Architecture for Intelligent Big Data Analysis based on Automatic Service Composition

Thenuwara Hannadige Akila Sanjaya Siriweera

A DISSERTATION
SUBMITTTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Graduate Department of Computer and Information Systems

The University of Aizu

2019

The thesis titled

Architecture for Intelligent Big Data Analysis based on Automatic Service Composition

by

Thenuwara Hannadige Akila Sanjaya Siriweera

is reviewed and approved by:

Chief Referee
Professor
Incheon Paik — Jan. 31, 2019

Professor
Qiangfu Zhao — Jan. 31, 2019

Professor
Alexander P. Vazhenin — Feb 1, 2019

Professor
Keitaro Naruse — Jan. 31, 2019

The University of Aizu

2019

This page intentionally left blank.

I dedicate this to

- My mother, father

  and

- The father of free education, Sri Lanka

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

Glossary

ASC	Automatic Service Composition
BDA	Big Data Analysis
CRSIP-DM	CRoss Industry Standard Process for Data Mining
WS	Web Service
CWS	Composite WS Service
RA	Reference Architecture
SA	System Architecture
UML	Unified Modelling Language
GP	Graph Plan
HTN	Hierarchical Task Network
Mutex	Mutually Exclusive
Mutin	Mutually Inclusive
HDFS	Hadoop Distributed File System
GSSN	Global Social Service Network
LSSN	Linked Social Service Network
TS	Transaction Service
QoS	Quality of Service
CTQoS	Customizable Transaction and QoS
TSP	Transaction Service Properties
GA	Genetic Algorithm
MCKP	Multi-Constraint Knapsack Problem
ABC	Artificial Bee Colony
MR	Map Reduce
DAG	Directed Acyclic Graph
PoW	Partial order Workflow
POP	Partial Order Planner
SoPOP	Service-oriented Partial Order Planner

This page intentionally left blank.

Preface

This thesis presents my work for the fulfillment of the requirement for the Doctor of Philosophy in Computer Science and Engineering, Graduate School of Computer Science and Engineering, the University of Aizu, Japan. The study was carried out in the period from April 2016 to March 2019.

This page intentionally left blank.

Abstract

Big data analytics (BDA) is the preferred analytical approach to managing high volumes of highly varied data generated at high velocity (3V-data), which is becoming prevalent in the data sciences. Moreover, BDA is evolving for many V's (mV-data) such as high variability and veracity and this leads to appearing field such as Deep Learning. It is an extreme challenge to store and process the mV-data. Moreover BDA process consumes mV-data is raising extreme concerns. Understanding, addressing quality, dealing with outliers, modeling for analysis and displaying meaningful results are considered as concerns with respective to the data science. This implies, BDA process is diversely stepped, heavily resource and time consuming job. These two limitations are hampering the meaningful adoption of the BDA across the research and industry domains. Therefore, automating the BDA is a cognitive approach for the research and industry, which are suffering most. Besides that dramatic expansion of services related to data analysis shows a bright prospect in data science. Then service composition becoming the preferred platform for the BDA. Therefore we proposed a novel architectural design process to automate the BDA process based on automatic service composition (ASC), with the cross-industry standard process for data mining being used as the data science underpinning BDA. Proposed ASC comprised five main stages, which are, planning, discovery, selection, verification-refinement and execution. Each stage is comprised of respective stage-specific one or more combinations of concerns, such as constraint-awareness, NP-complete and domain-specific concerns. Then it is essential to address respective concerns in adoptable for constraints, heuristic and ready to comply with domain concerns. Consequently, we proposed dedicated solutions for each stage of the ASC process. We employed AI techniques for the concerns, which are encountered during the process. Our experiments demonstrate that the proposed solutions are well behaved and efficiently facilitate to accomplish to satisfy the overall architecture to automate intelligently, adoptable and effectively satisfy the BDA requirement.

This page intentionally left blank.

Chapter 1 Introduction

Data populates exponentially in various dimensions such as high volume, velocity, and variety (3V) are called as the popular term of Big Data. Nevertheless, the Big Data leverages to the additional dimensions in addition to the 3V are high variability and need for high veracity (5V), which is appearing in fields such as deep learning [1], [2] and continuously evolving the connotation of the Big Data for higher degree of V's [3]. Human and machines are two sources of Big Data. That implies, the Big Data pours from every conceivable direction. Cisco claimed, we already reached the Zettabyte-era (10^{27} or 2^{70} : ZB) in 2015 from the Terabyte-era (10^{12} or 2^{40} : TB) perspective to the volume of the internet traffic.[1] Big Data raises an extreme challenge to the data science with respective to the Big Data analysis (BDA). Nevertheless, BDA shows bright prospect to the gateway to the Big-Money, therefore research and business domains prefers the BDA [4], [5].

However, recent data science techniques such as BDA process consume excessive resources and time. These limitations are hampering the meaningful adoption of the data science across research and industry domains [6]–[8]. However, such a data science processes require highly diverse and rigorous stages to be successful, which involves very large resources and makespan (the overall time for the process). This constrains the full meaningful effects of a state-of-the-art data science product. Automating the data analytics process offers one of the most cognitive solutions to these

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

concerns [1], [9]. Automation of data analytics such as BDA may enable data scientists to accomplish their task in days rather than the traditional period of months.[2]

BDA arises complex situation in its data mining process due to the diversified analytical requirements and multi-disciplinary data set's. We have to select a comprehensive data mining methodology to fulfil its diversified requirements in efficient way. We can choose a data-mining method for BDA and other necessary procedure using our experiences. CRoss Industry Standard Platform for Data Mining (CRISP-DM) is a useful standard for BDA [10]. However, the manual process of steps in CRSIP-DM for BDA hinders faster decision making on real time application for efficient data analytics. Further, CRISP-DM has to pass thorough and rigorous steps to complete a successfully the data-mining process, such as understanding data, addressing the data quality, dealing with outliers, modeling for analysis and displaying meaningful results etc.

As the first step of the research, we propose a novel architecture to automate BDA process intelligent manner based on Automatic Service Composition (ASC). Mainly it consists with 3 level of architectural views. Reference Architecture (RA), System Architecture (SA) and high level UML class diagram.

As the second step onwards, we have successfully achieved the respective stages of the ASC comprises with five main stages; which are planning stage, discovery stage, selection stage, verification-refinement (VR) stage and execution stage. Each stage of the ASC process encountered domain specific concerns and therefore we addressed those concerns considered them as the domain-specific multi-objective problems.

1.1 Architecture for Intelligent Big Data Analysis (BDA)

BDA is diverse stepped, multiple resource and time consuming process. It has to be passed various stages which are depending on each other’s to do a successful analytical

[2] http://news.mit.edu/2016/automating-big-data-analysis-1021

process. Further, it has to be paid serious attention and maintained standard discipline in the procedure of design and development process of the BDA. Therefore we are undertaking standard software design and development discipline to automate the BDA. An Architecture is playing the pivot role in software design and development paradigm. A software architecture is about making fundamental structural choices which are costly to change once implemented. Therefore we initiate with template architectural design, which is called RA. The RA is mostly used in enterprise level to cope with cumbrous software design and development projects. This initial RA can be used to test and evaluate problem domain with technologies considering various aspects of the whole process. Then we could be able to result a matured RA and rest of architectural design process could be able surpassed based on the matured RA. Therefore we could be able to avoid faults and lapses due to the architectural design. These were proved in SA design process. Because we could be able to maximally avoid difficulties in deriving a SA for our scenarios. Proposed approach is extended solution of our previous work [11] for the BDA domain.

1.2 Automatic Service Composition (ASC) for Intelligent BDA

In our research, we are exploring to automate CRISP-DM process using the ASC. As mentioned earlier, the ASC consists of five stages: planning, discovery, selection, VR and execution. The core idea is to orchestrate existing services to achieve a larger task, resulting in a new composite and value-added web service. For the given data analytics scenario, the goal is to find a suitable composition plan according to the client's particular requirements, without checking all combinations of all services and tasks. In the planning stage, an abstract workflow is constructed. Afterwards, the rest stages aim to identify optimal service composition for the identified abstract tasks. The performance of service composition is mainly based on the abstract workflow generated in the planning stage.

Service discovery is acting a key role as the second stage of the ASC process. In this stage, we focus on discovering most suitable set of services from the registry for each tasks of the workflow results by the planning stage. We have worked to achieve the discovery by considering two main factors, which are perspective to precise task and effective workflow.

Service selection is the third main stage of the ASC process. Here process selects the most cognitive service for the given tasks based on the respect QoS aware requirement given by the user. We have accomplished to achieve the selection by considering two key requirements considering end-user requirements. The first one is QoS, dynamic transaction awareness, the second one is the QoS and traffic awareness in the Big Data space.

VR is the fourth main stage of the ASC process. In this, we focus on preparing total order planner (TOP) by verifying and refining the composition plan directed by selection stage. We have accomplished to achieve the VR stage, which is dynamic constraints aware composite web service (CWS) based process.

Execution stage is the final stage of the ASC. In this stage, we execute the TOP, which is resulted by the VR stage that comprises order of sequence of CWS's to complete process of the ASC.

1.3 Original Contributions

Mainly our research is to Architecture for intelligent BDA. Following contributions are the key contributions in our research.

1. Architecture for intelligent BDA based on ASC
2. Planning stage
    a. Constraint driven dynamic workflow for the BDA based on Graphplan technique.
3. Discovery stage
    a. Domain ontology based service discovery

b. Social service network (SSN) with multiple-feature attributes based parameterized service discovery

4. Selection stage
    a. QoS and customizable transaction-aware selection for BDA
    b. QoS-aware rule-based traffic-efficient multiobjective selection in Big Data space
5. VR stage
    a. Constraint-aware service oriented partial order planner

1.4 Thesis Organization

The thesis mainly consists of four parts as shown in Figure 1.1.

In the chapter 2, the background of the study will be presented. We discuss the works related to the each area.

Part I: Background and Related Works

Chapter 2 presents the background study of architecture for BDA, and the ASC process. In chapter 2.1, we present the overview of the BDA process. And Chapter 2.2 also explained the overview of key technologies used in the overall process. Here we are discussed motivation scenario, CRISP-DM and ASC in respective Chapters 2.2.1, 2.2.2 and 2.2.3. Chapter 2.3 discuss the literature review on BDA architecture. Chapter 2.4, 2.5, 2.6, 2.7, and 2.8 are discussed the literature reviews of respective stages of the ASC; planning, discovery, selection, VR and execution.

Part II: Architecture for Intelligent BDA

Chapter 3 presents motivation for the architecture for the BDA and explain architecture deriving process in detail. We explain the procedure of architectural decision made and stages we went through the architectural design. In chapter 3.1, we present the architectural design process, Chapter 3.2 and 3.3 discussed reference and system architectures of the proposed method.

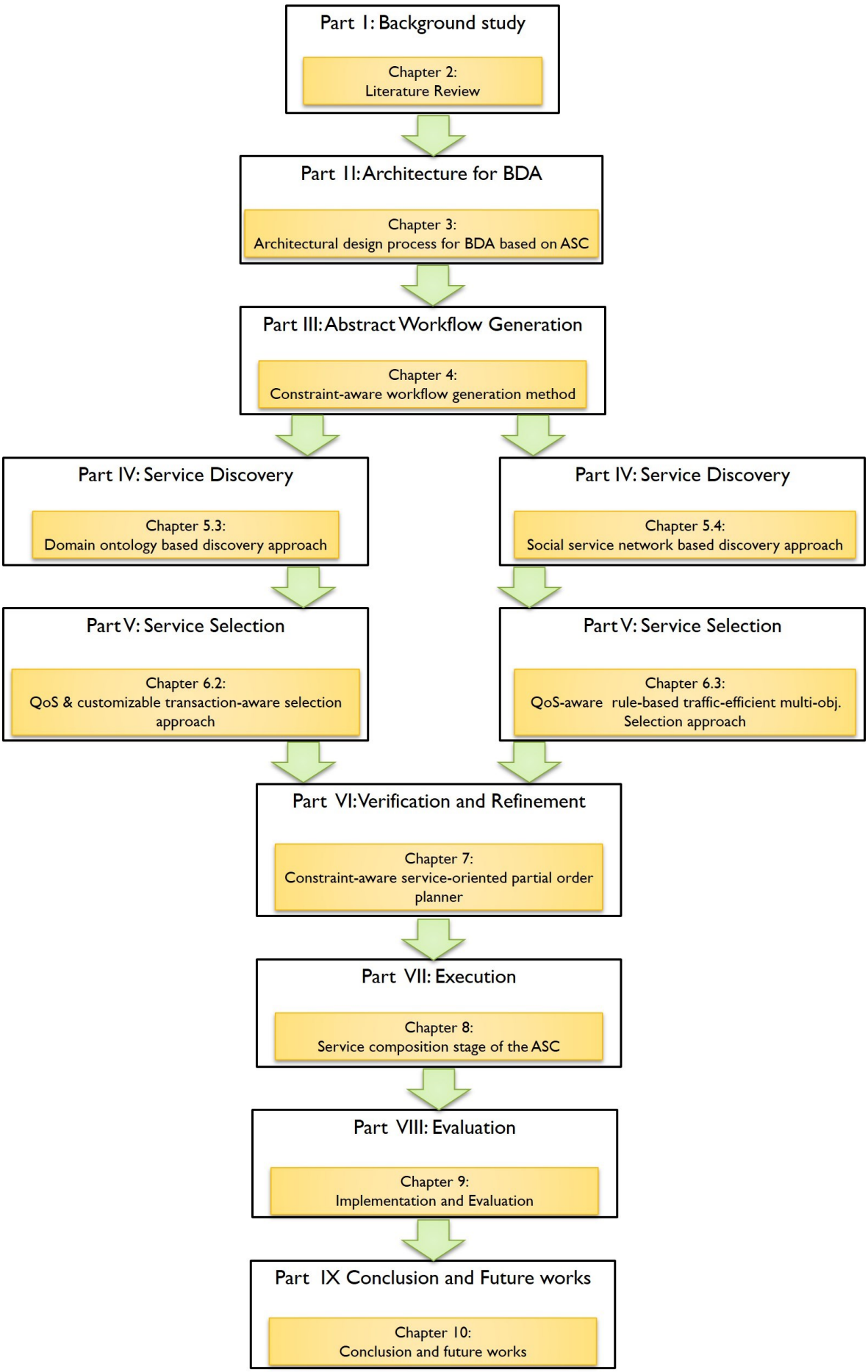


Figure 1.1: Thesis organization

Part III: Planning Stage of the ASC

Chapter 4 presents motivation for the planning stage implementation and the way we achieved the planning stage.

Part IV: Discovery Stage of the ASC

In chapter 5.1, we discussed the motivation scenarios for the discovery stage. Chapter 5.2, we discuss the introduction, which is applied for the both methods. Next Chapter 5.3 discuss the domain ontology based service discovery method and Chapter 5.4 discuss social service network based service discovery method.

Part V: Selection Stage of the ASC

In chapter 6.1, we discussed motivation scenarios for the service selection. Chapter 6.2 discuss the QoS and dynamic transaction-aware service selection method. Chapter 6.2 discussed the QoS-aware rule-based traffic-efficient multiobjective selection in Big Data space.

Part VI: VR Stage of the ASC

In Chapter 7.1, we discussed motivation scenario for the VR stage. Chapter 7.2 discuss the constraint aware service-oriented partial order planner.

Part VII: Execution Stage of the ASC

In Chapter 8.1, objectives and motivation scenario for the execution stage.

Part VIII: Experiments and Evaluation

In chapter 9, implementation and evaluation of our proposed architecture. Respective Chapter 9.1, 9.2, 9.3, 9,4, 9.5 and 9.6 discuss the respective experiments and evaluations of architecture, planning, discovery, selection, VR and execution stages of the ASC process.

Part IX: Conclusion and Future Works

In chapter 10, the thesis concluded and the future works are presented.

Chapter 2 Background and Related Works

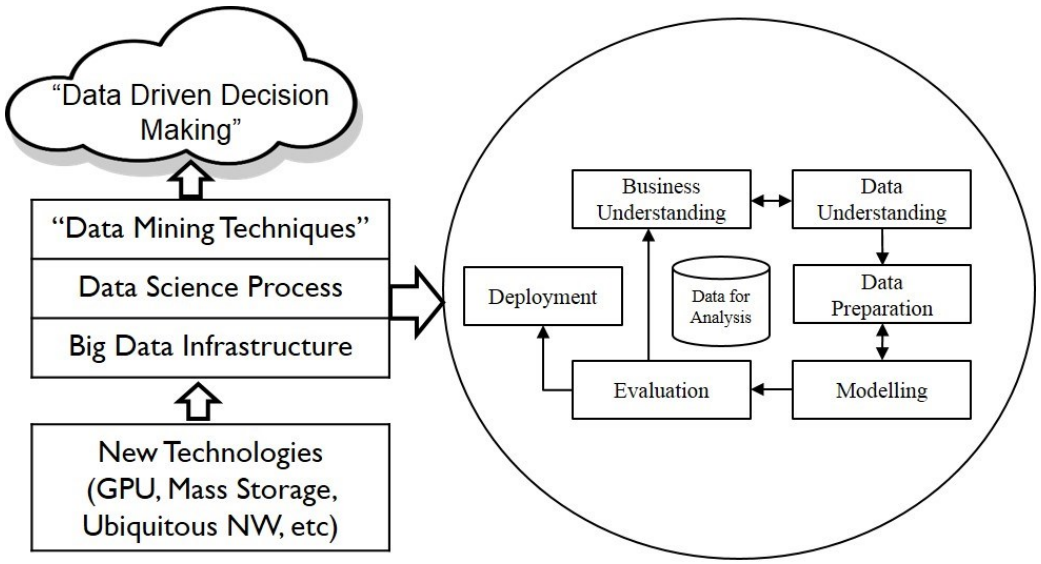
In this chapter, first we present overview of the Big Data Analytics. Chapter 2.1 discuss overview of the architecture and in chapter 2.2 discuss overview of the technologies used. Chapter 2.3 discuss the related works for the architecture for the BDA. In chapter 2.4, we discuss overview of the planning stage and related works in that. And finally chapter 2.5 discuss overview of the discovery stage and the related works in the discovery. Chapter 2.6 discuss the related works of the selection stage and Chapter 2.7 discuss the related works of the VR stage. Finally Chapter 2.8 discuss the related works of the execution stage of the ASC process.

2.1 Overview of the BDA

BDA is the process of collecting, organizing and analyzing large sets of data to discover patterns and other useful information. BDA helps organizations to better understanding the information contained within the data and help to identify the data that is most important to the business and future business decisions. Data warehouse will be processed by data science technology and mined by data mining techniques. Data manipulation will be done by a data science process and here we use, CRISP-DM. Figure 2.1 displays the high-level view of the standard process of the BDA. Fig. 2.1.a shows the standard data science processes, while Fig. 2.1.b shows how CRISP-DM conducts detailed data mining on stored data from a certain infrastructure.

Conventional BDA is a diverged stepped process due to the complex data science

process. Initially it has to be clarified the business requirement that is seeking to do an analytics based on the Big Data infrastructure. Next, it has to be selected most opportunistic data, which is tallying with the requirement. After that, it needs to be purified data set. In this case it has to go through thorough purification process to feed the analytical process. Some of the major steps of in the purification process are dealing with outliers, handle inconsistent, missing values, noise, and synonyms and handle uncertainty etc. After that, it needs to prepare to feed the Hadoop infrastructure and continue analytical process based on the model designed for the analytics. In simply, this is the reason to break the whole process in to diverge stepped, consumes many resources and time.



a. Standard data science process b. Stages of the CRISP-DM

Figure 2.1: Standard Process of the BDA

2.2 Overview of the Technologies used in

Here we discuss two key technologies we used in this process. Which are CRISP-DM and the ASC. BDA comprises two types of analysis, those are batch processing and real time processing. Batch processing conducts using high volume of stored collection of data after certain time of period. Batch processing involves in generating canned reports, interest accruals processing, forecasting and customer

profiling etc. Real time processing conducts using data which collects in real time. Here it involves continual input. Real time processing is popular analytical method for complex event processing, operational intelligence and continuously running programs which consumes stream data. We explain these two based on two motivation scenario. It will ease understanding the behaviors' of these two technologies within our architecture. Fig. 2.2 and 2.3 show the scenarios for the BDA.

2.2.1 Motivation Scenario:

2.2.1.1 Batch processing

A Japanese national institute dealing with advanced industry is initiating many projects in renewable-energy fields, one of which is conducting a weather-data analysis program in Fukushima Prefecture to investigate effective renewable-energy sources. Their aim is to pro-mote adoption of renewable energy, to find the most efficient energy sources and to decommission or maximally reduce the dependency on existing nuclear power plants in northern Japan. In this project, the institute is collaborating with two universities: Fukushima University and the University of Aizu. One of the contributions, involving a smart grid and energy IoT areas [11.

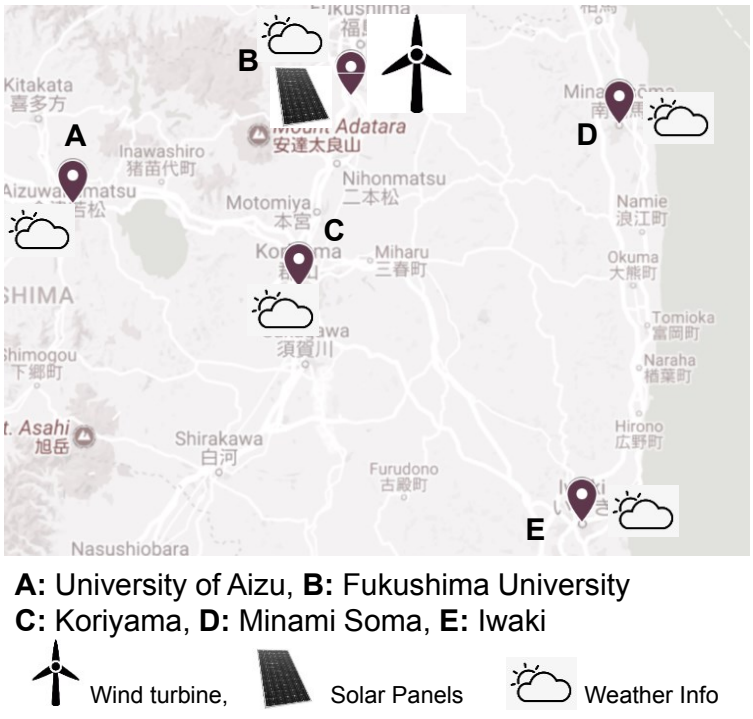


Figure 2.2: Batch processing scenario for the sustainable power solution

2.2.1.2 Real time processing

Japan metrological department is conducting a research to design a real time earthquake detection model and send lifesaving alerts to relevant authorities about the earthquake which are tend to be exceed magnitude 6 or above. The solution should be improve with machine learning to provide near real-time computation results. It needs to handle multiple sources of data. Solution should be easy to use Bulk transmission of alerts within 60 seconds of the foreshock. System should collect real time data, do analytics and send alters within given limited time of period to the respective earthquake prone areas.

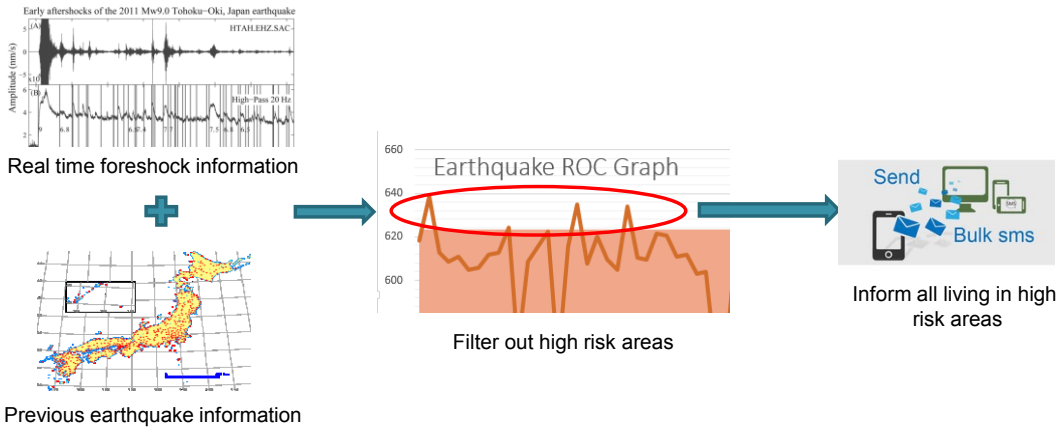


Figure 2.3: Real time processing scenario for the earthquake detection

2.2.2 CRISP-DM Process

In this section, we explain the CRISP-DM process and then we present motivating scenario for BDA. BDA aims to support decision making by discovering patterns and other useful information from large set of data, as shown in Figure 2.1. Standard data science processes, such as CRISP-DM, will conduct various data mining over data stored on certain infrastructure. And also Figure 2.1.b illustrates the six step-wise phases of operating CRISP-DM over big data. The initial phase is business understanding, which aims to understand the project objectives and requirements from the business domain. In the data-understanding phase, data scientists proceed with activities to get familiar with the data and identify data quality problems. The data preparation step covers all activities that prepare the raw data in order to yield the final dataset to be fed into modeling tool. In the modeling step, various modeling techniques

are applied to analyze the dataset. Before proceeding to the final deployment of the model outcome, a thorough evaluation phase is needed to ensure whether it meets the business requirements. Here onwards, we employs batch processing scenario to describe internal process of the CRISPDM process.

Business Understanding: The ultimate goal of the study is to find optimal renew-able energy sources that can reduce or halt the use of nuclear power plants. Towards this goal, the main objective is to determine the energy factors and their sustainability. To realize this objective, the researchers decided to create two profiles for renewable energy sources across Fukushima Prefecture. The first profile involves weather data collected from five locations and the second profile involves energy data collected from one location, as shown in Fig. 2.2.

Data Understanding: The first profile contains six types of weather data collected from five locations across Fukushima Prefecture: i.e., irradiance, temperature, wind direction, wind speed, humidity, and pressure. The second profile constrains three types of power data, i.e., voltage, current, temperature of the panel's photovoltaic surface, and wind turbine data.

Data Preparation: First, researchers identify the core influential factors from the weather and power data generated for the two profiles. For example, the weather profile includes irradiance, temperature, and wind speed, and the power profile includes voltage and current. The identified influential factors are then treated as variables upon which cluster analysis is performed. During this phase, some data preprocessing tasks are performed, including data-format conversion, missing-value handling, outlier handling, and Hadoop data preparation.

Modeling: Researchers feed the preprocessed data set into a big-data file system, such as the Hadoop Distributed File System (HDFS), and attempt to build a sophisticated model. Typical candidate methods are clustering and classification algorithms. For example, clustering is a natural method for generating types of segmented pro-files that contain algorithm classes. The researchers can then select an appropriate clustering algorithm, such as K-means. By tuning the parameters of the algorithm, such as the

number of clusters and maximum iterations, different clustering results may be obtained.

Evaluation: Various analytic techniques, such as discriminant analysis, can be used to verify the "reality" of the resulting cluster or classification results. Pre- and post-evaluations of the model have been proposed because the BDA results are highly sensitive and may influence decisions made during serious crises in the region.

Deployment: After the evaluation phase, the model can be deployed by the respective authorities, which then generate the required reports to the Japanese government. Based on the subsequent results, the advance industry institution can propose the most sustainably optimal power solution to the power crisis in northern Japan, especially in Fukushima Prefecture.

We propose automating the last four stages (data preparation, modelling, evaluation, and deployment) of the CRISP-DM. Each CRISP-DM stage is constrained by the prior stage. The stages are recallable based on explicit constraints, such as the expected accuracy of the evaluation stage.

Classical CRISP-DM can recall from either modeling to data preparation or evaluation to business understanding stages. However, considering BDA's domain-specific concerns in the automation process, scheduled automation of our planning process may require recall from modeling to data preparation, deployment to data preparation, and data preparation for deployment. The latter two recallable options are additions to classical CRISP-DM that aim to improve the deployed model at the production site by using new data to mature and update the model. This implies a data preparation to deployment recall facility. In addition, a production-level evaluation should be included in the deployed model. If the model fails or shows any malfunctioning, it should be readjusted using different data with the previously used parameters. Therefore, we propose to include a deployment to data preparation recall facility. In this way, we hope to improve the quality of BDA automation and affirm the quality of the BDA process.

2.2.3 ASC for CRISP-DM

ASC is a well-known technology for automating a diverse range of applications [12]. The ASC must handle the highly dynamic and constraint-oriented BDA problem domain in a sophisticated manner. The proposed ASC comprises five main stages: i.e., the *planning*, *discovery*, *selection*, *VR*, and *execution stages*.

Planning stage: This stage prepares the abstract workflow for analysis by considering BDA requirements and initial constraints during the process explained in the Chapter 4.

Discovery stage: This stage identifies candidate services to the workflow using the social service network with domain ontology and workflow awareness to satisfy the functional requirement of the given tasks in the workflow, as we discussed in Chapter 5.

Selection stage: This stage selects the candidate service from among the discovered services that best satisfy the quality of service (QoS) requirements. Here we mainly focused on two types of selections, which are customizable transaction aware and traffic efficient selection in Big Data space. We discussed in Chapter 6.

VR stage: This stage is essential to an ASC in the BDA domain. Therefore, this stage is newly proposed as an intermediate stage between the *selection* and *execution stages* initially proposed by Paik et al. [12]. At the beginning, VR stage converts the selected-service network into a partial order workflow (PoW) problem. The *verification* process inspects each planning requirement's selected services for *flaws (open goals* and *threats).* The *refinement* process addresses to the *flaws* and refines the PoW to achieve the original requirement of seamless automation. Then refined planner we called as the total order planner (TOP), which proceeds according to the status it has been achieved. *Status1: refined with new task POP* refers to a flawed PoW refined with a new task, requiring the POP to feed back to the discovery stage to discover services to the newly introduced. *Status2: refined with causal link POP* refers to a flawed PoW refined only with a *causal link* rather than a new task, enabling it to provide an appropriate out-linked support file and feed forward to the *execution stage*. *Status3: no refinement POP* refers to a flawless POP, which means the workflow can be considered concrete and

can feed forward to the *execution stage*. *Status4: VR failure* requires that the process feed back to the *planning stage.*

Execution stage: This stage executes the selected services and invokes the results. Execution stage automates the composition using exception handling transaction-aware service composition [2], [13]. It recalls the VR stage whenever it fails to achieve any of the preconditions defined in the various sub-goals.

2.3 Architecture for Intelligent BDA

Architecture is the first and most readable interpret real world problem in to the technical language. It needs well-structured and disciplined architectural design to the BDA process. It eases and smoothly directs the members which are involving in the design and development process of the BDA solution.

The literature on scalable intelligent architecture for BDA is scarce. But there are several models are introduced by researchers. There is an architecture, which provided reference architecture for the BDA and it results an indicative evidence [5]. It aims predictive analytics process. It does not consider the process automation or intelligent approach for the BDA process. Since it did not mention about further architectural design levels such as SA or UML diagram etc. In addition, intelligent multi agent solution provided for particular domain [6].

A memory centric real-time BDA also introduced and explained [7]. Health related real-time BDA solution for monitoring purposes discussed in detail [8]. And it is providing predictive BDA for aviation industry with considering various factors of the aviation industry [9]. Wu et al. [10] presented a HACE theorem that characterizes the features of the Big Data revolution, and proposes a Big Data processing model. They have analyzed several challenges at the data, model, and system levels to explore the Big Data. Shang et al. [11] proposed an approach to assist developers of BDA Apps for cloud deployments. They have also proposed a lightweight approach for uncovering differences between pseudo and large-scale cloud deployments.

Koliopoulos et al. [12] designed distributed framework for Weka, which is implemented on top of Spark, a Hadoop-related distributed framework with fast in-

memory processing capabilities and support for iterative computations. The framework provided big data Mining toolkit that exploits the power of distributed systems whilst retaining the standard Weka interface.

Recently several resech works focused on applying BDA in health care sector. Xu and Kumar [13] proposed machine learning based BDA framework that improve the quality and performance of Auxiliary Power Units health monitoring services. The framework provided predictive analytics capability for system monitoring, diagnostics, product reliability and system performance. Their work combined cloud computing, BDA and machine learning technologies. Abusharekh et al. [14] proposed another platform called H-DRIVE for big health data analytics. It is an integrated, end-to-end health data analytics service-oriented workbench designed to empower data analysts and researchers to design analytical experiments and then perform complex analytics on their health data.

Nural et al. [15] proposed to use Semantic Web technology to assist data analysts/data scientists in selecting, building and explaining models in predictive BDA. They tried to address issues related to choice on modeling technique, estimation procedure/algorithm and efficient execution. The framework is supported for semi-automated model selection using analytics Ontology. However, this solution is differ from our BDA solution. We proposed to automate the BDA process using ASC.

Most of solutions are domain specific and some of them are providing partial real time support to the analytical process. When it is comparing with our solution, we have introduced a domain independent scalable solution to the BDA process, which is extended solution of our previous works [11], [14].

2.4 Planning Stage of the ASC

Planning stage is the first stage of the ASC process. Here we have to directly dealing with the exact requirement to find the most cognitive workflow to satisfy the requirement. Literature related to workflow generation for BDA processes is scarce. Several approaches directly link to BDA workflow generation related to the service domain. Kumara et al. proposed an ontology-based workflow generation for BDA

based on the ASC process [15], which generates an abstract workflow for the given requirement using ontology rules. Gil delivered a course for studying BDA using workflow paradigms [16]. A key aspect of their work is the use of semantic workflows to capture and reuse end-to-end analytic methods used by experts to analyze big data. Gil et al. [17] proposed time-bound analytic tasks in big data sets through dynamic configuration, which allows an extension to workflow systems that enables them to consider user deadlines and rank them according to performance estimates. Gil et al. also proposed a semantic framework for automatic generation of computation workflows using distributed data and component catalogs [18]. Kranjc et al. proposed an online workflow generation for BDA [19] using a platform called ClowdFlows, Their method focuses on BDA processing in batch and real-time processing modes. Hauder et al. proposed a framework for efficient text analytics through automatic configuration and customization of scientific workflows [20]. They have developed a framework to assist scientists with data analysis tasks, particularly machine learning and data mining. This framework takes advantage of the unique capabilities of the Wings workflow system to reason about semantic constraints. Wang et al. proposed a GraphPlan-based uncertainty-aware ASC planning method [21]. Pistore et al. proposed knowledge-level planning for service composition based on GraphPlan, which allows us to avoid the explosion of the search space because of the usually large and possibly infinite ranges of data values that are exchanged among services, and thus to scale up the applicability of state-of-the-art techniques for the automated composition of web services [22].

2.5 Discovery Stage of the ASC

Discovery stage is the second stage of the ASC process. Here it should discover the most suitable lists of services to satisfy the tasks across the work flow, which is resulted by the planning stage. The literature on Web service discovery for the BDA domain is scarce. Alarcon et al. [23] presented the REST Web service description for graph-based service discovery in the Big Data domain. They proposed the graph-based

method to overcome the limitations of REST services in certain ways. An approach by Akila et al. [24] was based on ontology for the BDA domain. An agent-based approach was proposed by Rajendran et al. [25]. They proposed a discovery framework consisting of separate agents for ranking services based on QoS certificates obtained from publishers. Johnson et al. [26] proposed a service discovery method designed to be used in heterogeneous networks. This method was specifically proposed for use in military domains because networks in that domain are heterogeneous. Rong and Liu [27] proposed a context-aware approach to overcome information loss during the transformation from the user's requests to a formalized one. Hai-Cheng and Hong [28] proposed a layer-based semantic method to improve efficiency of matching within a repository.

Interesting hybrid approaches have been proposed by several researchers. Tsai et al. [29] proposed a keyword- and ontology-based method. A multiple-criteria decision-making method was proposed by Saaty [30] based on text and ontology. It considers all associated attributes between query and services. It seems noisy because it considers functional and nonfunctional aspects. Wen et al. [31] presented a Web service discovery method based on semantics and clustering. An ontology-based workflow composition and discovery method was proposed by Karakoc et al. [32]. Their proposed model allows users to select relevant service types.

2.6 Selection Stage of the ASC

We proposed two types of selection method under the selection stage. Therefore first we discussed the literature related to the QoS and dynamic transaction awareness. Second we discussed the literature related to the QoS and traffic-aware selection method.

At first, when its considering the transaction perspective, the long running processes hamper the automation process and loss the result-data, time and resources. In the BDA perspective, it will be a huge loss. Transactional properties of the services can be utilized to address above concerns in composition process [33]–[37]. Most of the

conventional approaches are not considering the risk of unexpected termination or errors during the composition process. Then transactional service (TS) coming into the topic. Here also, some approaches are only limited to transactional awareness other than QoS-awareness [38], [39]. However, Only TS aware approaches are not guaranteed the requirements of the multivariate Quantitative QoS awareness. Especially it is a serious drawback of the BDA composition.

Few approaches are proposing composition models considering both TS and quantitative QoS awareness of the composition. Only Haddad et. al. proposed ASC based TS and quantitative QoS aware composition model for the workflow automation under two key risk levels [33]. Moreover, they have proposed semantic TS properties (TSP) identification methodology. It allows defining transactional requirements for the workflow in general not domain specific. It proposed to find the local optimal not global optimal and not flexible for custom user requirements. Z. Ding et. al. proposed genetic algorithm (GA) based approach and it focused global optimal solution [34]. They employed GA with a penalty function for given workflow. Nevertheless, as per our study, it has not considered the workflow automation and proposed algorithm does not provide the flexibility to customization. J. Li et. al. proposed composition model for the Directed Acyclic Graph models but not guaranteed either automation and custom settings [35]. J. Cao et. al. proposed Ant Colony-based TS and Quantitative QoS aware selection for near optimal solution [36]. Y. Cadinale et. al. proposed TS and Quantitative QoS aware selection based on Petri net unfolding algorithm [37]. However, none of these approaches considered the user custom setting of TS for the given workflow or domain specific solution during the composition.

As the second concern, QoS and traffic-awareness in the Big Data perspective, Service selection based on Big Data space is scarce. Wang et al. [40] proposed the BigData Space service selection, a method based on heterogeneous QoS-aware and trust values. It focused on accumulating both qualitative and quantitative values into a single trade-off model. It can be considered a comprehensive qualitative and quantitative approach. However, it does not consider heterogeneous-selection requirements, which is the key difference in our approach. It considers a multi-CPU

IBM high-end server for a Big Data space, but not traffic issues. In contrast, we have used one of the most populated Big Data space, the Hadoop multicluster approach, as our Big Data environment. In addition, their method does not have the ability to deal with dynamic selection requirements. In our case, we have analyzed our solution for three distinct selection requirements. Moreover, they proposed qualitative and quantitative QoS-aware static service selection, whereas we propose a quantitative QoS-aware dynamic service selection.

A traffic-efficient Big Data processing model has been proposed by Xia et al. [41]. They proposed an architecture that could deal with both the front-end and back-end traffic in a Big Data space. Lin [42] discussed the issues that occur because of the Zipf phenomena in a case study. It focused on the limitations to the parallelization of the MR job. The Pareto phenomenon is the naturally observable concern in most cases.

Kim et al. [43] proposed a solution that can select services based on QoS and trust. A two-layered preference service-selection solution was proposed. It had to pass two distinct phases because of the two-layered approach. The overall process was layered, and this is also a qualitative and quantitative QoS-aware service-selection solution, but it is not a framework. A further advanced approach has been proposed by Wahab et al. [44], which focuses on providing a precise reputation-assessment mechanism in an open environment. This reputation-evaluation process is interesting, and it offers a means of manipulating the reputation for overall service selection. Wan et al. [45] proposed a cloud-based service-selection method. It also discussed trending concerns in the service domain and proposed an architecture for discussing how to deal with cloud services. Huang [46] proposed a QoS estimation method through online communication. This might be very useful during the selection process because of the QoS preferential estimation that reduces the overall selection time. It is based on both qualitative and quantitative concerns. However, none of these methods are designed to cope with Big Data space.

Kang et al. [47] proposed a method that considered globally optimal service selection for multiple competitive peer users. They also discussed an interesting topic, namely resolving conflicting requests and allowing them to find their optimal selection within

the range of service distributions. They proposed an agent as a solution, which also features in our proposal. It is the key element in avoiding conflicts in the selection scenario. Hadad et al. [48] proposed a QoS broker architecture to find the optimum WS, which can be a provided service, based on user queries. They proposed a selection solution that can offer automatic service composition (ASC), which is also one of the trending requirements in the industry. It focused on the transactional constraints of the ASC process and aimed to satisfy the selection requirements while considering the functional requirements, transactional properties, and QoS characteristics.

Gao et al. [49] proposed a QoS-aware service selection based on a genetic algorithm that was mainly oriented toward trust in the QoS. They designed a trust-oriented genetic algorithm called TOGA, and their aim was to find a near-optimal plan for the composition system. Zhang et al. [50] proposed an ant-colony-based service-selection algorithm for large-scale QoS preference selection. They proposed a clustering-guided method that uses a skyline-guided process to filter the candidate service classes and cluster them by shrinking. These methods are based on intelligent agent-based service selection. In our approach, we proposed a middle agent (not an AI agent) to address traffic congestion during the selection process. Moreover, we proposed ABC for multivariate QoS optimization in the service selection.

An interactive analytical process was proposed for Hadoop space by Chen et al. [51]. They focused on addressing the traffic congestion caused by Zipf and Pareto phenomena. They proposed a solution based on a cross-industry study of the efficient management of MR workloads. Ke et al. [52] proposed a traffic-aware partition-based Big Data method that focused on reducing internal traffic congestion during an MR job. They proposed an intermediate data-partition solution to address these concerns. Their solution used a decomposition-based distributed algorithm to deal with large-scale optimization. However, none of these solutions were related to service selection but were discussed as the internal and external traffic solutions to MR jobs. We are proposing a solution that can cope with both internal and external traffic congestion in an efficient manner. In addition, we have applied this solution to practical application areas such as the service selection domain.

Traffic occurrence and optimization beyond the data center has been studied by D. Ersoz et al. [53]. They considered cluster-based network traffic characterization for multi-tier data centers. Their focus was the characteristics of network behavior within a clustered, multi-tiered data center with respect to the inter arrival-time distribution of requests to individual server nodes and tiers. This approach gave insights about low-level traffic handling and communication between tiers.

Traffic and communication optimization in Big Data infrastructures has been studied by various scholars. J. Zhang et al. [54] proposed a method for optimizing data shuffling in parallel computation by user-defined functions in an MR process. In addition, they proposed a framework called SUDO, which is an optimization framework that reasons about data-partition properties, functional properties, and data shuffling. This differed from our work, where we proposed shuffling and data utilization in the parallel processing of the MR process. M. Aledhari et al. [55] proposed a deep-learning-based data-minimization algorithm for the fast and secure transfer of big genomic data. They focused on maximizing the channel utilization by decreasing the bits needed for transmission of the dataset. They proposed a novel deep-learning CNN-based algorithm that minimizes the dataset during transfer and protects the data from middle-man attacks and other types of attack, such as changing the binary representation of the dataset. Y. Zhao et al. [56] proposed a distributed graph-parallel computing system with lightweight communications. Their system, called Ligraph, processes large-scale graph data in a distributed mode with optimal communication overhead. Z. Yan et al. [57] proposed heterogeneous data-storage management with deduplication for cloud computing. They focused on encrypted data storage, management, and the deduplication process across the cloud environment. Comparing our method with this method, we selected a Big Data environment and the MR process because of their lightweight data and extensive resource-starvation processing with minimal overhead traffic for solving heterogeneous-selection optimization problems.

For threshold-based policies, service selection and scheduling are both NP-hard problems. Scheduling problems are applicable to many domains including services, communication, and planning. X. Chen et al. [58] proposed buffer-aware scheduling

with the adaptive transmission, which focused on obtaining the optimal trade-off between the average delay and power consumption. We focused on the QoS when aiming for optimal traffic efficiency. X. Chen et al. modeled the problem based on a Markov decision process and proposed an algorithm to obtain the optimal solution. A. Asadi and V. Mancuso [59] surveyed scheduling in wireless communication, describing opportunistic scheduling from various perspectives such as capacity, QoS, fairness, and distributed scheduling. Our proposal also considered if the processing-resource capacity for Big Data, with linear, combinatorial and multi-objective QoS, and with distributed computing would be satisfied with respect to resource starvation.

2.7 Verification and Refinement (VR) Stage of the ASC

Here, we discuss the literature related to BDA workflow generation and POP problems. In both areas, relevant literature is scarce. First, we consider the literature related to BDA workflow generation. Some approaches discuss dynamic workflow automation for BDA, but tend to be limited to one or a few stages of the automation. Sparks et al. proposed the KeystoneML pipeline, which enables preparation of a dynamic workflow for the analytics, particularly dynamic optimization of the modelling stage of the BDA process [13]. Forward-chaining planning with a flexible least-commitment strategy proposed by Oscar et. al [60]. It is a heuristic planner, which can be easily adopted to temporal or multi-agent planning. Wang. et al. proposed a method called as pipsCloud, it is a workflow management system for the analysis of dynamic remote-sensing data proposed [61].

Several approaches directly link BDA workflow generation to the service domain. Kumara et al. proposed ontology-based BDA workflow generation based on the ASC process [15], which generates an abstract workflow for the given requirement using ontology rules. Pistore et al. proposed an ASC framework for large-data analytics, which enables planning the composition of services by using their knowledge-level models [22]. These approaches have not discussed dynamic features of the BDA

workflow.

POP problems originate from two main issues: i.e., verification and refinement. V&R is related to the POP problem domain in AI planning. The planner does not need to search the whole space, but does need to verify that the planner can achieve the given goals. If a threat is found, a refinement process must be invoked. AI POP planning techniques for service domains are limited and those dedicated to BDA automation are scarce. Peer proposed a POP-based planner for the composition of Web services [62]. He proposed an algorithm for replanning a service composition. However, it does not consider composite Web services, which are often used to accomplish complex tasks in modern day BDA services. Szreter proposed a graph-based POP planning technique for service composition [63]. In their method, the planning problem becomes a graph problem, finding the subontologies and aggregating them to prepare the workflow for the requirements. However, it is not guaranteed to achieve the dynamic composition of composite services. Yan et al. proposed POP for the service domain based on a genetic algorithm. Their approach considered the POP planning problem as an incompletely observed problem space [64]. Wassink et al. proposed a method for a partial-ordered workflow-refining method for a service domain [65]. They extended the eBioFlow workflow system to include an ad-hoc editor. There have been several POP heuristic approaches proposed for the AI planning domain. UcPOP is possibly the best known. It works best for open goals rather than threats, behaving as a partially observable refiner in the service domain. The RePOP planner proposed by Nguyen and Kambhampati considers the POP as a planning graph in estimating the cost of achieving subgoals. They discuss the pessimism about the scalability of POP methods presenting several novel heuristic control techniques [66]. VhPOP is a versatile heuristic POP proposed by Younes and Simmons [67]. Authors discussed the solving problem by causal link planner and partly based on the UCPOP. UCPOP supports planning with durative actions by dealing with the techniques for temporal constraint reasoning.

2.8 Execution Stage of the ASC

The execution stage implies that executing the selected services (by selection stage) those are refined through the VR stage. Generally, in the service computing domain, execution is referred as service selection, and service composition. However according to the proposed architecture, we divided the generally known service composition process in to the five stages. Then we address the concerns occurred in those stages considering the domains (BDA and service) specific concerns. Then according to we proposed execution stage dedicates only to execute the verified and refined selected services.

Chapter 3 Architecture for Intelligent BDA

In this chapter, we present architectural design process for the BDA automation. The **Objectives** is the propose architectural design process for the BDA based on the ASC. **Key contributions** are, first we present the way we made the key decisions and architectural design process. Next, we discuss architectural style and flow of architecture design and etc. After that, Chapter 3.2 presents the reference architecture (RA) and the way we conclude the RA for the BDA process. Moreover, in chapter 3.3 presents the motivation scenario, which is made for the BDA domain and next the way derived the System Architecture (SA) based on RA and the given scenario. Finally chapter 3.4 presents high level UML class diagram and the way we decide the important decision on that. As the **future works**, we are looking forward to extend the architectural design process for the smart framework for the Data Science.

3.1 Architectural Design Process

The proposed solution is the extended solution of our precious works. Fundamentals in proposing the architectures are same the previous works. Moreover, we proposed further advanced solution, which addresses lacks behind the previous work. Architecture is the first and most readable interpret real world problem in to the technical language. We can choose the pathway of software engineering design and development due to the complexity of the requirement and the problem domain. If it

has very simple requirement then there is no use of design level approach. It can start the development in the beginning. If it has detailed requirement which involves multiple roles, then we can initiate it from the Entity Relationship diagram or Unified Modelling Language (UML) class diagram. Since the requirement complex further and then it needs further dive in to the design process. In middle level complexity cases we can do the System Architectural (SA) level approach. And if the requirement and problem domain raises serious concerns and increase the complexity, we have to go deeper in to the design level that is beyond the SA.

According to the standard software design and analysis techniques, we have to consider from the architectural styles. There are several types of architectural styles. It can be used based on complexity of the requirement and the problem domain. Below Table 3.1 shows the summary of requirement category vs architectural styles [27].

Table 3.1. Requirement categories vs architectural styles

#	**Category**	**Architectural Style**
1	*Structure*	Layered Architecture , Object Oriented, Component based
2	Communication	Service-Oriented Architecture (SOA), Message Bus
3	Deployment	Client/Server, N-Tier, 3-Tier
4	Domain	Domain Driven Design

In BDA perspective, our problem domain (BDA) is categorized in to the structure category. Because we have to deal with various technologies and tools, which are involving in BDA process and those can be categorized in to the three main groups. Table 3.2 shows the technologies and tools consist in each groups. These groups are infrastructure, technology and analytical.

Table 3.2. Identified layers vs technologies and tools

#	**Layer**	**Technologies and Tools**
1	*Infrastructure layer*	Hadoop 2.x, Service Registry, Data Warehouse (RDBMS Cluster), Analytics Database (Analytic Cluster)
2	*Service Layer*	Automatic Service Composition (ASC), Planning Agent, QoS Agent
3	*Analytical Layer*	Cross Industry Standard Process for Data Mining (CRISP-DM)

Then we have to select an architectural style which can be dealt with multi groups in efficient way. Therefore we have to do further study on each architectural types. Here below Table 3.3 shows the architectural styles vs usage description in respective problem domains.

As it mentioned earlier, Layered Architectural is the most cognitive and suitable the approach to beginning our architectural design process. This Architecture suits very well to our requirement. Therefore our architectural design process starts from the structured category and Layered style.

Table 3.3. Architectural styles vs usage descriptions

#	Architecture style	Usage description in given problem domains
1	**Layered Architecture**	Partitions the concerns of the application into stacked groups (layers).
2	**Service-Oriented Architecture (SOA)**	Refers to applications that expose and consume functionality as a service using contracts and messages.
3	**Client/Server Architecture**	Segregates the system into two applications, where the client makes requests to the server. In many cases, the server is a database with application logic represented as stored procedures.
4	**Component-Based Architecture**	Decomposes application design into reusable functional or logical components that expose well-defined communication interfaces.
5	**Message Bus**	An architecture style that prescribes use of a software system that can receive and send messages using one or more communication channels, so that applications can interact without needing to know specific details about each other.
6	**N-Tier**	Segregates functionality into separate segments in much the same way as the layered style, but with each segment being a tier located on a physically separate computer.
7	**Object-Oriented**	A design paradigm based on division of responsibilities for an application or system into individual reusable and self-sufficient objects, each containing the data and the behavior relevant to the object.

Next, we create a scenario based on BDA domain. Then we can create a System Architecture (SA) based on RA and the scenario. After that we can move to the UML class diagram level. Finally it provides sophisticated solid base to start the development from the scratch in free of mind.

3.2 Reference Architecture

In chapter 3.1 we concluded that the layered architecture is our initial step of design process. Then we move to the *Structured* category and *Layered* style architecture type. At the beginning, we cannot be stand with solid architecture due to the Research and Development (R&D) type problem domain. Then we have to start from the abstract

framework and that must be supported to maintained (improve) architecture based on the research. That means, in the beginning we have to concern two issues, which are layered architecture styles and abstract level initiation.

Then Reference Architecture (RA) comes as the obvious solution, which facilitate to make template solution for complex problem domain and facilitate to layered styles. The Rational Unified Process® (RUP®) states that such harvesting of best practices within the organization is the first step toward building a strong, versatile reference architecture. Briefly, a reference architecture consists of information accessible to all project team members that provides a consistent set of architectural best practices. These can be embodied in many forms: prior project artifacts, company standards, design patterns, commercial frameworks, and so forth. The mission of the reference architecture is to provide an asset base that projects can draw from at the beginning of the project lifecycle and add to at the end of the project [28].

We have come a cross *Category, Style* and *Type* of the architecture domain. By the way, it was result *Structured* and *layered* support 3 layered RA model. Figure 3.1 is shown the model we have designed for the Intelligent Big Data Analytics Automation process. This provided solid base to extract SA. The SA is a conceptual model that defines structure, behavior and more views of a system. Simply RA is layered solution, which gives high-level view how each components and technologies of the product behave and how it maintains interactions between each of them. This layered pattern connected closely to an architectural principle "loose coupling" [29].

In addition, according to the BD architectural planning and designing perspective we have studied five important observations while studying literature [5].

(1) It is clearly define core of a BD architecture.

(2) There is more than MapReduce: data sources, data mining processes, coordination and configuration engines, databases, monitoring etc. Further business intelligent systems and software components still will have a place in a BD architecture.

(3) Several architectural principles have been applied. Loos coupling and scalabilities are popular principles among them.

(4) "Data pipeline approach" is the cognitive and truly stands out. This indicates that BD architecture is like a pipeline through that data flows.

(5) There seem to be more consensuses about the principles and best practices.

3.2.1 Conclude the Reference Architecture

Angelov's framework is one of well-recognized frameworks to use in design and development of RA. We have been assisted that framework during the design and development of the RA for Intelligent BDA process [30]. According to the Angelov's framework, it has to do following analysis before creation of the RA. Main three dimensions have to identify and clearly defined. Goal, Context, and Design are needed to clearly define. Next, it has to be determined and studied following sub dimensioned under above mentioned. Under the goal, it should be studied "Why is it defined?" dimension. Under the context, it has to be studied "Where will it be used?", "Who defines it?" and "When is it defined?" sub-dimensions. And under the design, it should be studied "What is described?", "Detailed: how is it described?", "Concreteness: How is it described?" and "Representation: How is it represented?". Here below given detail information we have studied based on Angelov's framework of creating the RA.

Discussion about Main Three Dimensions

In this section, we discuss the main three dimensions based on Angelov's method. It is the way of conclude our RA for the BDA domain based on ASC.

Goal: ***Why is it defined?***

Here it needs to clarify goal of the reference architecture. There are two possibilities of the Angelov et. al. defined, those standardization and facilitation. Our main ambition is to standardize concrete architecture. Therefore, goal of the RA for BDA solution is *standardization*.

Context: ***Where will it be used?***

Here it needs to clarify the application context of the RA. This is for the organizations who are working for predictive analytics based analytical requirement data gathered from various data sources. Service computing will use as main technology. In addition, it should be able to generate concrete system architecture based on the RA.

Therefore, context of this approach is to *multiple organizations*.

Design: ***What is it described?***

Here it distinguishes, elements that can be defined in the RA. We need to distinguish components, connecters, interfaces, protocols, algorithms, policies and guidelines. We are using CRISP-DM as the data mining concepts behind the solution. Since we are based on the ASC as selected architectural principle of overall solution.

Above we have been discussed main three dimensions under the Goal, Context and Design. Next we discuss the sub-dimension which are under main two dimensions as follows.

Context: ***Who defines it?***

Here it distinguishes, who will define it? . It is about the stakeholders of the RA? Mainly two groups involved in. They are designers and providers. Mainly product will be use by who is doing predictive analytics in standard organizations.

When is it defined?

Here it needs to clarify the timing aspects of the RA. That is our solution needs to be time independent. That means RA will be outdated when components used in the RA are outdated. It has two possible artifacts, which are preliminary and classical. It is using Preliminary RA if there are no concrete components have used in the RA. We have to define and make clear idea about the components are used in, according to the system development and concrete system architecture development. It will result more practical and sustainable RA solution. Therefore, our approach is *classical*.

Design: ***Detailed, how is it described?***

Here it distinguishes, number of levels can be defined in "What is it described?". It avoids more than two instances of usage of elements and components behind the solution. Further it will go through the *levels and stages* of above mentioned technologies in overall solution.

Concreteness, How is it described?

Here it distinguishes, possible level of abstraction of RA. It is related to the level

of choices made in an architecture in terms of technology, application and users etc. Here it has three values, which are abstract, semi-concrete and concrete. We plan to achieve a *concrete architecture*.

Table 3.4. Dimensions and values of the reference architecture

Dimension	**Values**
Why is it defined?	*standardization*
↓	
Where will it be used	multiple organizations
Who defines it?	*standard organizations*
When is it defined?	*classical*
↓	
What is it described?	*ASC, CRISP-DM*
Detailed	*Levels and stages of above technologies*
Concreteness	concrete architecture
Representation	*semi-formal and formal element specifications*

Representation, How is it represented?

Here it distinguishes, possible levels of formalization of RA. We have planned to use ArchiMate 3.1.1 for development and design the RA. Three architectural layers planned to be used. ASC, that is one of main concepts behind the Service oriented architecture (SOA). That is we have *semi-formal and formal element specifications* about the using elements.

According to our initial analysis of RA on BDA domain is based on the Angelov's framework, Table 3.4 is summarized information as follows. Table 3.4 describes the dimension vs values we have identified during process. Each dimension represent identical value and this flow summarizes to way of conclude concreteness and representation of the RA. It is discussed the main technologies we used and organizational support etc.

Based on the Angelov's framework of RA, we have been concluded the Type 2 RA. That is the Standardization RA.

Development of the RA: For the RA perspective, we have identified main 3 building block layers, top level layer called as Analytical Layer, middle layer called as Technology Layer and bottom layer called as Infrastructure Layer. Let start to identify each layers in summarily:

Infrastructure layer: It mainly considers data warehouse and data mart layer. This contains Hadoop Eco system to manipulate BD infrastructure, web service pools and two relational data base managements (RDBMS) for data manipulation and to maintain analytical clusters. All of the above can exist in Intranet and Internet platforms. As an example, Hadoop cluster can be Geo distributed as data centers and then we have to

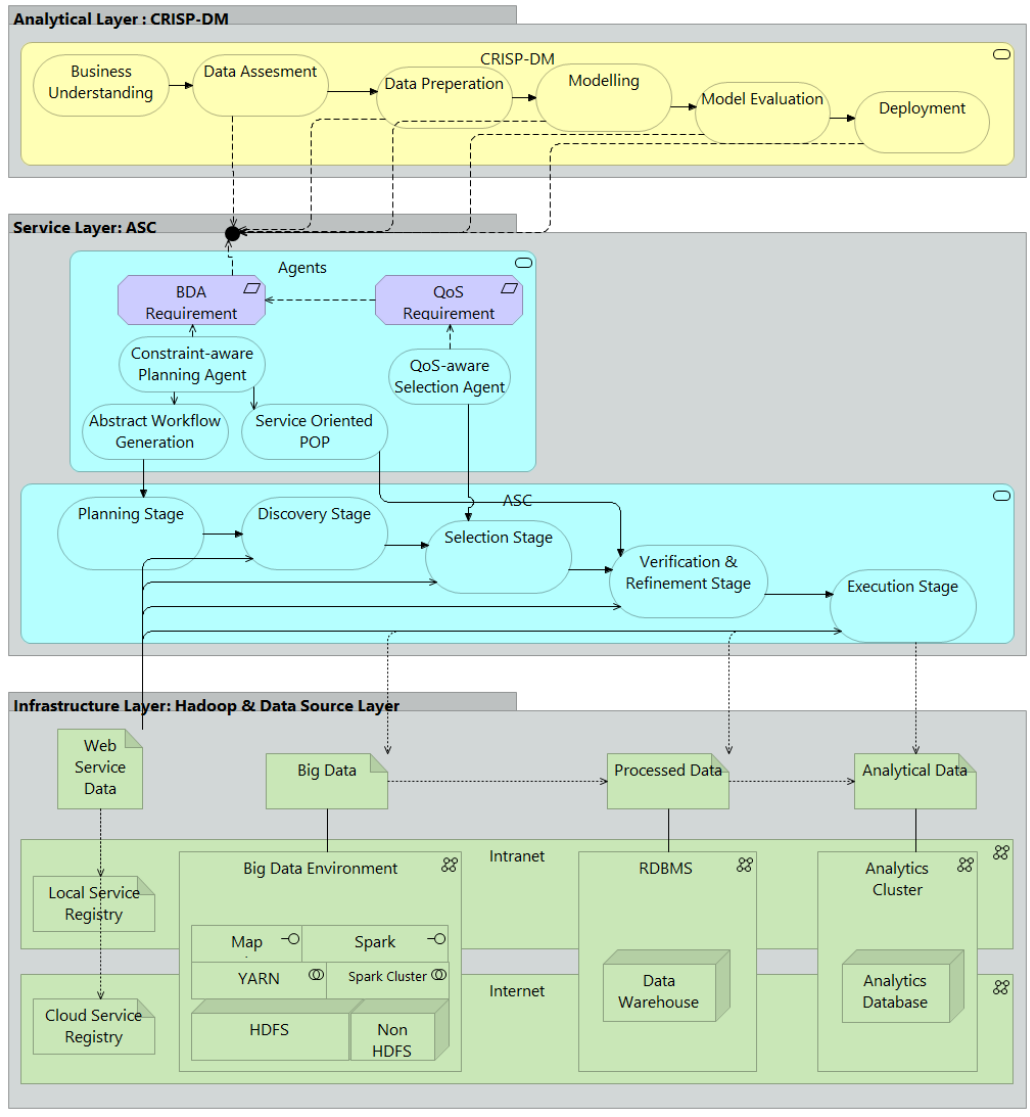


Figure 3.1. Reference architecture of the intelligent BDA

deal with hadoop beyond the intranet level.

Since web services also can be distributed along the internet and local network. One of two RDBMS ready to accept export data from Hadoop and data processing facility to the technology layer. The other RDMS is used for handling analytical cluster and related activities of the analytical process.

Service layer: Mainly this is dominated by ASC and it supports technologies such as quality of services agent and intelligent planning agent to provide intelligent workflow automation facility. An agent which comprises two sub agents, which are responsible for acquiring BDA requirements are the constraints, the other one is responsible for acquiring QoS related information of the services.

Therefore it will identify the requirement to utilize respective resources distributed along the system to fulfil the functional and non-functional requirements of the project.

Analytical layer: This layer is dominated by CRISP-DM to provide data mining process of the project. First two out of 6 stages of the CRISP-DM has been decided by manually and therefore ASC will be dealt with the rest of four stages to full-fill the data automation of mining requirement

3.3 System Architecture

In chapter 3.2 we concluded the RA for the BDA domain. Here we express our motivation for the SA and explain the SA.

3.3.1 Motivating Scenario

The motivation scenarios are discussed in Chapter 2.2.1.

3.3.2 Proposed System Architecture

ASC is the key technology behind the automation process. Architectural perspective we should be clarified the reason behind ASC as the key technology of the BDA process automation.

Our requirement behind the BDA is process automation. According to the Table 3.1, SOA is categorized in to the communication category. Then it (automation process)

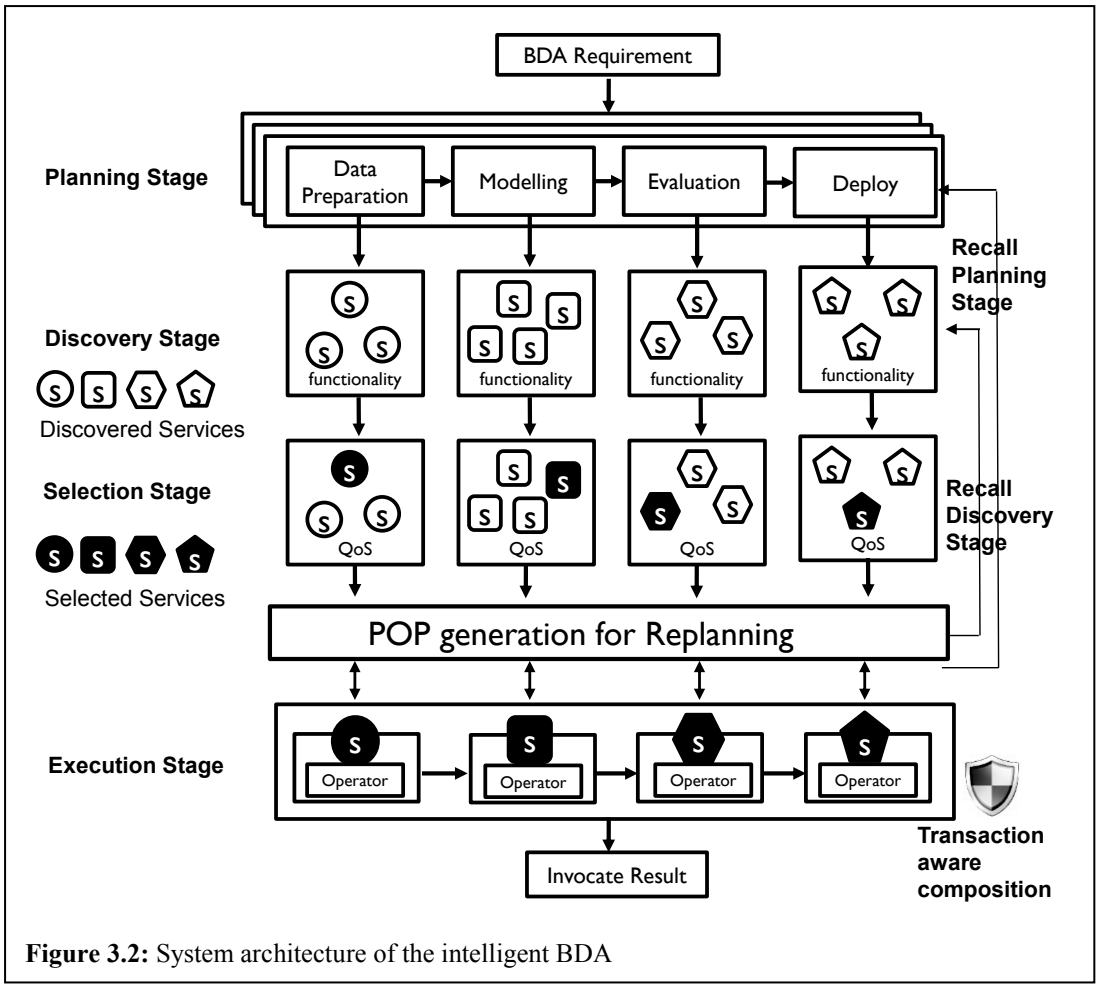


Figure 3.2: System architecture of the intelligent BDA

needs to communicate with various technologies and tools across the platform to full-fill automation requirement. And the ASC is the well-known dynamic process automation technology [2]. And also, ASC technology is derived based on SOA style. That means, we have selected one of the most suited technology for the process automation.

Next we move to the proposed architecture. Figure 3.2, proposed a SA for given scenario of the BDA process automation. We derived the SA based on the RA and applied our scenario to the SA. It clearly shows how each layer will behave during the execution time and result (output) will be produced in it is (ASC) execution stage.

ASC Process: This is the key technology of the solution. This allows identifying the functional and non-functional requirements for the analytics as follows.

R: Set of user's requests at the service level.

$G_{BDA} = \{G_{DP}, G_{Mod}, G_{Eval}, G_{Dep}\}$. : Set of sub-goals in the G_{BDA} objective of the BDA.

Planning: Π: G_{BDA} = {G_{DP}, G_{Mod}, G_{Eval}, G_{Dep}}. Here. G_{DP}, G_{Mod}, G_{Eval}, and G_{Dep} represent the subgoals for data preparation, modeling, evaluation and deployment stages, respectively. Then, according to the scenario explained in Section 2.2 they can be expressed as four main CRISP-DM subgoals.

Many researchers adopt the hierarchical task network (HTN) planner (Sirin, Parsia. 2004) technique to dynamically develop workflows. However, it is acknowledged that the formulation of HTN planning problem requires significant structural information. In contrast to their work, in this paper, we use graphplan-based workflow generation method. Our core idea is to utilize ontology to acquire hidden domain knowledge, in order to generate more application-specific abstract workflow. Based on ontology designed for CRISP-DM, we have developed approach for the workflow generation. A rule-based approach is developed for detailed inference. We have implemented SWRL rules to identify the abstract services according to the properties of dataset and user requirements.

Discovery: $\Delta: W \rightarrow I \ \therefore \ C_j = \{c_{j1}, c_{j2}, \ldots, c_{jp}\}$ Set of I selected service instances to be executed from the service instance set . C is the set of C_j where $1 \leq j \leq l$. In this work, we employed two types of selection methods which are described in Chapter 5.3 and 5.4.

Selection: WF_1 = {CWS_{11}, CWS_{12}, ⋯, CWS_{1n}}, here n is the total number of CWS's in the workflow. Available WS's for each of the CWS, CWS_{12} = {(WS_{11}, WS_{12}, ⋯ , $WS_{1.y1}$), (WS_{21}, WS_{22}, ⋯ , $WS_{2.y2}$), ⋯ , (WS_{n1}, WS_{n2}, ⋯ , $WS_{n.yn}$)}, where WS_{ij} represents the j^{th} candidate web service of CWS_i and y_n is the total number of candidate web services of CWS_i. In this work we proposed two types of selection methods, which are described in Chapter 6.2 and 6.3.

VR: $\pi = (\Upsilon, \prec, B, L)$ is defined as the partial order planning problem. Υ is the partially instantiated set of service tasks and ≺ is the set of ordering constraints on Υ of the form ($\Upsilon_i \prec \Upsilon_j$). is the set of binding constraints on the variables in the Υ_i tasks. For example, given Υ_i, the output ϑ_i and a Υ_{i+1} input of ι_l^{i+1}, then the variable binding

constraint should be $\vartheta_i = \iota_l^{i+1}$ or $\iota_l^{i+1} \in \text{Typecast}_{\vartheta_i}$. Finally, L is the causal link which supports the interlinking of two tasks or the introduction of a new input file to the workflow. In this work, we proposed CWS based constraint aware TOP generation method, which is described in Chapter 7.

Execution: : $C \rightarrow X$ **:** All the selected services of given concrete tasks will be executed according to the TOP resulted by the VR stage.

Chapter 4 Planning Stage of the ASC

In this chapter first we discussed the research philosophy during the discovery stage of the BDA process. In Chapter 4.1 and 4.2 discuss the motivation scenario and the introduction. Next, Chapter 4.3 discuss the proposed method for generating abstract workflow the BDA. The **Objective** is generating an abstract workflow for the BDA considering constraint awareness behind the process. **Key contributions** are proposed modelling for constraints aware planner, generate constraints aware workflow for the analytics and task simulation technique based on action and proposition used in the graph planner. As the **future works**, planner needs to be improved to deal with complex requirements while increasing the mutual exclusive constraints in the planning requirement.

Big data analytics (BDA) is the preferred approach to unleash the hidden knowledge from high volumes of varied data generated at high velocity with high variability and veracity. However, BDA consumes excessive resources and time. These limitations are hampering the meaningful adoption of BDA. Therefore, automating the BDA is a cognitive approach to the BDA domain, and service composition became the preferred platform for BDA. In previous work, we have proposed BDA based on automatic service composition (ASC). According to the ASC process; at the beginning, abstract workflow and before the execution of composition, concrete workflow for the analysis must be generated. BDA is highly dependent on diversified constraints and must undergo rigorous data-mining stages, which are comprised many subtasks. This cause

to increase the solution space. Large solution space, constraints-awareness, resource and time consuming are affecting abstract workflow generation. Therefore, we propose GraphPlan-based abstract workflow generation. After selecting services for the abstract workflow, we are left with a constraint-aware partial-order planning (POP) problem. We then propose an algorithm that addresses the service-oriented POP problem. Combining these two-staged planning techniques generates a concrete workflow. Experiments demonstrate that the proposed two-staged process is well behaved when generating a concrete BDA workflow.

4.1 Motivation Scenario

A Japanese national institute dealing with advanced industry is initiating many projects in renewable-energy fields, one of which is conducting a weather-data analysis program in Fukushima Prefecture to investigate effective renewable-energy sources. Their aim is to promote adoption of renewable energy, to find the most efficient energy sources and to decommission or maximally reduce the dependency on existing nuclear power plants in northern Japan. In this project, the institute is collaborating with two universities: Fukushima University and the University of Aizu. One of the contributions, involving a smart grid and energy IoT areas, is described in [68].

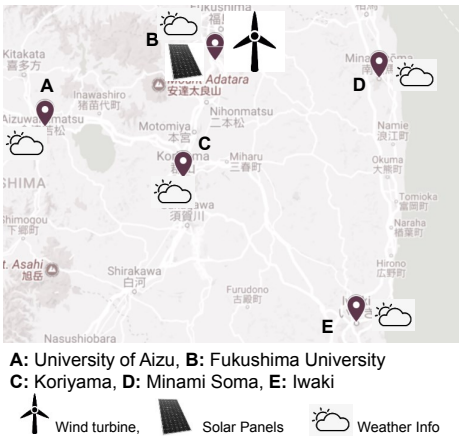


Figure 4.1.1: BDA scenario for sustainable power research

4.2 Introduction

Data science is facing severe concerns about handling big data efficiently. Big data is characterized by its multidimensionality; i.e., it is not only voluminous, but also highly variable and high velocity (3V data). Data involving additional dimensions of variability and veracity is called 5V data, which has potential for new directions in research and industry that cannot be realized by conventional business analytics techniques. The process of harnessing 5V data is called big data analytics (BDA). BDA is becoming the leading platform in the data analytics paradigm. The field has grown exponentially, with BDA insights enabling businesses to be fast, remain agile, and become the front-runners in their fields. Revenues from BDA are projected to exceed $200 billion in 2020, up from nearly $134 billion in 2016.3 However, the BDA process requires highly diverse and rigorous stages to be successful, which involves very large resources and makespan (the overall time for the process). This constrains the full meaningful effects of a state-of-the-art BDA product. Automating the BDA process offers the most cognitive solution to these concerns. Automation may enable data scientists to accomplish their task in days rather than the traditional period of months.4

BDA process automation has been discussed only rarely in the existing literature because of its complexity, which derives from its explicit and implicit constraints on its dynamic process and requirements with respect to time, resources, data, modelling, and deployment. However, some attempts were made to partially or fully automate aspects of the BDA process, such as data preparation, modelling, model optimization, or deployment [69], [13], [19], [20]. Existing approaches are mainly focused on automating one or more stages of the BDA process. We propose a BDA architecture based on automatic service composition (ASC) [11]. Our aim is to automate the data preparation, modelling, evaluation, and deployment of the cross-industry standard process for data mining (CRISP-DM) [10] based on ASC [12].

Workflow generation for BDA in ASC is one of the four main stages involved in the

[3] https://www.idc.com/getdoc.jsp?containerId=prUS41826116
[4] http://news.mit.edu/2016/automating-big-data-analysis-1021

ASC process. Existing works related to BDA automation have proposed the use of various techniques to prepare the workflow, which were either partial or involve manual intervention. According to both domain experience and literature reviews [69], [13], [19], [20], [15], [17], [16], we can observe that seamless workflow is the foundation of seamless automation of BDA. Furthermore, automated workflow is an intrinsic requirement of the ASC process. Therefore, we have divided the planning problem into two stages. The first is the planning stage, which prepares the abstract workflow for BDA by considering domain-specific concerns. The second stage occurs after the service selection stage but before the service composition (execution stage), and is the introductory replanning stage, which refines the workflow to prepare a concrete workflow for analysis.

The BDA workflow generation-related literature describes a variety of techniques. Zulkernine et al. initially used an abstract workflow prepared by a big-data consultant and mapped it via the ontology to generate a concrete workflow [69]. Kumara et al. proposed using ontology-rule-based workflows from a big-data consultant [15]. Sparks et al. proposed a dynamic workflow for analytics via a pipeline called "Keystoneml." However, it was limited to adjusting the modelling and optimization [13]. Likewise, most of the techniques are limited, with predefined or manual adjustments to the analysis workflow during the process. Yolanda et al. proposed time-bound dynamic workflow generation and a ranking method [17], which was restricted to focusing on time constraints. This is very limited, considering the complex constraints, dynamics of the BDA process, adoptability and automation in most of the literature. Replanning is essential during execution of the workflow to enable adaptation to constraints occurring on the fly. As an example, consider the classification of human and machine generated documents using deep learning in natural language processing. It would be an essential requirement to repeatedly fine-tune the hyper-parameters used to create a successful model until the preconditions defined for the model could be satisfied. Changes to the input files would be required to achieve a successful output. For these reasons, a replanning ability is required to adapt to a highly dynamic environment. Therefore, the highly dynamic process and the implicit and explicit constraints are the main reasons

for non-automated workflow generation. Some implicit constraints are mutually exclusive (mutex; tasks cannot occur at the same time, i.e., XOR logic) and others are mutually inclusive (mutin; events can occur simultaneously as well as independently, i.e., AND logic) [16], [70]. This causes the evolution in the dynamic workflow for BDA [71], [72]. Therefore, to automate the workflow generation for the BDA, the overall processes should be flexible enough to adjust explicitly to the dynamic constraints.

BDA workflow generation in the planning stage of the ASC should consider both explicit and implicit initial constraints on the data (e.g., types and limits of data sets), variables (e.g., optimal limits for model optimization), and resources (e.g., deployment platform). In addition, each stage of the CRISP-DM process may be further divided into many substages depending on the BDA requirements. This implies a large search space and the initial planning stage problem will be NP hard [73]. To overcome this concern, the workflow generation method should be a heuristic method. Furthermore, two of the main objectives behind the automation are to minimize the makespan and its overall resource consumption. Therefore, the solution workflow for BDA should be a shortened workflow, comprising a minimal number of horizontal tasks (achieved by maximizing the number of parallel actions).

To begin with, therefore, we need to prepare an abstract workflow for the BDA that considers explicit and implicit constraints, heuristic methods, and a shortened workflow for the analysis.

Abstract workflow generation based on AI planning techniques have been discussed for data analytics. A wide variety of methods are available to address different aspects of workflow requirements. Ontology-based workflows have been used to discuss semantic knowledge in the planning problem [15]. Finite-state machine-based workflows consider process in terms of a finite automaton. Petri-net-based workflows consider the transition of activities satisfied by pre- and post-conditions. However, none of these methods meet our fundamental requirements for the analysis, which involve complex constraints and shortened workflow.

Several types of AI planning techniques [74],[75],[76], such as STRIPS planning, consider state transitions that are not guaranteed to meet complex constraints and

shortened workflow. Among the task network planners, hierarchical task networks (HTN) support explicit constraints better than simple task networks, but none of them guarantee a shortened workflow. Moreover, they are not end-user friendly and need the domain author's involvement to include additional constraints such as implicit constraints to the planning problem. Forward-search planners, backward-search planners (BSP), and GraphPlan (GP) are considered good heuristic planners for complex-constraint satisfaction [76]. Among these three approaches, backward-searchable algorithms, i.e., BSP and GP, algorithmically work towards a shortened workflow with a minimum number of actions and maximum parallel actions [76]. Of these two approaches, GP has additional features that enable complex constraint handling more efficiently. These include special provisions for implicit-constraint handling of mutex combinations via a technique called mutex handling and for mutex tuples via the no-good table technique [76].

Issues in GP and the proposed method for abstract workflow generation involve an improved GP with unground/non-primitive tasks. Because GP prepares the planner as a collection of actions called ground/primitive tasks, such as a ground action for changing an effect of a given proposition to remove null values. But a BDA workflow contains a collection of unground tasks. An example is a defect-handling task, which contains a collection of ground actions related to defect handling, such as remove null values, missing values, truncation of data, and type mismatch. Unlike HTN, existing GP methods do not provide task networks. To address this issue, we propose an approach called the task simulation method, which prepares the unground tasks for the workflow. This offers two main benefits to BDA automation: i.e., providing real-world workflows for BDA and reducing the number of steps in the overall process.

4.3 Proposed method

4.3.1 System Modelling

Here, we discuss the proposed constraint-based search plan for the BDA process. In our motivation scenario, our BDA domain is a constraint-aware user case. We selected

GP, which is a leading constraint-based AI planning technique based mainly on propositions, actions, preconditions, post-conditions, and the negative constraints imposed on them. In addition, we propose a task simulation technique to create detailed tasks in the BDA workflow based on actions and propositions. The concepts underlying the proposed model are defined as follows.

Definition 1: The goal of the planning problem is denoted by G_{BDA}, comprising the four main subgoals shown in Fig. 4.1. Therefore, we substitute the main problem by four subplanning problems, denoted as:

$$G_{BDA} = \{G_{DP}, G_{Mod}, G_{Eval}, G_{Dep}\} \quad (1)$$

Here. G_{DP}, G_{Mod}, G_{Eval}, and G_{Dep} represent the subgoals for data preparation, modeling, evaluation and deployment stages, respectively. Then, according to the scenario explained in Section 2.2 they can be expressed as four main CRISP-DM subgoals.

Definition 2: The propositional planning problem for the BDA process is denoted by P and the goal state is denoted by S. It comprises three main components: ground state transition system Σ, initial state S1, and goal G_{BDA}:

$$P = \{S; \Sigma, S_1, G_{BDA}\} \quad (2)$$

Definition 3: The unground state transition system Σ comprises three tuples: i.e., set of states S, set of actions A, and propositions P of each state, as shown by (3).

$$\Sigma=(S,A,P) \quad (3)$$

According to our scenario, the unground states of the transition system represent the respective achievements of each of subgoals of the CRISP-DM. For example, the data proration ground state means that all the preconditions must be satisfied before proceeding to the modelling stage of the CRISP-DM.

Definition 4: The unground states vs. tasks relationship is defined as follows. If one or more actions create a given state Si, then Si is denoted by:

$$Si = \{\bigcup_1^m T_j : \text{m is number of } T_j\} \quad (4)$$

Here, the given state Si is a combination of a set of workflow tasks. For example, the data modeling state comes after the data preparation state. A combination of tasks must be performed, i.e., defect handling, clean data, and prepare HDFS file, to achieve the

completion of the data preparation unground state.

Definition 5: The ground states vs. ground action and proposition relationship are

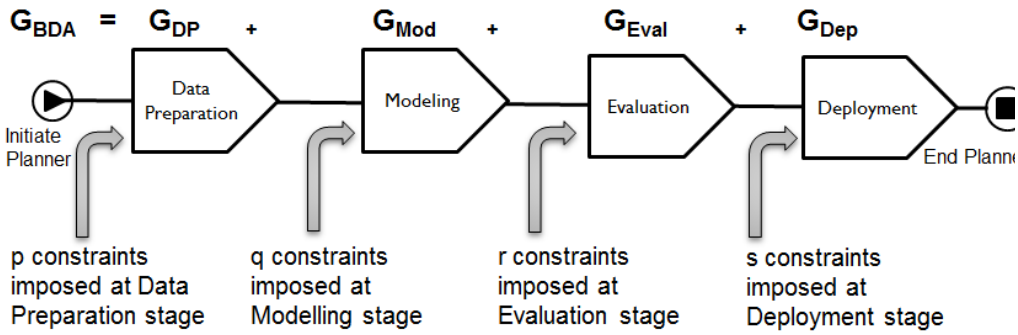


Figure 4.1: Sub-goals vs. set of constraints imposed on the planner

defined as follows. Ground state s_j is a combination of a given ground action A_j and proposition P_j:

$s_j = \{A_j \cup P_j$: sj is the jth element of S$\}$ (5)

Adjacent ground actions and propositions in GP create respective ground states in the BDA process. According to our scenario, three types of data in XML file format and all must be converted to CSV format. Ground actions in FileFormatConversion and the respective adjacent propositions caused by fileType=CSV create the respective ground state in the BDA process, i.e., FileFormatConversionXMLtoCSV.

Definition 6: The ground states vs. tasks relationship is defined as follows. One or more ground states s_j create a given unground state task Ti, which can be denoted by:

Ti = $\{\bigcup_1^n s_j$: n is the number of s_j elements$\}$ (6)

A given task Ti is a combination of a set of workflow unground states. According to our scenario, the cleaned data state comes from the uncleaned data state. A combination of ground-state processes must be performed: i.e., data passing to correct syntax errors, duplicate elimination to avoid duplicate content, and handle erroneous data to avoid contextual errors in the data.

Definition 7: The task simulation is denoted by F(T). It simulates tasks by conjugating actions and events defined in the state transition:

F(Ti) = $\{\sum_i^k (A_i \cup P_i)$: connection between adjacent action Ai and proposition Pi in the planner graph$\}$ (7)

Adjacent actions and propositions in GP create respective tasks in the BDA process. For example, an action FileFormatConversion and the respective adjacent proposition fileType=CSV create the respective detailed task in BDA.

Definition 8: The relationship between tasks and subgoals is defined using Definitions 1, 4, and 7. To achieve each subgoal, the respective states defined in the state transition system must be reached. Each ground state is a collection of decomposed tasks Ti. The subgoal GSUB may be denoted by:

$G_{SUB} = \{\bigcup_1^n S_i$: each S_i made by collection $T_i\}$ (8)

According to the given scenario, subgoals of the BDA process should be achieved by executing the respective collections of tasks for those subgoals.

Proposition 1: The existence of a solution to the planning problem P is valid if and only if a set of all possible goal states intersects with the set representing tasks that are reachable from the initial task T_1. That is, the resultant intersection must not be the empty set.

$G_{BDA} \cap \Gamma > (\{T_1\}) \neq \{\}$ (9)

Proof: To have a solution to our planning problem P, there must be a reachable solution. According to Definition 1, the subgoals are G_{DP}, G_{Mod}, G_{Eval}, and G_{Dep}. If any subgoals cannot be achieved, a solution is not available. According to Definition 2, the corollary is that there must be a reachable goal state, which is denoted by S. According to Definition 4, the state is a collection of tasks denoted by {Ti}. This means that each proposed subgoals G_{DP}, G_{Mod}, G_{Eval}, and G_{Dep} must be true in some reachable state for at least one planning solution, which means a task collection. This implies that the set of all subgoals GBDA intersects with a set of tasks initiated from the first task T_1. In addition, the intersection of sets must not be the empty set. At least one planning solution appears if and only if GP produces a solution.

□

4.3.2 Proposed Algorithm

The proposed method is based on the GP technique described by Ghallab et al.

[75]. The method, described in Algorithm 1, feeds the initial state Si a set of tasks and an initial graph with the first propositional layer as the input. In Algorithm 1, Lines 3 to 4 expand the graph, which may contain a solution. At Lines 7 and 10, the model tries to extract a solution from the graph, with Lines 7 to 13 performing a backward search of the GP algorithm. Line 2 defines i for the layer, ∇ for a no-good table, P0 for the initial proposition, and G for the graph with an initial proposition layer. In Line 3, the model starts to expand the graph until either it contains all propositions in its last layer or no mutex tasks exist in the graph and a fixed point of the graph, but it does not contain a solution. Meanwhile, Line 4 expands the graph for the next layer by adding respective action and proposition layers to the graph. Line 5 returns failure if the model found all propositions in the last layer or none of them is a mutex. In Line 6, η specifies the size of the no-good table. Line 6 extracts the plan again. In Lines 8 to 10, the model again tries to explore the graph to find a solution. In Lines 11 to 13, the model sets the termination condition. Then the model checks the size of the no-good table (η). If the size is unchanged, the model returns failure because, even if the process continues, if nothing was found to change, the opportunity to find a solution will remain nonexistent. Finally, Line 14 returns the proposed planner.

4.3.3 Plan Generation

According to the ongoing research previously described for the motivation scenario, two data profiles are available for identifying the correlation between weather effects and power generation. Data were generated every minute in six locations over the most recent three-year period. However, these data were stored without performing any filtering or cleaning processes. Fig. 4.2 shows how YAWL [77] was used to

Algorithm 1: Improved GraphPlan for BDA

Input: S: Initial State, Set of Tasks, Initial Graph

Output: $\prod$: Optimal Task Plan (OTP)

BEGIN

function GraphPlan ()

Initialize layer i, no good table ∇*, propositions* P_0 *and Init graph G*

while (*all propositions* $\cup$ *P* **or** *no mutex*) **and** reached at G_l^P **do** {

expand graph(G) $\rightarrow$ i+1; }

if $\cup P_i$ **or** *no mutex* **then return** failure

$\eta \leftarrow G_l^P \ ? \ |\nabla(\kappa)| : 0$

extract action vs Proposition $\cup$ ($A_i \rightarrow P_i$) planner; $\prod$

while $\prod$=failure **do** {

expand graph(G) $\rightarrow$ i+1

extract action vs Proposition $\cup$ ($A_i \rightarrow P_i$) planner; $\prod$ }

if $\prod$=failure **and** G_l^P **then** {

if size of η does not change **then return** failure

else $\eta \leftarrow |\nabla(\kappa)|$ }

return $\prod$: OTP

END

represent the workflows described here.

According to CRISP-DM, we need to pass through four stages to achieve successful BDA. According to the BDA workflow generation, the respective initial planning states, tasks, main goal, and subgoals should be defined. Here, we describe the plan generation for the goal set in the data preparation for CRISP-DM, previously

denoted by GDP. The proposition set used comprises Missing Value denoted by mv, File Format denoted by ff, Clean Data denoted by cd, Standard Data denoted by sd, Reasonable Data denoted by rd, File in HDFS denoted by fh and HDFS File denoted by hf. The action set used comprises Clean Data denoted by CD, Defects Handling denoted by DH, Format Convert denoted by FC, Data Standardization denoted by DS, Check Reasonable denoted by CR, Prepare HDFS File denoted by PH and Feed to HDFS denoted by FH. Next, we set the initial state as mvTrue, ffDB, cdFalse, sdFalse, rdFalse, fhFalse, and hfFalse. In this way, G_{BDA} represents the set of cleaned data ready for feeding to the HDFS.

According to our algorithm, GP starts with the initial propositional layers shown in Fig. 4.3(a). GP creates connections between propositions vs. actions and actions vs. propositions from the start to the end. Next, unnecessary connections between the proposition and action layers are removed in the backward search, which starts from the last propositional layer P5. Here, it checks the conditions specified for the plan preparation. Then, based on the conditions set for respective actions in the last layer, those actions create the connections from last propositional layer P5 to A5. The dashed lines represent the mutually exclusive relationships, and thick lines represent the accepted actions between a given action and the proposition layers in that phase. Fig. (a) shows the result after the backward search using the GP technique. Fig. (b) shows the task planner created via the task simulation technique specified in Definition 7.

The corresponding plan for the goal set for data preparation is the result of the connections between a set of actions and propotions. According to the task simulation in Definition 7, these action-to-proposition relationships are considered tasks.

Moreover, the GP results shown in Fig. 4.3 identifies two possible connections between A3 and P3, which implies parallel actions in that phase. However, although it would be possible to implement these links sequentially, the proposed method results in an effective planner with a possible shortened time and reduced resources. In parallelizing the mutually inclusive actions, the effect can reduce the overall effective horizontal length of the workflow dramatically. If there are parallel tasks in a given workflow, those tasks can run in parallel using distributed resources instead of running

sequentially. Therefore, it directly affects the makespan and resource consumption of the overall process. BDA is heavily resource dependent and time consuming. Therefore, this is very effective in reducing the overall executing time for the workflow relative to the time and resource-intensive consumption of conventional BDA processes. In this case, the final result of the G^{DP} is Defects Handling → Clean Data → [Data Standardization AND Check Reasonable] → Format Convert to CSV → Prepare HDFS File.

Achieving the goal set for the G_{BDA} involves clustering or classification being performed before the deployment stage of the CRISP-DM. The results of the respective requirements for the planning, including all four CRISP-DM stages, are shown in Fig. 4.4. The figure shows the planner's respective tasks using arrows and the respective sequential, AND, and XOR linking patterns for the planner. This is the simplest possible complex planning requirement that contains all three possible linking patterns (sequential, AND, and XOR). This confirms that it is possible to generate more complex patterns for more-complex constraints in BDA planning.

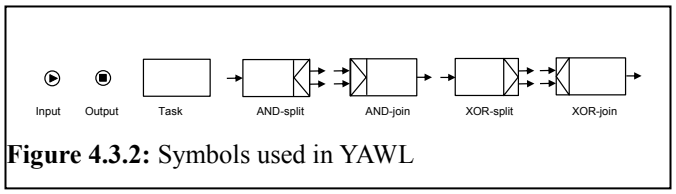
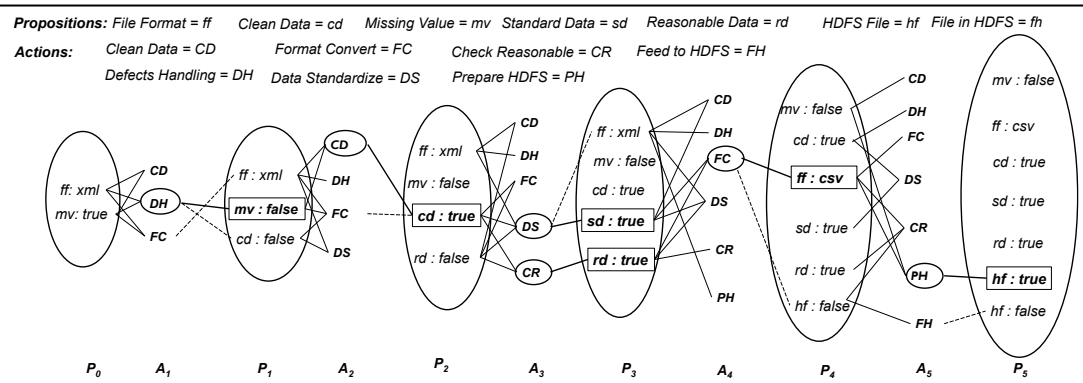
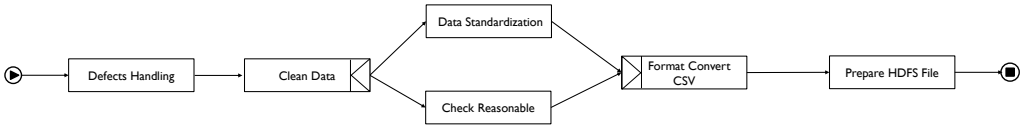


Figure 4.3.2: Symbols used in YAWL



a: GP for G_{DP} with its representation of actions and propositions



b: Generated plan with task representation according to definition 7

Figure 4.3.3: Plan generated for the data preparation stage of the CRISP-DM for the BDA scenario: G_{DP}

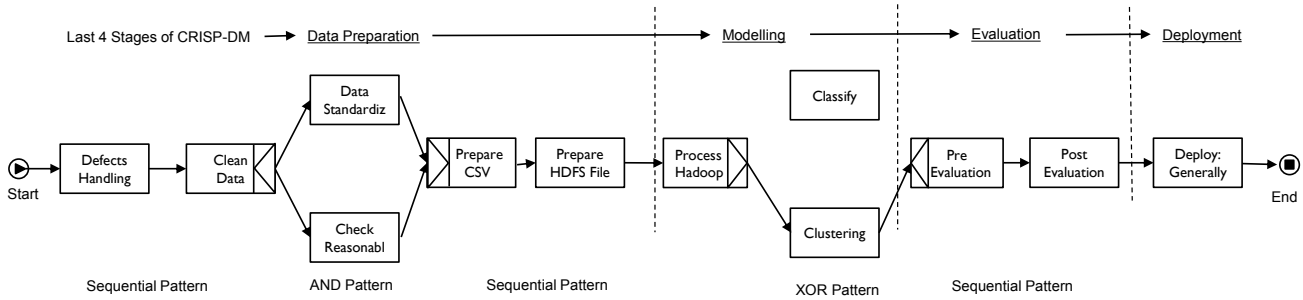


Figure 4.3.4: Plan generation for all four stages of the CRISP-DM in the BDA scenario: G_{BDA}

Chapter 5 Discovery Stage of the ASC

In this chapter first we discussed the research philosophy during the discovery stage of the BDA process. The **Objective** is discovering services for the workflow considering the functional requirements of the tasks. **Key contributions** are discussed in Chapter 5.1 and 5.2. In Chapter 5.1 and 5.2 discuss the ways of achieved the discovery stage. First we present the Domain Ontology based service discovery and next we present the Social Service Network with Multiple Feature Attributes based service discovery. As the **future works**, it needs to improve the accuracy and precision of the proposed method, improve domain ontology and linked social service network behind the solution.

In the era of Big Data, data analysis gives strong competition power to enterprises. As services for Big Data Analysis (BDA) become prevalent, analysis services with intelligence and autonomy using automatic service composition show very bright prospects in the BDA market. Service composition consists of four stages: workflow generation, discovery, selection, and execution. In this paper, we propose a novel service discovery approach that considers two key concerns in the discovery domain towards better quality as well as effective service composition. BDA services are fine grained according to the domain and functional behaviors. The services need a domain context-aware and precision-guided discovery approach. Therefore, we propose domain ontology-based service discovery. It is mainly focused on the BDA domain for precise service discovery considering all behavioral signatures between

queries and services. As for the second concern, components in composed services depend greatly on each other in situations such as workflow for data analysis. We show that linking services together considering sociability or user preference gives better discovery performance. We propose a Linked Social Service Network (LSSN) with multiple feature attribute-based service discovery for BDA. Our approach combines two advantages, the precision and sociability of Web services. The experimental results show that both of these methods perform well based on their perspectives, better than previous approaches.

5.1 Motivating Scenario

In this chapter, we discuss the motivation behind the discovery stage we have been working on. Web service discovery is playing a pivotal role in the process of the ASC. Here we proposed two novel approaches to discovering web services considering two key concerns in the discovery domain towards better quality as well as effective service composition.

5.1.1 Domain aware Precise Service Discovery

As the first concern we have identified is BDA services are fine grained according to the domain and functional behaviors. Then it needs domain context aware and exact functional supportive discovery approach. Here our domain is the BDA and functional behaviors represent steps under the stages of the CRISP-DM process. Therefore we proposed Domain Ontology based service discovery method to discover the precise services from the registry. And the classes and sub concepts of the ontology represents respective stages and steps of the CRISP-DM process. Which contains all possible tasks for the BDA process. Therefore under the precise service discovery perspective, we proposed Domain Ontology based service discovery. Which addresses respective domain as well as behavior signatures of the discovery requirement. Chapter 5.2 discuss the in-detail way of achieve the proposed method.

5.1.2 Facilitate to Effective Workflow Discovery

As the second concern we have identified is services in composed services are

depending each other highly on the domain such alike workflow for data analysis. Then it is better an approach such as Linked Service based discovery. That should be facilitate to do effective workflow discovery to satisfy the given workflow which was result by the planning stage. Therefore we proposed Social Service Network with Multiple Feature Attributes based service discovery to achieve above mentioned requirement. Chapter 5.3 explain the detail steps of the proposed method.

5.2 Introduction

In this section we discuss the common introduction for both methods. According to the Big Data Value Association of the European Union, governments in Europe could save $149 billion by using Big Data Analytics (BDA) with deep learning to improve operational efficiency. BDA can provide additional value in every sector where it is applied, leading to more efficient and accurate processes.[5] Deep learning is an emerging trend in BDA in parallel with predictive analytics. BDA is thus increasing its importance from every aspect and will open more innovation and opportunities than we expect.

In contrast to the exponentially growing importance of BDA, it is still a very time-consuming and resource-consuming process. Users have invested $44 billion in the BDA domain in 2014 alone from industries that are looking for fruitful analytical results [2]. The BDA process raises extreme challenges in data preparation, modeling for analysis and adoption of the matured models. For example, we must understand data, address data quality, and deal with outliers for more meaningful results in data preparation. In modeling, the process requires modeling, testing and re-modeling until it satisfies the requirements for analysis. Adopting a matured model can be considered the basic meaning of deployment, but the trustworthiness of the BDA process must be assured and the model should be precisely articulated to goals and business objectives.

Therefore, we believe that the automation of the BDA process is the most desirable approach to the BDA domain. As the first step, we have previously proposed

[5] http://www.bdva.eu/sites/default/files/europeanbigdatavaluepartnership_sria__v1_0_final.pdf

an intelligent BDA architecture based on Automatic Service Composition (ASC) [11]. ASC is a well-known technology for automating diverse stepped intelligent processes [4]. As the second step, we successfully achieved the planning stage of the ASC, which is the first stage of the ASC process [15], [78]. In this paper, we have addressed the second stage of the ASC process, which is the discovery stage for the BDA domain. We have addressed the two major concerns of effective service discovery and efficient service composition for the BDA procedure.

According to our experience and studies of the BDA domain, we have identified that BDA services are fine grained according to the domain and context. For example, in the data preparation stage, the composition system should distinguish ConvertFileXmltToCsv vs ConvertFileExcelToCsv and in the modeling stage, ClusteringWithKMean and ClusteringWithRepetitiveKMean. The composition system must also incorporate domain context-awareness and precision-guided service discovery for effective service discovery. Therefore, we propose a domain ontology-based service discovery method to identify the precise services from the service registry. Our domain is BDA and the functional behavior of the ontology represents the stages of the Cross-Industry Standard Process for Data Mining (CRISP-DM), which is the foundation of the data science process of the intelligent BDA architecture that we proposed in the first phase of the overall research [3]. The classes and subconcepts of the ontology represent the respective stages and steps of the CRISP-DM process. The domain ontology-based service discovery method aims to exploit semantic meaning of the matrixes that are used in the services to acquire hidden domain knowledge. It also includes a behavioral signature-level approach to ensure the highest possible precision rate. This method is oriented to discover the precise services from the service registry that fulfil effective service discovery for the ASC of BDA process automation.

For our second major concern, we have studied efficient service composition for ASC in the BDA domain. BDA services are highly dependent on each other in situations such as workflow for data analysis. For example, data preparation stage, modeling for analysis and deployment are unavoidable stages of the BDA process. Each stage depends on the prior stage and therefore the stages of the BDA process are heavily

interdependent. In most cases, BDA service consumers are not limited to a single service request from a service repository; they want to locate multiple services that can work together. This allows peer users to address more complex functions by combining services in an efficient manner. Satisfying complex functions is one of the biggest challenges in the BDA process. This means that according to the workflow, these services are consumed regularly and therefore show strong social interaction with peer services within the service network (registry). Therefore, it is better to use an approach such as linked service-based discovery [6]. This aims to facilitate efficient workflow discovery. Discovering workflows is the most recognized approach to efficient service composition. However, in most cases, such approaches are oriented to achieve solutions that are near optimal rather than more accurate [7]. From the BDA perspective, however, services are fine grained according to the domain and context. We therefore have an additional constraint on BDA for effective service composition for workflow discovery: to maintain accuracy as well as optimality. We propose a Linked Social Service Network (LSSN) with multiple feature attributes-based service discovery to achieve both constraints while seeking efficient service composition for the BDA domain.

5.3 Domain Ontology based Service Discovery

Domain Ontology based service discovery has three main stages and seven sub stages under main stages.

Stage 1: Initial Setup

a. Build the BDA Domain Ontology
b. Prepare the service registry

Stage 2: Clustering

c. Calculate Similarities between services available in the Service registry
d. Cluster the service registry based Similarities between services

Stage 3: Discovery

e. Find most suitable cluster group

f. Calculate Similarity : Task vs Cluster Group

g. Discovering Services

Here below we will explain each stages and their sub stages. Each stages and their sub stages are linked with each other. But no need to be repeated for every discovery process. That means, Stage 1 and Stage 2 need to be done only one time for given services in the service registry. All these seven steps mentioned above have been streamlined and once it is completed full cycle it just need execute only Stage 3 for different type of discovery requirements.

Here below Figure 5.3.1 shows the flow chart of the proposed method of the Domain Ontology based service discovery.

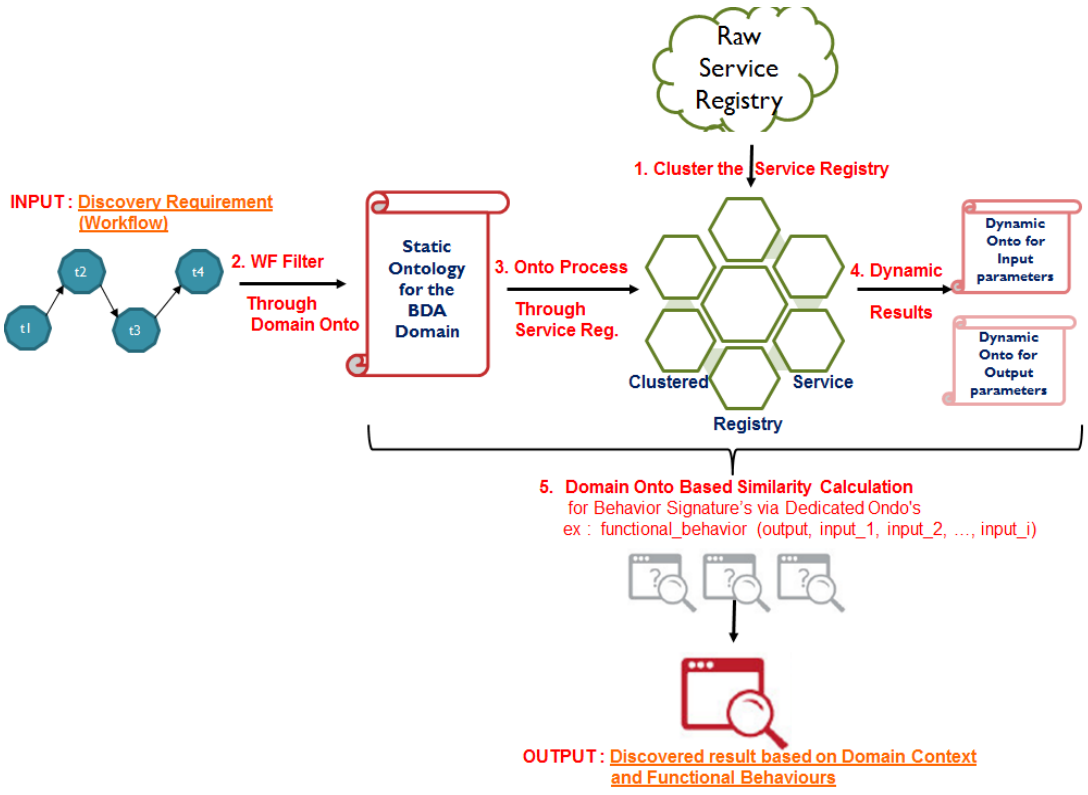


Figure 5.3.1: Flow of proposed method for the precise service discovery

Our aim is to address the domain context as well as the exact behavioral signature to satisfy most precise service discovery for the BDA domain. Then we have been using matured BDA domain that contains all possible tasks which are needed in

BDA process and respective parameters used to represent behavioral signature such as output and input parameters. Here below explained the way achieved the proposed method in step wise declaration. Experiments results show the overall performance of the method based on respective perspective.

5.3.1 Stage 1: Initial Setup

A. Build the BDA Domain Ontology

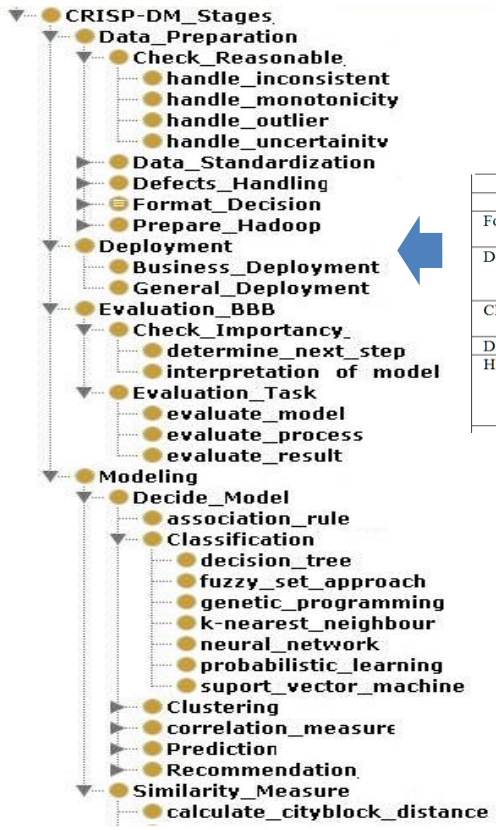
We defined domain ontology through generating ontology classes for the stages in CRISP-DM that required services. These services are decided based on the domain experiences and the knowledge. One of the biggest advantage is that this domain ontology can be update according to the requirements. Ontology is a specification of objects, categories, properties and relationships. Figure 5.3.2 displays the part of the ontology we made for the BDA.

For given service, that is lowest level sub concept of the ontologies has their own inputs and output variable. These have been defined as data properties of respective name individuals of them. Figure 5.3.3 displays sample data properties assignation for given named individual.

All of the possible tasks, which are generating from CRISP-DM process have defined in the Ontology. Here this ontology is considering as the domain ontology of the BDA process. Next our process is based on this BDA domain ontology. Then we use this ontology to calculate the similarity between abstract task and web services. If it complex task requirement, such if it needs same tasks with multiple different I/O parameters, we can define another object property concept under given sub concept and define required I/O for that.

All of the possible tasks, which are generating from CRISP-DM process have defined in the Ontology. Here this ontology is considering as the domain ontology of the BDA process. Next our process is based on this BDA domain ontology. Then we use this ontology to calculate the similarity between abstract task and web services. If it complex task requirement, such if it needs same tasks with multiple different I/O parameters, we can define another object property concept under given sub concept and define required I/O for that.

Next our discovery process is starting as follows. First of all we clustered the existing service registry based on semantic similarity between existing services in the service registry. It results list of groups, each group contained cluster center and a member list, which is set of services contained in service registry. After that our discovery process is proceeded with these clusters vs domain ontology that we have defined in the beginning of the process.



Data Preparation Step	
Task	Sub Tasks
Format Decision Task	Convert_SQLtoCSV, Convert_XMLtoCSV
Defects Handling Task	Handle_MissingValue, Handle_Duplicates, Handle_NoisyData
Check Reasonable Task	Handle_Outlier, Handle_inconsistent, Handle_Uncertainity
Data Standardization Task	doNormalization, Handle_Synonyms
Hadoop Infrastructure Task	PrepareKV, InterfaceHDFS, setHadoopNodes, setHadoopBlockSize, setMapReduceApps

Figure 5.3.2. Sample domain ontology vs part of the task table

B. Prepare the Service Registry

We have prepared the service registry with respective services which are needed to Big Data Analytical process.

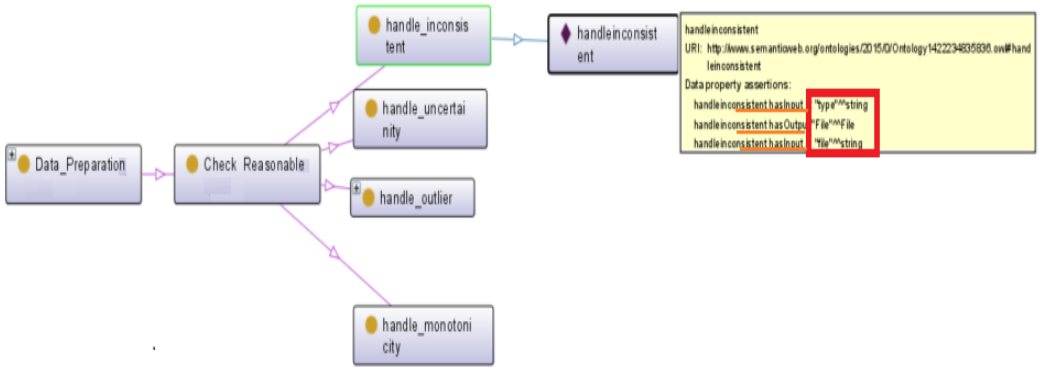


Figure 5.3.3: Sample way of assigned inputs and output to given service

5.3.2 Stage 2: Clustering

Here we clustered the service registry to reduce the search space. It is dramatically optimized the time factor of the discovering process. It needs to follow C and D steps to be completed the clustering process.

C. Calculate Similarities between services available in the Service registry

Similarity calculation is done by Hybrid Term Similarity calculation method [4].

D. Cluster the service registry based Similarities between services

Clustering process was one of the important part of the process. It has dramatic effect to the time factor. Here below we describe the clustering method and its important factor, which is cluster center identification method of the process. Here we use the input as the Hybrid Term Similarity Result of the C step.

5.3.2.1 About the Clustering Method

We have been using Ontology learning based clustering approach to cluster our service registry [4]. In this case, first we calculated the services similarity contained in the registry. Then we use Agglomerative clustering algorithm which is handling any type of similarities or distances between services in effective way. Since it can retrieve the main body of the data and has good computation power also. Below Algorithm 1 shows the high level view of the clustering algorithm.

This bottom-up hierarchical clustering method starts by assigning each service to its own cluster (Lines 1). It then starts merging the most similar clusters, based on

proximity of the clusters at each iteration, until the stopping criterion is met (e.g., number of clusters) (Lines 4 to 10). Several methods have been used to merge clusters, such as single-link and complete-link [34]. We use a centroid-based method where, for the proximity value, we use SimS(Si, Sj) between cluster centers. Figures 5.3.4 (a) and (b) show an example of the clustering steps, with Figure (c) showing a tree representation.

Cluster Center identification method

It was very important to identify most suitable cluster center to optimize cluster performance. In this sub-section. Here is the way of find the cluster center. To calculate the center, it presume the following condition. A center service of a service cluster has highest value of the summation of TF–IDF value of service name and average relative similarity among services in the cluster. According to the condition, it defined following equation. First, it calculate the center value CV for all services in the cluster (see Lines 7 in Algorithm 2) and then choose the service with the highest value as the cluster center.

$$cv(S_{i,c}) = \sum_{1}^{m}\left(\frac{Sim_s(S_i,S_j)}{m(m-1)}\right) + tfidf(S_{i,c})$$

Here, $CV(S_i,c)$ is center value of service Si in cluster C. Parameter m is the number of services in the cluster. $Sim_S(S_i,, S_j,)$ is similarity values between service S_i and S_j in cluster C. Average similarity value is between 0 and 1. Value $tfidf(S_i,c)$ is TF-IDF value of service name Si within the cluster C. It reflects how important a service is to a collection of services. To calculate the TF–IDF value, first we tokenize all the service names in the clusters into individual terms and then calculate the TF–IDF value of the individual terms as:

$$tfidf_{x,c} = tf_{x,c} \log\frac{n}{Cn_x} +$$

Here, $tfidf_{x,c}$ is the TF–IDF value for term x in cluster C. $tf_{x,c}$ is the term frequency of term x in cluster C. Cnx is the number of clusters that contain term x. Parameter n is the number of clusters. Finally, we calculate the average TF–IDF values for terms used in service name as the TF–IDF value of service name as shown in below and k is the number of individual terms used in service name and average TF-IDF value

is between 0 and 1.

$$tfidf(S_{i,c}) = \frac{\sum_1^k tfidf_{x,c}}{k}$$

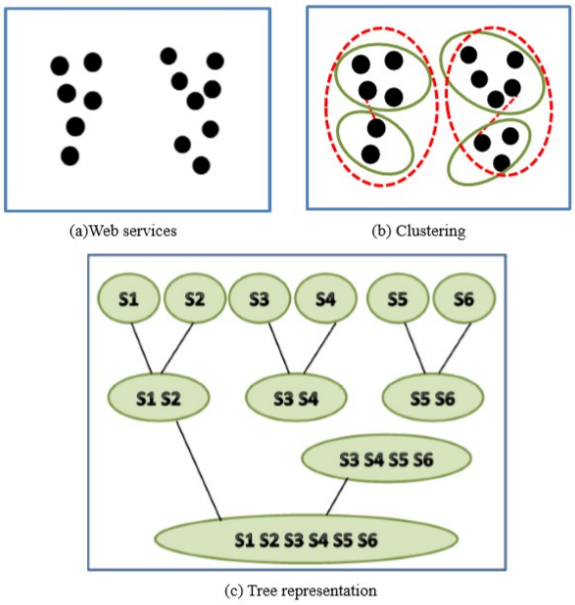


Figure 5.3.4: Agglomerative clustering

5.3.3 Stage 3: Discovery

In the planning stage of the ASC framework, a Workflow Generator creates an abstract workflow to satisfy the functional property of user requests. Here first we proposed an ontology-based method to generate the plan in our previous work: we can define, a plan for the planning problem as $\pi(a_1, a_2, \ldots a_n)$, where a_i is a classical action. The sequence of actions will form an abstract workflow guiding service composition. Idea of the service discovery is to find candidate web services for each action a_i of the abstract workflow. Discovering services are based on the weights of the similarity values of query (tasks) vs services in the service registry. It is the abstract description of the discovery process.

As it mentioned above, we have been following E, F and G steps to complete the proposed method. Before it presents E, F and G steps, it is an important to discuss the way of calculating similarities between Tasks vs Services.

5.3.3.1 Calculate Semantic Similarity between Tasks vs Services

The process of similarity calculation can be divided in to the main three steps.

Step 1: Semantic Features Extraction

Step 2: Ontology Learning (used only for inputs and outputs)

Step 3: Feature Similarity Calculation

Step 1: Semantic Features Extraction

In calculating the similarity between a web service and task. We extract the service features (inputs, output and service name of given WSDL) from WSDL files. And Tasks features (lowest level sub concepts and their data properties) from domain ontology.

Step 2: Ontology Learning

Here we used ontology learning process for inputs and outputs features. Because we need to identify only service-specific terms relevant to the service domain and more meaningful terms in generating the upper-level concepts of respective dynamically creating ontologies of inputs and outputs. In here, first we find the extracted feature (ex: input) is complex or not. If the feature is a complex term (ex: inputFile), then it divides in to two parts (input, File) based on the assumption that the capitalized letters indicate the start of a letter. In addition to that, we tokenize terms based on hyphen (-) as well.

Next, we find the TF-IDF values of all the tokenized words. These terms rank based on their values, with the highest-ranked word having the highest TF-IDF value and a threshold TF-IDF value is defined. Then it results service specific terms relevant to the service domain and more meaningful terms in generating the upper-level concepts. That means, we can generate mostly relevant BDA domain specific ontologies from respective features (input and output).

Next, it uses to generate the concepts and relationship between concepts using the TF-IDF value ranking and rules. It chooses a word of highest rank and generate a concept for that term. Then it selects all complex terms that make use of that word to build the complex term by taking it as its head. It uses a method call “Ontology Generation” in previous research [4]. Algorithm 2 presents the flow of creating ontologies for input and output features. Here we used dedicated ontologies rather than

single one to assure more precise semantic feature identification in different perspective which are service names, inputs and outputs in BDA domain.

This allows user to change the parameter (multi-parameters for given task) of respective tasks based on their requirements and retrieve most equivalent services discovery result beyond single perspective static domain ontology. As an example, assume for workflow_1 needs Format_Decision tasks required output is Boolean type, inputs are inputFileType and fileLocation. But to satisfy workflow_2 it needs same taks (Format_Decision) but required output is File and input is fileLocation only. Thus same tasks but looking for different parameters. That means, these services are looking for different services from the registry. In this case we need to add two different object property concept to the ontology and define their respective data properties to satisfy this kind of scenario. As an example we could be able find two different service from the registry for same tasks but with different parameters.

```
Input  Tc: Array of complex terms
Input  Tt: Array of tokenized terms
Input  θ : Threshold TF–IDF value
Output O : Ontology
1:      for each tokenized term tt, where TF–IDF value > θ in Tt do
2:        generateConcept(tt);
3:        for each complex term t in Tc do
4:          Ht = getHeadTerm(t);
5:              if(tt.equals(Ht))
6:                generateConceptsforAllLevel-ComplexTerms(t);
7:               end
8:              generateSubSuperRelationship(); // By Rule 1.
9:            end-for
10:     end-for
11:     for each complex term t in Tc do
12:       Ht = getHeadTerm(t);
13:       Mt = getModifierTerm(t);
14:       If (Ht is not a concept and Mt is a concept)
15:     generateDataProperty(); //By Rule 2.
16:       end
17:     end-for
18:     for each concept Ci do
19:       for each concept Cj do
20:     generateObjectPropertyforConceptModifier(); // By Rule 3.
21:       generateObjectPropertyforModifierOnly();  //By Rule 4.
22:        end-for
23:     end-for
```

.

Algorithm 2. Ontology Creating

Task_1: set_hdfs_node and parameters are output is Boolean, inputs are inputFile & location.

Task_2: set_hdfs_node but parameters are output is Boolean and input is inputFile only.

Let's consider our service registry has 2 services which parameters are as follows, Service_1 = Boolean set_hdfs_node (inputFile, location) and service_2 = Boolean set_hdfs_node (inputFile). According to our method of calculation similarity, it gave this result.

Task_1 vs Service_1 =1.0

Task_1 vs Service 2 = .9570000171661377

Task_2 vs Service_1 = 0.9641666859388351

Task_2 vs Service_2 =1.0

Due to use of base domain ontology and two sub domain ontologies for respective categories, it can be achieved more precise discovery result even it uses same tasks but different parameters.

Step 3: Feature Similarity Calculation

In similarity calculation process, we use the Hybrid Term Similarity Calculation method to calculate the similarities. Mainly we have three types of features to be calculated the similarities. Which are names, input and output. Here feature similarity calculations are doing based on BDA domain ontology and two sub derived dynamic domain ontologies described in above. Name feature uses above defined BDA domain ontology, Input and output features use dynamically resulting domain model ontologies as above mentioned. Hybrid Term Similarity concept contained two methods in calculating the similarities. Which are ontology based term similarity calculation and Information Retrieval (IR) based term similarity calculation.

Ontology based Term Similarity calculation Method

In calculation process of similarities between tasks vs services we check the existence of concepts with respective ontology. If the concept is exist then we compute the degree of semantic matching by applying different filters. We defined the following filters for calculating the similarity. If the given pair does not exist that pair proceed to IR based method to calculating the similarities.

Ontology is an explicit specification of a conceptualization. Relations describe the interactions between concepts or a concept's properties. We consider two types of relations, namely concept hierarchy (Subclass–Superclass) and triples (Subject–

Predicate–Object). Let C be a set of concepts $\{C_1, C_2, \ldots, C_n\}$ in the ontology. Here, C_i represents Si_F, which is a feature F (e.g., service name) of service Si. $LSC(C_i)$ is the set of least specific concepts (direct children) C_x of C_i . That is, C_x is an immediate sub-concept of Ci in the concept hierarchy. $LGC(C_x)$ is the set of least generic concepts (direct parents) C_i of C_x. $PROP(C_i)$ is the set of properties of concept C_i.

Definition 1: Exact Match: Service S_i is an exact match with abstract task a_j if$C_i \equiv C_j$, where $C_i$and C_j are concepts in ontology $O_p$and represent the service S_i and task a_j, respectively.

Definition 2: Plug-in Match: Service S_i plugs into a_j if $C_i \in LSC(C_j)$, where C_i and C_j are concepts in ontology $O_p$and represent the service S_i and task a_j, respectively. $LSC(C)$is the set of least specific concepts (direct children).

Definition 3: Sibling Match: Service S_i is a match (sibling) with a_j if $C_i \in LSC(C_k) \wedge C_j \in LSC(C_k)$. C_i and C_j are direct children of concept C_k, where C_i and C_j are concepts in ontology $O_p$and represent the service S_i and task a_j, respectively, with C_k being another concept in ontology O_p.

Definition 4: Subsumes Match: Task a_j subsumes service S_i if $C_j > C_i$, where C_i and C_j are concepts in ontology $O_p$and represent the service S_i and task a_j. $C_i$is more specific than C_j.

Definition 5: Logic Fail and Fail : Service S_i "logic-fails" to match a_j if C_i and $C_j$are in the same ontology $O_p$but fail for all types of match filter described above. Service S_i "fails" to match a_j if there no concepts available in ontology.

Logic-based matching considers the filters according to the following order, Exact > Plug-in > Sibling > Subsumes > Logic Fail> Fail.

If there is an exact match between two concepts, then the similarity is equal to the highest value, 1. If the matching filter is Plug-in, Subsumes, Property, Property-&-

Property, Sibling or Logic Fail, we then use (7) to calculate the similarity.

$$\mathrm{Sim}(C_i, C_j) = W_m + W_e * \mathrm{ESim}(C_i, C_j) \qquad (1)$$

The values of weights W_m and W_e are real values between 0 and 1 determined by the matching filters. $\mathrm{ESim}(C_i, C_j)$ is the edge-based similarity calculated from (2), where $d(C_i, C_j)$ is the shortest distance between concepts $C_i$andC_j. D is the maximum depth of the ontology.

$$\mathrm{ESim}(C_i, C_j) = -\log\left(\frac{d(C_i, C_j)}{2D}\right) \qquad (2)$$

If services fail to match via any matching filter except Fail, then WordNet is used.

Information Retrieval based Term Similarity Calculation Method

If it fails satisfy any of above mentioned filters, term pair proceed to Information Retrieval (IR) based term similarity calculation. It has two approaches, which are thesaurus based term similarity and Search Engine base term similarities.

Thesaurus-based term similarity: This method can be considered as a knowledge rich similarity-measuring technique, which requires a semantic network or a semantically tagged corpus to define the concept of a term in relation to other concepts or within the surrounding context. We use WordNet as the knowledge base. To calculate the semantic similarity of two terms we use an edge-count-based approach [35], which is a natural and direct way of evaluating semantic similarity in the taxonomy field.

SEB term similarity: One main issue for the above method is that some terms used in Web services may not be included in the thesaurus. We may therefore fail to obtain a reasonable similarity value for features (e.g., "IphonePrice" and "NokiaPrice"). However, the SEB method can overcome this problem because it analyzes Web-based documents. Further, it can identify the latent semantics in the terms (e.g., the semantic similarity between "Apple" and "Computer").

We used three algorithms called Web-Jaccard, Web-Dice and Web-PMI, as described in [36]. Below three equations shown the calculation of weights between terms. Here, H (P) and H (Q) are page counts for the queries P and Q, respectively. Value H (P∩Q) is the conjunction query P AND Q. All the coefficients are set to zero if H (P∩Q) is less than a threshold, c, because two terms may appear by accident on the same

page. N is the number of documents indexed by the search engine. Further, all similarity values are between 0 and 1

$$Web\_Jaccard(P,Q)\begin{cases}0, & if(H(P\cap Q))\le c\\ \dfrac{H(P\cap Q)}{H(P)+H(Q)-H(P\cap Q)}, & otherwice\end{cases}$$

$$Web\_Dice(P,Q)\begin{cases}0, & if(H(P\cap Q))\le c\\ \dfrac{2H(P\cap Q)}{H(P)+H(Q)}, & otherwice\end{cases}$$

$$Web\_PMI(P,Q)\begin{cases}0, & if(H(P\cap Q))\le c\\ \log_2\left(\dfrac{\frac{H(P\cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}}\right), & otherwice\end{cases}$$

First, it computes the pair similarity of the individual terms used in complex terms to calculate the feature similarity value as follows,

$$Sim(T_1,T_2) = \alpha Sim_T(T_1,T_2) + \beta Sim_{SE}(T_1,T_2)$$

Here, $Sim_T(T1, T2)$ is the thesaurus-based term-similarity score and $Sim_{SE}(T_1, T_2)$ is the SEB similarity score. Parameters α and β are real values between 0 and 1, with α + β =1 and represent weights for thesaurus-based and SEB similarities. Final similarity value is between 0 and 1.

Finally calculate the feature similarity value as follows,

$$Sim_p(S_1,S_2) = \sum_{p=1}^{l}\sum_{q=1}^{m}\frac{\max\_sim(x_p,y_q)}{(l+m)}$$

Where X_p and Y_q denote the individual terms, with l and m being the number of individual terms in a particular feature (service name) of services S_i and S_j, respectively. Further, feature similarity value is between 0 and 1.

E. Find the Most Suitable Cluster Group

We compares tasks vs cluster centers resulted by Stage 2 to find highest similar

cluster group for given task. Figure 5.3.5 shows the way of achieve it. It uses 5.2.3.1 described similarity calculation process to calculate similarity between tasks vs center services.

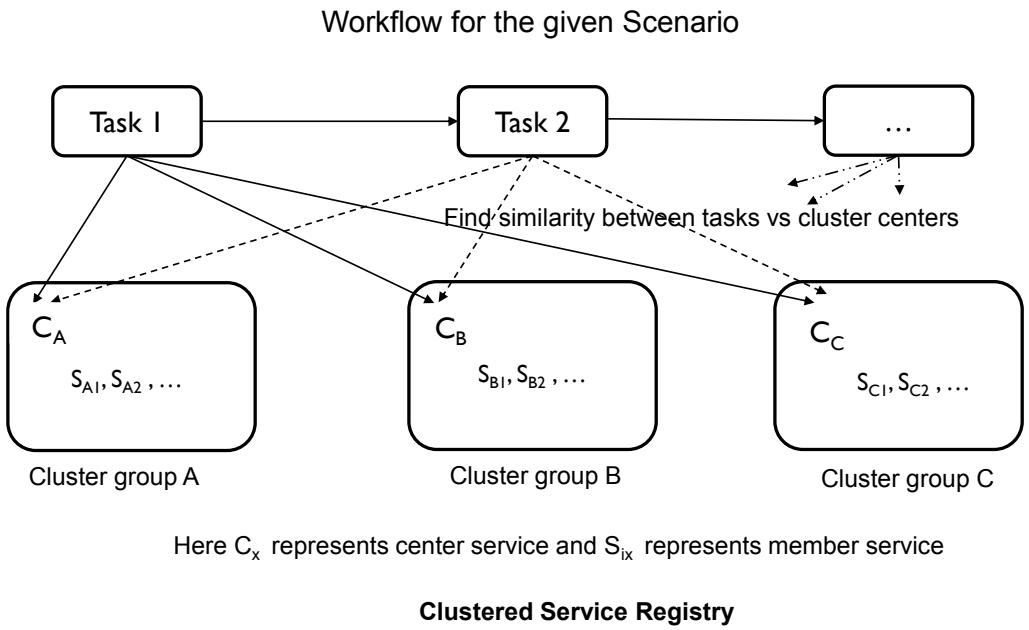


Figure 5.3.5: Find the optimal cluster group

According to the results of Similarity values between Tasks vs Centers, it is finding highest similar center. And then it considers that the center service contained group as the most similar group of services contained for given tasks. Discovering process for given task is proceeding based on the group selected by this step.

F. Calculate Semantic Similarity: Task vs Cluster Group

Here it calculates similarity between each Task vs respective cluster group members. And it uses 5.2.3.1 described similarity calculation process to calculate the similarities.

G. G. Discovering Services

In this step, we sort the result of above step and pick highest similar set of services as the discovered services for given tasks.

5.4 SSN with Multiple Feature Attributes based Service Discovery

This discovery method is based on the Renovated Global Social Service Network. And this discovery process also consists three stages and seven sub stages.

Stage 1: Initial Setup

a. Prepare the service registry
b. Build the Renovated Global Social Service Network

Stage 2: Clustering

c. Cluster the R-GSSN service registry

Stage 3: Discovery

d. Find most suitable cluster group
e. Calculate Semantic Similarity : Task vs Cluster Group
f. Calculate Sociable Similarity Value (SSV)
g. Discovering Services

Our aim is to facilitate to the effective workflow discovery for the BDA domain. Therefore, we have been using fine grained method of precise service discovery (5.2 explained) and Services Sociability (via Linked Social Service Network). As per given

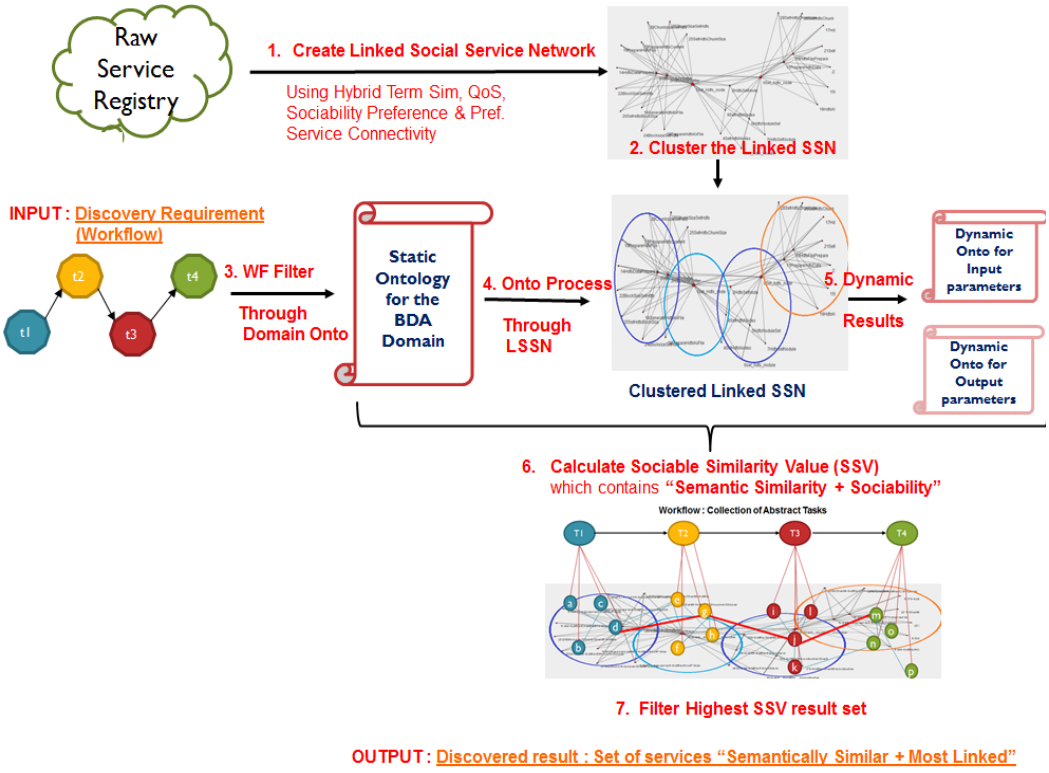


Figure 5.4.1: Flow of proposed method to facilitate to effective workflow discovery

above highest SSV gain by most linked semantically similar services within the network. Thus result most cognitive set of services to satisfy the given workflow in the requirement. That means, our solution is addressed to the domain context, exact behavioral signature and sociability factor to discovering services. Here below explained the way achieved the proposed method in step wise declaration. Experiments results show the overall performance of the proposed method based on respective perspective. Here below Figure 5.4.1 displays the flow chart of the LSSN with Multiple Feature Attribute based service discovery process.

5.4.1 Stage 1: Initial Setup

A. Prepare the Service Registry

We have prepared the service registry with respective services which have been

used in domain ontology based service discovery method.

B. Build the Renovated GSSN

Existing GSSN is created based on four factors. Which are Dependency Satisfaction Rate (DSR), QoS Preference, Sociability Preference and Preferential Service Connectivity (PSC). Then we replace DSR with Hybrid Term Similarity Value between Service Pairs. After that we proceed GSSN creation method and thus result a fresh GSSN. New GSSN is named as the Renovated GSSN (R-GSSN) and it can be identified following as key differences with the existing GSSN,

1. It closed similar services to each other
2. Reasonable Links distribution among the network
3. Better discovery performances

R-GSSN has higher impact of Semantic Similarity between services than GSSN. Therefore we could be able to achieve the more advanced GSSN. Then our discovery

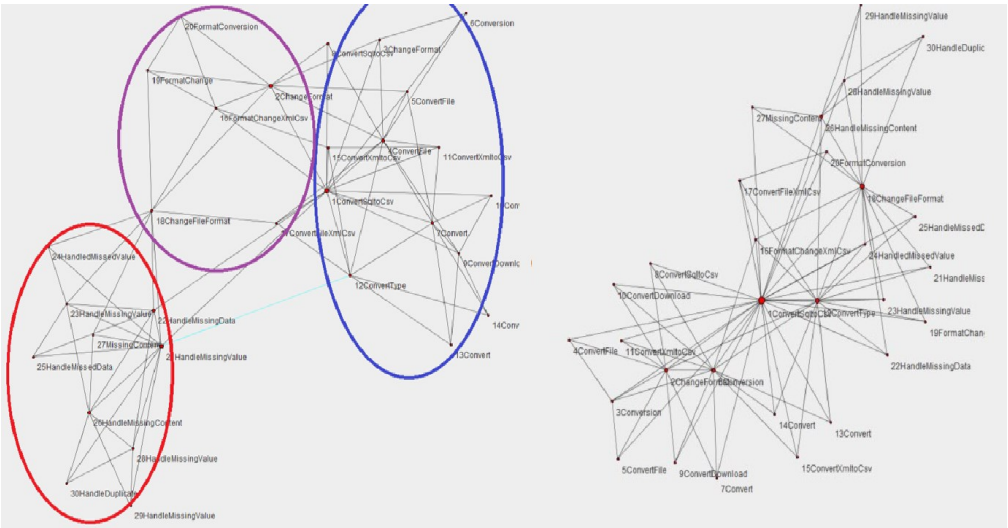


Figure 5.4.2: Sample R-GSSN (LSSN) vs GSSN for 30 services

process is based on this R-GSSN. Figure 5.4.2 shows the sample R-GSSN vs GSSN made by same 30 services.

5.4.2 Stage 2: Clustering

C. Cluster the R-GSSN Service Registry

Here we clustered the R-GSSN service registry to reduce the search space. It is dramatically optimized the time factor of the discovering process. It feeds R-GSSN weight value set resulted during the R-GSSN creation to the clustering algorithm

described in 5.2.2.1. Thus resulted a clustered R-GSSN and our discovery process proceed based on these clustered R-GSSN.

5.4.3 Stage 3: Discovery

D. Find the Most Suitable Cluster Group

We have followed same steps as it described in above mentioned Figure 5.3.5. Thus resulted a clustered R-GSSN and our discovery process proceed based on these clustered R-GSSN. Also we calculate similarity between task vs cluster center service's to find most suitable cluster group for given tasks.

E. Calculate Semantic Similarity: Tasks vs Cluster Group

Here it calculates similarity between each Task vs respective cluster group members. And it uses 5.2.3.1 described similarity calculation method to calculate the similarities.

F. Calculate Sociable Similarity Value

Here we accumulate the sociability and semantic similarity in to a one value. We called it as the Sociable Similarity Value (SSV). SSV calculation as follows.

$$SSV\ (T,S) = \ Sim_F * \ Sim\ (T,S) + \ Soc_F * \ Sociability$$

Here,

$Sim_F =$ "Sim Factor" is constant; here we use $Sim_F = 0.8$

$$Sim\ (T,S) = Semantic\ similarity\ between\ Task\ (T)\ vs\ Services\ (S)$$

$$Soc_F = [1 - (Sim_F * Sim(T,S))]$$

$$Sociability = \frac{\#\ of\ links\ populated\ by\ Services\ in\ LSSN}{Total\ \#\ of\ links\ within\ given\ cluster\ in\ LSSN}$$

G. Discovering Services

In this step, we sort the result of above step and pick highest SSV set of services as the discovered services for given tasks.

Chapter 6 Selection Stage of the ASC

In this chapter first we discussed the research philosophy during the selection stage of the BDA process. **Objective** is selecting services considering the QoS requirements, BDA and ASC domains specific concerns. **Key contributions** are; as the first method, we proposed QoS and customizable transaction aware modelling and algorithm. As the second method, we proposed QoS and traffic awareness model and algorithm for Big Data space service selection. As the **future works**, selection stage needs a trust model and adoptability for more qualitative QoS factors during the selection.

First, we proposed the QoS and customizable transaction aware selectin method, due to the services for BDA become prevalent, analysis services with intelligence and autonomy using ASC show bright prospects in the BDA market. Selection is one of the most important phases of successful ASC process. Moreover, it became competitive with the rise of demand for the services and criticalness of the BDA process. It is a challenge to accomplish a successful uninterruptable composition while serving diverse custom selection requirements. In the case of failure, it results in complete loss of time and resources. Traditional approaches are not applicable to handle failures during long running transactions. Instead, compensation suggests to being an error recovery. Therefore, analytics transactions scheduled as a composition of a set of compensable transactions. However, compensable services are a higher price and consume more time. Moreover, consumers equipped with diverse requirements. It is necessary to guarantee

the critical stages of workflow using compensable services rather than whole workflow. Therefore, we proposed customizable Transaction and QoS-aware service selection approach under five user custom settings based on genetic algorithm (GA) to address above concerns. QoS-awareness facilitated by multi-objective QoS criteria and GA is used for multivariate optimization. We conducted a thorough evaluation, and it shows proposed method effectively and efficiently reach the global optimal of the overall selection criteria.

Next we proposed the Big Data space based service selection, because The number of web services has increased dramatically during the last few years. This has resulted in an increase in the volume of candidate services for tasks in composition systems. This has led to growth in the variety of nonfunctional properties in service selection, resulting in uncertainty (veracity issues) among such properties, which has severely affected the NP-hard aspects of service selection. Despite this, consumers in many areas would like access to a variety of selection methods such as linear programming and dynamic programming techniques. An additional problem is that the composition length (the number of tasks) of the workflow has increased, with the incorporation of research domains such as data science. These trending composition issues are challenging the computational power of existing methods. Such concerns have opened the door to research involving big-data spaces. We propose a flexible, distributed selection algorithm that facilitates heterogeneous-selection methods to satisfy multiobjective composition requirements rather than rigid, specific composition requirements. However, service-selection processes in a big-data space will inevitably increase traffic congestion caused by the increased volume of internal communication, particularly external traffic such as Zipf and Pareto phenomena and internal traffic during shuffling. To address these concerns, we propose solutions for each case. Our experiments demonstrate that the proposed traffic-efficient multiobjective method is well behaved when selecting services in big-data spaces.

6.1 Motivating Scenario

Two selection scenarios are used for the given selection stage.

6.1.1 QoS and Customizable Transaction aware Selection

Already completed and related content is pending to be added to the thesis. Figure 6.1.1 shows the workflow for the given selection scenario. As it shown in Fig. 6.1.1, it has three sequential patterns ($CWS_1^S, CWS_2^S, CWS_3^S$), one AND pattern ($CWS_1^A$) and one XOR pattern ($CWS_1^X$). Existing TS aware selection methods proposed TS-awareness apply to the complete workflow. However, in BDA perspective, users are consists with a diverse range of requirements due to the limitation of budget and time. And also, existing patterns in the workflow will not be highly critical, and some patterns already trusted by experiences. Then it is not necessarily to assured by pricier and time consuming compensatable services. Therefore, we define five custom levels of selection criteria's as follows.

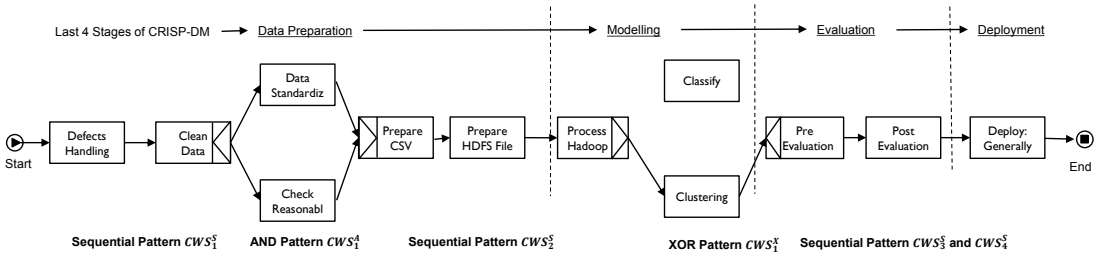


Figure 6.1.1. BDA workflow with three basic composition patterns

Level 1 (L1) Custom Sequential, AND Patterns and Quantitative QoS awareness: It guarantees the TS awareness for the selected *Sequential* and *AND patterns* in the BDA workflow. It will contain at least one of two patterns or multiples of given patterns.

Scenario 1: BDA user has $WF_{BDA,}$ and he needs to find a global optimal composition plan, which assured the TS-awareness in only during CWS_1^S and CWS_1^A places in the workflow while satisfying his multivariate QoS requirement. Then we proposed, L1 to satisfy scenario-1 selection requirement.

Level 2 (L2) Custom Sequential, XOR Patterns, and Quantitative QoS awareness: It guarantees the TS awareness of the selected *Sequential* and *XOR patterns* in the BDA workflow. It will contain at least one of two patterns or multiples of given patterns.

Scenario 2: BDA user has $WF_{BDA,}$ and he needs to find a global optimal composition plan, which assured the TS-awareness in only during CWS_1^S and CWS_1^X places in the

workflow while satisfying his multivariate QoS requirement. Then we proposed, L2 to satisfy scenario-2 selection requirement.

Level 3 (L3) Custom AND, XOR Patterns and Quantitative QoS awareness: It guarantees the TS awareness of the selected *Sequential* and *AND patterns* in the BDA workflow. It will contain at least one of two patterns or multiples of given patterns.

Scenario 3: BDA user has $WF_{BDA,}$ and he needs to find a global optimal composition plan, which assured the TS-awareness in only during CWS_1^A and CWS_1^X places in the workflow while satisfying his multivariate QoS requirement. Then we proposed, L3 to satisfy scenario-3 selection requirement.

Level 4 (L4) Complete workflow assured by TS awareness and Quantitative QoS awareness: It guarantees the TS awareness to the complete BDA workflow.

Scenario 4: BDA user has $WF_{BDA,}$ and he needs to find a global optimal composition plan, which assured the TS-awareness in in the complete workflow while satisfying his multivariate QoS requirement. Then we proposed, L4 to satisfy scenario-4 selection requirement.

Level 5 (L5) Except TS awareness, only Quantitative QoS awareness: It guarantees Quantitative QoS awareness throughout the workflow automation, not considered TS awareness.

Scenario 5: BDA user has $WF_{BDA,}$ and he needs to find a global optimal composition plan. He does not expect to the TS-awareness to the workflow and only looks forward to satisfying his multivariate QoS requirement. Then we proposed, L5 to satisfy scenario-5 selection requirement.

6.1.2 QoS-aware Rule-based Traffic-efficient Multi-objective Selection

We define service-selection requirements that are identified as three of the most common types. We use these three requirements to show the adaptability of the proposed method. One of these three requirements will be used in the relevant selection scenario

during the MR process in the Big Data space.

- QoS values for web services (WS) that are affected positively by the performance of the service are denoted by QoS^P, and there will be x QoS constraints for the given WS WS_{ij}. Here, $x,\ i,\ j\ > 0$, i represents the i^{th} task in a composition planner, and the j represents the j^{th} candidate service for the given i^{th} task.

- QoS values of WS that are affected negatively by the performance of the service are denoted by QoS^N, and there will be y QoS constraints for the given WS WS_{ij}. Here, $y,\ i,\ j\ > 0$ and $x + y$ represents the total number of QoS values for the given WS_{ij}.

- The weight values of QoS^P and QoS^N are represented by $w_\propto$ and w_β, respectively, such that $\sum_{\propto=1}^{x} w_\propto + \sum_{\beta=1}^{y} w_\beta = 1$ and $0 < w_\propto . w_\beta < 1$.

- T is the number of tasks in the workflow, such that $0 < i \leq T$. $v_\propto^{avg}$ is the average value of the particular QoS attribute among the given candidate services and $v_\propto^{max}$ and $v_\propto^{min}$ are the maximum and minimum values of the respective attributes.

$Q_{All}^{Constraints}$ is the collective value of negatively affected QoS attributes set by the user, with $Q_\beta^{Constraints}$ being an individual constraint.

Based on these notations, we define the heterogeneous-selection problem using Definitions (and Scenarios) 1, 2, and 3 below.

Definition 1: Linear optimal service selection. Select the composition plan that represents the global optimal solution with the most profitable overall QoS criteria.

Scenario 1: Users need to find the global optimal service composition sequence that

$$Max \sum_{i=1}^{T} \left(\sum_{\propto=1}^{x} \left(\frac{V(ws_{i\propto}) - v_\propto^{avg}}{v_\propto^{max} - v_\propto^{min}} \right) w_\propto + \sum_{\beta=1}^{y} \left(1 - \frac{V(ws_{i\beta}) - v_\beta^{avg}}{v_\beta^{max} - v_\beta^{min}} \right) w_\beta \right) \quad (1)$$

maximizes the overall profit of QoS criteria as shown in Eq. 1.

We propose a solution to the linear optimal service-selection requirement based on the Dijkstra algorithm described in Section III-A.

Definition 2: Combinatorial service selection. Select the composition plan that satisfies the QoS requirements, namely the upper limit of negatively affected QoS criteria and maximum normalized QoS criteria.

Scenario 2: Users need to find the global optimal service composition sequence that maximizes the overall profit and sets the maximum upper bound for negatively affected QoS criteria as shown in Eq. 1 and Eq. 2.

$$Max \sum_{i=1}^{T} \left(\sum_{\propto=1}^{x} \left(\frac{V(ws_{i\propto}) - v_{\propto}^{avg}}{v_{\propto}^{max} - v_{\propto}^{min}} \right) w_{\propto} + \sum_{\beta=1}^{y} \left(1 - \frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}} \right) w_{\beta} \right) \quad (1)$$

$$Subject\ to \sum_{i=1}^{T} \left(\sum_{\beta=1}^{y} \left(\frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}} \right) w_{\beta} \right) \leq Q_{All}^{Constraints} \quad (2)$$

We propose a solution to the combinatorial service-selection requirement based on the 0-1 MCKP algorithm described in Section III-B.

Definition 3: Multivariate optimal service selection. Select the composition plan that satisfies the combinatorial QoS-constrained set of each service, which is the resultant service in the composition plan that satisfies the respective minimum upper limit for negatively affected QoS criteria with respect to the maximum normalized QoS criteria.

Scenario 3: Users need to find the global optimal service composition sequence that maximizes each profit as shown in Eq. 3 subject to the maximum upper limits for each of the negatively affected QoS criteria as shown in Eq. 4. Here, the user sets a number β of constraints for β QoS attributes for candidate WSs for the j[th] task. Here $1 \leq k \leq \beta$.

$$Max \sum_{\propto=1}^{x} \left(\frac{V(ws_{i\propto}) - v_{\propto}^{avg}}{v_{\propto}^{max} - v_{\propto}^{min}} \right) w_{\propto} + \sum_{\beta=1}^{y} \left(1 - \frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}} \right) w_{\beta} \qquad (3)$$

$$Subject\ to \left(\frac{V(ws_{ik}) - v_{k}^{avg}}{v_{k}^{max} - v_{k}^{min}} \right) w_j \leq Q_k^{Constraint} \text{ Here } 1 \leq k \leq \beta \qquad (4)$$

We propose a solution to the multivariate service-selection requirement based on the ABC algorithm described in Section III-C.

The main constraint behind a successful selection process is optimal resource utilization. However, traffic congestion is one of the most constraining factors behind optimal resource utilization in a Big Data environment. This accounts for considerable inefficiencies in the overall process. According to our literature review, we identified two key types of traffic congestion that affect the MR process internally and externally. We, therefore, define key traffic congestion in these two categories as described in Chapter 6.3.2.

6.2 QoS and Customizable Transaction aware Selection

6.2.1 Introduction

Data analytics as a service is one of the leading service based industry, according to the industry survey, it increased the demand for the Amazon analytical services by 293% in 2016 compared to 2013. It has nearly 100% growth rate in each year for Amazon

Web Services6. Services became the prevalent platform of the data analytics, especially Big Data Analytics (BDA). However, BDA process is heavilly time and resource consuming job.

Therefore, we believe that the automation of the BDA process is the most desirable approach to the BDA domain. As the first step, we have proposed a comprehensive architectural design process for the BDA automation based on Automatic Service Composition (ASC) [11]. The ASC mainly contained four stages, planning, discovery, selection and execution [12]. Next, we have proposed novel approaches to achieve the planning and discovery stages [15], [79]. In this paper, we have addressed the selection stage of the ASC process while addressing concerns occurred in the BDA workflow.

BDA contained many tasks, and each task consumes extended periods and the considerably large amount of resources to accomplish given requirements. Nevertheless, these tasks are highly vulnerable due to the highly volatile environment such as bandwidth; infrastructure tends to result in unexpected terminations and errors.

Such interruptions of long running processes hamper the automation process and result loss of data, time and resources. In the BDA perspective, it will be a huge loss. Transactional properties of the services can be utilized to address above concerns in composition process [33]–[37].

However, conventional selection approaches do not consider the transactional awareness during the composition process [80]–[82]. They mainly focused on various aspects of multi-objective quantitative QoS-awareness (Quantitative QoS) and qualitative QoS-awareness (QoS_{QU}) for near optimal or global optimal during the selection. None of these approaches considering the risk of unexpected termination or errors during the composition process. However, the optimal composition does not guarantee the reliable composition during the execution.

Then transactional service (TS) coming into the topic. Here also, some approaches are only limited to transactional awareness other than QoS-awareness [38], [39]. However, Only TS aware approaches are not guaranteed the requirements of the

[1]www.statista.com/statistics/233725/development-of-amazon-web-services-revenue/

multivariate Quantitative QoS awareness. Especially it is a serious drawback of the BDA composition.

Few approaches are proposing composition models considering both TS and Quantitative QoS awareness of the composition [33]–[37]. As per our literature review, only Haddad et. al. proposed ASC based TS and Quantitative QoS aware composition model for the workflow automation under two key risk levels [33]. Moreover, they have proposed semantic TS properties (TSP) identification methodology. It allows defining transactional requirements for the workflow in general not domain specific. It proposed to find the local optimal not global optimal and not flexible for custom user requirements. Z. Ding et. al. proposed genetic algorithm (GA) based approach and it focused global optimal solution [34]. They employed GA with a penalty function for given workflow. Nevertheless, as per our study, it has not considered the workflow automation and proposed algorithm does not provide the flexibility to customization. J. Li et. al. proposed composition model for the Directed Acyclic Graph models but not guaranteed either automation and custom settings [35]. J. Cao et. al. proposed Ant Colony-based TS and Quantitative QoS aware selection for near optimal solution [36]. Y. Cadinale et. al. proposed TS and Quantitative QoS aware selection based on Petri net unfolding algorithm [37]. However, none of these approaches considered the user custom setting of TS for the given workflow or domain specific solution during the composition.

TS services are pricier and consume more time than regular services. However, it provides recovery mechanism to avoid unexpected failures. In the real world, especially BDA service users equipped with diverse requirements, due to the budget and time constraints. Existing transaction aware service selection approaches provides TS and quantitative QoS awareness for complete workflow. Risk levels of TS's and priorities of TP's are essential components in customization of TS-awareness. None of the existing methods are flexible to custom user selection settings for TS awareness to desired locations of the workflow such as critical stages.

To overcome above concerns, we proposed novel,

- TS Risk and TP prioritization for the BDA process and

- Customizable TS and Quantitative QoS aware service selection algorithm for the BDA planned to do in ASC. We used GA as based approach of multivariate optimization.

6.2.2 Preliminaries

In this section, we discuss the key techniques and technologies behind the overall solution. In this paper, we discuss only QoSQN preferences of the non-functional properties (NFP's) and as QoSQU preference of the NFP as the TS awareness according to our study and experience in BDA domain. First, we explain the TSP for web services and composite web services. Next, we explain the multivariate QoSQN properties of the BDA selection process. Here onwards, we use QoS to imply the QoSQN.

A. Transaction aware Compensable Service (TS)

TSP is the behavioral NFP of the web services (WS). It measures the ability to level of successful accomplishment of given task and secure interactivity locally or globally. S. Mehrotra and H. Korth proposed a solid fundamental study on the transactional models [83]. Moreover, J. Hadad et. al. discussed the semantic interpretation and how this transactional model utilizes to service selection domain [5]. Therefore, we inspired by studies of both [5], [15]. Based on their proposals, we lay the foundation of TS awareness of the services as follows under the section A.

TSP's of Web Services: We have identified following properties as TS properties of WS. Property 1- Pivot WS (p): A WS is p if it can be accomplished its task successfully; however, it cannot be rollback and fixed after effects. If it failed, no effects at all. That means result cannot undo semantically. Property 2- Compensatable WS (c): A WS is c if it is available another service, which can semantically undo the execution of the given service. Property 3- Retriable WS (r): A WS is r, it guarantees the successful execution of given number of invocations. Then we can combine these primary properties and can make the secondary properties as follows; Property 4- Pivot Retriable (pr): A WS, which is the pivot and can be retriable as well. Property 5- Compensatable Retriable (cr): A WS, which is compensatable and retriable as well.

TSP's of Composite Web Services (CWS): It is making a CWS to attain a common

goal by coordinating set of WS's. We have identified following TSP's of the CWS as follows. Property 6- Pivot CWS ($\bar{p}$): A CWS is $\bar{p}$, if it is completed successful completion, it effects cannot be semantically undone, fixed and if it failed no effects at all. Property 7- Compensatable CWS ($\bar{c}$): A CWS is $\bar{c}$, if its component WS's are compensatable. Property 8- Retriable CWS ($\bar{r}$): A CWS is $\bar{r}$, if its component WS's are retriable. Then we combine these primary TSP's of the CWS and define secondary TSP's of the CWS as follows; Property 9- Pivot Retriable CWS ($\overline{pr}$): A CWS is $\overline{pr}$, if it is pivot and retriable. Property 10- Compensatable Retriable CWS ($\overline{cr}$): A CWS is $\overline{cr}$, if it is compensatable and retriable.

B. Building Blocks of the Workflow

We have identified three fundamental components of the workflows occurred in BDA process. The Sequential pattern, AND pattern and XOR pattern as basic patterns used to make complex patterns in BDA as of our experience's and literature review [16], [70]. Fig. 6.2.1 shows their graphical representations. We define these three types as follows.

Sequential pattern (CWS^S): Tasks T_1 and T_2 Sequentially executed. That means, T_2 executes after successful completion of $_{T1}$. Then WS_1 and WS_2 assigned to T_1 and T_2, these results in the CWS^S pattern, and it represents as $WS_1 : WS_2$.

AND pattern (CWS^A): Tasks T_1 and T_2 execute in parallel. That means, it executes both T_1 and T_2 at the same time. WS_1 and WS_2 assigned to T_1 and T_2, It results in the CWSA pattern, and it represents as $WS_1 \parallel WS_2$.

XOR pattern (CWS^X): Tasks T_1 and T_2 , it executes either T_1 and T_2 . WS_1 and WS_2 assigned to T_1 and T_2 , This results in the CWS^X pattern, and it represents as $WS_1 | WS_2$.

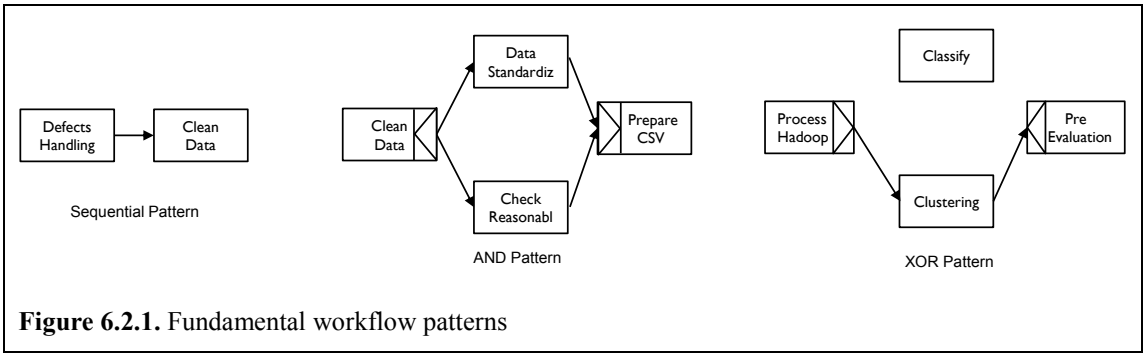


Figure 6.2.1. Fundamental workflow patterns

C. Quantitative QoS Properties of Service (QoSQN)

Here below Table 6.2.1 summarized the QoSQN properties of WS proposed to consider during the BDA service composition. These QoS criteria's consider for the multivariate QoS optimizations using the GA.

Assume, a workflow of a BDA is WF_1; it contains a set of CWS's, $WF_1 = \{CWS_{11}, CWS_{12}, \cdots, CWS_{1n}\}$, here n is the total number of CWS's in the workflow.

Available WS's for each of the CWS, $CWS_{12} = \{(WS_{11}, WS_{12}, \cdots, WS_{1.y1}), (WS_{21}, WS_{22}, \cdots, WS_{2.y2}), \cdots, (WS_{n1}, WS_{n2}, \cdots, WS_{n.yn})\}$, where WS_{ij} represents the j^{th} candidate web service of CWS_i and y_n is the total number of candidate web services of CWS_i

QoS values of WS_{ij} are $qav(ws_{ij})$, $qth(ws_{ij})$, $qre(ws_{ij})$, $qpr(ws_{ij})$ and $qti(ws_{ij})$ for each of the candidate web services of S_{ij}, v_{ij1} , v_{ij2}, v_{ij3}, v_{ij4} and v_{ij5}, where $1 \leq i \leq n$ and $1 \leq j \leq yn$.

Weights for QoS criteria, w1, w2, w3, w4, and w5 for qav(wsij), qth(wsij), qre(wsij), qpr(wsij) and qti(wsij), respectively, where w1 + w2 + w3 + w4 + w5 = 1.

(1), (2) and (3) use in GA algorithm in

$$F(ws_i) = 1 - \sum_{y=1}^{3} \left(\frac{v_y^{max} - v_y(ws)}{v_y^{max} - v_y^{min}} \right) * w_i + \sum_{y=1}^{2} \left(\frac{v_y(ws) - v_y^{max}}{v_y^{max} - v_y^{min}} \right) * w_i \qquad (1)$$

$$F(cws_i) = \sum_{y=1}^{n} ws_i \qquad (2)$$

6.2.3 CTQS: Customizable Transaction and Qos Aware Service Selection

Here we discuss the two proposed approaches under three sections. First, we explain the proposed architecture of the selection method. Next, we explain the custom transaction awareness for the service selection. Finally, we explain the multivariate optimization algorithm of the TS and QoS based on GA.

Table 6.2.1: Representation of QoS_{QN} of Web Services

QoS_{QN} Criteria of the WS	**Functional Representation**
Availability $q_{av}(WS)$	T_t / T_a
Throughoutput $q_{th}(WS)$	R_m / T_u
Reliability $q_{re}(WS)$	N_s / N_{sf}
Price q_{pr}(WS)	*Execution cost per unit*
Time $q_{ti}(WS)$	$T_s - T_r$

Acronyms used in the table as follows.

- T_t represents the total amount of time available, and T_a represents the possible time during last T_a time
- R_m represents completed maximum requests and T_u represents unit time
- N_s represents the total number of successful invokes and N_{sf} represents the total number of successful and fail invokes during given time.
- T_s represents the request sent time and T_r represents results received time.

6.2.3.1 Architecture of the proposed CTQS

Fig. 6.2.2 displays the proposed architecture for the BDA automation. We already achieved the planning and discovery stages of the ASC as of our previous works [15], [78], [79]. In the beginning, user feeds the BDA requirement to the planning stage, and it creates a workflow for the data analysis. Next, this workflow forwards to the discovery stage of the ASC to discover the candidate services to the each task of the workflow. After that, the output of the discovery stage goes to the selection stage. In selection stage, our aim is to find the global optimal plan, which satisfies multivariate QoS of NFP's and assured uninterrupted, error free, successful composition for available budget and time. Finally, the output of the selection stage, which is the ensured global optimal plan put forward to the execution stage to accomplish given BDA requirement. In this paper, we discuss only selection stage of the ASC process considering the concerns we have identified in BDA workflows and ASC process, neither execution nor other previous stages.

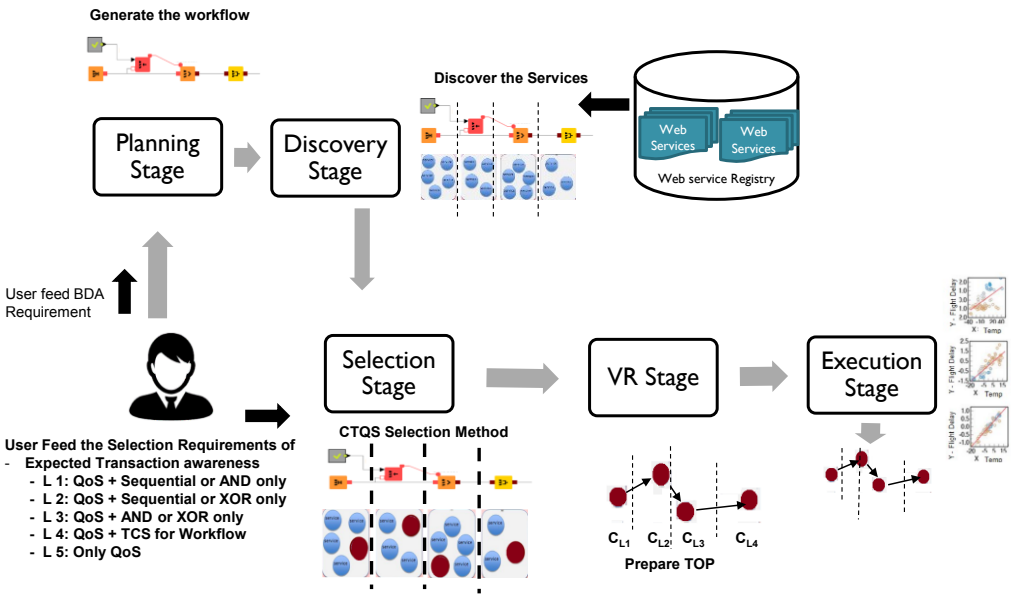


Figure 6.2.2. BDA automation and proposed service selection based on ASC

Next, we elaborate a sample workflow derived for the BDA. Fig. 6.2.3 displays the sample workflow with minimum complexity, which contained only basic patterns. Our data mining process is based on cross industry standard process for data mining. ASC

proposed to automate last four stages of the data mining process. As shown in figure, data preparation stage, contained CWS_1^S and CWS_1^A patterns. In modelling stage, it contains CWS_1^X pattern, finally evaluation and deployment linked by CWS_2^S. We continue the explanation and evaluation of our proposal based on this workflow. Let name this as the WFBDA. Our aim is to derive the global optimal plan for the WFBDA. Then, WFBDA = $\{ CWS_1^S, CWS_1^A, CWS_1^X, CWS_2^S \}$. CTQS method handles each composition of CWS's as identical critical stages and allows BDA user to select which parts he needs to be assured by TS-awareness and QoS-awareness. Next section we discuss the proposed way to approach customization of TS-awareness.

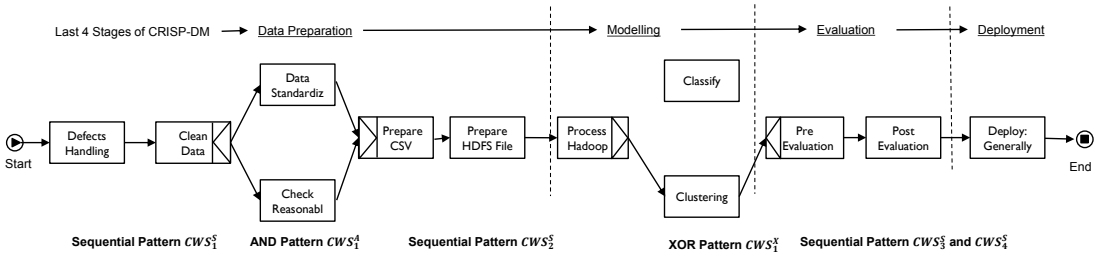


Figure 6.2.3. BDA workflow with three basic composition patterns

B. Customizable Transactional aware Service Selection

In part A, we explained fundamental behind the TSP's. Here, we explain the proposed method for the customizable TS (CTS) aware selection. Let define the risk levels and TSP priority identification of the for the BDA composition.

Risk and TSP Prioritization: According to the TSP's explained in section II, A; define the Lemma to identify the levels of the risks of TP's.

Lemma 1 (Risk levels of WS): TSP set of the WS, SWS = {p, c, r, pr, cr} for all candidate services which belong under each of Ti tasks of the given workflow. Here i ≤ n. n is a number of tasks in the workflow. The descending order list of the risk level of the TS priorities of WS is p, c, r, pr, and cr.

Proof: We prove this by contradictory reasoning.

- According to the Property 1 of the WS, p cannot rollback. Moreover, it tends to have an unsuccessful termination. It is only this property contained these two drawbacks. This is the worst-case can happen to a service. Then

TSP's p is the worse TSP of the WS.

- Property 2, c does not guarantee the successful execution rest of three other minimally guaranteed the successful execution due to retriable property. However, it can undo. Therefore, it is better than p but worse than other three of set SWS
- Property 3, r does not guarantee the pivot retriable or compensatable retriability. Therefore, it is worse than pr and cr. However, it ensures the successful execution after given number execution. Therefore, it is better than p and c properties.
- Property 4, pr does not guarantee the compensatable retriability. It guaranteed pivot retriability. Therefore, it is worse than cr. However, it consists of retriability and pivot property. Therefore, it is better than other three properties of SWS except for c.
- The Property 5, cr it guaranteed retriability and compensability, It is the best assurance can expect from given service. Therefore, cr is the best TCP property of the set of properties contained in the SWS.

Then based on Lemma 1, we define the priorities of TP's of the WS for BDA workflow as follows.

Definition 1 (Priorities of the TSP's of the WS): Priority levels of TS awareness of the TSP's of the WS's according to ascending order as follows: p < c < r < pr < cr. That means A WS cr is the most TS aware property of the WS and p is the least TS aware property of the WS.

Here onwards, we ignore the pivot (p) and retriable (r) properties for BDA automation. Because p does not provide assurance for the tasks of the workflow and r guaranteeing the successful execution only after given number of executions. Then both of properties reduce the effective of the TS awareness of the BDA.

Lemma 2 (Risk level of CWS): TSP's set of the CWS, SCWS = $\{\bar{p}, \bar{c}, \bar{r}, \bar{p}r, \bar{c}r\}$ for all composite services which are belongs under each of WP workflow. Here WP contained Tn is number of tasks in the workflow. The descending order list of risk level of the TS priorities of CWS is $\bar{p}$, $\bar{c}$, $\bar{r}$, $\bar{p}r$ and $\bar{c}r$.

Proof: We prove this by deductive reasoning.

- According to the Property 6 of the CWS, $\bar{p}$ means the collective results of individual p's of WS's of the CWS. Then collective combination of p's result the $\bar{p}$. Then it inherited all p qualities. Which are inability to undone and consistent execution.
- Likewise, according to the rest of Properties (7, 8, 9, 10) of the CWS, $\bar{c}$, $\bar{r}$, $\bar{p}r$ and $\bar{c}\bar{r}$ means the respective collective results of individual c, r, pr and cr of WS's of the CWS. Then collective combination of individual c, r, pr and cr's result respective, $\bar{c}$, $\bar{r}$, $\bar{p}r$ and $\bar{c}\bar{r}$. And, they inherited those TSP qualities as well. Therefore, according to the Lemma 1, CWS's risk quality should be behaved as descending order of is $\bar{p}$, $\bar{r}$, $\bar{p}r$ and $\bar{c}\bar{r}$.

Then based on Lemma 2, we define the priorities of the TSP's of the CWS for BDA workflow as follows.

Definition 2 (Priorities of the TSP's of the CWS): Priority levels of TS awareness of the TSP's of the CWS's according to ascending order as follows: $\bar{p} < \bar{c} < \bar{r} < \bar{p}r < \bar{c}\bar{r}$. That means, A WS $\bar{c}\bar{r}$ is the most TS aware property of the WS and $\bar{p}$ is the least TS aware property of the CWS.

As we done like to TS of the WS, here also, for the same reasons we ignore the pivot ($\bar{p}$) and retriable ($\bar{r}$) properties for BDA automation. Based on definition 1 and definition 2, we elaborated the composition rules of the TSP's of WS and CWS to on behalf of guaranteed TS awareness of the BDA process. TS awareness of WS: Table 6.2.2 shows the summarized information on a combination of TS awareness of the TSP's based on definition 1. We ignore the composition patterns, which are resulted by XOR pattern. Because, during the XOR, it results in one out of two based on the criteria. This applies to the CWS too. Then it results in another Sequential combination. Therefore, we consider only Sequential and AND patterns.

Rule 1 (Composition Rules of TSP's of WS): According to the Table 6.2.2, we have shown the acceptable composition patterns for the successful WS composition of the BDA process based on ASC. Therefore, here we conclude the acceptable composition patterns as follows; Acceptable Sequential patterns are c : c, c : cr, cr : cr and cr : c only.

Moreover, acceptable AND patterns are c // c, c // cr, cr // cr and cr // c. These are only patterns that guarantee the end to end TS-awareness for the BDA process. TS awareness of CWS: Table 6.2.2 shows the summarized information of a combination of TS awareness of the TSP's based on definition 2. Here we ignored all the combinations resulted by $\bar{p}r$, which are not guaranteed the TS awareness for the BDA process.

Rule 2 (Composition Rules of TSP's of CWS): According to the Table 6.2.2, we can derive the acceptable composition patterns for the successful CWS composition of the BDA process based on ASC. Therefore, here we conclude the acceptable composition patterns of the CWS as follows; Acceptable Sequential patterns are $\bar{c} : \bar{c}$, $\bar{c} : \bar{c}r$, $\bar{c}r : \bar{c}r$ and $\bar{c}r : \bar{c}$ only. And, acceptable AND patterns $\bar{c} // \bar{c}$, $\bar{c} // \bar{c}r$, $\bar{c}r // \bar{c}r$ and $\bar{c}r // \bar{c}$. These are only patterns that guarantee the end-to-end TS awareness for the BDA process.

Customization Criteria's of the TS awareness to the BDA workflow: As it shown in Fig. 6.2.4, it has two Sequential patterns (CWS_1^S, CWS_2^S), one AND pattern (CWS_1^A) and one XOR pattern (CWS_1^X). Existing TS aware selection methods proposed TS-awareness apply to the complete workflow. However, in BDA perspective, users are consists with a diverse range of requirements due to the limitation of budget and time.

TABLE 6.2.2: Composite TSP consideration of WS to the BDA

TSP1	*TSP 2*	*Result of CWS^L*	*Result of CWS^A*	*Consideration to BDA on ASC*
pr	*pr*	$\bar{p}r$	$\bar{p}r$	*No*
pr	*c*	$\bar{p}r$	$\bar{p}r$	*No*
pr	*cr*	$\bar{p}r$	$\bar{p}r$	*No*
c	*pr*	$\bar{p}r$	$\bar{p}r$	*No*
c	*c*	$\bar{c}$	$\bar{c}$	*Yes*
c	*cr*	$\bar{c}$	$\bar{c}$	*Yes*
cr	*pr*	$\bar{p}r$	$\bar{p}r$	*No*
cr	*c*	$\bar{c}$	$\bar{c}$	*Yes*
cr	*cr*	$\bar{c}r$	$\bar{c}r$	*Yes*

And also, existing patterns in the workflow will not be highly critical, and some patterns already trusted by experiences. Then it is not necessarily to assured by pricier and time consuming compensatable services. Therefore, we define five custom levels of selection criteria's in Chapter 6.1.1.

C. CTS and QoS_{QN} based Genetic Algorithm

We elaborate the algorithm to find global optimal of the proposed method using

Algorithm 1 QoS_{QN} and Custom TS based Genetic Algorithm

Input: - Result Service and Tasks set of Discovery stage of ASC
- Selection Level 1/ 2/ 3/ 4/ 5

Output: - Optimal Plan, - fittest_Qos

BEGIN

```
1:  Initialize the Population
2:      for each Individual in Population do
3:          if check selection Level (1/ 2/ 3/ 4/ 5) then
4:              populate individuals based on the Level requirement
5:          end if
6:      end for
7:  Find the Best Individuals from the Initial population
8:  Evolve Population
9:      for ∀ Individuals ∈ Population do
10:         if elitism true then
11:             elitismOffset set to 0 and save individual
12:         end if
13:         for each Individuals start from elitism offset to crossover do
14:             create parent1 and parent2 from tournament selection
15:             Probabilistically crossover parent1, parent2 & create
                new child and add in to population
16:         end for
17:         for each Individuals start from elitism offset to mutate do
18:             Probabilistically cross over the each pairs
19:         end for
20:     return new population
21: end for
```

END

Algorithm 1.

In the selection stage of the ASC, it uses the result of discovery stage use as one input [4] and the custom level (L1, L2, L3, L4 or L5) of the selection criteria as the

next. Feed the input to the proposed the GA based Algorithm 1.

- Initialize Population: Line number 1 to 6, at the beginning it populates generation based on custom level requirements. Gene's encoding scheme during initialization is shown in Fig. 6.2.4. Here, $1 \leq i \leq n$; n is the number tasks in the workflow. j and k are jth and kth candidate services of respective tasks. It follows the composition rules definitions 1 and 2.
- Evolve Population: Next, line number 7 to 21; it evolves population using initial population. Mainly, it executes three operations, which are crossover, mutation, and selection. It continues changing the fittest disregard its current fittest. We set the minimum to recommend mutation rate find optimal minimal for the given requirement. Due to the small mutation rate, it allows to explores more search space and it has more probability to reach global optimal than local optimal.

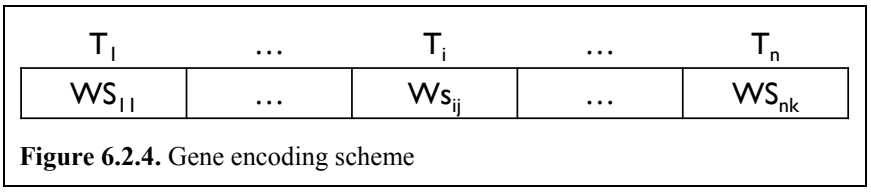


Figure 6.2.4. Gene encoding scheme

6.3 QoS-aware Rule-based Traffic-efficient Multi-objective Selection

6.3.1 Introduction

The expansion in services has led to increased opportunities. Studies show that the growth in revenue has more than doubled in bi-annually, with an increase more than 220% in service-based data-science-related activities in the Amazon Web Services Platform in 2015, 2016 and 2017 compared to the respective 2013, 2014 and 2015[7]. In addition, studies show a doubling of the volume of services in the ProgrammableWeb.com store each year. Moreover, ProgrammableWeb is becoming a

[7] www.statista.com/statistics/233725/development-of-amazon-web-services-revenue

popular platform for well-known providers such as IBM, Google, and Microsoft [40]. This confirms that growth in the consumer market and the availability of services is extensive.

There has been a dramatic surge in the availability of candidate services, which, in turn, has led to "resource starvation" in selection processes. The variety and uncertainty (veracity) of quality of services (QoS) among the high volume of candidate services increases the NP-hardness in the search for solutions [40].

Introducing new research areas such as data science (DS) also increases the complexity of the composition system. Lengthy processes such as data analysis (DA) for business intelligence in a DS field typically comprise four different phases: preparation, analysis, reflection, and dissemination [84]. In the preparation stage, data must be collected, formatted, and cleaned. In the analysis stage, the analysis, debugging, and inspection of substages must be cleared. In the reflection stage, there are comparisons and calls to re-execute the substages of the analysis stage. In the dissemination stage, steps such as displaying, deploying, and retrieving the statistics of the detailed process must be at least minimally performed. Each of these substages can contain multiple tasks and modern composition systems have lengthier processes than do their conventional counterparts. We recognize this as a challenge of modern service selection. In response, new DA research is emerging in DS, involving approaches such as Big Data analytics (BDA) [11], which is lengthier than general DA methods.

It is evident that the native service selection process is a well-known NP-hard problem [35], [36]. Moreover, conventional standalone processing platforms are reaching their limits in dealing with NP-hardness [40]. Thus, we move toward Big Data-related research topics. Hadoop and Spark are currently the most widely used Big Data processing platforms. However, the Spark platform is generally regarded as involving expensive in-memory processing and lacking its own file-management system. This hinders the study of native traffic congestion occurring during the selection process. Hadoop is the most popular native Big Data processing platform. It includes its own file-management system and facilitates a diverse range of techniques for file handling. Therefore, it allows studying the occurring-traffic during the selection in a more

effective manner. It also provides a gateway to addressing a diverse range of problems associated with research and industry domains in a cost-effective manner. MapReduce (MR) is a well-known fundamental programming technique in the Hadoop space. Therefore, we devised our selection method based on MR in the Hadoop space.

Service composition is a key component of the highly diverse DS [40], [85]–[87], including the fields of BDA and deep learning. While facilitating these highly diverse composition requirements, a peer user should have the freedom to select and switch between composition requirements for the given problem using the given algorithm without affecting the fundamental data structure, thereby avoiding complex steps during decision-making or changing preferences. However, there are no existing methods that facilitate such dynamically changing composition requirements and most have rigidly specific composition patterns, such as optimizations that are focused only on linear [88], [89] combinatorial [90], [91], [92], [93] or multivariate patterns [40], [94], [95], [96]. Therefore, it is very important to facilitate multiobjective selection methods, which are flexible with respect to different selection requirements without affecting the overall flow of the algorithm. Accordingly, our aim is to propose a multiobjective selection algorithm that facilitates user-driven flexibility about selection methods.

Given this aim, we propose three types of selection approaches by considering the three main types of composition requirements, the first being based on linear programming and the other two involving dynamic programming.

In some cases, users need to find the global optimal QoS requirements from a given set of candidate services. Here, to satisfy the linear programming requirements, we propose a Dijkstra algorithm-based method as a novel technique for selection domain.

In other cases, users need to satisfy a combinatorial optimization of their QoS requirements with respective to a maximum upper bound for the overall negatively affected values, while maximizing the overall positively affected QoS of the services. Here, we propose a 0-1 Multi-Constraint Knapsack Problem (0-1 MCKP) [90], [91], [97] to satisfy these requirements.

Finally, peer users may seek multivariate optimization of their QoS parameters for their composition system. To satisfy this requirement, we propose using selection

criteria based on the Artificial Bee Colony (ABC) algorithm. These three methods are integrated into the multiobjective selection method referred to in this paper.

Service-selection processes in Big Data spaces can involve excessive traffic congestion because of both internal and external factors in the MR process [54], [51]. Considering the MR algorithm as the base, shuffling traffic called the internal traffic occurs during the MR process, and ZipF, Pareto traffic occurs due to the hotness of the data and outside of the MR algorithm. Then they are called as the external traffic perspective to the MR algorithm. We refer to these two issues as "internal traffic" and "external traffic," respectively, throughout this paper.

Two commonly identified types of external traffic congestions are the Zipf and Pareto phenomena's [51]. They affect negatively to the overall performance of an MR job. Both phenomena's occur naturally in any environment, including local machines, local area networks, and clouds. However, they show particularly adverse behavior in the Hadoop space.

ZipF [98]–[100] and Pareto [99], [101], [102] are native and intrinsic traffic congestions occurring network data processing. However, these two factors show adverse behavior in the Big Data environment [51], [103].

The Zipf phenomenon in a Hadoop distributed file system (HDFS) refers to an access pattern distribution of replicas in a given file according to Zipf's law. This results in a "hot" replica (higher access rate) among replicas available in a given file. In the service-selection process, the aim is to deal with QoS service preferences during this process, but the hotness phenomena tend to favor services with lower QoS preferences. This leads to a reduction in the accuracy of the selection process. In addition, because of the limited number of replica preferences, it generates heavy traffic congestion and causes a reduction in overall performance. To address these concerns, we propose a QoS-aware service distribution method.

The Pareto phenomenon in general environments refers to 80% of the data usage being represented by 20% of the data. This is known as the "80/20 rule" [103]. However, the Hadoop process demonstrates the further adverse effect in the Hadoop environment. This phenomenon causes to increase the hot files. To address this traffic congestion

phenomenon, we propose a traffic-aware service-replica distribution method.

In addition to that, service selection is a natively NP-hard problem for finding the optimal utility (optimized for linear, combinatorial or multivariate) values of QoS in composition plan among services available. To address these issues, we employed the heuristic methods to find the optimal selection composition plan. Then it is obvious, the given selection process always tries to achieve the global/local optimal utility QoS plan among the given services. Then service selection also generates the 'hot' services and composition plans during the selection process. Therefore, we assume, it has more tendency to affect the intrinsic traffic congestions which are caused by ZipF and Pareto phenomena's than these two traffic congestions appear in the native traffic in the Hadoop environment.

Moreover, one of the most common reasons for internal traffic congestion is data communication between the map and reduce phases [104]. Usually, this shuffles a large chunk of the map results in the reduce phase based on the key value of the MR job, and thereby on the internal traffic of the MR job. Service selection generates an excessive amount of intermediate data and this tends to increase the internal traffic. Therefore, we propose a combiner-based intermediate MR agent to address this issue. We called this method the intermediate MR agent. This agent procedure results in two major benefits, firstly, this reduces traffic congestion between mapper and reducer by reducing the intermediate data, and secondly reduces the rest job of reducer phase. This results in curtailing the processing time of the reducer phase. These two are the main reasons behind the reflexing the traffic in the shuffling stage and workload of the reducer phase.

Experimental results show that our solution is well adapted to Big Data spaces. A shortened form of this paper has been presented previously in conference papers [89], [103]. Here, we expand on the theoretical and evaluation aspects of the optimization model and propose a multiobjective service-selection algorithm for Big Data spaces. We are among the first to propose bidirectional (internal and external) traffic optimization based on QoS awareness for multiobjective service-selection algorithms in a distributed environment. Our main contributions are:

The Multiobjective service-selection algorithm in a distributed environment.

QoS-aware rule-based traffic solutions to optimize traffic caused by the ZipF and the Pareto.

Combiner based traffic solution on optimizing the traffic in shuffling stage.

The remainder of the paper is structured as follows. In Section II, we introduce some preliminaries and formulate the problem statement. In Section III, we present our proposed solutions for multiobjective selection requirements and bidirectional traffic concerns. Section IV discusses the proposed traffic-efficient multiobjective selection algorithm. In Section V, we discuss evaluation, and Section VI considers related work. Section VII concludes the paper.

6.3.2 Preliminaries and Problem Statement

In Section A, we introduce some preliminaries to selection-requirement issues. In Section B, we elaborate on the selection data flow in the MR process. Section C formulates the selection traffic problem for Big Data space.

6.3.2.1 Preliminaries

In this subsection, we describe preliminary studies in the problem domain. First, we define the types of selection requirements and traffic congestion. Next, we define a problem statement for selection in the Big Data domain and model the selection problem under the external and internal traffic concerns.

We define service-selection requirements that are identified as three of the most common types. We use these three requirements to show the adaptability of the proposed method. One of these three requirements will be used in the relevant selection scenario during the MR process in the Big Data space.

- QoS values for web services (WS) that are affected positively by the performance of the service are denoted by QoS^P, and there will be x QoS constraints for the given WS WS_{ij}. Here, $x,\ i,\ j > 0$, i represents the i^{th} task in a composition planner, and the j represents the j^{th} candidate service for the given i^{th} task.
- QoS values of WS that are affected negatively by the performance of the service are denoted by QoS^N, and there will be y QoS constraints for the given WS WS_{ij}. Here, $y,\ i,\ j > 0$ and $x + y$ represents the total number of QoS values for the given WS_{ij}.

- The weight values of QoS^P and QoS^N are represented by $w_\propto$ and w_β, respectively, such that $\sum_{\propto=1}^{x} w_\propto + \sum_{\beta=1}^{y} w_\beta = 1$ and $0 < w_\propto . w_\beta < 1$.
- T is the number of tasks in the workflow, such that $0 < i \leq T$. $v_\propto^{avg}$ is the average value of the particular QoS attribute among the given candidate services and $v_\propto^{max}$ and $v_\propto^{min}$ are the maximum and minimum values of the respective attributes.
- $Q_{All}^{Constraints}$ is the collective value of negatively affected QoS attributes set by the user, with $Q_\beta^{Constraints}$ being an individual constraint.

Based on these notations, we defined the heterogeneous-selection problem using Definitions (and Scenarios) 1, 2, and 3 in Chapter 6.1.2.

Definition 4: External traffic congestion—The Zipf Problem: This refers to the fact that the relative probability of access or processing requests for the ith most popular data block is proportional to $1/1^\theta$. When $\theta = 1$, the data block-request distribution strictly follows Zipf's law. Otherwise, it follows a more general Zipf-like distribution. Generally, it follows a Zipf-like distribution for Hadoop in the corresponding data blocks [105], [106].

Chen et al. [51] described the Zipf distribution of an MR job for both the input and output files of the MR job. They experimentally demonstrated and discussed the resulting unusual phenomena in terms of Zipf's law. Zipf generates hot files (higher access rate) and cold files (lower access rate) during the MR job. This causes an increase in the internal traffic and considerably reduces the performance of the overall MR job. Henceforth, we will use the term hot for more popular files and the term cold for not so popular files. We propose a traffic solution to the Zipf problem in Definition 7 of Section III-D-1.

Definition 5: External traffic congestion—The Pareto Problem: The Pareto problem refers to the 80/20 rule. From a Hadoop perspective, the Pareto problem says that 80% of access or processing happens in 20% of the most popular data blocks. However, in actual cases, this 80/20 rule varies somewhat, becoming an "80/10" rule. It can then severely affect internal data communication and traffic in the MR job, increasing the ratio of hot vs cold files. Zipf and Pareto problems severely imbalance the data

population in HDFS. This leads to increased traffic to hot data and dramatically affects the overall traffic of the MR process. We propose a traffic solution to the Pareto problem in Definition 8 of Section III-D-1.

Definition 6: Internal traffic congestion - The Shuffling Problem: The mapper generates large chunks of intermediate data that are passed on to the reducer for further processing, which leads to massive network congestion. Shuffling data account for 58.6% of the cross-pod traffic and amounts to over 200 petabytes of data in an analysis of SCOPE jobs [54]. For shuffle-heavy MR tasks, this high traffic can incur a considerable performance overhead of up to 30–40%, as described in [107]. Therefore, we propose a traffic solution to internal traffic congestion in Section III-D-2.

6.3.2.2 Service Selection Data Flow in the MR Process

Fig. 6.3.1 shows the typical anatomy of an MR process. According to the figure, part of the selection operation is scheduled in the given MR job. The input split constrains information about the candidate services for the first tasks in the given composition requirement, combined with mapper, then continues a portion of the overall selection operation. Respective files are then split, merged, copied, and sorted before being fed to the reducer phase. A different set of intermediate keys are assigned to each respective reducer node. These sets are the input to the reduce tasks. Reduce tasks are responsible for reducing the value associated with intermediate keys. Therefore, a set of intermediate keys are sorted on a single node before being fed to the reducer.

The respective outputs of the reducers represent an optimal composition planner for the given selection criteria. This executes the respective key-related results from the reducers and outputs the respective optimal planner associated with each key (servicing the first tasks in the composition requirement).

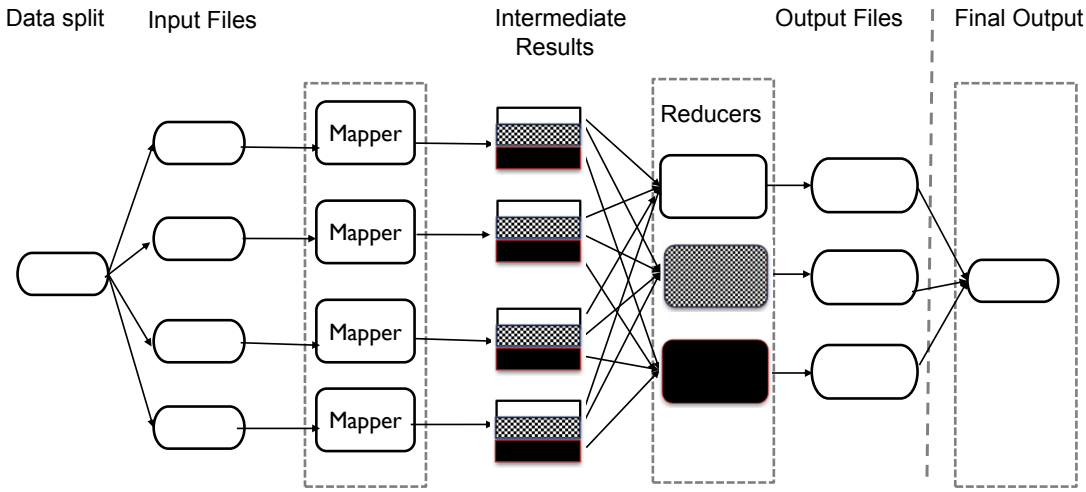


Figure 6.3.1. Anatomy of a typical MR process

6.3.2.3 Traffic Problem Statement

According to the network traffic and the data flow of the selection process in the MR job, we can state the problem as follows, in terms of an effective-selection job. Fig. 6.3.2 shows the anatomy of an MR job. It has to pass rigorous steps in the data flow while facing severe constraints, namely, internal and external traffic. C_{MR_i} is the cost of the ith job of the MR process for the selection. C_{Joint_i} is the traffic cost that affects the ith job of the MR process. In addition, $C_{Sel_i}$is the cost of the selection process during the ith job of the MR process. We can then describe the cost of the ith job of the MR process as shown in Eq. 5.

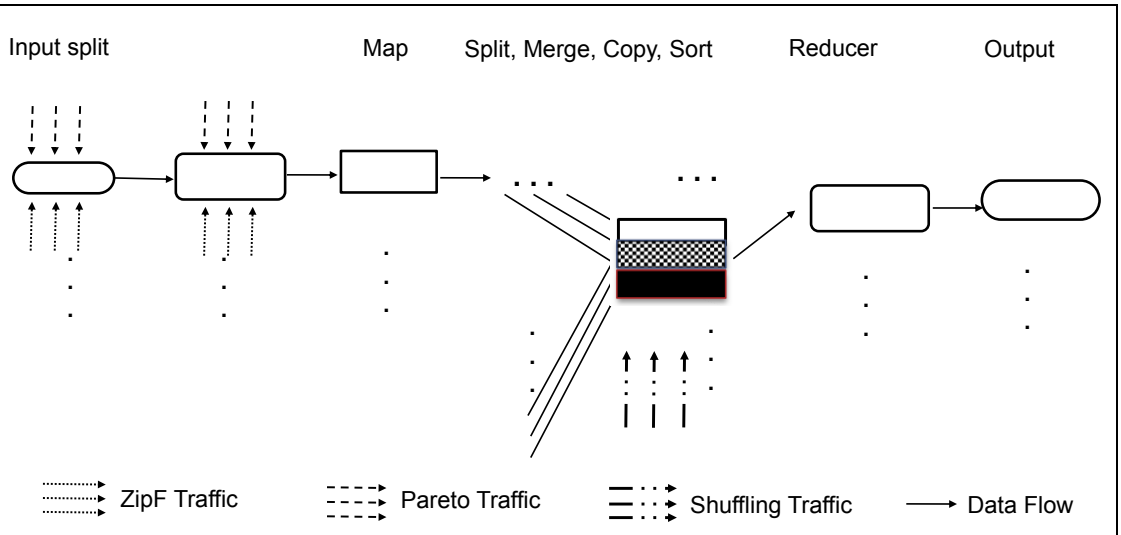


Figure 6.3.2. MR job vs traffic observed during the process

$$C_{MR_i} = C_{Joint_i} + C_{Sel_i} + a_1 \quad (5)$$

Here, $C_{Sel_i} = C_{Map_i} + C_{Red_i}$ involves the map and reduce phases during selection. $C_{Joint_i}$is mainly affected via internal and external traffic during the process. Therefore, we can give Eq. 6 as the traffic cost of the ith job of the process.

$$C_{Joint_i} = C_{Int_i} + C_{Ext_i} + a_2 \quad (6)$$

Here, C_{Int_i} is the internal traffic cost of the ith job of the process and C_{Ext_i} is the external traffic cost of the ith job of the process. These external and internal costs can then be further divided and expressed by Eq. 7 and Eq. 8 as follows.

$$C_{Int_i} = C_{S_i} + a_3 \quad (7)$$

$$C_{Ext_i} = C_{Z_i} + C_{P_i} + a_4 \quad (8)$$

C_{S_i}, C_{Z_i} and C_{P_i} are the costs of the shuffling, Zipf, and Pareto, respectively, of the ith job of the MR process, respectively, and $a_i: i = 1,2,3,4,5$ is constant. We can then represent C_{Joint_i} as shown in Eq. 9. Here, c is a constant.

$$C_{Joint_i} = C_{S_i} + C_{Z_i} + C_{P_i} + a_5 \quad (9)$$

Therefore, to have an effective and efficient selection process, we have to address the concerns described in Eq. 9. This means that the respective C_{Map_i}, C_{S_i}, C_{Red_i}, C_{Z_i} and C_{P_i} values need to be reduced to reduce the respective C_{Joint_i}. This directly affects C_{MR_i} , as shown in Eq. 5.

We can then deduce the Eq. 10 to show the effects of traffic congestion in reducing the overall traffic. This implies that we can reduce the overall execution cost of the MR job of the selection process when we minimize the collective internal and external traffic congestions, as shown below. Here γ is the number of jobs in the MR process.

$$\min_{i \in \gamma} \left(\sum \left(C_{S_i} + C_{Z_i} + C_{P_i} \right) \right) \quad (10)$$

$$leads\ to\ \min_{i \in \gamma} \sum C_{MR_i}$$

We prove this as a joint optimization of the traffic problems in Section III-D-3.

6.3.3 Proposed Solution

In this section, we present our proposed solutions to the issues described in Section II. In Sections III-A, III-B, and III-C, we present the proposed solutions for multiobjective selection requirements. In Section III-D, we develop the proposed solution for bidirectional traffic concerns.

6.3.3.1 Proposed Solution for a Linear Optimal Service Selection Requirement

We propose using a graph-theory-based linear programming technique to address the scenario described in Definition 1 in Section II. This is a novel approach to the service-selection domain. First, we prepare a directed acrylic graph (DAG) network from the candidate services, as shown in Fig. 6.3.3, and then use our proposed algorithm to calculate the longest path between the first and last vertex, as described by Zeng et al. [88]. Dijkstra employed a longest-path-finding algorithm to calculate the optimal selection composition plan for such a DAG network. Table 6.3.1 shows the respective QoS utility representations for the Dijkstra method.

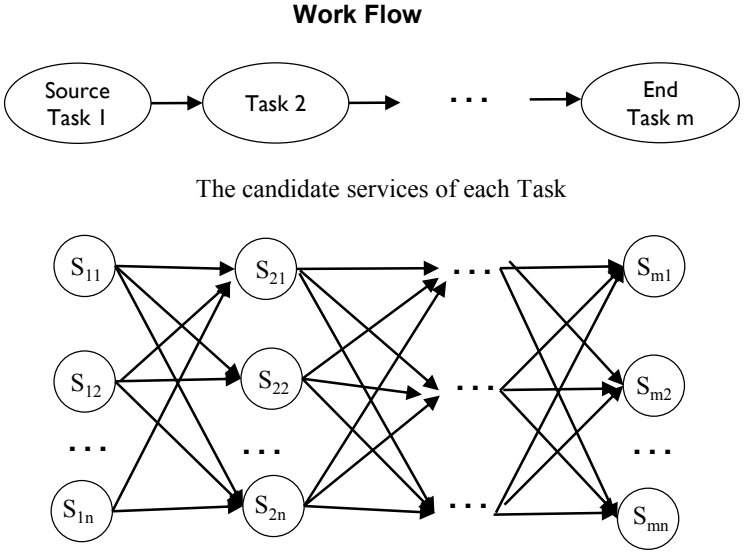


Figure 6.3.3. DAG network created by candidate services of each task

Plans are made by the given candidate services. We use Eq. 11 to calculate the

distance of vertices (services) between the ith and the (i+1)th tasks containing any two services, which is called the L(Si, Si+1). According to the Fig. 6.3.3 shown DAG graph, it does not allow to create edges in between services in the same task. Fig. 6.3.3 DAG allows edges between adjacent tasks only, where c is a constant. For service S_{i+1} , its utility value $U_{S_{i+1}}$ is given by Eq. 12.

$$L\,(S_i, S_{i+1}) = c + U_{S_{i+1}} \tag{11}$$

$$U_{S_{i+1}} = \sum_{\propto=1}^{x} \left(\frac{V(ws_{i\propto}) - v_{\propto}^{avg}}{v_{\propto}^{max} - v_{\propto}^{min}} \right) w_{\propto} + \sum_{\beta=1}^{y} \left(1 - \frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}} \right) w_{\beta} \tag{12}$$

TABLE 6.3.1. Tasks vs utility values of candidate services used in Dijkstra
Here, n is the number of candidate WSs and m is the number of tasks in the work flow.

Source Task	Task 2	…	End Task
S_{11} U.QoS: $U_{S_{11}}$	S_{21} U.QoS: $U_{S_{21}}$	…	S_{m1} U.QoS: $U_{S_{m1}}$
S_{12} U.QoS: $U_{S_{12}}$	S_{22} U.QoS: $U_{S_{22}}$	…	S_{m2} U.QoS: $U_{S_{m2}}$
…	…	…	…
S_{1n} U.QoS: $U_{S_{1n}}$	S_{2n} U.QoS: $U_{S_{2n}}$	…	S_{mn} U.QoS: $U_{S_{mn}}$

Here, we have taken c = 0. A DAG network is defined as a source task (Task 1) to the last task, which Task m services. This graph connects exactly m services. Each service represents a vertex and the distance between two services is calculated from Eq. 11. It does not create edges between services in the same Task itself. We define the graph G (V, E), in which V is created from Si and E is calculated by L(Si, Si+1). Source nodes are represented by Task-1 services and target nodes are represented by end-task services. If the cost of the (ij)th service is represented as Eij, we have to maximize

$\sum_{ij\in E} E_{ij}.S_{ij}$, subject to $S \geq 0$, for all i.

6.3.3.2 Proposed Solution for a Combinatorial Service-Selection Requirement

We propose a dynamic programming technique to address the combinatorial selection requirement. We use 0-1 MCKP to simulate the scenario described in Definition 2 in Section II. The utility-value calculations that represent the profit and weight used in 0-1 MCKP are given by Eq. 13 and Eq. 14. Table 6.3.2 shows the respective weights and profit consumption in the selection scenario.

Table 6.3.2 shows the number m of Tj (i ≤ m) tasks in the workflow. Each Tj contains

$$U_{i,j:P} = \sum_{i=1}^{T}(\sum_{\propto=1}^{x}\left(\frac{V(ws_{i\propto}) - v_{\propto}^{avg}}{v_{\propto}^{max} - v_{\propto}^{min}}\right) w_{\propto} + \sum_{\beta=1}^{y}\left(1 - \frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}}\right) w_{\beta}) \quad (13)$$

$$U_{i,j:,N} = \sum_{i=1}^{T}(\sum_{\beta=1}^{y}\left(\frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}}\right) w_{\beta}) \quad (14)$$

TABLE 6.3.2. Task vs utility values of candidate services used in 0-1 MCKP
Here, n is the number of candidate WSs and m is the number of tasks in the work flow.

Source Task	Task 2	…	End Task
S_{11} Profit:$U_{1,1:P}$ Weight: $U_{1,1:N}$	S_{21} Profit:$U_{2,1:P}$ Weight: $U_{2,1:N}$	…	S_{m1} Profit:$U_{m,1:P}$ Weight: $U_{m,1:N}$
S_{12} Profit:$U_{1,2:P}$ Weight: $U_{1,2:N}$	S_{22} Profit:$U_{2,2:P}$ Weight: $U_{2,2:N}$		S_{m2} Profit:$U_{m,2:P}$ Weight: $U_{m,2:N}$
…	…	…	…
S_{1n} Profit:$U_{1,}$	S_{2n} Profit:$U_{2,}$		S_{mn} Profit:$U_{m,n:P}$

n candidate services Sij (i ≤ n, j ≤ m). The profit that can be gained from Sij services is represented as $U_{ij}^{P\aleph}$ and the weight as $U_{ij}^{N\aleph}$. The custom QoS requirement is called the capacity, denoted as C.

Our 0-1 MCKP requirement is then expressed as follows. To maximize the normalized utility QoS P_u , $P_u = \sum_{i=1}^{m}\sum_{j=1}^{n} U_{i,j:P}$, subject to $\sum_{j=1}^{n} U_{i,j:N} . S_{ij} \leq C$, where $i \in M\{1,2,\dots,m\}$. Moreover, $\sum_{i=1}^{m} S_{ij} = 1$, $j \in N\{1,2,\dots,n\}$, and $S_{ij} = 0\ or\ 1$, where $i \in M, j \in N$. $U_{i,j:P}$ positively affects the utility QoS of the jth candidate WS of the ith task. $U_{i,j:,N}$ negatively affects the utility QoS of the jth candidate WS of the ith task.

6.3.3.3 Proposed Solution for a Multivariate Service-Selection Requirement

The WS selection requirement defined by Definition 3 in Section III is simulated using the ABC algorithm [108]. The utility functions for the positively affected QoS (profit) and negatively affected attributes are represented by Eq. 15 and Eq. 16, respectively. Table 6.3.3 gives the respective value distributions across the candidate services for the given task. The utility QoS of the jth candidate WS of the ith task is positively affected by $U_{i,j:P}$, whereas $U_{i,j,\beta:,N}$ is the βth negatively affected utility QoS of the jth candidate WS of the ith task. The user sets β constraints for the β QoS attributes for candidate WS of the jth task

Initially, the ABC algorithm initializes the generated "food source" randomly. Here, it uses Eq. 15 and Eq. 16 to set the respective utility values for the food sources.

Next, it sends the employed "bees" to the food sources (identified plans) and determines the amount of "nectar" (overall profit). It then calculates the fitness values for each food source. Greedy selection is applied to the current solution and its mutant. If the mutant solution is an improvement, it replaces the previous solution and the trial counter of the solution is reset. If a better solution cannot be found, the trial counter is incremented.

The algorithm then calculates the probability of the fitness value. Here, it evaluates

the nectar information from all employed bees and chooses a food source (identified solution) with a probability related to the amount of nectar (overall profit). If it cannot find a better solution, then it enjoins a "scout bee" to find a new food source in the same way as does the employee bee. It continues this process until the termination condition is reached.

$$U_{i,j:P} = \sum_{\propto=1}^{x} \left(\frac{V(ws_{i\propto}) - v_{\propto}^{avg}}{v_{\propto}^{max} - v_{\propto}^{min}} \right) w_{\propto} + \sum_{\beta=1}^{y} \left(1 - \frac{V(ws_{i\beta}) - v_{\beta}^{avg}}{v_{\beta}^{max} - v_{\beta}^{min}} \right) w_{\beta} \quad (15)$$

$$U_{i,j,k:N} = \left(\frac{V(ws_{ik}) - v_{k}^{avg}}{v_{k}^{max} - v_{k}^{min}} \right) w_{k} \qquad \text{Here } 1 \leq k \leq \beta \quad (16)$$

TABLE 6.3.3. Task vs utility values of candidate services used in ABC
Here, n is the number of candidate WSs and m is the number of tasks in the work flow.

Source Task	**Task 2**	**..**	**End Task**
S_{11}	S_{21}	..	S_{m1}
Profit:$U_{1,1:P}$	Profit:$U_{2,1:P}$		Profit:$U_{m,1:P}$
Neg. 1st : $U_{1,1,1:N}$	Neg. 1st : $U_{2,1,1:N}$		Neg. 1st : $U_{m,1,1:N}$
Neg. 2nd : $U_{1,1,2:N}$	Neg. 2nd : $U_{2,1,2:N}$		Neg. 2nd : $U_{m,1,2:N}$
..	..		..
Neg. ,β^{th}: $U_{1,1,}\beta_{:N}$	Neg. ,β^{th}: $U_{2,1,}\beta_{:N}$		Neg. ,β^{th}: $U_{m,1,}\beta_{:N}$
S_{12}	S_{22}	..	S_{m2}

6.3.3.4 Proposed Solutions for Bidirectional Traffic Concerns

In this subsection, we present the proposed solutions for the bidirectional traffic concerns. We first describe a solution for external traffic and then for internal traffic. Finally, in Section III-D-3, we present a joint optimization of bidirectional traffic concerns.

1. EXTERNAL TRAFFIC solution

We now describe our proposed solutions for the two external traffic concerns, namely

the Zipf and Pareto problems.

Zipf problem: QoS-aware service distribution (QSD rule)

We propose an efficient, rule-based traffic technique to address the concerns that occur during the selection process in an MR job. From our observations, Zipf raised the most concerns, especially in the split and map stages of the MR process shown in Fig. 6.3.2. These areas are highly correlated with specific files and their replica access for specific needs in the selection process, incurring Zipf phenomena and hot files. To address this concern, we propose the QSD rule.

Definition 7: QSD Rule. Service distribution according to the proportional values of the normalized utility QoS are inversely proportional to the hotness caused by the Zipf phenomena during the selection process.

Let f be the functional representation of the proportional distribution of services and φ the functional representation of the hotness that occurs from Zipf during service selection. The rules are formulated in Eq. 17 and Eq. 18.

$$f^*(S_i, S_{i+1}) = \frac{QoS\ (S_{i+1})}{QoS(S_i)} \tag{17}$$

$$f^*(S_i, S_{i+1}) = \frac{\varphi(S_{i+1})}{\varphi(S_i)} \tag{18}$$

Here, $f^*(S_i, S_{i+1})$ is the optimal proportional replica distribution of S_i and S_{i+1} services according to their normalized QoS proportions, as shown in Eq. 17, and $\frac{\varphi(S_{i+1})}{\varphi(S_i)}$ is the proportional value of the hotness caused by the Zipf of S_i and S_{i+1} services, as shown in Eq. 18.

We prove the QSD rule in terms of Theorem 1, Lemma 1, and Lemma 2, as below.

Theorem 1: Traffic caused by Zipf is proportional to the normalized QoS criteria of the services.

Proof: Consider that there are services called $S_1, S_2, \dots, S_n$, and their respective normalized QoS are $QoS_{S_1}, QoS_{S_2}, \dots, QoS_{S_n}$. The functional representation of the

QoS distribution is given as $\emptyset\left(U_{S_1}, U_{S_2}, \ldots, U_{S_n}\right)$. Traffic occurring by Zipf for respective $S_1, S_2, \ldots, S_n$ services is denoted as $T_{S_1}^Z, T_2^Z, \ldots, T_n^Z$, and the functional representation of the hotness distribution is given as $\mathrm{T}(S_1, S_2, \ldots, S_n)$. Zipf occurs at the splitting and map stages. At the split stage, it is trying to find an accessible replica of a given service from among the available replicas.

Here, it is most actively and exhaustively used for the highest normalized QoS replicas. This means that the highest traffic occurs for the highest QoS replicas, and the lowest traffic for the lowest QoS replicas received.

In turn, this implies that the splitting stage traffic for the ith job of the MR, $T_{S_{j,split}}^{MR_i,Z}$, is directly proportional to its U_{S_j}, where k_1 and c1 are constants.

Therefore, $$T_{S_{j,split}}^{MR_i,Z} = k_1 * U_{S_j} + c_1 \,, \qquad (19)$$

In addition, the mapper part is working hard to achieve the optimal composition plan among the given list of plans, as described in Sections III-A, III-B, and III-C. Here, exponential traffic occurs to the highest QoS because all three native techniques are designed to follow the optimal QoS. Therefore, the mapper stage traffic for the ith job of the MR, $T_{S_{j,Map}}^{MR_i,Z}$, is directly proportional to its U_{S_j}.That is,

$$T_{S_{j,Map}}^{MR_i,Z} = k_2\, U_{S_j} + c_2 \,, \qquad (20)$$

Where k_2 and c2 are constants. We now consider both Eq. 18 and Eq. 19:

$$T_{S_{j,Map}}^{MR_i,Z} + T_{S_{j,split}}^{MR_i,Z} = k_3\, U_{S_j} + c_3$$

Where $k_3 = k_1 + k_2$ and $c_3 = c_1 + c_2$ are constants.

This means that traffic caused by jth services for the ith job is given by Eq. 21:

$$T_{S_j}^{MR_i,Z} = k_3\, U_{S_j} + c_3 \qquad (21)$$

Moreover, based on Eq. 21, traffic caused by all services of the ith job can be expressed as shown in Eq. 22. Here, N is the number of services and γ is the number

$$\sum^{N} T_{S_j}^{MR_i,Z} = k_3 \sum^{N} (U_{S_j} + c_4) \quad (22)$$

of jobs in the MR process.

Finally, based on (22), traffic caused by all services of all jobs contained in the MR process is given by Eq. 23:

According to Eq. 23, therefore, the overall traffic caused by Zipf is proportional to the

$$\sum_{i=1}^{\gamma}\sum_{i=1}^{N} T_{S_j}^{MR_i,Z} = k_3 \sum_{i=1}^{\gamma}\sum_{i=1}^{N}(U_{S_j} + c_5) \quad (23)$$

normalized utility QoS of services.

According to Eq. 21, assuming services S_j and S_{j+1} and their $U_{S_j} > U_{S_j+1}$, the traffic imposed by Zipf for the ith MR job is shown by Eq. 24:

$$T_{S_j}^{MR_i,Z} > T_{S_j+1}^{MR_i,Z} \quad \{24\}$$

According to Eq. 24, traffic to S_j and S_{j+1} caused by Zipf for the entire MR process is:

$$\sum_{i=1}^{\gamma} T_{S_j}^{MR_i,Z} > \sum_{i=1}^{\gamma} T_{S_{j+1}}^{MR_i,Z} \quad (25)$$

Services that have higher normalized QoS have more traffic during the MR selection process. Then, from Eq. 23 and Eq. 25, traffic increments of S_j and S_{j+1} are proportional to their normalized QoS. This is represented by Eq. 26.

$$\frac{T_{S_{j+1}}^{MR_i,Z}}{T_{S_j}^{MR_i,Z}} = \frac{U_{S_{j+1}}}{U_{S_j}} \quad (26)$$

Lemma 1: Traffic caused by Zipf is inversely proportional to the service distribution in the normalized QoS.

Proof: According to the Eq. 26, traffic in the given service increases with increasing QoS. It then seeks more file instances of that service to overcome the demand. Let $f(S_i) = f(S_{i+1})$ be the initial service distribution of S_i and S_{i+1} services, with their QoS being $U_{S_i} > U_{S_i+1}$. Then, according to Eq. 24, Zipf traffic for these services is

$T^1_{S_i} > T^1_{S_i+1}$. This implies that S_i suffers more demand than S_{i+1}. These demands are

$$T^2_{S_i} / T^2_{S_i+1} = T^1_{S_i} / T^1_{S_i+1} = U_{S_{i+1}} / U_i \quad (27)$$

proportional to their utility QoS because Zipf traffic is proportional to their QoS. The new file distribution is then f^*, with $f^*(S_i) / f^*(S_{i+1}) = U_{S_i} / U_{S_i+1}$. New traffic on S_i and S_{i+1} is redeemed in the same proportional manner: $T^2_{S_i} < T^1_{S_i}$, $T^2_{S_i+1} < T^1_{S_i+1}$ and $T^2_{S_i} / T^2_{S_i+1} = T^1_{S_i} / T^1_{S_i+1} = U_{S_{i+1}} / U_{S_i}$. This means that Zipf traffic is inversely proportional to the f^*. We can express this as Eq. 27:

Lemma 2: *Traffic occurring with Zipf is proportional to the hotness caused by Zipf for a given service.*

Proof: According to Definition 4, Zipf hotness refers to the popularity generated by Zipf. Nevertheless, "very hot" implies more traffic, "average hot" implies average traffic, and "not hot" implies no traffic. This means that

the hotness (caused by Zipf) is equally proportional to the traffic (caused by Zipf) on that service. $H^1_{S_j}$ is the hotness caused by the Zipf.

Then, according to Eq. 27 and Eq. 28 below, it proves Definition 7 and can be expressed as Eq. 29.

$$T^1_{S_j} / T^1_{S_j+1} = H^1_{S_j} / H^1_{S_j+1} \quad (28)$$

$$H^1_{S_j} / H^1_{S_j+1} = U_{S_{j+1}} / U_{S_j} \quad (29)$$

□

Pareto problem: traffic-aware replica distribution (TRD rule)

The Pareto raised most of its concerns in the split and map stages of the MR process. These areas are highly correlated with specific replica accesses for specific needs when Pareto phenomena and hot replicas occur. To address this concern, we propose the TRD rule.

Definition 8: TRD Rule. Hotness caused by the Pareto is inversely proportional to the traffic-aware replica distribution.

We prove the TRD rule based on Theorem 1 above and Lemmas 3 and 4 below.

Lemma 3: Availability vs traffic: increasing the availability of densely hot services (caused by the Pareto replicas) is inversely proportional to the overall traffic.

Proof: We can apply Theorem 1 to the Pareto traffic in services, which is proportional to the normalized QoS criteria of the services. However, replicas in the HDFS represent particular services. Therefore, we can extend Theorem 1 to the replica level of the S_j service, assuming S_j is replicated by r_1, r_2,.., r_n, with n being the default replication factor in the HDFS. According to the Pareto rule, it makes hot replicas from available replicas among the given services. In addition, according to Theorem 1, we can address this traffic by increasing the number of instances of a particular file. This implies that an increment of hot replicas according to their popularity (hotness) is inversely proportional to the traffic.

Lemma 4: Traffic occurring with Pareto is proportional to the hotness caused by Pareto for the given replicas.

Definition 8: TRD Rule. *Hotness caused by the Pareto is inversely proportional to the traffic-aware replica distribution.*

We prove the TRD rule based on Theorem 1 above and Lemmas 3 and 4 below.

Lemma 3: *Availability vs traffic: increasing the availability of densely hot services (caused by the Pareto replicas) is inversely proportional to the overall traffic.*

Proof: We can apply Theorem 1 to the Pareto traffic in services, which is proportional to the normalized QoS criteria of the services. However, replicas in the HDFS represent particular services. Therefore, we can extend Theorem 1 to the replica level of the S_j service, assuming S_j is replicated by r_1, r_2,.., r_n, with *n* being the default replication factor in the HDFS. According to the Pareto rule, it makes hot replicas from available replicas among the given services. In addition, according to Theorem 1, we can address this traffic by increasing the number of instances of a particular file. This implies that an increment of hot replicas according to their popularity (hotness) is inversely proportional to the traffic.

Lemma 4: *Traffic occurring with Pareto is proportional to the hotness caused by Pareto for the given replicas.*

Proof: According to Lemma 2, the hotness caused by a particular replica will be directly proportional to the traffic. This can be applied to Pareto, where the hotness caused by the Pareto is directly proportional to the traffic.

According to Lemmas 3 and 4, we can conclude that the hotness caused by Pareto is inversely proportional to the traffic-aware replica distribution. □

2. INTERNAL TRAFFIC solution

According to our observations, the shuffling stage of the selection process faces heavy traffic congestion because of a large number of intermediate results. To address this concern, we propose a middle agent for the MR process that uses a combiner. This job-wise combiner method allows us to sort and short-list the shuffling results of each job for both the mapping and reducing sides. The result is a considerably reduced overall job in the reducer phase. We call this the intermediate MR (IMR) agent.

IMR Agent: The mapper phase generates large chunks of intermediate data that are passed on to the reducer phase for further processing. At the beginning of the reducer phase, large chunks of intermediate data are shuffled to facilitate reducer processing. This leads to massive network congestion. To address this, we propose to introduce an IMR agent that reduces network congestion and relieves the workload of the reducer phase. The IMR agent mainly handles the task of shuffling the intermediate chunks, which is separated from the reducer phase and plays a crucial role in reducing network congestion. It is important to note that the primary job of the IMR Agent is to process the output data from the mapper before being passed to the reducer phase to reduce the workload of the reducer phase and minimized the shuffling data chunk. Therefore, IMR results in two major benefits, firstly, it accomplishes the part of the workload of the reducer phase, and secondly, it reduces the shuffling data traffic. We have employed a combiner to implement the IMR Agent class.

3. Joint Optimization for the Traffic Problem Statement

Section II-C contains the problem statement for overall traffic congestion. According to Eq. 9, we have to minimize the traffic concerns occurring during the selection stage of the MR process. These are expressed as $\sum C_{MR_i}$, where i is the ith job of the MR

process. In terms of the proposed techniques explained earlier in Section III-D, we minimize the overall traffic concerns as follows.

According to the Definition 7, by distributing services according to the QRD rule, we will obtain:

$$\min_{i \in \gamma} \sum C_{Z_i}$$

$$\min_{i \in \gamma} \sum C_{P_i}$$

Then, according to the Definition 8, by distributing the service replicas according to the TRD rule, we will obtain:

Finally, by considering the internal traffic congestion by using the IMR agent, we will obtain:

By aggregating these three techniques for overall external and internal traffic, we minimize the overall traffic caused by the selection process. That is, we obtain:

$$\min_{i \in \gamma} \sum C_{S_i}$$

$$\min_{i \in \gamma} \left(\sum \left(C_{S_i} + C_{Z_i} + C_{P_i} \right) \right)$$

This represents a solution to the problem expressed by Eq. 10 in Section II-C.

6.3.4 Qos-Aware Traffic-Efficient Algorithm

This section describes the proposed algorithm for service selection in Big Data space. Finding the optimal service composition sequence is a cumbersome task, mainly because of the complex composition requirements sought within a large search space. It requires high-performance infrastructure to complete such a resource-intensive heavy-duty job. We propose an MR algorithm that aims to meet heterogeneous-selection requirements, achieve global optima for linear and combinatorial selection requirements, and near-optimal multivariate-selection requirements.

According to the definition 1, 2 and 3, the proposed solutions for the multiobjective selection requirements are described in previous section III-A, B, and C. The selection

solutions are running behind the distributed environment. In addition to that, traffic solutions are applied. Section A describes the initial setup and overall flow, which is called as driver procedure; Section B describes the algorithm of the mapper; Section C describes the algorithm of the IMR agent; Section D describes the algorithm of the reducer and retrieve the final output.

6.3.4.1 SELECTION PROCEDURE: Flow of the Driver

Selection procedure represents the driver of the MR process. Let's define the scenario for the better understanding the process.

Scenario 4: As shown in Fig. 6.3.3; Consider a case comprising four tasks (m = 4) and n candidate services for each task. Assume p batches are processed by each mapper. Let U_i be the normalized utility QoS. Next, assume that we have services S1, S2, and S3, with respective U_i values: U1 = 0.5, U2 = 0.75, and U3 = 0.25. The proportional values of these three utilities will then be U1:U2:U3 = 2:3:1.

Initial Setup: First, we prepared the availability of the service as follows. The HDFS needs to contain the respective service information according to Definitions 7 and 8 in Section III-D. According to the QRD rule, we follow the service distribution of each batch of services as follows. S1 has a C × 2 replications, S2 has C × 3 services, S3, and S4 have C × 1 replications. Here, C is a constant and represents the replication factor for the TRD rule. This is the preparation of the environment to address the external traffic.

Next, it decides the types of QoS-awareness (linear optimal, combinatorial and multivariate) need for the composition. If he needs the linear optimal then he should use the Dijkstra method and doesn't have to do anything, just need to start the selection process. If he needs combinatorial, then he should use the 0-1 MCKP and define the minimum lower bound for the collective value of negatively affected criteria's as shown in Eq. 2. If he needs multivariate, then he should use the ABC and define the minimum lower bound for each of negatively affected criteria's as shown in Eq. 4. Finally, the user sets the above parameters in the selection Procedure 1 line 3. Procedure 1 shows the flow of selection procedure and arrangement of the driver of the MR process. Lines 2 to 9 contains the driver with three sections: n number of the mappers, an IMR Agent, and

the reducer. At the beginning of the Selection (Line 3), value for the Threshold_Plans and the Type of Selection are assigned. Next, n number of mappers are specified (Line 5). Line 6 includes the IMR Agent in the procedure, which is responsible for reducing the traffic in the shuffling stage and thereby reducing the burden on the reducer stage. Line 7 includes the reducer class, which processes intermediate results given by the IMR Agent and outputs the optimal (or near-optimal) composition service sequences.

Procedure 1: Selection Procedure

```
1:  Setup initial service data according to the
      - QSD Rule to address the Zip
      - TRD Rule to address the Pareto

2: Selection Class {
3:    // Set initial parameters
        Threshold_Plans ← Number of Plans per Each Batch
        Selection_Requirement ← Type of Selection & QoS constraints

4:    Driver Class {
5:      // Set n number of Mapper Classes under the MultipleInputs
        MultipleInputs.addInput( job, Input_file_1, Mapper_1.class)
        MultipleInputs.addInput( job, Input_file_2, Mapper_2.class)
        . . .
        MultipleInputs.addInput(job, Input_file_n, Mapper_n.class)

6:      // Set Combiner Class
        job.setCombiner(IMR_Agent.class)

7:      // Set Reducer Class
        job.setReducer(Reducer.class)
8:  }end Driver Class
9:}end of Selection Class
```

6.3.4.2 MAPPER Algorithm

Algorithm 1 represents the proposed mapper algorithm for the selection procedure. Lines 1 - 15 represents the kth mapper of the Procedure 1 of the n number of mappers. In Line 2, it generates possible plans for the given search space. Next, Lines 3 to 14 show looping through the set of plans according to the arranged number of threshold plans. During this process, the procedure executes the selection Selection_Requirement

that was initially set by the user. This generates a set of possible optimal planners for respective batches of plans. Lines 4 to 6 involve the linear optimal selection requirement and the Dijkstra algorithm is invoked. Lines 7 to 9 involve the combinatorial selection requirement and the 0-1 MCKP algorithm is invoked. Lines 10 to 12 involve the multivariate-selection requirement and the ABC algorithm is invoked. The inputs and outputs of all three algorithms are synchronized to the same data structure, thereby facilitating smooth operation throughout the selection process without compromising the overall selection criteria. Each job of the mapper results in a chunk of composition results, initiated from the given candidate service of the first task. Line 13 writes the set of resulting plans to the context.

For the above example, the results are in the form of Level id vs optimal service selection sequence#profits, for $1 \leq k \leq n$, as follows.

1: $S_{11}@S_{21}@S_{31}@S_{41}$#2.964

<Results of all possible combinations from S11 to 4th Task of n mapper of p batches>

2: $S_{12}@S_{2n}@S_{3n}@S_{4n}$#1.460

<Results of all possible combinations from S12 to 4th Task of n mapper of p batches>

…

k: $S_{1k}@S_{23}@S_{34}@S_{4k}$#2.960

<Results of all possible combinations from S1k to 4th Task of n mapper of p batches>

…

n: $S_{1n}@S_{2n}@S_{3k}@S_{4n}$#1.360

<Results of all possible combinations from Sn1 to 4th Task of n mapper classes of p batches>

Algorithm 1: Mapper_k

```
1:procedure Map (key, service Information)
2:    Generate Plans ← Create All possible plans' service info & Model Plan
3:    for all plan_num < Threshold_Plans do
4:        if  Selection_Requirement = Linear Optimal then
5:            Mapper_Result ← Execute Dijkstra Algorithm
6:        end if
7:        else if  Selection_Requirement = Combinatorial then
8:            Mapper_Result ← Execute 0-1 MCKP Algorithm
9:        end else if
10:       else then //Selection_Requirement = Multivariate
11:           Mapper_Result ← Execute ABC Algorithm
12:       end else
13:   context.write (Mapper_Result ← Task_1 service, optimal_plan#profit)
14:   end for
15:end of Map() procedure
```

6.3.4.3 IMR Algorithm

Algorithm 2 represents the algorithm for IMR Agent. This procedure receives batches of intermediate results of the selection. Two main objectives are involved in the IMR. Reduce the shuffling traffic congestion, and reduce the workload of the reducer phase. This leads to a dramatic reduction in internal traffic and the multiple reducers. In Line 2, the procedure initializes the optimal profit and optimal plan sequence values. Lines 3 to 8 loop to find optimal profit under the given key value and find the optimal plan sequence under the key. Line 9 writes the key with the concatenated optimal plan and profit to the MR context.

During the IMR Agent class, the context is processed as an intermediate process, below is sample output from the IMR Agent. The information is KEY string value vs optimal plan sequence with the profit value for the given sequence, namely KEY vs optimal service selection sequence#profits, for $1 \leq k \leq n$.

KEY: $S_{11}@S_{21}@S_{31}@S_{41}$#2.964 // This is the optimal sequence among all possible combinations from S_{11} to 4^{th} Task of the 1st mapper.

<Results of all possible combinations from S11 for n mapper>

KEY: $S_{12}@S_{2n}@S_{3n}@S_{4n}$#1.460 // This is the optimal sequence among all possible

combinations from S_{12} to 4th Task of the 2nd mapper.

<Results of all possible combinations from S_{12} for n mapper>

…

KEY: $S_{1k}@S_{23}@S_{34}@S_{4k}$#2.960 // This is the optimal sequence among all possible combinations from S_{1k} to 4th Task of the kth mapper.

<Results of all possible combinations from S_{1k} for n mapper> …

KEY: $S_{1n}@S_{2n}@S_{3k}@S_{4n}$#1.360 // This is the optimal sequence among all possible combinations from S1n to 4th Task of the nth mapper.

<Results of all possible combinations from $_{S1n}$ for n mapper>

Algorithm 2: IMR Agent

```
// Designed to avoid multi cross shuffling, and continue same Key of Mapper
// used in as the Key of Reducer to address shuffling stage traffic concerns
1:procedure IMR Agent (key, values,  context)
2:     initialize optimal_profit and optimal_plan
3:     for all values do
4:         if existing_profit < new_profit then
5:             optimal_profit = new_profit
6:             optimal_plan = new_plan
7:         end if
8:     end for
9:     context.write (“KEY”, optimal_plan#profit)
10:end of IMR Agent procedure
```

6.3.4.4 REDUCER Algorithm

Algorithm 3 represents the flow of the reducer. This finds an optimal plan for the given key by splitting the optimal plan and profit string resulting from the IMR Agent. Part of the reducer job already completed by the IMR agent. Lines 3 to 8, procedure find the optimal plan and profit value for the particular key. Line 9 writes the plan and profit to the MR context. Below is sample output from the reducer. The information is an optimal (or near P = optimal) plan sequence vs profit value for the given sequence, where $1 \leq k \leq n$.

$S_{1k}@S_{21}@S_{31}@S_{41}$ vs 2.960

Algorithm 3: Reducer

```
1:procedure reducer (key, values,  context)
2:      initialize optimal_profit and optimal_plan
3:      for all values do
4:          if existing_profit < new_profit then
5:              optimal_profit = new_profit
6:              optimal_plan = new_plan
7:          end if
8:      end for
9:      context.write (optimal_plan, profit)
10:end of reducer procedure
```

This page intentionally left blank.

Chapter 7 Verification and Refinement (VR) Stage of the ASC

In this chapter, we present service oriented partial order planner. **Objectives** are verifying and refining the composite service based selection results to guarantee the seamless execution and satisfaction of the BDA requirement. **Key contributions** are we proposed complete modelling for composite service based constraints aware POP planner and algorithms for the planner, which is called as SoPOP. As the **future works**, SoPOP needs to improve with adoptability and inter-operability between execution stage and VR stage.

ASC is a cognitive solution to the data science, which is natively constraint aware, resource and time consuming jobs. However, during the service execution, partial-ordered workflow (PoW) for the analytics may violate the original constraints of the data analytics (objective), and automation of the process. Then violation of automation leads to consuming excessive resource and time. Therefore, verification and refinement (VR) are essential before executing the services to avoid the violation of objectives, and automation. Converting the selection results in to a PoW and VR are two main sub stages of the VR process. However, complex tasks are satisfied by composite web services (CWS). CWS Composing may lead to occur flaws, which are main caused to a violation of the objective and automation. Nevertheless, it is essential to reverse engineering the CWS selection result in to a partial-order-planning problem without losing the integrity of the CWS's. Therefore we proposed, CWS based PoW (CPoW) generation algorithm. Next, we proposed constraint-aware service-oriented partial-

order planner (SoPOP) algorithm based on the CPoW for successful VR thus result in the flawless planner. Experiments demonstrate the proposed CPoW and SoPOP are well behaved to have a successful VR.

7.1 Motivating Scenario

A user with BDA requirement as described in Section 2.2, processed his/ her requirement with proposed ASC architecture. At the beginning, we prepared a constraint aware workflow for the analytics as we described in Chapter 4. Then, the abstract workflow for the analytics and Fig. 7.1.1 top layer is the data preparation part of the CRISP-DM process. Fig. 7.1.1 middle layer represents the levels, and tasks which are contained under the levels. Each task is compromised with two pre and post-conditions to ease the understanding of the process. After that, it selects services for the respective tasks based on transaction aware QoS awareness [7]. Then it is essential to verify and refine the workflow for the analytics before starts the execution stage due to the possible user cases present in Fig. 7.1.2.

Levels represent the number of horizontal tasks, mutually inclusive (AND), mutually exclusive tasks (OR) considered as a one level tasks. Respective tasks need to be a satisfied set of pre and post-conditions before and after the execution. In this scenario, it displays as two pre and post conditions for each tasks.

Fig. 7.1.1.below shows respective CWS are selected for the respective tasks. Here we assume, a given CWS contained two atomic web services, each web service consists with two operations, set of input and an output.

Fig. 7.1.2 shows, possible flaws (open goals, threats which are violation of ordering constraint and automation) may be occurred during the execution those services in ASC while composing composite web services. Fig. 7.1.2.a presents the user case 1 : CWS_2 has a its own subgoal, i.e. post-condition called as type conflicts issue, which is not satisfied by the two of the selected two services under given CWS_2. This is called as the open goal. Fig. 7.1.2.b presents the user case 2: That the one of the input of the service of the CWS_2 uses different file format than the output of the CWS_1. Then it

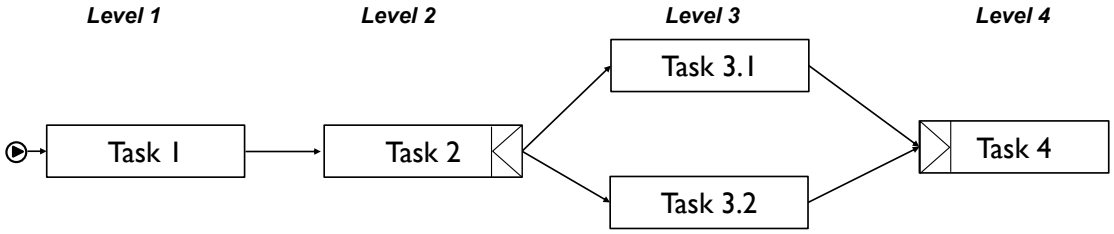
occurs input output signature conflict issue. In addition to that, the first service of the CWS_2 uses additional file to execute its operation. Then these two issues are called as violation of automation or binding constraint violation. And Fig. 7.1.2.c presents user case 3 : One of the precondition of the CWS_2 is not satisfied by none of the services of the CWS_1. This is known as the violation of ordering constraint.

Therefore, before the selection results going to the execution stage, it is required to;

– Verify the workflow to find the availability of the flaws, if they occur, and then it is an essential requirement to

– Refine those flaws to

- Satisfy the original constraints for the analytics, to satisfy overall G_{BDA},
- Maintain the original flow of the abstract workflow, to minimize consuming resource and makespan and
- Minimum deviation from the abstract flow for the analytics, to minimize extended consumption of resource and time after refinement. Constraint-aware Service Oriented Partial Order Planner



Level 1	Level 2	Level 3		Level 4
Task 1	Task 2	Task 3.1	Task 3.2	Task 4
Pre-cond. 1-1 Pre-cond. 1-2	Pre-cond. 2-1 Pre-cond. 2-2	Pre-cond. 3.1-1 Pre-cond. 3.1-2	Pre-cond. 3.2-1 Pre-cond. 3.2-2	Pre-cond. 4-1 Pre-cond. 4-2
Post-cond. 1-1 Post-cond. 1-2	Post-cond. 2-1 Post-cond. 2-2	Post-cond. 3.1-1 Post-cond. 3.1-2	Post-cond. 3.2-1 Post-cond. 3.2-2	Post-cond. 4-1 Post-cond. 4-2

CWS 1		CWS 2		CWS 3.1		CWS 3.2		CWS 4	
S1.1	S1.2	S2.1	S2.2	S3.1.1	S3.1.2	S3.2.1	S3.2.2	S4.1	S4.2
Op1.1.1 Op1.1.2$\cup_1^m \iota_j$ $\vartheta 1.1$	Op1.2.1 Op1.2.2$\cup_1^m \iota_j$ $\vartheta 1.2$	Op2.1.1 Op2.1.2$\cup_1^m \iota_j$ $\vartheta 2.1$	Op2.2.1 Op2.2.2$\cup_1^m \iota_j$ $\vartheta 2.2$	Op3.1.1.1 Op3.1.1.2 $\cup_1^m \iota_j$ $\vartheta 3.1.1$	Op3.1.2.1 Op3.1.2.2 $\cup_1^m \iota_j$ $\vartheta 3.1.2$	Op3.2.1.1 Op.3.2.1.2 $\cup_1^m \iota_j$ $\vartheta 3.2.1$	Op.3.2.2.1 Op.3.2.2.2 $\cup_1^m \iota_j$ $\vartheta 3.2.2$	Op4.1.1 Op4.1.2$\cup_1^m \iota_j$ $\vartheta 4.1$	Op4.2.1 Op4.2.2$\cup_1^m \iota_j$ $\vartheta 4.2$

Figure 7.1.1. Workflow for the data preparation stage respective tasks, their condition and selected CWS's

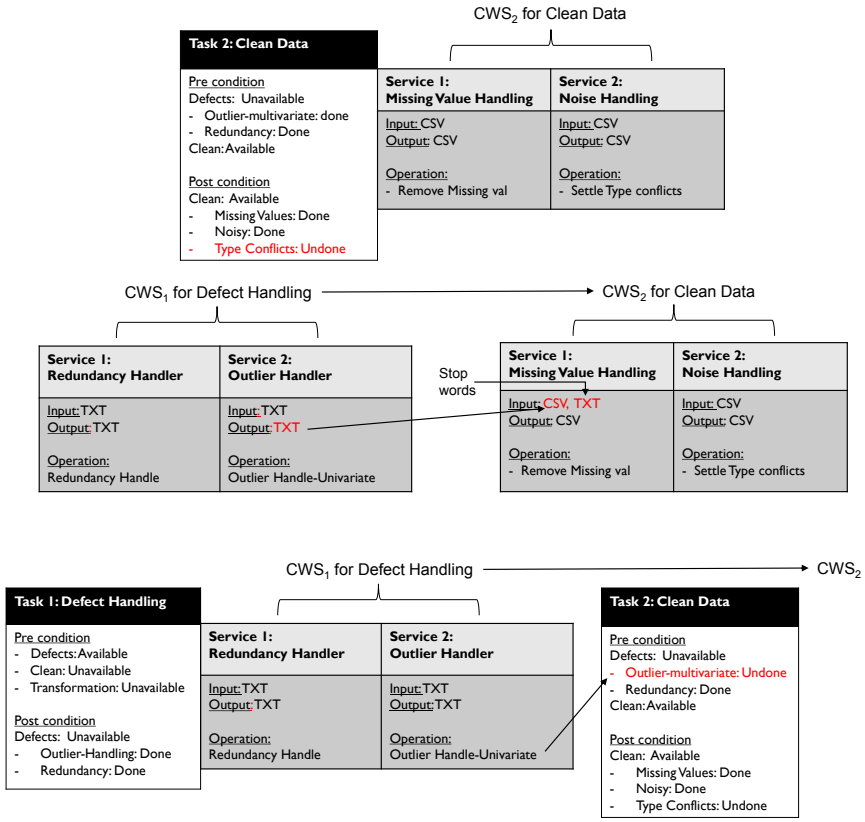


Figure 7.1.2 Probable flaws identified in ASC process composing CWS

7.2 Constraint-aware Service Oriented Partial Order Planner

7.2.1 Introduction

Already completed and related content is pending to be added to the thesis. The expansion in services has led to increased opportunities. Studies show that the growth in revenue has more than doubled in bi-annually, with an increase more than 220% in service-based data-science-related activities in the Amazon Web Services Platform in 2015, 2016 and 2017 compared to the respective 2013, 2014 and 2015.8 In addition, studies show a doubling of the volume of services in the ProgrammableWeb.com store each year. Moreover, ProgrammableWeb is becoming a popular platform for well-known providers such as IBM, Google, and Microsoft [40]. This confirms that growth

8 http://www.statista.com/statistics/233725/development-of-amazon-web-services-revenue

in the consumer market and the availability of services is extensive due to the applications related data science have been increased in an extensive manner [40], [109]–[111].

Modern data science, such as Big data analytics (BDA) evolves with multidimensional constraints in conventional business analytics such as ; excessive voluminous, highly variable and high velocity (3V data). BDA leverages to manage data that involves additional dimensions of high variability and the need for high veracity (5V data), which is appearing in fields such as deep learning (DL) [1], [2]. However, recent data science techniques such as BDA and DL processes consume excessive resources and time. These limitations are hampering the meaningful adoption of the data science across research and industry domains [6]–[8]. However, such a data science processes require highly diverse and rigorous stages to be successful, which involves very large resources and makespan (the overall time for the process). This constrains the full meaningful effects of a state-of-the-art data science product. Automating the data analytics process offers the most cognitive solution to these concerns [9], [112]. Automation of data analytics such as BDA may enable data scientists to accomplish their task in days rather than the traditional period of months.9

Data Analytics, such as BDA automation has been discussed in the existing literature considering various aspects such as constraints handling, which derives from its explicit and implicit constraints on its dynamic process and requirements with respect to; time, resources, data, modelling, and deployment. Existing literature has discussed the attempts considering their stages of the process, such as data preparation, modelling, model optimization, or deployment [13], [19], [20], [69]. We propose to automate BDA based on automatic service composition (ASC) [11]. Our aim is to automate the data preparation, modelling, evaluation, and deployment of the cross-industry standard process for data mining (CRISP-DM) [10] based on ASC [12].

Proposed ASC has main five stages, which are planning, discovery, selection, verification & refinement (VR) and execution involved in the ASC process. Existing

9 http://news.mit.edu/2016/automating-big-data-analysis-1021

works related to BDA automation have proposed the use of various techniques to prepare the workflow, which was either partial or involve manual intervention. According to both domain experience and literature reviews [13], [15]–[17], [19], [20], [69], we can observe that seamless workflow is the foundation of seamless automation of BDA. Furthermore, the automated workflow is an intrinsic requirement of the ASC process. Therefore, we have divided the planning problem into two stages. The first is the planning stage, which prepares the abstract workflow for analytics by considering domain specific concerns. The second stage occurs after the service selection stage but before the service composition (execution stage), called the VR stage, which refines the workflow to prepare a concrete workflow for analysis.

We proposed constraint aware workflow generation based on the Graphplan technique to prepare abstract workflow for analytics based on the ASC [78]. Implicit and explicit constraints are the main causes to increase the complexity of the workflows. According to the resulted workflow described in Chapter 4, these implicit constraints are mutually exclusive (mutex ; tasks cannot occur at the same time, i.e., XOR logic) and others are mutually inclusive (mutin; events can occur simultaneously as well as independently, i.e., AND logic) [16], [70]. This causes the evolution in the dynamic workflow for BDA [71], [72]. It is essential to maintain the original pattern of the work in the execution stage, which is main cause to reduce the resource consuming and the makespan [70], [77], [113], [114].

Nevertheless, during discovery stage, process discover the most suitable services for the given functional requirement of the tasks, which are satisfied by the composite web services (CWS), due to the complexity of the tasks in the workflow. As an example, defect-handling task, which is a un-ground/nonprimitive task, which made up by a set of ground/primitive tasks, which cannot be satisfied by an atomic service. The set of set of ground/primitive tasks such as remove null values, missing values, truncation of data, and type mismatch. CWS, which made of a collection of atomic services are used to satisfy those primitive tasks. In the selection stage, procedure selects the respective CWS's to satisfy the given tasks considering QoS requirements. Then composing of these CWS may occur flaws during the execution stage. Therefore, replanning (VR) is

essential before precedes to the execution stage. Then the selection result becomes a partially-ordered CWS based selection result. Therefore to continue the VR process, it is essential to prepare the partial-order planning (POP) problem from the existing selection result. This POP problem is called as the as the partial-order workflow (PoW). However, during the VR process, it may occur unavoidable constraints when the VR procedure uses an atomic services based PoW problem solving such as B. Wang et. al. [115] and P. Wang et. al.[116]. At first, generating PoW from the atomic services shows limited capabilities compared to CWS based selection results. Secondly, solving the PoW based on atomic service based POP solver are disrupting the integrity and smooth functioning of the selected CWS's for the task in data analytical workflows. Because such POP solvers add refinements in between the CWS services. As a third reason, solving POP based on atomic services may cause to violate the functional and qualitative objectives of selecting CWS for the task in the workflow. Considering the functional constraints perspective, generally a given CWS consists with collaborative functioning. Then including a service by a different provider in the middle of a given particular CWS may constrained the smooth functioning, consume additional time, and resource. In addition to that, such issues may ruin the qualitative objective such as transaction awareness of the CWS's [6], [34], [48]. Moreover, literature related to the service based partial order planners are scarce [62], [115]–[118]. However, they are discussed the atomic service based POP solving techniques. Therefore we proposed CWS based PoW (CPoW) generation modelling and algorithm.

Basically two types of flaws can occur in PoW. i.e., open goals (unsatisfied subgoals of respective task, i.e. post-condition such as the clean_data CWS failing to clean type conflicts issues) and threats (such as violation of constraint ordering and violation of automation). Violation of constraint ordering means the violation of the order of preconditions. For example, the action of a given selected service prior to the file_convert_to_csv CWS may violate the precondition of cleaned_data. Violation of automation means, violation of binding variable, ie; the output of a given service violates the input of the next service. Therefore, the workflow needs to undergo VR to prepare the flawless planner. This is known as the POP problem. However, POP should

be able to detect, rectify and verify all types of flaws in an efficient manner without disrupting the objective of data analytics and automation. A single mistake will lead to disrupting the whole process, waste of time and resources. Nevertheless, the resultant planner should be minimally deviated from the abstract workflow and should be maintained the original shape to maximally reduce consumption of resource and time, also procedure should guaranteeing to satisfy of original constraints (objectives) of analytical requirement. Therefore we proposed constraint aware service oriented partial-order planner (SoPOP) to address the composite service based POP problem. A shortened form of this paper has been presented previously in conference paper [119].

The joint approach of CPoW and SoPOP represents the VR stage of the ASC process. The process considers the POW preparation, and VR in a dynamic environment to guaranteeing seamless uninterrupted execution in minimum resource consumption and makespan. According to our studies and literature review, we are one of the first groups to propose a CWS based PoW and CWS based SoPOP. Our contributions are;

- CWS based PoW
- CWS based SoPOP

The scope of this paper is limited to the replanning the workflow after the selection stage of the ASC process (described in Section 2.3). In this paper, we do not discuss the adoptability of the SoPOP in the ASC process.

The remainder of this paper is structured as follows. In Section 2, we discuss the preliminaries and elaborate on the motivation scenario for BDA automation and POP solving. In Section 3, we present the proposed method. In Section 4, we describe our evaluation of the proposed method. In Section 5, we discuss related work, with Section 6 concluding the paper.

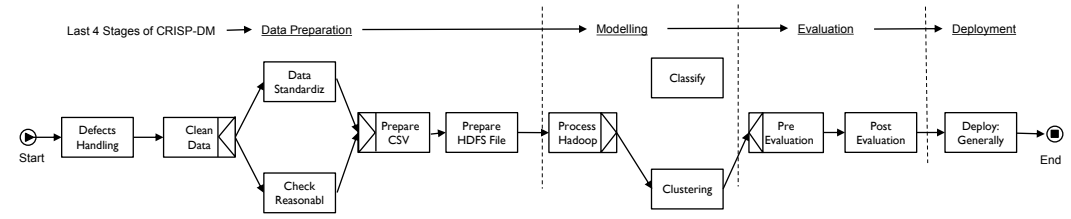


Figure 7.2.1 Workflow for the BDA

7.2.2 Proposed SoPOP

In this section, we discuss the proposed joint approach of the CPoW and ScPOP for constraint-driven service oriented POP generation. In Section 3.1, we proposed system modelling for the VR stage. In Section 3.2, we proposed the respective algorithms for the VR stage, which involves in CWS based PoW generation and SoPOP base on the CPoW to generate the TOP for the BDA in ASC.

7.2.2.1 System Modelling

In this section, we discuss the proposed modelling for the VR Stage of the ASC for the BDA process. In Section 3.1.1, we discuss the workflow generation for the BDA in ASC. In Section 3.1.2, we discuss the CWS based PoW generation. And, in Section 3.1.3, we discuss the TOP generation based on SoPOP method.

7.2.2.1.1 System Modelling for abstract workflow generation

We already discussed this in Chapter 4.2.1.

7.2.2.1.2 System Modelling for the PoW problem

Here onwards we discuss the system modeling of the POP problem of the ASC.

Here,$1 \leq i \leq n$ the number of tasks in the abstract workflow and k is the number of collections of actions (unground tasks) with their respective propositions. According to the ASC process for the given scenario, the planning stage results in selected services for the given tasks indicated by Definition 7. Then, according to the Definition 7, each of the tasks T_i is consistent with a corresponding set of actions and propositions. That is, they are consistent with the functional requirements attributed to them by the planner Π. This can be represented as $T_i = \cup_i^k (A_i \cup P_i)$, where k is the number of unground tasks.

Definition 10: Selection stage results are denoted by σ, an ordered collection of composite Web services (CWS) [6], in which each service maps to a corresponding introducing service task Υ_i:

$$\sigma = \{\cup_1^m CWS_i : \varphi\ (CWS_i) \longrightarrow \Upsilon_i\ \} \qquad (11)$$

Here, φ is the propositional mapper function for each Web service to a new type of

task called a "service task," denoted by Υ_i in the abstract workflow.

Definition 11: Atomic web service, called as is the fundamental building block of the CWS. The includes the set of inputs for a corresponding ws_i, denoted by $\bigcup_1^m \iota_j$ and its output ϑ:

$$ws = \{ \bigcup_1^m \iota_j ,\ \vartheta,\ \bigcup_{j=1}^k s_j^i \} \qquad (12)$$

Definition 12: Sub service state $\mathbf{s_i}$ is made up from a corresponding unground action and proposition P_i:

$$s_j = \{\bigcup_{l=1}^k \left(A_l^j \cup P_l^j\right) \} \qquad (13)$$

Then the corresponding's are make the CWS as defined below.

Definition 13: CWS is the high-level building block of the selection result set, each of CWS is made up by the collection of atomic web services, denoted as (14).

$$CWS_i = \left\{\bigcup_1^m ws_j : ws \longrightarrow \bigcup_1^m \iota_j ,\ \vartheta,\ \bigcup_{j=1}^k s_j^i \right\} \qquad (14)$$

In the process of the converting the selection result set in to a PoW problem, we introduced service task Υ_i.

Definition 14: A service task Υ_i includes collections of subservice states S_i

. In addition, it includes the set of inputs for a corresponding CWS_i, denoted by $\bigcup_1^m \iota_j$ and its output ϑ:

$$\Upsilon_i = \{ \bigcup_1^n (\bigcup_1^m \iota_j ,\ \vartheta,\ \bigcup_{j=1}^k s_j^i) \} \qquad (15)$$

Here, $1 \leq j \leq m$, k is the number of action–proposition pairs and $1 \leq i \leq n$, with n being the number of sub service states defined in (13). Based on these service tasks, we can define the PoW for the given problem.

Definition 15: The PoW; for a given problem is denoted by π:

$$\pi = (\Upsilon, \prec, B, L) \qquad (16)$$

Here, Υ is the partially instantiated set of service tasks and $\prec$ is the set of ordering constraints on Υ of the form ($\Upsilon_i \prec \Upsilon_j$). is the set of binding constraints on the variables in the Υ_i tasks. For example, given Υ_i, the output ϑ_i and a Υ_{i+1} input of ι_l^{i+1}, then the variable binding constraint should be $\vartheta_i = \iota_l^{i+1}$ or $\iota_l^{i+1} \in \text{Typecast}_{\vartheta_i}$. Finally, L is the

causal link which supports the interlinking of two tasks or the introduction of a new input file to the workflow.

3.1.3. System modeling for the SoPOP

For the given scenario, the T_i tasks of the Π planner defined in (9) and the Υ_i service tasks in PoW, π tend to differ in providing actions and propositions as well as violations of the automation. These issues are called flaws in the POP.

Definition 16: The flaws in the POP, denoted by $\mathcal{F}$, are deviations from the functionality that is required to be satisfied by the constraints in the planner. $\mathcal{F}$ is either an unsatisfied subgoal (an open goal) $\bar{G}$ or a threat $\bar{T}$:

$$\mathcal{F} = \{\bar{G}, \bar{T}\} \qquad (17)$$

Definition 17: The open goal type of flaw is denoted by $\bar{G}$ and is caused by an unsatisfied precondition associated with a given task T_i. Based on (6) and (13), it is the difference between the required-to-be-satisfied subgoals that it has been assigned to the particular task T_i, here we discussed as the post conditions (unground states) and those of the selected CWS: Relative complement of subgoals of the given task T_i of workflow Π in subgoals of respective Υ_i in σ.

$$\bar{G}_\iota = \{(\bigcup_{j=1}^{k} s_j^i \ of\ \Upsilon_i) \cap (\bigcup_{j=1}^{k} s_j^i \ of\ T_i)^c\} \qquad (18)$$

According to the given scenario, at the end of the data preparation stage, all the open goals should be refined before proceeding to the modeling stage. Any unsatisfied open goals must be refined before proceeding.

Definition 18: The threat type of flaw is denoted by $\bar{T}$ and is caused by one of two main reasons: violation of constraint ordering $\prec$ or violation of automation involving the binding constraints B.

$$\bar{T} = \{\prec, B\} \qquad (19)$$

Proposition 1: The POP $\boldsymbol{\pi_a} = (\boldsymbol{\Upsilon}, \prec, \boldsymbol{B}, \boldsymbol{L})$ is a solution to the planning problem Pa = $\{S; \Sigma, s_i, G_a\}$ if $\Phi(s_i, \pi)$ satisfies the subgoal G_a of the workflow set under the main goal G_{BDA}.

Proof: If π has no flow, the ordering constraints $\prec$ and variable binding B are consistent. Next, every sequence of $(\Upsilon_1, \ldots, \Upsilon_n)$ in all the service tasks in $\Upsilon - \{S_1, G_a\}$,

where S_1 is the initial state and G_a is the subgoal, implies totally ordered, grounded, and satisfied $\prec$ and B. If we apply the total order plan $(Y_1, \ldots, Y_n)$ to the initial state S_i, it should satisfy the G_a. That is, $\Phi(S_i, (Y_1, \ldots, Y_n)) \in G_a$.

7.2.2.2 Proposed Algorithm

In this section, we discuss the proposed algorithm for the VR stage of the ASC process. In Section 3.2.1, we discuss the algorithm for the CWS based PoW. In Section 3.2.2, we discuss the algorithm for the SoPOP.

7.2.2.2.1 CWS based PoW of the VR stage

The given selection results contained the collection of CWS as described in our previous work [6], [120]. Then each CWS associate with a collection of unground states, these states are associated with the collection of a pair of action and propositions. The $\bigcup_1^m \iota_j$, ϑ are collections of inputs and output of the CWS. Algorithm for the PoW problem preparation is shown in Algorithm 1.

Then based on the result, we mined respective functionalities, their proportions, list of inputs and outputs of the respective WSDL files associated with respective CWS. Based on this information we prepared the service oriented PoW. Line 2 gets the list of CWS associate with respective selection result called as sr. And the output of the algorithm is the PoW for the given selection result. From line 3 to 20 iterates the each of CWS and mined the respect information to prepare the PoW for the given selection result. Line 5 to 12 mine the information from wsdl and it prepares the list of states named as S.list() with associate the list of action and their propositions (A→P) operation list from the wsdl. Line 13 and 14 get the list of input; ι_j.list() and an output; of the CWS. Using the ι_j. list(), ϑ, and S.list(), it prepares the service task of the POP. Next, line 16, it retrieves the ordering constraint named as $\prec_i$ $(CWS_i \prec CWS_k \prec CWS_j)$ of the given CWS_k using the list of CWS. Line 17, it find types of binding variables named as B_i of input and outputs of the CWS. After that, line 18 it finds the causal links named as L_i of associated web services of the CWS. Finally, it creates the node of the PoW as π_i using the respective information $\Upsilon_i, \prec_i, B_i,$ and L_i.

7.2.2.3 SoPOP of the VR stage of the ASC

This section contained three sub sections, at the first SoPOP algorithm describes the algorithm for the SoPOP of the proposed VR stage of the ASC, next Verification algorithm next Section B describes the algorithm for the verification of the SoPOP. Finally, refinement algorithm describes the algorithm of the refinement of the SoPOP with respective user cases described in Section 2.4.

SoPOP algorithm

The proposed SoPOP has two main functions: verification and refinement. Algorithm 2 describes the pseudocode for the proposed SoPOP. The algorithm is based on the PoW. Inputs used in the process are $planReq_i$, $absWF_i$, and $selRes_i$; i represents the subgoals of the BDA workflow as shown in Fig. 7.2.1. Here i = 1, 2, 3, 4. Here 1 = Data Preparation, 2 = Modelling, 3 = Evaluation and 4 = Deployment. The $planReq_i$ is propositional planning requirement of the given subgoal, the $absWF_i$ is the abstract workflow resulted to satisfy for given planning requirement and the $selRes_i$ is the selection result of the CWS based services to satisfy given tasks in the abstract workflow.

SoPOP mainly contained two main functions, which are SoPOP and V&R. SoPOP represents the main method to prepare the total order planner (TOP) for the BDA. V&R represents the verification and refinement of the respective subgoals. In Line 2, the SoPOP process initializes the subgoal G_i using $planReq_i$, abstract workflow Π_i using $absWF_i$, σ_i, and $F_i(T)$ using the $selRes_i$. In Line 2, procedure prepares the PoW for the given subgoal using respective σ_i, which result of the selection as described in Section

3.2.1. The given selection results σ_i contain the collection of CWS specified in

(11) and described in our previous work [6]. Then, according to (11), (12), and (13), each CWS is associated with a collection of unground states, which are, in turn, associated with collections of action–proposition pairs, as shown in (18). The $\bigcup_1^m \iota_j$,

ϑ are collections of the inputs and output of the CWS:

$$CWS_i = \{\cup_{l=1}^{k}(A_l^j \cup P_l^j),\ \cup_1^m \iota_j,\ \vartheta\} \qquad (20)$$

Line 4, it prepares the respect TOP for the given subgoal and line 5 it prepares the TOP for the overall BDA process. During the V&R procedure, first it verify the availability of the open goals, and threats (violation of automation and ordering constraints) using the verification method as shown in Line 7. Next, the procedure refine respective flaws (open goals and threats) as shown in Line 8. After that, procedure check for availability of flaws as shown in Line 9 and Line 10 to 12, procedure works for return PoW as TOP or recall the V&R method itself, if PoW doesn't refined successfully.

Verification algorithm

Verification procedure verifies the possible flaws in the PoW. Algorithm 3 represents the pseudocode for the verification process. Inputs of the verification are Π_i, σ_i and PoW_i which are already prepared by the SoPOP method. The pseudocode mainly contains five sub methods which are verification, verify open goals, verify ordering constraints and verify binding constraint violation.

In Line 2, 3 and 4, procedure update its own PoW if it is available any given respective flaws, which are open goal, ordering constraint and binding constraint violation. And finally procedure returns the updated PoW with respective flaws occurred by selected CWS as shown in Line 5.

Line 6 to 11 represent the verification process for the open goal. Procedure loops the selection results σ_i, compare respective CWS with respective tasks in Π_i for post-condition violations. In Line 8, procedure finds the emptiness of the two sets of subgoals of particular tasks of both σ_i and Π_i result sets. If $postCond^{\Pi_j}$; called as A and $postCond^{CWS_j}$; called as B are sets, then the relative complement of A in B, also termed the set difference of B and A, is the set of elements in B but not in A. This is called relative complements of the A in B. And we represent the relative complement as $B \cap A^C$. If the resultant set is not emptied then the resultant set is considered as the

open goals of the given service task γ_j of PoW. Line 11 returns the updated PoW.

Algorithm 1: Prepare the PoW using the Selection Result

Input: sr: selection result
Output: pop: Partial Order Planner
BEGIN
function preparePOP(sr)
    CWS.list() ← getCompositeServiceList(sr)
    for CWS.list **each**
        WS.list() ← getWebServiceList(CWS_i)
        for WS.list() **each**
            op.list() ← getTypeofOperationList(wsdl)
            for op.list() **each**
                (*A*→*P*).list(): action&prop.list()
                ← getOperationResultList()
            end of for op.list()
            S_i. create() ← ((*A*→*P*).list())
            S.list() ← add(S_i)
        end of for WS.list()
    ι_j.list() ← getInputList(CWS)
    ϑ ← getOutPut(CWS)
    Υ_i ← (ι_j.list() , ϑ , S.list())
    $\prec_i$ ← getOrderingConstraints(CWS_i.list())
    B_i ← getBindingConstraints(CWS_i)
    L_i ← getCausalLink(CWS_i)
    $\pi_i.element = (\Upsilon_i, \prec_i, B_i, L_i)$
    pop.add($\pi_i.element$)
    end of for CWS.list()
    return pop
END

Algorithm 2: BDA Verification and Refinement stage SoPOP of the ASC

Input: $planReq_i, absWF_i , selRes_i$; i = 1,2, 3,4
Here 1= *Data Preparation*, 2 = *Modelling*, 3 = *Evaluation* and 4 = D*eployment*
Output: Total Order Planner: TOP_{BDA}
BEGIN
function SoPOP
    $G_i, \Pi_i , \sigma_i , F_i(T)$ ← Initialize($planReq_i, absWF_i , selRes_i$)
    PoW_i ← preparePoW(σ_i)
    TOP_i ← V&R(Π_i , σ_i , PoW_i)
    TOP_{BDA} ← $\cup_i TOP_i$
function V&R (Π_i , σ_i , PoW_i)
    $vPoW_i$ ← verification(Π_i , σ_i , PoW_i)
    $rPoW_i$ ← refinement($vPoW_i$)
    isRefined ← reVerify($rPoW_i$)
    if isRefined **then**
        return $rPoW_i$: TOP_i
    else return V&R ($\Pi_i , \sigma_i , rPoW_i$)
END

Line 12 to 17 represents the verification of the ordering constraints. Flow of the

Algorithm 3: Verification process of the SoPOP

Input: Π_i, σ_i, PoW_i ; i = *1,2, 3,4*
Here 1= *Data Preparation*, 2= *Modelling*, 3 = *Evaluation*, 4 = *Deployment*
Output: top: Verified PoW: $vPoW_i$
BEGIN
function verification (Π_i, σ_i, PoW_i)
  PoW_i ← updateOpenGoals(σ_i, Π_i, PoW_i)
  PoW_i ← updateOrderingConstraint(σ_i, Π_i, PoW_i)
  PoW_i ← updateIPOViolation(σ_i, PoW_i)
  return PoW_i
function updateOpenGoals(σ_i, Π_i, PoW_i)
  for σ_i **each** CWS_j
    if $postCond^{\Pi_j} \setminus postCond^{CWS_j}$!= null **then**
      $openGoals_j^{\Pi}$ ← $postCond^{\Pi_j} \setminus postCond^{CWS_j}$
      PoW_i ← update $openGoals_j^{\Pi}$
  return PoW_i
function updateOrderingConstraints(σ_i, Π_i, PoW_i)
  for σ_i **each** CWS_j
    if $preCond^{\Pi_j} \setminus preCond^{CWS_j}$!= null **then**
      $orderConst_j^{\Pi}$ ← $preCond^{\Pi_j} \setminus preCond^{CWS_j}$
      PoW_i ← update $orderConst_j^{\Pi}$
  return PoW_i
function updateVBViolation(σ_i, PoW_i)
  $LCWS$ ← getLevelvsCWSList(σ_i)
  for σ_i **each** $LCWS_j$
    if $inputList^{LCWS_{j+1}} \setminus outputList^{LCWS_j}$!= null **then**
      $vbViolation_j^{\Pi}$ ← $typeConflict^{LCWS_{j+1}} \setminus additionalInputs^{LCWS_j+1}$
      PoW_i^{SG} ← update $vbViolation_j^{\Pi}$
  return PoW_i
function reVerify(PoW_i)
  for PoW_i **each** γ_j
    if $openGoal_j$ **or** $vbViolation_j$ **or** $orderConst_j$ **exists then**
      return false
    else return true
END

Algorithm 4: Refinement process of the SoPOP

Input: Verified PoW: $vPoW_i$
Output: top: Refined PoW: $rPoW_i$
BEGIN
function refinement(PoW_i)
  PoW_i← refineOpenGoals(PoW_i)
  PoW_i← refineFlaws(PoW_i)
  return PoW_i
function refineOpenGoals (PoW_i)
  for PoW_i **each** γ_j
    if $openGoal_j$ **exists then**
      PoW_i **insert** γ_{new} to in front of γ_j
  return PoW_i
function refineFlaws (PoW_i)
  for PoW_i **each** γ_j
    if $orderCons_j$ **not exists AND** $vbViol_j$ **exists then**
      PoW_i**: insert** cl_{new} : ie, causal link with respective I/O to behind γ_j
    if $orderCons_j$ **exists AND** $vbViol_j$ **not exists then**
      PoW_i**: insert** γ_{new} behind γ_j
    if $orderCons_j$ **exists AND** $vbViol_j$ **exists then**
      PoW_i**: insert** γ_{new} with causal link with respective I/O to behind γ_j
  return PoW_i
END

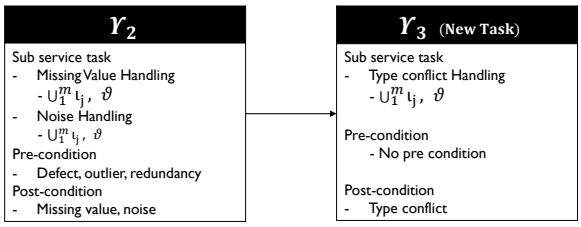
procedure follows the same rule and the way we verify the open goal, here we find the

relative complement of the $preCond^{\Pi_j}$ in $preCond^{CWS_j}$. Line 18 to 24 represents the verification process for the binding constraint violation known as violation of automation. Here procedure loops the CWS's with respective to their level id's and finds for relative complements of input variable signatures of the given first atomic web service of the CWS_{j+1} and outputs of the given last atomic web services of the CWS_j : CWS_j is prior CWS of the CWS_{j+1} . If it is not emptied, then procedure the update the possible binding constraint violation in PoW. In Line 24 procedure returns resultant PoW. In the re-verification method, procedure verifies for availability of any possible flaw of the PoW, if it find return false or else returns true.

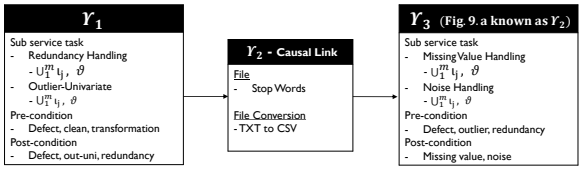
Refinement algorithm

Algorithm 4 represents the refinement of the SoPOP process. Here input of the refinement process is used as the verified PoW. Refinement process contains three sub methods, which are refinement, refine open goals and refine flaws. Fig. 7.2.2 shows respect refined user cases described in Section 7.1.

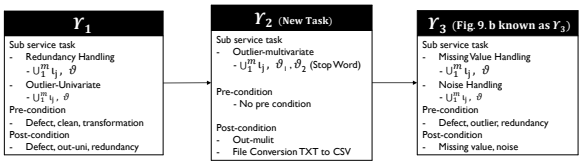
Line 1 to 4 represent refinement method and it calls open goals and threats refining methods. Line 5 to 9 represent the refining the open goals and line 10 to 18 represent the refining threats. During refining open goals, the procedure loops the PoW and if it founds any service task γ_j, which consists any of open goals, then the procedure adds new service task called γ_{new} in front of the γ_j. Fig. 7.2.2 shows the flaw refines with respective to the users cases described in Section 2.4. Then according to the Fig. 7.2.2.a, the procedure adds new service tasks γ_3 named as type conflict handling which doesn't have preconditions and only relevant post-conditions. That means, γ_3 is relatively minimally constraint aware tasks.



a: Refine an open goal adding new task in front of the give Task



b: Refine threat; violation of automation due to mis-matched and unavailable inputs are refined by adding a clausal link before to the respective Task



c: Refine threat; ordering constraints and automation violations are refined by adding new task before the respective Task

Figure 7.2.2: Proposed techniques of Refining open goals and threats

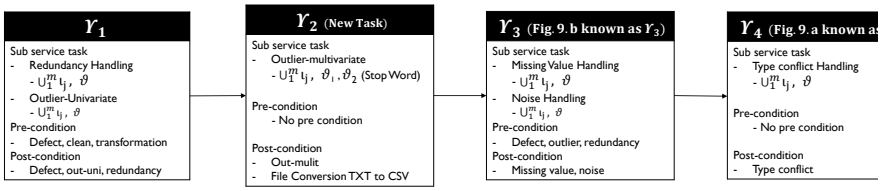


Figure 7.2.3: TOP-Refined Flaws of the PoW

Line 10 to 18, the procedure represents a refinement of the threats occurred in the PoW. Procedure loops the PoW, Line 12 procedure checks for any binding constraint violation, then it adds new causal links to satisfied that issue before the given γ_j. Fig. 7.2.2.b shows the adding new causal link called as γ_2 before that binding constraint violation occurred in service tasks. Line 14, procedure checks for availability of ordering constraints violations, if then it adds new service task instead of causal link prior to the given service task, which is already violated the pre conditions. Line 16, procedure check for violation of both ordering constraint and binding constraint, if it occurs such cases, then the procedure adds new service task prior to the given threat occurred task. New task is relatively more constrained than an added task to refine the

open goal, that means newly introduced task should be satisfied both post-conditions 7and respective binding variable. Fig. 7.2.2.c shows such user case, which adds such a new service to the user case that contained both types of threats. Fig. 7.2.3 shows the resultant snippet of the refined CPoW, which refined all flaws occurred in users cases described in Section 7.1 scenario.

Chapter 8 Execution Stage of the ASC

In this chapter, we discuss the way we implement the execution stage of the proposed solution. **Objective** of this stage is that execute the final stage, that means, selected service sequence to satisfy the given BDA requirement. As a **future works,** it needs to extend execution stage as the adoptable stage that facilitate the execution stage to dynamically deal with VR stage for the necessary improvements or error recovery. We employee previously discussed scenario to explain the execution stage.

8.1 Motivating Scenario

We automated the real time analytical scenario described in Chapter 2.2.1. According to the first two stages of the CRISP-DM, which are business and data understanding Fig. 8.1 shows the high level flow diagram for the scenario. Here our aim is to graphically view the results of the earth quake prone areas.

Then Fig. 8.2 represents the GUI of the proposed solution. At the beginning, it needs to selects the type of analysis and data file plans to use in the analytical process. After that, end-user needs to decide constraints of the data mining process, which are shown

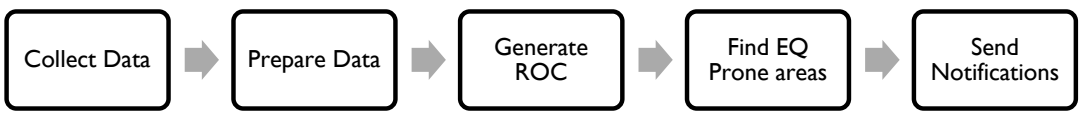


Figure 8.1: High level data flow diagram

as stages of the CRISP-DM under four stages. Next end-user needs to decide the way of automating, i.e. how he employ the ASC to automate the analytical process. ASC process has five stages which are comprised with respective sub-options as radio button selections. Finally, system allows to conduct the automation as the selection.

According to the shown selected criteria for BDA scenario, Fig. 8.3 shows the generated abstract workflow for the analytics. Abstract workflow comprised nine unground tasks. After generating the abstract workflow, system enables the discovery button. Fig. 8.4 shows the discovered CWSs for the given abstract workflow for the analysis. Each tasks comprised with five discovered services. After generating the discovery results, system enables the selection button. Fig. 8.5 shows web service selection result from the discovered web services. Based on the selection results, process conducts the verification and refinement process. Then based on that VR process, finally it prepares the TOP for the analytics. This shows in Fig. 8.6. Finally the process proceeds to the execution stage. Here it pop up the execution initiator which comprised with initiate button, progress bar and functioning web service notify during the analytical process as shown in Fig. 8.6.

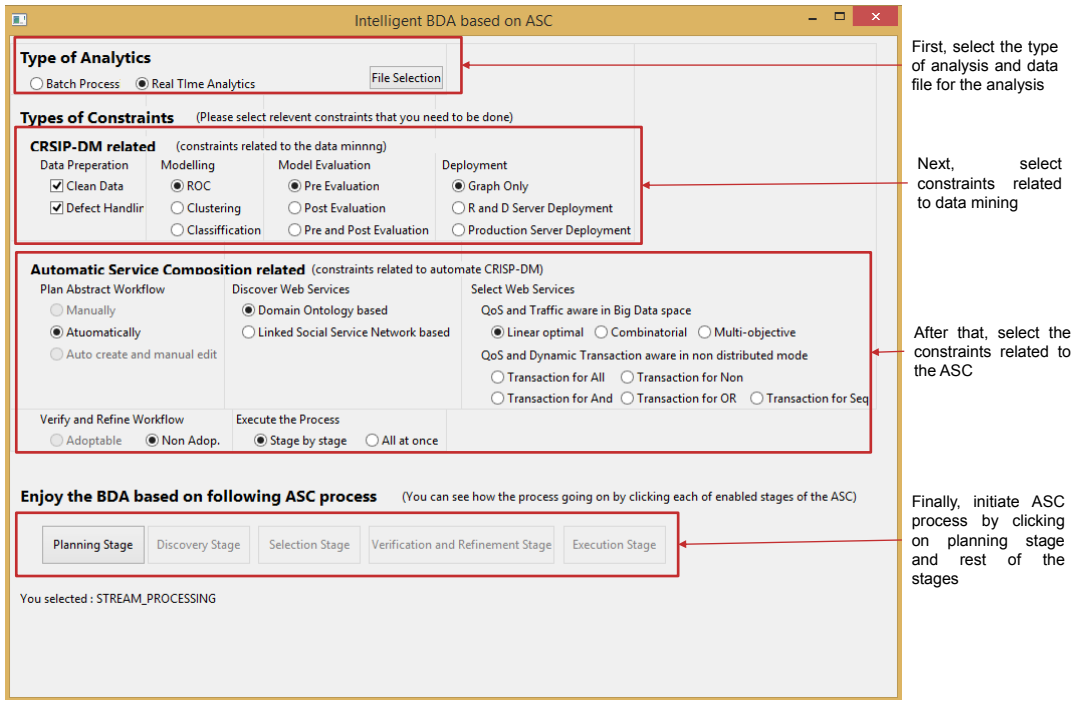


Figure 8.2: GUI of the proposed solution

Finally, user can initiate the composition process by clicking the “start” button. Fig. 8.8 shows the final status of the initiator window and final results of the given analytical scenario. According to that, initiator shows 100% progress and then it generates graph view we selected at the beginning as the result. If the end-user select deploying model,

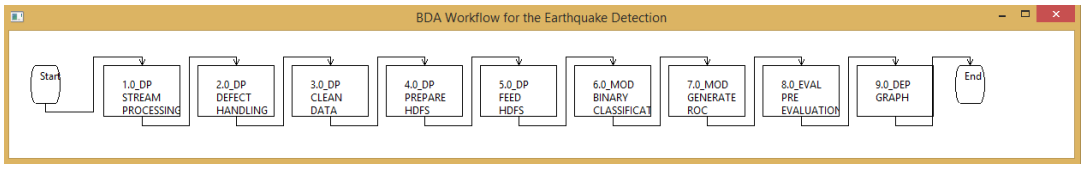


Figure 8.3: Abstract workflow for the scenario

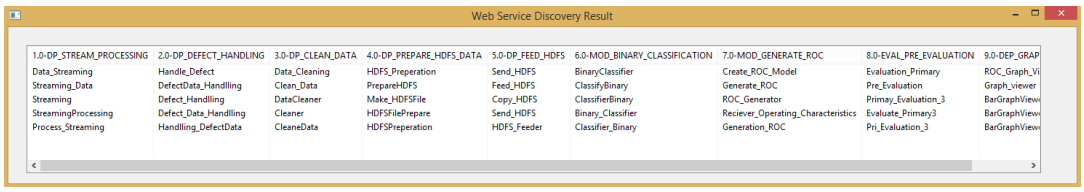


Figure 8.4: Web service discovery results for the Fig. 8.3 shown abstract workflow

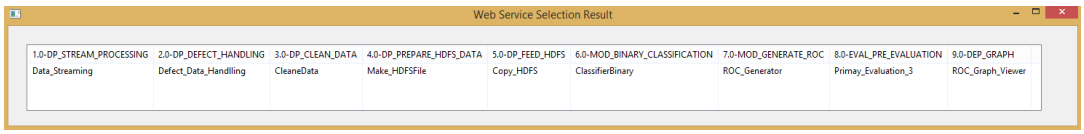


Figure 8.5: Web service selection results for the Fig. 8.4 shown discovery result

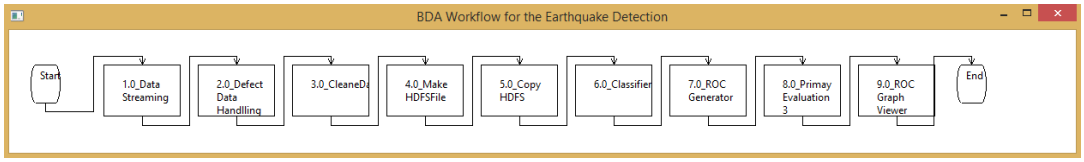


Figure 8.6: Verification and refinement results for the Fig. 8.5 shown service selection result

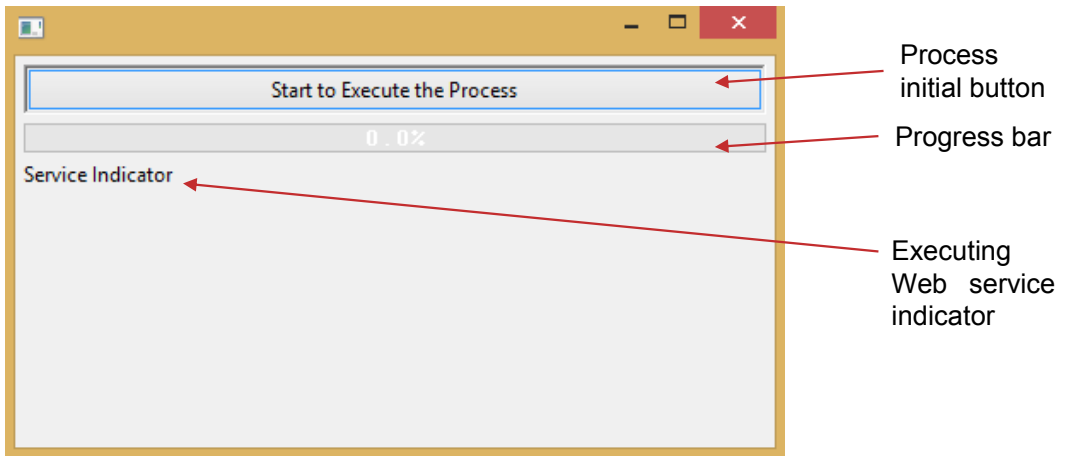


Figure 8.7: Execution process initiator

then it will save the model for the analysis in respective server location. According to the scenario, based on the graph it can identifies the threaten areas for the earthquake detection, which are shown over the threshold line of the graph. Relevant authorities can send notification to the areas detected as the threaten areas.

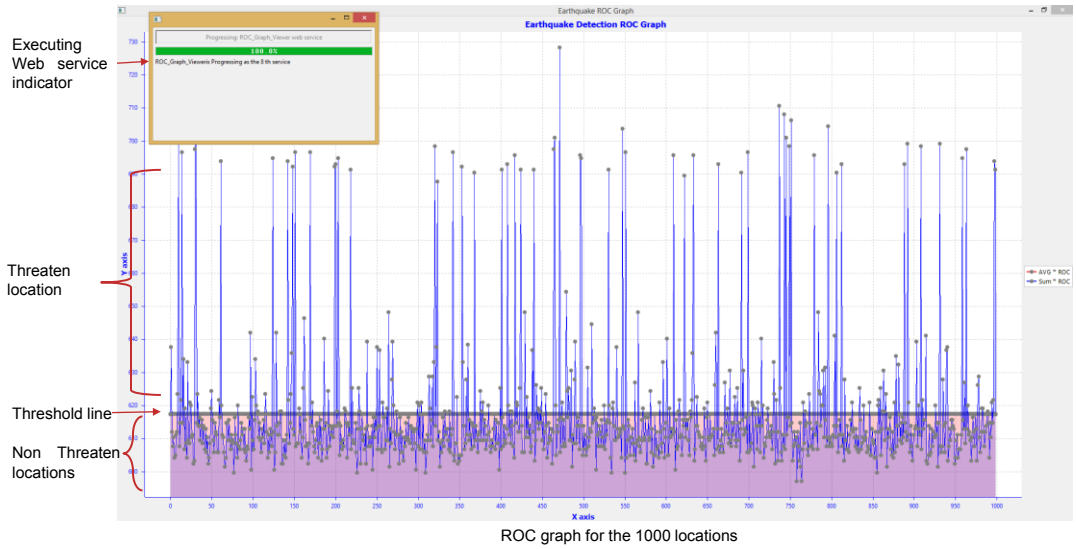


Figure 8.8: ROC graph and status of the execution stage initiator

According to that, user can select relevant analytical requirement and continue the BDA process based on the ASC. If it needs more features for the data mining in addition to the given features, then it should add relevant feature under one of four stages of the CRISP-DM, define reasoning behind the abstract workflow generation and web services to satisfy that functional requirements.

Chapter 9 Experiments and Evaluation

In this chapter, we proceed evaluation on proposed approaches to analyze the effect of main three stages of the research. Chapter 9.1 evaluates the proposed BDA architecture, chapter 9.2 evaluates proposed method to achieve the Planning Stage and Chapter 9.3 evaluates proposed both approaches to achieve the Discovery Stage of the ASC process. Chapter 9.4 evaluates proposed both approaches to achieve the Selection Stage of the ASC process. Next, Chapter 9.5 evaluates proposed approach to achieve the VR Stage of the ASC process. Finally, Chapter 9.6 evaluates the Execution stage.

9.1 Evaluate the Proposed Architecture for Intelligent BDA

Architecture evaluation can do in more stages of the software development process [37]. It can be used to evaluate various attributes, using various methods. There are four types of software architecture evaluation categories have identified.

(1) **Experienced-based** evaluation is based on the previous experience and domain knowledge of developers or consultants,

(2) **Simulation-based** evaluation relies on high level implementation some or all of the components in the architecture,

(3) **Mathematical modelling**, which measures operational quality of requirements [38]

(4) **Scenario-based** evaluation relies particular quality attributes by creating scenarios.

Here, we use scenario-based evaluation category to evaluate our architecture for the Intelligent BDA process.

9.1.1 Experiment Setup

There are three types of key methods in scenario-based have identified. Software Architecture Analysis Method (SAAM), Architecture Trade-off Method (ATAM) and Architecture Level Modifiability Analysis Method (ALMAM). Here we use SAAM to evaluate our three staged architecture's designed and developed for the BDA automation. And it was used experienced-based voting method to asses overall SAAM method.

SAAM consists of six main steps and we have followed our evaluation process according to them [39]. Three types of roles should be involved during the SAAM process. Before starting the main six steps of the SAAM, we have to identify SAAM Roles.

SAAM Roles

We have consulted fifteen industry experts, who have diverse expertization, which are needed in BDA process and used in this architectural process such as BDA, SOA, ASC, AI, Data Mining and Dynamic Workflow automation etc. These roles are involved in the evaluation process,

(a) Member of Evaluation team: Five experts, who are academic, industry or both levels capacities but not specifically data-science domain.

(b) Stakeholders: Five experts, who are academic, industry or both levels capacities and involved in specifically data-science domain.

Step 1 – Develop Scenario

As a brainstorm exercise we have defined following scenario as for our first step.

Scenario 1: ABC airport Company plans to analyze the flight delay data to identify the factors that, led to the past delay. The company hopes to reduce airline delay through such data analytics. BDA engineers work to find the correlation between temperature and flight delay.

This scenario was used as base scenario and variation of this (by changing the

parameters, such as collective ground operational time vs flight delay, Air traffic controllers mishaps vs flight delays etc.) are used as other respective variations of scenarios.

Step 2 – Describe Architectures

We held a session to describe each candidate architectures invented to BDA process and also introduced our requirement by natural-language specification with a summary overview.

Step 3 – Classify and Prioritize Scenarios

Here we identified direct scenario as Scenario 1 and indirect scenarios as variations, which are above mentioned. And levels of candidate architectures as stages of architectural design process. As the stage 1: RA was evaluated based on the given scenario, studied the way of achieve it and benefits and drawbacks of the candidate RA. Next, as of the stage 2: Studied the SA, the way of derived it from the RA. And let them to analyze, whether it satisfied given requirements of scenarios. Finally the high level UML class diagram was analyzed. We did the evaluation to studied ten key quality factors, which are Performance, Reliability, Availability, Security, Modifiability, Portability, Functionality, Variability, Subset-ability and Conceptual integrity.

Step 4 – Individually Evaluate Indirect Scenarios

In the step3, we gave top priority to analyze the direct scenario. Next we gave other variations to further analysis and study the tallying with three stages of candidate architectures. Let them to study above mentioned quality factors.

Step 5 – Assess Scenario Interactions

Next, we asked to evaluate architectures by considering with multiple scenarios vs changes over the components within architectures.

9.1.2 Evaluation

The process of evaluation is laid on step 6 and we have followed following step 6 and received the below result as shown in Figure 9.1.1.

Step 6 – Create an Overall Evaluation

Finally evaluation team marked estimation of success weights of scenarios vs stages

of candidate architectures and continued final report of estimation of quality factors based on their weights. We created a ranked questionnaire of quality factors and produced it to the evaluation team.

Ranks were given as follows their satisfaction of quality factors: 1 for poorly satisfied, 2 for minimally satisfied, 3 for averaged satisfied, 4 for good and 5 for perfectly satisfied. Figure 9.1.1 shows the sorted averaged scores vs quality factors.

According to the above mentioned procedure we conducted evaluation of the architecture. We compared the results of the previous works [11], which conducted evaluation with the same evaluation matrix.

At the beginning we compared, overall weights values between two architectural designs. Fig. 9.1.1 shows latest proposed architectural design scores more quality factors except security, modifiability and availability. Portability to security shows descending ordered weights values with respective to the 2019 results. However due to lack of confidence on security cause, that I explained the proposed system does not provide specific security improvement other than Hadoop does. However, it is strongly recommended to care for security improvements when it releases the next version of

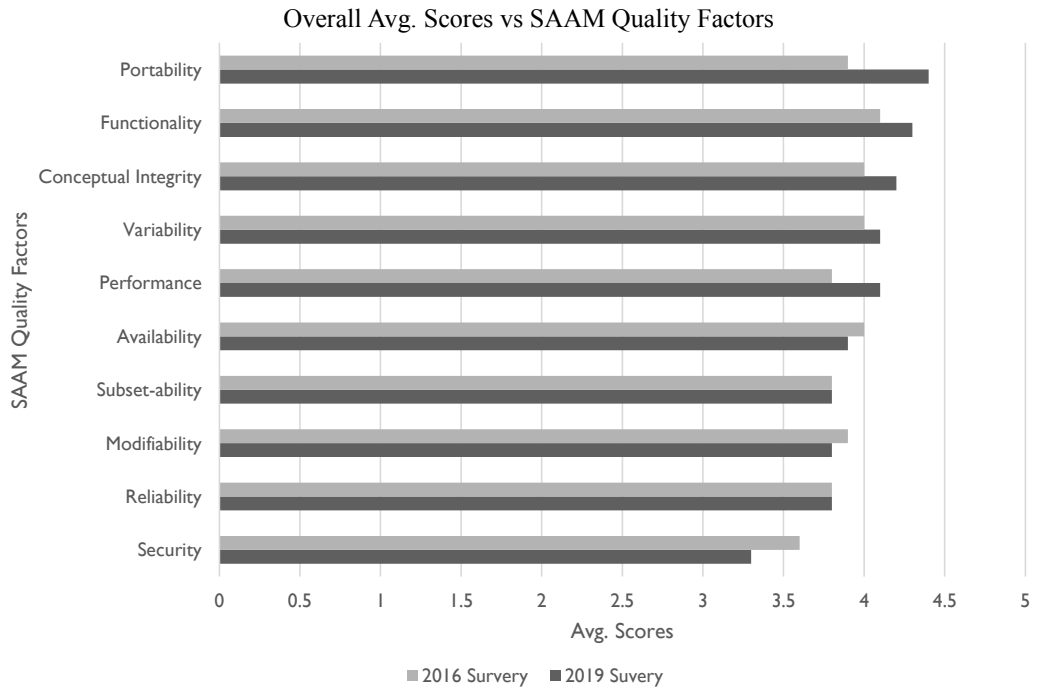


Figure 9.1.1. Overall average scores vs quality factors

the product. In addition to that, all quality factors are scored above the 'Average' satisfaction rate. Rest of all factors are scored beyond the 3.5 except security factor. It is the minimally scored quality factor. That implies architectural design and system release needs to consider more on the security in addition to other quality attributes.

Next we compared proposed architectures with respective to the Stakeholders and

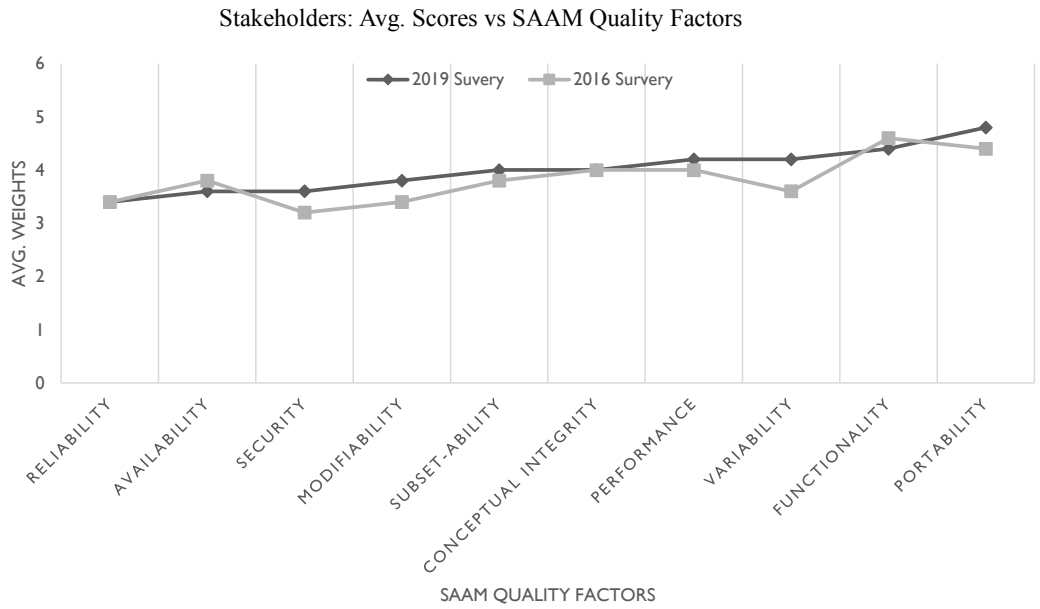


Figure 9.1.2. Stakeholders: Average scores vs quality factors

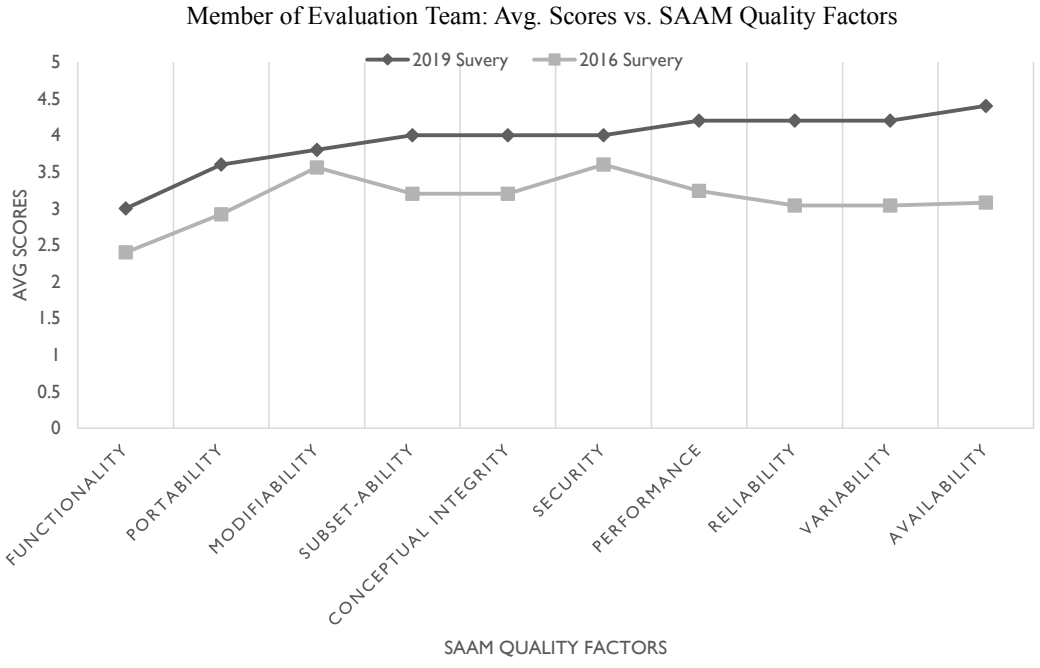


Figure 9.1.3. Overall average scores vs quality factors

SAAM members. Fig. 9.1.2 and 9.1.3 show the results. According to the result member of evaluation team shows more confident on current proposed architecture compared to the previous work. Key improvements of the current proposal are dedicated agent for constraints and QoS management and VR stage of the ASC process. Moreover, Stakeholders shows least confidence on Reliability while Members show least on Functionality. Highest of Stakeholders is the Portability while Members show in Availability. Main cause behind these differences are key difference between their experiences' on data science domain.

9.2 Evaluate the Planning Stage of the ASC

The proposed planner is based on a two-stage process. First it prepares an abstract planner for the BDA process in the planning stage. Next, it prepares a TOP planner, which is called as the concrete planner for BDA in the V&R stage.

Therefore, we organized our evaluation in terms of these two stages. Chapter 9.2 discusses the evaluation of the abstract-workflow planner, which involves an improved GP with related AI planners. Chapter 9.5 discusses the preparation of the TOP planner, which involves a POP that uses the proposed SoPOP and related POP techniques. Throughout the following experiments, we used two evaluation metrics:

1. The efficiency of the proposed methods. This measures the time taken to satisfy the requirements, for various parameter settings. Abstract workflow generation: We evaluated the performance of the methods for the various CRISP-DM stages while increasing the search space of the BDA process.
2. The effectiveness of the proposed methods. This measures the ability to satisfy the requirements for various types of loads. Abstract workflow generation: We evaluated the workflow generation while increasing the number of complex stages and measured the length of the workflow. Next, we studied the behavior in the worst case, using mismatched subgoals and

evaluating the processing time.

9.2.1 Experiment Setup

The experiments were performed on a machine with an Intel Core i7 processor and 16 GB RAM, running Windows 8.1 and Java 1.8. Each test case was executed 100 times, with the average being recorded as the result for that test case. To evaluate the planning stage, we compared the proposed method with a forward-search planning technique [74].

9.2.2 Evaluation

9.2.2.1 Efficiency

To evaluate the efficiency of this aspect of the proposed method, we conducted three experiments, measuring the processing time in each case. First, we evaluated the processing time for the four main stages of the BDA process. In the second experiment, we increased the search space. Finally, we increased the number of complex tasks (mutually inclusive, AND, and exclusive OR steps) in the workflow requirement.

Fig. 9.2.1 shows the experimental processing-time results for the various stages of the CRISP-DM. The proposed method clearly outperformed the heuristic planner. As the CRISP-DM progresses, the forward search shows exponential growth while GP maintains a relatively low processing time.

Next, we measured the performance while increasing the number of complex tasks (mutually inclusive, AND, and exclusive OR steps) in the workflow requirement. Fig. 9.2.2 shows the result of the experiments. The proposed method maintained a relatively low derivative, which implies that the proposed method can adapt to more-complex planning requirements better than other method. This is a significant benefit of using GP, confirming that our method converges efficiently for more-complex planning requirements.

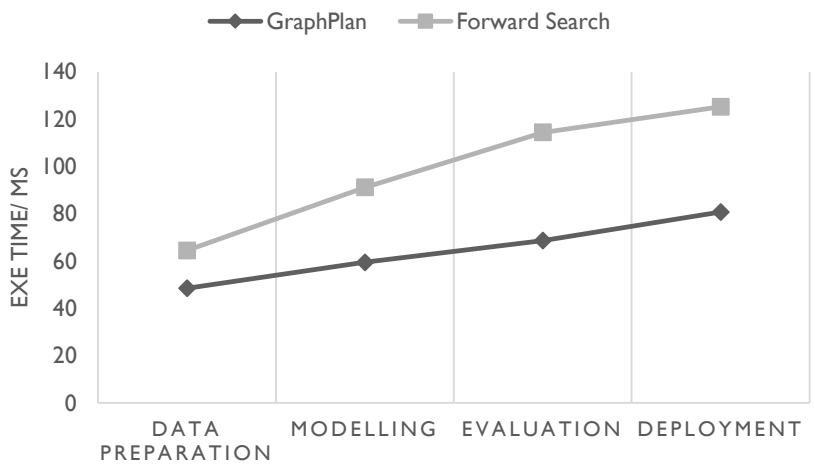


Figure 9.2.1: Processing time for the various CRISP-DM stages

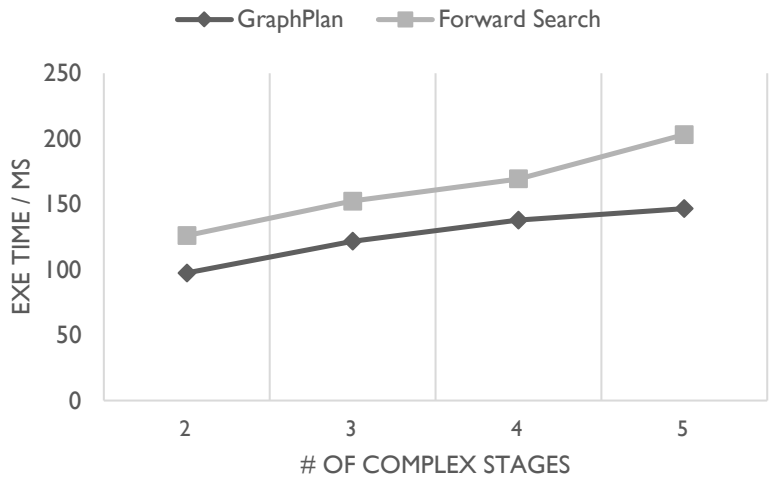


Figure 9.2.2: Processing time vs. number of complex stages

9.2.2.2 Effectiveness

To evaluate the effectiveness of the proposed method, we conducted two experiments. The first finds the shortest effective length of the planner (number of horizontal tasks) for various numbers of complex stages. The second evaluates the performance in the worst case.

In the first effectiveness experiment, a shortened workflow directly affects the makespan. For example, two complex stages and ten horizontal tasks are shown in Fig. 4.3.4. When more parallelizable tasks exist in a given workflow, those tasks can run in parallel in distributed resources instead of running sequentially. Fig. 9.2.3 shows the

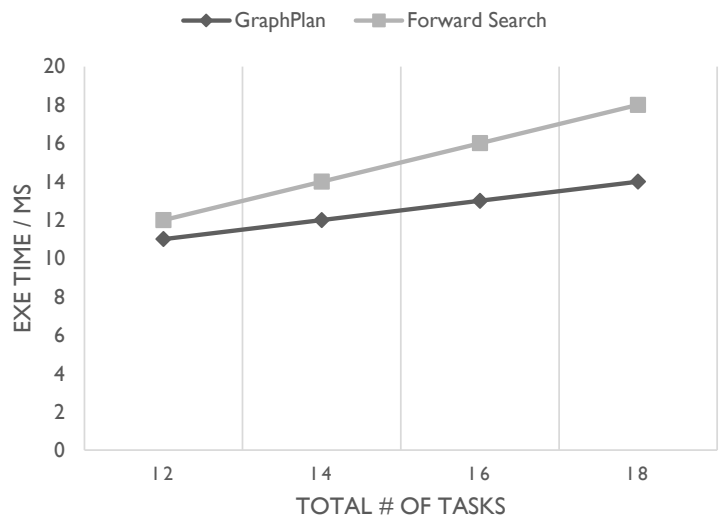


Figure 9.2.3: Effective length vs. number of complex tasks

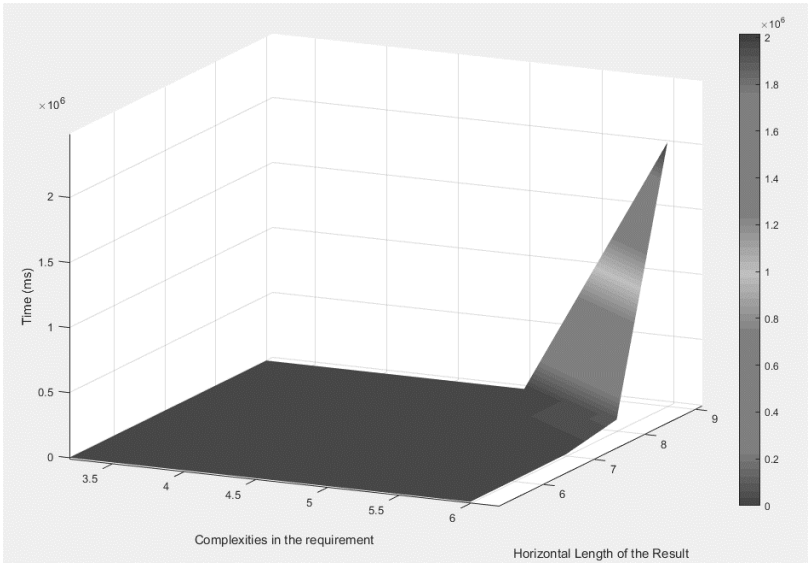


Figure 9.2.4: Performance evaluation vs. number of mismatched mutex

results of this experiment. The basic workflow starts with the workflow shown in Fig. 4.3.4. As the number of complex tasks in the requirement is increased, it increases the number of horizontal workflow tasks. The fig. 9.2.3 shows that our method always maintains the shorter and more effective planner, finding many parallel workflow tasks that can minimize the makespan of the overall process. This finding confirms that our method achieves the goal of BDA plan generation more effectively.

The second effectiveness experiment involved interdependent multiple subgoals. We achieved satisfactory heuristic results through handling these subgoals in most cases. However, in the worst subgoal cases, the heuristic method of the algorithm failed, recording exponential increases in processing time. To observe these cases, we evaluated the planner in terms of the placement of a mismatched mutex relationship with a subgoal, measuring the horizontal length of the result and its processing time. Fig. 9.2.4 shows the exponential growth in the processing time. The x axis represents the mismatched subgoals, the y axis represents the planner length, and the z axis represents the processing time, which was found to follow the relationship $z = (x^y - 1)/(x - 1)$. This implies an exponentially increasing search space, with the algorithm trying to achieve its subgoals via exponential complexity; i.e., implying that mismatched mutex requirements in the planner tend to result in exponential processing times.

9.3 Evaluate the Discovery Stage of the ASC

We have achieved the discovery stage based two perspectives. Which are discovering services precise functional behavior with domain context aware and effective workflow. Then we have proposed Domain Ontology based Service Discovery considering discovering precise functional behavior with domain context aware platform. And also we have proposed Social Service Network with Multiple Feature Attributes based discovery considering the effective workflow. These two aspects are two of most concerns in the discovery domain. In this chapter we discuss evaluation of these two discoveries.

We analyzed the performance of our both approaches by:

a. Evaluating effectiveness of discoveries in terms of discovering Accuracy
b. Evaluating quality of discoveries in terms of Success rate
c. Evaluating efficiency of discovery in terms of discovering Recall rate

In addition to that, we have studied our second approach further in to with its respective parameters,

d. Observing the properties of the GSSN
e. Observing the evolution of the GSSN over Number of links added/ or rewired

9.3.1 Experiment Setup

Implementation Details: Our approaches to the discovery stage address two concerns. Here we continue the evaluation process while measuring the achievements of each perspectives in parallel. We have created BDA domain specific WSDL service registry which contains two hundred services. Same service registry and discovery requirement (workflow, ie collection of abstract tasks) use for both cases. These two discovery processes consist with three main stages. These three stages contain seven steps and those are dealing seven different aspects of the discovery process. The Initial Stage has implemented to initialize and setup ground works of the discovery process. In addition to create the service registry, Domain Ontology based model build its domain ontology and R-GSSN is produced by the next approach. The Clustering Stage key factors to produce fast result and it provides windfall gain to the discovery. Then during the stage 3, it is really happening the discovery process this deals with real aspect of discovering with innovative approaches. Since the Initial stage does not need run always for given set of services. It just one time running requirement and it can re-use it result as many times for distinct requirements (workflow, ie set of abstract tasks). It is another key advantage of the staging and streamline. It is stimulating the effeteness of the discovery in terms of time factor.

Experimental Setup: In parallel we have been developed two other discoveries to compare with our two approaches. Which are Ontology based service discovery and GSSN based service discovery. Then we have been dealing with four types of discoveries to analyze and study gain and losses of proposed two novel approaches. All together, we have been using four semantic discovery methods to the evaluation. We have been using same service registry, discovery requirement as well as same three stages and seven steps to facilitate same environment to all four discoveries. We hope this will help to do the evaluation in fair manner. Since four discoveries are streamlined and once it started the operation from the head to tail and result the outputs of the discoveries. Throughout the experiments, we use the following two evaluation metrics.

1. Precision rate: measures how much accurately find required services. This measures the effectiveness of discovery approaches.
2. Recall rate: measures proportion of the precision. This measures the efficiency of discovery approaches.
3. Balance F measure: To assess the tradeoff between Precision rate and Recall rate. Which helps to study the harmonic mean between effectiveness and efficiency of our approaches.

Higher precision indicates better effectiveness. And Recall rate also varies from 0.0 to 1.0 and that indicates readiness to recognize as standard approaches. And here also 1.0 indicates readiness to the perfect recognition.

To evaluate precision rate and recall rate, we have used 300 BDA related services contained service registry with 30 quarries (abstracts tasks) composed discovery requirements in the BDA domain. The given service registry has 10 different types of services (which are designed with different behavioral signatures to satisfied diversified BDA requirements) to facilitate for a given task.

9.3.2 Evaluation

In this section, we discuss the proposed approaches of the discovery stage.

9.3.2.1 Effectiveness of the Proposed Approaches

Precision rate is one of the best approach to show the effectiveness. We evaluated the effectiveness by increasing the number of services 20 to 300 and number of tasks 2 to 30 in and observe top 10 of the discovered result for a given task. Then we found precision rate of the each task and finally calculated the micro average precision rate for the given set of services. As the number of desired task increases discovering time was also increased. Figure 9.3.1 shows that most precise service discovery result was given by our Domain Ontology based service discovery method. And LSSN based service discovery holds the second among four methods. That means, our approach to precise service discovery to effective service composition have been achieved. LSSN based method is maintaining second among precision calculation due it result not only precise, it considers sociability as well. But it holds second position among rest of two other semantic methods even with that additional constraints (sociability) we have been seeking throughout that process. It can be considered as one of the breakthrough of the LSSN based method.

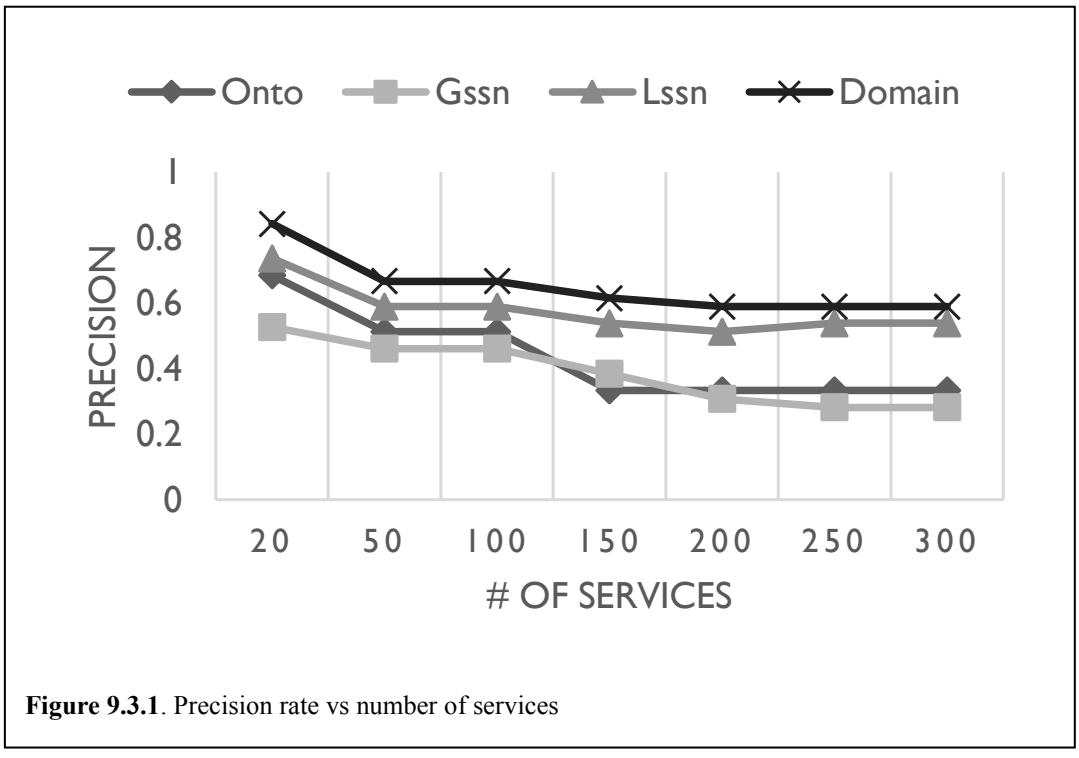


Figure 9.3.1. Precision rate vs number of services

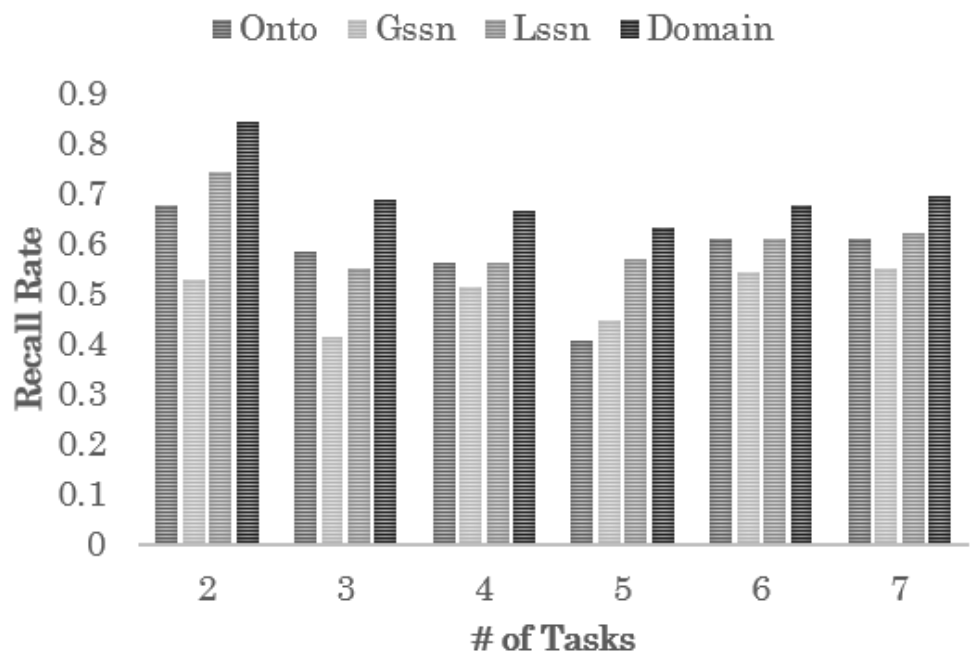


Figure 9.3.2. Recall rate vs number of tasks in the workflow

9.3.2.2. Effectiveness of the Proposed Approaches

Recall rate is one of the best approach to show the efficiency of given approaches. We evaluated the efficiency by increasing the number of tasks 2 to 20 for a given workflow observe top 10 of the discovered result for a given task. Then we found recall rate of the respective workflow and finally calculated the micro average recall rate for the given set of tasks in the workflow.

Figure 9.3.2 shows the recall rate vs number of tasks in the flow. It can clearly seen that our Domain Ontology method holds the most highest recall rate and LSSN based method holds the second among the rest of two other semantic methods. Which implies our two proposed approaches are discovering most efficient discovery result even it increases number tasks in the workflow. Since we have drawn a trend line using our LSSN (lowest among two proposed methods) method and it can be seen that our method holds above 0.6 during even up to 11 tasks in a workflow. That means Domain Ontology is most efficient among four of semantic discoveries and LSSN method holds it second place even with additional constraint (sociability) we have been looking for.

9.3.2.2.Effectiveness of the Proposed Approaches

As the third experiment, it needs to find anomalies of standard we have been founded during the experiments. And also it can be considered as the analysis of the tradeoff between two evaluation metrics, which are Precision rate and Recall rate. We have calculated micro average recall rate for the given set of services of the experiment process. A combine F measure can be considered as the perfect measure to assess the tradeoff between precision and recall rates. Here below shown the equation to calculate combined F measure.

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Here P represents Precision rate and R represents rate. For our evaluation, we used the balance F measure, therefore we substitute α = ½ and β=1 to above equation and it result the following equation,

$$F = 2PR/ (P+R)$$

Table 9.3.1 shows the respective Balance F measure graph of our experiment. It was also given same order of result as of Figure 9.3.1 and 9.3.2 have been resulted. These are given harmonic result. That means, our evaluation results can re-assure and defend that our result as the fair and accurate.

Table 9.3.1. Balanced F Measure of recall rate and precision rate

Service	GSSN	Onto	LSSN	Domain
20	0.526316	0.684211	0.736842	0.842105
50	0.461538	0.512821	0.589744	0.666667
100	0.461538	0.512821	0.589744	0.666667
150	0.384615	0.333333	0.538462	0.615385
200	0.307692	0.333333	0.512821	0.589744
250	0.282051	0.333333	0.538462	0.589744
300	0.282051	0.333333	0.538462	0.589744

According to our result of evaluation process, Domain Ontology based Service discovery is the most precise service discovery method within clan of semantic service discovery methods we have used for our evaluation. Since LSSN based method also performs well in the perspective and maintain good performance during the finding of the precise service discovery of the ASC process.

In addition to that, we evaluated the proposed LSSN with GSSN method to evaluate the social service network perspective to the service discovery. LSSN and GSSN are the methods that we have used for workflow discovery. Therefore, we compute the success rate of the workflow discoveries of LSSN vs GSSN with increasing numbers of services. Fig. 9.3.3 shows the results. We apply thick-dots-lead-

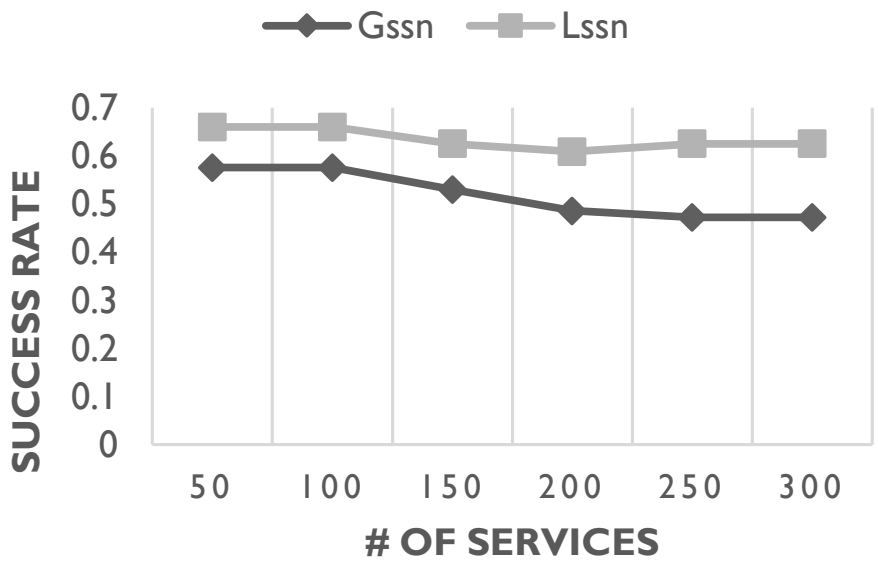


Figure 9.3.3. Success Rate between LSSN vs GSSN

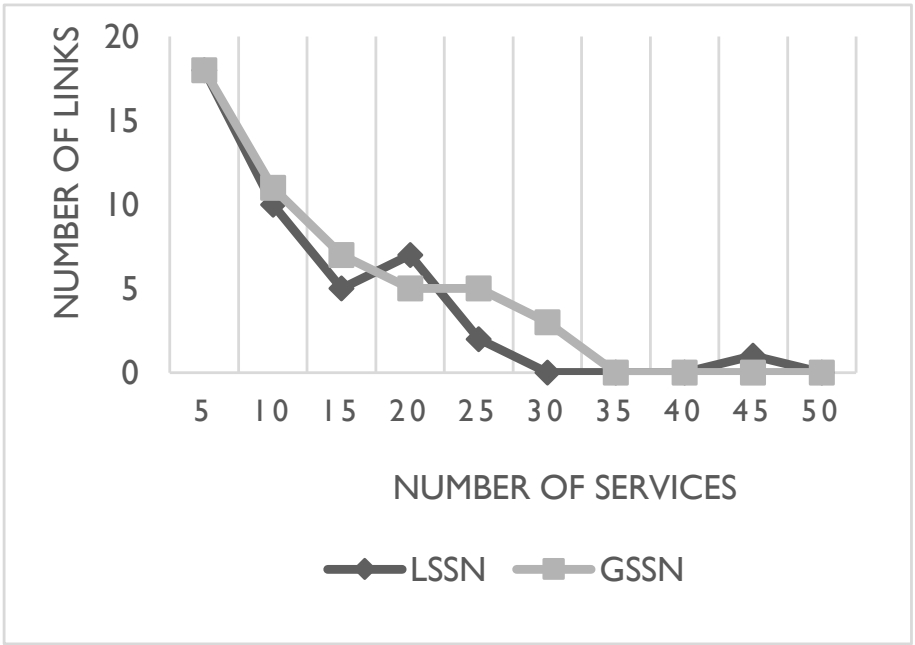


Figure 9.3.4. Scalability between LSSN and GSSN

you in both cases.

When we compare Figures 9.3.1 and 9.3.2, the LSSN method consistently comes second. The perspective behind our proposed domain ontology method and the domain ontology were designed to address only fine-grained concerns. Nevertheless, we considered an additional constraint, the sociability within the LSSN method. Because LSSN is in second place and performing better than the heterogeneous ontology and GSSN methods, even with this additional constraint it performs better than the other two methods. Moreover, it is clear that LSSN takes first place in Figure 10. That means that the workflow perspective is the best method because it has achieved the highest precision between the two workflow discovery methods (LSSN and GSSN) and is also successful over LSSN and GSSN for workflow discovery. It can be considered a breakthrough when we consider the near optimization of conventional workflow discoveries: the LSSN method finds a precise optimum rather than a near optimum.

Finally, we compute the scalability of both approaches. It is one of the most important factors for both GSSN and LSSN, because both work in networks. Fig. 9.3.4 shows that LSSN's link average is less than that of GSSN and it proves that our LSSN maintains good scalability with increasing numbers of services in the network. This is a good sign, because LSSN can provide satisfactory services to peer users for their needs (such as service discovery, recommendation, and composition) than the existing GSSN.

9.4 Evaluate the Selection Stage of the ASC

In this section, we evaluate two methods which are described in Chapter 6.2 and 6.3. Chapter 9.4.1 and 9.4.2 describe evaluation procedure and results of the respective Chapter 6.2 and 6.3.

9.4.1 Evaluate the CTQOS

To evaluate the behavior of our selection approaches, we conducted the extensive experiment on our approach. We focused on the accuracy and the efficiency of our approach. As of our studies in the domain, this is the first approach to QoS criteria's

and custom TS-aware.

For the accuracy evaluation, we have two main perspectives. First, it evaluates the accuracy of the QoS optimization of our proposed method. Next, we have to evaluate the accuracy of custom level methods. To simulate the evaluation of the first user case, we conducted an evaluation of our approach with a leading swarm intelligent multivariate optimization algorithm, which is the D. Karbagoda and B. Basturk proposed artificial bee colony algorithm (ABC) [108]. It is one of the most cited AI based multivariate optimizers. We prepare multivariate QoS based selection approach based on D. Karbagoda proposed method and conduct the overall accuracy of QoS optimization (scenario 5) of the proposed approach. For the second user case, we conduct each custom levels L1, L2 and L3 then find their average and compare their average with L4 to evaluate their Transaction and QoS aware ability.

For the efficiency evaluation, we consider evaluating the processing time. For that, at first, we conduct L5 method with ABC to show the performance of the comparative methods. Moreover, to evaluate the internal efficiency, we conduct above-mentioned second test case with changing a number of tasks and candidate services in the workflow. Evaluate Proposed Selection method in Big Data space

9.4.1.1 Experiment Setup

We used our experiment platform as Intel Core i7, Windows 8.1, 16GB RAM computer and Java 1.8 Enterprise edition. We used the 1000 services information that we already used in the previous method [79]. We used Fig. 6.1.1 shown scenario as of our workflow pattern for the evaluation. In the algorithms, for ABC, we set run time as 100, colony size to the two times of a number of plans, number food sources is equal to the number of plans, a number of employee bee and onlookers are equal to the colony size, one scout bee and 100 as the max cycle. We found these are optimal for the multivariate optimization for our selection. For our GA method, we use 100 as of initial population; tournament size 50 to 100, and mutation rate is 0.01. We conducted five times for each test cases and use the average as that the result of the particular test case.

9.4.1.2 Evaluation

9.4.1.2.1 Accuracy of the Proposed Approaches

Here we focused on the evaluate the overall accuracy of the QoS criteria's except compensability of the proposed method. According to the level information, it's the L5. We compare the L5 with the ABC method as shown in Fig. 9.4.1. To find the accuracy of the result, we measure deviation from the pre-identified global solution of multivariate QoS criteria's. For that, first, we set the candidate services according to the ascending order of values. Next, we calculate the error of deviation from global solution by measuring the deviation from each resulted candidate service from given approach and aggregate each error deviations. It results in a percentage of total error deviation. This reduced by 100 and find accuracy. Fig. 9.4.1 indicates proposed method outperform the ABC method by accuracy. It clearly shows that our method is performing very well in the multivariate optimization of the QoS criteria's.

Next, we compared the average of L1, L2, and L3 vs. L4. Here L1, L2, and L3 represent critical stages, and we simulated scenario 1, 2 and 3 described in Chapter 6.2. We maintained two critical stages for each scenario for L1, L2, and L3 user cases. Moreover, take an average of them to make a single representation of them. As we measure the accuracy of the above method here, also we find the accuracy with the pre-identified global solution of TS and QoS awareness. Next, it calculates the error deviation and finally gets the accuracy. Fig. 9.4.2 shows accuracy comparison of this user case. It shows the accuracy of compensability and QoS between L1, L2, and L3 vs. L4. It shows the highest accuracy scored by L4 mode. That means when it is increasing number of critical stages of the workflow, our proposed method doing better. It ensures the proposed method perform better to find global optimal when it perform with more critical stages.

9.4.1.2.2 Efficiency of the Proposed Approaches

To assess the processing efficiency of the proposed method, we conduct an above-mentioned 1st experiment and find the execution time. For the next experiment, we consider L1, L2, and L3 one instance. Then we compare the execution time of that with

L4 and L5 by changing the number tasks and candidate services. Respective Fig. 9.4.3, 9.4.4 and 9.4.5 shown results in those user cases.

Fig. 9.4.3 shows the ABC with combined L1, L2, L3 vs. L4, vs. L5 while increasing number candidate services with fixed number tasks. All L test results laid on each other. It shows significantly lowest maintained in all L test cases while increasing candidate services. However, ABC shows the exponential growth and significant gap between all L cases. That means, our proposed method remain almost unchanged. This caused; because it remains unchanged, the number of tasks in the workflow, population size and number of the evolution of the proposed GA based method. On the other side, a number of candidate services for each task is the main reason to affect the execution time of the ABC.

Fig. 9.4.4 shows the combined L1, L2, and L3, vs. L4 vs. L5 with fixed number of tasks and increasing number of services. All remained lower execution time. However, L4 shows the highest execution time. Moreover, L5 shows lowest possible. This caused; L4 scenario, it needs to find the services, which are satisfying composition rules 1 and 2 to all of the tasks to make a complete individual from existing services. However, the L5 case does not have such a constraints to make the individual genomes. Therefore, it shows lowest possible execution. L1, L2, and L3 have to limit 2 critical stages to satisfy their TS-awareness during preparing individuals. Therefore, it maintains lower execution time compared to the L4 and higher than L5.

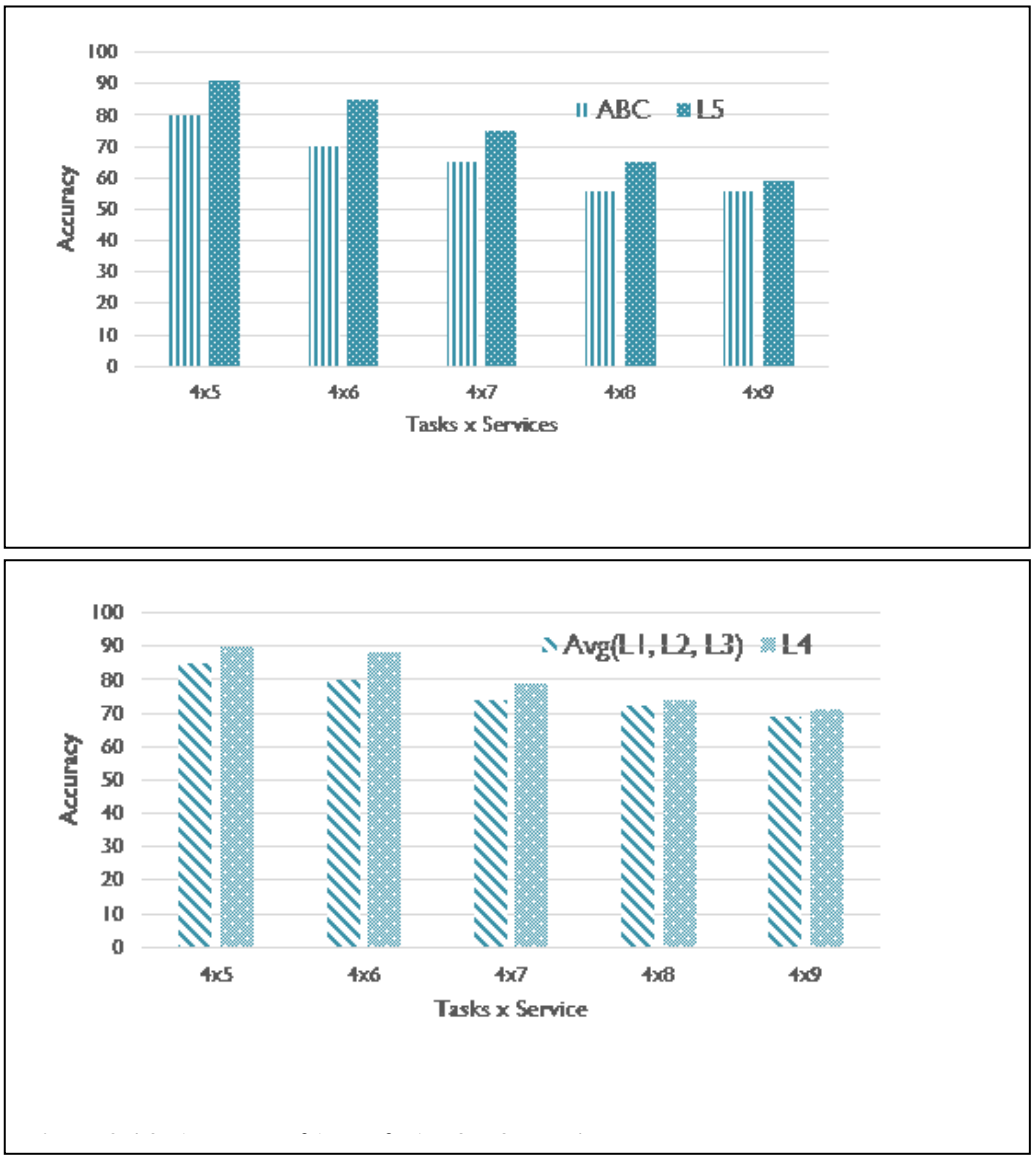


Figure 9.4.3: Processing Time of ABC vs. avg. of L1, L2, L3 vs L4 vs. L5 Increasing

Fig. 9.4.5 shows above experiments with fixed number of services and increasing

number of tasks. It gives same results as shown in Fig. 9.4.4. However, Fig. 9.4.4 maintains convex shape while Fig. 9.4.5 has concave shapes in combined L1, L2,L3, and L4. That means Fig. 9 has higher derivatives of the derivatives than Fig. 9.4.4. This implies number tasks are more effective to increase execution time than candidate services. According to the above experiments results, CTQS is more effective and efficient in the perspective of accuracy and processing time. And, it works better to find the global optimal in the BDA automation based on ASC process.

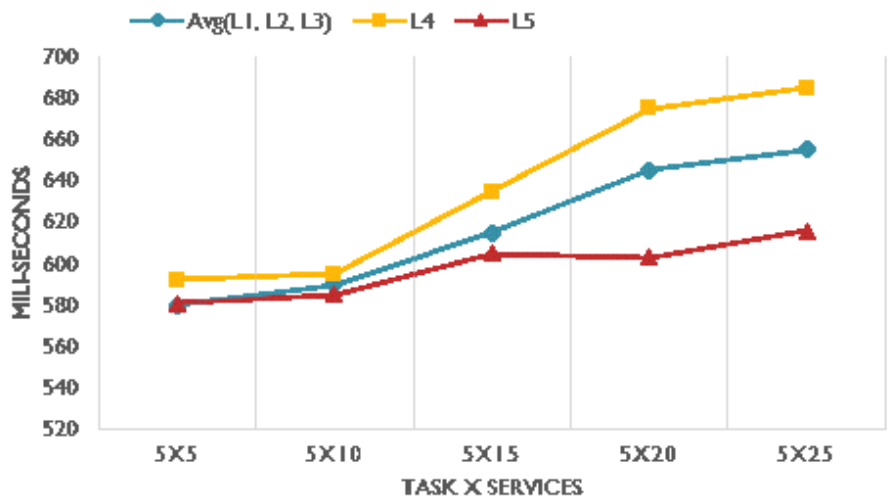


Figure 9.4.4: Processing Time of avg of L1, L2, L3 vs L4 vs. L5 Increasing Number of

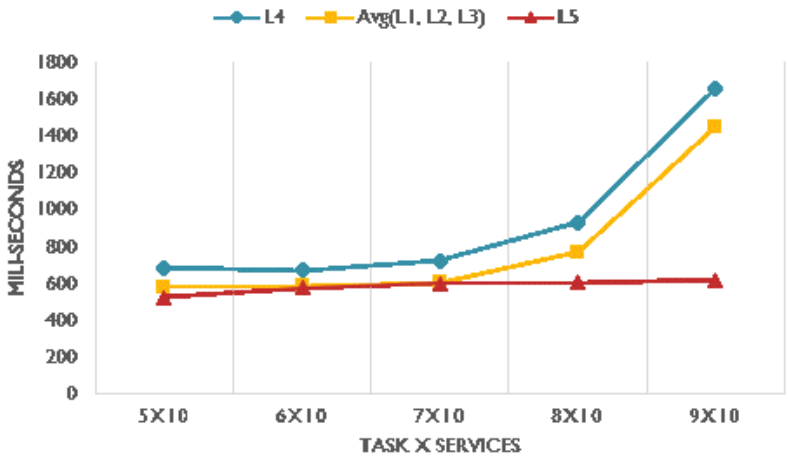


Figure 9.4.5: Processing Time of avg of L1, L2, L3 vs L4 Increasing Number of Tasks for

9.4.2 Evaluate the Selection method in Big Data space

We conducted experiments to evaluate the proposed method. We considered two key research metrics, namely the efficiency of the proposed method and the effectiveness of the heterogeneous-selection approaches in the Big Data space. We, therefore, performed experiments in these two main areas.

Efficiency: To evaluate the efficiency, we conducted experiments that observed the internal, external, and jointly optimized traffic efficiencies of multiobjective selection methods while increasing the number of plans and data nodes in the Hadoop cluster.

Effectiveness: To evaluate the effectiveness, we conducted experiments that observed, in terms of computational complexity, the internal, external, and jointly optimized traffic effectiveness of multiobjective selection methods while increasing the number of plans and data nodes. In particular, we observed the precision of the proposed methods.

Evaluation metrics: Our aim was to evaluate the efficiency and effectiveness of the internal, external, and joint traffic optimizations of multiobjective selection methods in a Hadoop environment. To achieve this, we defined four modes (Mode 1, Mode 2, Mode 3, and Mode 4) and their respective computational complexities (P, Q, R, and S) as shown in Eq. 30, Eq. 31, Eq. 32, and Eq. 33.

Next, based on the computational complexities of Eq. 30, Eq. 31, Eq. 32, and Eq. 33, we defined three traffic metrics C_{Int}, C_{Ext} , and C_{Joint} in Eq. 34, Eq. 35, and Eq. 36.

To calculate these measures, we executed multiobjective selection methods for the various modes by increasing the number of plans while increasing the number of nodes of the Hadoop environment. We maintained a default mode (Mode 1) with minimal replica and block distribution and no IMR agent. Modes 2, 3, and 4 were devised to satisfy particular traffic solutions as follows.

In Chapter 6.3, we proposed two approaches to address the key external traffic congestions occurring during selection. We integrated these two methods to conduct experiments that evaluated the efficiency of the proposed method under multiobjective selection approaches. First, we sorted the WSs based on their utility QoS values and partitioned the dataset into n sets. We then took the average QoS of each batch and

found their proportional values. We set the default mode as one and the remaining batches according to their proportional values. This is simulated in the QSD rule defined in Chapter 6.3. Next, we multiplied the respective proportional values by m to simulate the TRD rule, also defined Chapter 6.3. We then observed the traffic-aware replica distribution across the network (here, m and n are positive integers). According to these results, we distributed services and their replicas across the HDFS. We conducted experiments in Mode 3 and 4 with external traffic solutions to obtain the respective metrics.

In Chapter 6.3, we proposed an IMR agent approach to address the key internal traffic congestion occurring during selection. We conducted experiments in Modes 2 and 3 with an IMR agent to obtain the respective metrics.

$$P = \textit{Computational complexity of Mode 1 (Default): No external and internal traffic solutions are applied} \tag{30}$$

$$Q = \textit{Computational complexity of Mode 2: Internal solution is applied but external traffic solutions are not applied} \tag{31}$$

$$R = \textit{Computational complexity of Mode 3: External solutions are applied but internal traffic solution is not applied.} \tag{32}$$

$$S = \textit{Computational complexity of Mode 4: External solutions and internal traffic solutions are applied.} \tag{33}$$

$$C_{Int} = R - S \tag{34}$$

$$C_{Ext} = Q - S \tag{35}$$

$$C_{Joint} = P - S \tag{36}$$

$$E(C_{Int}) = C_{Int}/R \tag{37}$$

$$E(C_{Ext}) = C_{Ext}/Q \tag{38}$$

$$E(C_{Joint}) = C_{Joint}/P \tag{39}$$

9.4.2.1 Experiment Setup

The experiments were conducted in CentOs 7, with Hadoop 2.2 and Java 1.8 installed on a four-node Hadoop cluster. The master node contained an Intel Core i7 3.4-GHz 8-core processor and 8 GB RAM. Each data node contained an Intel Core i5 3.0-GHz 4-core processor and 8 GB RAM.

Dataset: We conducted our experiments with a real-world dataset provided by Hameda et al. [123] called the QWS dataset, which contained 2500 items of real service information for 10 types of QoS data. For our testing purposes, we use two negatively affecting QoS criteria (response time, and latency) and three positively affecting QoS criteria (availability, throughput, and reliability). We repeated each test case 30 times to obtain the average values for that test case. We prepared 2,000,000 to 10,000,000 planners in a Big Data environment from the QWS data.

9.4.2.2 Evaluation

9.4.2.2.1 Efficiency

We calculated the efficiency of the proposed internal, external. and jointly optimized methods as $E(T_{Internal})$, $E(T_{External})$, and $E(T_{joint})$, respectively, based on the computational complexities of four modes (Eq. 30, Eq. 31, Eq. 32, and Eq. 33) and three basic traffic metrics (Eq. 34, Eq. 35, and Eq. 36).

A. Internal Traffic Efficiency

We considered the internal traffic efficiency of the three selection methods, namely Dijkstra, 0-1 MCKP, and ABC. We calculated the internal traffic efficiencies using Eq. 37 above. Tables 9.4.1, 9.4.2, and 9.4.3 give results for the internal traffic efficiencies for each method. Here, M represents the number of plans, in millions. Fig. 9.4.6 shows the processing costs for Mode 1. This can be used as a reference for the processing costs for the various methods in the Hadoop environment.

For all three methods, the efficiency is suddenly reduced as the number of plans changes from 2M to 4M. The number of plans are increased by changing the number of mappers. For 2M, a single mapper is used. For 4M, two mappers are used. If a single

Table 9.4.1: Internal traffic efficiencies for the 0-1 MCKP method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	21%	19%	9%	14%	15%
3	25%	20%	14%	12%	19%
4	27%	20%	18%	25%	28%

Table 9.4.2: Internal traffic efficiencies for the ABC method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	10%	8%	13%	14%	16%
3	13%	11%	7%	8%	15%
4	17%	9%	13%	17%	21%

Table 9.4.3: Internal traffic efficiencies for the Dijkstra method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	21%	19%	9%	14%	15%
3	25%	20%	14%	12%	19%
4	27%	20%	18%	25%	28%

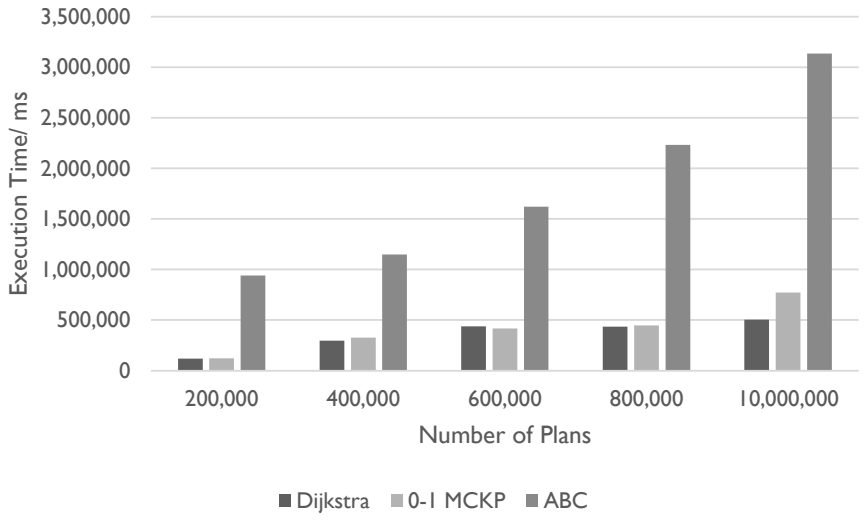


Figure 9.4.6. 'P' values for two data nodes of the Hadoop cluster

mapper is used, the internal efficiency will be higher than when two mappers are used

because using two mappers will generate more internal traffic. That is, increasing the number of plans will increase the number of mappers, thereby increasing the internal traffic. In our experiments, we gradually increased the number of mappers and the number of data nodes in the selection process. The internal traffic should increase when increasing the number of mappers because of the increased cross shuffling in the selection process. However, by using an IMR agent, the internal traffic efficiency is increased, from two mappers upwards. That is, as the internal traffic of the selection process increases, the IMR agent causes an increase in the internal traffic efficiency of the selection process. We found that, across the two-node, three-node, and four-node modes, the 0-1 MCKP method improved from an average 16% internal traffic efficiency to 23%. For Dijkstra, it was from 10.8% to 11.2%. For ABC, it was from 10.8% to 15.6%.

The 0-1 MCKP method demonstrated a higher traffic efficiency than the Dijkstra and ABC methods. According to our investigation of the internal data used in the shuffling stage, 0-1 MCKP generated more internal data than the other two methods. This implies that the IMR agent works more effectively when increasing the internal data of the process. More generally, the proposed IMR agent works efficiently when increasing the number of data nodes, the number of mappers, and the internal data in the selection process.

B. External Traffic Efficiency

We now consider the external traffic efficiency for the three selection methods. We calculated the external traffic efficiencies using Eq. 38 above. Tables 9.4.4, 9.4.5, and 9.4.6 give results for the external traffic efficiencies for the Dijkstra, 0-1 MCKP, and ABC methods, respectively. Again, M represents the number of plans, in millions.

As for internal traffic efficiency, a sudden reduction in efficiency occurs when the second mapper is introduced, as the number of plans increases from 2M to 4M. However, efficiency is increased as the number of mappers is further increased. In our experiments, we gradually increased the number of mappers and the number of data nodes in the selection process. This means that the popularity of the service data of the previous user case and the test case will directly affect the following test case. According to our hypothesis, popularity should be proportional to the hotness of the service data, which implies that increasing the popularity of service data should reduce external traffic. We observed that the proposed rules for external traffic successfully

Table 9.4.4:. External traffic efficiencies for the ABC method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	22%	18%	17%	21%	28%
3	27%	25%	21%	26%	34%
4	33%	27%	21%	32%	39%

Table 9.4.5:. External traffic efficiencies for the 0-1 MCKP method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	22%	18%	17%	21%	28%
3	27%	25%	21%	26%	34%
4	33%	27%	21%	32%	39%

Table 9.4.6:. External traffic efficiencies for the Dijkstra method

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	8%	7%	6%	14%	17%
3	21%	14%	12%	21%	26%
4	19%	18%	18%	26%	28%

reduced the overall traffic for the selection process. We found that, across the two-node,

three-node, and four-node modes, the ABC method improved from an average 21.2% external traffic efficiency to 30.4%. For 0-1 MCKP, it was from 12.2% to 22.6%. For Dijkstra, it was from 10.4% to 21.6%.

The ABC method demonstrated a higher traffic efficiency than the other two methods. According to Fig. 9.4.6, ABC also has the highest processing cost. This implies that ABC should have the highest external traffic. Therefore, the proposed method for external traffic efficiency works best for increased external traffic in the selection process.

Table 9.4.7: Jointly optimized traffic efficiencies for 0-1 MCKP

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	23%	19%	8%	14%	27%
3	38%	31%	21%	17%	36%
4	43%	28%	24%	37%	49%

Table 9.4.8: Jointly optimized traffic efficiencies for ABC

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6M	8M	10M
2	19%	13%	17%	22%	31%
3	27%	23%	15%	21%	36%
4	37%	23%	22%	36%	40%

Table 9.4.9:. Jointly optimized traffic efficiencies for Dijkstra

Number of Nodes	Traffic efficiency with respect to the number of plans				
	2M	4M	6NM	8M	10M
2	11%	14%	10%	22%	24%
3	14%	13%	12%	18%	22%
4	17%	16%	16%	21%	28%

C. Joint Optimization of Traffic Efficiency

Finally, we consider the efficiency of joint optimization of the internal and external traffic for the three selection methods. We calculated the jointly optimized traffic efficiencies using Eq. 39 above. Tables 9.4.7, 9.4.8, and 9.4.9 give results for the jointly optimized efficiencies for the three methods while increasing the number of plans and the data nodes in the Hadoop environment. Again, M represents the number of plans, in millions.

We found that, across the two-node, three-node, and four-node modes, the 0-1 MCKP method improved from an average 18.2% jointly optimized traffic efficiency to 36.2%. For ABC, it was from 20.4% to 31.6%. For Dijkstra, it was from 15.8% to 19.6%.

The 0-1 MCKP method demonstrated the highest joint optimization among the methods, with ABC being next best. The 0-1 MCKP method could achieve 49% efficiency by joint optimization when the number of data nodes was increased to four. For ABC, 40% could be achieved and Dijkstra could achieve a maximum of 28%. The selection process usually reduces the efficiency of joint optimization when the process introduces a second mapper. However, the process increases efficiency as the number of plans in the process increases. This implies that the proposed method for jointly optimized traffic-aware selection can work efficiently.

9.4.2.2.2 Effectiveness

For this metric, we mainly considered the computational complexities and the precision of the three methods in the Hadoop space.

A. Effectiveness of the Traffic Solutions

We calculated the effectiveness of the respective internal (C_{Int}) , external (C_{Ext}), and jointly optimized (C_{Joint}) methods by taking the average values for the various numbers of plans, (2 M, 4 M, 6 M, 8 M, and 10 M) in all three test cases (2, 3, and 4 data nodes). For example, we consider 2 million plans for all three data-node numbers in calculating

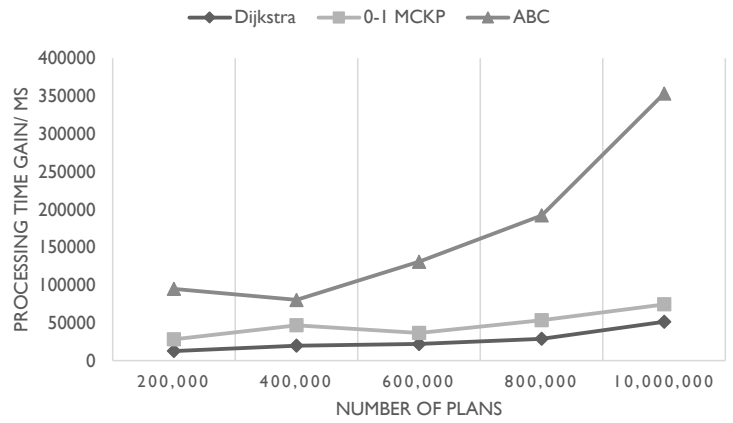


Figure 9.4.7: Average processing time gain by internal traffic solution for multiobjective selections in a multinode Hadoop cluster

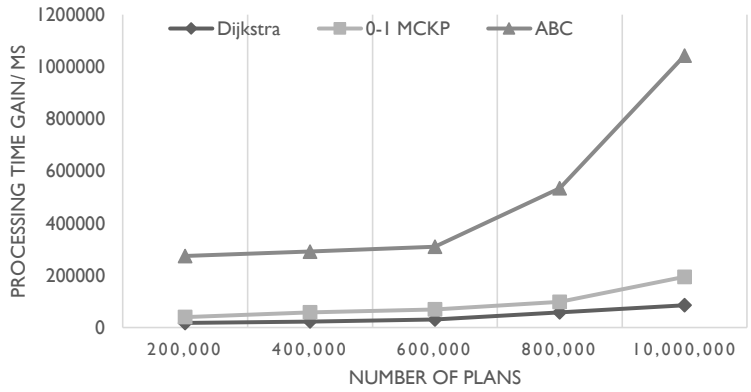


Figure 9.4.8: Average processing time gain by external traffic solution for multi-objective selections in a multi-node Hadoop cluster

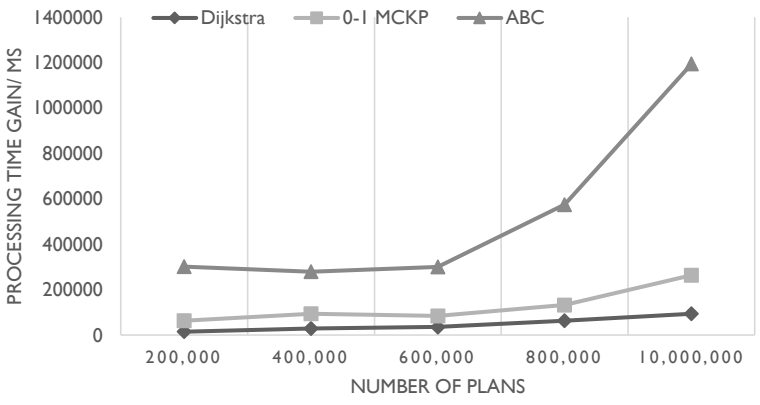


Figure 9.4.9: Average processing time gain by joint traffic solution in multiobjective selections in the multinode Hadoop cluster

the respective C_{Int} as $C_{Int,a}$, $C_{Int,b}$, and $C_{Int,c}$ values for ABC. We then specify the average

traffic effectiveness for ABC in Hadoop as ($C_{Int,a}$ + $C_{Int,b}$ + $C_{Int,c}$)/3. Likewise, we calculate the average values for all numbers of plans for the three selection methods.

The results are shown in Figs. 9.4.7, 9.4.8, and 9.4.9 with respect to their internal, external, and jointly optimized traffic costs. All three graphs maintain the same pattern with only slight deviations. The average line graph for ABC is concave in shape, whereas 0-1 MCKP and Dijkstra maintained a nearly linear relationship with execution time. This means that the derivative of the ABC line graph increases with an increasing number of plans, but the other two methods do not show such an exponential increment. That is, internal, external, and jointly optimized ABC traffic effectiveness for the ABC method increases exponentially, whereas the increase is linear for the other two methods. This is because the ABC method has a multiobjective selection requirement, in contrast to combinatorial and linear selection requirements. Numbers of service plans are directly affected to increase traffic in all three methods.

The 0-1 MCKP method maintained relatively high traffic costs for the various traffic types. This is because 0-1 MCKP has to solve combinatorial selection requirements, unlike the Dijkstra method. This implies that the linear optimal selection requirement solved by the Dijkstra method causes it to generate the lowest possible traffic cost among the three methods.

These three graphs plotted the caused and solved the average traffic for the various selection methods, implying that the proposed traffic solutions work effectively with respect to the various traffic concerns occurring with the various methods.

B. Effectiveness in Precision

In Chapter 6.3, we proposed a threshold plan for each batch, which is an approach that addresses the precision of the proposed method by reducing the search space and segmenting it in terms of batchwise content. To evaluate the proposed method from the perspective of the efficiency of the results, we conducted two different tests with two data sets.

The Dijkstra and 0-1 MCKP methods always result in a globally optimal solution. Therefore, we needed to consider only the ABC method to measure the precision of the

composition plan for multivariate optimization. We conducted experiments using the

$$D_{sum} = \sum_{i}^{n} Deviated\ Result \quad (40)$$

$$Average\ Precision = {}^{100 - D_{sum}}/_{n} \quad (41)$$

QWS dataset under various types of candidate services to find the average precision of the given result results for the ABC algorithm. We found the respective error deviations, as shown in Eq. 40 and calculated the average precision, as shown in Eq. 41. Here, D_{sum} is the deviation from the ascending ordered result and n is the number of attempts.

Fig. 9.4.10 shows the results with and without the batchwise operator for the QWS data set. It shows that the proposed method consistently maintained a relatively high precision, whereas the alternative would have a drastic reduction in precision. This indicates that the batchwise operator is the main reason for maintaining the effectiveness of the results in a consistent manner.

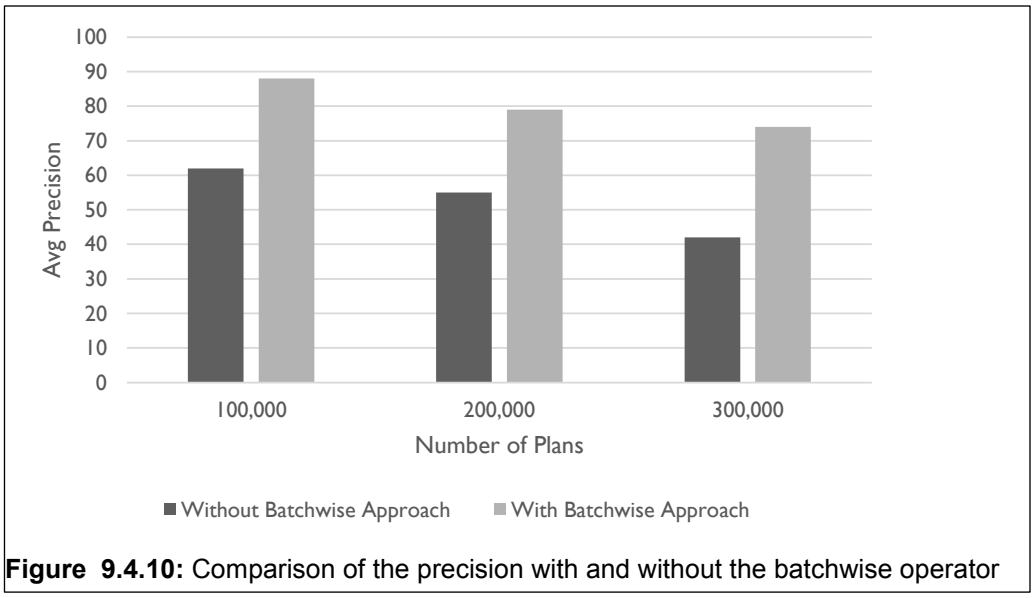


Figure 9.4.10: Comparison of the precision with and without the batchwise operator

9.4.2.2.3 Summary

Summaries of the respective methods are discussed in the following. To find the summarized average values of the respective methods, first we calculate the average values of three rows separately and then calculate the average of the respective three average values of Table 9.4.1 to 9.4.9.

Internal traffic efficiencies of each method are 19% by 0-1 MCKP, 13% by ABC and 11% by Dijkstra. The highest internal traffic efficiency is earned by the 0-1 MCKP and lowest by the ABC. That means, for combinatorial requirement (0-1 MCKP) result in the highest number of intermediate results while multivariate requirement (ABC) has minimal intermediate results during the process. All three methods show relatively low increment value while increasing number of nodes for a particular number of plans and increasing the number of plans for a particular number of nodes.

External traffic efficiencies of each method are 26% by ABC, 18% by 0-1 MCKP and 17% by Dijkstra. The highest external traffic efficiency is earned by the ABC and lowest by the Dijkstra. That means, during the execution of the multivariate algorithm (ABC), process earned the highest external traffic efficiency compared to the other two methods. Relatively ABC has an exponential increase of processing cost and this is caused by the computation complexity of multivariate composition requirement. However, ABC results in more hot data due to the highest computation complexity than the other two methods. In return, ABC results in the highest efficiency as well. This implies, proposed external traffic solution works better while the presence of more hot data in Big Data space. Meantime, Dijkstra has minimal computation complexity compared to the other two methods and it results in relatively lowest efficiency. This means the computation complexity of the objective function shows a roughly proportional relationship to the external traffic.

Joint traffic efficiencies of each method are 28% by 0-1 MCKP, 25% by ABC and 17% by Dijkstra. The highest joint traffic efficiency is earned by the 0-1 MCKP and minimal by the Dijkstra method. Dijkstra has minimal computation complexity, therefore linear optimal algorithm earned minimal joint traffic benefits. However, joint traffic efficiency doesn't show the proportional relationship to the computation complexity of the objective function.

ABC shows the highest traffic effectiveness compared to the other two methods. It maintains a considerable gap between the other two methods. This caused by the computation complexity of the ABC method. Respectively 0-1 MCKP and Dijkstra show relatively low and increment in all three methods (internal, external and joint)

while increasing the number of plans in the environment as shown in Fig. 9.4.7, 9.4.8 and 9.4.9.

According to the Fig 9.4.6, the highest computation cost is shown by the ABC and lowest by the Dijkstra. However, ABC shows exponential growth and the considerable gap in the processing time compared to the other two methods.

9.5 Evaluate the VR Stage of the ASC

The proposed VR stage is mainly based on a two-stage process. First procedure converts the CWS based selection results in to the POP problem, we already discussed as the CPoW. Next, the procedure solves planning problem by preparing a TOP planner using the SoPOP method.

Therefore, we organized our evaluation in terms of these two stages. Chapter 7 discusses the evaluation of the atomic service and CWS based PoW, which involves a POP problem for the SoPOP. Chapter 7 discusses the further evaluate TOP preparation, which involves a POP that uses the proposed SoPOP and related POP technique that feasible to prepare the TOP based on the CWS. Throughout both sets of experiments, we used two evaluation metrics:

1. The **efficiency** of the proposed methods. This measures the efficiency factors in satisfying the requirements, for various parameter settings.
    a. PoW and TOP generation based on service (atomic and CWS) based methods: We evaluated the performance of the methods for the various CRISPDM stages while increasing the search space of the BDA process.
    b. TOP generation compared to the general POP solver: We evaluated the performance of the methods in various CRISP-DM stages while increasing the search space of the BDA process and in flaw-refining efficiency.
2. As The **effectiveness** of the proposed methods. This measures the ability to satisfy the requirements for various types of loads.

a. PoW and TOP generation based on service (atomic and CWS) based methods: We evaluated effectiveness of the TOP using preparation for a given BDA workflow.

b. TOP generation compared to the general POP solver: We evaluated the performance in the presence of flaws in the workflow.

The experiments were performed on a machine with an Intel Core i7 processor and 16 GB RAM, running Windows 8.1 and Java 1.8. Each test case was executed 100 times, with the average being recorded as a result of that test case. The service selection repository was the E. Al-Masri and QH Mamoud service registry [121], as we prepared 100 Web services for BDA. The evaluation scenario was based on Section 2.4 discussed scenario.

9.5.1 Experiment Setup

To evaluate the POP problem, in this manuscript we discussed as the PoW, we manually compared the proposed method with an existing service based POP methods B. Wang et. al. [115], and P. Wang et. al. [116]. We use ScAPoP proposed by B. Wang et. al. and WSPR proposed by P. Wang et. al.

In addition to that, we are one of the very first to propose the CWS based POP. Therefore for further evaluation for the CWS based POP solvers, atomic service based POP solvers such as ScAPoP and WSPR methods did not consider, due such methods are violate the functional and qualitative objectives of the CWS based selection results.

Therefore, we compared the proposed method with a method, which is flexible to generate the POP problem. According to our problem domain, we have to address three types of flaws (open goal, ordering constraint and binding variable constraints) under two main types of flaws (open goal and threat). However, according to our literature review, related recent works which are discussed solving such issues explicitly are scarce. As of our studies, we identified forward search planner (FSP), proposed by Oscar Sapena et. al. [60], as one of most compatible planner for above mentioned flaw-refining. Therefore, we select FSP as one of the related methods in the evaluation. In addition to that, there was no related CWS based PoW to prepare the POP problem for

flaw solving method. Therefore, we prepared the PoW based on CPoW method throughout the evaluation.

9.5.2 Evaluate service oriented perspective

To evaluate the proposed method w.r.t. to the service oriented POP's, we manually compared with ScAPOP and WSPR. We manually prepared respective PoW's and TOP's according to the logical explanation given by the respective methods for the user cases.

9.5.2.1 Efficiency

To evaluate the efficiency of this aspect of the proposed method, we conducted two experiments, measure the deviation of the PoW and TOP from the original workflow resulted in the planning stage. First, we evaluated the deviation of the PoW from the abstract workflow for the four main stages of the BDA process. In the second experiment, we increased the search space. Finally, we increased the number of stages of the CRISP-DM and prepared the TOP. In the related method, the PoW is called the directed graph G(V, E) for the planning problem. V represents the vertices, i.e. task (or services), and E represents the connecting edges between tasks (services). We prepared the PoW's, and TOP's of all three methods based on the theoretical explanation according to the respective literature review's.

Fig. 9.5.1 shows the experimental results for the stages of the CRISP-DM. The proposed method clearly outperformed the atomic web service based PoW generator. As the CRISP-DM progresses, the CPoW maintains zero deviation from the original result resulted in the planning stage. Both of the atomic services based POP solvers generate the respective PoW with respective to the atomic services of the workflow. The scenario discussed in Chapter 7.1, that we used in the evaluation contains two web services under each CWS's and then each web service represents one vertex according to the G(V, E) made by the respective proposed methods. WSPR method unable to cope with complex workflow patterns such as AND, XOR. Therefore, WSPR returns a sequential pattern for the any given composition patterns. Data Preparation stage contained an AND pattern, then WSPR shows additional deviation from the abstract

workflow. Importance of the maintaining minimum deviation from the original workflow reflects when the procedure refining the PoW and at the execution stage of the final result.

Next, we measured the deviation of the TOP while increasing the stages of the CRISP-DM. Fig. 9.5.2 shows the result of the experiments. We included one of each type of flaw (one open goal for one task, both ordering constraint and binding constraint for one task) at each stage of the CRISP-DM. CPoW requires only two compensation task to satisfied flaws occurred at each stage. However, the other two methods need three additional steps to satisfy flaws. Therefore the proposed method maintained a relatively low deviation, while the other two methods show a higher deviation than the proposed method. This implies that the proposed method can adapt to more-complex planning requirements better than atomic service based methods. This is a significant benefit of using CWS based PoW, confirming that our method converges more-complex POP requirements in an efficient manner.

According to these experiments, the proposed CPoW method performed better than the atomic service based method in terms of efficiency of preparing the PoW and the TOP.

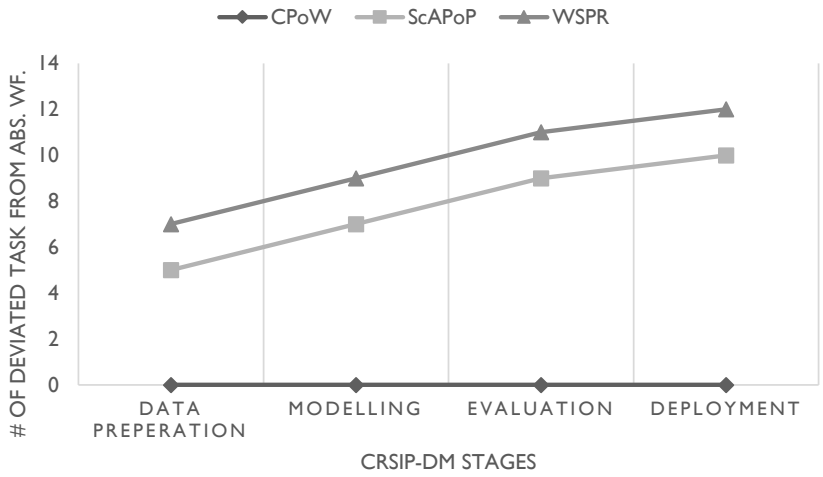


Figure 9.5.1: Horizontal deviation of the PoW compared to the abstract workflow

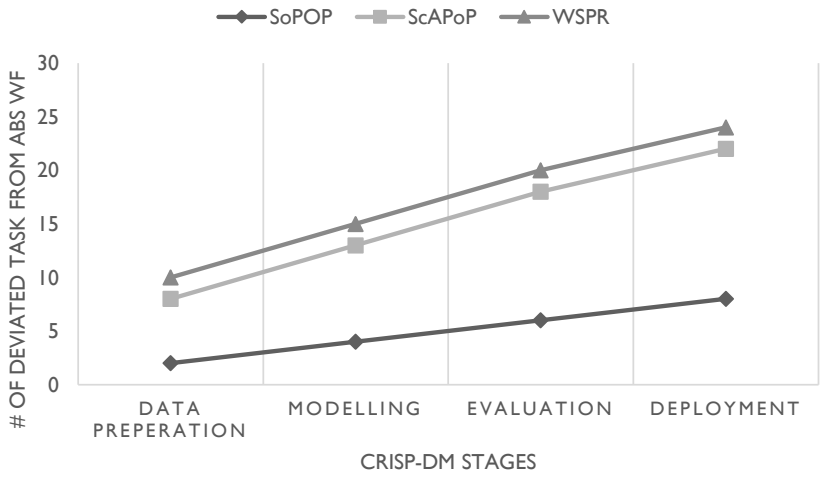
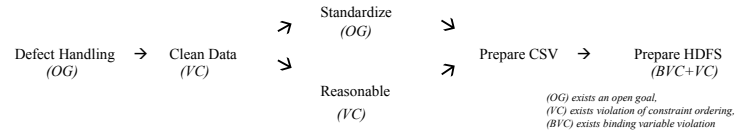


Figure 9.5.2: Horizontal deviation of the TOP compared to the TOP

9.5.2.2 Effectiveness

To evaluate the effectiveness of the proposed SoPOP, we prepared the PoW and TOP for each method. Fig. 9.5.3 shows the flawed scenario made by G_{DP} vs respective PoW and TOP made by WSPR, ScAPoP and SoPOP. The Fig.a represents the flawed scenario made for the evaluation. The selected composition of CWS and atomic services are the same as the scenario discussed in Chapter 7.1. Fig.b, c and d represent the respective PoW's and TOP's prepared according to the WSPR, ScAPoP and proposed method.

According to the Fig. b, c, the WSPR and ScAPoP methods address the POP problem by considering the each of the atomic services as a vertex (task) of the graph. Then these two methods are preparing the PoW, which included 10 tasks for the POP problem. In addition to that, PoW of the WSPR is unable to cope with complex workflow patterns such as AND and XOR contained workflows. WSPR procedure address only as the sequential POP problem. This reason severally affects the overall resource and time consumption of the workflow. Moreover, both methods are refining the flaws by adding refining vertices by considering the issues of atomic web services, not the objectives of CWS's. Then respective procedure may include refining tasks in between CWS's. This



a: Flawed Scenario made by G_{DP} to prepare PoW and TOP

PoW

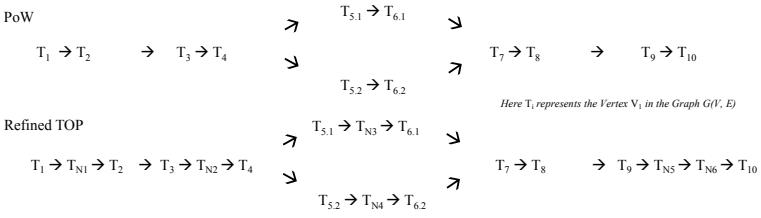
$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow T_6 \rightarrow T_7 \rightarrow T_8 \rightarrow T_7 \rightarrow T_8 \rightarrow T_9 \rightarrow T_{10}$

Here T_i represents the web service (WS) of in the Graph G(WS, E)

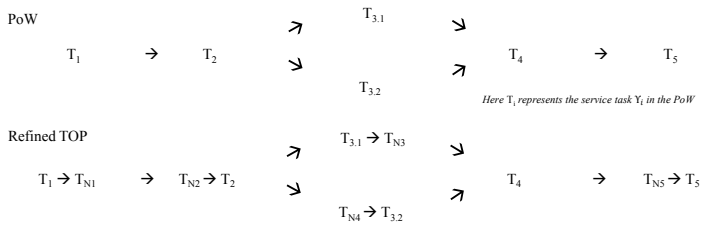
Refined TOP

$T_1 \rightarrow T_{N1} \rightarrow T_2 \rightarrow T_3 \rightarrow T_{N2} \rightarrow T_4 \rightarrow T_5 \rightarrow T_{N3} \rightarrow T_6 \rightarrow T_7 \rightarrow T_{N4} \rightarrow T_8 \rightarrow T_7 \rightarrow T_8 \rightarrow T_9 \rightarrow T_{N5} \rightarrow T_{N6} \rightarrow T_{10}$

b: PoW and refined TOP according to the WSPR method



c: PoW and refined TOP according to the ScAPoP method



d: PoW and refined TOP according to the CPoW and SoPOP

Figure 9.5.3: Flawed scenario made by G_{DP} vs respective PoW, TOP made by WSPR, ScAPoP and SoPOP

may violate the fundamental objectives such functional and qualitative aspects of selecting CWS. Moreover, Fig.d shows that the proposed PoW maintains zero deviation compared to the abstract workflow G_{DP} and TOP included the respective refined task without disrupting the integrity of already selected CWS's.

These experiments confirm that the proposed SoPOP method maintain minimum deviation and provides backend support to prepare the more effective TOP for data analytics. According to these experiments, the proposed method outperforms the existing service oriented method by maintaining minimum deviation from the abstract workflow and effectively shortened workflows with its original workflow pattern.

9.5.3 Evaluate POP perspective

To evaluate the proposed method as a general POP solvers perspective, we compared with Oscar Sapena et. al. proposed method, which is the forward search planner with least commitment strategy, we called as the FSP. To conduct the experiments, we convert the PoW problem in to the FSP readable way. Moreover, FSP proposed for robot user case. Therefore, we adopted FSP to comply with the analytical user case. Then we conducted the experiments as described below.

9.5.3.1 Efficiency

To evaluate the efficiency of the proposed method based on CWS based PoW, we conducted two experiments. The first involved the flaw-refinement efficiency with respective to the execution time. The second measured the flaw refinement efficiency with respective to the horizontal length of the TOP. Each stages of the CRISP-DM compromised a one type of flaw from all types of flaws and one type of logical gates as described logical flaws in Chapter 4.

The first experiment's results are shown in Fig. 9.5.4. The experiment evaluates flaw refinement factor with respective to the flaws in refining time. We define $f_1(VR^t, \mathcal{F}^r)$ as the flaw VR time factor for various numbers of flaws and tasks. Here, VR^t is processing time of the VR the flaws. $n(\mathcal{F}^r)$ is the number of flaws refined. Lowered f_1 means the higher solving efficiency of the flaws.

$$f_1(VR^t, \mathcal{F}^r) = VR^t / n(\mathcal{F}^r) \qquad (21)$$

According to Fig. 9.5.4, f_1 of the both methods maintain considerable gap and slight concave shape. However, the proposed method steadily maintains the lowered value compared to the FSP method. This causes that the VR process of the proposed method solves flaws in more flexibly compared to the FSP method. That means, the proposed method able to VR the flaws in minimum effort while increasing the search space of the POP problem.

The second experiment's results are shown in Fig. 9.5.5. The experiment evaluates the refinement factor with respective to the horizontal length. We define $f_2(\gamma^{TOP}, \mathcal{F}^r)$ as the inverse deviation factor of the refined TOP. Here, $n(\gamma^{TOP})$ is the length of the TOP. Higher f_2 means the TOP contained minimum deviation compared to the abstract workflow.

$$f_2(\mathcal{F}^r, \gamma^{TOP}) = n(\mathcal{F}^r)/n(\gamma^{TOP}) \qquad (22)$$

According to Fig. 9.5.5, the f_2 of the both cases are increasing linearly. However, the proposed method shows higher f_2 while FSP maintains lower. This implies, that the proposed method always prepare the TOP with minimum deviation. This caused, TOP achieved by the minimal number of newly added refined task compared to the FSP method. This shows that the proposed SoPOP shows efficient TOP generation with minimal deviation from the abstract workflow. This benefit caused by that the proposed method refine the both types of threats as conjugate flaw, while the other methods refine them as individually and discard the logical pattern of problem domain.

Therefore these experiments imply, the proposed method offers better flaw refinement factors and this implies SoPOP more efficient in preparing TOP with respective to the general POP solvers. That means, the proposed method well behaved before and after the POP VR process.

9.5.3.2 Effectiveness

To evaluate the effectiveness of the proposed method, we evaluated the VR while increasing the number of flaws and search space (stages of the CRISP-DM process).

Fig. 9.5.6 shows the results of the experiment. It increases the gap between two methods while increasing the search space. And FSP has higher concave shape

compared to the SoPOP. This implies, the derivative of the FSP is higher than the SoPOP. That mean, the proposed SoPOP method is the fastest, and did not slow down significantly while increasing the number of flaws and search space. SoPOP inherits these benefits, due it is refining flaws in sequentially and recall VR as the recursive method. Therefore, SoPOP shows good performance advantage compared to the FSP. This confirms that the proposed method outperforms the other method when increasing the flaws and search space. Based on the above experiment, the proposed SoPOP shows the highest effectiveness with respect to the VR time and horizontal length to generate more effective TOP.

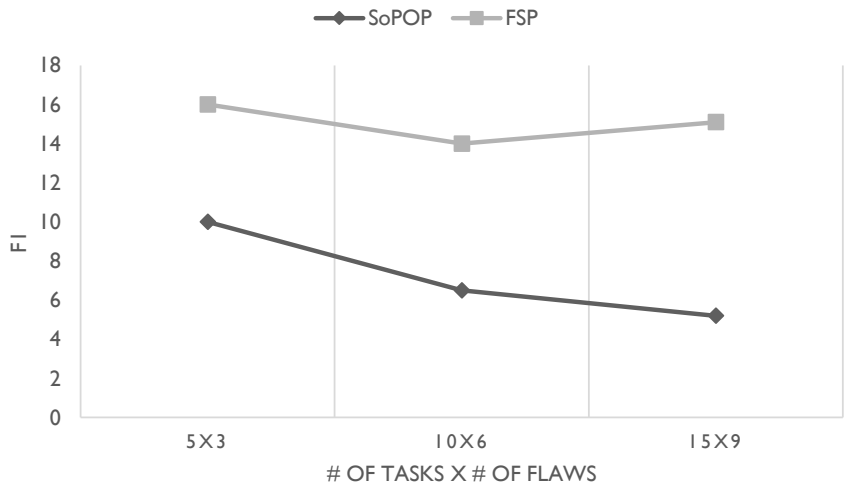


Figure 9.5.4: $f_1(VR^t, \mathcal{F}^r)$ vs number of tasks and flaws

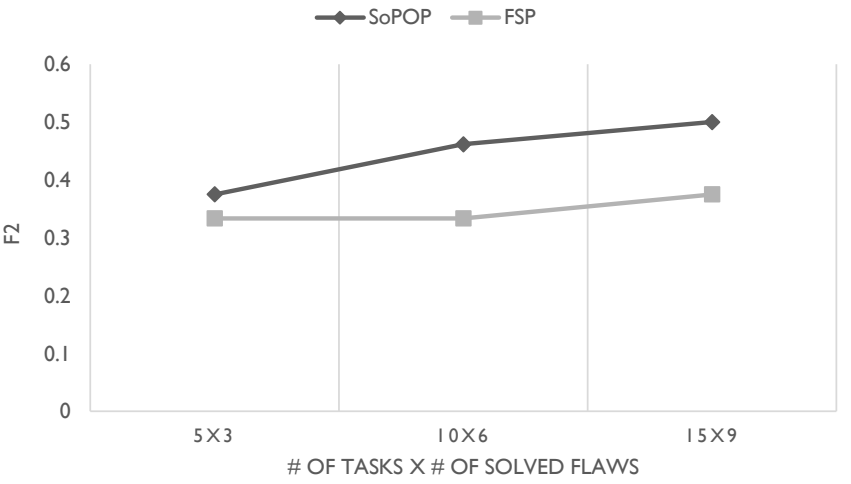


Figure 9.5.5: $f_2(\mathcal{F}^r, \gamma^{TOP})$ vs number of tasks and flaws

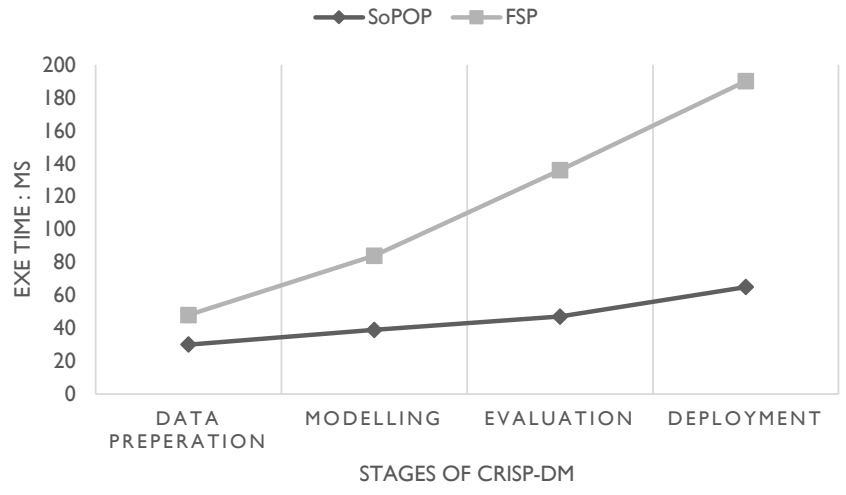


Figure 9.5.6: VR time for the various CRISP-DM stages

9.6 Evaluate the Execution Stage of the ASC

We already discussed Planning, Discovery, Selection and VR stages of the proposed ASC architecture. The proposed Execution stage responsible to execute the resulted TOP by the VR stage.

Therefore, I organized the evaluation in term of two stages. Chapter 9.6.2.1 discuss the evaluation of efficiency behind the proposed method. Section 9.6.2.2 discuss the evaluation of effectiveness behind the proposed method.

1. The efficiency of the proposed method, this measures the efficiency factors in satisfying requirements by changing the constraint parameters and overserving results.
2. The effectiveness of the proposed, this measures the effectiveness factors in satisfying requirements by changing the service related factors and observing the results.

9.6.1 Experiment Setup

Up to the date, fully automated workflow generation for the BDA based on the ASC has not been proposed. Ardagna et. al recently proposed model based BDA, however it is very limited to support the automate BDA [124]. Moreover frameworks other than the ASC such as, KNIME and RapidMiner neither generate automated workflow nor verify and refine the workflow [125], [126]. All the above method need extensive knowledge in service composition or BDA or both. Therefore we compared the proposed method by changing the parameters of itself.

Prototype solution comprises 16 unground tasks. Therefore, we prepared the service registry which contained 80 CWSs. Each CWS comprises with two atomic web services as we discussed in VR stage. Discovering services for the respective tasks is one time job and therefore, we used clustered service registry for the evaluation. Three cluster registries are used, which are contained three, four and five candidate CWS's for the each tasks. Qualitative and quantitative QoS values of the CWS are predefined. Moreover QoS requirements of the given workflow already set in selection related requirements. Impacts of each stages of the proposed ASC have discussed in previous

sections 9.2, 9.3, 9.4, and 9.5. Therefore here we discuss the overall prototype solution and its results.

9.6.2 Evaluation

9.6.2.1 Efficiency

To evaluate the efficiency change the constraints related to the We already discussed Planning, Discovery, Selection and VR stages of the proposed ASC architecture. The proposed Execution stage responsible to execute the resulted TOP by the VR stage. Fig. 9.6.1 shows the GUI of the proposed solution. We changed the constraints related data mining process and observe the behaviors.

Fig. 9.6.2 and 9.6.3 show the two different types of data mining constraints and associated abstract workflow, discovered services, selected services and VR TOP's. Consider the Fig. 9.6.2, from the top of that, first it selected the data mining constraints of the batch processing BDA requirement, next it shows the relevant abstract workflow for the analysis. The abstract workflow contained 7 tasks. After it shows the discovered

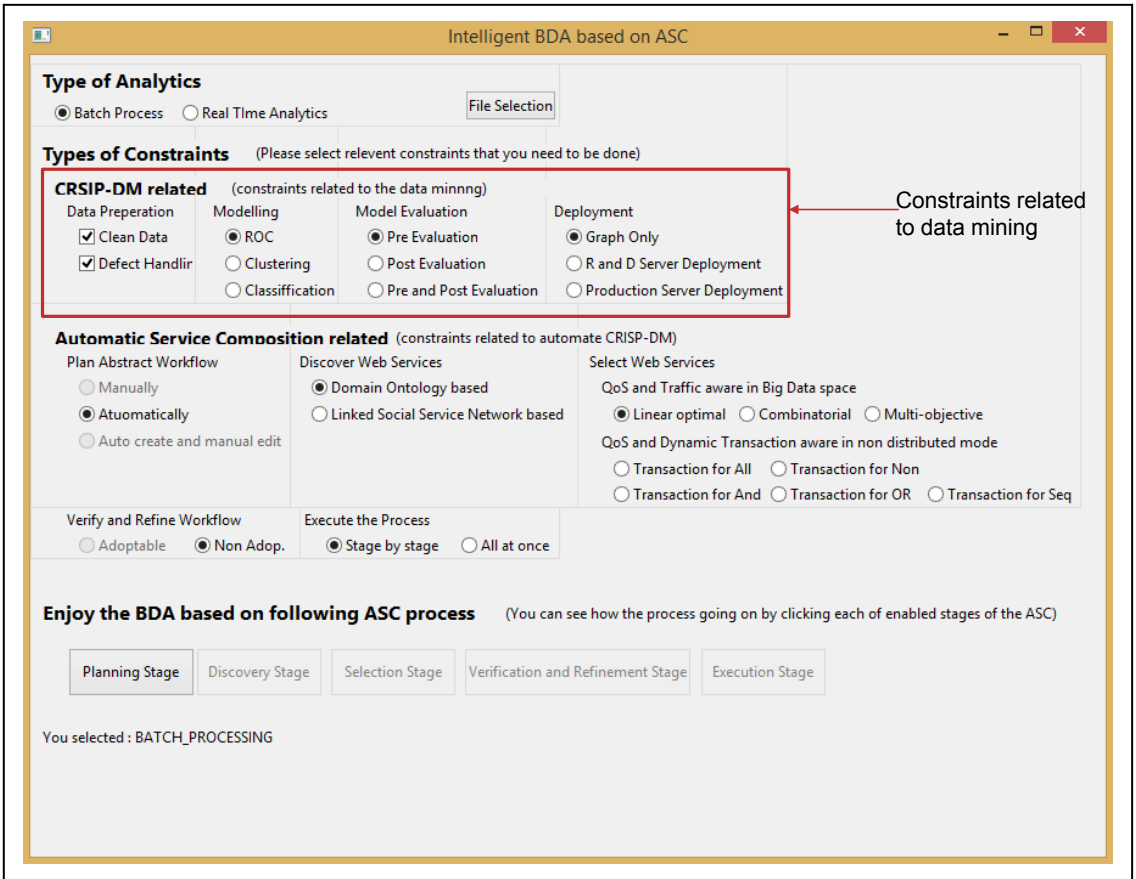


Figure 9.6.1: Proposed GUI for the BDA automation based on ASC

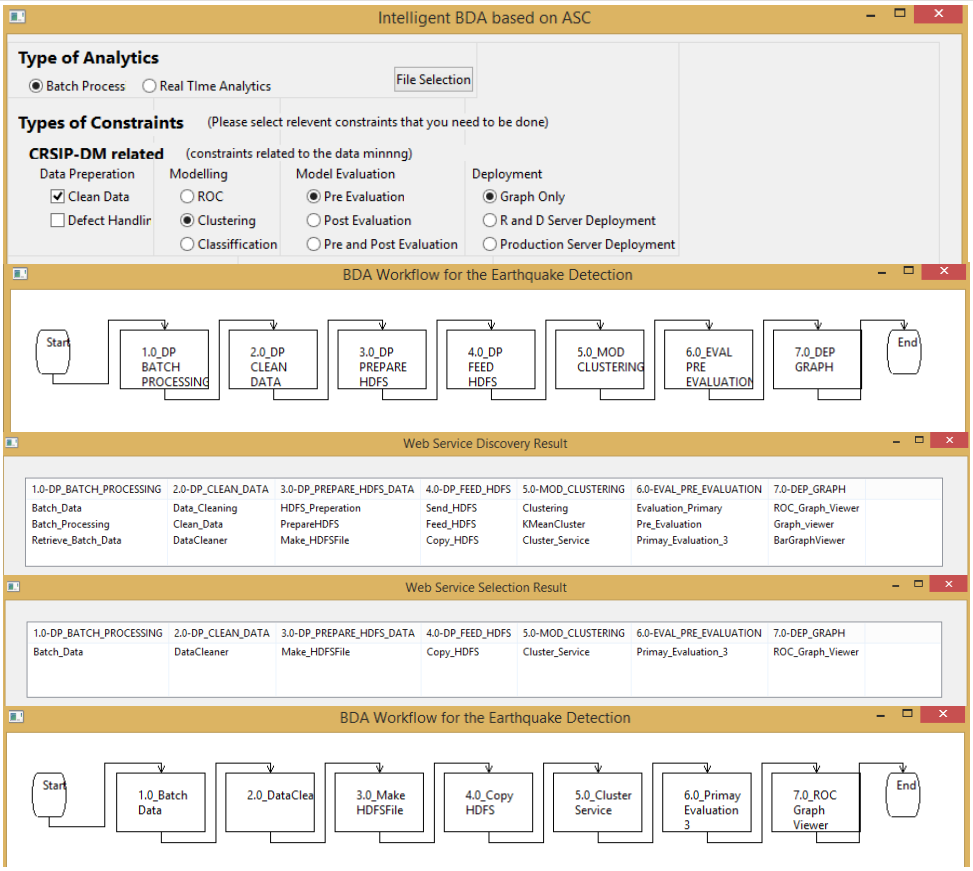


Figure 9.6.2: Selected data mining constrains and results of the ASC stages for the batch processing

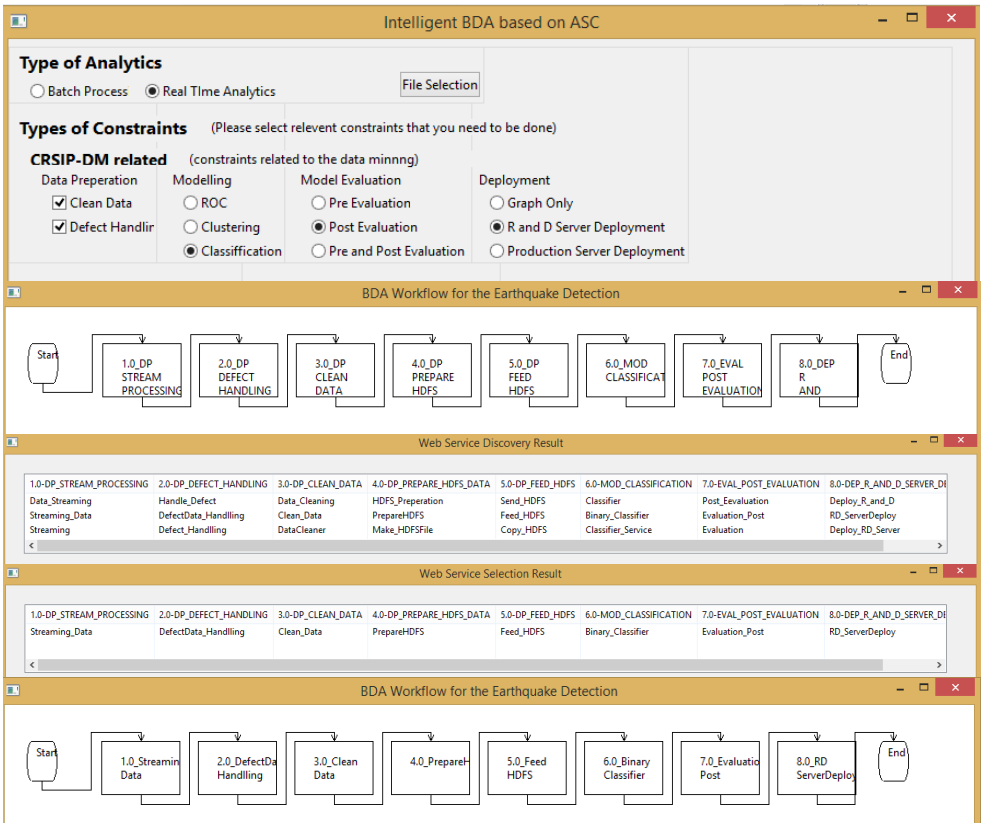


Figure 9.6.2: Selected data mining constrains and results of the ASC stages for the real time analytics

services for the each tasks. Given results derived from the clustered web service registry,

which contained three services for the each tasks. After that, process shows the service selection results. Finally, process prepared the TOP by the VR stage. Likewise, Fig. 9.6.3 shows the data mining constraints for the real time processing BDA requirement and respective solutions resulted by the each stages of the process, which contained 8 tasks.

Fig. 9.6.2 and 9.6.3 confirmed that the proposed solution is efficiency adopted dynamic data mining constraints despite of their diversity. This dynamic workflow generation features allows the proposed constraint aware workflow generation based on the Graphplan [78]. Once the end user select the constraint behind his BDA process, then the proposed Graphplan technique reasoning the constraints and generates the abstract workflow automatically. Rest of the stages of the ASC process is originally follow the abstract workflow results generated by the Planning stage. Nevertheless, according to the proposed solution, it allows to add more features to the data mining process, ie. CRISP-DM. It needs to add relevant add relevant constraint reasoning for the new adding CRISP-DM feature. This implies that the proposed method is customizable and adoptable for the future needs as it occurs.

Moreover, two figures shows that it generates dynamic results according to the workflow resulted by the Planning stage. Results of Discovery, Selection and VR stages are identical to each other. This implies, that the proposed solution is efficiently supporting the dynamic processing.

9.6.2.2 Effectiveness

Next we calculate the processing effectiveness for the overall process with different use cases. First we conducted experiment with dynamic web service clusters which contained three, four and five candidate services and observe processing time of the each stages of the ASC process. Next we conducted experiments with a given service cluster while changing the number records in raw data file for the BDA. For these two we employed discussed the scenario in Chapter 8.

Figure 9.6.4 shows the results of first experiments. It shows that the number of

candidate service effect to the overall effectiveness. It causes, in this scenario it comprises 9 abstract tasks. Then when it is with three candidate services, it manages 3^9 number of plans, when it has four and five of candidate services it manages 4^9 and 5^9 number of plans. Then it increases the number of plans in selection and cause to increase the processing time. However, it consumes considerably low processing time when it compares with the overall processing time. That means, number of candidate services negatively affect to the overall processing time.

Next Fig. 9.6.5 shows the results of the second experiments. It shows processing time of the execution stage is rapidly increasing the processing time and shows no effect at all with the other stages of the ASC process.

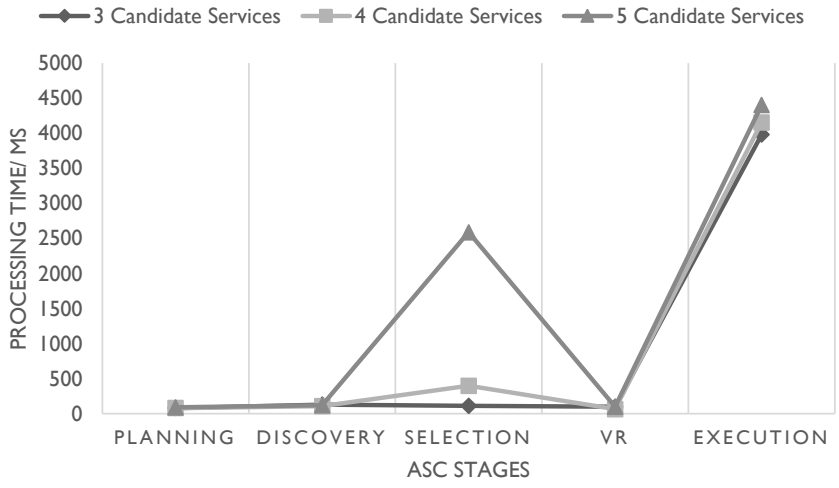


Figure 9.6.4: Effectiveness for dynamic service cluster

Figure 9.6.5: Effectiveness for dynamic raw data file for the BDA

Then based on the Fig. 9.6.4 and 9.6.5, we can conclude number of candidate services has very little effect to the overall effectiveness and file size involved in the BDA requirements is main cause behind the overall effectiveness. That means, the file size involving the BDA is unavoidable constraints due it is depend on the end user. However rest of the constraint behind the process such as data mining constraint has no effect to the effectiveness of the ASC process except the last execution stage. Moreover, number of candidate services also has mild effect, however compared to the overall process it can be considered as the ignorable effect.

Then based on the Section 9.6.1.1 and 9.6.1.2, we can conclude the proposed method is efficiency support to the dynamic constraints, i.e. BDA requirements in intelligent manner and adoptable for future needs as well. Nevertheless proposed solution almost independent from number of services and effectively accomplishes the BDA requirements. This implies, the proposed solution provides efficient and effective platform for the intelligent BDA automation.

This page intentionally left blank.

Chapter 10 Conclusion and Future Works

Architectural perspective, we have been working to automate the BDA process based on ASC and CRISP-DM. As the first step of the research, we have proposed standard architectural design process for the BDA domain and it was result three staged architectures. RA, SA and high level UML class diagram are represents these stages and those were evaluated by industry expert using SAAM, a standard architectural evaluation method. Final result gave an assurance that our proposed architecture can recommended to adopt smoothly to the BDA domain. Once it is completed the development process of the all four stages of the ASC process this architecture will be affirmed and then it will be an empirical proposal for the BDA. Further on architectural perspective, we plan to study and observe behaviors of the key technologies with relevant substitution in that time and the Deep Learning use cases. Moreover, system needs to extend in to smart framework for the Data Science.

Abstract workflow generation perspective, relevant to the planning stage of the ASC, we proposed workflow generation for BDA process based on the GraphPlan technique. Mainly we focused on constrain awareness, search space, and resource consuming during plan execution. Proposed task simulation technique allows to generate the improved task planner rather than default action planner result by the GraphPlan. Our experiment results affirm, that the proposed method efficiently managed the constrained awareness and adoptable to large search space. And, it shows, that the proposed method could achieve most effective planner, perspective to the

resource consumption during the execution of the resulted planner. That means, our objective behind the workflow generation for the BDA is satisfied in effective and efficient manner. In our future work, we are working to extend this proposal to adhere to the replanning and automata of the BDA planning processes.

Discovering web services perspective, according to the approaches to the discovery of the ASC process, we have proposed two methods considering the two major concerns in the BDA domain and discovery stage. Which are precise service discovery based on domain and context and facilitate to achieve the workflow in the effective way. According to result of evaluation, it was proved that our proposed methods are performing well in their perspective. Then it can be used one of these methods, based on the criticality of the requirement. It can use Domain Ontology method if the discovery requirement of the BDA is keen for the precise service and if it's critical in workflow it can be used the LSSN method. We plan to mature the Ontologies used in discovery as the future works in discovery stage. And we plan to accumulate in to single cognitive ontology that both ontologies used in plan and discovery stages. Also we plan to extend our LSSN method, as the workflow discovery method in future.

Selecting web services achieve under two key selection requirements. As of the QoS and transaction awareness perspective, proposed novel approach to customizable TS awareness for the BDA automation on ASC by considering diverse requirements of the BDA process. Experiment results show we proposed method is outperform existing multivariate optimization method and reaches the goals while consuming minimal resources. That means, TCQS perform better and reach the global optimal, very efficient manner. Our aim was to proposed flexible and assured composition method for BDA automation based on ASC. Experiment results show that our method is highly effective and efficient to achieve the desired goal. As the future works, we are working to address the uncertainty and concerns occurred in trustworthy of the service domain and workflow adaptability for more effective BDA automation based on ASC.

And as of Big Data space selection requirement perspective, We have proposed a multiobjective service selection solution that considers external and internal traffic congestions in Big Data space. Proposed method addresses the concerns arising from

the flooding of services and then the selection processes can be handled more efficiently in Big Data space. Data traffic caused by the ZipF and Pareto are called as the external traffic congestions. Data traffic caused by the shuffling stage of the MR process is identified as the internal traffic. We have proposed a complete model for traffic control while dealing with the selection process in MR jobs. Novel QoS-aware traffic-efficient methods have been proposed for external traffic congestion. In addition, we have introduced a middle agent to address the internal traffic congestion, which eases the reducer workload in an efficient manner. We adopted three selection criteria's for the multiobjective selection methods, which are linear optimal, combinatorial and multivariate QoS optimizations. These methods are based on both linear programming and dynamic programming. The proposed distributed algorithm can adapt easily to dynamic selection requirements. Our experimental results demonstrate that the proposed method handles traffic congestions efficiently and effectively in producing composition plans for multiobjective selection requirements. Internal traffic efficiency shows relatively low compared to the external traffic efficiency. Only external traffic efficiency shows a nearly proportional relationship to the computation complexity of the objective functions. The Proposed method is a well-behaved QoS-aware rule-based traffic-efficient service-selection method for Big Data space. In future work, we aim to investigate qualitative QoS criteria and uncertainty in the QoS values for the selection method.

Verification and refinement is the next stage we accomplished in ASC process. We have proposed a method for constraint-driven composite service-oriented POP technique. According to our literature review, we are among the very first to propose a CWS based constraint aware POP for data analytics in ASC. The main idea is to prepare a flawless workflow in ASC. In the VR stage, we prepare a TOP for the BDA, which guarantees flaw-free seamless workflow. Our experiments on the POP's VR showed that the SoPOP outperformed related techniques in terms of both efficiency and effectiveness. That means, two-stage POP solving technique generating seamless data analytical workflows in a very efficient and effective manner. In future work we investigate the execution stage of the ASC process to accomplish BDA automation.

Composing web services according to the given sequence of order by VR stage represent the Execution stage of the ASC. According to the experiment results, proposed solution almost independent from stages of the ASC, number of services and effectively accomplishes the BDA requirements. This implies, the proposed solution provides efficient and effective platform for the intelligent BDA automation. As of the future works, system needs to provides manual workflow generation alike KNIME, RapidMiner, also auto generation and manual edit facility as well. GUI of the proposed solution needs to be improved with more customer oriented manner with necessary improvements. Moreover, we plan to extend the proposed solution as smart framework for the BDA and other data science methods such as Deep Learning.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Incheon Paik for the continuous support from my master studies to the PhD and for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing the thesis. I could not have imagined having a better advisor and mentor for my postgraduate studies.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Qiangfu Zhao, Prof. Alexander P. Vazhenin and Prof. Keitaro Naruse for their insightful comments and encouragement, also for their reviews which incented me to widen my research from various perspectives.

I thank my fellow lab members for their encourage-full discussions and critics those were lead me to see different perspective of my research motivation.

I would like to thank my family: my mother, father (deceased), my wife, my brothers and sisters for their immense support (motivation, economically, and spiritually) throughout the entire life in general. They managed most of my responsibilities and maximally reduced the burdens on my side. Their contribution was greatly helped me to continue the postgraduate studies in peace of mind. Last but not the least, I would like to thank to the father of free education of Sri Lanka, Dr. C. W. Wijekoon Kannangara, who had done his ultimatum to give free education from kindergarten to the university to all of the Sri Lankan citizen. It is obvious, that I wouldn't be at this stage if none of them (my family members and father of free education Sri Lanka) were exist.

This page intentionally left blank.

References

[1] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, p. 1, 2015.

[2] K. Kersting and U. Meyer, "From Big Data to Big Artificial Intelligence?" Springer, 2018.

[3] C. Wu, R. Buyya, and K. Ramamohanarao, "Big Data Analytics = Machine Learning + Cloud Computing," *Big Data Princ. Paradig.*, pp. 1–27, 2014.

[4] F. J. Ohlhorst, *Big data analytics: turning big data into big money*. John Wiley & Sons, 2012.

[5] F. Provost and T. Fawcett, "Data Science and its Relationship to Big Data and Data-Driven Decision Making," *Big Data*, vol. 1, no. 1, pp. 51–59, 2013.

[6] T. H. A. S. Siriweera, I. Paik, and B. T. G. S. Kumara, "QoS and Customizable Transaction-aware Selection for Big Data Analytics on Automatic Service Composition," in *Services Computing (SCC), 2017 IEEE International Conference on*, 2017, pp. 116–123.

[7] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big data for remote sensing: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 11, pp. 2207–2219, 2016.

[8] Q. Luo, C. Guo, Y. J. Zhang, Y. Cai, and G. Liu, "Algorithms designed for compressed-gene-data transformation among gene banks with different references," *BMC Bioinformatics*, vol. 19, no. 1, p. 230, 2018.

[9] R. Dukaric and M. B. Juric, "BPMN Extensions for Orchestrating Cloud

Environments Using a Two-layer Approach," *J. Vis. Lang. Comput.*, 2018.

[10] C. Shearer, "The CRISP-DM model: the new blueprint for data mining," *J. data Warehous.*, vol. 5, no. 4, pp. 13–22, 2000.

[11] T. H. Akila S. Siriweera, Incheon Paik, Banage T. G. S. Kumara, and C. K. Koswatta, "Architecture for intelligent big data analysis based on automatic service composition," *Int. J. Big Data*, vol. 2, no. 2, pp. 1–14, 2015.

[12] I. Paik, W. Chen, and M. N. Huhns, "A scalable architecture for automatic service composition," *IEEE Trans. Serv. Comput.*, vol. 7, no. 1, pp. 82–95, 2014.

[13] E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht, "Keystoneml: Optimizing pipelines for large-scale advanced analytics," *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. pp. 535–546, 2017.

[14] T. H. Akila S. Siriweera, I. Paik, B. T. G. S. Kumara, and K. R. C. Koswatta, "Intelligent Big Data Analysis Architecture Based on Automatic Service Composition," in *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, 2015.

[15] B. T. G. S. Kumara, I. Paik, J. Zhang, T. H. Akila S. Siriweera, and K. R. C. Koswatte, "Ontology-Based Workflow Generation for Intelligent Big Data Analytics," in *Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015*, 2015.

[16] Y. Gil, "Teaching Big Data Analytics Skills with Intelligent Workflow Systems," *Proc. 30th Conf. Artif. Intell. (AAAI 2016)*, pp. 4081–4088, 2016.

[17] Y. Gil *et al.*, "Time-Bound Analytic Tasks on Large Datasets through Dynamic Dynamic Configuration of Workflows," *Proc. Eighth Work. Work. Support Large-Scale Sci.*, pp. 88–97, 2013.

[18] Y. Gil, P. Gonzalez-Calero, J. Kim, J. Moody, and V. Ratnakar, "Automatic Generation of Computational Workflows from Workflow Templates Using Distributed Data and Component Catalogs," *Proc. Sixth Symp. Educ. Adv. Artif. Intell.*, pp. 1–77, 2009.

[19] J. Kranjc, R. Orač, V. Podpečan, N. Lavrač, and M. Robnik-Šikonja,

"ClowdFlows: Online workflows for distributed big data mining," *Futur. Gener. Comput. Syst.*, vol. 68, pp. 38–58, 2017.

[20] M. Hauder, Y. Gil, and Y. Liu, "A framework for efficient data analytics through automatic configuration and customization of scientific workflows," *Proc. - 2011 7th IEEE Int. Conf. eScience, eScience 2011*, pp. 379–386, 2011.

[21] P. Wang, P. Wang, Z. Ding, C. Jiang, M. Zhou, and Y. Zheng, "Automatic Web Service Composition Based on Uncertainty Execution Effects Automatic Web Service Composition Based on Uncertainty Execution Effects," *IEEE Trans. Serv. Comput.*, vol. 9, no. January 2015, pp. 551–565, 2016.

[22] M. Pistore, A. Marconi, P. Bertoli, and P. Traverso, "Automated composition of web services by planning at the knowledge level," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 1252–1259, 2005.

[23] R. Alarcon, R. Saffie, N. Bravo, and J. Cabello, "REST web service description for graph-based service discovery," in *International Conference on Web Engineering*, 2015, pp. 461–478.

[24] T. H. Akila S. Siriweera, I. Paik, and B. T. G. S. Kumara, "Onotology-based service discovery for intelligent Big Data analytics," in *IEEE 7th International Conference on Awareness Science and Technology, iCAST 2015 - Proceedings*, 2015.

[25] T. Rajendran and P. Balasubramanie, "An optimal agent-based architecture for dynamic web service discovery with qos," in *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on*, 2010, pp. 1–7.

[26] F. T. Johnsen, T. Hafsøe, and M. Skjegstad, "Web services and service discovery in military networks," *14th ICCRTS, Washingt. DC, US*, 2009.

[27] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, and others, "Web services description language (WSDL) 1.1." Citeseer, 2001.

[28] G. Wen-yue, Q. Hai-cheng, and C. Hong, "Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system," in *Information Management and Engineering (ICIME), 2010 The 2nd*

IEEE International Conference on, 2010, pp. 601–604.

[29] Y.-H. Tsai, S.-Y. Hwang, and Y. Tang, "A hybrid approach to automatic web services discovery," in *Service Sciences (IJCSS), 2011 International Joint Conference on*, 2011, pp. 277–281.

[30] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Serv. Sci.*, vol. 1, no. 1, pp. 83–98, 2008.

[31] T. Wen, G. Sheng, Y. Li, and Q. Guo, "Research on Web service discovery with semantics and clustering," in *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*, 2011, vol. 1, pp. 62–67.

[32] E. Karakoc, K. Kardas, and P. Senkul, "A workflow-based Web service composition system," in *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, 2006, pp. 113–116.

[33] J. El Hadad, M. Manouvrier, and M. Rukoz, "(reading)TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition," *Serv. Comput. IEEE Trans.*, vol. 3, no. 1, pp. 73–85, 2010.

[34] ZhiJun Ding, JunJun Liu, YouQing Sun, ChangJun Jiang, and MengChu Zhou, "A Transaction and QoS-Aware Service Selection Approach Based on Genetic Algorithm," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 45, no. 7, pp. 1035–1046, 2015.

[35] H. Liu, Z. Zheng, W. Zhang, and K. Ren, "A global graph-based approach for transaction and QoS-aware service composition," *KSII Trans. Internet Inf. Syst.*, vol. 5, no. 7, pp. 1252–1273, 2011.

[36] J. Cao, G. Zhu, X. Zheng, B. Liu, and F. Dong, "TASS: Transaction assurance in service selection," *Proc. - 2012 IEEE 19th Int. Conf. Web Serv. ICWS 2012*, pp. 472–479, 2012.

[37] Y. Cardinale, J. El Haddad, M. Manouvrier, and M. Rukoz, "Web service selection for transactional composition," *Procedia Comput. Sci.*, vol. 1, no. 1, pp. 2689–2698, 2010.

[38] F. Montagut, R. Molva, and S. T. Golega, "Automating the composition of

transactional web services," *Int. J. Web Serv. Res.*, vol. 5, no. 1, p. 24, 2008.

[39] S. Bhiri *et al.*, "Extending workflow patterns with transactional dependencies to define reliable composite Web services," in *Web Services, 2007. ICWS 2007. IEEE International Conference on*, 2008, vol. 5, no. 1, p. 145.

[40] H. Wang, C. Yu, L. Wang, and Q. Yu, "Effective BigData-Space Service Selection over Trust and Heterogeneous QoS Preferences," *IEEE Trans. Serv. Comput.*, vol. 99, no. X, 2015.

[41] Y. Xia, J. Chen, X. Lu, C. Wang, and C. Xu, "Big traffic data processing framework for intelligent monitoring and recording systems," *Neurocomputing*, vol. 181, pp. 139–146, 2016.

[42] J. Lin and others, "The curse of zipf and limits to parallelization: A look at the stragglers problem in mapreduce," in *7th Workshop on Large-Scale Distributed Systems for Information Retrieval*, 2009, vol. 1, pp. 57–62.

[43] Y. Kim and K.-G. Doh, "A Trust Management Model for QoS-Based Service Selection.," in *WISA*, 2012, pp. 345–357.

[44] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "A survey on trust and reputation models for Web services: Single, composite, and communities," *Decis. Support Syst.*, vol. 74, pp. 121–134, 2015.

[45] Z. Wan, F. J. Meng, J. M. Xu, and P. Wang, "Service composition pattern generation for cloud migration: a graph similarity analysis approach," in *Web Services (ICWS), 2014 IEEE International Conference on*, 2014, pp. 321–328.

[46] X. Huang, "UsageQoS: Estimating the QoS of Web services through online user communities," *ACM Trans. Web*, vol. 8, no. 1, p. 1, 2013.

[47] G. Kang, J. Liu, M. Tang, X. Liu, and K. K. Fletcher, "Web service selection for resolving conflicting service requests," in *Web Services (ICWS), 2011 IEEE International Conference on*, 2011, pp. 387–394.

[48] J. El Hadad, M. Manouvrier, and M. Rukoz, "TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition," *IEEE Trans. Serv. Comput.*, vol. 3, no. 1, pp. 73–85, 2010.

[49] H. Gao, J. Yan, and Y. Mu, "Trust-oriented QoS-aware composite service

selection based on genetic algorithms," *Concurr. Comput. Pract. Exp.*, vol. 26, no. 2, pp. 500–515, 2014.

[50] C. Zhang, H. Yin, and B. Zhang, "A novel ant colony optimization algorithm for large scale QoS-based service selection problem," *Discret. Dyn. Nat. Soc.*, vol. 2013, 2013.

[51] Y. Chen, S. Alspaugh, and R. Katz, "Interactive analytical processing in big data systems: A cross-industry study of mapreduce workloads," *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 1802–1813, 2012.

[52] H. Ke, S. Member, and P. Li, "On Traffic-Aware Partition and Aggregation in MapReduce for Big Data Applications," vol. 2, pp. 1–12.

[53] D. Ersoz, M. S. Yousif, and C. R. Das, "Characterizing network traffic in a cluster-based, multi-tier data center," in *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*, 2007, p. 59.

[54] J. Zhang *et al.*, "Optimizing Data Shuffling in Data-Parallel Computation by Understanding User-Defined Functions.," in *NSDI*, 2012, vol. 12, p. 22.

[55] M. Aledhari, M. Di Pierro, M. Hefeida, and F. Saeed, "A Deep Learning-Based Data Minimization Algorithm for Fast and Secure Transfer of Big Genomic Datasets," *IEEE Trans. Big Data*, 2018.

[56] Y. Zhao, K. Yoshigoe, J. Bian, M. Xie, Z. Xue, and Y. Feng, "A Distributed Graph-Parallel Computing System with Lightweight Communication Overhead," *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 204–218, 2016.

[57] Z. Yan, L. Zhang, W. Ding, and Q. Zheng, "Heterogeneous Data Storage Management with Deduplication in Cloud Computing," *IEEE Trans. Big Data*, 2017.

[58] X. Chen, W. Chen, J. Lee, and N. B. Shroff, "Delay-optimal buffer-aware scheduling with adaptive transmission," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2917–2930, 2017.

[59] A. Asadi *et al.*, "A survey on opportunistic scheduling in wireless communications," *IEEE Trans. Softw. Eng.*, vol. 15, no. 2, pp. 311–327, 2013.

[60] O. Sapena, E. Onaindía, and A. Torreño, "Forward-chaining planning with a

flexible least-commitment strategy," *Front. Artif. Intell. Appl.*, vol. 256, no. January, pp. 41–50, 2013.

[61] L. Wang, Y. Ma, J. Yan, V. Chang, and A. Y. Zomaya, "Pipscloud: High performance cloud computing for remote sensing big data management and processing," *Futur. Gener. Comput. Syst.*, 2016.

[62] J. Peer, "A POP-Based Replanning Agent for Automatic Web Service Composition.," in *ESWC*, 2005, vol. 3532, pp. 47–61.

[63] M. Szreter, "A graph-based reduction in Planics abstract planning, based on partial orders of services," in *CS&P*, 2016, pp. 165–170.

[64] Y. Yan, Y. Liang, and H. Liang, "Composing business processes with partial observable problem space in Web services environments," in *Web Services, 2006. ICWS'06. International Conference on*, 2006, pp. 541–548.

[65] I. Wassink, M. Ooms, and P. van der Vet, "Designing workflows on the fly using e-BioFlow," in *Service-Oriented Computing*, Springer, 2009, pp. 470–484.

[66] X. Nguyen and S. Kambhampati, "Reviving partial order planning," in *IJCAI*, 2001, vol. 1, pp. 459–464.

[67] H. L. S. Younes and R. G. Simmons, "VHPOP: Versatile heuristic partial order planner," *J. Artif. Intell. Res.*, vol. 20, pp. 405–430, 2003.

[68] S. Kikuchi, A. Nakamura, and D. Yoshino, "Evaluation on Information Model about Sensors Featured by Relationships to Measured Structural Objects," *Adv. Internet Things*, vol. 6, no. 03, p. 31, 2016.

[69] F. Zulkernine, P. Martin, Y. Zou, M. Bauer, F. Gwadry-Sridhar, and A. Aboulnaga, "Towards cloud-based analytics-as-a-service (claaas) for big data analytics in the cloud," *Big Data (BigData Congress), 2013 IEEE International Congress on*. pp. 62–69, 2013.

[70] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "Qos aggregation for web service composition using workflow patterns," in *Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*, 2004, pp. 149–159.

[71] F. Casati, S. Ceri, B. Pernici, and G. Pozzi, "Workflow evolution," *Data Knowl.*

Eng., vol. 24, no. 3, pp. 211–238, 1998.

[72] F. Casati, S. Ceri, S. Paraboschi, and G. Pozzi, "Specification and implementation of exceptions in workflow management systems," *ACM Trans. Database Syst.*, vol. 24, no. 3, pp. 405–451, 1999.

[73] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *J. Big Data*, vol. 2, no. 1, p. 21, 2015.

[74] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artif. Intell. Prentice-Hall, Egnlewood Cliffs*, vol. 25, p. 27, 1995.

[75] T. Malik Ghallab Dana Nau Paolo, "Automated Planning and Acting," *Cambridge University Press, 2016*. pp. 1–546, 2016.

[76] A. T. Gerhard Wickler, "AI Planning," *AI Planning MOOC Course Materials*. 2015.

[77] W. M. P. Van Der Aalst and A. H. M. Ter Hofstede, "YAWL: yet another workflow language," *Inf. Syst.*, vol. 30, no. 4, pp. 245–275, 2005.

[78] T. H. A. S. Siriweera, I. Paik, and B. T. G. S. Kumara, "Constraint-Driven Dynamic Workflow for Automation of Big Data Analytics Based on GraphPlan," *2017 IEEE Int. Conf. Web Serv.*, pp. 357–364, 2017.

[79] T. H. Akila S. Siriweera, I. Paik, J. Zhang, and B. T. G. S. Kumara, "Big data analytic service discovery using social service network with domain ontology and workflow awareness," in *Proceedings - 2016 IEEE International Conference on Web Services*, 2016.

[80] Z. Ye, X. Zhou, and A. Bouguettaya, "Genetic algorithm based QoS-aware service compositions in cloud computing," in *International Conference on Database Systems for Advanced Applications*, 2011, pp. 321–334.

[81] M. Tang and L. Ai, "A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010, pp. 1–8.

[82] B. Huang *et al.*, "Genetic algorithm based QoS-aware service compositions in cloud computing," in *International Conference on Database Systems for Advanced Applications*, 2011, vol. 8, no. 4, pp. 1–8.

[83] Sharad Mehrotra and Henry F. Korth, “A transaction model for multidatabase systems,” *Proceeding Int. Conf. Distrib. Comput. Syst.*, vol. June, pp. 56–63, 1992.

[84] P. Guo, “Software tools to facilitate research programming,” *Ph.D. Diss. Stanford Digit. Repos.*, no. May, p. 230, 2012.

[85] W. Song and H.-A. Jacobsen, “Static and dynamic process change,” *IEEE Trans. Serv. Comput.*, vol. 11, no. 1, pp. 215–231, 2018.

[86] S. Hu, V. Muthusamy, G. Li, and H.-A. Jacobsen, “Distributed automatic service composition in large-scale systems,” in *Proceedings of the second international conference on Distributed event-based systems*, 2008, pp. 233–244.

[87] G. Li, V. Muthusamy, and H.-A. Jacobsen, “A distributed service-oriented architecture for business process execution,” *ACM Trans. Web*, vol. 4, no. 1, p. 2, 2010.

[88] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “Qos-aware middleware for web services composition,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, 2004.

[89] T. H. Akila S. Siriweera, I. Paik, and B. T. G. S. Kumara, “QoS-Aware Traffic-Efficient Web Service Selection over BigData Space,” *2016 IEEE Int. Conf. Comput. Inf. Technol.*, pp. 197–203, 2016.

[90] T. Yu and K.-J. Lin, “Service selection algorithms for Web services with end-to-end QoS constraints,” *Inf. Syst. E-bus. Manag.*, vol. 3, no. 2, pp. 103–126, 2005.

[91] M. Alrifai, T. Risse, P. Dolog, and W. Nejdl, “A Scalable Approach for QoS-Based Web Service Selection.,” in *ICSOC Workshops*, 2008, pp. 190–199.

[92] Q. He, J. Yan, H. Jin, Y. Yang, and others, “Quality-Aware Service Selection for Service-Based Systems Based on Iterative Multi-Attribute Combinatorial Auction.,” *IEEE Trans. Softw. Eng.*, vol. 40, no. 2, pp. 192–215, 2014.

[93] L. Wang, S. Guo, X. Li, B. Du, and W. Xu, “Distributed manufacturing resource selection strategy in cloud manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 9, pp. 3375–3388, 2018.

[94] M. Alrifai, D. Skoutas, and T. Risse, “Selecting skyline services for QoS-based

web service composition," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 11–20.

[95] J. Zhou and X. Yao, "A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition," *Int. J. Adv. Manuf. Technol.*, vol. 88, no. 9–12, pp. 3371–3387, 2017.

[96] X. Xue, Y.-M. Kou, S.-F. Wang, and Z.-Z. Liu, "Computational experiment research on the equalization-oriented service strategy in collaborative manufacturing," *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 369–383, 2018.

[97] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 881–890.

[98] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things," *IEEE Trans. Ind. Informatics*, 2018.

[99] S. Kanrar and N. K. Mandal, "Traffic analysis and control at proxy server," in *Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on*, 2017, pp. 164–167.

[100] M. Furqan, C. Zhang, W. Yan, A. Shahid, M. Wasim, and Y. Huang, "A collaborative hotspot caching design for 5G cellular network," *IEEE Access*, vol. 6, pp. 38161–38170, 2018.

[101] H. Beyranvand, M. Lévesque, M. Maier, J. A. Salehi, C. Verikoukis, and D. Tipper, "Toward 5G: FiWi enhanced LTE-A HetNets with reliable low-latency fiber backhaul sharing and WiFi offloading," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 690–707, 2017.

[102] K. Wang, X. Li, H. Ji, and X. Du, "Modeling and optimizing the LTE discontinuous reception mechanism under self-similar traffic," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5595–5610, 2016.

[103] T. H. Akila S. Siriweera and I. Paik, "Service Selection on BigData-Space based on Heterogeneous QoS Preferences," pp. 1–4, 2016.

[104] J. Zhang *et al.*, "Optimizing Data Shuffling in Data-Parallel Computation by

Understanding User-Defined Functions."

[105] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, vol. 1, pp. 126–134.

[106] D. G. Dolgikh and A. M. Sukhov, "Parameters of cache systems based on a Zipf-like distribution," *Comput. Networks*, vol. 37, no. 6, pp. 711–716, 2001.

[107] F. Ahmad, S. Lee, M. Thottethodi, and T. N. Vijaykumar, "MapReduce with communication overlap (MaRCO)," *J. Parallel Distrib. Comput.*, vol. 73, no. 5, pp. 608–620, 2013.

[108] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[109] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "Hybrid optimization algorithm for large-scale QoS-aware service composition," *IEEE Trans. Serv. Comput.*, vol. 10, no. 4, pp. 547–559, 2017.

[110] A. Bukhari and X. Liu, "A Web service search engine for large-scale Web service discovery based on the probabilistic topic modeling and clustering," *Serv. Oriented Comput. Appl.*, pp. 1–14, 2018.

[111] Q. He *et al.*, "Localizing Runtime Anomalies in Service-Oriented Systems," *IEEE Trans. Serv. Comput.*, vol. 10, no. 1, pp. 94–106, 2017.

[112] J. Viil and S. N. Srirama, "Framework for automated partitioning and execution of scientific workflows in the cloud," *J. Supercomput.*, vol. 74, no. 6, pp. 2656–2683, 2018.

[113] F. Marozzo, D. Talia, and P. Trunfio, "A workflow management system for scalable data mining on clouds," *IEEE Trans. Serv. Comput.*, 2016.

[114] W. M. P. van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distrib. parallel databases*, vol. 14, no. 1, pp. 5–51, 2003.

[115] B. Wang, A. Haller, and F. Rosenberg, "Generating workflow models from OWL-S service descriptions with a partial-order plan construction," *Proc. - 2011*

IEEE 9th Int. Conf. Web Serv. ICWS 2011, pp. 714–715, 2011.

[116] S.-C. Oh, D. Lee, and S. R. Kumara, "Web service planner (wspr): An effective and scalable web service composition algorithm," *Int. J. Web Serv. Res.*, vol. 4, no. 1, pp. 1–22, 2007.

[117] M. Szreter, "A graph-based reduction in Plan ics abstract planning , based on partial orders of services (Extended Abstract)."

[118] M. Kuzu and N. K. Cicekli, "Dynamic planning approach to automated web service composition," *Appl. Intell.*, vol. 36, no. 1, pp. 1–28, 2012.

[119] T. H. A. S. Siriweera and I. Paik, "Constraint-aware Dynamic Partial Order Plan Generation for Big Data Analytics based on Automatic Service Composition," in *Constraint-aware Dynamic Partial Order Plan Generation for Big Data Analytics based on Automatic Service Composition*, 2018.

[120] T. H. A. S. Siriweera and I. Paik, "QoS-Aware Rule-Based Traffic-Efficient Multiobjective Service Selection in Big Data Space," *IEEE Access*, vol. 6, pp. 48797–48814, 2018.

[121] E. Al-Masri and Q. H. Mahmoud, "The QWS dataset," *2013-03-12]. http:? www. uoguelph. ca/~ qmahmoud/qws/index html*. 2008.

[122] A. L. Blum and M. L. Furst, "Fast planning through planning graph analysis," *Artif. Intell.*, vol. 90, no. 1–2, pp. 281–300, 1997.

[123] E. Al-Masri and Q. H. Mahmoud, "Discovering the best web service," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1257–1258.

[124] C. A. Ardagna, V. Bellandi, M. Bezzi, P. Ceravolo, E. Damiani, and C. Hebert, "Model-based big data analytics-as-a-service: take big data to the next level," *IEEE Trans. Serv. Comput.*, 2018.

[125] M. R. Berthold *et al.*, "KNIME-the Konstanz information miner: version 2.0 and beyond," *AcM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 26–31, 2009.

[126] M. Hofmann and R. Klinkenberg, *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

Publications

[Academic Journals (Refereed)]

[1]. [1] T. H. Akila S. Siriweera, and Incheon Paik, QoS-aware Rule-based Traffic-efficient Multi- objective Service Selection in Big Data Space", IEEE Access. August 2018. (Impact factor 3.55)

[2]. [2] T. H. Akila S. Siriweera, Incheon Paik, and Banage T. G. S. Kumara, and C. K. Koswatta, Architecture For Intelligent Big Data Analysis Based On Automatic Service Composition, Int. J. Big Data, vol. 2, no. 2, pp. 1–14, 2015.

[Proceedings at International Conferences (Refereed)]

[3]. T. H. Akila S. Siriweera, I. Paik, and B. T. G. S. Kumara, "Constraint-Driven Dynamic Workflow for Automation of Big Data Analytics Based on GraphPlan," 2017 IEEE Int. Conf. Web Serv. (ICWS), pp. 357–364, 2017.

[4]. T. H. A. S. Siriweera, I. Paik, and B. T. G. S. Kumara, "QoS and Customizable Transaction-aware Selection for Big Data Analytics on Automatic Service Composition," 2017 IEEE 14th International Conference on Services Computing (SCC),, 2017, pp. 116–123.

[5]. B. T. G. S. Kumara, I. Paik, T. H. Akila S. Siriweera, and K. R. C. Koswatte, "QoS Aware Service Clustering to Bootstrap the Web Service Selection," in Proceedings - 2017 IEEE 14th International Conference on Services Computing (SCC) 2017.

[6]. I. Paik, Yutaka Koshiba, and T. H. Akila S. Siriweera , "Efficient Service Discovery Using Social Service Network Based on Big Data Infrastructure",

11th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip, Korea University, Seoul, Korea, September 18-20, 2017.

[7]. T. H. Akila S. Siriweera, I. Paik, J. Zhang, and B. T. G. S. Kumara, “Big data analytic service discovery using social service network with domain ontology and workflow awareness,” in Proceedings - 2016 IEEE International Conference on Web Services (ICWS) 2016.

[8]. T. H. Akila S. Siriweera, I. Paik, and B. T. G. S. Kumara, “QoS-Aware Traffic-Efficient Web Service Selection over BigData Space,” 2016 IEEE Int. Conf. Comput. Inf. Technol. (CIT), pp. 197–203, 2016.

[9]. T. H. Akila S. Siriweera and I. Paik, “Service selection on bigdata-space based on heterogeneous QoS preferences,” in 2016 IEEE International Conference on Consumer Electronics-Asia, ICCE-Asia 2016,

[10]. B. T. G. S. Kumara, I. Paik, T. H. Akila S. Siriweera, and K. R. C. Koswatte, “Cluster-based web service recommendation,” in Proceedings - 2016 IEEE International Conference on Services Computing, (SCC).

[11]. R. A. H. M. Rupasingha, I. Paik, B. T. G. S. Kumara, and T. H. Akila S. Siriweera, “Domain-aware web service clustering based on ontology generation by text mining,” in 7th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEEE IEMCON 2016, 2016.

[12]. T. H. Akila S. Siriweera, I. Paik, and B. T. G. S. Kumara, “Onotology-based service discovery for intelligent Big Data analytics,” in IEEE 7th International Conference on Awareness Science and Technology, iCAST 2015 - Proceedings, 2015.

[13]. T. H. Akila S. Siriweera, I. Paik, B. T. G. S. Kumara, and K. R. C. Koswatta, “Intelligent Big Data Anal199ysis Architecture Based on Automatic Service Composition,” in Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015, 2015

[14]. B. T. G. S. Kumara, I. Paik, J. Zhang, T. H. Akila S. Siriweera, and K. R. C. Koswatte, “Ontology-Based Workflow Generation for Intelligent Big Data

Analytics," in Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015, 2015.

[15]. K. R. C. Koswatte, I. Paik, B. T. G. S. Kumara, and T. H. Akila S. Siriweera, "Meta-ontology for innovative product design with semantic TRIZ," in Proceedings of 2015 International Conference on Electrical and Information Technologies, ICEIT 2015, 2015.