power industry are concerned in AM, and many efforts have been made to implement AM into power systems [4–7]. The main purpose of an AM programme is to manage the each type of the asset and improve the associated performance optimally [2] [6].

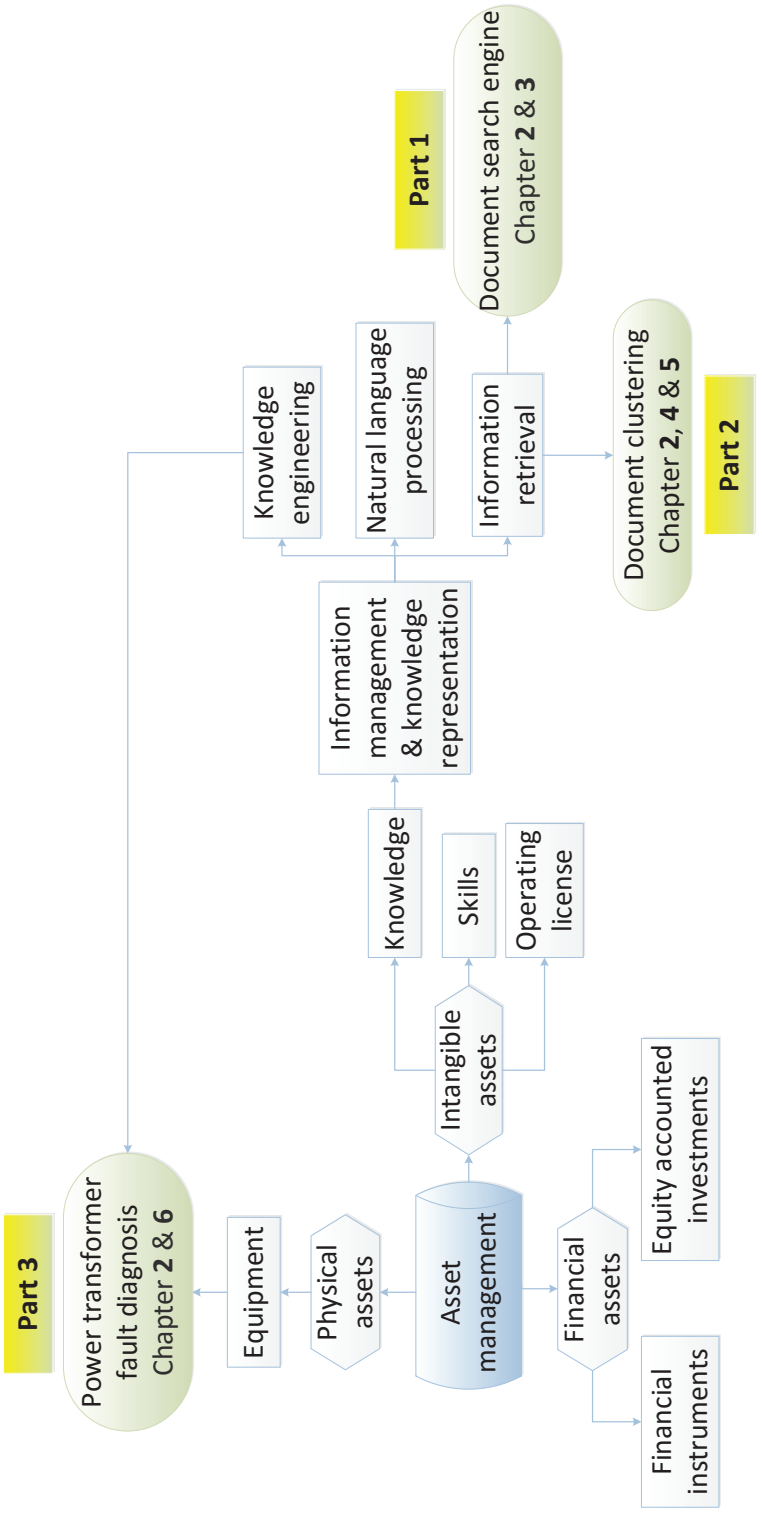


Figure 1.1: Asset management aspects addressed in this thesis

1.2 Asset Management Aspects concerned in this Thesis

A power substation is one of the most important apparatuses in power systems. Among the basic elements of power system AM mentioned in Section 1.1, this research mainly focuses on two types of AM in power substations, i.e., a power substation-related document repository (PSD) and the power transformers.

Three intelligent approaches are presented in this thesis, including an improved document ranking method in ontology-based document search engine (ODSE) using evidential reasoning (ER) for PSD, consensus clustering (CC) algorithms for ontology-based PSD, and an ontology-based Bayesian networks (BNs) model for power transformers fault diagnosis (PTFD). The relationship between these approaches and the AM of power substations is summarised in Figure 1.1, in which the corresponding chapters are illustrated concerning each approach employed in this thesis. The details are introduced in the following sections.

1.2.1 Document searching for a power substation-related document repository

A large domain document collection always contains massive of documents concerning this research domain only. These documents normally consists of knowledge, skills, strategies, key concepts, latest research findings, etc. discussing the domain-related research or industrial topics. Thus, a domain document repository can be regared as a set of intangible assets in the corresponding field, of which an effective management can provide the experts in this field with fast knowledge retrieval so that both learning and working efficiency can be improved significantly. Nowadays, a great number of text files, which are collected in power company databases, are causing considerable difficulties in information retrieval (IR) [8] and thus result in reduced efficiency of substation information reuse. In current power enterprises, a document search engine is usually recognised as a common tool for the IR of power substations. However, several drawbacks still exist

in such a system that can greatly reduce the accuracy of an IR process. Therefore, in the first part of this thesis, an ER-based ODSE is introduced for accurately retrieving user-sought information of power substations. In addition, the proposed ODSE is carried out on the PSD, which has been built by our research group [9], and contains more than 100,000 power substation-related documents (all in English) generally including six aspects, i.e., system operation and control, PTFD, power generation, voltage stability safety evaluation, intangible information management, and other optimised strategies for maintenance.

1.2.2 Cluster analysis of a power substation-related document repository

The domain text materials are collected from different sources, e.g., articles, technical reports, emails of the power system engineers consulted, etc., in different file formats, e.g., pdf, word, html, text, etc. If the number of the text materials is not large and unstructured, each text file can be classified and allocated into a tree structure based on the topics of the text files by the domain experts manually. However, if a large number of unstructured files from different sources is obtained simultaneously, it is essential to manage these text knowledge automatically. Cluster analysis or clustering methods is a tool to accomplish such purpose, aiming to group the text file into a list of categories automatically [10]. Briefly, documents with similar topics have high similarities, while documents within the same cluster are different from documents in the other clusters. As an unsupervised learning strategy, clustering has the automated processing capacity for documents without being concerned with the training process, e.g., classification (supervised learning), and annotating the documents manually in advance [11].

From IR point of view, it aims to search targeted text resources from a large text repository. A search engine always returns thousands of results in response to a user's query, making it difficult for users to browse or to identify the relevancy of each response. In this case, the document repository could be organised into a set of meaningful clusters automatically, which provides an efficient way for power

engineers to browse and navigate. In addition, as the documents within the same cluster have high similarities, the un-retrieved documents but in the same cluster with the retrieved document can also be regarded as a relevant search. Thus, the search results can be widened by adding other documents in the same cluster. As a consequence, more relevant search results are obtained with the aid of clustering so that the accuracy of document search can be improved. The cluster analysis of the PSD is introduced in the second part of this thesis.

1.2.3 Power transformer fault diagnosis system

A power transformer, which has the characteristics of capital intensive, robust, long-lived, and not easily relocatable, is a major component of a power substation. Normally, a transformer has operated for at least 40 to 50 years, which can pose high risks for the safety of operations. Also, a power transformer always costs millions of pounds and weighs approximatedly 250 tons [12], so that the replacement of a power transformer is extremely costly. AM for power transformers aims to balance the cost, improve the performance and reduce potential risks, which is normally achieved by fault diagnosis and management decisions [13]. If an early fault of a power transformer is detected before it leads to a disastrous fault, the unit may be repaired on site or replaced accordingly with a scheduled arrangement. An automation system is very important to the operation of modern power transformers. It is a good way to provide conceptually specific description and build the foundation for knowledge sharing [14] [15]. The power substation maintenance is always regarded as a "knowledge-based" domain, as the decision of fault diagnosis in power substations is determined from the comparison of current status and experience obtained from similar situations in the past [16]. Also, the ontology knowledge has been proven of great importance for the interconnection and construction of large-scale software systems to deal with power substation-related topics in many previous investigations [17–19]. The third approach of this thesis focuses on the ontology-based PTFD with additional function of uncertainty reasoning.

1.3 Brief Reviews of related Research Areas

1.3.1 Document searching in information retrieval

In a power company, most power substation-related information are digitally stored, e.g., technical reports, a large number of backup documents of substations, plenty of substation maintenance records, and other documents are increasing rapidly. Therefore, it becomes more important that an intelligent IR tool should be used concerning both practical and investigative aspects.

In the past few years, a variety of IR models have been developed for document ranking in document search engines, e.g., the probabilistic model, inference network model, and vector space model (VSM), etc., of which VSM is the dominant one [20]. The main function of a search engine is to discover the information in relation to a query input. When a query is submitted, which is typically in the format of several keywords, a search engine retrieves relevant documents according to the query. The result then is usually returned as a list of relevant documents that are ranked in a descending order of their relevance scores concerning the query. In practice, a well-formatted query can explicitly illustrate the required information by a user and thus lead to a high search accuracy. However, the search accuracy of a search engine may be greatly restricted due to unclear and incomplete queries. In order to overcome this problem, query expansion (QE) has been introduced as a viable solution. Generally, QE presents a process of expanding a query input with its related terms [21]. With an expanded query, the documents that do not contain the same keywords of the original query, but are correlative to such inferred terms, can be retrieved. Consequently, the search scale in a search process is suitably broadened and a more accurate result may be obtained by retrieving more relevant documents.

Various QE techniques have been developed that are mainly based upon the mechanisms of relevance feedback [21] and statistical term co-occurrence [22]. In most cases, an improved search accuracy can be achieved using these techniques. However, a significant drawback of the above two QE techniques is that the related terms of a query input are obtained by analysis of the context of documents stored in

a document repository. Thus, if there are not sufficient documents used for analysing before a search process, the relatedness between related query terms and an original query term cannot be ensured [2] [9].

As words always have various meanings, the document search engine cannot distinguish which meaning the user intends to input. For instance, the query "switch" means "to turn on or turn off a button", but also means "change". If this kind of query is submitted, both of the meaning-related documents will be retrieved. As a result, the retrieved accuracy will decrease. In this case, the ontology-based QE is proposed. In theory, an ontology model is 'an explicit specification of a conceptualisation' [23]. It is a formal representation of the entities that can exist in a domain of application as well as the explicit relationship between the terms in different hierarchies [15].

One of the most advanced ontology models for QE in IR is to use WordNet as a knowledge model, and in [24], a set of tests was carried out based upon the Text REtrieval Conference (TREC) document repository [25]. Briefly, WordNet is a lexical database, which can be regarded as an ontology model, consisting of large amount of words, and their synonyms, hyponyms, etc [26]. A semantic IR system introduced by Castells, in which a knowledge base is constructed by annotating documents of a document repository, uses an ontology-based semi-automatic annotation mechanism [27]. The test results have shown that the semantic IR model achieves high performance in IR compared with that of a keyword-based IR system.

1.3.2 Document clustering in information retrieval

Clustering is a multivariate statistical procedure that starts from a dataset containing information about a sample of entities, aiming to re-organise these entities into relatively homogeneous groups [28]. In other words, clustering is a general concept for creating a classification of objects from unstructured data [29]. Furthermore, clustering is the division of data into groups of similar objects. Each group is called a cluster, consisting of objects that are similar among themselves and dissimilar from objects of other groups.

Document clustering has become an increasingly crucial analysis tool for large collections of documents. It is based on the idea of cluster analysis, in which relevant documents tend to be more similar to each other than to non-relevant documents, and to compose the same cluster [10]. Document clustering improves the precision or recall in IR systems [8] [30] and is an efficient way of finding the nearest neighbours of document [31]. Basically, document clustering consists of three steps, i.e., document dataset pre-processing, applying suitable clustering algorithms to obtain a pre-defined number of document clusters, and results validation [32].

The first step of document clustering, i.e., pre-processing, is similar to document searching in IR, which aims to provide a way to represent a document in a mathematical model. In a document search engine, the relevance score between a query vector and a document is mostly computed based on VSM [22] [32–35]. In document clustering, terms in the document repository can be regarded as a query. As a result, a document-term matrix (or term-document matrix) is generated based on VSM. Subsequently, weightings of all the terms in each document are calculated by term frequency-inverse document frequency method (tf-idf). Therefore, the original document dataset is pre-processed into a weighted document-term matrix. Each row of this matrix represents a document in the document repository, and each column stands for terms with assigned weights. However, using VSM for document representation ignores the relationships among important terms that may not appear simultaneously in the document repository. In other words, they only relate documents that use identical terminology [36]. This problem is similar to the documents retrieval in IR. For instance, one document concerning transformer diagnosis using condition assessment and another document concerning the same topic using fault isolation sometimes may be clustered into different groups.

Meanwhile, a large variety of clustering algorithms have been proposed including k-means, hierarchical clustering algorithms, fuzzy c-means, and others [10]. Many researchers have concluded that the results obtained from varied single clustering algorithms, or even the same algorithm with different initial states or iterative steps, are very dissimilar. It is not appropriate to decide which clustering

result is correct or not, as they are all obtained by using equally plausible clustering algorithms [37]. In this case, the concept of CC, which is a combination of various clustering solutions, is proposed [37] [38]. It aims to achieve a comprehensive result with better performance than each single clustering algorithm, and the solution should be similar to all the clustering results [38].

1.3.3 Knowledge representation in power transformer fault diagnosis

Power substations concern condition assessment, fault diagnosis, operation decision-making, and maintenance of power transformers [13]. As a crucial device, accurate transformer incipient fault diagnosis can extend the service life of power transformers and further increase grid reliability to avoid power blackouts. In industrial practice, a number of methods, e.g., dissolved gas analysis, thermal modelling, partial discharge analysis, and frequency response, have been used for such purpose [39] [40]. These methods can warn about impending problems, provide early diagnosis, and ensure a transformer's maximum uptime [41]. The actions taken on a power transformer are normally based on electrical experts' knowledge, experience, and expertise by comparing present and past measurement data [39]. In this case, experts from various areas should be involved in these operations to manage a large amount of information, which is associated with complex and comprehensive concepts and knowledge of power transformer operations. Therefore, advanced techniques with computers or machines for knowledge representation, automated data analysis, and decision-making, become more important in handling the complex data for power substations.

A knowledge base is a special kind of database, containing knowledge and information from a specific area. It is used to store and manage complex and comprehensive concepts as well as their relations. The knowledge-based systems utilise knowledge bases to solve complex problems. Transformer incipient fault diagnosis is conventionally based on an expert-system, which is a computer system simulating the decision-making ability of human experts and based on

a knowledge-based system. The expert-system is constructed with rule-based knowledge representation, since it is expressed by experts according to some rules, e.g., IF-THEN rules, to make inferences or choices [42]. Also, it consists of two components, i.e., knowledge base and inference engine.

Although an expert-system holds strong pertinence, it has weak extensibility. In knowledge engineering, the heart of the expert-system development process is to transfer the problem-solving expertise from a knowledge source to a program [40] [43]. Ontology is a mechanism that can represent domain knowledge in a machine readable manner, which has a underlying basis of formal logic [44]. It is a way of specifying explicitly the elements and their relationships in a domain such that they are reusable. In addition, the formal nature of ontology enables the integration of data from heterogeneous sources. In this case, some ontology-based systems have been proposed, and proven to be effective in PTFD [39] [45].

1.4 Motivation and Objectives

1.4.1 Drawbacks of traditional ontology-based document search engines

The accuracy of a search engine is enhanced with an ontology-based QE method. However, there still exist some drawbacks needed to be concerned in the ODSEs. For instance, most existing approaches focus more on the algorithms, seeking the best expanded query terms as well as their numbers so that the best search accuracy could be achieved. There is no existing mechanism employed to organise the expanded terms. Also, the hierarchical relationships among related terms, in our case, i.e., the original query term, synonyms and hyponyms are ignored. As a consequence, although mutual weights are assigned differently to a pair of terms in an expanded query, the emphasis of the original query could be biased in a search process. In addition, various methods are used to determine the relevance scores between the terms of an expanded query and a document. However, the relatedness between the expanded query and the document is normally calculated

by the weighted sum of these generated relevance scores, based upon VSM. Hence, the relevance scores generated by the query terms are treated independently during the combination process, which may reduce the accuracy of a final search result.

1.4.2 Bottlenecks of current document clustering methods

The basic idea of involving background knowledge in the document repository is to use WordNet, providing sets of synonyms and extending VSM. In this case, synonyms are resolved and more general concepts are introduced to identify related topics. Several strategies have been introduced in [46], including e.g., 'Add concepts', 'Replace terms by concepts', and 'Concept vector only'. Better clustering results can be achieved when each strategy is compared with the normal text pre-representation for selected document datasets. However, these strategies can either increase the dimensionality of the text data or decrease the amount of information of the raw dataset. Thus, they are not practical for solving the large volume document clustering problems. Also, limitations exist in the WordNet, as there is no synonym or hyponym mapped to a power substation concept in some cases.

In addition, many CC algorithms are originally designed for sample datasets, which have much smaller features than the document datasets. Also, there is no existing comparison study on both sample datasets and text datasets among these CC algorithms. Although there are still some studies discussing CC algorithms to cluster a document depository, e.g., [47] [48], they are all for a general document repository with testing purpose of the algorithms. There is no such application for a specific research domain (e.g., power substations) operating as a tool for power engineers or research experts to do both IR and clustering. Moreover, one type of the CC algorithms can be regarded as the optimisation problems [37], in which the settings of parameters may cause significant variations in the result. Thus, the parameter settings of the typical CC algorithm are worthy of further study.

1.4.3 Limitations of conventional ontology-based power transformer fault diagnosis systems

Conventional fault diagnosis systems either involve BNs to handle simple uncertainty reasoning, e.g., dissolved gas analysis data to reinforce the capability of the traditional dissolved gas analysis method [13], or implement an ontology model to explicitly express the interconnection between fault types and symptoms in order to provide inference without considering the uncertainties. There is no existing PTFD system concerning the benefits of both. The limitation of ontology in such a system is that it can only be utilised as a tool of knowledge representation. In real diagnosis systems, the number and type of signal collection increase dramatically with the increasing of the system scale and complexity, e.g., temperature, vibration, noise, discharging, etc. Knowledge can be collected from different sources, and it can be either mutually related or complementary. If new probabilistic information is obtained, an existing BN prefers being merged with the new knowledge rather than being reconstructed according to the new situation. The procedure to build a BN and construction of the conditional probability tables (CPTs) [49] are complex and time-consuming. It is essential to find a way to modify the original BN with new knowledge from different sources and avoid re-constructing a new BN.

1.4.4 Objectives of this research

This thesis presents the development of three intelligent power substations AM approaches to address the problems mentioned above:

- A novel approach to document ranking in an ODSE using ER is presented. A substation ontology (SONT), which is developed in the context of power substations, is used for QE. A multiple attribute decision making (MADM) tree model is proposed to organise expanded query terms. Then, an ER algorithm, based on the Dempster-Shafer (DS) [50] theory, is used for evidence combination in the MADM tree model. The proposed approach is discussed in a generic framework for document ranking, which is evaluated using document queries in the domain of the substation. The performance

of this approach is compared with both traditional methods and one of the recently developed methods.

- The document data representation for clustering in this research is inspired by Jing [51], in which a WordNet-based distance measure is proposed. In contrast, SONT is applied to modify the traditional VSM for document representation (SONT-based VSM), which is concerned with the semantic relations between terms. A new document representation is generated using a term mutual information matrix with the aid of SONT-based VSM. In addition, compared with two other novel CC algorithms, i.e., non-negative matrix factorisation-based CC (NNMF-CC) [52] and information theory-based CC (INT-CC) [53], weighted partition via kernel-based CC algorithm (WPK-CC) [54] is utilised to solve the CC issue for PSD. Meanwhile, the genetic algorithm (GA) is employed to WPK-CC for PSD, as there are limitations in the original WPK-CC for document clustering. Subsequently, selected mechanisms of the genetic operators in GA are compared and improved, resulting in comprehensive parameter settings for the PSD CC.

- This research aims to propose an ontology model for PTFD, in which one of the ontology languages, i.e., web ontology language (OWL) [14] is supplemented with additional expressive power for representing and reasoning with uncertainty. This task is achieved by combining with a BN, which is a probabilistic directed acyclic graph model for uncertainty knowledge representation and reasoning [55]. A set of translation rules, which is inspired by Ding and Peng [56], is employed for translating an OWL file into a BN. Also, the probabilistic knowledge iterative proportional fitting procedure (IPFP)-based algorithms [57–59] are utilised to refine an existing ontology-based BN model with probabilistic constraints. Finally, the proposed ontology-based BN is demonstrated by a small-scale PTFD illustration.

1.5 Thesis Outline

This thesis is structured as follows:

Chapter 2 : mainly provides the fundamental background knowledge concerned in this thesis, which begins with introducing the basics of ontology, including semantic webs, ontology languages, a brief review of DLs, and the applications of ontology. In addition, a historical literature review of IR and some mathematical models utilised in IR are presented, in which VSM is demonstrated in detail. Moreover, the optimisation techniques are briefly introduced, among which two types of meta-heuristics, i.e., simulated annealing algorithm (SA) and GA are presented. Finally, the basic concepts of BNs are given, including the fundamental basis of BNs, CPTs and probabilistic inference in a PTFD system using BNs. This chapter establishes the fundamental basis of the three approaches.

Chapter 3 : focuses on the development of the ER-based ODSE. It starts from an introduction of SONT for QE. Subsequently, the methodology of transferring an expanded query (with SONT) into the MADM tree model is discussed. Then, a brief review of the ER and the DS theory are introduced. Subsequently, the ER developed based on the DS theory for the evidence combination of the MADM tree model is explained. Furthermore, the implementation of integrating the ER algorithm with an ODSE is demonstrated. Four document search engines are designed for the purpose of comparison. Finally, the comparison results and performance of each search engine are evaluated.

Chapter 4 : mainly evaluates the impact of the selected CC algorithms. Firstly, the basics of clustering are introduced. Secondly, a review of single clustering algorithms is presented. Then, the idea and relevant information of CC are delivered in detail. In addition, three novel CC algorithms, i.e., NNMF-CC, WPK-CC, and INT-CC are presented. Finally, all the CC algorithms mentioned are applied to selected sample datasets and document repositories. Finally, the performance is evaluated by both internal and external validation methods.

Chapter 5 : presents the approach of CC for SONT-based PSD using WPK-CC. Firstly, a brief introduction is given, including the idea of implementing ontology to document data representation and a theoretical comparison between two optimisation methods. Secondly, SONT is applied to modify the traditional VSM, which is concerned with the semantic relations between terms. A new document representation is generated using a term mutual information matrix with the aid of SONT. Meanwhile, GAs are applied to WPK-CC in order to perform CC for PSD. Three simulation studies are designed, of which the results are evaluated in detail by the validation method of purity.

Chapter 6 : dedicates to the development of ontology-based BNs for PTFD. Firstly, a brief introduction of BNs and ontology in PTFD system is given. Secondly, the idea of knowledge integration is presented. Also, the related algorithms, e.g., IPFP, C-IPFP, E-IPFP, and D-IPFP, are illustrated. They are applied to modify BNs with probabilistic constraints. Subsequently, the ontology-based BNs is introduced in detail. Finally, a sample PTFD scenario based on the proposed ontology-based BNs is demonstrated.

Chapter 7 : concludes the thesis by giving a summary of the results of each chapter. Suggestions for possible future work are also listed.

1.6 Major Contributions of this Research

The major contributions arising from this thesis are summarised as follows:

- In the first and second parts of this thesis, a domain ontology model, i.e., SONT, has been extended and built based on Yang's work [2] in the context of a power substation. SONT has been applied to two approaches in this thesis, i.e., QE and modifying the traditional VSM with semantic relations between terms. A well-defined ontology model can be utilised by other ontology-based research regarding power substations. Also, it is capable of developing other power substation-related projects and reduces relevant

costs for re-developing a new ontology model. Meanwhile, as an ontology model, SONT is flexible and expandable with no ambiguity. In addition, both approaches are applied on a power substation-related document repository that has been constructed by our research group. The PSD is a large collection of document that concerns only power substation including technical reports, substation maintenance records, published papers, etc. The PSD restricts the searching process to be under the domain of power substations, and the search engine improves both the recall and the precision of searching so that the retrieved documents are highly relevant to the power engineers' requirements. Subsequently, power engineers can provide relevant actions to power substations.

- In the first approach of this thesis, the work on the development of an ODSE using ER is presented. The terms of an expanded query is organised into an MADM tree model during a search process. The ER algorithm, based on the DS theory, is then proposed for the combination of the relevance scores generated between the terms of the expanded query and a document. Test results show that the proposed ER-based approach is a suitable solution for document ranking in an ODSE and that the search accuracy of an ODSE has been improved, compared with that of an ODSE without the proposed ER-based approach and one of the recent proposed IR models.

- In the second approach of this thesis, three advanced CC algorithms have been reviewed. There is no existing quantitative comparison among these algorithms when applied to the document datasets, especially for the PSD. This thesis describes, for the first time, the original work on the development of document clustering concerning the combination of three aspects, i.e., document dataset representation, CC algorithm implementation, and methodology for modifying the existing CC algorithm. In addition, the mechanisms of GA have been discussed and compared. Typically, a document clustering mutation scheme, i.e., DC-mutation, has been proposed, which is based on the Baker's linear ranking method (LR) [60]. With the proposed approach,

clustering can be either implemented to the entire PSD and other power substations related-document repositories or to a retrieved document set so that users only need to browse a small number of accurately retrieved results. As a consequence, this approach improves the efficiency of knowledge acquisition for power engineers or academic staff and contributes to power substation asset management.

- In the third approach of this thesis, an application framework has been presented to supplement ontology-based PTFD systems with uncertain knowledge representation and reasoning based on BNs. This framework aims to quantify a certain fault with BNs, based on the knowledge embedded in a transformer ontology regarding relationships of fault types and symptoms. This approach is the first PTFD system to combine the benefits of ontology representation and BNs reasoning. In addition, knowledge integration methods are applied to this framework so that if new probabilistic knowledge is obtained, it can be regarded as a probabilistic constraint and merged into the existing diagnosis system rather than re-constructing a new BN. As a consequence, the proposed approach is capable of performing probabilistic inference, and acts as a supportive tool for electrical engineers to diagnose complex system faults.

1.7 Publications

List of the publications produced from this research:

1. W. H. Tang, **L. Yan**, Z. Yang and Q. H. Wu, "Improved Document Ranking in Ontology-based Document Search Engine Using Evidential Reasoning", *Software IET*, vol.8, no.1, pp.33,41, February 2014.

2. **L. Yan**, C. H. Wei, W. H. Tang and Q. H. Wu, "Development of Novel Asset Management Systems for Power Transformers based on Ontology", *IEEE PES Asia-Pacific Power and Energy Engineering Conference*, Hong Kong, 8-11 December 2013.

3. **L. Yan**, Y. L. Xin and W. H. Tang,"Consensus Clustering Algorithm for Asset Management of Power Systems, *The 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT 2015)*, Changsha, China, 26-29 November, 2015.

4. **L. Yan**, W. H. Tang, Q. H. Wu and J. Smith, "Consensus Clustering for Ontology-embedded Document Repository of Power Substation using Kernel-based Method", Submitted to *CSEE Journal of Power & Energy Systems*, February, 2016.

Chapter 2

Background Knowledge and Literature Review

Before presenting the three approaches, it would be helpful to introduce some fundamental background knowledge concerned in this thesis. Section 2.1 provides the initial motivation of the Semantic Web and the relationship between ontology and the Semantic Web, followed by the common applications of ontology. Meanwhile, ontology applied in IR and knowledge engineering are highlighted by introducing the generic process of developing domain ontology models. Subsequently, the Semantic Web, ontology language and DLs are presented. In Section 2.2, a historical literature review of IR is firstly introduced. Then, three mathematical models utilised in IR, i.e., Boolean model, probabilistic model and VSM, are illustrated. Furthermore, VSM combined with tf-idf method, which is implemented in this thesis to process the PSD, is also demonstrated in detail. Median partition, which belongs to one of the CC categories and is based on the optimisation problem, is under the scope of this research. In this case, the optimisation techniques are introduced in Section 2.3. Specially, the common genetic operators of the GA that are employed in this thesis to modify the GA for PSD CC are illustrated. Finally, Section 2.4 illustrates the essential concepts of BNs, including the basic concepts in probability theory, CPTs of a BN, and the probabilistic inference with a BN in PTFD systems, etc.

2.1 Ontology

Tim Berners-Lee, who invented the World Wide Web (WWW) and hypertext markup language (HTML), brought up the concept of the Semantic Web firstly in 1998 [61]. He aimed to provide meanings or semantics to the existing web data. The Semantic Web activity is a cooperated project by WWW consortium (W3C), US defense advanced research project agency, and EU information society technologies (IST) programme. In early 1990s, ontology was used as a technical term in computer science [23], and it is regarded as the core of the Semantic Web. As mentioned in Section 1.3.1, the most recognised definition of ontology is 'a formal specification of a conceptualisation' that was proposed by Thomas Gruber in 1993 [23]. Thus, a well-defined ontology model provides a 'clear and rigorous vocabulary' [62].

Ontology techniques have been applied to many intelligent systems [15] and been investigated in many areas, including IR [24] [27] [63] [64], knowledge engineering [65], natural-language processing [66], etc. The most common applications of ontology techniques are summarised in Figure 2.1 [67].

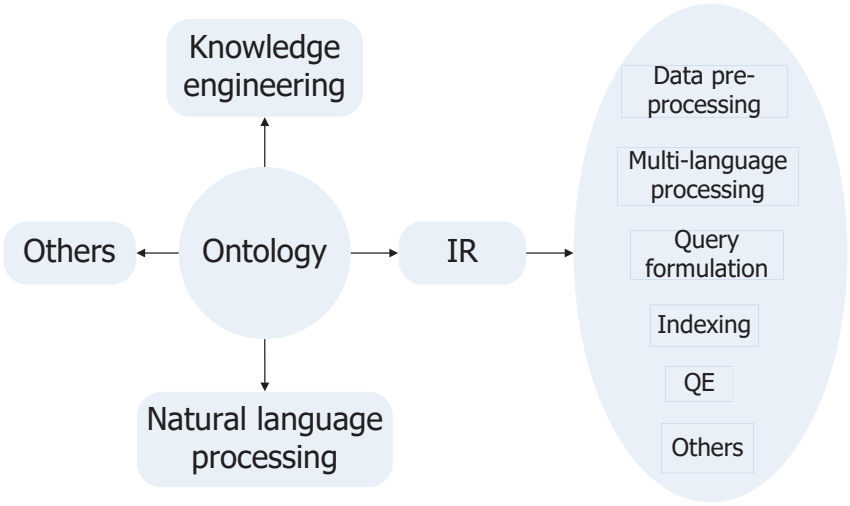


Figure 2.1: Common applications of ontology techniques

In practice, ontology techniques focus on constructing domain ontology models, in which various concepts and their relations only concerning this domain, are described. In our case, a domain ontology demonstrates power substation-related

concepts, properties of the concepts, and interrelations. A domain ontology model integrates heterogeneous information and provides common vocabulary for knowledge sharing. In this thesis, ontology has been applied into two aspects, i.e., IR and knowledge engineering, consisting of three applications as mentioned in Section 1.4.4. The related information, e.g., relevant theories, methodologies, involved mathematics, framework implementations, etc. are presented in detail within different chapters as illustrated in Figure 1.1.

2.1.1 The Semantic Web

The contents of a web page can be easily viewed and comprehended by human, but computers are not able to comprehend web page directly. The departure point of the Semantic Web is to make web pages understandable by computers, and relevant semantics should be embedded in the existing web data. Basically, in order to promote the data exchange between humans and computers, the Semantic Web aims at providing a higher level of data, which is named metadata (data about data), to the current WWW, so that machines can understand. The Semantic Web is recognised as an efficient tool for information representation and sharing on the web. Resource description framework (RDF) [68], which is based on extensible markup language (XML) syntaxes [69], is a standard model for that purpose. The XML language used by RDF is called RDF/XML. It unambiguously interprets identifiers by using uniform resource identifier (URI) and XML (xmlns) namespace mechanism [70] declarations enclosed in an opening rdf:RDF tag. Briefly, RDF is a framework that supports resource description or metadata. The basic items of RDF are called RDF triples, including "subject" (a specific resource), "predicate" (a property), and "object" (value). It can be concluded as "the $< subject >$ has $< predicate >< object >$". The example of an RDF/XML document presented in Listing 2.1 describes that "http://www.w3.org/" has a title of "World Wide Web Consortium" [71]. Also, it can be illustrated as an "RDF graph" as shown in Figure 2.2.

RDF Schema (RDFS) [72] is a simple data-typing model of RDF. It has the function of controlling the set of terms, properties, domains, and ranges of

properties. The “rdfs:subClassOf” and “rdfs:subPropertyOf” are used to define resources.

Listing 2.1: An RDF/XML example from http://w3.org/RDF/Validator

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.w3.org/">
    <dc:title>World Wide Web Consortium</dc:title>
  </rdf:Description>
</rdf:RDF>
```

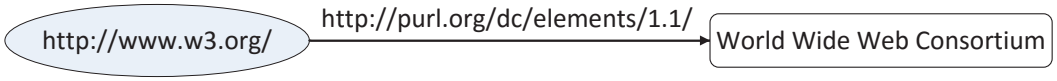


Figure 2.2: An example of an RDF/XML graph

Also, for an advanced system, it is not enough that simply describes the concepts and their relations. The limitation of RDF is that it is unable to perform reasoning, and all the relationships between classes and properties in a domain can not be identified in RDFS [73]. In order to overcome the limitation of RDF, another XML-based language, i.e., DARPA agent markup language (DAML), was proposed in 2000 [73]. DAML addresses the shortcoming of RDFS, providing a machine with the ability of making simple inferences. For instance, if “fatherOf” is a “subProperty” of “parentOf”, and “Sam” is the “fatherOf” “Peter”, the machine can infer that “Sam” is also the “parentOf” “Peter” [73]. Meanwhile, ontology inference layer (OIL) [74] is an extension of RDFS, providing a layered approach for web-based representation. The language syntax of OIL is also based on XML, of which the formal semantics and reasoning services, are from DLs. DAML+OIL [75] combines features of both, supplying a richer set of vocabularies to describe the resources on the web. Afterwards, W3C recommendeds web ontology language (OWL) [76], which is based on DAML+OIL, aiming to give a standard language for semantic information representations, and to provide more features than RDFS

for defining classes, properties and interrelations so that software applications can easily understand the instance information.

2.1.2 Ontology languages

The concept of ontology in philosophy is about the study of the nature of beings and existence in the universe. In natural science, ontology is built based on commonly agreed vocabularies for representing and sharing knowledge [77]. Another rigorous definition of ontology is that 'the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality' by Smith in 2003 [78]. The nature of ontology is explained as 'providing a definitive and exhaustive classification of entities in all spheres of beings' [78]. In the computer science community, the term ontology in the context of information exchange and knowledge sharing refers to formal descriptions of particular domains and provides a common understanding about this domain [79]. According to Thomas Gruber's definition of ontology, the relationship among a specific domain, conceptualisation model and an ontology specification is illustrated in Figure 2.3.

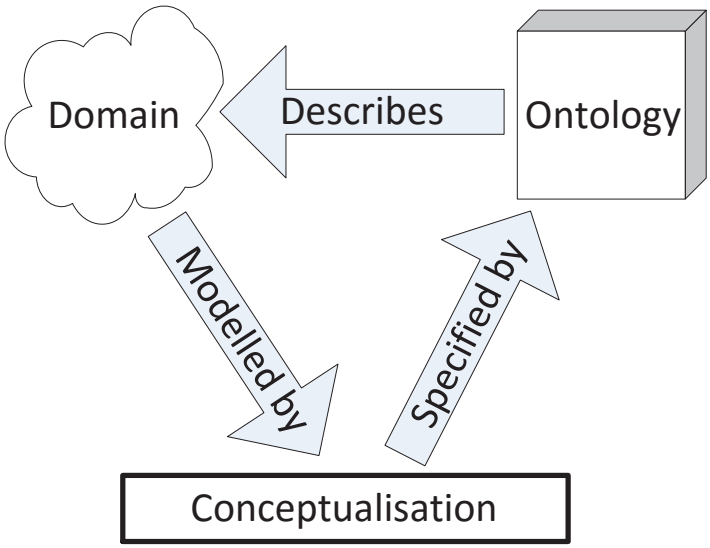


Figure 2.3: Ontology defined for a specific domain

Ontology is explicitly represented in knowledge representation language. Over

the last decades, a lot of ontology definition languages have emerged, including the markup languages introduced in Section 2.1.1, i.e., RDFS, DAML, OIL, DAML+OIL, and OWL, among which OWL is the most expressive one. There are three sub-languages of OWL, i.e., OWL Lite, OWL DL, and OWL Full [76]. Basically, OWL Lite is based on the syntax of RDFS with some additional properties so that it can express conception definitions and axioms. It is the simplest description language in OWL categories. In contrast, OWL DL is a reasoning mechanism that is developed based on DLs, while OWL Full interprets a more complete vocabulary for defining entities in an ontology model. The limitation of OWL Full is that it has no computational guarantees in logic. The advantages and disadvantages of these OWL languages are concluded in Table 2.1 [76]:

Table 2.1: Advantages and disadvantages of OWL sub-languages

Category	Advantages	Disadvantage
OWL Lite	easy to support with software	limited expressiveness
OWL DL	decidability	under certain restrictions
OWL Full	Very expressive	not decidable

In this study, a domain ontology model, i.e., SONT, was built by the Protégé ontology development software [80] and employed for QE in an ODSE. Briefly, the Protégé is an open-source platform, which was developed by Stanford Medical Informatics, providing tools to construct domain models and knowledge-based system with ontologies [80]. According to the definitions in Section 2.1.1, the initial components of SONT are illustrated in Listing 2.2, including a set of xmlns, and URI. The interpretations of each declaration can refer to [81].

Also, SONT is utilised for PSD representation in ontology-based document clustering. Compared with two other OWL sub-languages, OWL DL has been employed for programming SONT and also applied to develop the ontology-based BNs for PTFD, as it offers capability for discovering latent relationships between the concepts of a domain ontology model based on a logical reasoning method.

Listing 2.2: Initial xmlns and URI in SONT

```
<rdf:RDF
  xmlns="http://www.owl-ontologies.com/SONT.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/SONT.owl">
```

Table 2.2: OWL constructors for classes versus DLs concepts

OWL constructor	DLs	Example
owl: THING	$\top$	
owl: NOTHING	$\bot$	
intersectionOf $(C_1...C_n)$	$C_1 \cap ... \cap C_n$	Human $\cap$ Male
unionOf $(C_1...C_n)$	$C_1 \cup ... \cup C_n$	Doctor $\cup$ Lawyer
complementOf (C)	$\neg C$	$\neg$ Male
oneOf $(a_1...a_n)$	$\{a_1...a_n\}$	{John, Mary }
restriction(r allValuesFrom (C))	$\forall$r.C	$\forall$hasChild.Doctor
restriction(r someValuesFrom (C))	$\exists$r.C	$\exists$hasChild.Doctor
restriction(r minCardinality (C))	$\geq$ nr.C	$\geq$ 2 hasChild.Lawyer
restriction(r maxCardinality (C))	$\leq$ nr.C	$\leq$ 2 hasChild.Lawyer
restriction(r value (a))	$\exists r.\{a\}$	$\exists$ citizenOf{ UK }

2.1.3 Description logics

DL is a family of logic based knowledge representation formalism that originated from semantic networks and frame-based systems [82–84]. All the OWL languages mentioned in Section 2.1.2 are inspired by DLs, which are the underlying logical basises, and inference mechanisms of these languages [82].

Basically, a domain can be described by DLs with concepts, roles, individuals and relevant operators [82]. Concepts refer to classes that are interpreted by a set of objects. Roles correspond to relations that are defined as binary relations

Table 2.3: OWL class relationships versus DLs inclusions

OWL axiom	DLs	Example
class(A partial $C_1...C_n$)	$A \subseteq C_1 \cap ...C_n$	Human $\subseteq$ Animal
class(A complete $C_1...C_n$)	$A \equiv C_1 \cap ...C_n$	Man $\equiv$Human$\cap$Male
subClassOf(C_1 C_2)	$C_1 \subseteq C_2$	Human $\subseteq$ Animal $\cap$ Biped
equivalentClass(C_1 C_2)	$C_1 \equiv C_2$	Man $\equiv$ Human $\cup$ Male
disjointWith(C_1 C_2)	$C_1 \subseteq \neg C_2$	Male $\subseteq \neg$ Female
sameAs(a_1 a_2)	$\{a_1\} \equiv \{a_2\}$	PhDStudent $=$ PhDCandidate
differentFrom(a_1 a_2)	$\{a_1\} \equiv \neg \{a_2\}$	PhDStudent $\neq$ MaterStudent

between objects. Individuals are instances of the corresponding concepts. Classes of individuals are described by concepts, which is denoted by linking the super-concepts with any additional restrictions using a set of operators. A hierarchy exists between sub-concepts and super-concepts. The top level of the hierarchy is defined as THING ($\top$) that is a super-concept of any other concepts, and the bottom is NOTHING ($\bot$) that is a sub-concept of all other concepts. A DL-based knowledge base normally contains two components: terminological knowledge base (Tbox) and assertional knowledge base (Abox) [82]. Tbox is composed of concepts and roles defined for a domain and a set of axioms used to assert relationship, e.g., subsumption, equivalence, disjointness, etc., with respect to other classes or properties. Abox is a set of assertions on individuals by using concepts and roles in Tbox. Table 2.2 shows a comparison between OWL constructors for classes and DLs concepts [85]. Table 2.3 presents OWL class relationships and DLs inclusions. The third columns of Table 2.2 and Table 2.3 present the corresponding examples regarding to the constructors of OWL DL, respectively [82]. The employment of these operators is presented in the subsection, and more information can be found in Section 3 and Section 6.

2.1.4 A generic process of building domain ontology models

Since the basics of ontology have been presented, this subsection provides a common process of building a domain ontology model. Before creating an ontology

model, it is essential to be aware of the goal and the scope of the model. In the proposed approaches, the knowledge base concerns power substations only, in which the concepts and their relations are determined by power engineers, followed by the formalisation using Protégé. Typically, SONT is similar to WordNet, which can be regarded as a lexical database, describing the synonyms and hyponyms of concepts and basically using the OWL constructors, e.g., owl:subClassOf for defining concepts hierarchy, owl:disjointWith for distinguishing concepts, owl:sameAs for linking two individuals, etc. In contrast, the ontology-based BNs for PTFD is capable of performing reasoning that apart from the ones in SONT, more logic constructors should be investigated, e.g., owl:complementOf, owl:equivalentClass, owl:unionOf, owl:intersectionOf, etc. To sum up, the general process of building domain ontology models in this thesis is illustrated in Figure 2.4.

2.2 Information Retrival

2.2.1 Historical literature review of information retrieval

The initial purpose of IR is to search useful information from a large collection of information resource according to the demand of a user. The popularisation of the idea of IR was initially introduced by Vannevar Bush in 1945, aiming to establish a goal of fast accessing to the contents of the world's libraries [86]. The term "IR" was coined by Calvin Mooers in 1948 [87] and was started to be widely used since 1950. With the invention of computers, there had been a number of sutdies working on computer searching for information in the mid of 1950s. During that time, the original ideas were to use words as index and employ the overlapping frequencies as the relevance score between the index (query) and a document [88]. Subsequently, a few of IR systems, e.g., Cranfield evaluations [89] and system for the mechanical analysis and retrieval of text (SMART) [33], were built in 1960s. The advent of the recall and precision technology, which is a milestone in IR systems, aims to evaluate retrieval systems that were established in 1967 [89]. Briefly, the effectiveness of an IR system is computed as a recall and precision curve. The details of this part are described detailedly in Section 3.4.3.

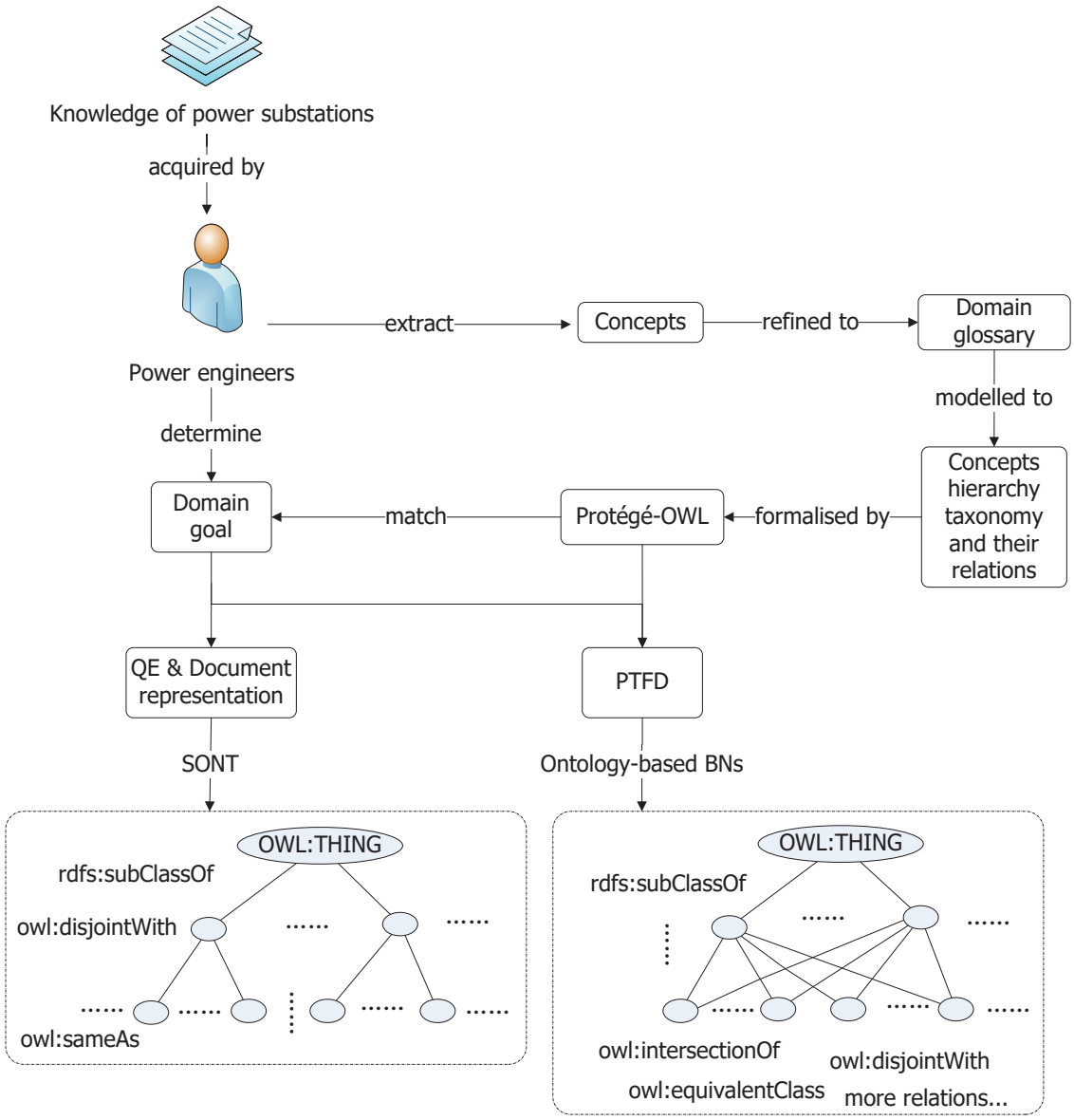


Figure 2.4: A generic process of developing SONT and ontology-based BNs

A series of experiments were carried out on some test collections created in 1960s, even though the largest collection at that time was only 1400 abstracts of aeronautical documents [90]. According to the outcomes of the experiments, it was noted that users often specified very short questions, but the systems could do better searches with longer queries. In this case, with the development of SMART, new retrieval techniques were proposed. Relevance feedback [21], which is one of the effective methods, augments the user's query by adding terms from known relevant documents. In 1970s, the retrieval became to mature into real systems. Owing to the development of the computer typesetting and the word processing, lots of texts were available in machine-readable form. In 1975, VSM was proposed and employed to determine the relevance score between a query input and a specific document repository by Salton [20]. The methodologies, also including single term measure, synonym recognition presented in SMART, paved the way for further developments in the field of IR.

IR techniques were not used for large document repositories until 1992. In the same year, the text retrieval conference (TREC) [25] datasets were established to host a series of workshops that was co-sponsored by the national institute of standards and technology (NIST) and the disruptive technology office of the U.S. department of defense. With the aid of TREC, a number of new methods have been published for IR of large corpus. TREC aims to encourage the research of IR from large text collections, and increase the speed of the lab-to-product transfer of novel IR technologies. A simple IR model for a document repository is illustrated in Figure 2.5.

The idea of applying IR techniques on the internet web pages started from 1994. Many famous search engines or IR systems were developed, including Yahoo [91] and Google [92]. Figure 2.6 illustrates the most important events of the history of IR. Specially, besides text data a web page contains a set of auxiliary information, e.g., HTML, URL, hyperlink, multi-media, etc [93]. The document ranking method of web search engines are not discussed in this thesis, as the web-based information is concerned for document ranking, which is out of the research scope of this study. In the subsection, a set of document ranking techniques of document search engines

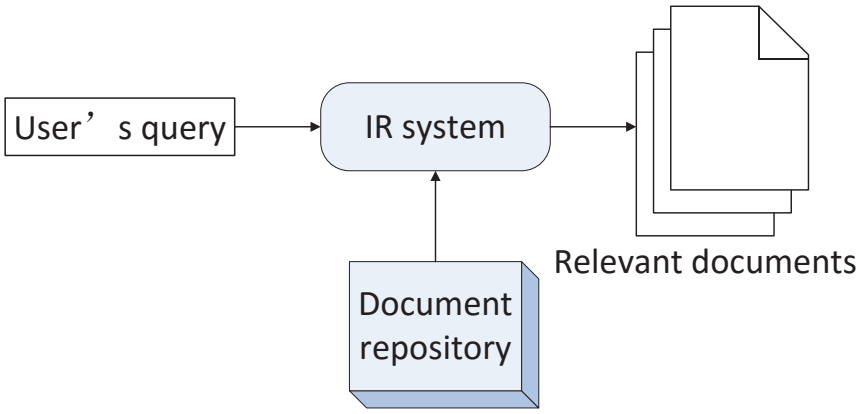


Figure 2.5: The general components of an IR model

is introduced.

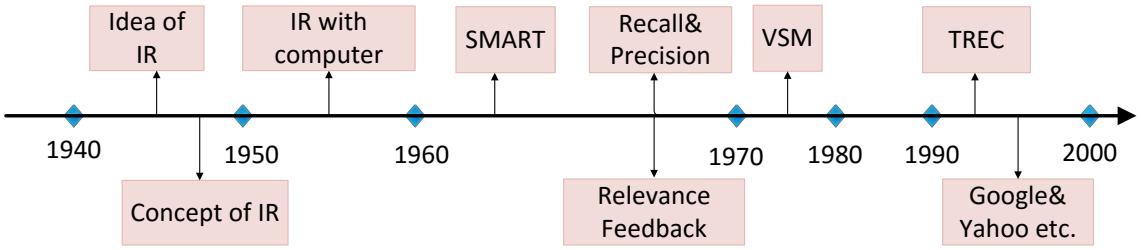


Figure 2.6: Development of modern IR

2.2.2 Mathematical models in information retrieval

In the past decades, a large number of IR models have been developed for document ranking in document search engines. A document search engine aims to discover the information in relation to a user’s input query, and returns a list of relevant documents from a large document repository with a descenting order according to the relevance score. In this subsection, several mathematical models for document ranking are introduced, including Boolean model, probabilistic model, and VSM.

Boolean model

The Boolean model is the first operational mathematical model for IR, proposed in 1973 [94]. It is based on the Boolean operators, i.e., AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). The documents are set of terms, and queries are Boolean expressions composed of terms and operators. A document is regarded to be relevant to a query expression if and only if it satisfies the query expression. Such Boolean model based IR systems are easy to be implemented. However, they have some significant limitations. For instance, there is no way to show the degree of match for a document. In other words, a document can be either satisfied with a query or completely unrelated. The user's information can be retrieved only if the user's query is clear and accurate, otherwise, this model may retrieve nothing related to the requirements. Meanwhile, there is no ranking of retrieved document that makes it difficult for users to navigate their required information.

Probabilistic model

The idea of probabilistic model for IR was firstly proposed in 1960 by Maron and Kuhns [95]. The common applied probabilistic model for IR was subsequently updated by Robertson and Jones in 1976 [96], aiming to examine statistical techniques for exploiting relevance information to weight search terms. After that, several new probabilistic models were proposed e.g., CHI-1, CHI-2, KLD and Okapi BM25 [97], among which Okapi BM25 was the most widely employed ranking formulation in the field of IR. These methods are all based on a general principle that the retrieved documents from a document repository should be ranked in the descending order of their probabilistic relevance regarding to an input query.

One of the representations of the probabilistic models is the non-randomness-based weighting (NRW) model, which was proposed by Chou and Cheng in 2011 [34], and it has been selected as a comparative approach in this research. It is a ranking algorithm for QE based on a term's appearing probability in a single document. The NRW has been proven to be competitive with four other QE weighting functions, including Okapi BM25, CHI-1, CHI-2 and KLD [34]. It is a reasonable comparison algorithm to verify our proposed ER-based ODSE approach.

Basically, the NRW model utilises the concept of probability measurement and the concept of adjustment to develop an expanded query weighting function through the summation of weights. A document repository is defined as $D = \{d_1, d_2, ...d_n\}$, where n is the number of documents in the repository and $d_j \in D$ denotes a single document. Also, q_0 is the initial input query, and a set of expanded queries is represented by $Q = \{q_1, ..., q_v, ..., q_r\}$, where $q_v \in Q$ denotes one of the expanded queries. The weighting function of NRW is defined by equation (2.2.1) and the derived weights are further re-weighted under the Rocchio's framework [98].

$$\omega(q_v) = \left(\sum_{d_j \in R} P_{d_j}(q_v) \cdot \log_2 \frac{P_{d_j}(q_v)}{P_R(q_v)} \cdot \frac{Sim(d_j, q_0)}{\sum\limits_{d_j \in R} Sim(d_i, q_0)} \right) \cdot \frac{\log_2(n/n_{q_v})}{\log_2(n)}, \quad (2.2.1)$$

where $\omega(q_v)$ is the weight assigned to the expanded query term q_v, R is the top-retrieved document set, $P_{d_i}(q_v)$ is the appearance probability of term q_v in d_j, $P_R(q_v)$ is the appearance probability of q_v in the whole document set, $Sim(d_j, q_0)$ the value of the similarity measure between the document d_j and the initial query q_0, and n_{q_v} is the number of documents, in which the term q_v appears.

Vector space model

Briefly, VSM [20] is a mathematical model used to determine the relevance score between a query input and a specific document in an indexed document repository. Following the notation defined in the probabilistic model, in a document ranking process based on VSM, each keyword extracted from a document d_j is stored as a component of the vector that is called term vector. In a simpler way, the similarity measure between a query input and a document is modelled as the distance between a query vector Q and a document vector d. To quantify the vectors, the weight of a query q_v in the document vector is calculated by weighting methods, e.g., the term-frequency (tf) method, tf-idf method, etc. Finally, the total relevance score between the query vector and the document vector is computed with a cosine function [99]. In tf-idf method, it consists of two components, i.e., tf_{d,q_v} and df_{q_v}. tf_{d,q_v} stands for the frequency of a query term q_v occurring in Q for d and df_{q_v} is denoted as the number of documents in the indexed document repository containing

q_v. The query term weight ω_{d,q_v} is obtained by equation (2.2.2).

$$w_{d,q_v} = tf_{d,q_v} \times idf_{q_v} = tf_{d,q_v} \times \log \frac{n}{df_{q_v}}, \tag{2.2.2}$$

where n is still the total number of documents in the whole document repository and idf_{q_v} is defined as the inverse document frequency of q_v. It is shown that if a term q_v appears more frequently in a single document d or the number of documents in the document repository with term q reduces, the value of query term weight ω_{d,q_v} increases. Also, the weight of term q_v in query Q can be also obtained by tf-idf, as shown in equation (2.2.3).

$$w_{Q,q_v} = tf_{Q,q_v} \times idf_{q_v} \tag{2.2.3}$$

Once w_{d,q_v} and w_{Q,q_v} of each query term in *Q* are obtained, the total relevance score between the query vector and the document vector is computed with a cosine function. For instance, $\overrightarrow{V}(d)$ and $\overrightarrow{V}(Q)$ are defined as the vectors of *d* and *Q*, respectively and depicted in Figure 2.7. As indicated in the figure, the angle between the two vectors is α. With the cosine function, the relevance score between $\overrightarrow{V}(d)$ and $\overrightarrow{V}(Q)$, i.e., $Sim_{d,Q}$, is defined in equation (2.2.4) [100].

$$\begin{aligned} Sim_{d,Q} = \cos(\alpha) &= \frac{\overrightarrow{V}(d) \cdot \overrightarrow{V}(Q)}{\mid \overrightarrow{V}(d) \mid\mid \overrightarrow{V}(Q) \mid} \\ &= \sum_{v=1}^{r} RS_v, \end{aligned} \tag{2.2.4}$$

where *r* is the total number of the query terms in *Q* and RS_v $(v = 1, \ldots, r)$ is the relevance score derived between q_v and *d*, and

$$RS_v = \frac{\omega_{d,q_v} \times \omega_{Q,q_v}}{\sqrt{\sum_{v=1}^{r} \omega_{d,q_v}^2} \times \sqrt{\sum_{v=1}^{r} \omega_{Q,q_v}^2}}. \tag{2.2.5}$$

Equation (2.2.4) shows the relevance score between *Q* and *d* is computed as the sum of RS_v $(v = 1, \ldots, r)$, generated between the query terms of the query *Q* and the document *d*. In this study, the relevance scores of the query terms, derived from a domain ontology-based QE process using SONT, are combined with the proposed ER-based approach, as explained detailedly in the next chapter.

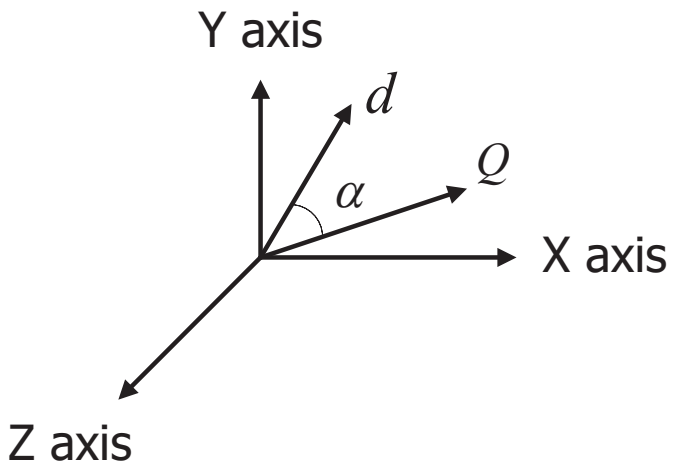


Figure 2.7: VSM model

Meanwhile, VSM is also employed in the document clustering pre-processing. Similarly, each document is represented by a term vector. Compared with VSM in document ranking, the term vector, i.e., $t = \{t_1, t_2, ..., t_m\}$, is composed of the unique terms across all documents in the corpus and denoted by the term frequency, i.e., $tf = \{tf_1, tf_2, ..., tf_m\}$ [32]. Thus, a document repository can be represented by a document-term matrix or term-document matrix. Subsequently, the similarity between each pair-wise term can be examined by a reasonable distance measure, resulting in a set of clusters, in which documents within the same cluster have smaller distance than documents in the other clusters.

The procedure of "document to term vector" is achieved based on the Apache Lucene [101] search engine library that is presented in detail in Section 3.3. Also, it should be noticed that "document to term vector" is not simply decomposing the documents to the words with the original form in the document. More related processing techniques are involved, e.g., "stop-word removal", "stemming", etc., which are also illustrated in Section 3.3.

2.3 Optimisation Techniques concerned in this Thesis

Optimisation is regarded as a mathematical discipline, aiming to find the maximum or minimum value for a given objective function with respect to all

involved parameters and given constraints [102]. It has been widely used in many research areas, e.g., robotic, medicine, economic, etc. In these fields, a targeted problem is always expressed as an optimisation problem seeking a minimum or maximum. There are large numbers of classical optimisation methods proposed to solve such issues in a finished time period, e.g., the linear programming for a linear function of decision variables [103], the quadratic programming and the Newton method [102], the dynamic programming [104], the Simplex method [105], the gradient method [106], etc. However, for these optimisation methods, the objective functions are limited by some characteristics, for example, convexity, continuity or the derivability [107]. If the objective function is non-homogeneous or cannot be expressed analytically according to the parameters, the classical optimisation methods become obsolete [107]. Heuristic technique is a great revolution in the optimisation field and can be utilised to complex problems with no limitations on the form of the objective function. Also, a reasonable computational time is allowed for the process to find the targeted value, but the optimality of the solution cannot be guaranteed. In other words, a heuristic returns an approximate solution that is worse than the optimum, but the solution is good enough in a reasonable amount of time. Many heuristic algorithms are very specific and problem-dependent. In most cases, they are developed for specific problems [108]. Meta-heuristics, which belong to stochastic optimisation algorithms, aiming to deal with general optimisation problems. It is defined as 'a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimisation algorithms' by Sörensen [108]. As mentioned in Section 2.1.1, meta-data means data about data. Similarly, "meta" in meta-heuristics means "one level above". In other words, meta-heuristics work with support to heuristics.

Meta-heuristics can be used to handle complex optimisation problems in different areas. Here, the complex problems mean that they cannot be optimised to a guaranteed bound using any exact method within a reasonably computational time limit [109]. Meta-heuristics usually contain an iterative procedure, which repeats during the optimisation until they reach a specified stopping criterion. Almost all the meta-heuristics are inspired by the nature world, e.g., physics, biology, ethology,

etc, involving random variables, and consisting of several parameters that need to be initialised for a random searching.

The two meta-heuristic, i.e., the simulated annealing algorithm (SA) and the genetic algorithms (GAs) are implemented to optimise the consensus functions of WPK-CC and INT-CC, respectively. In the subsections, the SA and the GA are presented in detail. Meanwhile, the non-negative matrix factorisation method, which is employed in NNMF-CC, belongs to the a non-linear optimisation problem and it is introduced in Section 4.2.1.

2.3.1 Simulated annealing algorithm

The idea of the SA, was inspired from the physical process of annealing technique that was proposed by Metropolis in 1953 [110]. Basically, the process consists of two steps: 1. heat a material to a high temperature; 2. cool down slowly in order to enhance its crystals' size. In the first step, the high temperature of the material reflects the atoms in the material have high energies, therefore, they start to change positions and perform large random movements in the material. In the second step, the energies of the atoms and their movement capacities reduce during the slow cooling process until to an equilibrium state. Kirkpatric and Cerny applied this idea to the optimisation field based on Metropolis algorithm in 1983 [111] [112].

Briefly, the objective function in an optimisation problem is similar to the energy $\mathcal{E}$ of a material. The algorithm starts by an initialised random or constructed solution $\mathcal{S}$ and a fictitious temperature T. At each iterative steps, a new solution $\mathcal{S}'$ is randomly selected in the neighbourhood of the current solution $\mathcal{S}$, i.e., neighbour($\mathcal{S}$). The condition of that solution $\mathcal{S}'$ can be selected as a new solution, depends on T and comparisons between $\mathcal{S}$ and $\mathcal{S}'$ regarding to $\mathcal{E}$, i.e., $\mathcal{E}(\mathcal{S})$ and $\mathcal{E}(\mathcal{S}')$, respectively. If $\mathcal{E}(\mathcal{S}') \leq \mathcal{E}(\mathcal{S})$, S is replaced by $\mathcal{S}'$ and $\mathcal{S}'$ becomes to a new solution. However, in the SA, if $\mathcal{E}(\mathcal{S}') > E(\mathcal{S})$, $\mathcal{S}'$ also has chances to be accepted based on a probability, i.e., $\text{prob}\,(T, \mathcal{E}(\mathcal{S}'), \mathcal{E}(\mathcal{S})) = \exp\left(-\frac{E(\mathcal{S}')-E(\mathcal{S})}{\beta T}\right)$. Here, β is the Boltzmann constant, which is related to the energy at the individual atom level with temperature, and $0 < \beta < 1$. The temperature T decreases during the iterations. Therefore, the search begins with a high probability to accept the deteriorating moves, and this

ability gradually decreases along with time.

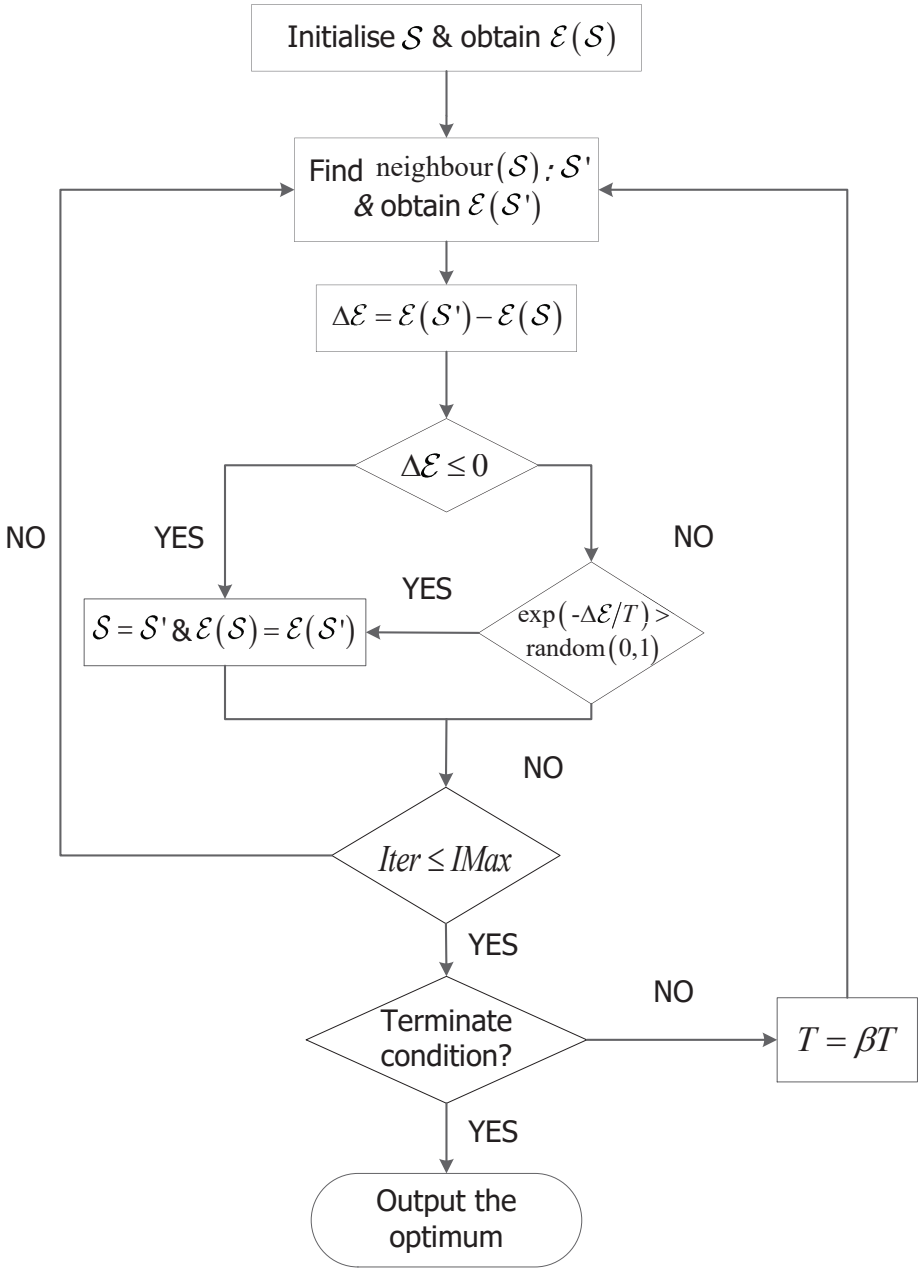


Figure 2.8: The working process of the SA

The SA is a special kind of greedy algorithm, e.g., hill climbing algorithm, with stochastic factors [113]. Hill climbing algorithm begins with an arbitrary solution. Then, it replicates many times in order to search for better solutions incrementally

until no further improvements can be found for the solution. It is a simple and popular way for searching a local optimum, however, the solution of hill climbing algorithm may not be a global optimum. Compared to hill climbing algorithm, the SA is a good way to avoid local optimum. Even if the new state is worse than previous state, it still has a chance to be accepted depending on the temperature. Therefore, after several iterations, it may get rid of a local optimum and obtain a globe optimum. In addition, according to Metropolis algorithm, the rate of cooling is controlled by β. If a large β is set, the probability to find a globe optimum increases, but the searching process becomes time-consuming. On the contrary, if a small β is selected, the time of searching process decreases, but it always returns a local optimum. The working process is shown in Figure 2.8. "$Iter$" is the iterative steps and "$IMax$" is pre-defined maximum iterations. Terminate condition is also a user pre-defined status. In this case, if $\mathcal{E}$ becomes to zero, the SA stops.

2.3.2 Genetic algorithms

Genetic algorithms are another well-known stochastic meta-heuristics, which belong to the evolutionary algorithm family. The GA was firstly established by Holland in 1975 that is inspired from Darwin theory, i.e., the natural evolution [114]. For living creatures, natural selection and reproduction are the basic mechanisms. In other words, the most adapted individuals in a large population will survive with the environment, and other individuals, which are not able to adapt to the environment will be obsolete. Subsequently, the selected individuals will reproduce their offsprings based on some genetic operators, i.e., crossovers, mutations, etc.

The procedure in the natural evolution is similarly to a meta-heuristics optimisation, in which a number of solutions can be replaced by other more fitted solutions until the approximate optimal solution occurs [114]. In other words, the old population with a number of individuals, through pre-defined iterations, will be replaced by a new population with the same number of individuals, which have better fitness. The fitness in the GAs is regarded as a fitness function referring to the objective function in optimisation. The individuals in the GAs are called chromosomes, of which the most common representation is a fixed-length binary

string (e.g., 111111000000). In this case, an initial set of solution should be encoded to a set of binary string based on defined rules, forming an "u" initial population. Subsequently, a selection procedure operates, in which fitnesses will be calculated for each chromosome. Basically, the chromosomes with better fitnesses have more chances or higher probabilities to be selected for reproduction, which are based on a random value, i.e, P_{s}, and $0 < P_{\mathrm{s}} < 1$. The selection procedure is followed by two genetic operators, i.e., crossover and mutation. The crossover and mutation operations can be achieved by the bits swap or transform, respectively. Both crossover and mutation procedures are determined by pre-defined probabilities, i.e., crossover rate (P_{c}) and mutation rate (P_{m}), respectively, where $0 < P_{\mathrm{c}} < 0$ and $0 < P_{\mathrm{m}} < 1$. If a random number is not satisfied with P_{c} or P_{m}, the relevant chromosomes will be directly copied and allocated into the new population. Also, the position of crossover and mutation in a selected chromosome or binary string are both randomly chosen. Crossover ensures the algorithm has a wide range of searching. Mutation means that any one gene or several genes in a chromosome may changes. It aims to keep the variety of population, ensure the ability of global searching and avoid pre-mature. Finally, the obtained optimal chromosome should be decoded to the original form of solution. The generic working process of the GA is illustrated in Figure 2.9.

In this study, a GA is implemented to optimise the consensus function of INT-CC. A clustering result can be represented by an integer vector, where the length of the vector shows the number of objects in the dataset, the values represent the corresponding cluster label and the maximum integer stands for the number of cluster. Thus, the representation of chromosomes in a GA in terms of cluster analysis is based on integer vector, and this principle of coding method is detailedly demonstrated in Section 4.1.3. The following subsections illustrate each genetic operator of GAs.

Selection mechanisms

Selection operates after a new population generated. The basic rule of selection mechanism is that chromosomes with better fitnesses have more chances or higher

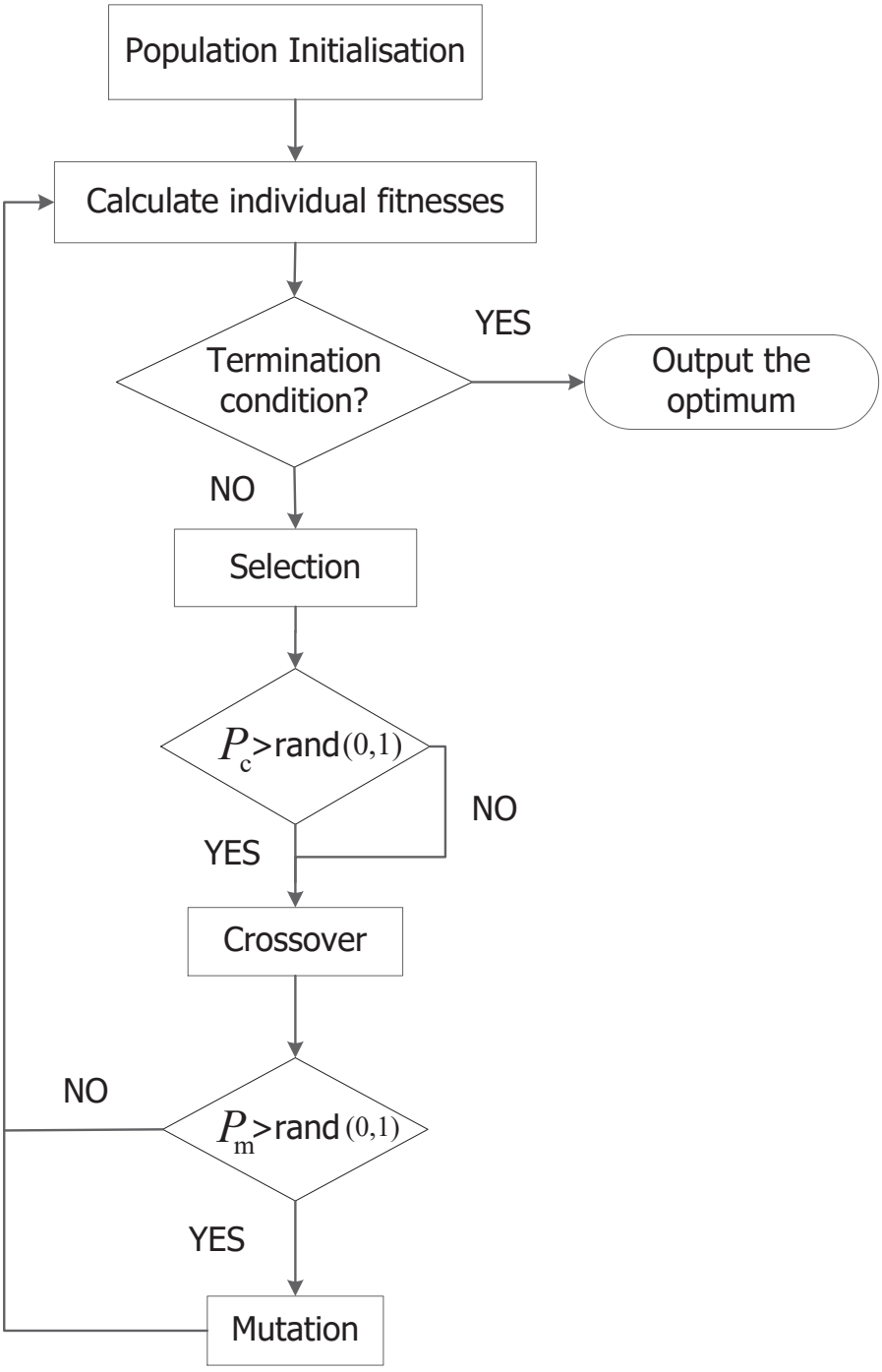


Figure 2.9: The working process of the GA

probabilities to be selected to produce offsprings.

1. Roulette wheel selection (RW)

   It works as a roulette wheel, and chromosomes are selected according to their fitnesses. The better fitness of a single chromosome, the more chances it can be selected. Figure 2.11 illustrates an example for RW. Suppose that there are four chromosomes in a population with different fitnesses, i.e., 2, 4, 6 and 8. Firstly, the sum of all chromosome fitnesses in population is calculated, i.e., 20 and each chromosome is allocated to a section in an imaginary roulette wheel according to their fitness proportions. The wheel rotates as many times as necessary to select the full set of parents for the next generation. As it is shown in Figure 2.11, chromosome 4 has a higher probability (e.g., 40%) than other individuals to be selected.

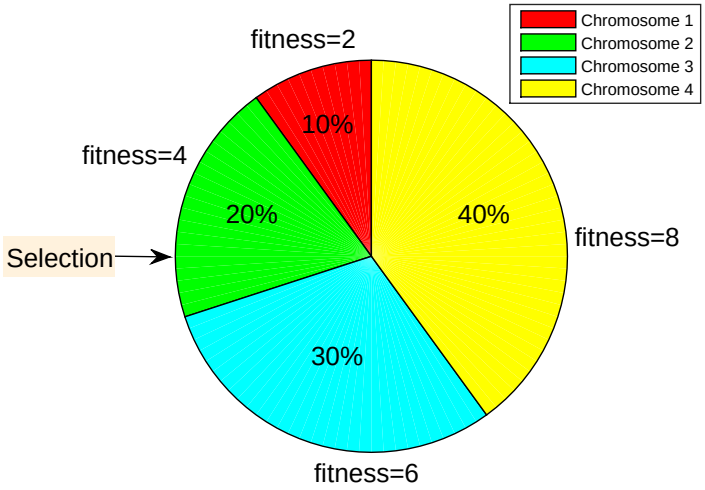


Figure 2.10: An illustration of RW

2. Baker's linear ranking selection (LR)

   The fitness may vary dramatically. If the best chromosome has a much better fitness, other chromosomes will have very few chances to be chosen [60]. Therefore, the LR is implemented to overcome the limitation as mentioned

above. It firstly ranks the population and then every chromosome receives fitness from the ranking as shown in equation (2.3.1):

$$p_i = \frac{1}{N}\left(\eta_{\max} - (\eta_{\max} - \eta_{\min}) \cdot \frac{i-1}{N-1}\right), \tag{2.3.1}$$

where, i stands for the ranked index (1 is the best). $\eta_{\max}$ and $\eta_{\min}$ are the expected fitness values of the best and worst members, respectively. The constraints are $\sum p_i = 1$, $\eta_{\min} = 2 - \eta_{\max}$ and $\eta_{\max} \in [1, 2]$. Figure 2.11 illustrates an example of the LR. The fitness of chromosome 4 is 50, i.e., 89% (probability of selection), which is much larger than other chromosomes. In this case, other chromosome has very few chances to be chosen, i.e., 11% in total. Suppose $\eta_{\max}$ is 1.5. After performing ranking, each chromosome has been re-assigned to a new section in the wheel so that every chromosome has a chance to be selected. As a consequence, ranking keeps the diversity of the population.

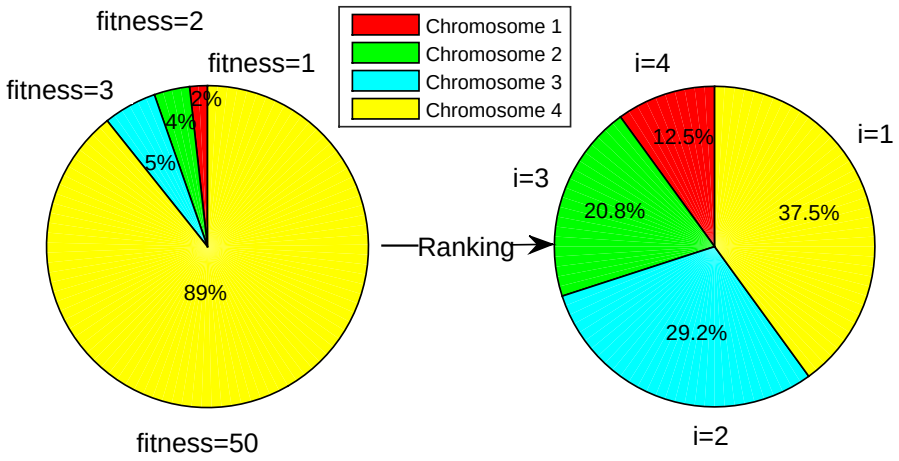


Figure 2.11: An illustration of LR

3. Elitism selection

   As both the crossover and the mutation process in a random way, some of the best chromosomes may lose in the offspring. The idea of elitism is to copy

the chromosome(s) with the best fitness(es) to the next generation directly without the crossover and mutation operations.

4. Elitism combined with RW or LR selection
   Elitism is always followed by other selection mechanisms that are applied to accomplish the selection from the rest chromosomes. In this research, the elitism scheme combined with RW and LR schemes are evaluated. The best 2% chromosomes are directed copied to the next generation, and the rest chromosomes are selected based on Elitism with RW (ERW) or elitism with LR (ELR). Also, the ERW was implemented in INT-CC as presented in Section 4.4

Crossover mechanisms

1. One-point crossover
   This is the most common crossover scheme, in which the integer string from the beginning of the chromosome to a randomly selected crossover point is copied from one parent chromosome, and the rest is copied from the second chromosome.

2. Two-point crossover
   There are two crossover points randomly selected, and each offspring acquires a portion of genes from its parent chromosomes in turns according to the crossover point. The following example shows the GA with two-point crossover for clustering issue:

Parent1	1112|221233|33
Parent2	1111|222213|23
Offspring1	1112|222213|33
Offspring2	1111|221233|23

3. Uniform crossover
   Apart from the above crossover mechanisms, there exists multi-point crossover

and an extreme case of multi-point crossover is defined as uniform crossover, where alleles are selected from either parent chromosomes with a probability of 50% or 0.5 [115]. In this case, a set of random variables $Ma = \{p_1, p_2, ..., p_n\}$ is generated, where n is the number of alleles or documents, $p_i \in Ma$ and $0 < p_i < 1$. According to Ma, a mask is obtained, in which it only contains 0 or 1 and the size of the mask is the same as Ma. The element of the mask equals to 1 when its corresponding located p_i is larger than 0.5, otherwise it is 0. The offspring is obtained based on the mask. Considering the above example, the process of uniform crossover can be concluded as following:

Mask	100110100111
Parent1	111222123333
Parent2	111122221323

Offspring1	111222121333
Offspring2	111122223323

4. Binomial crossover

   Every allele is chosen from the first parent with a probability of P, otherwise it is selected from the second parent with a probability of $(1 - P)$. In this case, uniform crossover mentioned before can be regarded as a special type of Binomial crossover, as $P = 0.5$.

Mutation mechanisms

1. Bit-flip mutation

   It is the most common mutation for the GA, in which mutation occurs depending on a pre-defined mutation rate P_m. The mutation point is also randomly selected.

2. Adaptive Mutation

   To monitor the population diversity and avoid premature convergence, the

adaptive mutation is applied in our research. Basically, high mutation rates are used for similar parents mate, whereas low mutation rates are set to dissimilar parents mate. The adaptive mutation rate is illustrated by equation (2.3.2) [116].

$$p_m\left(p_i, p_j\right) = p_{m_L} + \left(p_{m_U} - p_{m_L}\right) \cdot \left[\frac{I\left(p_i, p_j\right)}{len}\right]^2; \tag{2.3.2}$$

where $p_{m_L} = 0.05$ and $p_{m_U} = 0.2$ are lower and upper mutation bounds in this research, respectively. $I(\cdot)$ denotes the number of identical genes between two parents, and len is the length of each parent.

3. Mutation for document clustering

   It is a common way that the GA follows the binary coding. Therefore, if mutation occurs, one gene will change from "0" to "1" or from "1" to "0". However, in a clustering result, the chromosome is a vector of integer e.g., "11122132144455" (5 clusters). The selected mutation point can be mutated to any cluster number. In this research, if a mutation point is chosen, all alleles in the population will be analysed. In general, if a cluster number occurs more frequently than others, it will has a higher probability that other cluster number could be mutated to. If mutation occurs at the second gene, i.e., "**1**", of a chromosome "1**1**1222233344" (the length denotes the number of documents is 12) and a vector of its alleles is "111223" (the length denotes the *PopSize* is 6), the potential mutated gene can be "2", "3" or "4". Without considering the original gene "1", the probabilities to be mutated to "2", "3" and "4" are $2/3$, $1/3$, and 0, respectively. In order to consider every possible label, the LR is again used. Thus, the probabilities are changed to $1/2$, $1/3$, $1/6$, respectively.

2.4 Bayesian Networks

2.4.1 Basics of Bayesian networks

Probability theory is one of the most powerful approaches to capture the degree of belief about uncertain knowledge. A statement with a probability $P = 0$ means a belief that such statement is false, $P = 1$ shows the belief that the statement is true, and $P \in (0, 1)$ represents the degree of belief in the truth of such statement. The following parts illustrate four fundamental concepts of BNs, i.e., conditional probability, joint probability, marginal probability and Bayes's theorem.

- Conditional probability: it also refers to posterior probability, which can be explained by the probability of event A occurs given a condition that event B has occurred, and it is denoted by $P(A|B)$.
- Joint probability: given at least two events, e.g., A and B, it illustrates the probability that both A and B occur, which is represented by $P(A \cap B)$ or $P(A, B)$.
- Marginal probability: it is also called prior probability, simply means the probability that one event, e.g., B occurs, and it is expressed as $P(B)$. The relationship among the above three concepts is demonstrated in equation (2.4.1).

$$P\left(A\,|B\right) = \frac{P\left(A, B\right)}{P\left(B\right)}. \tag{2.4.1}$$

- Bayes's theorem: equation (2.4.1) concerns the conditional probability of A given B. Similarly, the conditional probability of B given A is shown in equation (2.4.2).

$$P\left(B\,|A\right) = \frac{P\left(A, B\right)}{P\left(A\right)}. \tag{2.4.2}$$

Combining equation (2.4.1) and equation (2.4.2):

$$P\left(A\,|B\right)P(B) = P(A, B) = P\left(B\,|A\right)P(A). \tag{2.4.3}$$

Therefore, from equation (2.4.3), the Bayes's theorem is given by equation (2.4.4):

$$P\left(A\,|B\right) = \frac{P\left(B\,|A\right)P\left(A\right)}{P\left(B\right)}. \tag{2.4.4}$$

$P(A|B)$ and $P(A)$ also refer to the posterior probability of a hypothesis A conditioned on some evidence B and the prior probability of A, respectively. Bayes's theorem allows unknown probabilities to be calculated from known cases.

2.4.2 Bayesian networks

BNs also refer to Bayesian belief networks or belief networks, which represents the dependence between variables and decomposes the JPD into a set of CPTs associated with individual variables. In that case, a BN of n variables consists of a directed acyclic graph of n nodes and a number of arcs. Figure 2.12 shows a BN with five binary variables (each variable has only two statuses, i.e., true or false). Commonly, BNs can be regarded as a representation of JPD [13]. The JPD

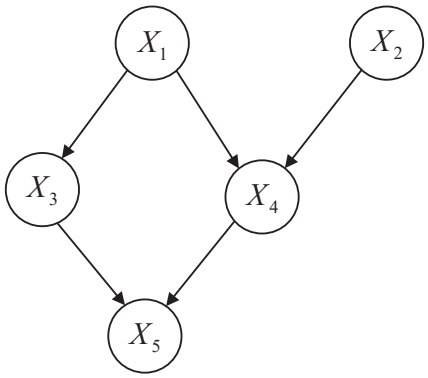


Figure 2.12: A five-node BN

of all variables involved can be used to calculate the answer to any probabilistic queries about the modelling domain. The JPD of the above BN is represented by $P(X_1 = x_1, X_2 = x_2, ..., X_5 = x_5)$, where $x = \{x_1, x_2, ..., x_5\}$ stands for a joint assignment or an instantiation to the set of all variables X. In a simpler notation, the joint probability can be factorised by equation (2.4.5).

$$P(x_1, x_2, ..., x_5) = P(x_1) \times P(x_2|x_1) \times P(x_3|x_1, x_2) \times P(x_4|x_1, x_2, x_3) \times P(x_5|x_1, x_2, x_3, x_4) \quad . \tag{2.4.5}$$

The multiplication refers to the chain rule of probability theory. In addition, the value of a particular node is conditional only on the values of its parent nodes [56]. It is noted that X_1 and X_2 are absolutely independent; X_3 is independent of X_2 and X_4 given X_1; X_4 is independent of X_3 given X_1 and X_2; X_5 is independent of X_1 and X_2 given X_3 and X_4. Thus, equation (2.4.5) is simplified to equation (2.4.6).

$$\begin{aligned} P(x_1, x_2, ..., x_5) = P(x_1) \times P(x_2) \times P(x_3 | x_1) \times \\ P(x_4 | x_1, x_2) \times P(x_5 | x_3, x_4) \end{aligned} . \tag{2.4.6}$$

Assume the five-node BN is parts of a large BN with n nodes, i.e., $X = \{X_1, X_2, ..., X_n\}$ ($n \gg 5$), the above two equations can be generalised to equation (2.4.7) and equation (2.4.8):

$$\begin{aligned} P(x_1, x_2, ..., x_n) &= P(x_1) \times ... \times P(x_n | x_1, ..., x_{n-1}) \\ &= \prod_{i=1}^{n} P(x_i | x_1, ..., x_{n-1}) \end{aligned} . \tag{2.4.7}$$

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i | \text{Parents}(X_i)). \tag{2.4.8}$$

2.4.3 Conditional probability tables

If the number of the variable in a domain increases sharply, the JPD becomes intractably large. As shown in Figure 2.12, JPD with five nodes has $2^5 = 32$ probabilities. In this case, BNs provide a more compact representation, i.e., CPTs are attached to each node, than simply describing the probability of every joint instantiation of all variables. CPTs are used to quantify the strength of the relationship between variables. Once the structure of a BN is confirmed, the CPTs for each node should be specified. A CPT describes the conditional probability of each node value for each possible combination of values of its parent nodes. The sum of each row of the CPT must be 1. The illustration of a BN with its CPTs is shown in Section 6.2.

2.4.4 Probabilistic inference

Probabilistic inference with BNs aims to compute the posterior probability distributions. Before utilising a BN to do inference, the structure of a BN and the

CPTs are specified. Normally, the structure of a BN, especially in a PTFD, the variable nodes and the relationships of the nodes are determined by the experts of power system. The CPTs are acquired by analysing the historical data.

A typical example of a BN approach for PTFD is the dissolved gas analysis presented in [13]. The main purpose of dissolved gas analysis in PTFD is to identify the fault types of the unit based on dissolved gas ratios. In [13], the BN is constructed by the fault type level, including thermal faults and discharge faults, and the three types of gas ratio on the symptom level. The typical combination of gas ratios reflects the corresponding fault types. Subsequently, the CPTs of each variable nodes are obtained by statistically analysing 40 sets of dissolved gas data. The decision of a fault type of a given symptom is made by the probabilistic inference. For instance, if a thermal fault has three states, i.e., $[\text{Normal}, \text{Low temperature overheating}, \text{High temperature overheating}]$ and the derived probability of the node thermal fault is $[0.02, 0.04, 0.94]$, the inference result indicates that this power transformer has a high probability under the fault of high temperature overheating.

According to BNs approach for dissolved gas analysis, the generic PTFD based on BNs is illustrated by Figure 2.13. The procedures are described as follows:

1. Design a graphical model based on the transformer fault types and fault symptoms;
2. Collect a group of historical data containing the relevant faults and symptoms;
3. Determine the CPTs for each node in the BN by statistical analysis;
4. Update the CPTs of the BN with new fault cases and the status of each fault type (occurs or not occurs) is indicated by a high probability;
5. Provide relevant repair suggestion to the observed fault type.

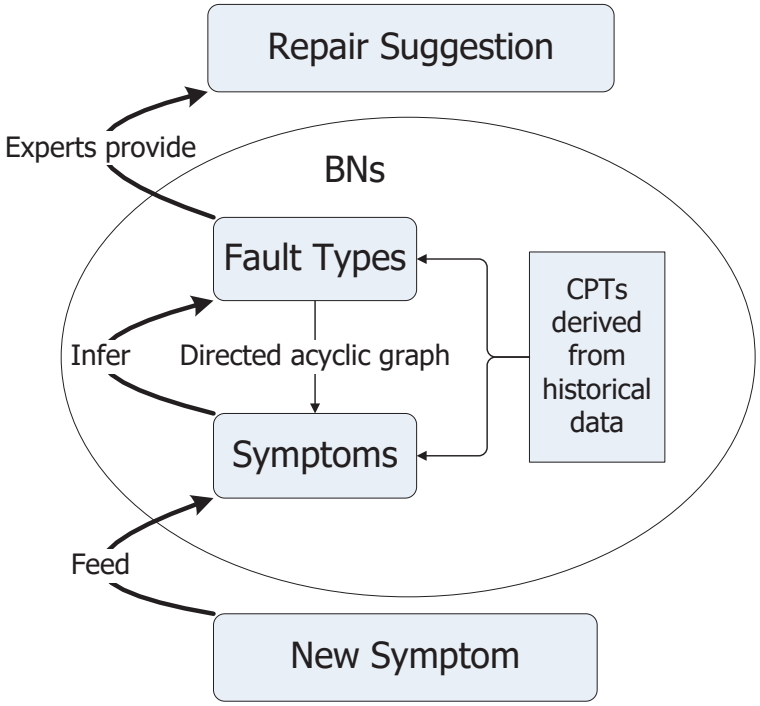


Figure 2.13: BNs approaches for PTFD

2.5 Summary

The main purpose of this chapter is providing the relevant background knowledge and literature review of proposed approaches. As the three intelligence approaches in this thesis are based on the ontology techniques, this chapter firstly presented an overview of ontology and a brief introduction of its applications concerned in the corresponding aspects. In addition, the two approaches, i.e., ER-based ODSE and CC for PSD, are members of the field of IR. Therefore, it would be helpful to provide an introduction of IR in this early chapter. This chapter started from a brief introduction of the Semantic Web, ontology languages, and DLs, followed by roughly sketching the implementation of ontology in this research. The general process of designing SONT and the framework of ontology-based BNs were demonstrated. Subsequently, a historical literature review of IR was presented, including the main purpose and evolution of IR. Afterwards, a set of mathematical models in IR was illustrated. One of the comparatives of ER-based ODSE, which is a probabilistic model-based approach, i.e., NRW, was described mathematically. In addition, the VSM with tf-idf weighting was highlighted, as it is one of the primary tasks to process PSD. Meanwhile, as the CC algorithms in this thesis are based on the median partitions approach, which are the basis of the optimisation problem, the optimisation techniques involved were reviewed. Finally, the introduction of the basics of BNs, followed by a traditional BN inference application in PTFD was given.

Chapter 3

Ontology-based Document Search Engine Using Evidential Reasoning

This chapter presents a novel approach to document ranking in an ODSE using ER. The structure of this chapter is organised as follows. Section 3.1 presents the details of developing SONT, which is followed by an illustration of utilising a MADM tree model to organise expanded query terms. Section 3.2 demonstrates the implementation of ER algorithm to perform decision making in order to meet the fulfilment of document ranking. An ER algorithm, based on the DS theory, which is employed for evidence combination in the MADM tree model, is demonstrated. Then, the basic probability assignment of the MADM model is integrated by ER. Subsequently, the relative weights of expanded queries (attributes in MADM) are assigned by analytic hierarchy process (AHP) method. In Section 3.3, four document search engines, i.e., SE_1, SE_2, SE_3, and SE_4 are designed for comparison purpose, where the working processes of SE_1, SE_2, and SE_3 are introduced. SE_4 is the NRW approach that has been introduced in Section 2.2.2. Finally, a number of simulation studies, which are carried out on the PSD, are illustrated in Section 3.4. The results show that the proposed approach, i.e., SE_3, provides an advantageous solution to document ranking, and the precision at the same recall levels for ODSE searches are improved significantly with ER-embedded, in comparison with a traditional keyword-matching search engine (SE_1),

an ODSE without ER (SE_4), and the NRW approach.

3.1 Introduction

As the ontology model for QE in IR systems was introduced in the last chapter, the procedure to achieve the ER-based ODSE should be illustrated. The first component of ER-based ODSE, i.e., using a MADM tree model to organise the expanded queries, is specified in this subsection.

3.1.1 Query expansion by an ontology model of power substations

Following the process of developing a domain ontology model illustrated in Figure 2.4, this subsection introduces the construction of SONT in detail, which is based on [2] [9]. Considering the domain scope of SONT, "Power System" is defined as "Thing" at the top ontology level, and a top-down development process is utilised to define the corresponding classes and the class hierarchies in Protégé. The second level classes consist of nearly all important aspects of power substations, including "Action", "Attributes", "Device", "Other assets", "Units", and "Status". Figure 3.1 shows a screenshot of classes and hierarchies defined in SONT from Protégé editor. For instance, for the class "Action", it contains four subclasses, namely "Monitoring", "Restoration", "Protection", and "Vibration". The following level is composed of "Fault diagnosis", "Distortion", etc, followed by "Fault detection", "Fault isolation", etc. Meanwhile, if an open source graph visualisation software, i.e., Graphviz, is built in Protégé, SONT can be graphically depicted [117]. Due to the space limitation, only the class "Action" is shown in Figure 3.2.

Currently, SONT has 413 classes (or concepts), 67 properties, and 31,579 individuals. Briefly, as one of the components of ontology, a class is a general concept of a set of individuals. For instance, "Transformer" can be defined as a class, then "'Voltage Transformer" is a subclass of "Transformer". The

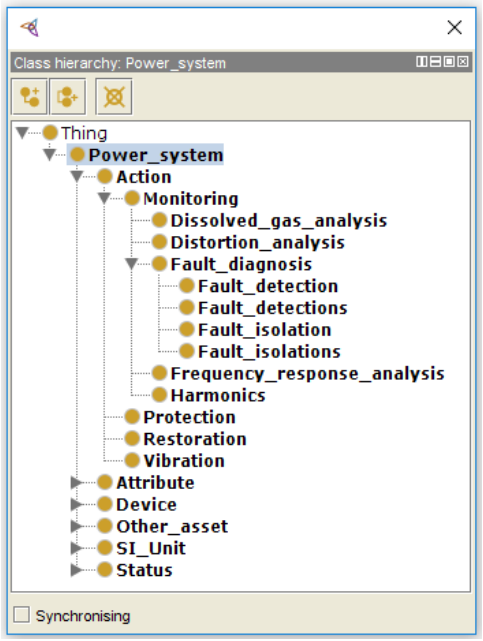


Figure 3.1: Classes and hierarchies of SONT defined in Protégé editor

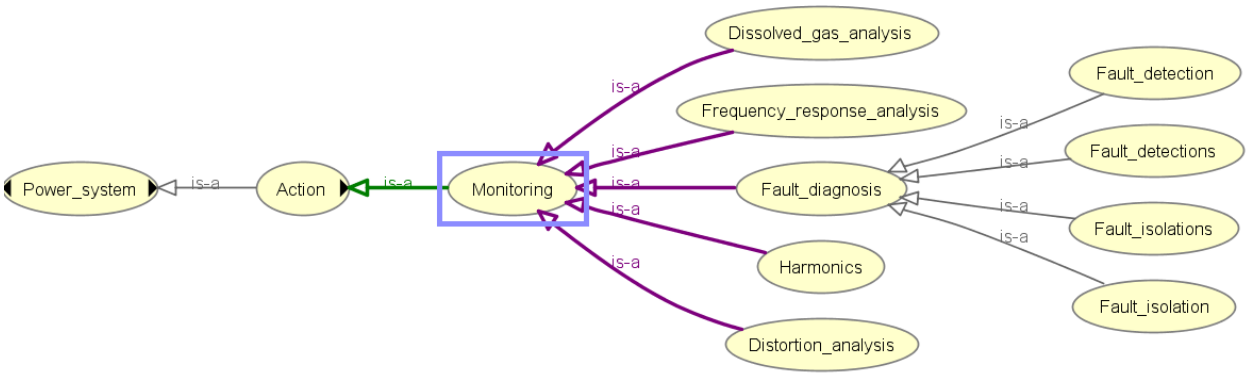


Figure 3.2: The class “Action” and its hierarchies

relationship "subclass" is a property. In the class of "Voltage Transformer", a number of individuals can be identified, such as "Voltage Transformer A", "Voltage Transformer B", etc.

In this study, a given query input can be extended with its synonyms and hyponyms only. It is noted that the classes in the fifth level of SONT in Figure 3.1 are hyponyms of their superclass "Fault diagnosis". Also, there are seven individuals defined for "Fault diagnosis" as shown in Figure 3.3 that form the synonym set of "Fault diagnosis", as they are equivalent power substation-related concepts.

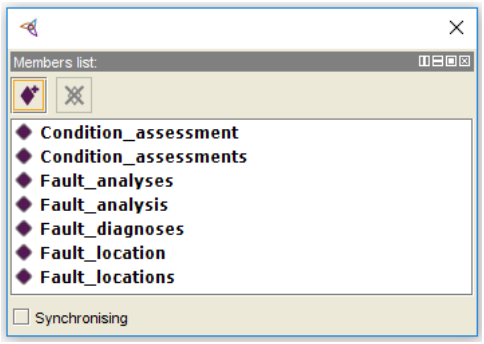


Figure 3.3: Individuals defined for "Fault diagnosis"

Following the above definition, a synonym set and a hyponym set for a query input "*Fault diagnosis*" can be derived as (3.1.1) and (3.1.2), respectively. Meanwhile, parts of the relationships among the above terms can be described by an OWL file as shown in Listing 3.1 without considering the URI and xmlns.

$$\{Fault\ diagnoses,\ Condition\ assessment,\ Condition\ assessments,\ Fault\ location,\ Fault\ locations,\ Fault\ analysis,\ Fault\ analyses\} \quad (3.1.1)$$

$$\{Fault\ detection,\ Fault\ detections,\ Fault\ isolation,\ Fault\ isolations\} \quad (3.1.2)$$

It is noted that the synonym set and the hyponym set defined in SONT contain both the singular and the plural of a term, e.g., "$Condition\ assessment$" and "$Condition\ assessments$", as the two forms of the term are regarded as two independent words in a document search process with the implementation of a document search engine. Thus, when a query term is not expanded with its singular

or plural, a number of documents, which do not contain the correlative form, cannot be retrieved during a subsequent document search process. As a result, the number of relevant documents retrieved by a search engine could be reduced.

Listing 3.1: OWL file for defining the synonym set and the hyponym set

```
<owl:Class rdf:ID="Fault_diagnosis">
    <owl:sameAs rdf:resource="#Condition_assessment"/>
      ......
</owl:Class>
......
<owl:Class rdf:ID="#Fault_isolation">
    <rdfs:subClassOf rdf:resource="#Fault_diagnosis"/>
     ......
    <owl:disjointWith rdf:resource="#Fault_detection"/>
     ......
</owl:Class>
```

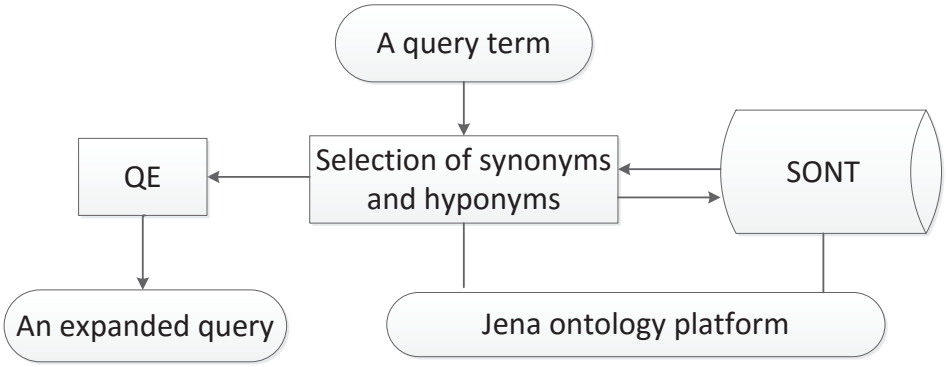


Figure 3.4: A QE process with SONT

A QE process with SONT is illustrated in Figure 3.4. A Jena [118] ontology platform, which is capable of developing semantic web applications, consisting of RDF API, OWL API, and a rule-based inference engine, is utilised for accessing SONT. In the QE process regarding the context of power substations, the advantage of using SONT, compared with employing a textual thesaurus is that a query term matches an ontology concept name of SONT concerning the domain of power

substations. Consequently, the meaning of the term can be restricted within the domain of substations and effectively disambiguated before a QE process, which is not achievable by published textual thesauri. Thus, a high accuracy of a QE process can be achieved by SONT. For instance, a QE process of "$Condition\ assessment$" is also implemented with WordNet, in which none of the synonyms or the hyponyms can be mapped for "$Condition\ assessment$". As a consequence, QE with SONT, compared to QE using WordNet, a higher search accuracy is possibly achieved for a document searching process with SONT.

3.1.2 Using the multiple attribute decision making tree model to present expanded queries

Figure 3.5 illustrates a generalised MADM tree model, consisting of a set of nodes with a hierarchical structure. Also, there are two different levels in the MADM tree model, i.e., attribute level and factor level. An evidence combination process is devoted to calculating the value of *Overall evaluation*. The attributes ($Attribute_i\ (i = 1, \ldots, I)$) that distribute in the attribute level should be evaluated. Meanwhile, the values of these attributes can either be obtained from external input, or derived by the evaluation in the factor level. For instance, the value of $Attribute_i$ is determined by the $Factor_{i,u}$ set, where $u = 1, \ldots, U$.

In the proposed ER-based ODSE, since the original query term is expanded by SONT, a MADM tree model can be utilised to present this query term and all the expanded query terms, aiming at mapping each expanded query term into a hierarchical tree model. Similarly, the tree model consists of two hierarchical levels, where each synonym and the original query are located in the same level, and the hyponyms lie on the level lower. The document ranking process firstly combines all the relevance scores between each query term in the tree model and a document, and subsequently generates the overall relatedness between the original query term and the document.

For illustration purpose, the query term "*Fault diagnosis*" can be organised into a tree model by considering hierarchical relationships among the synonym

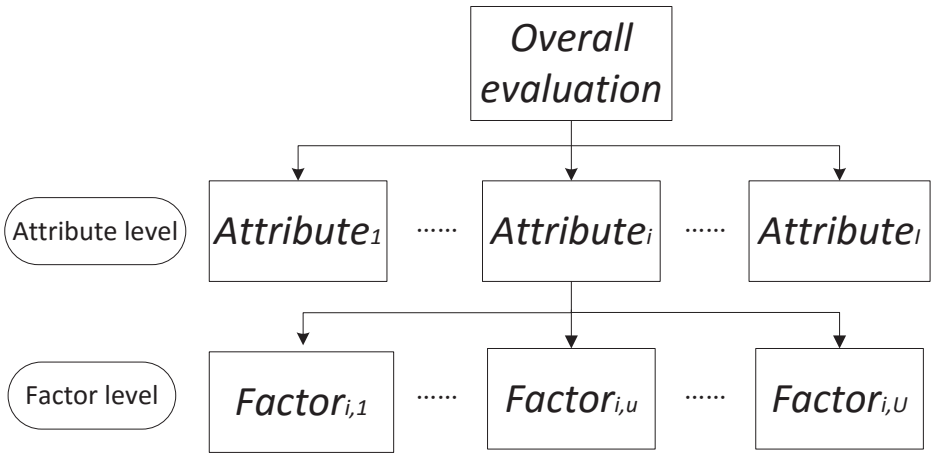


Figure 3.5: A general structure of a MADM tree model

set (3.1.1) and the hyponym set (3.1.2). The relevance score between the query term "*Fault diagnosis*" and a document is denoted as RS_{D} that represents the value of node *overall evaluation*. The overall relatedness between the query term and a document is concerned with all the expanded terms, which provide auxiliary evidence for determining RS_{D}. Also, the notation RS with a term subscript, e.g., the plural of "*Fault diagnosis*", i.e., $RS_{\text{Fault diagnoses}}$, is utilised to represent the relevance score between any single synonym or hyponym and a document. Subsequently, the decision making structure can be developed regarding the query term "*Fault diagnosis*" as illustrated in Figure 3.6. The components of the attribute level and factor level are derived based on the hierarchy as mentioned before. Meanwhile, the $overall\ evaluation$ of the four hyponyms can be investigated by the analysis of each relevance score, and denoted as OE_{Hyponym}. Thus, all the query terms with their relevance score are organised into a MADM tree model.

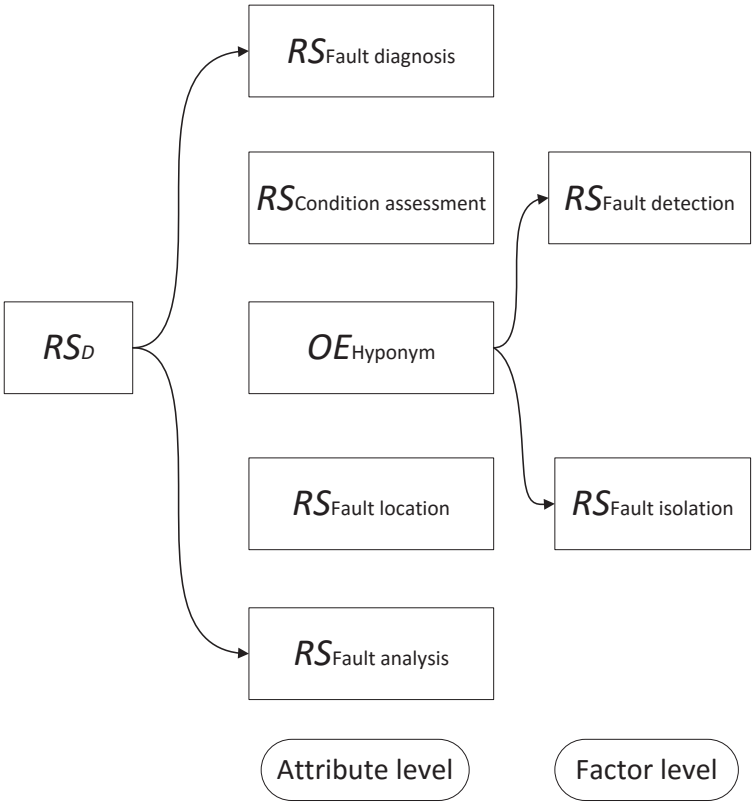


Figure 3.6: A tree model used for the combination of multiple relevance scores regarding to the synonyms (including the plural forms) and hyponyms of the query terms

3.2 Evidential Reasoning for Document Ranking

3.2.1 Brief introduction of the evidential reasoning

Since the original query term, the synonym set, and the hyponym set are mapped to a MADM tree model, the overall relatedness RS_{D} of this MADM tree model is computed by the ER algorithm. As mentioned in Section 3.1.2, the query terms from the expanded sets can be regarded as auxiliary evidence for determining the RS_{D}. The fulfilment of such purpose is based on the ER algorithm that combines all the evidence and provides decision making with uncertainties. The ER algorithm is capable of handling the problems such as assessment, decision analysis, evaluation problems, etc [119]. In the document ranking process, the ER algorithm is based upon the MADM tree model and the evidence combination rules of DS theory. Thus, the basis of DS theory is introduced in the following section.

3.2.2 Dempster-Shaper combination rules for evidence combination

The DS theory, which was firstly proposed by Dempster and expanded by his student Shafer [120–123], aiming at providing a mathematical model that combines evidence with uncertainty. Only the combination rules of DS theory is included in this thesis in order to meet the requirement of evidence combination.

In the DS theory, a sample space is defined as a frame of discernment Θ. A hypothesis (or singleton) H_s is defined as one element of Θ such that $H_s \subseteq \Theta$. In DS theory, all the hypothesis H_s in Θ should be mutually exclusive and exhaustive [123]. If one Θ consists of n different H_s, the Θ has 2^n subsets Ψ, including ϕ and Θ, where $\Psi \subseteq \Theta$. In such case, each subset of Θ can be assigned with a probability mass, denoted as $m(\Psi)$, which is named as the basic probability assignment in the DS theory, and $0 \leq m(\Psi) \leq 1$. If a piece of evidence is given, the basic probability assignment indicates belief in a hypothesis. According the definition above, the properties of the basic probability assignment can be

summarised as follows:

$$\begin{aligned} &\sum_{\Psi \subseteq \Theta} m\left(\Psi\right) = 1, \quad m\left(\phi\right) = 0; \\ &0 \leq m\left(\Psi\right) \leq 1 \qquad \text{for all } \Psi \in \Theta \end{aligned} \tag{3.2.1}$$

If there are two pieces of evidence, i.e., *A* and *B*, so that two basic probability assignments to a subset Ψ of Θ, i.e., $m_A(\Psi)$ and $m_B(\Psi)$, are provided. Thus, a combined probability assignment can be obtained by $m_{AB}(\Psi) = m_A(\Psi) \oplus m_B(\Psi)$. Finally, the evidence combination rule used for combining two probability assignments $m_A(\Psi)$ and $m_B(\Psi)$ is defined as follows:

$$\begin{aligned} m_{AB}(\phi) &= 0, \\ m_{AB}(\Psi) &= \sum_{h_1 \cap h_2 = \Psi} \frac{m_A(h_1) m_B(h_2)}{1 - K}, \\ K &= \sum_{h_1 \cap h_2 = \phi} m_A(h_1) m_B(h_2), \end{aligned} \tag{3.2.2}$$

where h_1 and h_2 denote two evaluation elements, which are selected from Θ in all possible ways, in which the intersection of h_1 and h_2 is Ψ. *K* is reflected by the conflicting situations where both $m_A(h_1)$ and $m_B(h_2)$ are not equal to zero, but the intersection $h_1 \cap h_2$ is ϕ.

3.2.3 Integrating basic probability assignments with evidential reasoning algorithm

The purpose of utilising the DS theory is to combine evidence in a MADM tree model. This section introduces the implementation of the DS theory for the document ranking task. From the DS combination rules illustrated in the last section, a hypothesis H_s can be defined as the relatedness between an expanded query term and any of its relevant documents in the document repository. The sample space Θ in the document ranking process can be treated as all the relatedness values computed from a query term and its relevant documents. Meanwhile, each individual query term, i.e., either the original query term or the expanded queries, can be regarded as a piece of evidence supporting the hypothesis, i.e., the relatedness, between the

query term and a document. Also, the relatedness value can be calculated by the probability assignments of the evidence, which are obtained by the relevance scores.

Assuming that, D_{repo} represents a document repository, consisting of n different documents. There are W documents retrieved from D_{repo} by a Lucene-based ODSE based upon all the query terms regarding to the query "*Fault diagnosis*". The relevant document set can be denoted by D_{rele}. As a result, each query term regarding to "*Fault diagnosis*" will generate W relevance score with respect to the corresponding document in D_{rele}. An illustration, in which one of the synonyms of "*Fault diagnosis*", i.e., " *Condition assessment*" with a set of relevance score regarding D_{rele}, is shown as follows:

$$\{RS_{\text{ca},1}, \ldots, RS_{\text{ca},w}, \ldots, RS_{\text{ca},W}\}, \tag{3.2.3}$$

where $RS_{\text{ca},w}$ represents the relevance score between the expanded query terms and one document D_w in D_{rele}, and the value of $RS_{\text{ca},w}$ should be within $[0, 1]$.

As the relatedness between "*Fault diagnosis*" and D_w is denoted as a hypothesis H_s in the DS theory, $RS_{\text{ca},w}$ can be considered as a confidence degree of "*Condition assessment*" assigned to D_w. In addition, there are *W* relevant documents of "*Fault diagnosis*" existing in D_{rele}, and the upper limit of the total belief committed to these documents by "*Condition assessment*" is 1. Each relevance score $RS_{\text{ca},w}$ can be normalised by equation (3.2.4).

$$\overline{RS}_{\text{ca},w} \;=\; \frac{RS_{\text{ca},w}}{W}. \tag{3.2.4}$$

The evidence set located at the attribute level of Figure 3.6 can be defined as e_i $(i \;=\; 1, \ldots, L_j)$, where L_j is defined as the number of evidence e_i. A basic probability assignment, that the evidence e_i supports the overall relatedness between *Fault diagnosis* and the document D_w then, is expressed as m_i^w. Also, a confidence degree confirmed by evidence e_i, e.g., $\overline{RS}_{\text{ca},w}$, is denoted by $\beta_{D_w}(e_i)$. Therefore, the basic probability assignment m_i^w, i.e., a belief, is determined by the following equation (3.2.5) [124].

$$m_i^w = \lambda_i \beta_{D_w}(e_i), \tag{3.2.5}$$

where $\lambda_i = [\lambda_1, \ldots, \lambda_{L_j}]^{\mathrm{T}}$ expresses the relative weight assigned to evidence e_i in the evidence set. Obviously, if there exists only one piece of evidence in the attribute level, m_i^w is equal to $\beta_{D_w}(e_i)$. The algorithm employed for determining the relative weights λ_i is introduced in the next subsection.

3.2.4 Methodology for assigning weights to the attributes

The relative weights for the evidence set derived in the last section can be generated by utilising the standard analytic hierarchy process (AHP) [122] [125]. Briefly, AHP is a multi-criteria decision support methodology used in management science [126]. A set of grades is assigned to the important values of evidence, which is based on the same standard. The fundamental grades utilised in this study are illustrated in Table 3.1. In AHP, a pair-wise comparison method is utilised

Table 3.1: Grade in the AHP

Values	Definition
1	Equally important or preferred
3	Slightly more important or preferred
5	Strongly more important or preferred
7	Very strongly more important or preferred
9	Extremely more important or preferred
2, 4, 6, 8	Intermediate values to reflect compromise

to a pair of evidence. The grades designed for locating evidence from an AHP pair are confirmed by search engine experts and power system engineers [126]. If $A_{m,r}(m = 1, ...L_j; r = 1, ...L_j)$ denotes a matrix and which the element $\xi_{\mathrm{m},r}$ represents the ratio comparison of mutual importance values between pair-wise evidences in an AHP. Then, a typical ratio comparison matrix is shown in equation

(3.2.6).

$$A_{m,r} = \begin{bmatrix} \xi_{1,1} & \xi_{1,2} & \cdots & \xi_{1,L_j} \\ \xi_{2,1} & \xi_{2,2} & \cdots & \xi_{2,L_j} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{L_j,1} & \xi_{L_j,2} & \cdots & \xi_{L_j,L_j} \end{bmatrix}. \tag{3.2.6}$$

The normalised matrix is obtained by each element divided by the sum value of each column that is expressed by equation (3.2.7).

$$\overline{A_{m,r}} = \begin{bmatrix} \overline{\xi_{1,1}} & \overline{\xi_{1,2}} & \cdots & \overline{\xi_{1,L_j}} \\ \overline{\xi_{2,1}} & \overline{\xi_{2,2}} & \cdots & \overline{\xi_{2,L_j}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\xi_{L_j,1}} & \overline{\xi_{L_j,2}} & \cdots & \overline{\xi_{L_j,L_j}} \end{bmatrix} \tag{3.2.7}$$

In this research, the mutual importance values between the relevance scores generated by an original query term and a synonym, the synonym and OE_{Hyponym}, the original query term and OE_{Hyponym} of the attribute level in Figure 3.6 are defined as 2:1, 2:1 and 3:1, respectively. Also, the weights are equally assigned to the elements of the factor level in Figure 3.6, as they are all the hyponyms of the original query term, which have equal influences to determine the overall relatedness between an expanded query term derived from the original query term and a document. An example of the implementation of the AHP technique is illustrated in Section 3.4.2.

3.2.5 Document ranking by Dempster-Shafer combination rules

Implementing the DS combination rules in ER aims to generate a set of values, each of which represents the overall relatedness between an expanded query term and a specific document. For the evidence set e_i as mentioned in Section 3.2.3, a combined probability assignment that indicates the relatedness between "*Fault diagnosis*" and a specific document of D_{rele} can be defined as $m_{e_i}^{H_s}(H_s \subseteq D_{\text{rele}})$. Furthermore, the remaining belief that is not assigned after commitments to all the documents of D_{rele} is defined as $m_i^{D_{\text{rele}}}$, which is $m_i^{D_{\text{rele}}} = 1 - \sum_{w=1}^{W} m_i^w$. Then, the formulas used for determining the overall relevance score between "*Fault diagnosis*" and D_w are illustrated in equation (3.2.8) and (3.2.9) [124]:

$$\{D_w\} \ : \ m^w_{e_{i+1}} = K_{e_{i+1}}(m^w_{e_i} m^w_{i+1} + m^w_{e_i} m^{D_{\text{rele}}}_{i+1} \ + \ m^{D_{\text{rele}}}_{e_i} m^w_{i+1}), \qquad (3.2.8)$$

$$\{D_{\text{rele}}\} \ : \ m^{D_{\text{rele}}}_{e_{i+1}} = K_{e_{i+1}} m^{D_{\text{rele}}}_{e_i} m^{D_{\text{rele}}}_{i+1}, \ w = 1, \ldots, W, \qquad (3.2.9)$$

where

$$K_{e_{i+1}} = \left[1 - \sum_{\tau=1}^{W} \sum_{\rho=1, \rho \neq \tau}^{W} m^{\tau}_{e_i} m^{\rho}_{i+1} \right]^{-1}, \ i = 1, \ldots, L_j - 1.$$

In the case that a query term is composed of more than one keyword, the overall relevance score generated by each of the keywords can be defined as a piece of evidence of the query term. Therefore, the final relatedness between such a query term and a document can then be derived with the above formulas by combining all the available evidence, with equally assigned mutual weights.

3.3 Document Search Engines Designed in this Thesis

Implementing the ER-based ODSE aims at improving the search accuracy of a document retrieval process, compared with that achieved by an ODSE using the weighted sum algorithm of VSM. There are four different document search engines, namely SE_1, SE_2, SE_3 and SE_4, respectively, which have been developed based on the Apache Lucene [101] search engine library and implemented for the tests. SE_1 represents a traditional keyword-matching document search engine with a weighted sum algorithm, which does not employ any QE techniques. SE_2 and SE_3 are two ODSEs implemented by a weighted sum algorithm and the ER algorithm, respectively. The NRW model is employed in SE_4 for comparison purposes. Briefly, the first part of SE_4 is exactly the same as SE_1. Compared with SE_1, SE_4 involves a second-pass retrieval consisting of three steps. Firstly, a pseudo-relevant document set is obtained from the retrieved documents. Secondly, a set of candidate expanded query terms is selected from the pseudo-relevant document set. Thirdly, the expanded query terms, which are used for the second-pass retrieval, are achieved by re-weighting the candidate expanded query terms based on the NRW

model. More details of the implementation can refer to the original paper [34]. The mechanisms of SE_1 to SE_4 are shown in Figure 3.7 to Figure 3.10, respectively. In a search process, all the returned documents regarding a query term are ranked from the highest relevance score to the lowest relevance score. Subsequently, the ranked document list is shown to the users as a result.

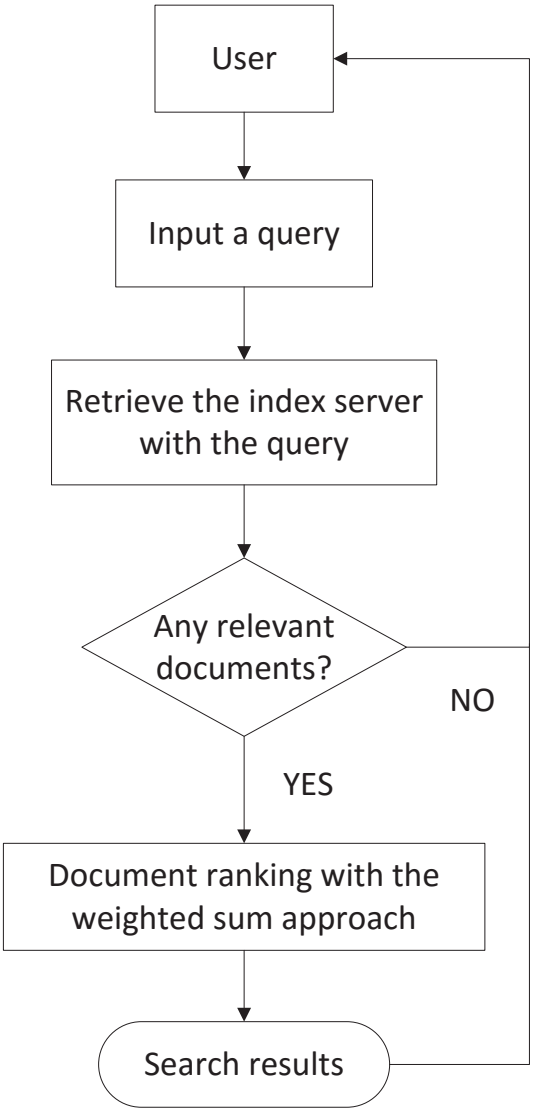


Figure 3.7: The mechanism of SE_1

In a Lucene-based document search engine, another main service offered besides

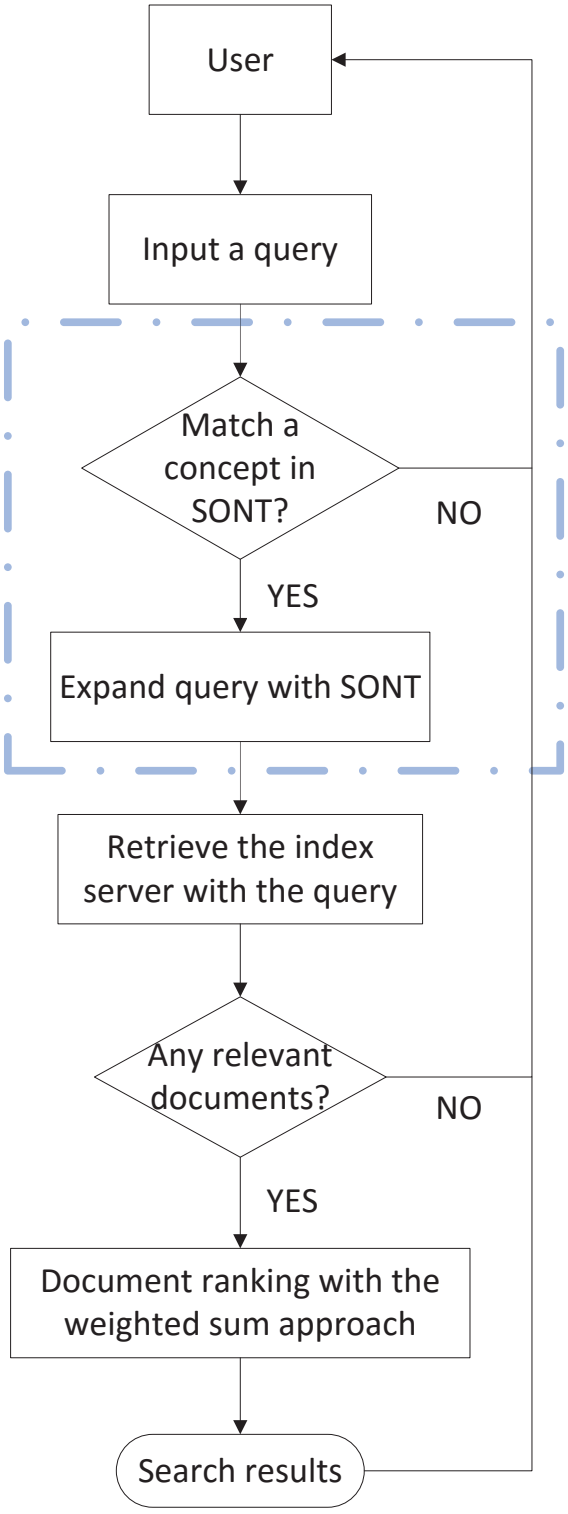


Figure 3.8: The mechanism of SE_2

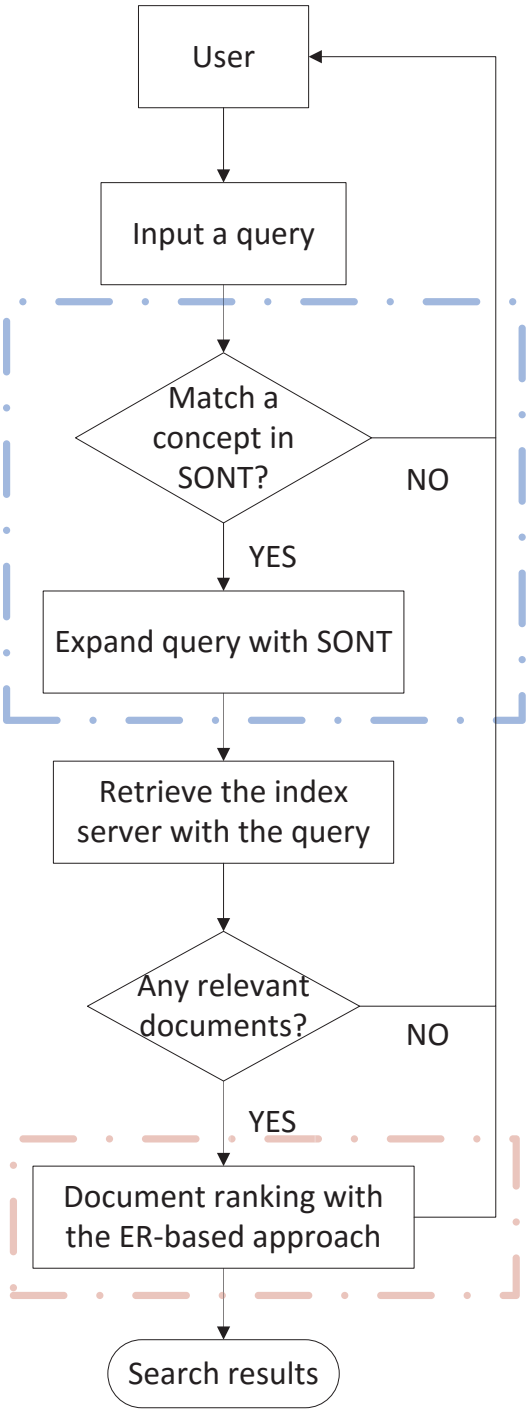


Figure 3.9: The mechanism of SE_3

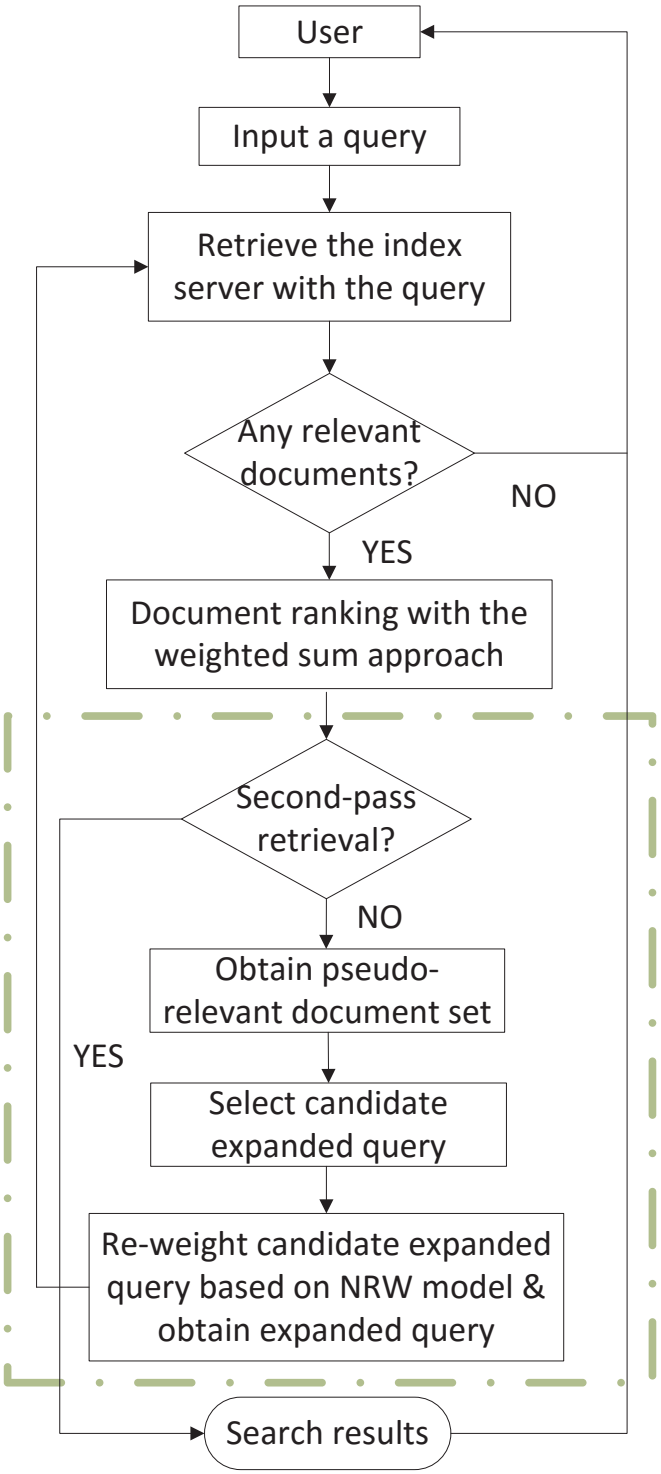


Figure 3.10: The mechanism of SE_4

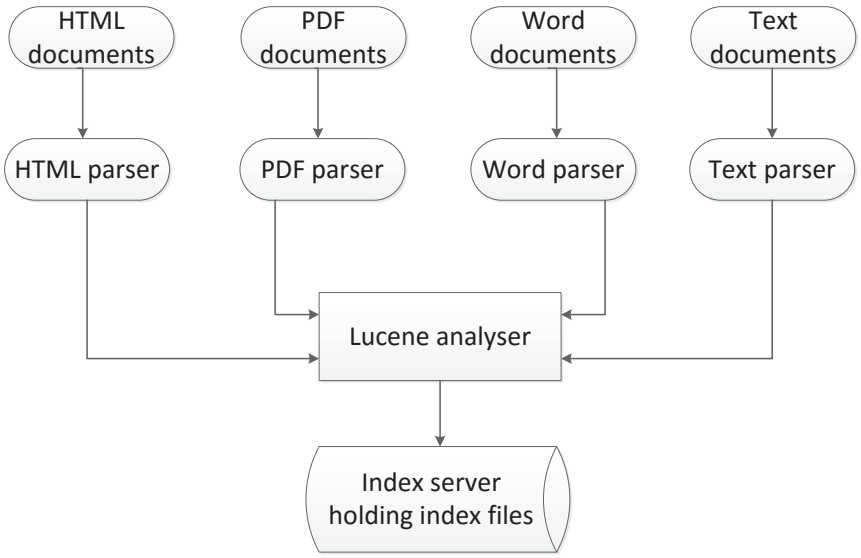


Figure 3.11: An index server generation process with Lucene

document searching is the indexing process. The indexing service of Lucene is used to store all useful content of documents existed in a specific document repository. The index server reduces search speed and improves search accuracy in document search processes with a submitted query term. The process of generating an index server with Lucene starts from employing several document parsers, as shown in Figure 3.11. It aims to extract text contents from different types of documents in a document repository. Subsequently, the obtained text contents are re-processed by the Lucene analyser into a set of index files, which is stored in the index server. The Lucene analyser contains two components, i.e., tokeniser and linguistic processor. Tokeniser transforms the content of a document into a sequence of terms, eliminates the punctuation and performs common stop words removal (e.g., “a”, “an”, “and” and “in”, etc. are removed). There exists a large number of stop words in every document that is not helpful for searching individual documents. For the linguistic processor, all the terms are changed to lowercase and the common morphological and inflectional forms are eliminated by suffix stripping algorithm [127], including stemming (e.g., automated $\rightarrow$ automate) and lemmatisation (e.g., criteria $\rightarrow$ criterion).

With the implementation of Lucene, each document is decomposed as different fields. In Lucene, the fields of a document can be either employed for searching documents and displaying a search result, or only displaying a search result. The index server used in this study is generated with Lucene according to the above index server generation procedures. An example of the Lucene analyser is given below:

1. Document 1: An ontology for fault diagnosis in electrical networks. [45]
   Document 2: Automated fault diagnosis at Philips medical systems. [128]
   Document 3: Ontology-based fault diagnosis for power transformers. [39]

2. Tokenisation: the three documents are transformed to tokens (Documents $\rightarrow$ tokens).
   "ontology", "fault", "diagnosis","electrical", "networks", "Automated", "fault", "diagnosis", "Philips", "medical", "systems", "Ontology", "fault", "diagnosis", "power", "transformers".

3. Linguistic process: tokens are changed to lowercase, and followed by stemming and lemmatisation (tokens $\rightarrow$ terms).
   "ontology", "fault", "diagnosis","electrical", "network", "automate", "fault", "diagnosis", "philips", "medical", "system", "ontology", "fault", "diagnosis", "power", "transformer".

4. Terms in indexer: terms obtained from last step form a dictionary as shown in Table 3.2.

5. Combine the same term to generate the posting list as illustrated in Figure 3.12."ID" denotes the document ID, "tf" and "df" refers to the term frequency and document frequency as introduced in Section 2.2.2.

Now, the index server is already built with the steps above. A relevance score obtained between a query term and a document is determined with a Boolean model combined with VSM. During the process of generating relevance scores between a term from a query term and a large number of documents, the Boolean model is utilised to filter the unrelated documents from the document set regarding the whole

Table 3.2: Term dictionary

Terms	Document ID		Terms with alphabet order	Document ID
ontology	1		automate	2
fault	1		diagnosis	1
diagnosis	1		diagnosis	2
electrical	1		diagnosis	3
network	1		electrical	1
automate	2		fault	1
fault	2		fault	2
diagnosis	2	$\Rightarrow$	fault	3
philips	2		medical	2
medical	2		network	1
system	2		ontology	1
ontology	3		ontology	3
fault	3		philips	2
diagnosis	3		power	3
power	3		system	2
transformer	3		transformer	3

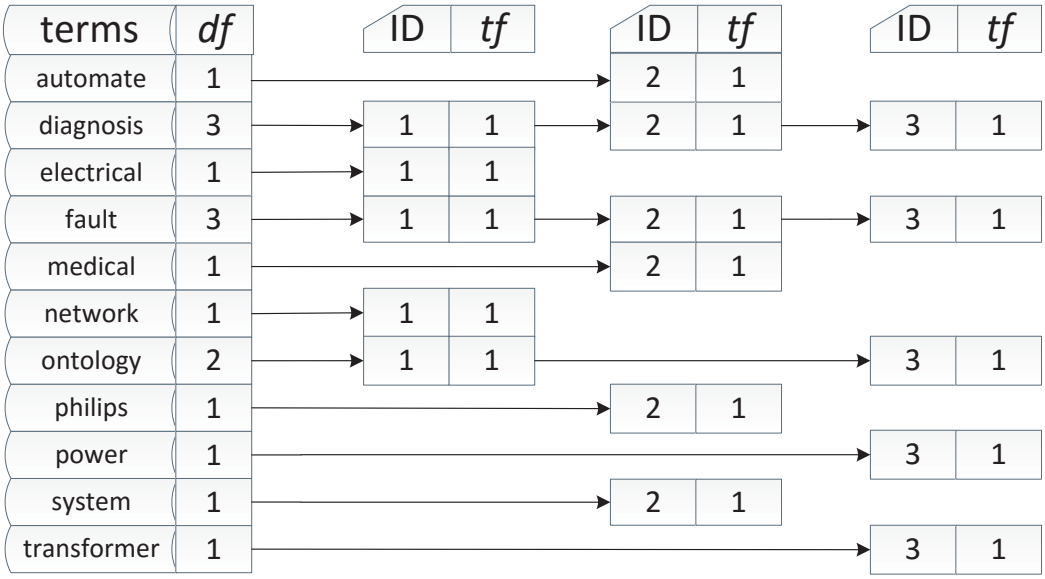


Figure 3.12: Posting list

query terms, based upon a Boolean logic. For instance, if a user's query is "ontology $\wedge$ power", document 3 will be retrieved. Afterwards, the relevance scores between each term of the query and the documents that exist in the filtered document set are calculated based on VSM, as introduced in Section 2.2.2. It is noted that if only a Boolean model is applied and the query input is in the form of "ontology $\wedge$ powers", there will be no document can be retrieved. This limitation also exists in other QE techniques. In contrast, SONT for QE contains both singular and the plural of a query term, so that the term "powers" can be mapped to its synonym set, i.e., "power". As a consequence, the accuracy of document retrieval can be improved.

3.4 Simulation Studies

3.4.1 Implementation configuration

This subsection demonstrates the implementation of the proposed ER-based ODSE. All the case studies are based on [2] [9]. Each search engine designed in Section 3.3 is employed to the same document repository, and the performance is

evaluated by the recall and precision technology. As presented in Table 3.3, the PSD, which contains 136,735 documents in total, has been applied throughout the simulations. All the keywords defined in SONT are employed as keywords to search relevant publications in the document repository. Meanwhile, the document pool in Table 3.3 refers to the pooling method [129], which is implemented for assisting to evaluate the performance of search engines and introduced in Section 3.4.3. Two query sets are utilised for investigating the search performance of each search engine. The query sets are selected by considering widely used query terms related to the power substations. The first set consists of 10 unique-keyword queries, and 10 combined-keyword queries compose the second set, as illustrated in Table 3.4. It is worth of mentioning that the tree model and keywords used in the tests are recommended by power system engineers concerning substation condition monitoring and assessment, which are summarised from technical reports and emails of the power system engineers consulted. Meanwhile, all the documents used for the performance evaluation are named differently, so that each of them could be considered as a distinct individual.

Table 3.3: Statistical information of the PSD in the simulation studies

Number of documents	136,735
Unique-keyword queries	10
Combined-keyword queries	10
Average number of documents per document pool	86.3
Average number of relevant documents per document pool	32.5

3.4.2 An illustration for the purposed document search engine

An example of implementation for SE_3 regarding the query term, i.e., "*Fault diagnosis*", is illustrated in this section, aiming to demonstrate the process of the proposed ER-based document ranking. In the first step of the search process, the query input "*Fault diagnosis*" is expanded with its synonym set and hyponym set, as depicted in (3.1.1) and (3.1.2), respectively. Subsequently, there are 24,933

Table 3.4: Queries employed for the simulation studies

Unique-keyword query set	Combined-keyword query set
1. Substation	1. Power system & Frequency response analysis
2. Transformer	2. Winding & Distortion analysis
3. Coolant	3. Relay & Fault location & Power system
4. Circuit breaker	4. Harmonics & Distortion & Power quality
5. Fault diagnosis	5. Transmission line & Protection & Relay
6. Dissolved gas analysis	6. Temperature & Overload & Thermal modelling
7. Relay	7. Transformer & Dissolved gas analysis
8. Switch	8. Fault analysis & Partial discharge
9. Thermal model	9. Transformer & Vibration & Mechanic
10. Voltage	10. Bus bar & Protection

documents in the PSD retrieved and identified to be the relevant documents of "*Fault diagnosis*" with the aid of the Boolean model in Lucene, as introduced in Section 3.3. Afterwards, the relevance scores between the document and the terms of the expanded query are computed. Then, the relevance score between the document and the expanded query is calculated with the ER algorithm. Finally, the documents are presented in a descending order according to generated overall relevance scores, which are delivered to users as final search results.

The document ranking process based on the DS theory as mentioned in Section 3.2 for two different documents is demonstrated as follows. As depicted in Table 3.5, two documents, i.e., the first two documents in Section 3.3 are denoted by D_{a} and D_{b}, respectively. The overall relevance scores regarding the query "*Fault diagnosis*" for D_{a} and D_{b} are expressed as RS_{a} and RS_{b}, respectively.

As mentioned in Section 3.1.2, the overall relevance score RS_{D} between the query term"*Fault diagnosis*" and a document is determined by the combination of relevance scores of itself, its synonyms and hyponyms. These relevance scores are treated as a set of evidence concerning the generation of the overall relevance scores RS_{D}. Based on the MADM tree model illustrated in Figure 3.6, the outputs of the

Table 3.5: Two documents used for ranking

	Document name
D_{a}	An ontology for fault diagnosis in electrical networks
D_{b}	Automated fault diagnosis at Philips medical systems

four hyponym branches can be treated as four attributes of the node OE_{Hyponym}. The relevance scores obtained by the four hyponyms regarding D_{a} and D_{b} are shown in Table 3.6. Then, the normalised relevance scores are calculated as shown in Table 3.7 using equation (3.2.4). These relevance scores can be further treated as the confidence degrees $\beta_{D_w}(e_i)$ according to equation (3.2.5). In the study, the

Table 3.6: Relevance scores in the factor level

	Fault detection	*Fault detections*	*Fault isolation*	*Fault isolations*
D_{a}	0	0	0	0
D_{b}	0.0202	0	0.0265	0

Table 3.7: Normalised relevance scores in the factor level

	Fault detection	*Fault detections*	*Fault isolation*	*Fault isolations*
D_{a}	0/24933	0/24933	0/24933	0/24933
D_{b}	0.0202/24933	0/24933	0.0265/24933	0/24933

relative weights λ_i are assigned as the same value 1/4 for each branch in the factor level of Figure 3.6. Therefore, the probability assignment m_i^w confirmed by "*Fault detection*" is generated as $0.0202/24933 \times 1/4 = 0.0202/99732$. Hence, when all the probability assignments are obtained for D_{a} and D_{b}, the values of OE_{Hyponym} are obtained as 0 and 4.6825×10^{-7} with equations (3.2.8) and (3.2.9) for D_{a} and D_{b}, respectively. Subsequently, the outputs of the evidence located at the attribute level

of Figure 3.6 are listed in Table 3.8, when normalised by equation (3.2.4).

$$\begin{bmatrix} 1 & 2 & 2 & 2 & 3 & 2 & 2 & 2 & 2 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/3 & 1/2 & 1/2 & 1/2 & 1 & 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 1/2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.4.1}$$

In order to integrate the relevance scores of Table 3.8 into scaled inputs for generating RS_{a} and RS_{b}, the relative weights λ_i are derived using AHP. As stated in Section 3.2.4, the mutual importance values between an original query term and a synonym, a synonym and OE_{Hyponym}, the original query term and OE_{Hyponym} are defined as 2:1, 2:1 and 3:1, respectively. Thus, matrix introduced in Section 3.2.4 can be expressed by equation (3.4.1).

Table 3.8: Relevance scores in the attribute level

	Fault diagnosis	*Fault diagnoses*	*Condition assessment*	*Condition assessment*
D_{a}	0.3782/24933	0/24933	0/24933	0/24933
D_{b}	0.5021/24933	0.0202/24933	0/24933	0/24933
OE_{Hyponym}	*Fault location*	*Fault locations*	*Fault analysis*	*Fault analyzes*
0	0/24933	0/24933	0/24933	0/24933
4.6825E-7	0.0362/24933	0/24933	0/24933	0/24933

With the AHP algorithm, the relative weight set of λ_i then is obtained as [0.2052, 0.1057, 0.1057, 0.1057, 0.0548, 0.1057, 0.1057, 0.1057, 0.1057]$^{\mathrm{T}}$. Finally, RS_{a} and RS_{b} are calculated as 3.1123×10^{-6} and 4.3971×10^{-6} with equations (3.2.8) and (3.2.9), respectively. Compared with the relevance score 3.1123×10^{-6} of D_{a}, it shows that D_{b} has a higher relatedness to the query "*Fault diagnosis*" with the

relevance score 4.3971×10^{-6}. As a consequence, D_{b} is returned with a higher ranking order in the final search result, i.e., a relevant document list, compared to that of D_{a}. Therefore, the document ranking scheme illustrated above can be implemented to rank a number of documents in a similar way given a query. In the next section, a recall and precision curve method, used for verifying the search performance of all the four search engines, is described.

3.4.3 Performance evaluation method

Typically, the recall and precision techniques, which are frequently used for evaluating search engine performance as reported by Chou [34], are considered as two important performance indices employed for evaluating the effectiveness of a document search engine. The NRW model was proven as the most competitive search engines compared to other four expansion term weighting functions in [34]. For each comparison, ten different recall levels are chosen. The gradient of the precision vs. recall curve [34] using NRW is smaller than other approaches. In addition, at each recall level, NRW has higher precision. The recall and precision method is a direct way to evaluate the performance of different search engines. Therefore, the method of calculating recall and precision is selected in tests to verify the search performance of $SE_1,\ SE_2,\ SE_3$ and SE_4. Given a query, its relevant documents in a document repository are defined as a set *R*. A set of relevant documents *H* is obtained by a document search engine after performing a search process. The recall and precision are illustrated in equation (3.4.2) and (3.4.3) [89].

$$R_c = \frac{H \cap R}{R}, \tag{3.4.2}$$

$$P_r = \frac{H \cap R}{H}, \tag{3.4.3}$$

where R_c is the recall value and P_r represents the search precision. Alternatively, as shown in Table 3.9, R_c is the proportion of retrieved documents amongst the relevant documents, i.e., $R_c = \frac{A}{A+C}$, and P_r is the proportion of relevant documents amongst retrieved documents, i.e., $P_r = \frac{A}{A+B}$, where "A","B","C" and "D" denote the number of documents.

Table 3.9: Evaluation metrics

	Relevant	Non-relevant	Total
Retrieved	A	B	A+B
Not retrieved	C	D	C+D
Total	A+C	B+D	A+B+C+D

In order to obtain the set *R* for each of the queries in Table 3.4, the pooling method is implemented, which is introduced by Coelho and Calado [129]. For each of the 20 queries shown in Table 3.4, SE_1, SE_2, SE_3 and SE_4 are utilised to implement a search process using the document repository, respectively. The top 50 ranked documents of each search process then are pooled into a document set for the corresponding query. Typically, such a document set is defined as a document pool. Then, by eliminating the reduplicate items in each document pool, the average amount of documents of the 20 document pools is derived as 86.3, as listed in Table 3.3.

For each of the 20 document pools, the documents are classified as *relevant* and *non-relevant* by power system engineers regarding the corresponding query. As a result, *R* of a specific query is known as the *relevant* document set. Moreover, the mean of document amount of the total 20 queries is obtained as 32.5 as illustrated in Table 3.3.

Implementing such a pooling method avoids evaluating all the documents in the document repository for a specific query. More significantly, *R* of all the 20 queries is determined in a logical way, and thus may lead to a more accurate test result. Since the method does not guarantee all the relevant documents of a query can be found and located in the corresponding document pool, therefore a recall value obtained in a search process then is defined as a *relative recall*. With each of the two query sets of Table 3.4, the average precisions of SE_1, SE_2, SE_3 and SE_4 are calculated at 10 different recall levels separately ranged from 10%, 20% to 100%. Also, the results of SE_1, SE_2 and SE_3 in the following section are reproduced from [2] [9].

3.4.4 Results and discussion

Figure 3.13 shows the average precision of the 10 unique-keyword queries at different recall levels. As can be observed from the figure, SE_1 presents the lowest precision throughout the search processes compared with the other three search engines. This may be due to the following reasons:

1. The synonyms of the original queries are not considered within the search processes of SE_1. Therefore, the documents which contain the synonyms, but not holding the original queries, cannot be retrieved by SE_1;

2. The hyponyms of the original queries may influence the search performance as well, which means cases without considerations of subclasses in SE_1 may reduce search accuracies;

3. For SE_2 and SE_3, more relevant documents are found by the expanded query and the ER-based document ranking techniques respectively. At the same recall levels H may be smaller in the results of SE_2 and SE_3 in tests, however $H \cap R$ may be higher due to the improved ranking of each relevant paper. As a result, the precision values of SE_2 and SE_3 are higher than that of SE_1 with respect to the same recall values, which are also verified in the tests.

In addition, the average precision of SE_3 is higher than those of SE_2 at all the recall levels. It indicates that the accuracy of a keyword-matching search engine can be improved with a QE technique provided by a domain ontology, the rough-and-tumble organisation of the relevance scores generated by the expansion terms can still restrict the search precision. Therefore, the potential of the proposed ER-based approach to improve the search accuracy of an ODSE can be verified with the unique-keyword queries.

Figure 3.14 presents the average recall-precision results of the 10 combined-keyword queries. Compared with Figure 3.13, the precision values of the four search engines have been improved. This means that utilising multi-keyword in one search process can refine the search scale, and thus improve the accuracy of document search engine outputs. Among the three search engines, SE_3 delivers the

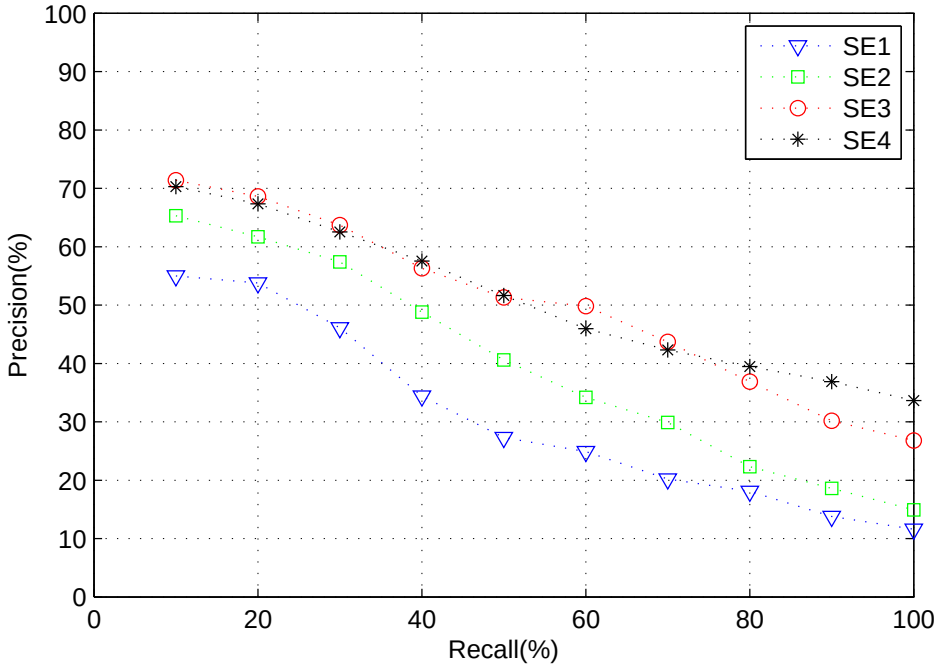


Figure 3.13: Average precision-recall curves with 10 unique-keyword queries

best precision at every recall level, while the precision achieved by SE_2 is much higher than that of SE_1. Compared between SE_3 and SE_4, SE_3 outperforms SE_4. A possible reason is that some of the retrieved expansion terms in SE_4 are less related to the initial query, while the expanded terms derived by SONT in SE_3 are closely related to the initial query. Therefore, for both the unique-keyword query set and the combined-keyword query set, the ER-based document ranking approach has demonstrated its capability of improving the search accuracy of an ODSE in terms of precision at the same recall levels.

Finally, the overall evaluation results of all the 20 queries are shown in Figure 3.15. Conventionally, a high precision is more important when it is generated at a low recall level, as users always start to read the retrieved documents with higher relevance scores. Therefore, as shown in Table 3.10, at the recall level of 10%, the precision of SE_3 is 80.9%, which is much higher than those of the other three search engines. The above results clearly show that, in an ODSE, the ER algorithm provides a suitable solution for combining multiple relevance scores of an expanded

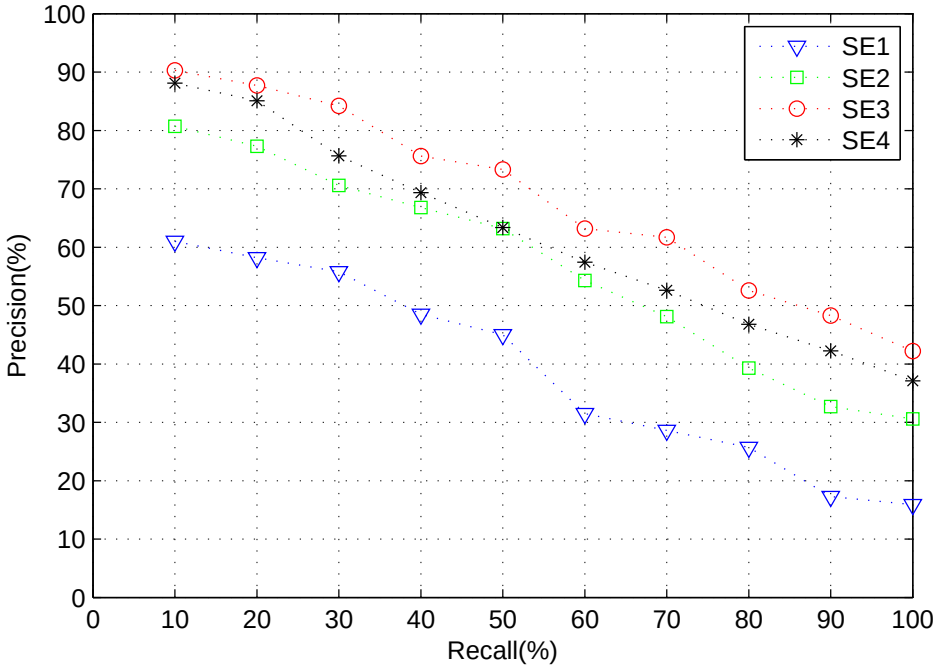


Figure 3.14: Average precision-recall curves with 10 combined-keyword queries

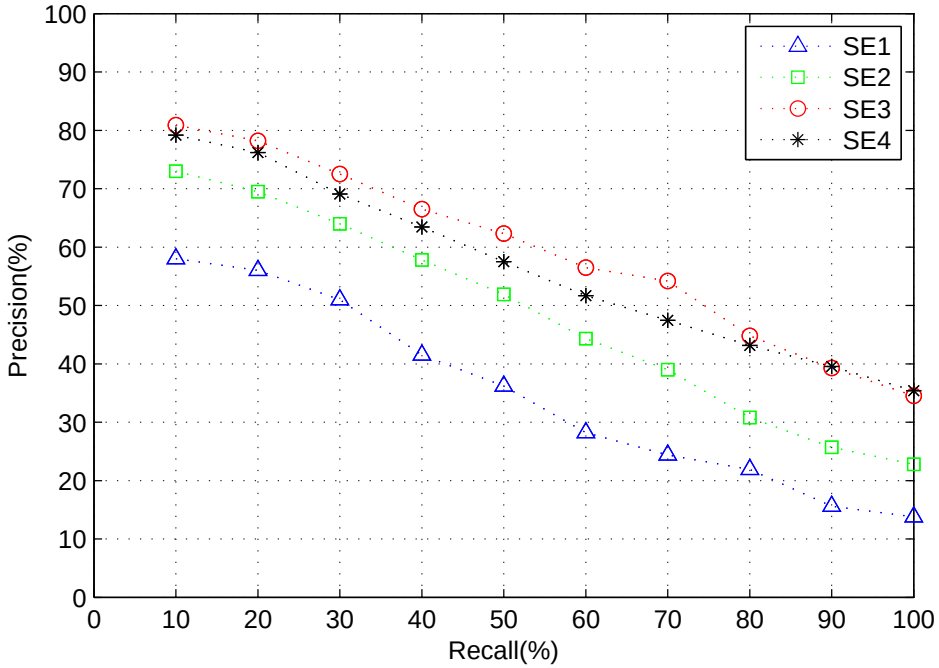


Figure 3.15: Average precision-recall curves with all 20 queries

Table 3.10: Average precisions of four search engines on recall levels of 10% and 20%

Method	10%	20%
SE_1	58.0	56.0
SE_2	73.0	69.5
SE_3	80.9	78.1
SE_4	79.2	76.2

query. Consequently, the search accuracy of an ODSE can be improved with the ER-based document ranking approach.

The main theoretical contribution of the proposed approach is the introduction of ER for organising an expanded query into a hierarchical tree model, in which the hierarchical relationships among the involved query terms is considered. Moreover, the practical contribution is the construction of an ontology model dedicated to power substation fault diagnosis. As mentioned in Section 3.1.1, the ontology model SONT contains a large number of classes and instances, which covers most important aspects of power substations, e.g., monitoring, status, devices, etc. Thus, it can be used as an open platform for expanding SONT. In addition, as reported in recent research, there are some new approaches on QE. However, most of them are based on the mechanisms of relevance feedback.

Recently, the adaptive co-training method and the proximity-based approach were proposed [98] [130]. Both of the two approaches are based upon Rocchio's model as NRW and have been proven to be much more competitive than other basic retrieval models. However, as mentioned in Section 1.3.1, if there are not sufficient documents used for analysis before a search process, the relatedness between related terms and an original query cannot be ensured. In comparison, SONT is a well-defined domain ontology, which functions as WordNet, containing a set of concepts related to power substation as well as their synonyms and hyponyms. It is a more direct and effective way to realise QE. It is believed that an ontology specially designed for substation fault diagnosis is very useful in practice, which is scalable by adding new diagnosis terms and rules in SONT.

3.5 Summary

In order to tackle the problems existing in the traditional ODSEs, an ER-based document ranking approach has been proposed in this chapter. It is the first time that the terms of an expanded query are organised into a MADM tree model in the search process of an ODSE. The ER algorithm was proposed for the combination of the relevance scores generated between terms of an expanded query and a document. This novel approach used in an ODSE aims to generate the overall relatedness between an expanded query and a document. Practically, it can be implemented with other QE techniques, e.g., relevance feedback-based techniques and statistical co-occurrence-based techniques, without considering the way of how an expanded query is generated. The traditional keyword-matching search engine, the NRW model, and the two ODSEs with and without the ER algorithm were tested using a number of queries related to IR of substations, respectively. Compared with the keyword-matching search engine and the NRW method, the proposed approach outperformed in terms of both recall and precision. The final results demonstrated that the ER-based approach has provided a viable solution for ranking documents in an ODSE and the search precision of the ODSE has been improved significantly at the same recall level with ER embedded.

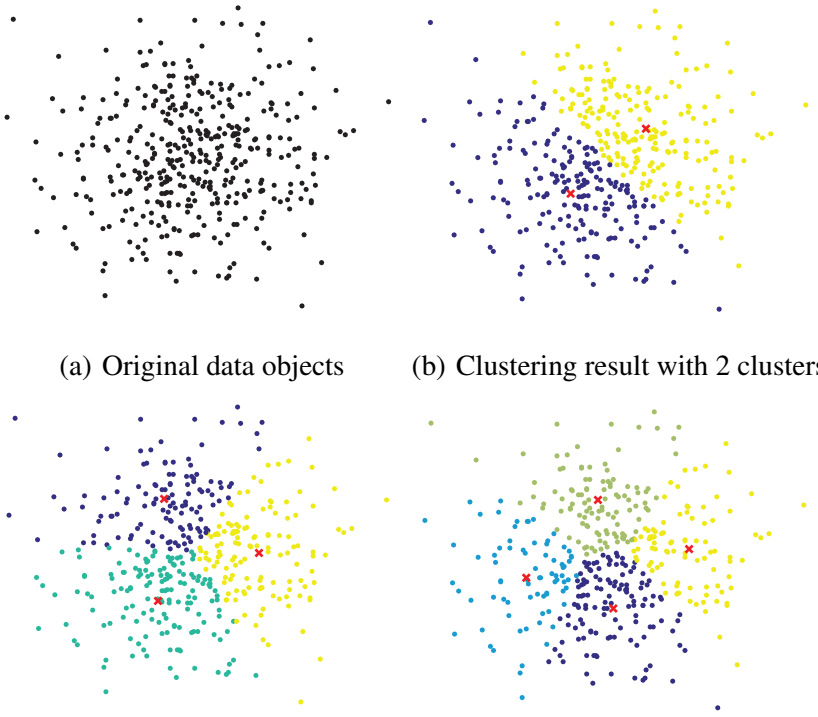
Chapter 4

Performance Evaluation of Clustering Algorithms

This chapter and Chapter 5 concentrate on the approach of CC for PSD. Before presenting our approach, this chapter focuses on evaluating the performance of clustering algorithms based upon a set of simulation studies, aiming to find an advantageous algorithm to handle the issue of document clustering. In Section 4.1, a brief introduction of cluster analysis is presented, followed by the common clustering applications, in which the generic document clustering procedure is provided. Subsequently, the purpose of CC, which is capable of improving clustering result, is illustrated. Section 4.2 demonstrates three consensus algorithms, i.e., NNMF-CC [52], WPK-CC [54], and INT-CC [53], which are selected in this thesis for comparison and implementation purposes. Also, a set of validation methods are presented in Section 4.3. In Section 4.4, all the clustering algorithms are tested on three sample datasets and four document datasets. The performance are evaluated by the selected validation methods. The results show that WPK-CC outperforms other algorithms on both types of datasets.

4.1 Introduction

As briefly discussed in Section 1.3.2, clustering or cluster analysis refers to a procedure that automatically arranges data into meaningful groups. Each group of objects is named as a cluster, in which each object is similar to the others and dissimilar to the objects in other groups. This type of data analysis is defined as a tool to examine the underlying class of each data [32]. It is also applied as an exploratory data analysis to discover unknown knowledge, in which the characteristics of the intrinsic contents that users are interested but unable to analyse. In this subsection, a systematic introduction of cluster analysis is given.



(a) Original data objects (b) Clustering result with 2 clusters

(c) Clustering result with 3 clusters (d) Clustering result with 4 clusters

Figure 4.1: Clustering results of different cluster numbers, i.e., 2, 3 and 4, respectively

In the machine learning community, clustering belongs to unsupervised learning, aiming to find hidden structure in unlabelled data [131]. It is different from supervised learning, which uses training samples from a given set of categories to learn a model [11]. In other words, the same set of data potentially has

different clustering outcomes, according to different initial parameter settings (e.g., number of clusters) for clustering, while priori classes information is assigned to a supervised learning problem. Normally, a dataset is a set of data objects, in which each data object consists of a fixed number of attributes (features) or weighted terms (e.g., a document-term vector). Each object is defined as a point in a multi-dimensional space. Clustering solution of a dataset is obtained by being applied with a suitable clustering algorithm with relevant settings of parameters. Subsequently, the quality of the clustering solution will be quantitatively evaluated by the validation methods. Figure 4.1 is an illustration of one dataset with different clusters. Figure 4.1(a) shows the original dataset, and three clustering results with cluster number of 2, 3 and 4 are shown in Figure 4.1(b), Figure 4.1(c) and Figure 4.1(d), respectively. The red crosses in each figure denote the center of each cluster.

4.1.1 Common applications of clustering

Cluster analysis is widely applied to many research areas, e.g., bio-informatics [132], image analysis [133], multimedia signal processing [134], marketing research [135], etc. Among all the applications, cluster analysis can be summarised as four major purposes as follows:

Data reduction : In most cases, a data collection is very large, which makes the data processing difficult to be achieved. Cluster analysis is utilised to group data into a number of "sensible" clusters, of which each cluster has a much smaller size so that the complexity of process can be reduced.

Hypothesis generation : It aims to infer some hypotheses concerning the nature of the data.

Hypothesis testing : Cluster analysis is used for the verification of the validity of a specific hypothesis.

Prediction based on groups : The resulting cluster from the process of cluster analysis is characterised based on its characteristics of the patterns. Therefore,

if an unknown new pattern is given, the cluster to which it is more likely to belong can be determined.

4.1.2 Document clustering

In the application of document clustering, a document repository refers to a dataset and each document in the document repository is one object. As summerised in Section 1.3.2, documents with a similar topic have high similarities, while documents within the same cluster disscuss different topics from documents in the other clusters. There is no need to provide labelled categories so that each document can be managed into the corresponding group based on the label [32]. The generic procedure of document clustering is presented in Figure 4.2 [32]. The first step, i.e., document repository pre-processing, has been illustrated in Section 2.2.2 and Section 3.3, in which the original document dataset is pre-processed into a weighted document-term matrix. Each row of this matrix represents a document in the document repository and each column stands for parsed term with assigned weight. The remaining steps of document clustering, i.e., applying clustering algorithms and results validations, are presented in the following sections.

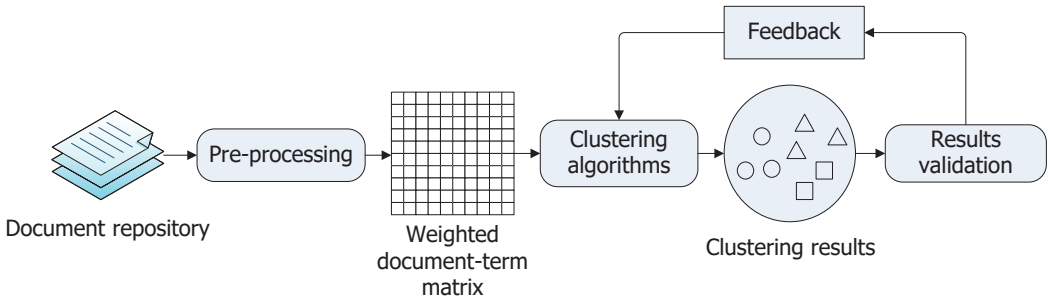


Figure 4.2: The generic document clustering procedure

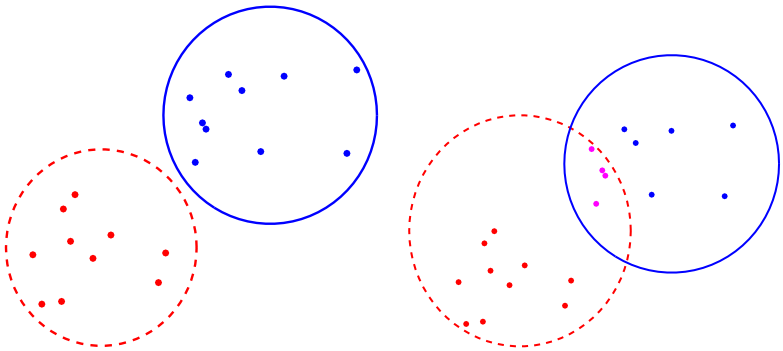
4.1.3 Clustering algorithms

A plenty of algorithms for clustering have been proposed, which can be mainly divided into three categories, i.e., exclusive clustering, overlapping clustering, and

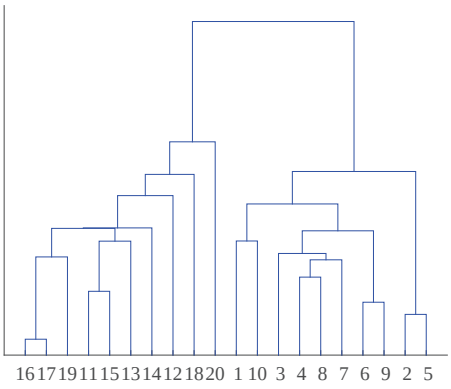
hierarchical clustering [32] [131]. The exclusive clustering is based upon an exclusive way, each data object is only allocated to one cluster. Normally, the algorithms in this type repeatedly refine a condition in order to find an optimal value of an objective function. in which an iterative refinement process that attempts to optimise a given objective function. In contrast, the fuzzy sets are involved in the overlapping clustering algorithms, of which each object in the clustering result belongs to more than one clusters with different degrees. As for a hierarchical clustering, the clustering result is in a tree model with hierarchy, which could be obtained by either a top-down or bottom-up mechanism. Figure 4.3 illustrates example clustering results for each type of clustering algorithm that is applied on a twenty-object dataset. The types of clustering algorithm selection is always problem dependent and based on user's objective. Although many documents or texts concern varied contents and different topics, they normally have a main emphasis and purpose. A document holds a large percentage of contents and uses many keywords to describe its main topic. In a particular domain, e.g., power substations, each research document or report has very high pertinence. Therefore, a set of clear and explicit clusters of document is essential for power experts to obtain papery knowledge and fast learning. Thus, in this research, only the first type of clustering algorithm, i.e., exclusive clustering, is studied. In this case, documents in the same cluster have significant similarity with respect to relevance to the information needs.

k-means

K-means is one of the best-known exclusive clustering algorithms [136], which is easy to be implemented, performing as a foundation algorithm for researchers to design new algorithms for clustering. Basically, it is an iterative procedure that is guaranteed to converge, though not always to the best solution, and "k" refers to the number of clusters. K-means revolves around the placement and replacement of "k" centroids denoted as c_j. The centroid of a cluster is defined to be the average of the vectors in that cluster, i.e., if a vector $\{x_1, x_2, \ldots, x_n\}$ forms a cluster then the centroid of that cluster is $(1/n) \sum_{i=1}^{n} x_i$ as the red crosses illustrated in Figure 4.1.



(a) Exclusive clustering (b) Overlapping clustering



(c) Hierarchical

Figure 4.3: A twenty-object dataset clustered by three types of clustering algorithm

Each data object is put in the cluster associated with the nearest centroid. The algorithm aims at minimising the objective function as equation (4.1.1).

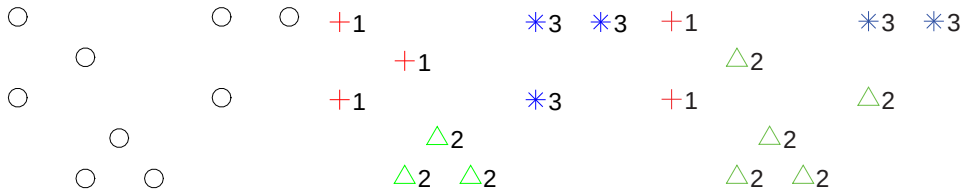
$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2, \tag{4.1.1}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is the chosen distance measure (Euclidean distance) between a data point $x_i^{(j)}$ and the cluster centroid c_j. These steps are repeated until the centroids remain stationary or there's no data point moving to any groups [136].

The form of a clustering result

The clustering result is an integer vector, in which each element represents a cluster label. Figure 4.4 shows an example of nine objects with three underlying

classes and a three-cluster result, i.e., $k = 3$. Thus, the clustering result in Figure 4.4(c) is denoted as $\{(1,1,1),(2,2,2,2),(3,3)\}$ or "111222233".



(a) Original dataset (b) The underlying classes of the original dataset (c) The clustering results with assigned labels by any clustering algorithm

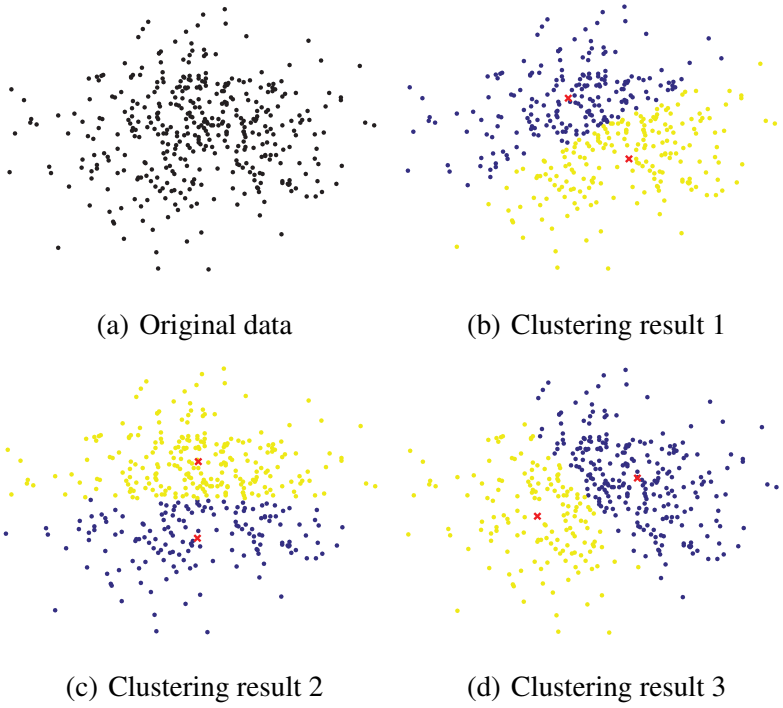
Figure 4.4: A nine-objects dataset with three underlying classes v.s. labelled clustering result

4.1.4 Limitations of single clustering algorithms

The output of k-means is highly sensitive, because different initial points or different iterative steps may generate alternative clustering results. Apart from k-means, many researchers have concluded that the single clustering algorithm with different initial states or iterative steps can result in diverse results as illustrated in Figure 4.5. It is not appropriate to decide which clustering result is correct or not, as they are all obtained by equally plausible clustering algorithms. Moreover, the evaluation of the results is associated to a validation method, of which the evaluation results can not be ensured. There is no cluster validity indexes impartially evaluating the results of any clustering algorithms [37] [38].

4.1.5 Consensus clustering

The supplement method for the single clustering algorithm refers to a combination procedure, named consensus clustering or cluster ensemble [37]. This procedure aims at combining all the clustering results of a same dataset from different clustering algorithms. Normally, there are several properties of CC reported in numerous of studies, including robustness, consistency, novelty, stability,



(a) Original data (b) Clustering result 1

(c) Clustering result 2 (d) Clustering result 3

Figure 4.5: Three clustering results by different runs of k-means

etc [137]. From the properties of CC, it can be summarised that CC algorithms have better average performances than that of the single clustering algorithms; the resultant result should be as similar as all clustering results obtained from single clustering algorithms; it receives a solution that no single clustering algorithm can reach; and noise or incorrect results from single clustering algorithms have little influence to the CC solution. In general, given a set of objects, CC methods mainly have fundamental steps, i.e., generation and consensus function, as listed in Table 4.1.

Table 4.1: Two principle steps of consensus clustering

Generation	To generate a set of clustering results, i.e., partitions
Consensus function	To integrate the obtained partitions from the generation step

The generation step intends to generate a set of alternative clustering results named partitions. These results are obtained by equally plausible clustering algorithms. Subsequently, the consensus functions are utilised to combine the

alternative clustering results according to the properties as mentioned above. There are two main types of consensus functions, i.e., objects co-occurrence and median partition [138]. The first approach aims to determine the correct cluster label for each object in the consensus partition, e.g., relabelling and voting [139]. It analyses the frequency of an object or objects belonging to the same cluster. Subsequently, a voting process is used, in which each object should vote for the cluster to which it will belong in the consensus partition [139]. For the second approach, the CC result is obtained by the solution of optimisation, aiming to find the consensus partition, which maximises the similarities or minimises the dissimilarities within all partitions [138]. In practice, different CC algorithms based on the median partition approach focus on designing the (dis)similarity measures between partitions that the overall (dis)similarity measure will be minimised or maximised, respectively.

Studies focus more on the consensus functions, which are based on the median partition approach, than the ones based on the objects co-occurrence, as the median partition approaches solve the consensus issue more rigorous [138] [140]. In spite of that, the correspondence between these two approaches does not to be unique, as some methods are used in both approaches with different peculiarities or purposes [37] [53] [139]. For instance, INT-CC, which belongs to the median partition approach, utilises relabelling method to solve the cluster label correspondence problem. For instance, if there are three 3-cluster partitions, i.e., Partition 1: $\{(1,1,1),(2,2,2,2),(3,3)\}$, Partition 2: $\{(2,2,2),(3,3,3,3),(1,1)\}$ and Partition 3: $\{(1,2,1),(2,1,2,2),(1,3)\}$, objects are assigned different cluster labels in these three partitions. However, Partition 1 and Partition 2 actually are the same clustering results, but the cluster labels are not identical. Therefore, the relabelling procedure is necessarily involved before considering the consensus function so that Partition 2 can be modified as $\{(1,1,1),(2,2,2,2),(3,3)\}$. Subsequently, the median partition approach is applied to combine all the partitions.

In this research, more efforts are carried out on the median partition approach. As it is based on optimisation, a brief literature review of optimisation techniques is given, and the optimisation techniques, which are involved in this research, are presented in the next section.

4.2 Consensus Clustering Algorithms

In this part of the work, three novel consensus algorithms are introduced, i.e., NNMF-CC, WPK-CC, and INT-CC. Before presenting each agorithm, the relevant notations are defined. Suppose $X = \{x_1, x_2, ..., x_n\}$ denotes a set of "n" objects, where each x_i is a tuple of some m-dimensional space, s partitions $P_X = \{P_1, P_2, ..., P_s\}$ of the data points in X are generated by k-means with different runs. Following previous definitions, for document clustering, n, is the number of documents, and m denotes the number of extracted word terms in the document repository. Each partition P_l, $l = 1, ..., s$ consists of a set of clusters $C^l = \{C_1^l, C_2^l, ..., C_k^l\}$ where k is the number of clusters for partition P_l and $X = \bigcup_{c=1}^{k} C_c^l$. The consensus partition is denoted as P^*, which is the combination of all partitions generated from different initialisations of k-means.

For a median partition approach, the consensus partition is investigated by an optimisation problem, aiming to find the median partition, which has the maximum similarity between all the partitions. Thus, the median partition is defined by equation (4.2.1) [37].

$$P^* = \arg \max_{P \in P_X} \sum_{j=1}^{s} Sim(P, P_j), \tag{4.2.1}$$

where Sim is a similarity measure between two partitions. The median partition is defined as the partition that maximises the summation of the similarities of all pair-wise partitions. Equivalently, it can also be explained and denoted by minimising the dissimilarities within all partitions. For each CC algorithm, relevant methods (e.g., NNMF, kernel methods or information theory) are utilised to construct the similarity measure or dissimilarity measure for the median partitions. Subsequently, the obtained objective functions are solved by selected optimisation methods. The details information of each CC algorithm are presented in subsections.

4.2.1 Non-negative matrix factorisation-based consensus clustering algorithm

NNMF-CC was proposed by Li and Ding [52], which is based on non-negative matrix factorisation (NNMF) [141] referring to the problem of factorising a given non-negative matrix into two matrix factors. This algorithm starts from defining a connectivity matrix, which is used to demonstrate the relationship between element-wise distance in two partitions. Thus, a primary objective function is obtained. Afterwards, cluster indicators are designed according to the connectivity matrix so that the objective function is transferred into a symmetric NNMF and solved by NNMF multiplicative update rules. The essential contents of NNMF and NNMF-CC are presented in the subsections.

Basics of NNMF

NNMF [141] is one of the representative of median partition, aiming to decompose a non-negative $m \times n$ matrix of column data $X = \{x_1, x_2, ..., x_n\}$ into the product of an $m \times r$ of feature vectors, $W = \{w_1, w_2, ..., w_r\}$, and an $n \times r$ coefficients matrix with column data $H = \{h_1, h_2, ..., h_n\}$, so that $X \approx WH^T$., where H^T represents the transpose of H. Alternatively, it is shown as equation (4.2.2).

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \approx \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1r} \\ w_{21} & w_{22} & \dots & w_{2r} \\ \vdots & \ddots & \ddots & \vdots \\ w_{m1} & w_{m2} & \dots & w_{mr} \end{bmatrix} \times \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ h_{r1} & h_{r2} & \dots & h_{rn} \end{bmatrix} \tag{4.2.2}$$

The decomposition is created by solving the following non-linear optimisation problem, as shown in equation (4.2.3).

$$\frac{1}{2}\left\|X - WH^T\right\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\left\|X_{:i} - W(H_{:i})^T\right\|^2, \tag{4.2.3}$$

where $\|X\|_F = \sqrt{\sum_{ij} x_{ij}^2}$ denotes the Frobenius norm [52]. One can minimise it with respect to each of the rows of H separately [141]. This results in solving a

sequence of quadratic problems as shown in equation (4.2.4).

$$\min_{h\geq 0} F(h) \quad \text{where } F(h) = \frac{1}{2}\left\|x - Wh\right\|^2. \tag{4.2.4}$$

Supposing a current approximation of the solution is $\overline{h} > 0$, the equation (4.2.4) can be formulated to the form of equation (4.2.5).

$$\min_{h\geq 0} \overline{F}(h) = \min_{h\geq 0} \frac{1}{2}\left[\left\|x - Wh\right\|^2 + (h-\overline{h})^T K_{\overline{h}}(h-\overline{h})\right], \tag{4.2.5}$$

where $K_{\overline{h}} = Dv - W^T W$ with $v = \frac{\left[W^T W\overline{h}\right]}{\left[\overline{h}\right]}$ and D is diagonal matrix. Because $K_{\overline{h}}$ is proven to be the positive semi-definiteness in [141], $\overline{F}(h) \geq F(h)$ for all h and especially $\overline{F}(\overline{h}) = F(\overline{h})$. Furthermore, the function is also convex so that in order to obtain the minimum h^*, the derivative $\nabla_h \overline{F}$ is calculated by equation (4.2.6).

$$\nabla_h \overline{F} = W^T W h - W^T x + K_{\overline{h}}(h - \overline{h}) = 0, \tag{4.2.6}$$

Thus,

$$(W^T W + K_{\overline{h}})h^* = W^T x - K_{\overline{h}}\overline{h}. \tag{4.2.7}$$

It is noticed that $W^T W + K_{\overline{h}} = Dv$ and $K_{\overline{h}}\overline{h} = 0$. Therefore,

$$h^* = \overline{h}\frac{W^T x}{W^T W\overline{h}}. \tag{4.2.8}$$

Since h^* is the global minimum of $F(\overline{h})$, we have $\overline{F}(h^*) \leq \overline{F}(\overline{h})$. Moreover, $F(\overline{h})$ is constructed to satisfy $\overline{F}(h) \geq F(h)$ for all h. Therefore, the relationships, i.e., $F(h^*) \leq \overline{F}(h^*) \leq \overline{F}(\overline{h}) = F(\overline{h})$ is obtained. In other words, the cost function contains a descent. As illustrated in Figure 4.6, the functions $F(h)$ and $\overline{F}(h)$ are plotted and the relationship of $F(h^*)$, $\overline{F}(h^*)$, $\overline{F}(\overline{h})$ and $F(\overline{h})$ is presented. The function $\overline{F}(h)$ is called auxiliary function. Solving for all rows of H results in the desired updating rule for H. The updating rule for W can be derived similarly. The multiplicative update rules for NNMF are summarised as follows [141]:

1. Initialize W^0, H^0 and $i = 0$;

2. REPEAT;

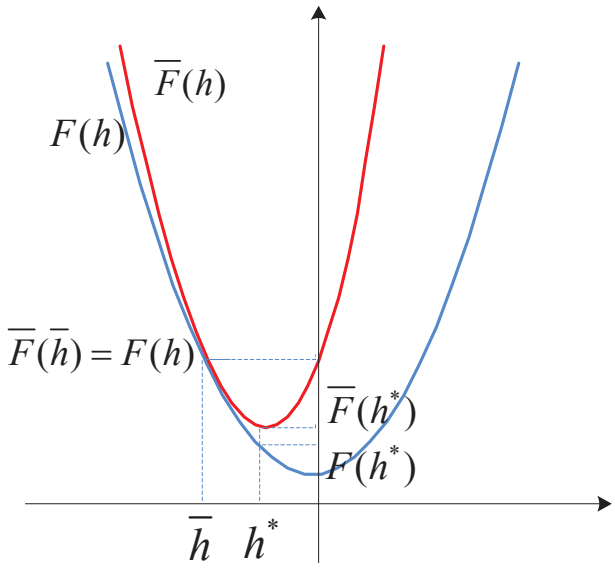


Figure 4.6: The multiplicative update rules for NNMF

3. $W^{i+1} = W^i \cdot \frac{XH^i}{W^i(H^i)^T H^i}$;
4. $H^{i+1} = H^i \cdot \frac{X^T W^{i+1}}{H^i(W^{i+1})^T W^{i+1}}$;
5. $i = i + 1$;
6. UNTIL Stopping condition.

Orthogonal NNMF

The orthogonality of matrix factors in NNMF was proposed by Ding and Li [142]. The one-sided W-orthogonal NNMF is defined in equation (4.2.9):

$$\min_{W \geq 0, H \geq 0} \|X - WH\|^2, \ \ s.t. \ \ W^T W = I, \tag{4.2.9}$$

where I is an identity matrix. The advantage of orthogonal NNMF is the uniqueness of the solution. For any given solution (W, H) of NNMF: $X = WH$, there are more than one matrix (A, B) such that equation (4.2.10) holds:

$$AB = I, WA \geq 0, WB \geq 0. \tag{4.2.10}$$

Therefore, (WA, WB) is also the solution with the same $\|X - WH\|^2$. The orthogonality condition $W^TW = I$ in the NNMF ensures that there is no matrix A and B satisfying both equation (4.2.10) and the orthogonality condition $(FA)^T(FA) = I$, unless A and B are permutation matrices [142]. Also, it is a continuous consideration that utilising the orthogonality on both W and H simultaneously in NNMF, as illustrated in equation (4.2.11).

$$\min_{W\geq 0, H\geq 0} \|X - WH\|^2, \ s.t. \ \ W^TW = I \ , H^TH = I. \tag{4.2.11}$$

The double orthogonality is very restrictive, producing a rather poor matrix low-rank approximation. In this case, an extra factor S was added to the objective function by Ding and He [143], which is used to absorb the different scales of X, W and H. S provides additional degrees of freedom such that the low-rank matrix representation remains accurate. Also, W gives row clusters, and H shows column clusters. The objective function is expressed by equation (4.2.12):

$$\min_{W\geq 0, H\geq 0, S\geq 0} \|X - WSH\|^2, \ s.t. \ \ W^TW = I \ , H^TH = I. \tag{4.2.12}$$

It is noted that if the X contains a matrix representing pair-wise similarities, i.e., $X = X^T$, the row clusters W will be equal to the column clusters, i.e., $W = H$. Hence, the objective function can be converted to a symmetric NNMF as presented in equation (4.2.13), where W is substituted by H.

$$\min_{H\geq 0, S\geq 0} \left\|X - HSH^T\right\|^2, \ s.t. \ \ H^TH = I. \tag{4.2.13}$$

Distance measure between partitions in NNMF-CC

The first step of NNMF-CC is to define a distance measure between two partitions. Following the notations defined at the beginning of Section 4.2, the distance between two partitions P_1 and P_2, as denoted by $d(P_1, P_2) = \sum_{i,j=1}^{n} d_{ij}(P_1, P_2)$, where the element-wise distance is defined by equation (4.2.14)

$$d(P_1, P_2) = \begin{cases} 1 & (i,j) \in C_k(P_1) \quad \text{and} \quad (i,j) \notin C_k(P_2); \\ 1 & (i,j) \in C_k(P_2) \quad \text{and} \quad (i,j) \notin C_k(P_1); \\ 0 & \text{otherwise,} \end{cases} \tag{4.2.14}$$

where $(i,j) \in C_k(P_1)$ means that object i and object j belong to the same cluster in partition P_1 and $(i,j) \notin C_k(P_1)$ means the i and j belong to different clusters in partition P_1. Furthermore, the connectivity matrix $M_{ij}(P_l)$ is defined by equation (4.2.15).

$$M_{ij}(P_l) = \begin{cases} 1 & (i,j) \in C_k(P_l); \\ 0 & \text{otherwise.} \end{cases} \tag{4.2.15}$$

The relationship between element-wise distance $d(P_1, P_2)$ and connectivity matrix $M_{ij}(P_l)$ is $d(P_1, P_2) = |M_{ij}(P_1) - M_{ij}(P_2)| = [M_{ij}(P_1) - M_{ij}(P_2)]^2$, due to $|M_{ij}(P_1) - M_{ij}(P_2)| = 0$ or 1.

Objective function

As the consensus partition P^* minimises the dissimilarity between itself and all the partitions, the objective function based on the distance measure can be expressed by equation (4.2.16).

$$\min_{P^*} J = \frac{1}{s}\sum_{l=1}^{s} d(P_l, P^*) = \frac{1}{s}\sum_{l=1}^{s}\sum_{i,j=1}^{n} \left[M_{ij}(P_l) - M_{ij}(P^*)\right]^2. \tag{4.2.16}$$

If $M_{ij}(P^*)$ is denoted as U_{ij}, which is the solution to this optimisation problem. In addition, the consensus (average) association between two objects i and j is denoted as $\widetilde{M}_{ij} = \frac{1}{s}\sum_{l=1}^{s} M_{ij}(P_l)$. The average squared difference from the consensus association $\widetilde{M}_{ij}$ is denoted by ΔM^2, which is expressed by $\Delta M^2 = \frac{1}{s}\sum_l \sum_{ij} \left[M_{ij}(P_l) - \widetilde{M}_{ij}\right]^2$. The smaller the value of ΔM^2 is, the closer the partitions are. Thus, equation (4.2.16) can be re-written by equation (4.2.17).

$$J = \frac{1}{s}\sum_l \sum_{ij} \left[M_{ij}(P_l) - \widetilde{M}_{ij} + \widetilde{M}_{ij} - U_{ij}\right]^2 = \Delta M^2 + \sum_{ij} \left(\widetilde{M}_{ij} - U_{ij}\right)^2. \tag{4.2.17}$$

Therefore, the objective function for consensus clustering can be further deduced to equation (4.2.18).

$$\min_U \sum_{i,j=1}^{n} \left(\widetilde{M}_{ij} - U_{ij}\right)^2 = \min_U \left\|\widetilde{M} - U\right\|_F^2. \tag{4.2.18}$$

To find the consensus partition becomes investigating the consensus association. As it is an optimisation problem, the constraints of the objective function need to be considered. The solution connectivity matrix U of the consensus clustering problem is characterised by a set of constraints [144]. The number of constraints increases sharply, when the data set becomes larger [144]. The optimisation of the consensus function becomes complicated, when a large amount of constraints are involved.

The large numbers of constraints problem can be solved by clustering indicators $Q = \{0,1\}^{n\times k}$. In each row of Q, there is only one "1" and the other entries must be "0" [145]. For instance, consider a six-object partition with three clusters: $X = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6\}\}$, here $n = 6$, $k = 3$. The connectivity matrix is shown in equation (4.2.19),

$$M_{ij} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2.19}$$

and the cluster indicator is presented by equation (4.2.20).

$$Q_{nk} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.2.20}$$

It is noted that $(QQ^T)_{ij}$ is equal to the inner product between row i of Q and row j of Q. Furthermore, when i and j belong to the same cluster, then row i must be identical to row j, thus $(QQ^T)_{ij} = 1$. Otherwise, if i and j belong to different clusters, the inner product between row i and row j is zero. Hence, the connectivity matrix U can be expressed in equation (4.2.21).

$$U = QQ^T, \quad \text{or} \quad U_{ij} = (QQ^T)_{ij}. \tag{4.2.21}$$

Thus, the consensus clustering problem becomes:

$$\min_{Q} \left\| \widetilde{M} - QQ^T \right\|^2, \tag{4.2.22}$$

where Q is restricted to an indicator matrix. For constraint that in each row of Q, there is only one non-zero element can be expressed as $\left(Q^TQ\right)_{kc} = 0$ for $k = c$. Also $\left(Q^TQ\right)_{kk} = |C_k| = n_k$. In this case, a diagonal matrix is defined as $D = diag\left(Q^TQ\right) = diag\left(n_1, ..., n_k\right)$. D shows the number of elements in cluster C_k and $Q^TQ = D$. For instance, D for the cluster indicator (4.2.20) is shown below:

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $[2, 3, 1]$ is consistent to the example six-object partition with three clusters. Therefore, the optimisation problem is further developed to equation (4.2.23).

$$\min_{Q^TQ=D, Q\geq 0} \left\| \widetilde{M} - QQ^T \right\|^2. \tag{4.2.23}$$

If Q is substituted by a new factor $\widetilde{Q}$, so that:

$$\widetilde{Q} = Q(Q^TQ)^{-1/2}. \tag{4.2.24}$$

Thus,

$$QQ^T = \widetilde{Q}D\widetilde{Q}^T, \quad \widetilde{Q}^T\widetilde{Q} = Q\left(Q^TQ\right)^{-1}Q = I. \tag{4.2.25}$$

The consensus clustering becomes the optimisation, as illustrated in equation (4.2.26)

$$\min_{\widetilde{Q}^T\widetilde{Q}=I, \widetilde{Q}, D\geq 0} \left\| \widetilde{M} - \widetilde{Q}D\widetilde{Q}^T \right\|^2, \quad s.t. \quad D \text{ diagonal} \tag{4.2.26}$$

Suppose $H = \widetilde{Q}$, $S = D$ and the consensus association is replaced by V, CC of a set of partitions is equivalent to a symmetric NNMF problem, as presented in equation (4.2.27).

$$H^*, S^* = \arg\min_{H\geq 0, S\geq 0} \left\| V - HSH^T \right\|^2, \quad s.t. \quad H^TH = I, \tag{4.2.27}$$

where H^* and S^* refer to the optimal solutions, and $U = H^*S^*H^{*T}$ denotes the connectivity matrix of the consensus partition P^*.

Method for solving the objective function

To solve the objective function in equation (4.2.27), the multiplicative update rules are given in [52], as illustrated by equations (4.2.28) and (4.2.29).

$$H_{jk} \leftarrow H_{jk}\sqrt{\frac{(VHS)_{jk}}{(HH^TVHS)_{jk}}}, \tag{4.2.28}$$

$$S_{kc} \leftarrow S_{kc}\sqrt{\frac{(H^TVH)_{kc}}{(H^THSH^TH)_{kc}}}. \tag{4.2.29}$$

The correctness and convergence are similarly proven as presented in the early part of Section 4.2.1 [142]. Finally, H indicates the clustering indicators and S denotes the number of objects in each cluster.

4.2.2 Weighted partition via kernel-based consensus clustering algorithm

WPK-CC was proposed by Vega-pons and Correa-Morris [54]. This algorithm involves an intermediate step, named partition relevance analysis, in which each partition is validated by some validation methods such as variance, connectivity, silhouette width, and Dunn index [54] [146] [147]. A set of weights is assigned to the partitions. Furthermore, the similarity measure between partitions is based on analysing each subset of X and can be further transformed into a Hilbert space according to kernel methods [148]. Thus, the objective function is expressed by the similarity measure between partitions, and solved by the SA.

Basics of kernel methods

The basic idea of kernel methods [148] is to transfer a dataset into a new equivalent higher-dimensional space named feature space, aiming to find a simpler (e.g., linear) relations among these data in such space. The non-linear relationships in the original space between objects could be more easily identified in the feature space. For example, in Figure 4.7, the data in original 2-D space is classified by an irregular curve, which is difficult to be obtained and evaluated. On the other

hand, they are easily classified by a flat surface in the feature space. Each single object is transferred to the equivalent high dimensional data according to some ways, e.g., $(x_1, x_2) \rightarrow (z_1, z_2, z_3)$. Data in higher dimensional space has to face two problems, i.e., curse of dimension and complicated computations [148]. However, in kernel methods, "kernel trick" investigates the relationship between a pair of objects (x_i, x_j) using a kernel function $\kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes a dot product and $\phi(x_i), \phi(x_j)$ stand for two objects in feature space. That is to say, it becomes non-essential to find the certain way (i.e., ϕ) to transfer data from the original space to the feature space. If the kernel function is known, the inner product of two points in the feature space can be ensured. To consider a complex problem in original space is subsequently substituted by analysing the data properties in the feature space (e.g., distance, angle, etc), the only effort need to be done is obtaining the inner products in such space.

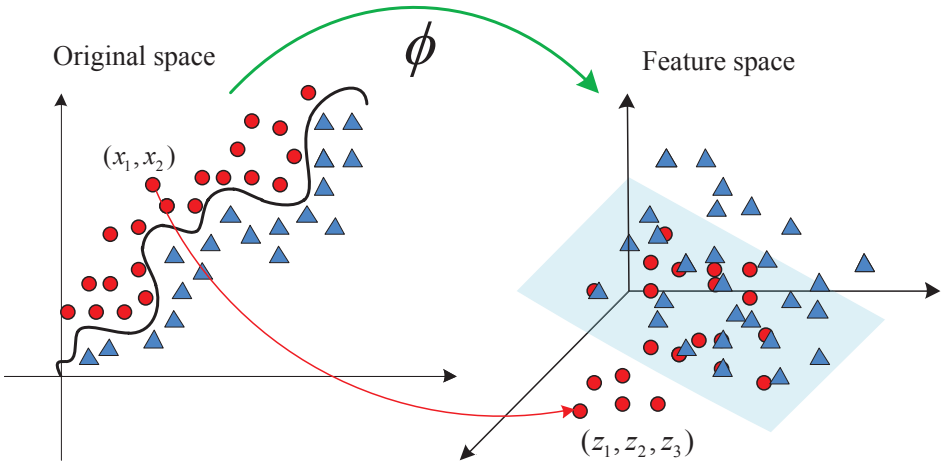


Figure 4.7: Data in the original space and the equivalent feature space

In practice, kernel function κ is represented by an $n \times n$ symmetric kernel matrix K, such that $\mathrm{K_{ij}} = \kappa(x_i, x_j)$. According to Mercer theorem, every positive semi-definite matrix can be regarded as a kernel function [148]. In other words, if the matrix $\mathrm{K_{i,j}}$ is positive semi-definite, function κ is determined to be a kernel function.

Partition relevance analysis

In this step, a set of weights is assigned to the partition set based upon the corresponding property evaluations given by different indexes, which is called property validity indexes (PVIs), and represented by $I = \{I_1, I_2, ...I_{\mathrm{t}}\}$, where t refers to the number of PVIs. In total, there are four validation methods involved in this thesis, which are introduced in Section 4.3. For each $I_{\mathrm{j}} \in I$, the summation of index I_j for each partition P_l is computed, i.e., $A_j = \sum_{l=1}^{s} I_j(P_l)$. A function φ_j, which represents the average I_j for all partitions, is defined by $\varphi_j(P) = {I_j(P)}/{A_j}$, where $\sum_{l=1}^{s} \varphi_j(P_l) = 1, \forall j = 1, ...t$. Therefore, φ_j is related to the distribution function of certain discrete random variable Y_j as illustrated in equation (4.2.30).

$$E(I_j) = E(Y_j) = -\sum\nolimits_{l=1}^{s} \varphi_j(P_l) \log(\varphi_j(P_l)), \tag{4.2.30}$$

where $E(Y_j)$ is the entropy of Y_j [149]. The entropy properties indicate that $E(I_j)$ reaches the maximum value when $\varphi_j(P_1) = ...\varphi_j(P_s)$. According to the continuity property of $E(Y_j)$, the higher values of $E(Y_j)$ imply the stronger likeness among the $I_j(P_l)$ values. Therefore, $E(I_j)$ can be a good measure to represent the property related to the index I_j. Hence, the weight ω_l assigned to each partition P_l, which denotes the relevance of each partition, is expressed in equation (4.2.31) [146].

$$\omega_l = \sum_{j=1}^{t} \left(E\left(I_j\right) \left(1 - \left| I_j(P_l) - \frac{1}{s} A_j \right| \right) \right). \tag{4.2.31}$$

The entropy $E(I_j)$ is used as a measure of the distinction between property index values I_j. The second factor of equation (4.2.31) is an evaluation of $I_j(P_l)$, based on the absolute value of the difference of $I_j(P_l)$ and the mean value $(1/s)A_j$.

Similarity measure between partitions

The WPK-CC focuses on analysing each subset S of X. The basic significance of subset S given P is denoted by $\mu_B(S|P) = {|S|}/{|C|}$, where $S \subseteq C$ for some cluster $C \in P$ and $|\cdot|$ denotes the number of objects. The similarity measure between partitions is defined by $\widetilde{k} : P_X \times P_X \to [0, 1]$, where P_i and P_j are two

instance partitions in P_X, such that:

$$\widetilde{k}\left(P_i, P_j\right) = \frac{k\left(P_i, P_j\right)}{\sqrt{k\left(P_i, P_i\right) k\left(P_j, P_j\right)}}, \tag{4.2.32}$$

where the function $k : P_X \times P_X \rightarrow R+$ is given by:

$$k\left(P_i, P_j\right) = \sum_{S \subseteq X} \delta_S^{P_i} \delta_S^{P_j} \mu\left(S|P_i\right) \mu\left(S|P_j\right),$$

and

$$\delta_S^P = \begin{cases} 1 & if \ \exists C \subseteq P, S \subseteq C \\ 0 & \text{otherwise.} \end{cases}$$

This similarity measure is proven to be positive semi-definite in [146], which means it is a kernel function. Therefore, there exists a map from P_X into a Hilbert space $\mathcal{H}$, $\widetilde{\phi} : P_X \rightarrow \mathcal{H}$ such that $\widetilde{k}\left(P_i, P_j\right) = \left\langle \widetilde{\phi}(P_i), \widetilde{\phi}(P_j) \right\rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the dot product in the Hilbert Space $\mathcal{H}$.

Objective function

Following equation (4.2.1), the objective function of WPK-CC can be obtained as shown in equation (4.2.33).

$$P^* = \arg \max_{P \in P_X} \sum_{l=1}^{s} \omega_l \widetilde{k}(P, P_l). \tag{4.2.33}$$

To solve this problem in the original P_X space is a very difficult combinatorial problem. As the similarity measure $\widetilde{k}$ is positive semi-definite, the equivalent problem in the reproducing kernel hilbert space $\mathcal{H}$ can be undertaken. Subsequently, the objective function in equation (4.2.33) can be converted to equation (4.2.34).

$$\widetilde{\phi}\left(P^*\right) = \arg \max_{\widetilde{\phi}(P) \in \mathcal{H}} \sum_{l=1}^{s} \widetilde{\omega}_l \cdot \left\langle \widetilde{\phi}\left(P\right), \widetilde{\phi}\left(P_l\right) \right\rangle_{\mathcal{H}}, \tag{4.2.34}$$

where $\widetilde{\phi}(P^*)$ and $\widetilde{\omega}_l$ are normalised $\phi(P^*)$ and ω_l, respectively.

In $\mathcal{H}$, the problem can be solved easily, and next step is to obtain the median partition solving the pre-image problem, i.e., it is not necessary to find the exact P^*,

as the exact solution does not have to exist. If the consensus partition is denoted by $\psi = \widetilde{\phi}(P^*)$, the approximate solution $\widehat{P}$ is defined in equation (4.2.35)

$$\widehat{P} = \arg\min_{P \in P_X} \left\| \widetilde{\phi}(P) - \psi \right\|_{\mathcal{H}}^2 \tag{4.2.35}$$

with

$$\left\| \widetilde{\phi}(P) - \psi \right\|_{\mathcal{H}}^2 = \widetilde{k}\left(P, P\right) - 2\sum_{l=1}^{s} \widetilde{\omega}_l \widetilde{k}\left(P, P_i\right) + \sum_{i=1}^{s}\sum_{j=1}^{s} \widetilde{\omega}_i \widetilde{\omega}_j \widetilde{k}\left(P_i, P_j\right).$$

Method for solving the objective function

To solve the pre-image problem, the SA is utilised. In this situation, the states of the system are partitions, and the idea is to start from an initial partition, which is the partition with the best performance (i.e., P_b), through an iterative process, and to obtain a very close partition to the consensus one. The cost function (energy) is the objective function. $P_{\text{neighbour}}$ represents the next state of the current state, i.e., partition P, if one element is moved from the cluster its belonging cluster, to another. According to equation (4.2.35), the notations are listed in Table 4.2, and each notation for the SA is equivalent to that in WPK-CC.

Table 4.2: The SA for solving the objective function of WPK-CC

Notations	SA	WPK-CC
Objective function	$\mathcal{E}$	$\widehat{P} = \arg\min_{P \in P_X} \left\| \widetilde{\phi}(P) - \psi \right\|_{\mathcal{H}}^2$
Initial state	$\mathcal{S}_0$	$P_0 = \arg\min_{P \in P_X} \left\| \widetilde{\phi}(P_b) - \psi \right\|_{\mathcal{H}}^2$
Initial energy	$\mathcal{E}_0$	$F_0 = \left\| \widetilde{\phi}(P_b) - \psi \right\|_{\mathcal{H}}^2$
Next state	$\mathcal{S}'$	$P_{\text{next}} = P_{\text{neighbour}}$
Next energy	$\mathcal{E}'$	$F_{\text{next}} = \left\| \widetilde{\phi}(P_{\text{neighbour}}) - \psi \right\|_{\mathcal{H}}^2$

4.2.3 Information Theory-based consensus clustering algorithm

Generally, this method aims at minimising an information theoretical criterion function using the GA, which was proposed by Luo and Jing [53]. This method uses a metric between partitions based on the entropy.

Dissimilarity measure between partitions

The generalised conditional entropy [53] is used to define the dissimilarity measure between two partitions. The classical notion of entropy and conditional entropy can be generalised using the notion of a generator [150]. Traditionally, Shannon entropy is used for assessing the clustering quality. It can be obtained by a concave, sub-additive function e.g., Shannon entropy $f(p) = -plog(p)$, which allows to generalise the notion of entropy and conditional entropy. Also, other generators are evaluated by Luo, e.g., Gini index $f_{gini}(p) = p - p^2$, among which the generator based on the Shannon entropy outperforms others. Therefore, the evaluation for INT-CC in this thesis is based upon Shannon entropy.

If f is a generator, and $P_1 = \{C_1^1, C_2^1, ..., C_k^1\}$, $P_2 = \{C_1^2, C_2^2, ..., C_k^2\}$ are two partitions for dataset X, where $X = \bigcup_{c_1=1}^{k} C_{c_1}^1$ and $X = \bigcup_{c_2=1}^{k} C_{c_2}^2$. The f-entropy of partition P_1 is defined in equation (4.2.36).

$$E^f(P_1) = f\left(\frac{|C_1^1|}{|X|}\right) + ... + f\left(\frac{|C_k^1|}{|X|}\right), \tag{4.2.36}$$

where $|\cdot|$ denotes the number of objects. The f-impurity of a subset $S \subseteq X$ relative to the partition P_1 [150] is presented in equation (4.2.37).

$$IMP_{P_1}^f(S) = |S|\left(f\left(\frac{|C_1^1 \cap S|}{|S|}\right) + ... + f\left(\frac{|C_k^1 \cap S|}{|S|}\right)\right). \tag{4.2.37}$$

The specific f-impurity of a subset $S \subseteq X$ relative to the partition P_1 is the value [150]:

$$imf_{P_1}^f(S) = f\left(\frac{|C_1^1 \cap S|}{|S|}\right) + ... + f\left(\frac{|C_k^1 \cap S|}{|S|}\right). \tag{4.2.38}$$

Therefore, the f-entropy of P_1 relative to P_2 is defined by equation (4.2.39).

$$E^f\left(P_1 \,|P_2\right) = \sum_{c_2=1}^{k} \frac{\left|C_{c_2}^2\right|}{|X|} imp_{P_1}^f\left(C_{c_2}^2\right) = \frac{1}{|X|}\sum_{c_2=1}^{k}\left|C_{c_2}^2\right|\sum_{c_1=1}^{k} f\left(\frac{\left|C_{c_1}^1 \cap C_{c_2}^2\right|}{\left|C_{c_2}^2\right|}\right). \tag{4.2.39}$$

The dissimilarity between two partitions is given in equation (4.2.40).

$$d^f\left(P_1, P_2\right) = E^f\left(P_1 \,|P_2\right) + E^f\left(P_2 \,|P_1\right). \tag{4.2.40}$$

Here, it is easy to find that when P_1 is close to P_2, giving many elements in common in their classes, both $E^f\left(P_1 \,|P_2\right)$ and $E^f\left(P_2 \,|P_1\right)$ are close to 0.

Objective function

According to the dissimilarity measure, the consensus function is summarised as shown in equation (4.2.41).

$$P^* = \arg\min_{P \in P_X} \sum_{l=1}^{s} d^f\left(P, P_l\right) = \arg\min_{P \in P_X} \sum_{l=1}^{s} \left(E^f\left(P\,|P_l\right) + E^f\left(P_l\,|P\right)\right). \tag{4.2.41}$$

Method for solving the objective function

The GA has been employed to solve the objective function (4.2.3). As mentioned in Section 2.3.2, the chromosomal population is represented by a set of random integer vectors using values between 1 and k, i.e., $P = \{P_1, ..., P_q, ..., P_u\}$, which contains u chromosomes and $P_q \in P$. When it reaches $IMax$, it will terminate. The best fitness regarding to the chromosome in the population is P^*. Otherwise, a new population will be generated. Meanwhile, in this part of the work, the genetic operators used include the ERW, one-point crossover, and bit-flip mutation [53], which have been systemically illustrated in Section 2.3.2.

4.3 Validation Methods

Validation methods concern the quality of clusters obtained by an algorithm. Validation methods aim to analyse obtained clustering results and provide an evaluation feedback. In addition, they are also utilised to compare the performance of different algorithms and the impact of different parameter settings for a specific algorithm.

Validation methods mainly contain two categories, namely internal and external validation. The internal measurements need to be used, aiming to explore the intrinsic properties of the data (e.g., real dataset). If the true classes of a set of data are available, external validation techniques can be used, providing an objective way of evaluation (e.g., test dataset) [32] [146].

In this research, both internal and external validation methods are utilised to evaluate the performance of clustering algorithms. In addition, the internal

validation methods are also applied in PVIs for WPK-CC.

4.3.1 Internal validation

1. Variance (VI) measures the compactness of the clusters in the partition, as illustrated in equation (4.3.1).

$$VI = \frac{1}{\sqrt{\frac{1}{n}\sum_{C_i \in P}\sum_{x_j \in C_i} dist\left(x_j, \eta_i\right)}}. \tag{4.3.1}$$

   where $dist$ is the distance function and η_i is the centroid of cluster C_i;

2. Connectivity (CI) evaluates the degree of connectedness of clusters in the partition by investigating the numbers of neighbours for one object that they all belong to the same cluster, as shown in equation (4.3.2).

$$CI = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{nc} v_{i,j}}, \tag{4.3.2}$$

   where

$$v_{i,j} = \begin{cases} \frac{1}{j} & if \quad x_i \in C_h \cap nn(i,j) \in C_h = \emptyset \\ 0 & \text{otherwise} \end{cases},$$

   here, $nn(i,j)$ is the jth nearest neighbour of the object x_i. nc is the number of neighbours.

3. Silhouette width (SI) shows how well an object lies within the correspondence cluster that is denoted by equation (4.3.3).

$$SI = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i - a_i}{\max(b_i, a_i)}, \tag{4.3.3}$$

   where a_i represents the average distance between the object x_i and the rest of objects in its cluster, and b_i is the average distance between x_i and the objects in its closer cluster.

4. Dunn index (DI) examines the ratio between the smallest inter-cluster distance and the largest intra-cluster distance, as defined in equation (4.3.4).

$$DI = \min_{C_i \in P}\left\{\min_{C_j \in P}\frac{dist(C_i, C_j)}{\max_{C_h \in P} diam(C_h)}\right\}, \tag{4.3.4}$$

where $diam(C_h)$ is the maximum intra-cluster distance within cluster C_h and $dist(C_i, C_j)$ is the minimal distance between pairs of objects x_a and x_b such that $x_a \in C_i$ and $x_b \in C_j$.

4.3.2 External validation

1. Purity is the percentage of the total number of objects that were correctly clustered [35]. The purity of a single cluster C_j is defined as the fraction of objects in the cluster that belong to the dominant class contained within that cluster:
$$P(C'_i, C_j) = \frac{1}{n_j} \max_i \{N_{ij}\},$$
where N_{ij} denotes the size of the intersection $|C_i' \cap C_j|$ between the class C_i' and cluster C_j; n_j is the numbers of data in cluster C_j. The overall purity of a clustering is defined as the sum of the individual cluster purities, weighted by the size of each cluster, which is illustrated in equation (4.3.5).
$$P(C', C) = \sum_{j=1}^{k} \frac{n_j}{n} P(C', C_j) \tag{4.3.5}$$

2. F-measure is based on the recall and precision criteria, which have been introduced in Section 3.4.3. Each cluster is regarded as the result of a query operation, and each underlying class is considered as the target set of documents for the query [151]. F-measure is given by the harmonic mean of precision and recall level from a pair of clustered results and natural class. Similarly, from the cluster analysis point of view, the F-measure is defined by equation (4.3.6).
$$F_{\text{ij}} = \frac{2 \cdot r_{ij} \cdot p_{ij}}{r_{ij} + p_{ij}}, \tag{4.3.6}$$
where r_{ij} and p_{ij} stand for recall and precision levels, respectively. The overall F-measure is obtained by equation (4.3.7).
$$F(C', C) = \sum_{i=1}^{k'} \frac{n_i'}{n} \max_j \{F_{ij}\}, \tag{4.3.7}$$
where k' is the number of class and n_i' denotes number of data in class C_i'.

4.4 Simulatition Studies of the Clustering Algorithms

4.4.1 Data collections

In this thesis, seven datasets have been selected for the clustering algorithm evaluation purpose, including three sample datasets as shown in Table 4.3 and four document repositories as presented in Table 4.4. In each table, "n" represents the number of object, and "m" is the number of features for each object. Class means the number of the internal or underlying classification (cluster) for each dataset, and objects-per-class shows the original number of data objects in each cluster. In other words, the information of classes and objects-per-class are obtained from the datasets themselves, which can be regarded as some external information for evaluating the performance of clustering algorithms.

Table 4.3: Three sample datasets from UCI machine learning repository

Dataset	Description	n	m	classes	objects per class
Iris[1]	Types of iris plant	150	4	3	50-50-50
Wine[2]	Types of wine	178	13	3	59-71-48
Glass[3]	Types of glass	214	10	7	87-70-17-76-13-9-29

Table 4.4: Four document repositories

Dataset	Description	n	m	classes	documents per class
bbcsports[4]	Sports news articles from BBC	737	4,613	5	101-124-265-147-100
bbc[5]	Articles from BBC	2,225	9,635	5	510-386-417-511-401
TDT2-6[6]	Subset of TDT2	6,523	36,771	6	1844-1828-1222-811-411-407
PSD	Power substation document corpus	136,735	700,083	6	41223-21482-32115- -10101-22583-9231

[1] Available from https://archive.ics.uci.edu/ml/datasets/Iris
[2] Available from https://archive.ics.uci.edu/ml/datasets/Wine
[3] Available from https://archive.ics.uci.edu/ml/datasets/Glass+Identification

4.4.2 Implementation configuration

In this section, the experiment starts from a generation step. K-means with different initialisations is utilised to generate 20 partitions. CC algorithms are successively applied to the partitions, aiming to find the consensus partition. Four internal and two external validation methods introduced in Section 4.3 are employed for algorithms evaluation. Besides, the four internal validation methods are examined to assign relevant weights to each partitions in WPK-CC. In the final results, each outcome of the internal validation method is normalised to $[0, 1]$ for comparison purpose. The experiment working process is illustrated in Figure 4.8. In addition, the parameter settings of each CC algorithm are almost based on the original papers, as shown in Table 4.5. The termination of each iteration for CC algorithms are set to be $IMax = 10000$. Moreover, the results of k-means for each

Table 4.5: The experiment parameter settings of each algorithm

CC algorithms	Parameter settings
NNMF-CC	$H^T H = I$
WPK-CC	$T = 100$,$\beta = 0.98$
INT-CC	$u = 50$, $P_{\text{m}} = 0.1$, $P_{\text{c}} = 0.8$

dataset is obtained from the average clustering results of 20 partitions, while the results of CC algorithms are computed by the average of 10 independent algorithm runs.

4.4.3 Simulation results

Table 4.6 to Table 4.8 show the validated results for the sample datasets. Table 4.9 to Table 4.12 give the validated results for text datasets. Figure 4.9 and Figure 4.10 reveal the visual illustration of external validated results of each clustering algorithm for all the datasets. From the tables, CC algorithms have better performance than the average results of k-means according to the internal validation

[4]Available from http://mlg.ucd.ie/datasets/bbc.html
[5]Available from http://mlg.ucd.ie/datasets/bbc.html
[6]Available from http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

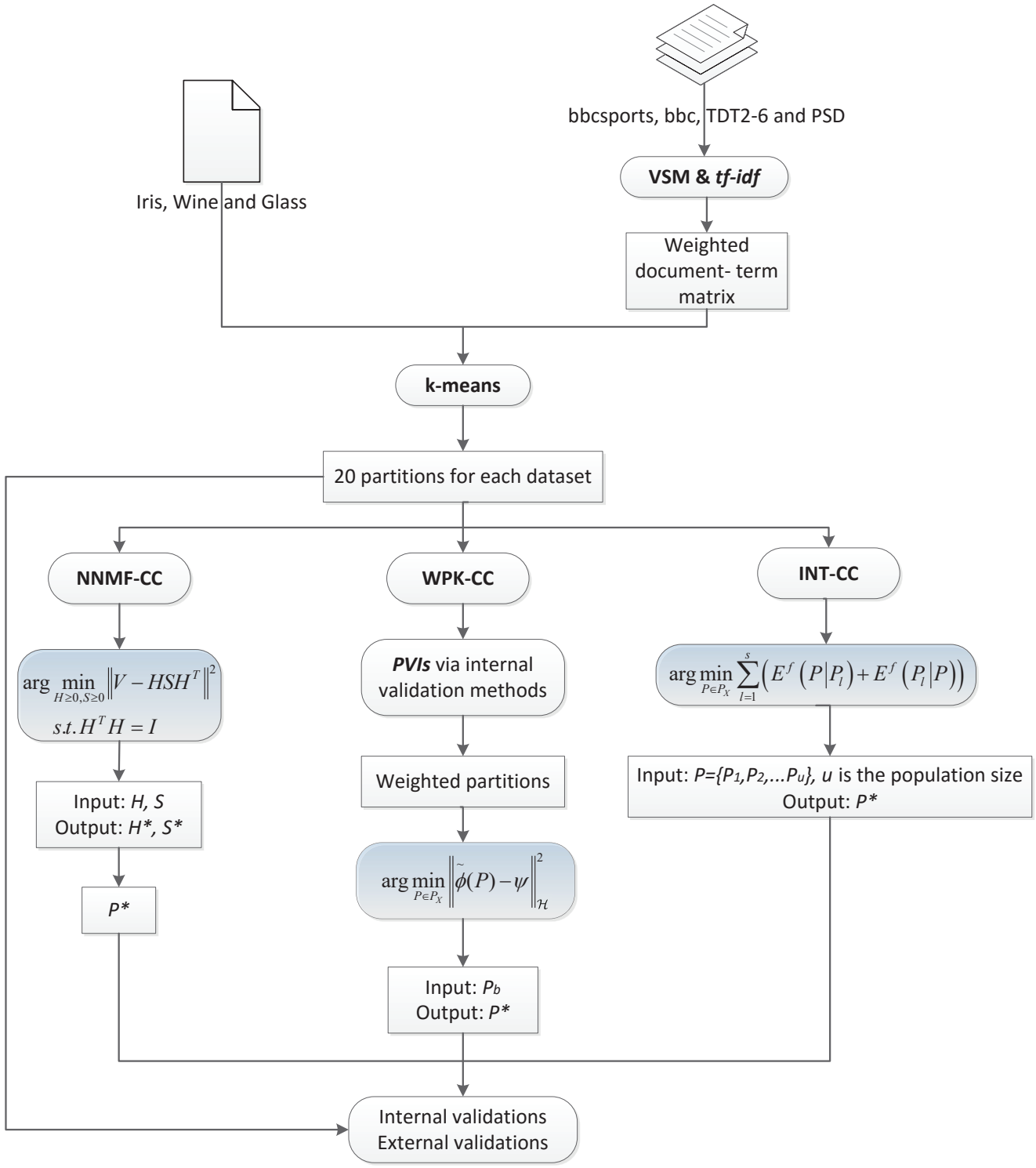


Figure 4.8: The experiment working process

Table 4.6: Validated results for the Iris dataset

	VI	CI	SI	DI	purity	F-measure
k-means	0.852	0.728	0.944	0.731	0.743	0.777
NNMF-CC	0.905	0.863	1	0.931	0.840	0.832
WPK-CC	1	1	0.988	1	0.852	0.899
INT-CC	0.943	0.371	0.911	0.905	0.790	0.784

Table 4.7: Validated results for the Wine dataset

	VI	CI	SI	DI	purity	F-measure
k-means	1	0.851	0.809	0.211	0.376	0.527
NNMF-CC	0.850	1	1	0.842	0.427	0.542
WPK-CC	0.925	0.821	0.915	1	0.431	0.610
INT-CC	0.950	0.776	0.872	0.579	0.393	0.533

Table 4.8: Validated results for the Glass dataset

	VI	CI	SI	DI	purity	F-measure
k-means	0.929	0.869	0.894	0.895	0.381	0.538
NNMF-CC	0.810	0.934	1	1	0.492	0.621
WPK-CC	0.857	0.918	0.979	0.737	0.572	0.602
INT-CC	1	1	0.936	0.421	0.451	0.644

methods. However, it seems that it is difficult to compare each CC algorithm with the internal validations, as we have no background knowledge to determine which properties should be considered as a good measure. The intrinsic cluster of a set of data can influence the internal validations. As a result, a clustering result with a higher value on one validated result cannot be ensured that it has good validated results on other validations. Also, the users may aim to maximum one of the characteristics of the data, e.g., if a user wants to obtain a clustering result with good connectivity value for the data, the partitions that highlight the connectivity value can be assigned by a high weight. In this case, with no background knowledge, the internal validations can partially reflect the performance of clustering algorithm. Therefore, the internal validations are more likely to be implemented as a reference to assign weights to each partition. The decision is based on the occurrence, i.e., a small weight is assigned to a partition that holds an apparently different clustering result to others, as it might be wrongly generated from generation step; if a partition has an average behaviour, indicating a general pattern of the data, a higher weight will be assigned to such partition. It is consistent to the partition relevance analysis discussed in Section 4.2.2.

For the external validation results, it can be concluded that all the CC algorithms have significant improvements, compared to the single clustering algorithm as shown in Figure 4.9 and Figure 4.10. Among them, WPK-CC outperforms NNMF-CC and INT-CC in all purity levels and most F-measures. These can be explained by WPK-CC involving PVIs to each partition. As some of the partitions from generation step might be wrongly generated, which can be regarded as noise partitions. The partition relevance analysis is a method to assign a small weight to these noise partitions. It avoids a simple average of the set of partitions producing a worse clustering result. For sample datasets, NNMF-CC outperforms INT-CC in most datasets. However, NNMF-CC and INT-CC have comparable performances in text datasets.

From the dataset point of view, the sample dataset, especially the iris, has much more remarkable purity level and F-measure with the implementation of every clustering algorithm. For selected sample datasets, we can find $n >> m$, which

Table 4.9: Validated results for the bbcsport document repository

	VI	CI	SI	DI	purity	F-measure
k-means	0.750	0.833	0.978	0.727	0.433	0.601
NNMF-CC	0.846	0.909	0.844	0.8636	0.453	0.663
WPK-CC	1	1	1	0.955	0.512	0.701
INT-CC	0.692	0.879	0.978	1	0.456	0.654

Table 4.10: Validated results for the bbc document repository

	VI	CI	SI	DI	purity	F-measure
k-means	0.355	1	0.524	0.226	0.279	0.439
NNMF-CC	0.367	0.8605	0.524	1	0.390	0.447
WPK-CC	0.402	0.721	0.667	0.538	0.415	0.546
INT-CC	1	0.628	1	0.774	0.403	0.440

Table 4.11: Validated results for the TDT2-6 document repository

	VI	CI	SI	DI	purity	F-measure
k-means	0.471	0.670	0.791	0.129	0.421	0.452
NNMF-CC	1	1	0.938	0.613	0.512	0.471
WPK-CC	0.447	0.550	0.896	1	0.534	0.666
INT-CC	0.435	0.540	1	0.710	0.492	0.463

Table 4.12: Validated results for the PSD

	VI	CI	SI	DI	purity	F-measure
k-means	0.854	0.470	0.550	0.533	0.317	0.378
NNMF-CC	1	0.330	0.776	0.689	0.353	0.421
WPK-CC	0.537	1	0.875	0.978	0.384	0.435
INT-CC	0.512	0.510	1	1	0.332	0.353

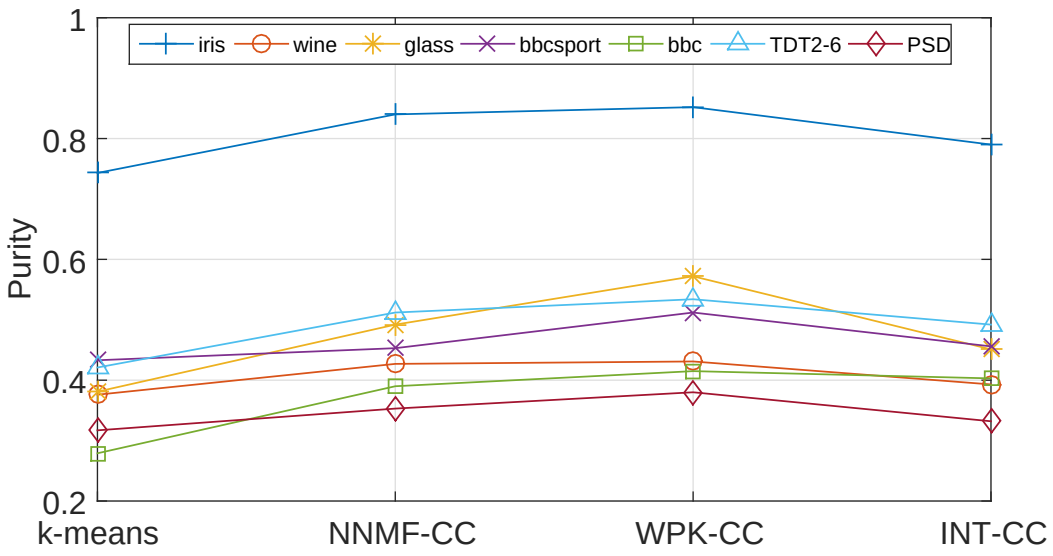


Figure 4.9: Purity of each clustering algorithm for different datasets

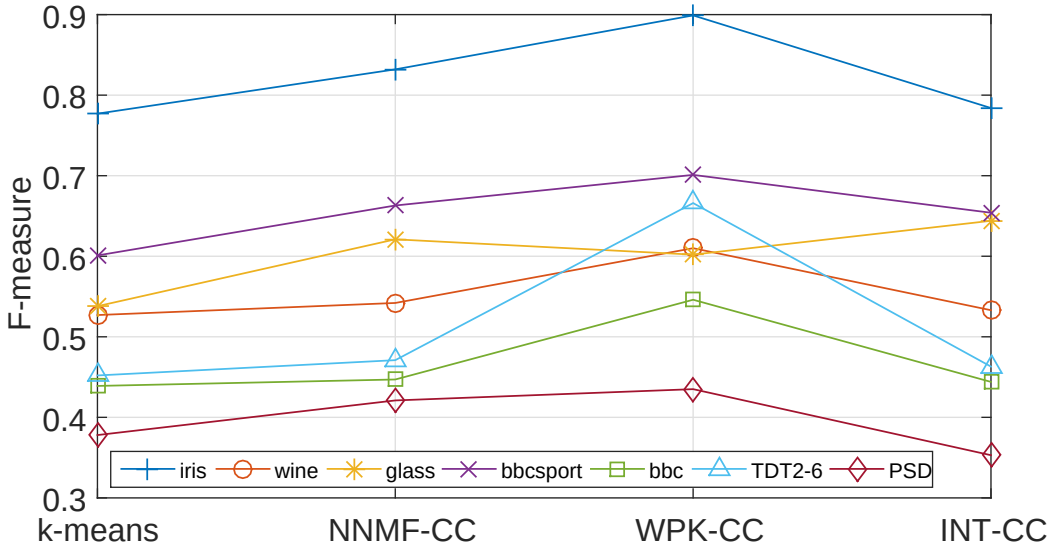


Figure 4.10: F-measure of each clustering algorithm for different datasets

means the number of objects are much more than its attributes. It is ensured that the key features, which contribute to the clustering, have more significant meanings. They directly indicate the properties for a single object. As in iris dataset, there are four features, namely sepal length, sepal width, petal length, and petal width. In a document dataset, the number of documents are normally much smaller than selected features (terms), i.e., $n << m$. The features are weighted terms or words in the document corpus. The document dataset cannot avoid the influence of noise from irrelevant words or incorrect words. Therefore, the poor performance of document clustering are expected. The main focus of this chapter is to evaluate the accuracy of the clustering results obtained from different clustering algorithms. Nevertheless, it is worth mentioning that the iteration pattern of each CC algorithm is different, resulting in significant impacts for the computational time. NNMF-CC updates the cluster indicators in each generation of the iterations, aiming to find an optima cluster indicator that directly indicates cluster of each object. Either the similarity measure or dissimilar measure between partitions, the construction of the objective function of both WPK-CC and INT-CC are based on analysis of the subsets of the dataset. If the size of a dataset is large, the computational time of the iterations increases dramatically. Moreover, the objective function of WPK-CC is solved by SA, in which a new solution in each generation is obtained by a single cluster label variation. As a result, the CC using WPK-CC has much slower convergence speed, compared with NNMF-CC. In contrast, the GA used in INT-CC provides a broader searching strategy based on the genetic operators. Although the evaluation performance of INT-CC is worse than WPK-CC, as it is not able to ensure the global optimum can be achieved, it is less time consuming than WPK-CC.

4.5 Summary

Document clustering methods are utilised to group documents in a corpus into a number of sensible clusters, in which each of the cluster has a smaller size. Also, they reduce the complexity of searching relevant document regarding to a user's query, which can be employed to the IR system. Most document clustering studies

only focused on single clustering algorithms upon different document repositories. Also, many proposed CC algorithms received better results, compared with the traditional CC algorithms. These algorithms were all originally designed for sample datasets, which have much smaller features than document datasets. This chapter provided a set of performance evaluations of three novel CC algorithms on both sample datasets and document datasets. It has verified that the competitive aspect of CC also applies to the cluster analysis of a document repository.

This chapter started from a systemic introduction of cluster analysis, including the definition of clustering with its common applications, the generic document clustering procedure. In addition, the general types of clustering algorithms were presented, in which k-means was highlighted in this thesis, devoting to the generation step of CC. Three CC algorithms were demonstrated in detail, followed by presenting a set of clustering results validation methods. Finally, a set of simulation studies were designed for the performance evaluation of clustering algorithms. The results showed that WPK-CC outperforms the other CC algorithms on every dataset. Inspired by the characteristics of the GA, more attentions have been received that the best performing WPK-CC can be improved by employing the GA to some extent. This part of the work is illustrated in the next chapter.

Chapter 5

Consensus Clustering for Ontology-embedded Document Repository of Power Substations using Kernel-based Method

This chapter presents a novel CC approach for PSD and contributes to the intangible AM of power substations. This chapter is organised as follows: Section 5.1 firstly presents the motivation that the background knowledge of a document repository should be involved in the document representation for clustering, followed by a theoretical comparison between the two meta-heuristics, i.e., the SA and the GA. Subsequently, the proposed method for the PSD representation is demonstrated in Section 5.2. It begins with introducing the SONT-based VSM in detail. Subsequently, the term mutual information and a Manhalanobis distance based on the correlation factor matrix are illustrated for the PSD representation. Finally, three simulation studies are designed for comparison purpose in Section 5.3, resulting in an optimal solution for PSD CC using the proposed WPK-CC combined with GA (WPKGA).

5.1 Introduction

From Chapter 4, it is concluded that CC for a document repository consists of the text pre-processing, implementing clustering algorithms, consensus functions and methods of solving consensus functions (e.g., meta-heuristics). From Section 4.1.3 to Section 2.3, many efforts have been carried out on the algorithms for clustering. It is a logical and continuous consideration that how the document pre-processing and suitable method to solve the consensus functions can influence the performance of document CC.

As a text repository normally contains many documents, documents with a similar topic are always filled with synonyms or hyponyms. Even if the CC methods are applied, the result of document CC cannot reveal the intrinsic document topics. As a consequence, the clustering result cannot be improved significantly. The document data pre-processing for clustering in this research is inspired by [51], in which a Wordnet-based distance measure, was proposed. Wordnet is utilised as the background knowledge, but it aims to generate a term mutual information matrix. Subsequently, a new data model is obtained by combining the consideration of correlation between terms and the traditional VSM. In this study, SONT is implemented for representing the document dataset, i.e., PSD. This procedure operates after tokenisation and linguistic processing. Section 5.2 explains the mechanism of the SONT-based distance measure for the PSD clustering in detail.

In Section 2.3, two meta-heuristics, i.e., the SA and the GA have been introduced and applied to solve the consensus functions, i.e., WPK-CC and INT-CC, respectively. Both of the SA and the GA are stochastic search algorithms that begin with random initialisation. The SA can be considered to be equivalent to the GA when the population size is only one. In other words, the SA aims to find a single approximate solution. The GA works on a population, which consists of a number of individuals or chromosomes, seeking a set of optimal results, of which the best individual is the final solution. Therefore, the SA can be regarded as a special type of the GA only including mutation and no crossover. The parameters in the SA, i.e., initial temperature and the temperature decreasing rule, are more likely to be determined empirically. For example, if the rate of temperature deducing is low, a

global optimum could be returned. However, in this case, the convergence speed becomes extremely slow. On the other hand, if the temperature deduces quickly, the SA will converge fast and always returns a local optimum. It is difficult to find a balanced "β" in the SA, which concerns both accuracy and efficiency. In document CC, the method to find a state neighbour in the SA is as similar as the mutation operation in the GA.

Apart from the basic steps of the GA, i.e., selection, crossover and mutation, normal optimisation problem via the GA also includes two steps, i.e., encoding and decoding. In other words, the initialised random individuals should be encoded into binary strings, formulating the chromosomes, and afterwards, the obtained "best" chromosome will be decoded into the corresponding form of result according to the problem. In a clustering application as mentioned in Section 2.3.2, the chromosomes are integer strings with fixed lengths, of which each component represents a cluster label for a single document. Therefore, utilising the GA to deal with clustering problem is more straightforward compared to other algorithms, as the form of solutions in a clustering problem is similar to a chromosome in the GA. In addition, the GA returns a set of solutions rather than a single solution, permitting more chances to obtain a more approximate solution. In this case, the GA is a better approach to solve the consensus functions than the SA or other optimisation methods. Moreover, the GA contains various genetic operators and parameters in each step of approximation as mentioned in in Section 2.3.2, which allows a more thorough optimisation for a CC application.

5.2 Document Repository of Power Substations with Ontology-based Vector Space Model and Term Mutual Information

From the linguistics point of view, some studies have verified that there exist mutual relations between the terms in a document-term vector [152] [153]. Therefore, it is essential to take these term relationships into consideration rather

than simply using a traditional term-based VSM [154] [155].

As mentioned in Section 3.1.1, SONT is programmed according to the context of power substations only, which is the first ontology model specifically defined regarding the domain of power substations, containing synonyms and hyponyms of each concepts. Compared with WordNet, SONT is a more specific ontology model representing a particular domain with no ambiguousness. Therefore, each concept in SONT and its synonym set and hyponym set represent a specific meaning. These relationship among terms in a document-term vector is called background knowledge of the terms. In the document CC, we aim to integrate the background knowledge to the traditional term-based VSM.

Following the notation defined in Section 2.2.2, if a set of documents D is denoted by $D = \{d_1, d_2, ..., d_n\}$, which is based on the term-based VSM, each document d_j can be represented by $t_{ji} = \{t_{j1}, t_{j2}, ..., t_{jm}\}$ or $\{t_1, t_2, ...t_m\}$ (as terms in the vocabulary extracted from the corresponding document collection are fixed). The corresponding term weight is denoted by $d_j = \{\omega_{j1}, \omega_{j2}, ..., \omega_{jm}\}$, and ω_{ji} is the weight of term t_{ji}. The weight can either be the term frequency, i.e., $tf_j = \{tf_{j1}, tf_{j2}, ..., tf_{jm}\}$ or based on tf-idf. Typically, in a document-term vector, this method begins with examining whether a term t_{i_1} is semantically correlated to the other term t_{i_2} with SONT. In other words, this step aims to check the synonym and hyponym set of each term, and an indicator, i.e., $\delta_{i_1 i_2}$, is defined to present the semantic information between two terms. If t_{i_2} is a synonym or hyponym of t_{i_1}, $\delta_{i_1 i_2}$ will be set to a coefficient, otherwise, $\delta_{i_1 i_2}$ will be set to zero. Thus, with the $\delta_{i_1 i_2}$ embedded, the term frequency can be modified by equation (5.2.1).

$$\widetilde{\omega}_{ji_1} = \omega_{ji_1} + \sum_{\substack{i_2=1 \\ i_2 \neq i_1}}^{m} \delta_{i_1 i_2} \omega_{ji_2}. \tag{5.2.1}$$

It is noticed that the synonym set and the hyponym set with regards to a same term in a specific domain describe a similar topic. For instance, transformer fault diagnosis and transformer fault analysis are similar concepts, and they have a same hyponym, i.e., fault detection. The hierarchy among the concepts regarding the cluster analysis becomes less important than it is for QE in an ODSE in this research, as we only aim to obtain a set of document clusters, and documents in the same

cluster discusses similar power substation-related topics. For this purpose, the semantic information indicator $\delta_{i_1 i_2}$ can be set to a fixed value. An example has been illustrated as follows: there are three documents, i.e., $\{d_1, d_2, d_3\}$. d_1 and d_2 mainly introduce the topic of PTFD, while d_3 is about the power transformer design. The term frequency, which is based on the traditional VSM, is given by Table 5.1 (a). As introduced in Section 3.1.1, diagnosis, assessment and detection are semantically related to each other. According to equation (5.2.1) with $\delta_{i_1 i_2} = 0.8$, the term-based VSM can be transformed into the SONT-based VSM as illustrated in Table 5.1 (b). The Euclidean distance based on the traditional VSM between two documents is $dis(d_1, d_2) = \sqrt{(d_1 - d_2)(d_1 - d_2)^T} = \sqrt{\sum_{i=1}^{m} (\omega_{1i} - \omega_{2i})^2}$. If the term frequency (tf_{ji}) is the weight measure, the distance between d_1 and d_2, d_2 and d_3, d_1 and d_3 are 14.8997, 15.0333 and 15.4919, respectively. In the SONT-based VSM, the above distances are updated to 8.1142, 23.8546 and 19.1833, respectively. The influence of involving the semantic relations for VSM is remarkable, as the distance between d_1 and d_2 decreases and the distances between d_1 and d_3, d_2 and d_3 increases significantly. As a consequence, d_1 and d_2 have more chances to be clustered into a same group in the cluster analysis, and d_3 will be assigned to another cluster.

Although the PSD contains different topics and the documents can be divided into several categories in terms of these topics, they are all under the context of power substations. According to Harris distributional hypothesis, the words or terms occur in the same contexts tend to have similar meanings [156]. Thus, there exist some syntactic surface dependencies between a pair of terms simultaneously occur in the document repository. The syntactic surface dependencies are defined as term mutual information (TMI) and computed by the cosine similarity. The TMI between term t_1 and term t_2 is illustrated in equation (5.2.2), in which the similarity of each pair of terms, i.e., mutual information matrix (MIM), in a given document repository can be computed as expressed in equation (5.2.3).

$$TMI_{t_1 t_2} = \frac{\sum_{j=1}^{n} \widetilde{\omega}_{j1} \widetilde{\omega}_{j2}}{\sqrt{\sum_{j=1}^{n} \widetilde{\omega}_{j1}^2} \cdot \sqrt{\sum_{j=1}^{n} \widetilde{\omega}_{j2}^2}}. \tag{5.2.2}$$

(a) Term-based VSM

	d_1	d_2	d_3
power	5	8	6
transformer	10	12	10
fault	5	2	0
diagnosis	10	0	0
detection	0	8	0
assessment	0	6	0
magnetic	0	0	5
circuit	0	0	5
optimisation	0	0	8

$\rightarrow$

(b) SONT-based VSM

	d_1	d_2	d_3
power	5	8	6
transformer	10	12	10
fault	5	2	0
diagnosis	10	11.2	0
detection	8	12.8	0
assessment	8	12.4	0
magnetic	0	0	5
circuit	0	0	5
optimisation	0	0	8

Table 5.1: The comparison between the term-based VSM and the SONT-based VSM

$$MIM = \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1i} & \dots & \sigma_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{j1} & \dots & \sigma_{ji} & \dots & \sigma_{jm} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{m1} & \dots & \sigma_{mi} & \dots & \sigma_{mm} \end{bmatrix}, \tag{5.2.3}$$

where σ_{ji} denotes the mutual similarity between term t_j and t_i. It is noticed that the similarity of (t_j, t_i) is equivalent as (t_i, t_j). Thus, MIM is symmetric. In addition, there is no difference between the same term, i.e., (t_i, t_i). Therefore, MIM can be normalised, which is illustrated by equation (5.2.4).

$$M = \begin{bmatrix} 1 & \dots & \sigma_{i1} & \dots & \sigma_{j1} & \dots & \sigma_{m1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{i1} & \dots & 1 & \dots & \sigma_{ji} & \dots & \sigma_{mi} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{j1} & \dots & \sigma_{ji} & \dots & 1 & \dots & \sigma_{mj} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \dots & \sigma_{mi} & \dots & \sigma_{mj} & \dots & 1 \end{bmatrix} \tag{5.2.4}$$

The elements in M are all greater than or equal to zero so that M is symmetric

positive semidefinite [157]. It is similar to the equations (4.2.24) and (4.2.25) as mentioned in Section 4.2.1, where the mutual information matrix can be decomposed by an orthogonal matrix A and a diagonal matrix D, as presented by equation (5.2.5).

$$M = ADA^T = A\sqrt{D}\sqrt{D}A^T = (A\sqrt{D})(A\sqrt{D})^T = BB^T, \tag{5.2.5}$$

where B is defined as the correlation factor matrix, and $B = A\sqrt{D}$. According to the Euclidean distance based on the term frequency between two documents, the distance with the term mutual information matrix can be denoted by equation (5.2.6).

$$md(d_1, d_2) = \sqrt{(d_1 - d_2)M(d_1 - d_2)^T}. \tag{5.2.6}$$

This distance measure refers to a Mahalanobis distance, where the matrix M is defined as the dimensions correlation coefficient appearing in Mahalanobis distance [158]. It is noticed that equation (5.2.6) turns into a Euclidean distance, if M is the identity matrix. That is to say, the inner relations between terms are not considered. According to equation (5.2.5), the distance $md(d_1, d_2)$ can be modified as follows:

$$\begin{aligned} md(d_1, d_2) =& \sqrt{(d_1 - d_2)M(d_1 - d_2)^T} \\ =& \sqrt{(d_1 - d_2)BB^T(d_1 - d_2)^T} \\ =& \sqrt{(d_1 - d_2)BB^T(d_1 - d_2)^T} \\ =& \sqrt{((d_1 - d_2)B)((d_1 - d_2)B)^T}, \\ =& \sqrt{(d_1B - d_2B)(d_1B - d_2B)^T} \\ =& \sqrt{(\widehat{d_1} - \widehat{d_2})(\widehat{d_1} - \widehat{d_2})^T} \end{aligned} \tag{5.2.7}$$

where $\widehat{d_1} = d_1B$ and $\widehat{d_2} = d_2B$. Thus, the Mahalanobis distance between d_1 and d_2 is equivalent to the Euclidean distance between $\widehat{d_1}$ and $\widehat{d_2}$. The difference between these two distances is that Mahalanobis distance involves the patterns of correlation in the dataset, while Euclidean distance does not.

5.3 Simulation Studies

In this thesis, three simulation studies are designed and applied to the PSD. Briefly, Section 5.3.1 presents the case study 1 that discusses the PSD clustering with background knowledge embedded. Case study 2 in Section 5.3.2 compares the original WPK-CC with the SA and WPK-CC combined with the GA. Besides, case study 3 in Section 5.3.3 analyses the impact of the genetic operators of GA to the PSD CC. The basic procedure is similar to the simulation studies presented in Section 4.4, starting from a generation step, which is used to obtain 20 partitions by k-means with random initialisations. As mentioned before, the PSD generally contains six topics, and this study only focuses on $k = 6$. Purity is the only validation method considered in this section, as the internal validation methods, i.e., VI, CI, SI and DI, refer to reference knowledge to assign each partition with a relevant weight according to the partition relevance analysis. Also, the difference between the underlying classes of a dataset and the resulted clustering result can reflect the performance of the algorithm directly. In addition, case studies designed in Section 4.4.2 only focus on the great significance of CC algorithms on both sample datasets and text datasets. Considering the size of the PSD (number of documents), i.e., 136,735, an improvement of 0.001 in purity assigns more than 100 documents into the correct cluster. In this case, an extra decimal space is added, and the CC result in purity with the percentage form in order to indicate more accurate results is presented. Figure 5.1 illustrates the overall flowchart for each simulation study, and the numbers by each arrow represent the corresponding cases. Specially, all the genetic operators are tested in the case study 3, and the WPKGA in the case study 2 only utilises the operators denoted by italic and underline. The details of each case study are demonstrated in the following three subsections.

5.3.1 The impact of ontology model for document representation

This case study focuses on the influence of involving the background knowledge to the PSD with SONT-based VSM, which is named as the modified PSD and denoted by MPSD. The Euclidean distance measure between two documents are

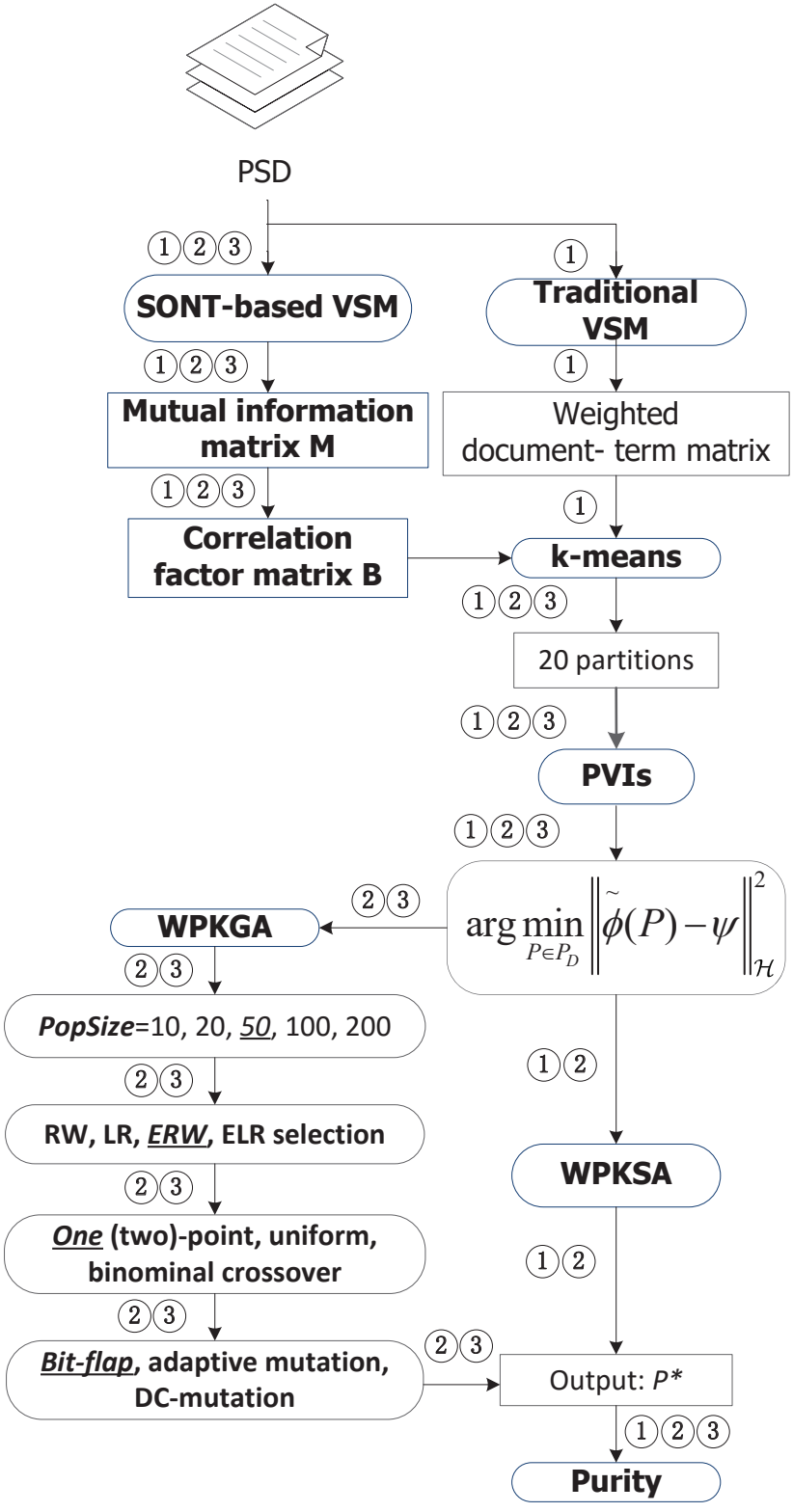


Figure 5.1: The flowchart of each simulation study, and the WPKGA in the case study 2 only concerns the *italic* terms

transformed to a Mahalanobis distance, which takes the correlation between each pair of terms into account. It is more reasonable to involve the term mutual information than ignoring the inter relations among terms. According to equation (4.1.1) and equation (5.2.7), the distance between a document and its cluster centroid can be denoted by equation (5.3.1).

$$md(d_j, C_l) = \sqrt{(d_j B - C_l B)(d_j B - C_l B)^T} = \sqrt{(\widehat{d_j} - \widehat{C_l})(\widehat{d_j} - \widehat{C_l})^T}, \quad (5.3.1)$$

where $\widehat{d_j}$ is the transferred document vector and $\widehat{C_l}$ is the l_{th}cluster's centroid. Thus, the Euclidean distance between a document and its cluster centroid is illustrated by equation (5.3.2).

$$d(\widehat{d_j}, \widehat{C_l}) = \sqrt{(\widehat{d_j} - \widehat{C_l})(\widehat{d_j} - \widehat{C_l})^T}. \quad (5.3.2)$$

Thus, the standard k-means can be applied to the MPSD. The case configuration follows the settings in Section 4.4.2. The results have been shown in Table 5.2. The comparison between clustering algorithms on PSD is consistent with that discussed in Section 4.4.3. WPK-CC outperforms the other clustering algorithms. With the SONT embedded PSD, each algorithm reaches an improved purity. It is noted that the purities of MPSD with CC algorithms have more significant improvement than that with average k-means. It may be due to that the noise partitions or wrongly generated partitions cannot be completely avoided during the generation step. In this case, compared with the PSD, the growth rate of using MPSD may not be significant, when using the standard k-means. However, the PSD contains 136,735 documents, which means that even an improvement of 0.33% for MPSD with k-means assigns more than 400 documents correctly into the underlying clusters compared with the normal PSD without considering the relationship between terms. From this point, each CC algorithm has remarkable performance, as the improvement of best performing WPK-CC with MPSD is 1.72%. Thus, more than 2300 documents are correctly clustered compared with applying WPK-CC to the original PSD. The comparison among different CC algorithms is consistent with that introduced and discussed in Section 4.4.3.

Anyhow, the clustering results have been improved by involving the term mutual information. SONT is implemented to add background information to the original

dataset. As a consequence, there are more relevant documents being assigned into the same cluster so that the accuracy of the cluster result is improved significantly.

Table 5.2: Purity analysis of each clustering algorithm on the PSD and the MPSD

Purity (%) Algorithm / Dataset	k-means	NNMF-CC	WPK-CC	INT-CC
PSD	31.72	35.29	38.43	33.19
MPSD	32.05	37.28	40.15	34.66
Increment (%)	0.33	1.99	1.72	1.47

5.3.2 The improvement of GA-embeded kernel-based consensus clustering algorithm

This case study aims to analyse the performance of consensus function with suitable meta-heuristics, i.e., the SA and the GA. The performance of each CC algorithm has been evaluated in previous sections. Compared with the other two CC algorithms, WPK-CC is advantageous with the aid of the partition relevance analysis, which demonstrates competative results on both PSD and MPSD. The similarity measure between two partitions in WPK-CC is proven to be a kernel function so that the optimal partition in the partition space can be obtained from its Hilbert space. The SA is applied to find the optimal solution of the consensus function. The principles of the SA and the GA have been introduced in Section 2.3.1 and 2.3.2, respectively. As a special type of GA, the SA has been theoretically compared to the GA in Section 5.1. Also, in INT-CC, the entropy generator has been proven that it has the minimal error rate for selected sample datasets CC [53]. However, when compared with other CC algorithms, it performs worse than WPK-CC. It is caused by either the method of obtaining the dissimilarity measure or the method of solving the consensus function. Considering all the aspects above, this case study aims to implement the GA to solve the consensus function of WPK-CC (denoted by WPKGA), and the result is compared to the original SA-embedded

WPK-CC (denoted by WPKSA). In addition, the SONT-based VSM with term mutual information PSD, i.e., MPSD, is the test dataset in the rest of the case studies.

WPKSA aims to find the minimum of the objective function as shown in equation (4.2.35). It is noted that the first and third term of equation (4.2.35) are fixed numbers, as $\tilde{k}(P, P)$ denotes the similarity measure between an intermediate solution partition and itself, and $\sum_{i=1}^{s}\sum_{j=1}^{s}\tilde{\omega}_i\tilde{\omega}_j\tilde{k}\left(P_i, P_j\right)$ shows the sum of weighted similarity measures among 20 partitions generated from k-means. Therefore, the objective function of WPKGA is equivalent to the search of the maximum of the second term of equation (4.2.35), i.e., $2\sum_{i=1}^{s}\tilde{\omega}_i\tilde{k}\left(P, P_i\right)$. In any case, both WPKSA and WPKGA are applied to seek the optimum of the objective function, aiming to find a clustering result, which can be regarded as an optimum depending on a relevant evaluation. Thus, both WPKSA and WPKGA follows the generation step, in which the k-means is employed to generate 20 partitions. Subsequently, each of the algorithm aims to solve the optimisation problem, i.e., equation (4.2.35). WPKSA seeks the neighbour of the best performing partition until the algorithm converges, operating as GAs with only mutation. In contrast, WPKGA starts from initialising fixed numbers of chromosomes (a population), followed by selection, crossover and mutation operations. Meanwhile, in the simulation studies presented in Section 4.4 and Section 5.3.1, each algorithm is terminated by a fixed number of iteration step. As stochastic algorithms, the SA and the GA involve iteration processes before obtaining the results and the termination condition is not guaranteed to know. In general, there are three termination conditions for the GA, i.e., a fixed number of generation is reached; an upper limit to the number of evaluations of the fitness function is reached; and the difference between the maximum fitness and the average fitness in a population is not significant [159]. In previous case studies, only the first termination criterion is followed. The second termination criterion requires some background knowledge about the problem to allow the estimation of a reasonable maximum fitness, as the purity is required to be as higher as possible. The third is an adaptive form, as it considers the nature of the iterations. Thus, several termination conditions for the SA and the GA are defined and presented in Table 5.3.

Table 5.3: Termination status of the SA and the GA related algorithms

	Termination status	
SA	$IMax = 10000$	$Energy\ \mathcal{E} = 0$
GA		$\frac{\lvert F_{\text{avg}} - F_{\text{best}} \rvert}{\lvert F_{\text{avg}} \rvert} \leq \varepsilon$

The iterations WPKSA terminates, when it reaches to the pre-defined maximum $IMax$ generation or the objective function $Energy\ \mathcal{E}$ becomes zero. For the GA related algorithms, they terminate when either the iteration satisfies the $IMax$ or the difference between the best objective value F_{best} and the corresponding average fitness or purity in the population, i.e., F_{avg} is not more significant. For instance, if a threshold $\varepsilon = 0.001$ is set, 40.15% and 40.12% are the best purity and the average purity, respectively. It means the best chromosome only correctly assigns around 40 documents more than the average performance of the entire population. When comparing 40 with the total number of the PSD, the selection of ε is reasonable.

Figure 5.2 illustrates the generations against purity of WPKSA and WPKGA on MPSD. Both the average purity and the maximum purity for each generation of WPKGA are presented. It is noted that WPKSA starts the iterations from a higher purity, i.e., 25.88%, and WPKGA start from apparently lower purities, i.e., less than 15%. As mentioned in Section 4.2.2, the states in WPKSA are partitions, and the idea is to start from an initial partition, which is the partition with the best performance (i.e., P_b), through an iterative process, and to obtain a very close partition to the consensus one. On the other hand, the initial population of WPKGA is randomly generated. Therefore, the difference between WPKSA and WPKGA in Figure 5.2 at the starting point of the iterations is predicted. In addition, WPKGA converges at less than 8000 iterative steps, and WPKSA is terminated by the pre-defined maximum generation, i.e., 10000. At each state of WPKSA, only one object changes to its neighbour. The similarity measure between partitions in WPK-CC is based on the intersection of objects. If the size of a dataset is too large, i.e., the length of the clustering result is very long, each generation will generate very little improvements, resulting in an extremely slow convergence speed. In contrast, WPKGA operates with a population of chromosomes, and the mechanisms of the

GA guarantee the variety of results. Therefore, WPKGA is expected to be more suitable to handle clustering problem with a large size of document repository.

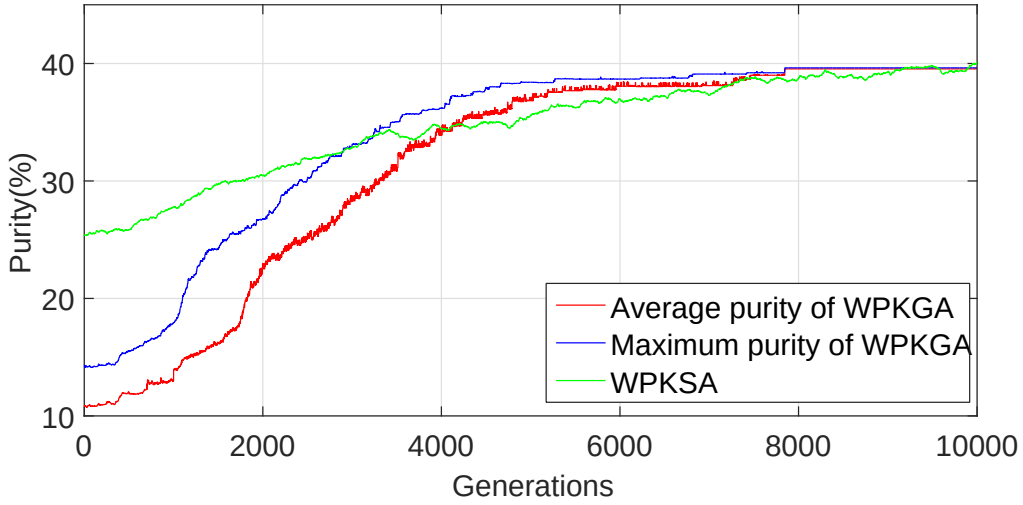


Figure 5.2: The comparison among the average purity of WPKGA of the entire population, the maximum purity of WPKGA in the population, and the purity of WPKSA

Table 5.4 presents the termination condition of WPKSA and WPKGA. The "max" and "average" denote the maximum and the average purity of 50 resultant chromosomes in each generation of WPKGA, respectively. Both of the final maximum purity (39.63%) and the average purity (39.60%) of the resultant GA population perform better than that using NNMF-CC (37.28%) and INT-CC (34.66%), and the maximum purity (39.63%) is smaller than the result of WPKSA (40.15%). It indicates that WPKGA is more competitive to WPKSA, when comparing WPKGA with NNMF-CC and INT-CC. In practice, researchers always take the best performing chromosome as the final optimum. This case study shows that the GA embedded WPK-CC outperforms other advanced CC algorithms. Compared to the SA, the GA is composed of many mechanisms in each operation of GA. It is worthy of study to examine the influence of each mechanism and how they can contribute to the clustering issues. In addition, this case study focuses on both the average performance of a population and the most significant chromosome in a population, which indicates the general performance of WPKGA is competitive

enough to other CC algorithms. In the following simulation studies, only the best performing chromosome in the final population is observed.

Table 5.4: Results of the PSD CC using WPKSA and WPKGA

	WPKSA	WPKGA	
		max	average
Purity (%)	40.15	39.63	39.60
Termination condition	$IMax$	7847	7847

5.3.3 The performance evaluation of genetic operators for proposed consensus clustering algorithm

This simulation study evaluates the impact of different mechanisms of the genetic operators and parameter settings of the GA, and the optimal settings for each operator of the GA are implemented to solve the MPSD CC. The performance of each setting is also evaluated by purity. The initial operator and parameter configurations are consistent to the settings given in Section 4.2.3 and Section 4.4.2. Each comparison is carried out based on the optimal settings from previous procedure of the GA. Firstly, there are five different *PopSize* selected for the performance evaluation, i.e., *PopSize* $= 10, 20, 50, 100$ and 200. The generations of each WPKGA with different *PopSize* are illustrated in Figure 5.3. Table 5.5 shows different *PopSize* of WPKGA and their corresponding convergence condition, i.e., generations and purities. For *PopSize* $= 10, 20, 50$ and 100, results show that the increase of *PopSize* significantly improves the purity of the clustering result, while the convergence speed decreases. In previous cases, *PopSize* was 50, which produced a better result (39.63%) compared with *PopSize* of 10 (32.23%) and 20 (36.56%). When *PopSize* increases to 100, it results in a significant higher purity (41.72%) and slower convergence speed (8712) than other *PopSize*. When *PopSize* is 200, WPKGA is terminated by $IMax$ and the purity (40.24%) is slightly worse than *PopSize* of 100. In addition, when *PopSize* changes from 10 to 50, remarkable improvement of the purity is obtained along with the increment of *PopSize*. In

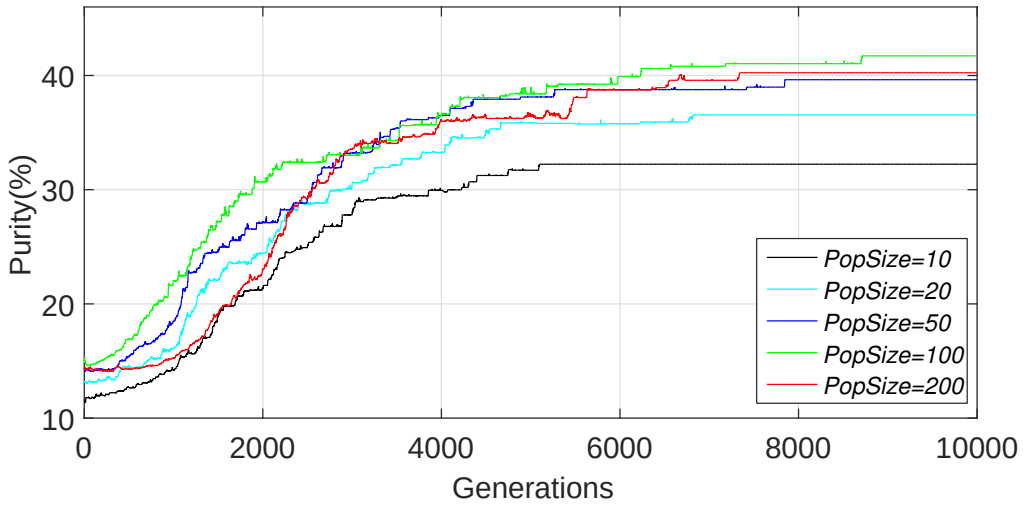


Figure 5.3: WPKGA for the MPSD CC with different *PopSize*

contrast, when *PopSize* increases from 50 to 200, the improvement of purity is less significant. It can be concluded that if *PopSize* is too small, population diversity become very low. As a result, in each iteration, the obtained new population has less chance to perform crossover or mutation and lead to the algorithm pre-mature and produces weak result with a faster convergence speed. On the contrary, if *PopSize* is large, the fitness level doesn't increase much and may become even worse. Thus, *PopSize* $= 100$ is selected to be an optimal setting for the first step of WPKGA on the MPSD.

Table 5.5: Purity and convergence of WPKGA with different *PopSize* on the MPSD

	10	20	50	100	200
Purity (%)	32.23	36.56	39.63	41.72	40.24
Termination Condition	5096	6819	7847	8712	$IMax$

Figure 5.3 demonstrates the iterations and convergence of WPKGA on MPSD for consensus clustering. To improve the readability, make it easier for analysis, and indicate statistical significance, the error bars are employed to present the results for the remaining comparisons. The purpose of using error bars to demonstrate the CC results is to present the average performance of each GA mechanism and parameter

setting in 10 WPKGA independent runs on the MPSD. Also, the generations of WPKGA before its convergence condition are eliminated, which makes the results simpler and conciser to be read, compared with the original "generation against purity" plot. In addition, the standard deviation (SD) of 10 WPKGA runs of a GA mechanism or parameter setting is also shown in the same figure so that the stabilisation of each mechanism can be evaluated as well. Briefly, error bars illustrate the variability of data, which can be used to either present the error or uncertainty of measurements or indicate the SD of a set of data, calculated by ten independent runs of the algorithm. In our case, the average purity indicates the strength and the weakness of CC algorithms. The method of presenting the results in the model of error bars is consistent with Figure 5.3, where "Y" axis stands for the purity and "X" axis shows the generations. The top ends of each bar denote the average purity of 10 WPKGA runs, locating on the average convergence generations, and the red line segments represent the SD of each 10 WPKGA runs. A long error bar indicates that the concentration of the values obtained is low (small SD), while a short error bar represents that the resultant values have low concentration (large SD). In addition, each bar represents one mechanism or one of the GA parameter settings.

Figure 5.4 shows the iterations of WPKGA for MPSD CC with four selection mechanisms, i.e., ERW, ELR, RW, and LR, when *PopSize* is 100. It is worth mentioning that there are different ways to draw error bars, e.g., standard error of the mean or SD. The SD quantifies how much the values vary from one another. The standard error of the mean is the SD of the sample mean's estimate of a population mean. In a standard error of the mean error bar, the upper error (U) should be maximum value minus the average value, and for the lower error (L), it is denoted by the value of the average value minus the minimum value. In our case, we only consider the variation of the results, and both U and L represent the SD. The results show that WPKGA with ERW (7187 generations) terminates faster than other selection mechanisms and the LR has the slowest convergence speed, which is terminated by $IMax = 10000$. As ERW keeps and directly copies the best 2% chromosomes to the next generation without crossover and mutation operations, it

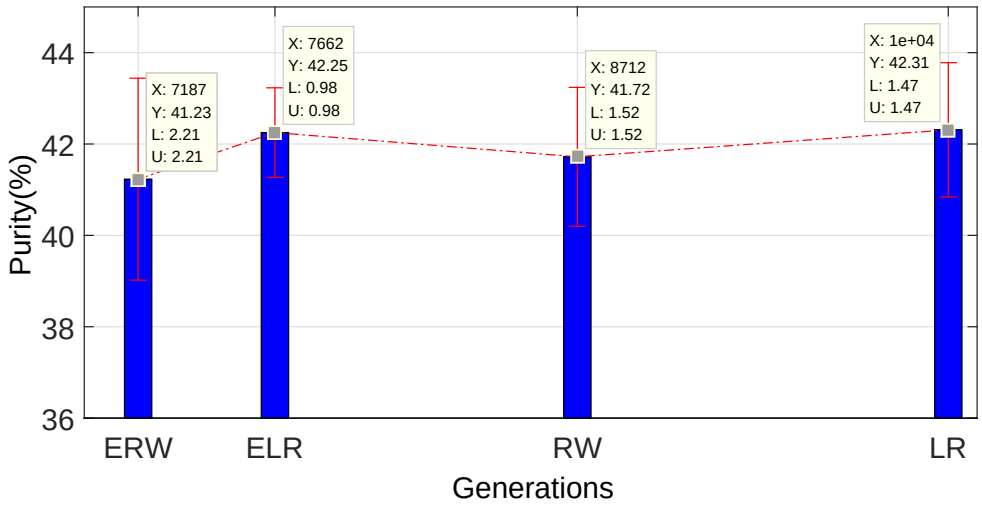


Figure 5.4: WPKGA for the MPSD CC with different selection mechanisms, and *PopSize* is 100, where "X" represents the generation at convergence for each mechanism; "Y" shows the average purity; "L" & "U" denote the lower and upper SD

avoids the disruption of best chromosomes. The LR has the highest purity (42.31%), which is comparable to ELR (42.25%). Both of these two mechanisms perform better than RW (41.72%), and ERW (41.23%) has the smallest purity amongst the four selection mechanisms. The SD of ERW (2.21%) is larger than others that means the result of WPKGA with ERW is less stable than the others. Either the good CC results or the bad results with lower purity are obtained. For the mechanism of LR, it overcomes the limitation of RW, which is if the best chromosome has an outstanding fitness, the other chromosomes will have few chances to be selected. Although LR mechanism sacrifices the convergence speed, it keeps the diversity of population and avoids the algorithm from pre-mature, resulting in a local minimum. ELR combines the advantages of LR and ERW, producing a remarkable CC result with an acceptable termination status for MPSD. Therefore, ELR is identified as an optimal mechanism of the simulations.

Figure 5.5 illustrates the impact of different crossover rates, i.e., $0.6, 0.7, 0.8$ and 0.9 in WPKGA. Generally, the crossover rate should be high (e.g., 0.9) so that most individuals can be involved into the genetic process. The results show that at

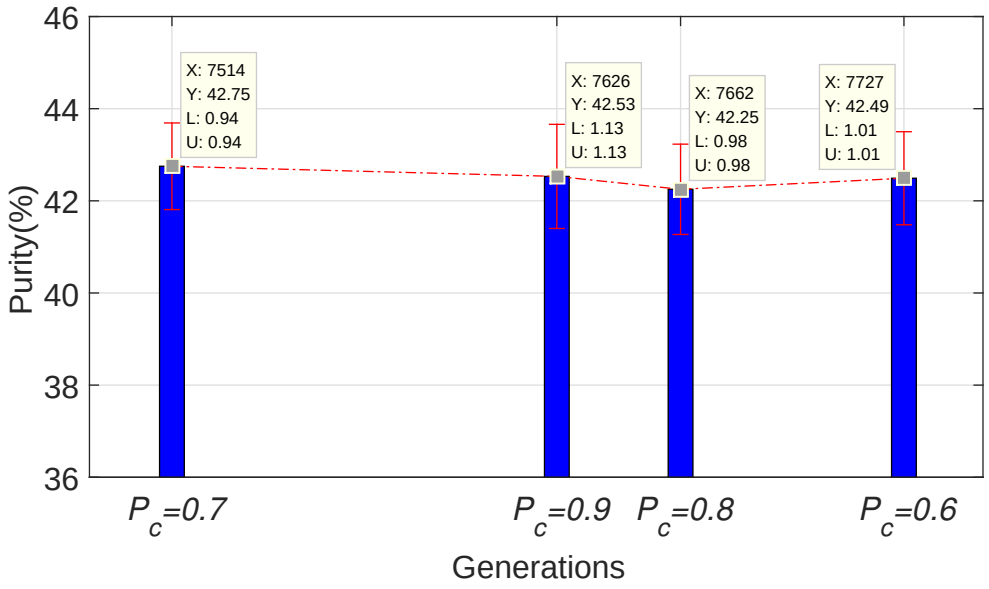


Figure 5.5: WPKGA for the MPSD CC with different crossover rates, when *PopSize* is 100, and ELR selection applies

$P_c = 0.7$, the purity obtained is the best (42.75%), and it converges faster than other crossover rates with the smallest SD. The purity and SD of $P_c = 0.6$ and $P_c = 0.9$ are similar, i.e., (42.49%) and (1.01%) v.s. (42.53%) and (1.13%). However, $P_c = 0.6$ has a slower convergence speed (7727) than $P_c = 0.9$ (7626). Although $P_c = 0.8$ produces the worst purity than other crossover rates, the stability (SD$= 0.98\%$) is better than $P_c = 0.6$ and $P_c = 0.9$. It can be concluded that the crossover rate is not as higher as better, it also depends on the specific problem or the other parameter settings. Thus, for the remaining of the comparisons, P_c is set to be 0.7.

Figure 5.6 presents different crossover types, i.e., one-point crossover, two-point crossover, binominal crossover with probability (P) of 0.1, 0.2, 0.3, 0.4 and 0.5. When P equals to 0.5 of binominal crossover, it also refers to the uniform crossover. Here, it is not necessary to consider the cases, where $P > 0.5$, due to the symmetry of binominal crossover. The results illustrate that one-point crossover has the worst performance on purity (42.75%), convergence speed (7514) and SD (0.94%) and two-point crossover performs only slightly better than one point, i.e., purity (42.67%), convergence speed (7401) and standard deviation (0.93%).

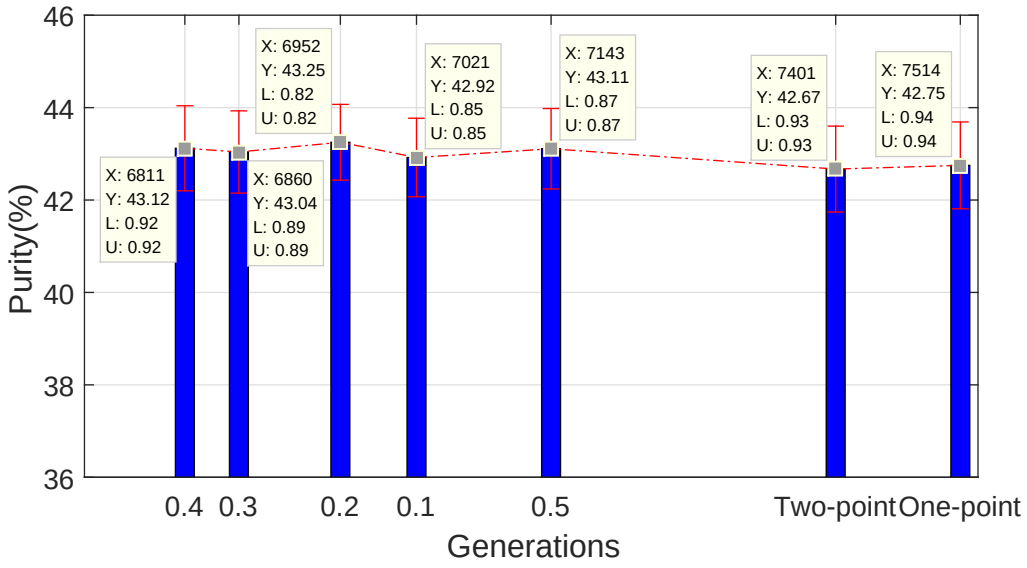


Figure 5.6: WPKGA for the MPSD CC with different crossover mechanisms, when *PopSize* is 100; ELR selection applies; and P_c is 0.7. The numeric values on the "X" axis represent the probability of binominal crossover

Binominal crossovers with all probabilities have competitive purities. Among them, binominal crossover with $P = 0.1$ has the smallest purity (42.92%) with an average SD (0.85%), and the uniform crossover converges slower (7143) than the other binominal crossover probabilities. The purity of $P = 0.2$ (43.25%) is the best performing binominal crossover with the smallest SD (0.82%) and the convergence speed (6952) is only slower than $P = 0.3$ (6811) and $P = 0.4$ (6860). Binominal crossovers allow the offspring chromosomes to search all possibilities of recombining those different genes in parents. Considering all the aspects discussed above, binominal crossover with the probability of 0.2 is selected as the optimal crossover mechanism.

Figure 5.7 shows WPKGA for the MPSD CC with different mutation rates (P_m), i.e., $0.05, 0.1, 0.15$ and 0.2. Among all the mutation rates, $P_m = 0.15$ outperforms the other rates on purity (43.96%) with a middle level of SD (0.93%). Although $P_m = 0.05$ converges faster than other P_m, it has a distinct large SD (1.31%) and a low purity (41.38%), which is only slightly larger than *PopSize*$= 50$ in Figure 5.3 without any advanced settings as mentioned before. $P_m = 0.2$ reaches to the

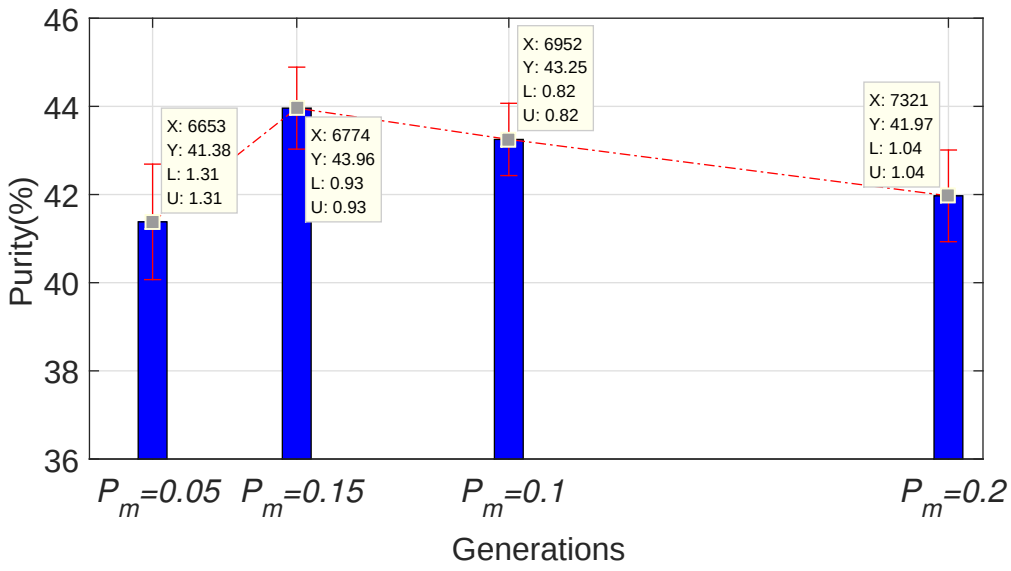


Figure 5.7: WPKGA for the MPSD CC with different mutation rates, when *PopSize* is 100; ELR selection; and crossover rate is 0.7 with a binominal crossover probability of 0.2

convergence status slower than the others (7321). The results show that mutation rate also cannot be either too high or too low. If P_m is set high, the search will turn into a primitive random search. If P_m is too low, the diversity of population can not be ensured. As it is difficult to ensure the mutation rate, adaptive mutation is implemented. In each generations, the mutation rate will reset automatically depending on the property of the population. Only the maximum and minimum mutation rates are defined pre-defined before running WPKGA.

Figure 5.8 compares different mutation types. $P_m = 0.15$ is set to the bit-flip mutation. For the adaptive mutation, the rate range is specified as $[0.05, 0.2]$. DC-mutation represents the mutation mechanism for document clustering based on the adaptive mutation. The results show that DC-mutation outperforms adaptive mutation and bit-flip mutation on each aspect, i.e., purity (45.74%), convergence speed (6531) and SD (0.81%). Among them, bit-flip mutation performs the worst with purity (43.96%), convergence speed (6774) and SD (0.93%).

DC-mutation is specially designed for document clustering. Since the chromosomes for document clustering are not based on binary coding, the selected

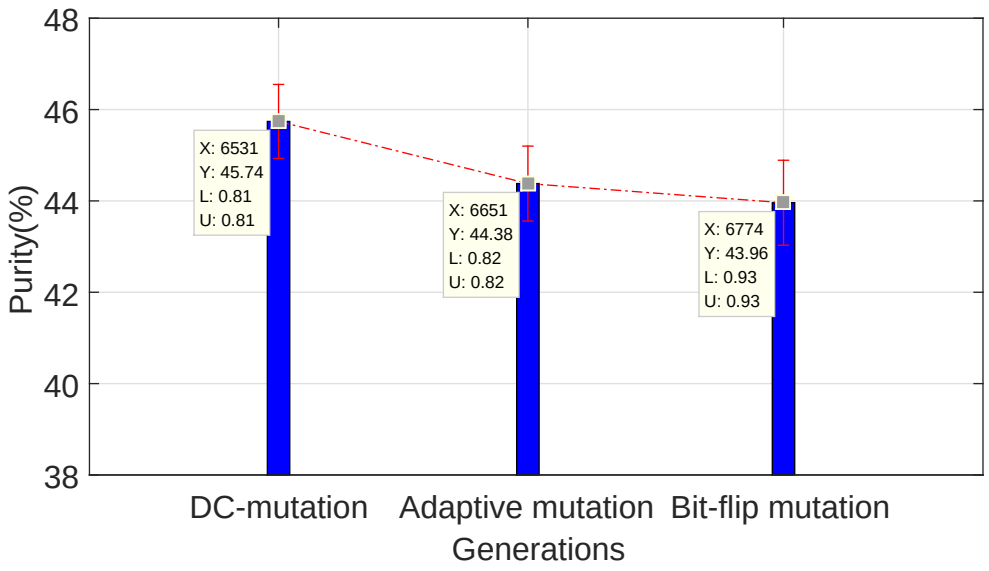


Figure 5.8: WPKGA for the MPSD CC with different mutation mechanisms, when *PopSize* is 100; ELR selection; crossover rate is 0.7 with a binominal crossover probability of 0.2; mutation rate is 0.15; and the range of the adaptive mutation rate is $[0.05, 0.2]$

point is able to mutate to any other integers based on the cluster number. It is ensured that the GA tends to be convergence, when the difference between the average performance and the best performing chromosome is not significant. In other words, most chromosomes tend to be as same as possible (The Hamming distances among chromosomes decrease) along with the generations. Therefore, the alleles of each chromosome are analysed to produce a vector with the same length of the population. The cluster number with the most occurrence may not be the most proper gene, which can be mutated to, however, it is reasonable to assign it with a higher probability. In this case, LR is implemented so that the cluster number with the most occurrence has more opportunity to be mutated to. LR also guarantees that the gene can mutate to any statuses with probabilities, and the probabilities are neither too large nor too small.

5.4 Summary

This chapter proposed an improved PSD clustering method, concerning three aspects, i.e., background knowledge-involved PSD representation, an advantageous CC algorithm according to the relevant comparison, and significant improvements to the original WPK-CC with the GA. SONT has been applied to add background knowledge to the PSD, concerning the semantic correlation among terms. Subsequently, the original PSD was modified by the term mutual information and the correlation factor matrix was obtained. Thus, the modified PSD enhanced the performances of each clustering algorithm.

Meanwhile, WPKGA has been compared with the original WPKSA. The GA was proven to be more suitable to handle document clustering issues. As the GA contains various mechanisms and parameters in each operation, comparisons of these operators have been evaluated and discussed. It can be concluded that the PSD clustering has been improved significantly, which is able to provide power engineers a more accurate and convenient way for important text files mining. In this case, the clustered PSD is capable of improving the efficiency of browsing and navigating relevant documents, which further benefits the AM of power substations in terms of intangible assets.

Chapter 6

Ontology-based Bayesian Networks for Power Transformer Fault Diagnosis

This chapter presents an ontology-based PTFD framework with the ability of uncertainty reasoning. Firstly, a brief introduction of BNs and ontology in PTFD is presented in Section 6.1. The algorithms of knowledge integration, including IPFP, C-IPFP, E-IPFP and D-IPFP, are employed to refine an existing BN with probabilistic constraints, which are given in Section 6.2. Subsequently, Section 6.3 demonstrates an ontology-based BN in detail, which consists of structural translation, CPTs construction, the probabilistic knowledge representation in OWL, and the framework implementation. Finally, a simple PTFD example implemented by ontology-based BNs is illustrated in Section 6.4.

6.1 Introduction

In practice, the information or knowledge acquired by the experts is not guaranteed certain, completed and consistent. The aim of managing the uncertainties in a system is to solve the issues such as how to represent uncertain information and knowledge, how to combine the uncertain knowledge, and how to make a decision

according to the uncertain knowledge [160]. The knowledge base is regarded as the core of an intelligent system, e.g., expert-system for PTFD, in which it contains incomplete or imprecise information. The standard logic is not able to handle these information. Therefore, the ways to describe and infer the uncertainty knowledge in a system become crucial issues.

BNs are tools, which combine probability statistics and graph theory, and perform uncertainty reasoning and data analysis in a complex domain [55] [161]. Basically, the network structure in BNs qualitatively illustrates the relationships among variables in a domain and the CPTs for each node in the network quantitatively describe mutual relations of the nodes [161]. In addition, the chain rule in the BN defines a joint probability distribution (JPD), which is regarded as the knowledge base of the problem model [162]. In previous studies, it has been verified that IEEE/IEC dissolved gas analysis coding scheme can be directly transferred into a BN, and the drawbacks of missing codes can be overcome [13] [163]. Also, BNs analyse the structure of the domain based on the the principle of probability statistics, and decomposes the complex JPD into a series of simple modules. Therefore, the complexity of the probability inference decreases and knowledge becomes easier to be acquired [164]. In PTFD systems, BNs can be used to capture the probabilistic relations between fault types and the corresponding symptoms. If any symptoms are determined, a BN is able to calculate the probabilities of all the potential fault types. The details of BNs are introduced in Section 2.4.

In Section 1.3.3, the application of ontology-based systems for PTFD has been briefly introduced. The limitation of such system is that a statement in an ontology is either true or false. It can be explained by an example, i.e., ontology can give that A belongs to B, but it is not able to show what the percentage A belongs to B is. The current ontology language is unable to quantify the possibility that a statement is true. In a PTFD system, various of uncertainties are contains. A fault symptom always reflects different fault types. The fault type, which is the main cause of that fault symptom is always examined and judge by probability, i.e., the degree of closeness of A to B. In such case, the uncertain information can not be directly represented in an ontology system. Inspired by Tang [13] and Ding's [56] studies,

this chapter aims to complement ontology with the ability of uncertainty reasoning using BN, and utilise this framework to handle the problem of PTFD.

6.2 Knowledge Integration

It is noted that the BN approach in [13] just involves 40 sets of dissolved gas data. From statistics point of view, a more training dataset will provide more reliable probabilistic knowledge to the BNs. In addition, the CPTs can also be obtained by other methods or sources, e.g., employing automated machine learning to discover the parameters of each CPT in [165]. In this case, this section describes the purpose and process of the knowledge integration.

The construction of an intelligent system or knowledge base is a gradual process, so that it always contains supplements and updates of the original knowledge base. The knowledge integration, which is based on the probabilistic knowledge update to a knowledge base, is an important issue in uncertainty reasoning [59] [166]. Briefly, knowledge integration aims to integrate a set of new constraints into an existing knowledge base (or JPD). In other words, it complements an existing JPD with probabilistic knowledge and seeks a comprehensive knowledge base, which satisfies with the new constraints [167]. For this purpose, Kruithof proposed iterative proportional fitting procedure (IPFP) [57], which is based on the minminum K-L divergence [168], iterating a given constraint set and performing the probabilistic knowledge integration. Also, IPFP is improved to conditional-IPFP (C-IPFP) by Bock, in which it is able to handle the conditional probabilistic constraints [169]. The limitation of IPFP and C-IPFP is that it can only be utilised to the case that knowledge base is JPD and not capable to deal with BNs directly. In that case, IPFP and C-IPFP are further developed to an extended-IPFP (E-IPFP) by Peng and Ding, so that it can be used to handle probabilistic knowledge integration for BNs [59]. Subsequently, E-IPFP is modified by Peng and Ding to decomposed-IPFP (D-IPFP) and the computational complexity is simplified. The details of these algorithms are demonstrated in the following section.

6.2.1 Related works

According to the probability theory frame, the domain problem is represented by a set of random variables, i.e., $X = (X_1, X_2, ..., X_n)$, where $P(X) = P(X_1, X_2, ..., X_n)$ denotes an n-dimensional JPD if for every assignment $x = (x_1, x_2, ..., x_n) \in X$,$0 \leq P(X) = P(X_1, X_2, ..., X_n) \leq 1$ and $\sum_{x \in X} P(X = x) = 1$ as x runs through all possible assignments of X.

Probabilistic constraints

The new probabilistic knowledge set is defined as a low dimensional probabilistic constraints, which is a subset of X, and denoted by equation (6.2.1).

$$R = \{P_1(Y^1), P_2(Y^2), ..., P_m(Y^m)\}, \tag{6.2.1}$$

where $Y^j \subseteq X$, m is the number of constraints.

Kullback-Leibler divergence

It is also known as K-L distance or relative entropy, which is a measure to reflect the difference between two JPDs. If O is the set of JPDs over random variables $X = (X_1, X_2, ..., X_n)$, and $Q, Q^* \in O$. The K-L divergence is defined by equation (6.2.2) and $0 \cdot \log \frac{0}{Q} = 0, Q^* \cdot \log \frac{Q^*}{0} = \infty$.

$$I(Q^* \| Q) = \sum_{x \in X, Q^*(x) > 0} Q^*(x) \log_2 \frac{Q^*(x)}{Q(x)}. \tag{6.2.2}$$

Probabilistic knowledge integration

Basically, it is the process, seeking a new JPD, which satisfies a probabilistic constraint set, i.e., given a JPD $Q(X)$ and a probabilistic constraint set R, a new JPD $Q^*(X)$ can be achieved, which satisfies R. Furthermore, the new JPD $Q^*(X)$ has minimum K-L divergence with initial JPD $Q(X)$.

I-projection

The probability distribution from a given set minimising K-L divergence with respect to a given distribution is called I-projection and illustrated by equation (6.2.3).

$$I(Q^* \| Q) = \min I(Q^* \| Q),$$

where

$$Q^*(X) = \begin{cases} 0 & \text{if} \quad Q(Y) = 0 \\ Q(X) \cdot \frac{P(Y)}{Q(Y)} & \text{if} \quad Q(Y) > 0 \end{cases} . \tag{6.2.3}$$

Normally, the constraint set contains more than one constraint. Thus, the problem becomes to an iterative process, in which repeatedly uses the constraints until the iteration converges.

6.2.2 Iterative proportional fitting procedure

The iterative process discussed in the last subsection is called IPFP, which is concluded as follows and Example 1 demonstrates the working process of IPFP.

1. Initial state: $Q_0(X)$, $R = \{P_1(Y^1), P_2(Y^2), ..., P_m(Y^m)\}$;
2. For $k = 1$, repeatedly do the following iterative process until the result converges:
   (a) $i = ((k-1) \bmod m) + 1$;
   (b)
   $$Q_k(X) = \begin{cases} 0 & \text{if} \quad Q_{k-1}(Y^i) = 0 \\ Q_{k-1}(X) \cdot \frac{P_i(Y^i)}{Q_{k-1}(Y^i)} & \text{if} \quad Q_{k-1}(Y^i) > 0 \end{cases} ;$$
3. $k = k + 1$.

Example 1:

Figure 6.1 presents a four node BN over variables $X = \{X_1, X_2, X_3, X_4\}$ with distribution $Q_0(X)$ and all random variables are binary, i.e., each node contains two states: true (T) or false (F). R_1 is a set of constraints, i.e., $R_1 = \{P_1(X_1) =$

$(0.55, 0.45), P_2(X_2) = (0.45, 0.55), P_3(X_3) = (0.35, 0.65)\}$, Table 6.1 shows the original JPD ($Q_0(X)$), JPD of the first fitting step ($Q_1(X)$) and the final JPD ($Q^*(X)$) of this BN with R, respectively. The K-L divergences between the JPDs in each fitting step $Q_k(X)$ and the original JPD are illustrated in Figure 6.2. The KL-divergence between $Q^*(X)$ and $Q_0(X)$ is 0.4767. The modified BN by the given constraint set R is shown in Figure 6.3. It can be verified that the resulted BN satisfies the three constraints of R_1, i.e., $Q^*(X_1 = T) = P_1(X_1 = T) = 0.55$, $Q^*(X_2 = T) = P_2(X_2 = T) = 0.45$ and $Q^*(X_3 = T) = P_3(X_3 = T) = 0.35$. In addition, the JPD obeys the chain rule of BNs such that $Q^*(X_1, X_2, X_3, X_4) = Q^*(X_1) \cdot Q^*(X_2|X_1) \cdot Q^*(X_3|X_1) \cdot Q^*(X_4|X_2, X_3)$.

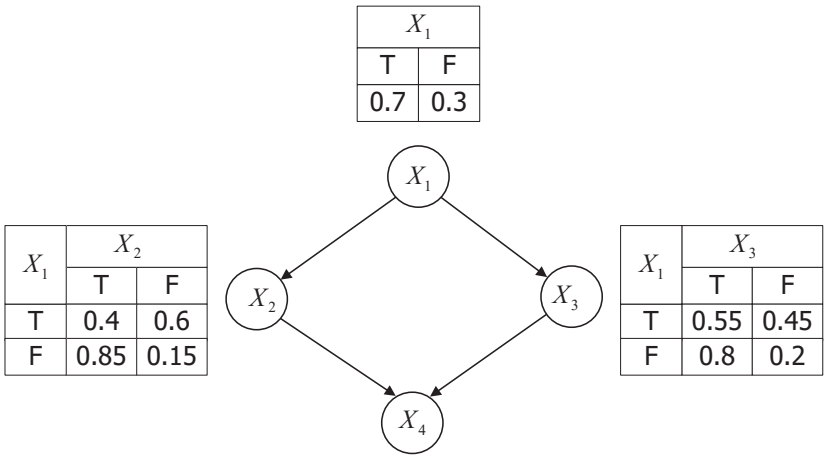


Figure 6.1: A four-node BN with its CPTs

6.2.3 Conditional-iterative proportional fitting procedure

In Example 1, all the probabilistic constraints are marginal probability. In practice, many probabilistic constraints are conditional probability distribution. In

Table 6.1: The original JPD ($Q_0(X)$), JPD of the first fitting step ($Q_1(X)$)) and the final JPD (($Q^*(X)$)) of the four-node BN

X_1	X_2	X_3	X_4	$Q_0(X_1, X_2, X_3, X_4)$	$Q_1(X_1, X_2, X_3, X_4)$	$Q^*(X_1, X_2, X_3, X_4)$
T	T	T	T	0.1386	0.1166	0.0264
T	T	T	F	0.0154	0.013	0.0029
T	T	F	T	0.11088	0.0933	0.0858
T	T	F	F	0.01512	0.0127	0.0017
T	F	T	T	0.17325	0.2049	0.0733
T	F	T	F	0.05775	0.0683	0.0244
T	F	F	T	0.1134	0.1341	0.1953
T	F	F	F	0.0756	0.0894	0.1302
F	T	T	T	0.1836	0.1544	0.1441
F	T	T	F	0.0204	0.0172	0.016
F	T	F	T	0.04488	0.0377	0.1434
F	T	F	F	0.00612	0.0051	0.0196
F	F	T	T	0.027	0.0319	0.0471
F	F	T	F	0.009	0.0106	0.0157
F	F	F	T	0.0054	0.0064	0.0384
F	F	F	F	0.0036	0.0043	0.0256

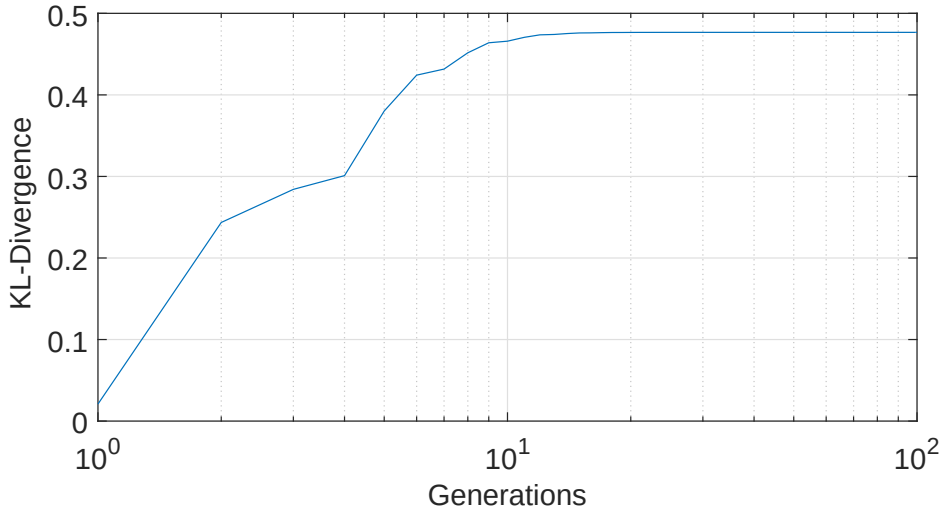


Figure 6.2: K-L divergence of $I(Q_k(X)||Q_0(X))$ using IPFP

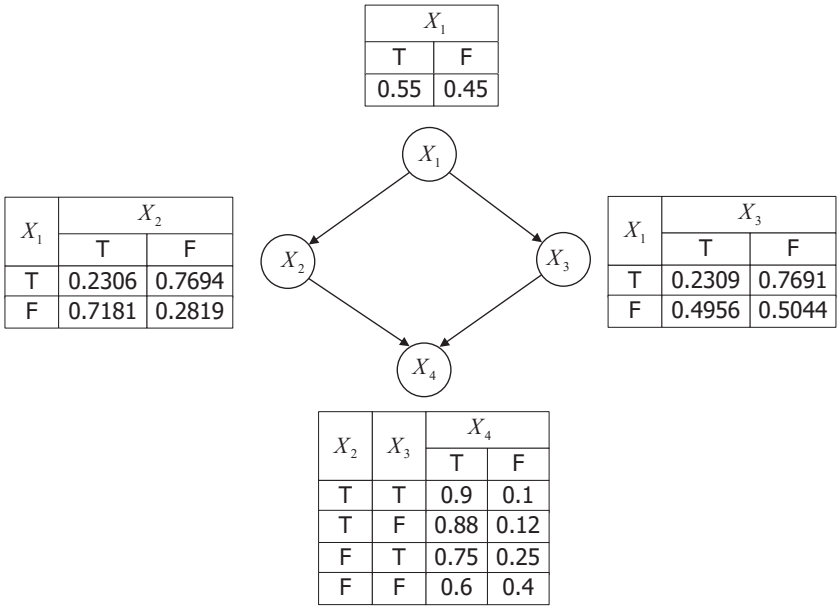


Figure 6.3: Resulted BN with constraint set R_1 using IPFP

this case, IPFP is upgraded to C-IPFP by Bock and Cramer, which is suitable to deal with the constraint in the form of $P(Y\,|Z\,)$, where Y is conditional to Z.
The algorithm is concluded as:

1. Initial state: $Q_0(X)$, $R = \{P_1(Y^1), P_2(Y^2), ..., P_m(Y^m)\}$;
2. for $\mathrm{k} = 1$, repeatedly do the following iterative process until the result converges:
   (a) $i = ((k-1) \bmod m) + 1$;
   (b)
$$Q_k(X) = \begin{cases} 0 & if \quad Q_{k-1}(Y^i\,|Z^i) = 0 \\ Q_{k-1}(X) \cdot \frac{P_i(Y^i|Z^i)}{Q_{k-1}(Y^i|Z^i)} & if \quad Q_{k-1}(Y^i\,|Z^i) > 0 \end{cases};$$
3. $k = k + 1$.

6.2.4 Extended-iterative proportional fitting procedure

If more than one variables occur in a constraint set, spanning over more than one CPT, the $Q^*(X)$ is not guaranteed to be satisfied with the chain rule. This issue is illustrated by Example 2.

Example 2:

A probabilistic constraint is applied to the same four-node BN in example 1, i.e., $R_2 = \{P_1(X_1, X_4) = (0.1322, 0.2863, 0.4231, 0.1584)\}$. The resulted JPD by IPFP is presented in Table 6.2(a) and the resulted BN with its CPTs is shown in Figure 6.4. It is verified that $Q_1^*(X)$ is consistent to R_2 (shown in Table 6.3(a)). However, it does not obey the chain rule, e.g., $Q_1^*(X_1 = T, X_2 = T, X_3 = T, X_4 = T) = 0.0342$, while $Q_1^*(X_1) \cdot Q_1^*(X_2|X_1) \cdot Q_1^*(X_3|X_1) \cdot Q_1^*(X_4|X_2, X_3) = 0.4185 \times 0.2744 \times 0.4891 \times 0.7519 = 0.0422$. If $Q_1^*(X)$ is modified by extracting CPTs in Figure 6.4, the resulted JPD $Q_2^*(X)$ is illustrated in Table 6.2(b). In this case, although the chain rule is satisfied, it is not consistent to the constraint set R_2 (shown in Table 6.3(b)).

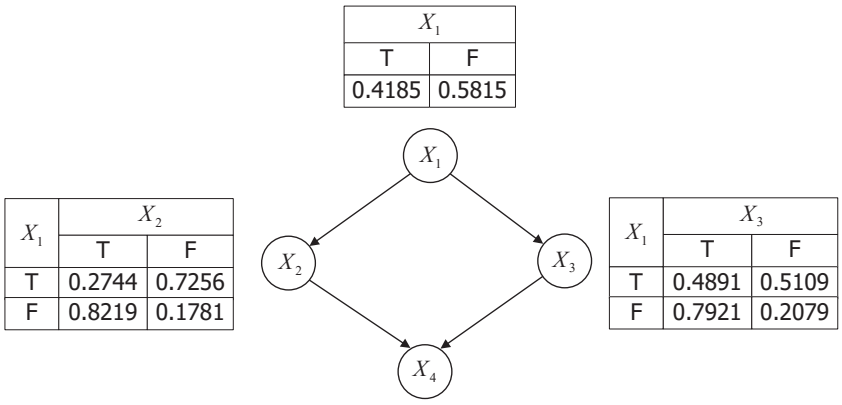


Figure 6.4: Resulted BN with constraint set R_2 using IPFP

Table 6.2: Resulted JPD $Q^*(X)$ with constraint set R_2 using IPFP

(a) JPD obtained by IPFP

X_1	X_2	X_3	X_4	$Q_1^*(X_1, X_2, X_3, X_4)$
T	T	T	T	0.0342
T	T	T	F	0.0269
T	T	F	T	0.0273
T	T	F	F	0.0264
T	F	T	T	0.0427
T	F	T	F	0.1009
T	F	F	T	0.028
T	F	F	F	0.1321
F	T	T	T	0.2978
F	T	T	F	0.0826
F	T	F	T	0.0728
F	T	F	F	0.0248
F	F	T	T	0.0438
F	F	T	F	0.0364
F	F	F	T	0.0088
F	F	F	F	0.0146

(b) JPD obtained by extracted BN

X_1	X_2	X_3	X_4	$Q_2^*(X_1, X_2, X_3, X_4)$
T	T	T	T	0.0422
T	T	T	F	0.0139
T	T	F	T	0.0388
T	T	F	F	0.0198
T	F	T	T	0.0574
T	F	T	F	0.0911
T	F	F	T	0.0311
T	F	F	F	0.1241
F	T	T	T	0.2847
F	T	T	F	0.0939
F	T	F	T	0.0658
F	T	F	F	0.0336
F	F	T	T	0.0317
F	F	T	F	0.0503
F	F	F	T	0.0043
F	F	F	F	0.0172

Table 6.3: Extracted $P_1(X_1, X_4)$ from resulted JPD

(a) R_2 of JPD $Q_1^*(X)$

X_1	X_4	
	T	F
T	0.1322	0.2863
F	0.4231	0.1584

(b) R_2 of JPD $Q_2^*(X)$

X_1	X_4	
	T	F
T	0.1695	0.2489
F	0.3865	0.1951

For a typical BN, there are two important factors:

- It is a directed acyclic graph, which is denoted by G;
- It must obey the chain rule as defined by equation (2.4.8).

If a BN N describes a set of n variables $X = (X_1, X_2, ..., X_n)$ with distribution $Q(X)$ and a set of constraints R, finding the N^* should meet the following three requirements:

- $G = G^*$ (Both networks have the same structure);
- $Q^*(X)$, the distribution of N^*, satisfies all constraints in R;
- K-L divergence $I(Q^*(X) \| Q(X))$ is minimum among all distributions that meet requirements 1 and 2.

To solve the BN modification problem defined above. Ding modified the original IPFP to E-IPFP to handle the requirement that the solution distribution should be consistent with the structure of the given BN, i.e., G_0. Thus, the requirement is regarded as a structural constraint, i.e., $R = \prod_{i=1}^{n} Q_{k-1}(X_i | \text{Parents}(X_i))$ in IPFP. Here, $Q_{k-1}(X_i | \text{Parents}(X_i))$ are extracted from $Q_{k-1}(X)$.

E-IPFP involves the structural constraint as the $(m+1)^{th}$ constraint $P_{m+1}(X)$. The algorithm of E-IPFP is shown as follows:

1. Initial state: $Q_0(X)$, $R = \{P_1(Y^1), P_2(Y^2), ..., P_m(Y^m)\}$;
2. Starting with $\mathrm{k} = 1$, repeat following iterative process until the result converges:
   (a) $i = ((k-1) \bmod (m+1)) + 1$;
   (b) if $i < m+1$
      i. if $(R_i \in R_m)$ (Using IPFP for marginal constraints)

$$Q_k(X) = Q_{k-1}(X) \cdot \frac{P_i(Y^i)}{Q_{k-1}(Y^i)};$$

ii. else if $(R_i \in R_c)$ (Using C-IPFP for conditional constraints)

$$Q_k(X) = Q_{k-1}(X) \cdot \frac{P_i(Y^i \,|Z^i)}{Q_{k-1}(Y^i \,|Z^i)};$$

(c) else extract $Q_{k-1}\left(X_i \,|\text{Parents}\left(X_i\right)\right)$ from $Q_{k-1}(X)$ according to G_0;

$$Q_k\left(X\right) = \prod_{i=1}^{n} Q_{k-1}\left(X_i \,|\text{Parents}\left(X_i\right)\right)$$

(d) $\mathrm{k} = \mathrm{k} + 1$;

3. return $N^*(X)$ with $G^* = G_0$.

Table 6.4 refers to the resulted JPD with R_2 by E-IPFP. The K-L divergence between the original JPD and the obtained JPD in each fitting step is illustrated in Figure 6.5. The final BN with its CPTs is presented by Figure 6.6. In this case, all the requirements defined before are satisfied in the new BN obtained by E-IPFP.

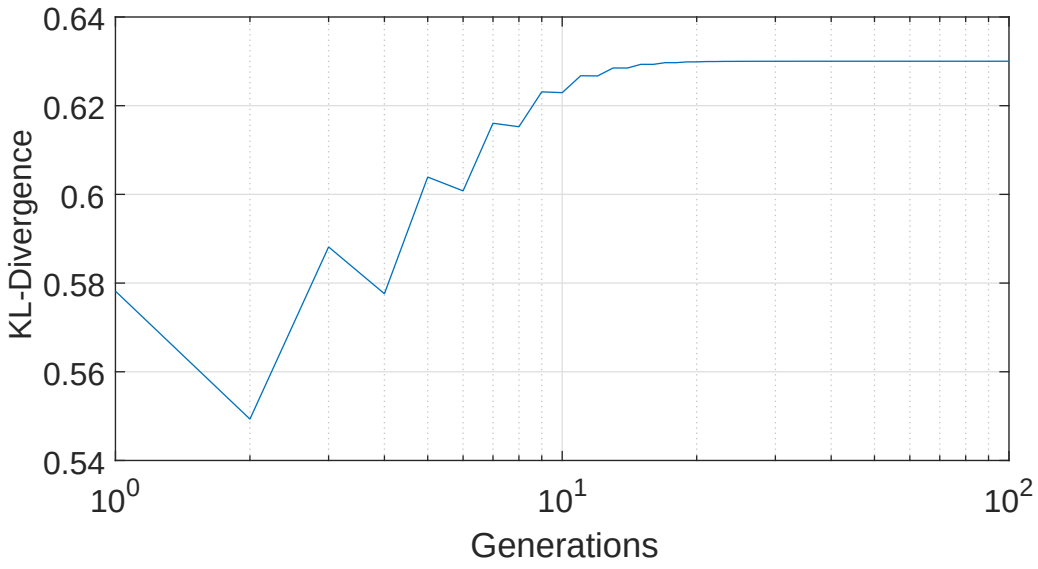


Figure 6.5: K-L divergence of $I(Q_k(X)||Q_0(X))$ using E-IPFP

6.2.5 Decomposed-iterative proportional fitting procedure

The computation of IPFP, C-IPFP and E-IPFP are all carried out on the entire JPD at every iteration. If a BN contains large numbers of nodes, the computational

Table 6.4: Resulted JPD with constraint set R_2 using E-IPFP

X_1	X_2	X_3	X_4	$Q^*(X_1, X_2, X_3, X_4)$
T	T	T	T	0.0324
T	T	T	F	0.0066
T	T	F	T	0.0322
T	T	F	F	0.0146
T	F	T	T	0.0471
T	F	T	F	0.1042
T	F	F	T	0.0205
T	F	F	F	0.1609
F	T	T	T	0.3336
F	T	T	F	0.0684
F	T	F	T	0.0671
F	T	F	F	0.0305
F	F	T	T	0.0205
F	F	T	F	0.0454
F	F	F	T	0.0018
F	F	F	F	0.0142

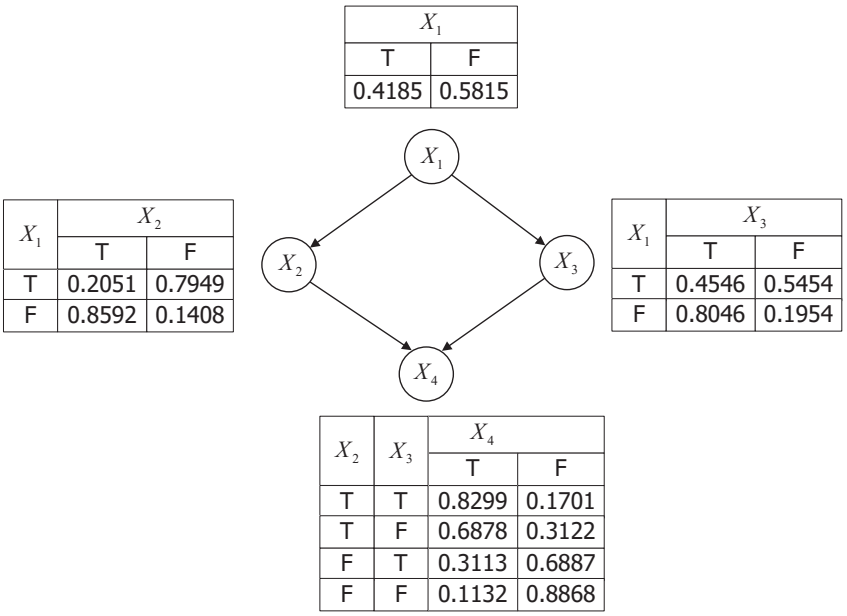


Figure 6.6: Resulted BN with constraint set R_2 using E-IPFP

cost will increase dramatically. Considering the interdependencies imposed on the distribution by the network structure, Ding proposed D-IPFP, aiming to update some selected CPTs only. The working process of D-IPFP is shown below:

1. Initial state: $Q_0(X)$, $R = \{P_1(Y^1), P_2(Y^2), ..., P_m(Y^m)\}$;
2. Starting with $\mathrm{k} = 1$, repeat following iterative process until the result converges:
   (a) $i = ((k-1) \bmod (m+1)) + 1$;
   (b) do E-IPFP with $P(Y^i)$ on $\{Y_i, \mathrm{Mb}(Y_i)\}$;

$$Q_k\left(Y^i, \mathrm{Mb}(Y^i)\right) = Q_{k-1}(Y^i, \mathrm{Mb}(Y^i)) \cdot \frac{P_i(Y^i)}{Q_{k-1}(Y^i \,|\mathrm{Mb}(Y^i))};$$

   (c) Extract the CPTs based on $Q_k\left(Y^i, \mathrm{Mb}(Y^i)\right)$, and update the BN;
   (d) $\mathrm{k} = \mathrm{k} + 1$.

In D-IPFP, the $\mathrm{Mb}(\cdot)$ represents the Markov blanket of a node, e.g., A, where consists of the set of parents, children and children's parents node of A [170].

Markov blanket defines a relationship of conditional independent such that a given node with its Markov blanket is conditional independent to other nodes.

6.3 Ontology-based Bayesian Networks

In Chapter 2, the ontology has been briefly introduced. The component of an ontology defined in OWL is the taxonomical concept and their hierarchy. As presented in Section 2.1.3, the logical relations and class axioms shown in Table 2.2 and Table 2.3 are for that purpose. This research is inspired by Ding and Peng's work [56], considering six constructors, i.e., three class axioms including rdfs:subClassOf, owl:equivalentClass, and owl:disjointWith; three logical operators consisting of owl:unionOf, owl:intersectionOf and owl:complementOf, with no regard to the constructors related to properties, individuals and datatypes.

6.3.1 Structural translation

Structural translation is the first step of the combination of OWL and BNs, especially for the structure of a BN, i.e., directed acyclic graph. The "subjects" and "objects" in RDF triples of an OWL file are translated into a set of nodes of a BN, and a directed arc is drawn between two related nodes according to "predicate" in the OWL file. There are two types of nodes in the translated BN, i.e., concept nodes (C-nodes) and logical nodes (L-nodes), which are utilised to indicate the concept classes and the logical relationships between C-nodes, respectively. The translation follows a set of translation rules, which is summarised below:

- Each concept class (e.g., X) is transferred into a single node with binary states, i.e., true (x) or false ($\bar{x}$).
- The constructor "rdfs:subClassOf" indicates the relationship between parent nodes and their child nodes. As presented in Figure 2.12, X_4 is the subclass of X_1 and X_2.
- According to the description of DLs in Table 2.2 and Table 2.3, the constructor of OWL: owl:complementOf, owl:disjointWith, owl:equivalentClass,

owl:unionOf and owl:intersectionOf are illustrated by the left parts of Figure 6.7 to Figure 6.11, respectively.

It is noted that all the L-nodes have no in-arcs, ensuring there is no cycles in translated BN. For instance, if A is equivalent to B, "rdfs:subClassOf" will define that A is a subclass of B and B is a subclass of A. As a consequence, a cycle between A and B is formed.

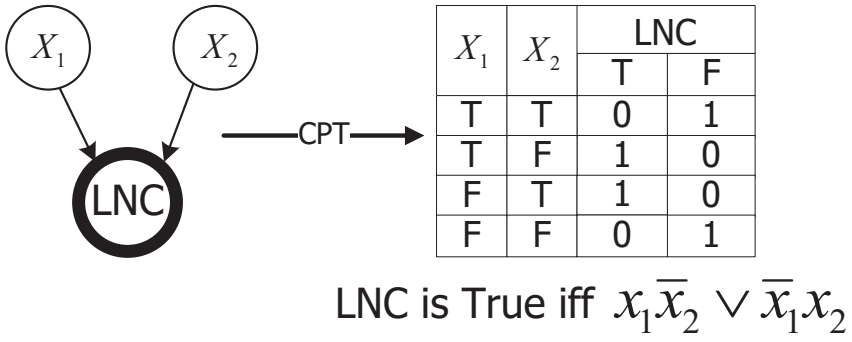


Figure 6.7: L-node owl:complementOf (LNC) and its CPT

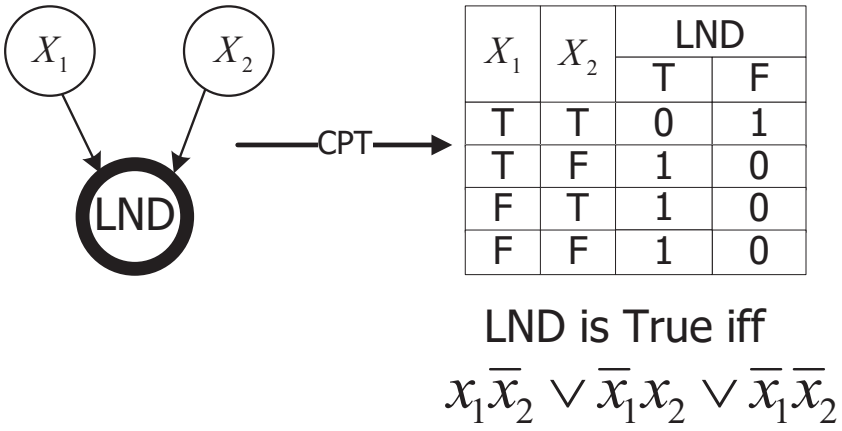
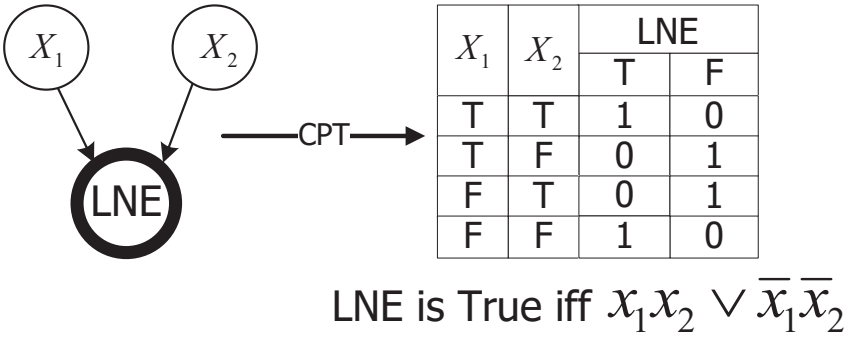


Figure 6.8: L-node for owl:disjointWith (LND) and its CPT

6.3.2 Conditional probability table construction

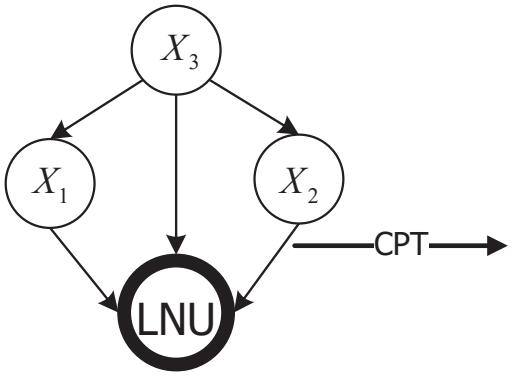
To accomplish the full translation from an OWL taxonomy to a BN is similar to the process of building BNs. Once the network structure is obtained, the remaining



X_1	X_2	LNE	
		T	F
T	T	1	0
T	F	0	1
F	T	0	1
F	F	1	0

LNE is True iff $x_1x_2 \vee \overline{x}_1\overline{x}_2$

Figure 6.9: L-node for owl:equivalentClass (LNE) and its CPT

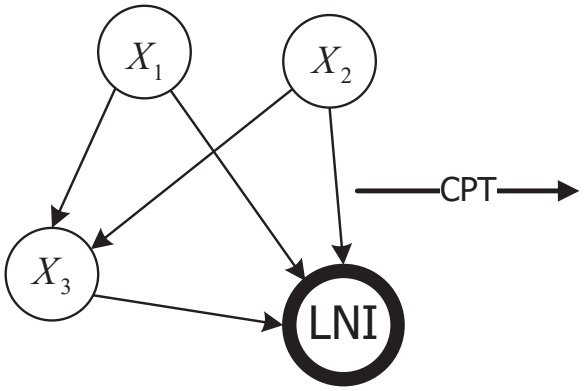


X_1	X_2	X_3	LNU	
			T	F
T	T	T	1	0
T	T	F	0	1
T	F	T	1	0
T	F	F	0	1
F	T	T	1	0
F	T	F	0	1
F	F	T	0	1
F	F	F	1	0

LNU is True iff

$xx_1x_2 \vee x\overline{x}_1x_2 \vee xx_1\overline{x}_2 \vee \overline{x}\overline{x}_1\overline{x}_2$

Figure 6.10: L-node for owl:unionOf relation (LNU) and its CPT



X_1	X_2	X_3	LNI	
			T	F
T	T	T	1	0
T	T	F	0	1
T	F	T	0	1
T	F	F	1	0
F	T	T	0	1
F	T	F	1	0
F	F	T	0	1
F	F	F	1	0

LNI is True iff

$$xx_1x_2 \vee \overline{x}\overline{x}_1x_2 \vee \overline{x}x_1\overline{x}_2 \vee \overline{x}\overline{x}_1\overline{x}_2$$

Figure 6.11: L-node for owl:intersectionOf (LNI) and its CPT

issue is to construct the CPTs for each node in the translated BN. CPTs for L-nodes can be determined by the logical relations. The CPTs of the five types of L-nodes corresponding to owl:complementOf, owl:disjointWith, equivalentClass, owl:unionOf and owl:intersectionOf are presented by the right parts of Figure 6.7 to Figure 6.11, respectively. Also, the true condition for each case is included.

In order to keep the BN consistent to the OWL semantics, all the states of the L-nodes should be set to true. In this case, the CPTs of C-nodes are conditional dependent to the true states of L-nodes (denoted by LnT), i.e., $P(X_C|LnT)$. The CPTs of C-nodes can be achieved either by the domain experts' statistical analysis so that the prior probabilities of concepts and pair-wise conditional probabilities of concepts are returned or random initialisation and modified by probabilistic constraints using knowledge integration methods, i.e., E-IPFP or D-IPFP. Both of the patterns to construct CPTs of C-nodes are followed by probabilistic integration, as the constraints are always updated as mentioned before. In summary, the 2(a)

step of E-IPFP is modified as illustrated by equation (6.3.1).

$$Q_k\left(X, LnT\right) = Q_{k-1}\left(X, LnT\right) \cdot \frac{R_i\left(Y^i\right)}{Q_{k-1}\left(Y^i|LnT\right)}. \tag{6.3.1}$$

6.3.3 Representation of probabilistic knowledge in OWL

In ontology, uncertainty knowledge can be regarded as the probabilistic constraint between concept classes and their relations. Both the marginal probability and the conditional probability are represented by OWL. Following Ding's research, a probability is regarded as a kind of resource and two OWL classes are defined, i.e., "MarginalProb" and "ConditionalProb". A marginal probability $P(A)$ of a variable A is defined as an instance of class "MarginalProb", and a conditional probability $P(A|B)$ of A is an instance of "ConditionalProb". These two classes have several properties as illustrated in Figure 6.12, where the number of "hasCondition" is no less than one and others are mandatory to one. The class "Variable", which has two properties, i.e., "hasClass" and "hasState", is defined to set the range of "hasCondition" and "hasVariable". "hasClass" indicates the concept class this probability is about and "hasState" presents the state of the probability, i.e., true or false. The illustration of these definitions is given in the subsection.

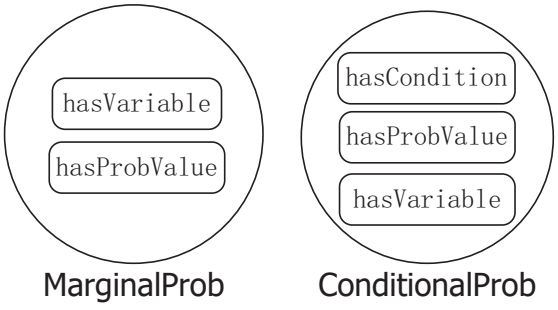


Figure 6.12: Properties of class "MarginalProb" and "ConditionalProb"

6.3.4 Framework Implementation

The implementation of the ontology-based BN can be achieved based on Java. Firstly, Jena is used, which is similar to the QE process mentioned in Section 3.1.2,

extracting ontology knowledge from an OWL file. Validation is made so that a given ontology file is syntactically valid and semantically consistent. Secondly, the OWL API in Jena is applied to collect concepts and their relations defined in an OWL file. Once all the relevant information is obtained, the NeticaJ API [171] is applied to generate a structured BN with CPTs of L-nodes based on the structure translation rules. NeticaJ API is developed by Norsys company, which is one of the most powerful BNs development softwares [172]. Subsequently, given a set of initial CPTs, a set of probabilistic constraints is integrated into the BN by using E-IPFP or D-IPFP. Finally, the BN with its belief bars is presented so that one can use this translated BN for reasoning.

This framework consists of three types of input, i.e., OWL ontology, initial probabilistic data, and probabilistic constraints. According to Figure 2.4 in Section 2.1.4, the common fault types and their corresponding symptoms of a power transformer are firstly extracted from the power substation database by power engineers, followed by being modelled to the concepts hierarchy taxonomy and formalised by Protégé to OWL ontology. The initial CPTs of each C-node can either be obtained by analysing historical data or randomly generation (as the CPTs of C-nodes will be modified based on the probabilistic constraints). L-nodes aim to provide the logic relations among concepts and guarantee the structure of the BN. Subsequently, if more sets of sample data are analysed so that the conditional probability between any related C-nodes or marginal probability are changed. These probabilistic data form a set of probabilistic constraints, which is further employed to refine the existing BN for PTFD. Thus, the ontology-based BN is built. Once the framework is implemented, a user can perform reasoning services for the PTFD. For example, if any symptom is detected, a user can set the belief bar of the corresponding node to "True" or "1". Therefore, the CPTs of relevant C-nodes will be modified automatically accordingly. Thus, the most likely fault location is investigated by a higher probability. Finally, suitable maintenance strategies or replacement services can be employed to the power transformer. It is worth mentioning that a user can also analyse the historical data and extract probabilistic constraints, followed by refining the existing ontology-

based BN model and applying related reasoning services. Figure 6.13 illustrates the flowchart of the ontology-based BN for PTFD. The following section gives a simple illustration of the framework implementation on a small-scale PTFD system.

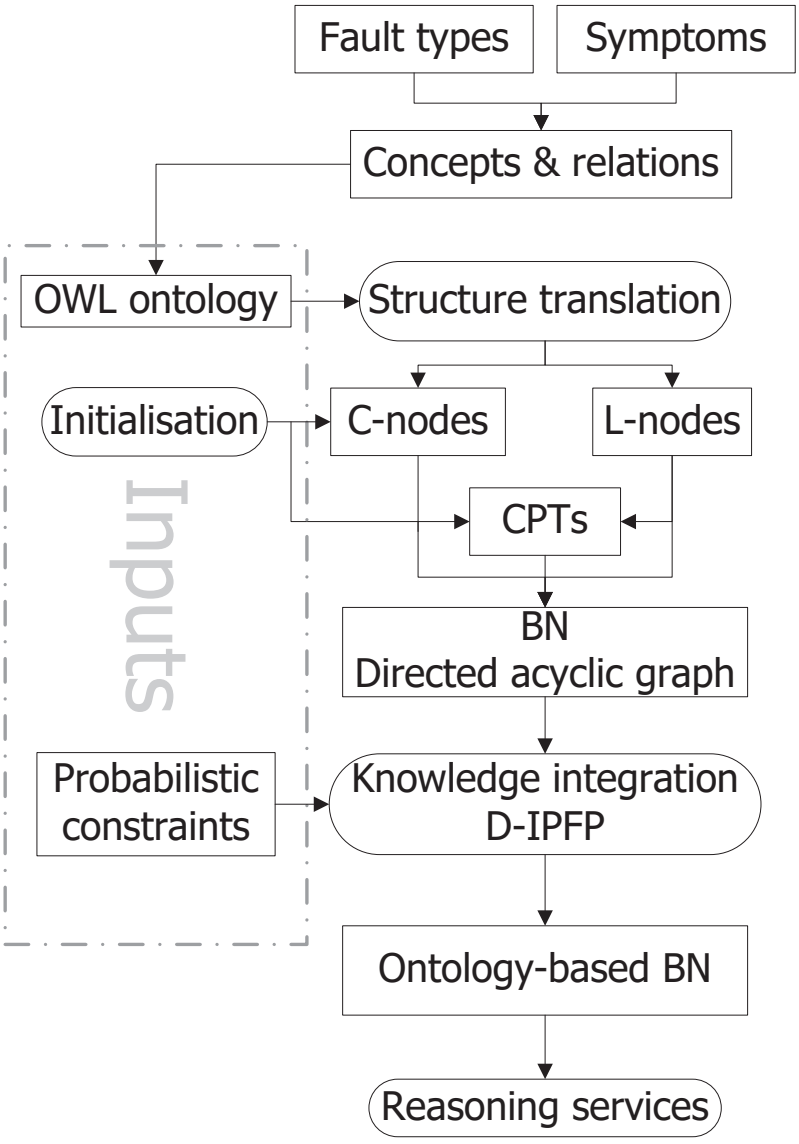


Figure 6.13: The flowchart of the ontology-based BN for PTFD

6.4 A Simple Illustration for Power Transformer Fault Diagnosis using Ontology-based Bayesian Networks

6.4.1 Building the initial ontology-based BN

A transformer winding is normally wrapped with paper, being located in oil. An insulation fault may happen, when trace water occurs either in the paper or oil. These two phenomena is regarded as a relationship of union. This structure is mapped into a binary variable node in the translated BN. The concept classes and their relations in an OWL file are presented in List 6.1. Following the translation rule, the translated BN of the above example is illustrated in Figure 6.14. In

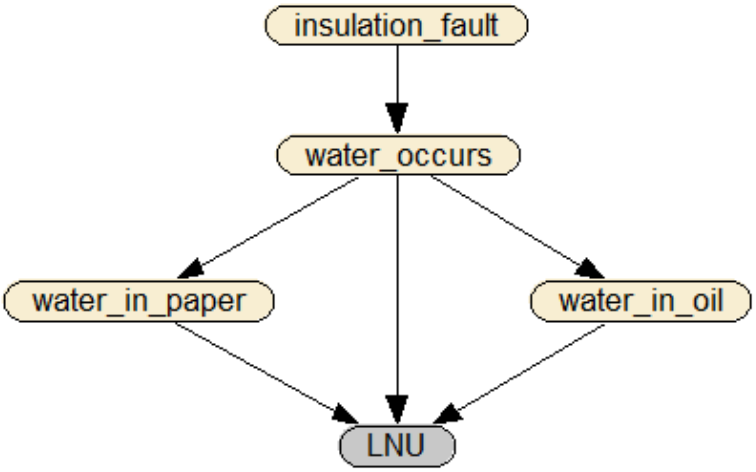


Figure 6.14: Translated BN for the union relation of the insulation fault example

addition, an assumption is made that if the insulation fault occurs in the paper, there is no trace water in the oil. Thus, "fault in paper" and "fault in oil" can be regarded as two disjoint classes. Also, two intersection relations can be observed in the above example, as "water in paper" is the intersection of "fault in paper" and "water occurs"; "water in oil" is the intersection of "fault in oil" and "water occurs". Finally, the translated BN with the initial CPTs and belief bars of each C-node (the CPTs of L-nodes are shown in Section 6.3.2) is illustrated in Figure 6.15.

A set of probabilistic constraints regarding the PTFD example, i.e., $R_3 = \{P_1(\text{ins ulation\_fault} = \text{T}) = 88.88\%, P_2(\text{water\_occurs} = \text{T}|\text{insulation\_fault} = \text{T}) = 35.90\%, P_3(\text{fault\_in\_paper} = \text{T}|\text{insulation\_fault} = \text{T}) = 50\%, P_4(\text{fault\_in\_oil} = \text{T}|\text{insulation\_fault} = \text{T}) = 40\%, P_5 = (\text{water\_in\_paper} = \text{T}|\text{water\_occurs} = \text{T}, \text{insul ation\_fault} = \text{T}) = 54.7\%\}$, is given.

Listing 6.1: OWL file for concept classes

```
<owl:Class rdf:ID="water_occurs">
   <rdfs:subClassOf rdf:resource="#insulation_fault"/>
      <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#water_in_oil"/>
      <owl:Class rdf:about="#water_in_paper"/>
   </owl:unionOf>
</owl:Class>
```

Listing 6.2: OWL file for probabilities

```
<Variable rdf:ID="a">
    <hasClass>insulation_fault </hasClass>
    <hasState>True</hasState>
</Variable>
<Variable rdf:ID="b">
    <hasClass>water_occurs </hasClass>
    <hasState>True</hasState>
</Variable>
<MarginalProb rdf:ID="P(a)">
    <hasVariable>a</hasVariable>
    <hasProbValue>0.8888</hasProbValue>
</MarginalProb>
<ConditionalProb rdf:ID="P(b|a)">
    <hasCondition>a</hasCondition>
    <hasVariable>b</hasVariable>
    <hasProbValue>0.359</hasProbValue>
</ConditionalProb>
```

Due to the space limitation, the OWL file for the first two probabilistic constraints, i.e., P_1 and P_2 in R_3, are shown in List 6.2. Applying D-IPFP, the initial BN shown in Figure 6.15 is modified to the BN with belief bars and CPTs illustrated

in Figure 6.16. All the CPTs have changed. It is noted that the CPT of the concept class "insulation fault" is consistent to the constraint P_1, i.e., $P(\text{insulation\_fault} = \text{T}) = 88.88\%$. Also, if the insulation fault occurs in the diagnosis system, i.e., $P(\text{insulation\_fault} = \text{T}) = 100\%$, the resulted BN is presented in Figure 6.17. The conditional probability of "fault in paper", "fault in oil" and "water occurs", given "insulation fault" is true, are 50%, 40%, and 35.9%, respectively, which are consistent to the constraint set $R_3 = \{P_2, P_3, P_4\}$ in R_3. The constraint P_5 is also achieved by setting both "insulation fault" and "water occurs" to true (100%) as shown in Figure 6.18.

6.4.2 Modification by new probabilistic knowledge and belief updates

At present, the fundamental PTFD system is constructed by extended OWL using BN. If the trace water occurs, power engineers can examine the relevant parts of the system according to obtained probabilities and provide actions. However, it can be observed that "fault in paper" and "fault in oil" shown in Figure 6.18 have similar probabilities (i.e., 54.7% and 45.3%), leading to the difficulty in the decision making. In this case, both the concept classes should be examined. If more reliable knowledge is obtained, showing that "water in oil" is more likely to cause an insulation fault, these knowledge will be integrated into the fundamental diagnosis system.

If R_4 is the new probabilistic knowledge and $R_4 = \{P_1(\text{insulation\_fault} = \text{T}) = 56.32\%, P_2(\text{water\_occurs} = \text{T}|\text{insulation\_fault} = \text{T}) = 49.5\%, P_3(\text{fault\_in\_paper} = \text{T}|\text{insulation\_fault} = \text{T}) = 3.57\%, P_4(\text{fault\_in\_oil} = \text{T}|\text{insulation\_fault} = \text{T}) = 94.3\%, P_5 = (\text{water\_in\_paper} = \text{T}|\text{water\_occurs} = \text{T}, \text{insulation\_fault} = \text{T}) = 0.85\%\}$, the modified BN using D-IPFP is shown in Figure 6.19. If the state of "insulation fault" and "water occurs" are set to be true in succession as illustrated in Figure 6.20 and Figure 6.21, respectively, the "fault in oil" has much higher probability than "fault in paper". The result is consistent to the new constraint set R_4.

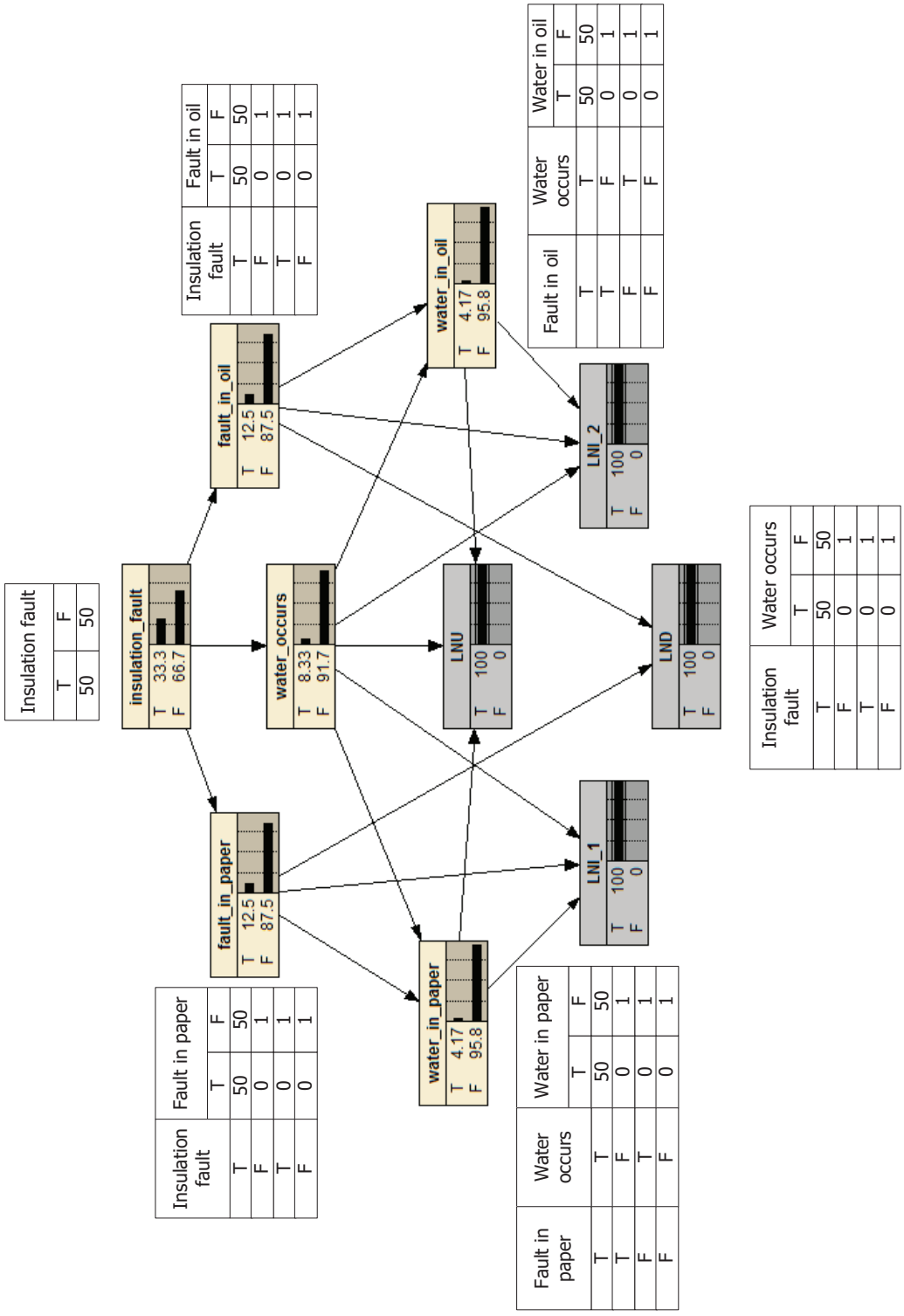


Figure 6.15: Translated BN with the initial belief bars and CPTs (%) of C-nodes for the insulation fault example

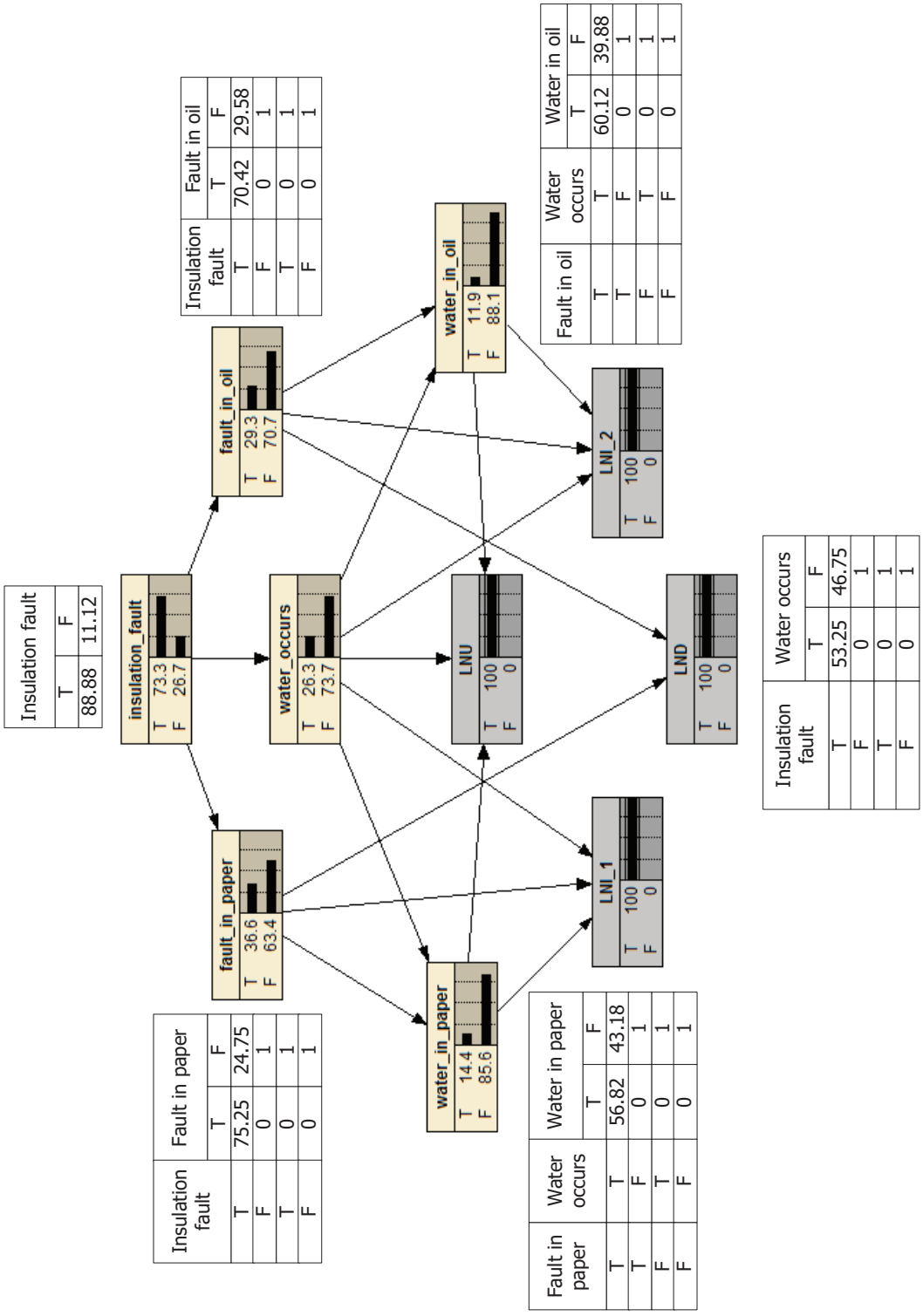


Figure 6.16: Modified BN by constraint set R_3

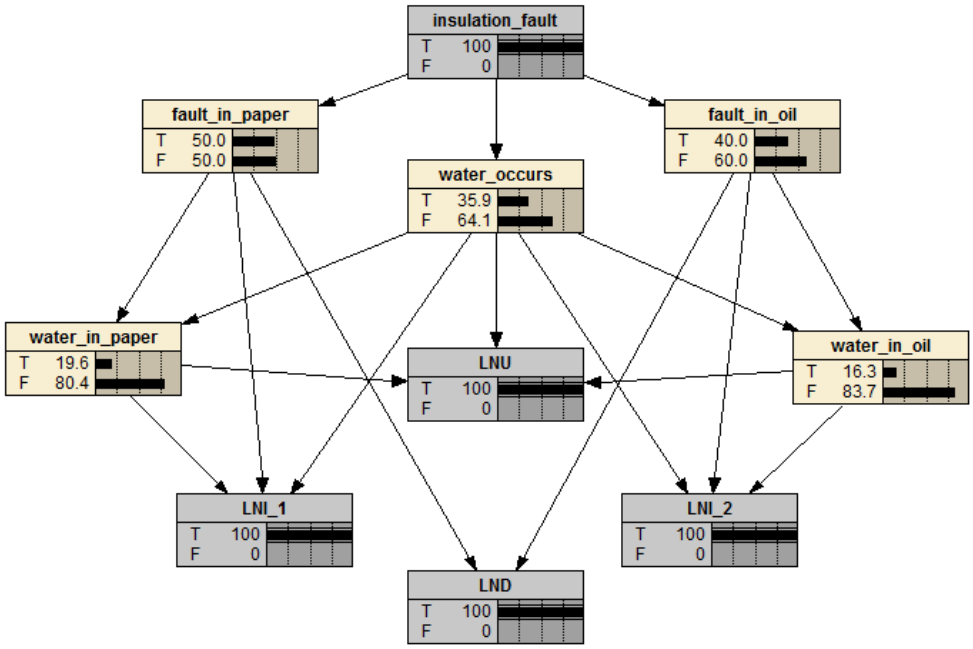


Figure 6.17: Modified BN by constraint set R_3, when "insulation fault" is true

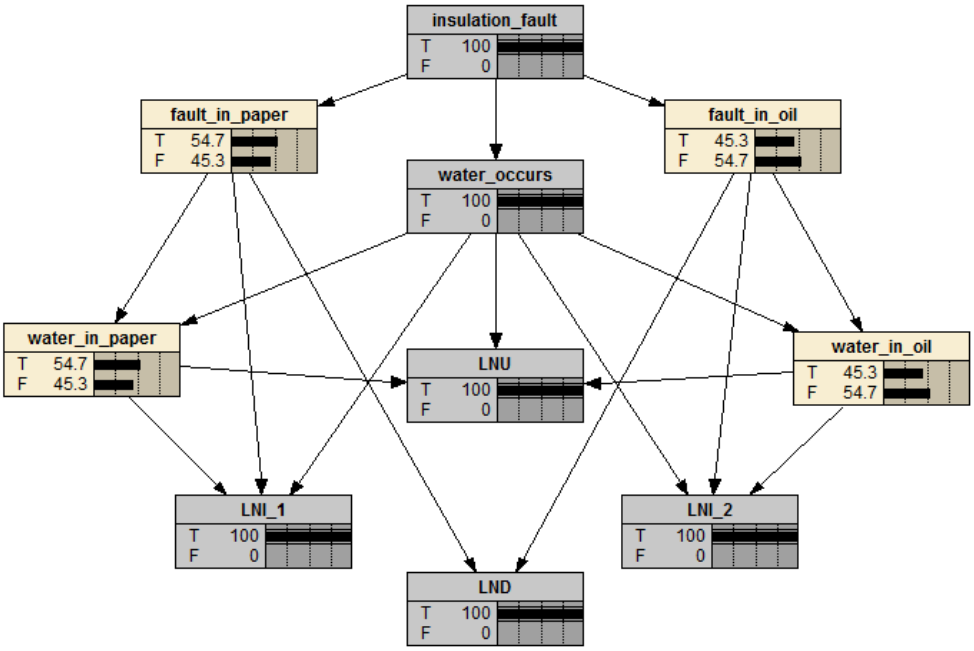


Figure 6.18: Modified BN by constraint set R_3, when "insulation fault" and "water occurs" are true

If the system receives a signal showing that an insulation fault occurs in oil as shown in Figure 6.22, as a disjoint concept, the insulation fault is not occurring in the paper. Therefore, water occurs in oil, which can be the only reason leading to the insulation fault. Also, the superclass "water occurs" is indicated with a probability, as there exist other factors causing an insulation fault, which are not listed in the example BN. This type of diagnosis can be extended to a large diagnosis system for reasoning. In addition, if water occurring in oil is determined, the obtained BN is presented in Figure 6.23. This hard evidence combined with the true states of L-nodes gives the result of diagnosis, where water occurs in the oil, resulting in an insulation fault, and there is no water in the paper.

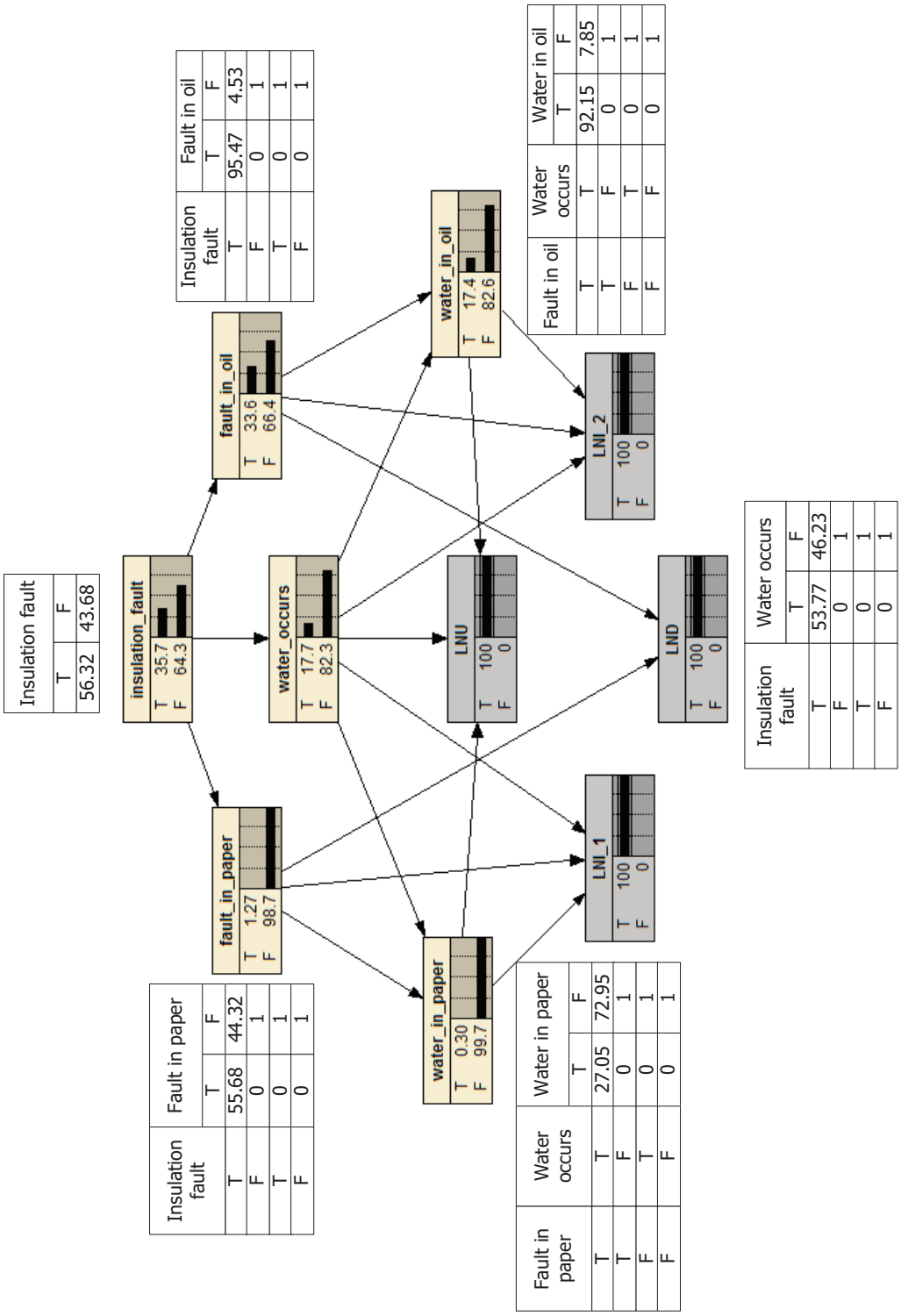


Figure 6.19: Modified BN by constraint set R_4

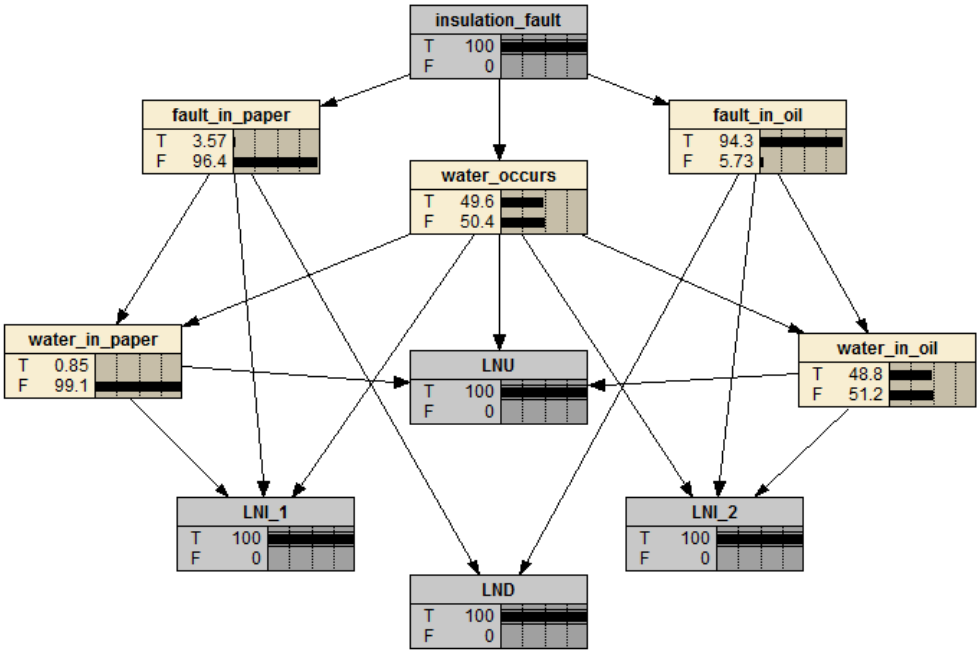


Figure 6.20: Modified BN by constraint set R_4, when "insulation fault" is true

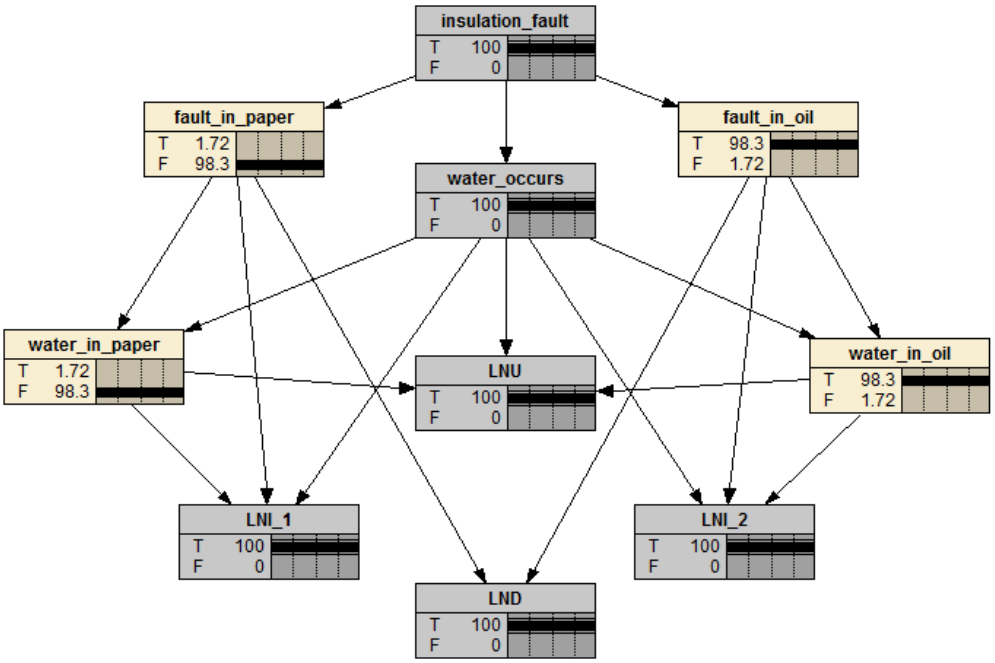


Figure 6.21: Modified BN by constraint set R_4, when "insulation fault" and "water occurs" are true

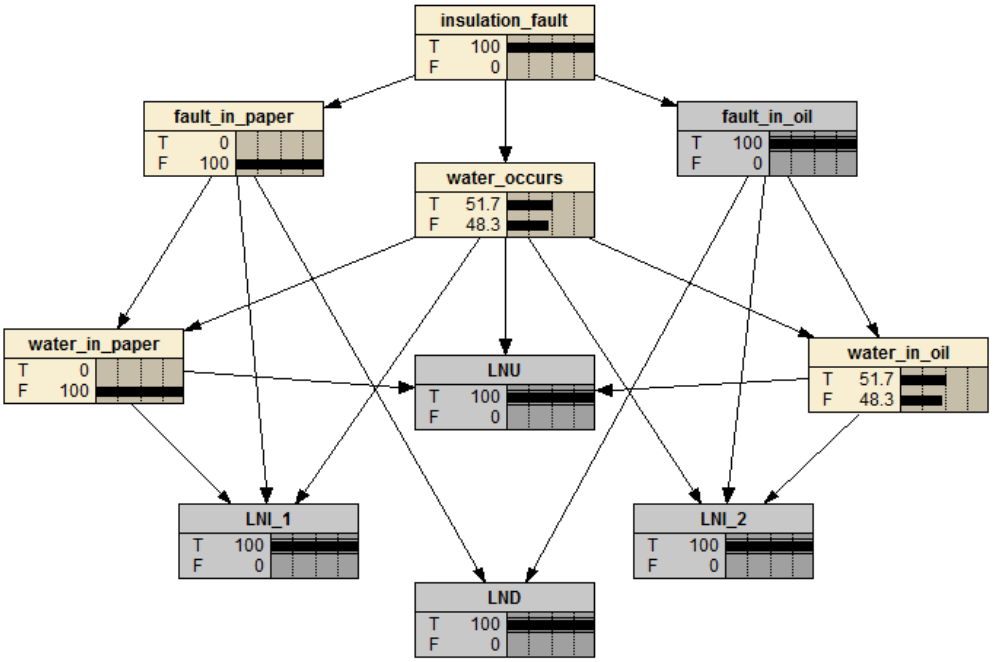


Figure 6.22: Modified BN by constraint set R_4, when "fault in oil" is true

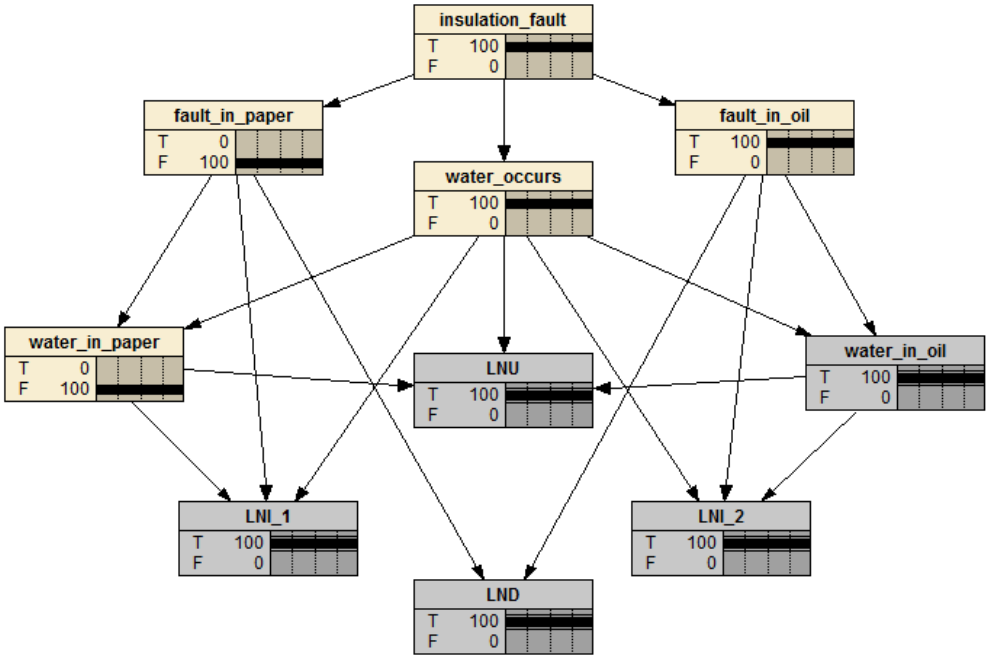


Figure 6.23: Modified BN by constraint set R_4, when "water in oil" is true

6.5 Summary

Ontology-based PTFP systems hold strong pertinence and extensibility that have been employed for domain knowledge representation. The limitation exists in the current ontology languages, as it is unable to handle the incomplete and inaccurate knowledge, and perform uncertainty reasoning. This chapter presented an ontology reasoning method, i.e., ontology-based BNs, to provide the current PTFD systems with the ability of probabilistic inference, and graphically presenting the inference outcomes. This framework was original inspired by Ding and Peng. Nevertheless, it was firstly employed to an ontology-based PTFD system. Knowledge integration algorithms including IPFP, C-IPFP, E-IPFP, and D-IPFP have been demonstrated with examples, in which E-IPFP is capable of dealing with the probabilistic knowledge integration in BNs and D-IPFP is an upgraded E-IPFP. A set of translation rules was defined so that the concepts and their relations in ontology can be mapped to BNs. As a consequence, the ontology reasoning can be transferred into BNs reasoning, which keeps the semantics of ontology and take advantage of the reasoning ability of BNs. Finally, this method was demonstrated by a small-scale PTFD system. The results illustrated in Section 6.4 simulated the process of PTFD, verifying that it is viable to overcome the limitations existing the current ontology-based PTFD.

Chapter 7

Conclusions

This chapter concludes the thesis and summarises the major achievements of the presented research work in the field of AM in power substations. Firstly, the summary of the research results illustrated in this thesis is given in Section 7.1, in which the major contributions of this research are highlighted. Subsequently, suggestions for future research are listed in Section 7.2.

7.1 Summary

The departure point of the research work aims to apply artificial intelligence methods to the AM of power substations. The power substation-related document repository and power transformer are the objects of study in this research, as they belong to intangible assets and physical assets of power substations, respectively. Three intelligence approaches have been proposed, i.e., ER-based ODSE, CC for ontology-based PSD using modified WPKGA, and an ontology-based BN for PTFD, in which the first two approaches belong to the ontology application in the research field of IR, and ontology employed in the knowledge engineering aspect was applied to the third approach. In the preceding chapters, the following work and promising results obtained, have been illustrated.

In Chapter 1, the definition of AM and the aspects of AM in power systems were given. Then, the concerning of AM related to this research, i.e., document searching

in power substations, document cluster analysis, and PTFD, were presented. Subsequently, brief reviews of related research areas are introduced. Afterwards, the significance of employing novel intelligent techniques for solving the limitations existing in the conventional substations AM solutions was explained. Finally, the thesis outline and the major contributions of this research were illustrated, which were followed by a list of published or submitted academic papers by the thesis author.

As each of the proposed approaches is based upon ontology, the basics of ontology were firstly given in Chapter 2, in which the Semantic Web, ontology, DLs and their relations were presented. Subsequently, an introduction of the generalised ontology building procedure concerned in this thesis was illustrated. Also, the applications of ontology were indicated. In addition, a historical literature review of IR was given, followed by introducing a set of mathematical models in IR. VSM was demonstrated in detail, as it was employed in the ER-based ODSE and the document representation for cluster analysis. Two optimisation techniques of meta-heuristics, i.e., the SA and the GA, have been reviewed, as the CC algorithms employed in this research are based on optimisation. Moreover, a number of mechanisms of each genetic operator of the GA were presented, concerning the selection mechanism, crossover mechanism, and mutation mechanism of the GA. Typically, in the mutation scheme, a mutation method for document clustering was presented by applying Baker's linear ranking method. Finally, the basics of BNs were demonstrated, including the directed acyclic graph structure of BNs, CPTs, and probabilistic inference in PTFD systems. This chapter can be regarded as the foundation of the proposed ontology-based approaches.

The development of SONT, which was employed for QE in an ODSE, was first introduced in Chapter 3. Subsequently, the procedure of organising a query and its expanded query terms into a MADM tree model was discussed. Then the ER algorithm, which is based on DS theory, was presented. Afterwards, the multi-criteria decision support methodology AHP was applied for generating the relative weights among attributes defined in the MADM tree model. DS combination rules were applied to generate the final relatedness between a given query and a document.

Subsequently, the experimental work of the proposed ER-based document ranking approach was reported. Apart from the ER-based ODSE, three document search engines were employed for comparison purpose under the same test scheme, i.e., the traditional keyword-matching search engine without QE and ER, the ODSE without ER, and the NRW approaches. The search engines were based on Lucene, which were illustrated by an example. Before the comparison, a simple search scenario using the proposed search engine was demonstrated. The simulation results have clearly shown that the ER-based ODSE is capable of combining multiple relevance scores generated between the terms of an expanded query and a document, and it has achieved the highest search accuracy amongst the four search engines. Thus, the ER-based approach can be employed as a viable solution in ODSE.

Chapter 4 started from introducing the basics of cluster analysis and document clustering. One of the most popular single clustering algorithms, i.e., k-means, was presented. Subsequently, the CC was involved in this research to tackle the limitations of single clustering algorithms. Three novel CC algorithms were reviewed in detail and studied in this research, i.e., NNMF-CC, WPK-CC, and INT-CC. Also, the validation methods of clustering algorithms were presented. Finally, implementations of selected algorithms were carried out on three sample datasets and four document datasets including the PSD. Given the initial states of each algorithm, the results have shown that the WPK-CC outperforms the other algorithms on both sample datasets and document datasets in purity and F-measure, while the single clustering algorithm, i.e., k-means, has the worst performance. There is no existing comparison between these CC algorithms, and they have never been applied to document clustering. This part of the work provides a reference, utilising CC algorithms to address document clustering and stimulated further study of CC for PSD.

According to the results obtained in Chapter 4, the improvement of WPK-CC for PSD was presented in Chapter 5. If CC for the PSD using WPK-CC is regarded as a system, the performance of this system is improved by modifying the original term-based VSM by SONT and employing the GA to optimise WPK-CC. A theoretical comparison between the SA and the GA was provided, specifically

from the clustering problem point of view. Afterwards, the SONT-based VSM and term mutual information were presented, which was inspired by the WordNet-based VSM. The employment of SONT has overcome the limitations of WordNet, as some of the terms in PSD are restricted to the specific domain of power substations. Also, an illustrative example for SONT-based VSM was provided. Finally, three simulation studies were designed, aiming to compare CC for the PSD with term-based VSM and SONT-based VSM; the CC algorithm of the original WPK-CC (or WPKSA) and WPK-CC combined with GA (or WPKGA); and WPKGA with different mechanisms. In each comparison step, the better performing strategy was kept, and finally the most comprehensive method for PSD CC was achieved.

Chapter 6 demonstrated an ontology-based BN framework for PTFD. Firstly, the capacity of BNs to deal with uncertainty reasoning in PTFD systems was briefly stated. Also, the limitation of conventional ontology-based systems for such purpose was further discussed. Secondly, the knowledge integration was presented, in which four probabilistic knowledge integration algorithms, i.e., IPFP, C-IPFP, E-IPFP, and D-IPFP, were illustrated with the aid of examples. Subsequently, the ontology-based BN was introduced, consisting of a set of structural translation rules and CPTs construction for both L-nodes and C-nodes. Also, the framework implementation was discussed, which began with defining two types of OWL file, i.e., concepts with their relations and the probabilities. Finally, an illustration of this framework was provided for a simple PTFD system. The results have shown that the ontology-based BN is able to handle the uncertainty reasoning and to provide power engineers probabilistic information so that actions can be taken according to the resulting probability.

7.2 Limitations of the Present Study and Suggestions for Future Work

In this section, the limitations of the present study are discussed. For each raised limitation, several related points that deserve further investigation are addressed.

- SONT is manually developed, which is time-consuming. Apparently, this is an unsuitable way to manually develop a more general ontology model or a complex domain ontology model. The automatic construction for extending SONT from a power substation domain to a power system domain will be addressed in future works.

- For the first approach of this thesis, there are some advanced topics worthy of study in the future. Because of the extra computation procedures of ER, the efficiency of the developed search engine using ER is slightly lower than other search engines without ER embedded. It costs more time than other search engines in a search process. Considering the concept of recall and precision, a perfect search engine has a constant 100% precision as recall increases. However, for a realistic search engine, the curve will always involve decreased precision, which has been reported in many research papers. Figures 3.13 to 3.15 illustrate the search to find the most competitive search engine. However, they are not designed for efficiency evaluation. This issue will be investigated in our future research.

- For the second part of this thesis, CC for PSD is studied independently of the context of IR. The only focus of this approach is to achieve better performance of the clustering results. There is no direct link between document clustering, document ranking, and user's experience. Thus, document clustering returns a set of meaningful clusters, which will be connected to the ER-based ODSE, followed by a set of accuracy and efficiency evaluations. Meanwhile, as an application, the combined ODSE with document clustering method in IR will be designed and presented by a user interface.

  Secondly, the number of CC algorithms concerned in this thesis is still limited. Although they are proven to be more advanced than many existing CC algorithms, they were tested only on sample datasets. Are there any CC algorithms or even single algorithms more suitable for document clustering concerning both accuracy and efficiency? Hence, more investigations and comparisons will be carried out.

Thirdly, the cluster number k is based on the original class of each dataset. Apparently, the contents of one document are not limited to only one topic. In this case, different k should be analysed. Compared with $k = 6$ for PSD, a larger k will generate a set of clusters with smaller sizes, which holds more specific meanings. Additionally, it is expected to achieve higher accuracy for the results of document searching, combined with the ER-based ODSE. Therefore, k with different values will be studied.

- The ontology-based BN in the third approach in this thesis is only demonstrated by a simple illustration. All the probabilistic constraints were based on reasonable assumptions. Thus, this framework will be implemented to the real PTFD system. In this case, the concept classes, e.g., fault types, symptoms, and etc, will further studied. Meanwhile, the statistical analysis will be carried out on the historical data, providing the real probabilistic knowledge to the system.

  Secondly, the states of each node in an ontology-based BN system are limited to two, i.e., true or false. However, in a real diagnosis system, each node may have more than two states. As discussed in Section 2.4.4, a thermal fault has three states, i.e., [Normal, Low temperature overheating, High temperature overheating]. To use the ontology-based BN, the original thermal fault will be split into three nodes, "thermal fault Normal", "thermal fault Low temperature overheating", and "thermal fault High temperature overheating" with six states in total. This type of transformation is meaningless, time-consuming, and increasing the complexity of the BNs. Therefore, transferred BNs with multiple states will be studied.

  Thirdly, the probabilistic constraints in this research are consistent. However, the knowledge can be obtained from different sources. For instance, as the relationship between "woman" and "man" is complementary, $P(\text{woman}) + P(\text{man}) = 1$. If two probabilistic constraints $P(\text{woman}) = 0.66$ and $P(\text{man}) = 0.88$ are obtained, the knowledge integration algorithms are proven to be non-convergent. Consequently, non-consistent constraints will

also be addressed in future works.

References

[1] National Asset Management Steering Group Australia. International infrastructure management manual, 2002.

[2] Z. Yang. *Intelligent information retrieval and fault diagnosis for the asset management of power substations*. PhD thesis, University of Liverpool, 2008.

[3] M. Beardow. Economics of asset management: drawing it together, 2003.

[4] L. Bertling, R. Allan, and R. Eriksson. A reliability-centered asset maintenance method for assessing the impact of maintenance in power distribution systems. *Power Systems, IEEE Transactions on*, 20(1):75–82, 2005.

[5] J. Crisp, D. Birtwhistle, et al. System dynamics modelling: application to electricity transmission network asset management. *Australian Journal of Electrical & Electronics Engineering*, 2(3):263, 2005.

[6] R.Merritt. Asset management keeps plants running smarter. 13(3):6, 2000.

[7] M. Judd, S. McArthur, J.R. McDonald, and O. Farish. Intelligent condition monitoring and asset management. partial discharge monitoring for power transformers. *Power Engineering Journal*, 16(6):297–304, 2002.

[8] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[9] W. H. Tang, L. Yan, Z. Yang, and Q. H. Wu. Improved document ranking

in ontology-based document search engine using evidential reasoning. *IET software*, 8(1):33–41, 2014.

[10] X. Rui and II Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, May 2005.

[11] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems 6*. Citeseer, 1994.

[12] H. Patricia and B. William. Large power transformers and the u.s. electric grid. 2012.

[13] W. H. Tang, Z. Lu, and Q. H. Wu. A bayesian network approach to power system asset management for transformer dissolved gas analysis. In *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pages 1460–1466. IEEE, 2008.

[14] W3C. *owlnamespace*, (accessed September 15, 2012). `https://www.w3.org/TR/2004/REC-owl-guide-20040210/#Namespaces`.

[15] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5):907–928, 1995.

[16] J. Q. Feng, Q. H. Wu, and J. Fitch. An ontology for knowledge representation in power systems. *Proc. of IEE Control 2004*, 35:1–5, 2004.

[17] J. Q. Feng, J. S. Smith, Q. H. Wu, and J. Fitch. Condition assessment of power system apparatuses using ontology systems. In *2005 IEEE/PES Transmission&Distribution Conference&Exposition: Asia and Pacific*, 2005.

[18] J. Xie, D. Y. Shi, Z.l. Yang, and X. Z. Duan. Research on an ontology based power system common graphics exchange approach. In *Universities Power Engineering Conference, 2007. UPEC 2007. 42nd International*, pages 290–295, Sept 2007.

[19] M. Qian, J. Z. Guo, and Y. H. Yang. An ontology for power system operation analysis. In *Electric Utility Deregulation, Restructuring and Power Technologies, 2004.(DRPT 2004). Proceedings of the 2004 IEEE International Conference on*, volume 2, pages 597–601. IEEE, 2004.

[20] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[21] K. Eguchi, H. Ito, A. Kumamoto, and Y. Kanata. Adaptive and incremental query expansion for cluster-based browsing. In *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, pages 25–34. IEEE, 1999.

[22] W. W. Chu, Z.Y. Liu, and W. L. Mao. Textual document indexing and retrieval via knowledge sources and data mining. *Communication of the Institute of Information and Computing Machinery (CIICM), Taiwan*, 5(2), 2002.

[23] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[24] E. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR94*, pages 61–69. Springer, 1994.

[25] D. Harman. Overview of the first trec conference. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 36–47. ACM, 1993.

[26] Princeton University. *WordNet: A lexical database for English*. https://wordnet.princeton.edu/, 2005.

[27] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272, 2007.

[28] M. S. Aldenderfer and R. K. Blashfield. Cluster analysis. sage university paper series on quantitative applications in the social sciences 07-044. 1984.

[29] Q. He. A review of clustering algorithms as applied in ir. *Graduate School of Library and İnformation Science University of İllinois at Urbana-Compaign*, 6, 1999.

[30] G. Kowalski. Information retrieval systems: theory and implementation. *Computers and Mathematics with Applications*, 5(35):133, 1998.

[31] C. Buckley and A. Lewit. Optimisation of inverted vector searches. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 97–110. ACM, 1985.

[32] D. Greene. *A State-of-the-art Toolkit for Document Clustering*. PhD thesis, Trinity College, 2007.

[33] G. Salton. The smart retrieval systemexperiments in automatic document processing. 1971.

[34] S. Chou, C. Y. Cheng, and S. Huang. A ranking algorithm for query expansion based on the term's appearing probability in the single document. *Online Information Review*, 35(2):217–236, 2011.

[35] M. Deepa and P. Revathy. Validation of document clustering based on purity and entropy measures. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(3):147–152, 2012.

[36] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 541–544. IEEE, 2003.

[37] S. Vega-Pons and J. Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[38] A. Topchy, A.K. Jain, and W. Punch. Clustering ensembles: models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866–1881, Dec 2005.

[39] D. Wang, W. H. Tang, and Q. H. Wu. Ontology-based fault diagnosis for power transformers. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1–8. IEEE, 2010.

[40] F. Hayes-Roth, D. Waterman, and D. Lenat. Building expert systems. 1984.

[41] L. Yan, C. H. Wei, W. H. Tang, and Q. H. Wu. Development of a novel asset management system for power transformers based on ontology. In *Power and Energy Engineering Conference (APPEEC), 2013 IEEE PES Asia-Pacific*, pages 1–6. IEEE, 2013.

[42] D. L. Ma, W. J. Zhang, and W. Yao. Establish expert system of transformer fault diagnosis based on dissolved gas in oil. In *Information Science and Cloud Computing Companion (ISCC-C), 2013 International Conference on*, pages 681–685. IEEE, 2013.

[43] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197, 1998.

[44] P. Baker, C. Goble, S. Bechhofer, N. Paton, R. Stevens, and A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6):510–520, 1999.

[45] A. Bernaras, I. Laresgoiti, N. Bartolome, and J. Corera. An ontology for fault diagnosis in electrical networks. In *Intelligent Systems Applications to Power Systems, 1996. Proceedings, ISAP'96., International Conference on*, pages 199–203. IEEE, 1996.

[46] A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. *Institute AIFB, Universität Karlsruhe*, 2003.

[47] Y. Zhang and T. Li. Consensus clustering+ meta clustering= multiple consensus clustering. In *FLAIRS Conference*, 2011.

[48] Y. Zhang and T. Li. Extending consensus clustering to explore multiple clustering views. In *SDM*, pages 920–931. SIAM, 2011.

[49] F. Jensen. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[50] A. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 205–247, 1968.

[51] L. P. Jing, L. X. Zhou, and M. K. Ng. Ontology-based distance measure for text clustering. In *Proc. of SIAM SDM workshop on text mining, Bethesda, Maryland, USA*, 2006.

[52] T. Li, C. Ding, M. Jordan, et al. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 577–582. IEEE, 2007.

[53] H. L. Luo, F. R. Jing, and X. B. Xie. Combining multiple clusterings using information theory based genetic algorithm. In *Computational Intelligence and Security, 2006 International Conference on*, volume 1, pages 84–89. IEEE, 2006.

[54] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper. Weighted cluster ensemble using a kernel consensus function. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 195–202. Springer, 2008.

[55] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288, 1986.

[56] Z. L. Ding, Y. Peng, and R. Pan. Bayesowl: Uncertainty modeling in semantic web ontologies. In *Soft Computing in Ontologies and Semantic Web*, pages 3–29. Springer, 2006.

[57] J. Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52:E15–E25, 1937.

[58] E. Cramer. Probability measures with given marginals and conditionals: l-projections and conditional iterative proportional fitting. *Statistics and Decisions-International Journal for Stochastic Methods and Models*, 18(3):311–330, 2000.

[59] S. Y Zhang, Y. Peng, and X. P. Wang. An efficient method for probabilistic knowledge integration. In *2008 20th IEEE international conference on tools with artificial intelligence*, pages 179–182. IEEE, 2008.

[60] F. Alabsi and R. Naoum. Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system. *Journal of Emerging Trends in Computing and Information Sciences*, 3(7):1053–1058, 2012.

[61] Tim B. L. *HTML*, (accessed April 22, 2012). `http://www.w3.org/TR/2000/CR-rdf-schema-20000327/`.

[62] N. Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[63] C. B. Fu, G.H.and Jones and A. I. Abdelmoty. Ontology-based spatial query expansion in information retrieval. In *On the move to meaningful internet systems 2005: CoopIS, DOA, and ODBASE*, pages 1466–1482. Springer, 2005.

[64] D. Bonino, F. Corno, L. Farinetti, and A. Bosca. Ontology driven semantic search. *WSEAS Transaction on Information Science and Application*, 1(6):1597–1605, 2004.

[65] L. M. Chen, N. R. Shadbolt, and C. Goble. A semantic web-based approach to knowledge management for grid applications. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):283–296, 2007.

[66] R. Navigli, P. Velardi, and A. Gangemi. Ontology learning and its application to automated terminology translation. *Intelligent Systems, IEEE*, 18(1):22–31, 2003.

[67] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information processing & management*, 43(4):866–886, 2007.

[68] W3C. *Resource Description Framework (RDF)*, (accessed September 31, 2013). `http://www.w3.org/RDF/`.

[69] W3C. *Extensible Markup Language (XML)*, (accessed October 3, 2013). `http://www.w3.org/XML/`.

[70] W3C. *Extensible Markup Language (XML) Schema*, (accessed October 20, 2013). `http://www.w3.org/XML/Schema`.

[71] W3C. *Validation Service*, (accessed May 3, 2012). `http://www.w3.org/RDF/Validator/`.

[72] W3C. *Resource Description Framework (RDF) Schema Specification 1.0*, (accessed November 3, 2013). `http://www.w3.org/RDF/`.

[73] P. Michael. *The DARPA Agent Markup Language*, (accessed October 3, 2015). `http://www.daml.org//`.

[74] D. Fensel, I. Horrocks, F. Van, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pages 1–16. Springer, 2000.

[75] W3C. *Annotated DAML+OIL Ontology Markup*, (accessed May 31, 2012). `http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218/`.

[76] W3C. *OWL Web Ontology Language Reference*, (accessed October 3, 2015). `http://www.w3.org/TR/owl-ref/`.

[77] W. V. Quine. On what there is. *The Review of Metaphysics*, 2(1):21–38, 1948.

[78] B. Smith. Ontology: philosophical and computational. *The Blackwell Guide to the Philosophy of Computing and Information. Blackwell, Oxford*, 2003.

[79] L. W. Lacy. *OWL: representing information using the Web Ontology Language*. Trafford Publishing, 2005.

[80] W3C. *Proétgé*, (accessed October 13, 2012). `http://protege.stanford.edu/`.

[81] W3C. *Semantic Web Homepage*, (accessed September 15, 2015). `http://protegewiki.stanford.edu/wiki/OWLViz`.

[82] F. Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

[83] J. F. Sowa. *Principles of Semantic Networks: Explorations in the representation of knowledge*. Morgan Kaufmann, 2014.

[84] M. Minsky. A framework for representing knowledge. 1975.

[85] *OWL DL Semantics*, (accessed October 20, 2015). `http://www.obitko.com/tutorials/ontologies-semantic-web/owl-dl-semantics.html`.

[86] V. Bush. The atlantic monthly. *As we may think*, 176(1):101–108, 1945.

[87] C. N. Mooers. *Application of random codes to the gathering of statistical information*. PhD thesis, Massachusetts Institute of Technology, 1948.

[88] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317, 1957.

[89] C. Cleverdon. The cranfield tests on index language devices. In *Aslib proceedings*, volume 19, pages 173–194. MCB UP Ltd, 1967.

[90] C. Cleverdon. Evaluation tests of information retrieval systems. *Journal of Documentation*, 26(1):55–67, 1970.

[91] Yahoo. *Yahoo*, (accessed March 3, 2016). `https://uk.yahoo.com/`.

[92] Google. *Google*, (accessed March 3, 2016). `https://www.google.co.uk/`.

[93] B. Peng. *On efficiency optimization and effectiveness evaluation of search engine retrieval System*. PhD thesis, Peking University, 2004.

[94] F. W. Lancaster and E. Gallup. Information retrieval on-line. Technical report, 1973.

[95] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

[96] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

[97] S. E. Robertson and S. Walker. Okapi at trec-1. In *TREC*, volume 8, pages 21–30, 1992.

[98] J. Miao, J. Huang, and Z. Ye. Proximity-based rocchio's model for pseudo relevance. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 535–544. ACM, 2012.

[99] A. L. Gançarski, A. Doucet, and P. R. Henriques. Attribute grammar-based interactive system to retrieve information from xml documents. *IEE Proceedings-Software*, 153(2):51–60, 2006.

[100] C. Manning, P. Raghavan, H.h Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[101] C. Doug. *Apache Lucene*, (accessed May 23, 2012). `http://lucene.apache.org/`.

[102] J. Nocedal and S. Wright. *Numerical optimisation*. Springer Science & Business Media, 2006.

[103] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[104] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[105] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[106] M. Avriel. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[107] B. Jarraya and A. Bouri. Metaheuristic optimization backgrounds: a literature review. *International Journal of Contemporary Business Studies*, 3(12), 2012.

[108] K. Sörensen and F. W. Glover. Metaheuristics. In *Encyclopedia of operations research and management science*, pages 960–970. Springer, 2013.

[109] D. I. Boussaï, J. Lepagnot, and P. Siarry. A survey on optimisation metaheuristics. *Information Sciences*, 237:82–117, 2013.

[110] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[111] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[112] V. Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.

[113] J. Hoffmann. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. In *Foundations of Intelligent Systems*, pages 216–227. Springer, 2010.

[114] L. Schmitt. Theory of genetic algorithms. *Theoretical Computer Science*, 259(1):1–61, 2001.

[115] G. Syswerda. Uniform crossover in genetic algorithms. 1989.

[116] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(4):656–667, 1994.

[117] Owlviz. `http://protegewiki.stanford.edu/wiki/OWLViz`.

[118] B. McBride. Jena: Implementing the rdf model and syntax specification. In *SemWeb*, 2001.

[119] W. H. Tang and Q. H. Wu. *Condition monitoring and assessment of power transformers using computational intelligence*. Springer Science & Business Media, 2011.

[120] S. Reed, Y. Petillot, and J. Bell. Automated approach to classification of mine-like objects in sidescan sonar using highlight and shadow information. In *Radar, Sonar and Navigation, IEE Proceedings-*, volume 151, pages 48–56. IET, 2004.

[121] L. H. Min and C. S. Chang. Application of dempster-shafer's theory of evidence for transformer incipient fault diagnosis. 2009.

[122] A. Awasthi and S. S. Chauhan. Using ahp and dempster–shafer theory for evaluating sustainable transport solutions. *Environmental Modelling & Software*, 26(6):787–796, 2011.

[123] J. B. Yang and M. G. Singh. An evidential reasoning approach for multiple-attribute decision making with uncertainty. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(1):1–18, 1994.

[124] L. S. Guo, C. X. Guo, W. H. Tang, and Q. H. Wu. Evidence-based approach to power transmission risk assessment with component failure risk analysis. *IET generation, transmission & distribution*, 6(7):665–672, 2012.

[125] S. A. Farghal, M. S. Kandil, and A. Elmitwally. Quantifying electric power quality via fuzzy modelling and analytic hierarchy processing. *IEE Proceedings-Generation, Transmission and Distribution*, 149(1):44–49, 2002.

[126] D. Anderson, D. Sweeney, T. Williams, J. Camm, and J. Cochran. *An introduction to management science: quantitative approaches to decision making*. Cengage Learning, 2015.

[127] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[128] W. M. Lindhoud. *Automated fault diagnosis at Philips medical systems: a model-based approach*. PhD thesis, MSc thesis, Delft University of Technology, Delft, The Netherlands, 2006.

[129] T. A. S. Coelho, Pável P. Calado, L. V. Souza, B. Ribeiro-Neto, and R. Muntz. Image retrieval using multiple evidence ranking. *Knowledge and Data Engineering, IEEE Transactions on*, 16(4):408–417, 2004.

[130] J. Huang, J. Miao and B. He. High performance query expansion using adaptive co-training. *Information Processing & Management*, 49(2):441–453, 2013.

[131] T. Hastie, R. Tibshirani, and J. Friedman. *Unsupervised learning*. Springer, 2009.

[132] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

[133] H. F. Lau and M. D. Levine. Finding a small number of regions in an image using low-level features. *Pattern Recognition*, 35(11):2323–2339, 2002.

[134] H. Zhong, J. B. Shi, and M. Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–819. IEEE, 2004.

[135] G. Punj and D. W. Stewart. Cluster analysis in marketing research: review and suggestions for application. *Journal of marketing research*, pages 134–148, 1983.

[136] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[137] A. Topchy, A. Jain, and W. Punch. A mixture model for clustering ensembles. In *Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining*, page 379. Society for Industrial and Applied Mathematics, 2004.

[138] R. J. G. B. Hruschka, E. R. Campello and A. A. Freitas. A survey of evolutionary algorithms for clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(2):133–155, March 2009.

[139] B. Fischer and J. M. Buhmann. Bagging for path-based clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11):1411–1415, 2003.

[140] A. P. Topchy, M. H. Law, A. K. Jain, and A. L. Fred. Analysis of consensus partition in cluster ensemble. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 225–232. IEEE, 2004.

[141] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorisation.

[142] T. Ding, C. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[143] X. F. Ding, C. He and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610. SIAM, 2005.

[144] S. Régnier. Sur quelques aspects mathématiques des problèmes de classification automatique. *Mathématiques et Sciences humaines*, 82:13–29, 1983.

[145] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006.

[146] J. Vega-Pons, S. Correa-Morris and J. Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition*, 43(8):2712–2724, 2010.

[147] J. Handl, J. Knowles and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[148] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[149] A. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):835–850, 2005.

[150] E. D. Cristofor. *Information-theoretical methods in clustering*. PhD thesis, Office of Graduate Studies, University of Massachusetts Boston, 2002.

[151] F. Beil and X. W. Ester, M.and Xu. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–442. ACM, 2002.

[152] D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics, 1990.

[153] S. A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 120–126. Association for Computational Linguistics, 1999.

[154] P. Velardi, P. Fabriani, and M. Missikoff. Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 270–284. ACM, 2001.

[155] P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res.(JAIR)*, 24:305–339, 2005.

[156] Z. S. Harris. Mathematical structures of language. 1968.

[157] J. K. Cullum and R. A. Willoughby. Real symmetric matrices. In *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Vol. II Programs*, pages 11–118. Springer, 1985.

[158] D. J. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT press, 2001.

[159] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole. On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence–SBIA 2004*, pages 405–413. Springer, 2004.

[160] L. W. Zhang and H. P. Guo. Bayesian network introduction. *Structure Learning*, pages 172–191, 2006.

[161] J. Pearl. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257, 1987.

[162] S. Russell and P. Norvig. Artificial intelligence: a modern approach. 1995.

[163] J. T. Quan, L. Ruan, and Z. C. Xie. The application of bayesian network theory in transformer condition assessment. In *Power and Energy Engineering Conference (APPEEC), 2013 IEEE PES Asia-Pacific*, pages 1–4. IEEE, 2013.

[164] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[165] K. B. Korb and A. E. Nicholson. *Bayesian artificial intelligence*. CRC press, 2010.

[166] A. Calì and T. Lukasiewicz. *An approach to probabilistic data integration for the semantic web*. Springer, 2008.

[167] J. Vomlel and R. Vomlel. Methods of probabilistic knowledge integration. 1999.

[168] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86, 1951.

[169] H. H. Bock. A conditional iterative proportional fitting (cipf) algorithm with applications in the statistical analysis of discrete spatial data. *Bull. ISI, Contributed papers of 47th Session in Paris*, 1:141–142, 1989.

[170] J. P. Pellet and A. Elisseeff. Using markov blankets for causal structure learning. *The Journal of Machine Learning Research*, 9:1295–1342, 2008.

[171] Norsys. *Norsys software corp.*, (accessed September 16, 2012). `https://www.norsys.com/netica.html`.

[172] K. Murphy. *Software Packages for Graphical Models*, (accessed September 15, 2012). `http://www.cs.ubc.ca/~murphyk/Software/bnsoft.html`.