MASTER

Optimizing shared service center performance by assessing simulated task assignment methods

Vermeeren, W.

Award date:
2021

Optimizing shared service center performance by assessing simulated task assignment methods

Master thesis

March 21st, 2021

Eindhoven University of Technology

Department of Industrial Engineering & Innovation Sciences

Thesis supervisor dr. ir. R.M. Dijkman

Author W. Vermeeren BSc.

Student number 0814647

Study Program Operations Management & Logistics

Abstract

A shared service center (SSC) is a business unit within a larger organization, tasked with the responsibility of delivering a certain consolidated service or function to other units of the organization. An element of the design of a SSC is the way tasks are being assigned to its agents. In current literature, many ways of assigning tasks can be found. In this thesis, those task assignment methods are retrieved from literature and applied within the context of a shared service center, with the aim of optimizing its operational performance.

Specifically, the task assignment methods *First in, first out* (FIFO), *Earliest Deadline First* (EDF), *Workload Balancing* (WldB), *Worklist Balancing* (WlsB) and *Shortest Processing Time* (SPT) are selected from literature and compared in a simulation. Eight agents work on eleven different task types, with varying arrival rates and processing rates, with agents being randomly and scheduled unavailable, with different levels of workload and with varying deadlines and ways of handling tasks. The service level, as the percentage of tasks completed before the deadline, and agent utilization, as the percentage of time agents spend working, are compared for each method under varying circumstances and are used to determine the earlier mentioned performance.

The simulation shows that when agents of a shared service center are not continuously available during working hours, methods with individual agent queues lead to higher waiting times, a lower service level and a higher utilization rate, compared to methods with a single assignment queue. In different setups of a shared service center, FIFO outperforms EDF and SPT, caused by lower average lead times. If the assignment queue has grown to levels in which service levels are no longer being met, SPT is best in decreasing the workload, as tasks are finished by the fastest agent.

The conclusions enable PBNL, the host of the case study, to pick FIFO as most suitable assignment method and apply SPT when workloads have increased to an extent that deadlines are no longer met. Both recommendations fit within both PBNL's implementation scenario's: one with and one without Workflow Management System. The research also proves PBNL can initially not work with less agents than planned and that PBNL should actively prevent the inflow of tasks that cannot be finished with a single action, as these types of tasks have the largest contribution to missed deadlines.

Preface

This master thesis gives the results of the research project I performed to finish my master Operations Management & Logistics at the Technical University of Eindhoven. For the past 6 months, I have been given the chance to put all my acquainted knowledge to the test by helping a Dutch leading private bank in their preparations of a reorganization. Via this preface, I would like to express my gratitude for this opportunity.

I want to thank Remco Dijkman, my first supervisor for his help and support, and especially for the flexible and fitting way of guidance that throughout my research process kept matching with my changing personal needs. Where for a long time, I saw my thesis mainly as a major final obstacle I would have to pass purely on willpower, my project became interesting and motivating quite early, making it something I enjoyed more than I had imagined before. The smooth cooperation, good discussions and subtle moves in the right direction have helped in this and make me thankful.

Furthermore, I want to express great thankfulness to Johan Pannekoek, my company supervisor. The opportunities Johan has given me in the past few years to develop myself in a team and an environment of such a high level, is something I am extremely grateful for and which is truly inspiring. I have learned too many lessons from Johan to mention in this preface, but I would like to say that having a friend as a manager and supervisor was an experience I will never forget.

Lastly, I want to thank my family, housemates, friends and fellow students for the moral support and the amazing time I had as a student.

I hope you enjoy reading this thesis.

Wouter Vermeeren
Amsterdam, 2021

Management Summary

This master thesis shows the results of a research on the impact of task assignment method on the performance of a shared service center. In this management summary, the problem context is explained first. Then, the methodology is given. Thirdly, the task assignment methods are shared, followed by the results. Lastly, recommendations for PBNL, the company hosting the research, are given.

Problem Context

PBNL, a Dutch leading private bank, is implementing a new shared service center concept to consolidate a set of administrative tasks from its local offices, to new regional operating ‘Business Support’ (BS) centers. A shared service center is a business unit within a larger organization, tasked with the responsibility of delivering a certain consolidated service or function to other units of the organization. An element of the design of a SSC is the way tasks are being assigned to its agents. In current literature, many ways of assigning tasks can be found.

The goal of this research project is to deliver recommendations on task distribution, suitable for implementation within a shared service center, by applying currently existing distribution methods in a data-based simulation and assessing their impact on service level, defined as the percentage of tasks finished before the deadline of 4 days, and their impact on agent utilization, defined as the percentage of time Business Support agents are working on tasks. By doing so, Business Support can service their local offices with the fastest possible service, but with as few deployed agents as possible.

Methodology

This research project is structured as a case study, with PBNL as host of the research, and as a design science study. First, a literature review is performed to retrieve task assignment methods from literature. Then, a data analysis is performed, on both task behavior and agent behavior. By analyzing the worklist used during a pilot in which the new way of working of Business Support was tested, arrival rates for the 11 different task types were found. By analyzing the journal logs of four assistants, processing rates were found. Based on the implementation plans of Business Support, a conceptual queueing model was made and transformed into a simulation. The arrival and processing rates are entered into the simulation and the different assignment methods are tested. Results of this simulation have led to recommendations for PBNL.

Assignment Methods

21 different task assignment methods are retrieved from literature. When analyzed, the methods were found to be categorizable in three groups. With task-selecting methods, the most suitable task is retrieved from the assignment queue when an agent requests a task.

With an agent-selecting method, a suitable agent is found for a task when it enters the assignment queue. Tasks can be assigned to agents directly and can be placed in individual agents' queues, when agents are occupied. When a complete set of (planned) tasks and agents is available, the third category, set matching, can be applied, which results in an optimal distribution of tasks to agents.

In this thesis, three task-selecting methods and two agent-selecting methods are tested. The three task-selecting methods are FIFO, which selects the task that entered the queue first, EDF, which selects the task with the soonest deadline and SPT, which selects the task the to-be-assigned agent can perform the quickest. The two agent-selecting methods are Worklist Balancing, which assigns tasks to the agent with the least number of tasks in its queue and Workload Balancing, which assigns tasks according to predetermined assignment probabilities, aiming to balance the overall workload of agents.

Results

The simulation is verified and validated. Based on the parameters from the data analysis, the lead time generated by the simulation is only 67% of the real world found lead time. For this and other reasons, different simulations are performed, with varying circumstances.

The results of the initial simulation with the parameters from the data analysis are shown in Figure I. With a deadline of 4 days, it is expected that with the three task-selecting methods, a service level of 0.95 is achieved with an agent utilization of only 0.74. The agent-selecting methods are found to be unusable, due to the fact agents at PBNL are not continuously available during working hours and then the existence of individual agents' queues leads to a significant increase in waiting times, which lead to a lower service level. The fact that agents start working with a filled queue, after a day off, also causes a higher overall utilization rate. Of the three task-selecting methods, FIFO shows the highest service level and lowest utilization rate. However, the simulated lead time of 460 minutes per task deviates from the value of 610 minutes found in the real-world validation.

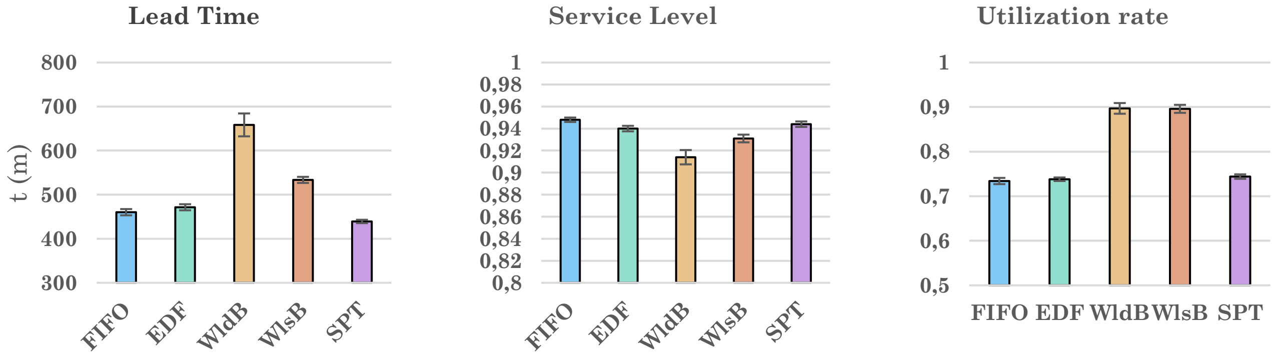

Figure I: Selection of Results for the Simulation with parameters derived from the initial data analysis

The real-world situation is further approached by altering the simulation parameters in two ways. First, the workload is increased with 20%. Then, randomly distributed unavailability is added for 8% of the agents' time, simulating random breaks. As shown in Figure II, agents now are utilized up to 95% of their time, achieving a service level of 92% with FIFO scoring best again. The three different methods, however, show relatively small differences on utilization rate and service level. When in this real-world approach setup, the eighth agent is removed from the simulation, the system becomes unstable and the number of tasks increases more rapidly than Business Support can process them.

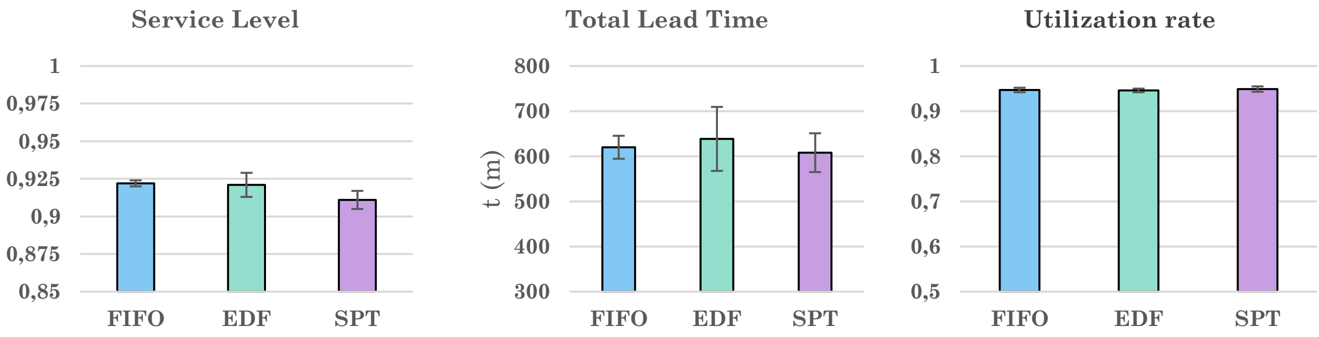

Figure II: Selection of Results for Simulation most approaching the validation (7)

From seven different test setups, additional results are found. When workload is increased with 20% by adjusting the arrival rates, the system becomes unstable for agent-selecting task methods. However, when agents are not scheduled a weekly day off, and are available continuously, the disadvantage of individual agent queues and thus the big difference between task and agent selecting methods mostly disappears. However, tasks-selecting methods still outperform the agent-selecting ones.

When the deadline of just 2 of 11 tasks is changed to 1 day, mimicking private bankers who send an urgent task to Business Support, the service level shows a drop to 0.8 for all assignment methods, except for FIFO, which drops to 0.91. Tasks that are not finished in a single action but in more steps, due to clients or other colleagues having to take action first, are tasks with a rearrival time. This rearrival time has been found to have a significant impact on the service level. When tasks rearrive after 3.5 days, instead of the estimated 2 days, the service level drops with 6% for FIFO, and with 9-12% with the other methods. The last additional founding is that when the agents' production capacity can't handle the inflow of tasks, the SPT algorithm is the quickest in in reducing the redundant tasks in the system.

Recommendations

The task assignment method PBNL chooses to implement, can be applied for both two different business scenarios, between which PBNL has not yet decided. In the first scenario, a workflow management system is implemented, allowing the assignment of tasks to take place automated and based on data. In the second scenario, there is no software system and the manager of Business Support and the agents themselves are responsible for the execution of an assignment method. In both scenario's, FIFO is recommended as the best task assignment method. As BS agents will be unavailable on scheduled moments, the agent-selecting tasks are no valid option. Within the three task-selecting methods, FIFO outperforms SPT and EDS slightly under regular circumstances, but with a bigger difference when circumstances change and for example, more tasks are sent with urgency. In the case of large unsustainable backlogs, when service levels missed anyway, SPT is recommended as it finishes tasks by assigning the quickest agent.

Furthermore, it is recommended to monitor and analyze the percentage of tasks that cannot be finished in a single action, as multi-step tasks with long rearrival times have a significant impact of the service level. It is found to be impossible to operate a BS with only 7 agents, as there is not enough capacity for the incoming tasks and the system becomes unstable. However, it is recommended to repeat the data analysis after a few months after implementation, as the possibility exist agents become quicker when they get more experienced and then efficiency still can be achieved.

Table of Content

List of Figures

List of Tables

1. Introduction

The first chapter of this thesis gives an introduction to the research. Firstly, a short description of the topic and context is given. Then, the design of the research is further explained. The third paragraph explains how the thesis contributes to the existing scientific literature. The research problem described in this thesis is approached with a case study. The organization hosting the research is introduced in the fourth section of this chapter. In the last paragraph, the research method is explained and linked to the structure of this thesis.

1.1. Topic and Context

This research focuses on the topic of task assignment within the context of shared service centers. The concept of a shared service center will be explained shortly and will be followed by some definitions with respect to tasks and task assignment.

A shared service center (SSC) is defined as *"a dedicated unit, including people, processes and technologies, that is structured as a centralized point of service and is focused on defined business functions"* (Gartner, 2021). With other words, a shared service center is a business unit within a larger organization, tasked with the responsibility of delivering a certain consolidated service or function to other units of the organization.

The IT helpdesk of an international paper company is an example of a shared service center. With all processes, responsibilities, knowledge and expertise on IT concentrated at the head office of the company, and available to all employees by phone and mail, there is no longer a need for IT service on regional offices and distribution centers. The IT helpdesk services all other business units, allowing itself to build expertise on all matters related to IT, and allowing other business units to focus on their specific functions, like sales, logistics or client support.

Shared service centers can be deployed for many kinds of services, such as human resources (HR) and accounting (Richter & Brühl, 2017). The implementation of a SSC and the corresponding consolidation of business activities can serve multiple purposes. By centralizing activities and enabling employees to specialize in a coherent set of tasks, efficiency can be achieved, costs can be reduced. Also, quality in service can be improved (Schulz & Brenner, 2010).

A SSC can service its organization in a proactive way, when for example a quality control center of a mechanical firm reaches out to its employees to remind them on training and the timely renewal of certifications. It is also possible for a SSC to service the organization in a reactive way. This happens when, for example, an employee wishes to change the bank account to which salary is deposited and sends this request to HR by opening a ticket via the intranet of the organization.

An element of the design of a SSC is the way tasks are being assigned to its employees. In order to satisfy the requirements of the internal customer, the ability to allocate the right employee to the right task at the right moment, is of a critical importance (Ernst, 2004). This task assignment problem, which is described as critical to success of a shared service center, since it influences the extent to which the desired efficiency and quality of performed tasks is achieved, is the topic of this research (Wang, 2007).

The research area of task assignment is broad and not unbounded. The annotated bibliography of Ernst (2004) gives insight in overlapping research areas, as workforce planning, demand modelling, shift scheduling and task assignment, all touching on the topics of employee availability and work allocation, but all doing so from a different perspective, with different goals, methods, input and output.

Ernst (2004) describes task assignment as the process of '*allocating tasks, or a set of tasks, with a specified start and end time and skill requirements, between a group of workers who have typically already been assigned to a set of working shifts'*. Throughout this thesis, the definition of Ernst is being followed, with the side note that end time tasks for a specific task may not always be known in advance, as the processing time of tasks are subject to certain levels of arbitrariness. End times in the form of deadlines however, still apply.

With task assignment being defined as the allocation of tasks to employees, task assignment methods are defined as sets of rules of logic, which applied on the context of the tasks and employees, guide the decision on which tasks, agents, or task-agent combinations should be next in line, or planned later in time.

Throughout this thesis, the term *agent* will be used for employees who are handling tasks and are part of the task assignment problem. In literature, the terms *worker* and *robot* are used in same or similar manners.

1.2. Research Design

This section discusses the design of the research, starting with the problem statement. Then the goal of the research is being given and lastly, the research question is stated.

1.2.1. Problem statement

The problem central in this research, is the problem of task assignment in shared service centers. In current literature, many ways of assigning tasks can be found. With the well-known *first in, first out* (FIFO) method as an example of a simple way to decide which task should be performed first, the current state of the art provides with numerous other methods, varying on aspects as needed data, applied logic and produced outcome.

Since the task assignment method is an element of the organizational design of a shared service center, it is in the interest of the implementing organization to choose a design that matches its goals. For this however, it should be known what the impact of certain task assignment methods will be on the performance of the SSC.

By addressing the research problem, this impact will be clarified. The effects of different ways of assigning tasks to agents will be investigated, making it possible to compare them on performance indicators as average lead times, percentage of timely deliveries and number of agents needed. These insights enable shared service implementors to choose a task assignment method that suites their specific needs and situation.

1.2.2. Research goal

The goal of this research as follows:

Deliver recommendations on task distribution, suitable for implementation within a shared service center, by applying currently existing distribution methods in a data-based simulation and assessing their impact on service level and agent utilization.

1.2.3. Research questions

In order to deliver the recommendations mentioned in the research goal, the following sub-questions will be addressed:

a. Which task distribution methods are given in recent, relevant literature?

b. Which task distribution methods are applicable on transactional shared service centers with a focus on administrative processes?

c. Which properties of processes and employees are necessary to perform an accurate simulation?

d. What is the impact of task assignment methods on the service level and agent utilization of a shared service center?

1.3. Scientific Contribution

The scientific contribution of this thesis lies on the cutting edge of two fields of study. On the one hand, this thesis contributes to the area of task assignment, since a unique set of task assignment methods is compared on performance. Within the existing field of literature, no comparison of the FIFO, EDF, SPT, WldB and WlsB is found. The in this thesis proposed novel way of classifying is a contribution the science and the same follows for the proposed framework for the design of a task assignment method.

On the other hand, a need on non-financial quantitative metrics of shared service centers is identified by Richter & Brühl (2017), who demonstrate a high fragmentation of academic literature about SSCs and identify multiple opportunities for future research. Where literature on the performance on the individual task assignment methods is abundant, for the specific context of a shared service center, with scheduled and random unavailability, no turnover times, random urgent tasks and absent storage or transport issues, no literature on possible task assignment methods is present, let alone being compared.

1.4. Case Study

The research is structured as a case study, in which the real implementation of a shared service center in an existing enterprise is subject of analysis and supplements the research on aspects in which real world data or information can be applied. This way, the research aims at yielding both general applicable findings and specific implementation recommendations for the host of the case study, which will be introduced in this section.

1.4.1. Company Description

The host of the research is a Dutch banking group. Seated in Amsterdam, the banking group focuses on serving European clients, both persons and businesses. Regular individuals are served by business line *Retail Banking*. Affluent customers are served by business line *Private Banking,* which itself is divided in multiple business units, all serving a single west European country.

The research takes place at the *Private Banking* business unit that serves the Dutch customer. This unit will furtherly be referred to as PBNL (Private Banking Netherlands). At this moment, all ±30,000 customers are being served by their own *Private Banker* (PB). The ± 200 Dutch bankers are distributed over 23 local offices, across the entire country and clustered in six regions. As is shown in Figure 1, a simplified conceptual organizational chart of PBNL, bankers are assisted by *Assistant Private Bankers* (APB). While being responsible for the general management of customer relations, bankers are also responsible for developing commercial opportunities and linking customers to colleagues of departments responsible for delivering a specific product or service. Assistant bankers support bankers with commercial activities and with the administrational elements of bank or client-initiated processes.

PBNL offers clients the service of investment management. When linked to a regional operating *Investment Manager* (IM), customers are offered the possibility to invest their assets in financial markets and products, with a predetermined goal and within a periodically checked mandate, adjusted to the customers wishes and risk-based limits. As shown in Figure 1, all IM's work, supported by an *Assistant Investment Manager* (AIM), on a regional level, allowing them to service all customers linked to the local offices of their region. Each regional investment center contains 6-10 investment managers and approximately 4-6 assistants.

Besides the (assistant) bankers and (assistant) investment managers as commercial staff, each region is being assisted by a non-commercial regional Operating Office (OO), responsible for several activities concerning the areas of risk management, compliance, process control and change. With six regions, accommodating the 23 local offices, there are six operating offices, all providing work to 5-7 employees.

For the entire group, three strategic pillars are given that guide developments and efforts undertaken by all underlying business lines and unit. Firstly, the bank wants to support their clients' transition to sustainability. Furthermore, the bank wants to reinvent the customer experience. Lastly, the bank wants to build a future-proof bank. Part of this last goal is the banks ambition to reduce its cost-to-income ratio to 56-58%, from 61.2% in 2019. Wherever possible, work should be organized as efficient as possible.

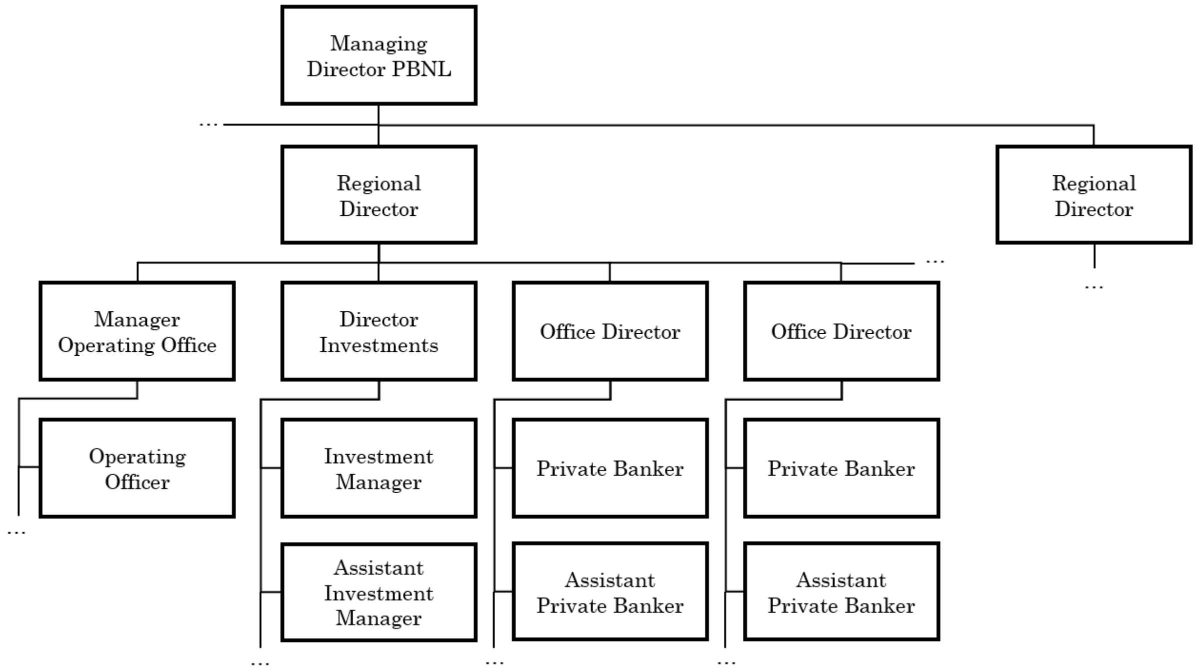

Figure 1: Simplified organizational chart of PBNL, showing functions in scope of the research

1.4.2. Problem Context

To align PBNL's business organization with the earlier mentioned strategic pillars, a project team has initiated a business model transformation in 2019. Currently, detailed plans have been developed, changing PBNL's client service model in multiple ways. One of the upcoming changes is the implementation of a shared service concept.

In the current organizational design, the functions of assistant Private Banker (APB) and assistant Investment Manager (AIM) include tasks focusing on commercial support of bankers on one hand, and on administrative processing of a wide variety of tasks on the other. In order to enable the APB and AIM to fully fulfill the commercial aspects of their function and to generate more quality and efficiency within administrative processes, PBNL has decided to relocate administrative tasks and responsibilities from the local APB's and AIM's to new, still to be established regional shared service centers, called Business Support (BS).

A conceptual overview of the implementation of Business Support, as a shared service center, is given in Figure 2, and will be explained more thoroughly. By the transformation, the six regions are merged into four new ones, shrinking the management layer of PBNL. The functions of Private Banker (PB), Assistant Private Banker (APB), Investment Manager (AM), Assistant Investment Manager (AIM) and Operating Officer (OO) have been analyzed on activity-level which has resulted into a selected list of activities that will be removed from their original functions, and will be transferred to the new regional Business Support and will there be inserted in of two new functions. The activities coming from the PB, APB, IM and AIM, all relating to clients, will together create the function of employee *Client Support* (CS). The activities coming from the Operating Office, which itself will merge completely into the new BS, will create the function of *Process Support*.

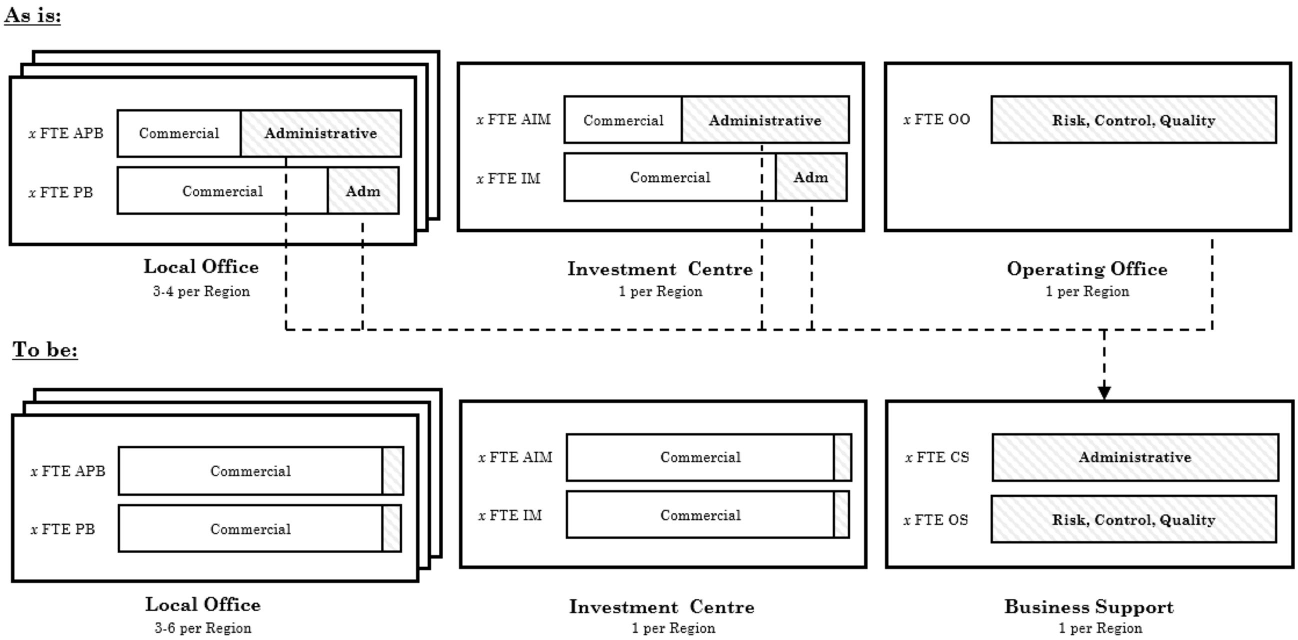

Figure 2: Conceptual representation of the implementation of Business Support

In order to understand the way BS will operate after its implementation and to understand assumptions, references and information that will be mentioned later in this thesis, this section will be concluded by a description of the tasks of the function *Client Support* and by a description of the way BS will function internally and in cooperation with its stakeholders.

Tasks and activities forming the function *Client Support* are:

1. *Set up New Client: a*ccounts are opened and set up for new clients, or for new banking entities within existing client accounts
2. *Set up New Product: n*ew products are opened and set up, or requested at departments within the banking group
3. *Change Client Data: p*ersonal information, such as phone numbers and addresses, are changed, or preferences are modified
4. *Change Banking Product or Contract: e*xisting products and products are changed. For example, a client wants to merge the saving accounts of himself, his partner and their joint account.
5. *Change Investment Product or Contract: a* change is made in an investment product. For example, a client wants to insert cash from a savings account into an investment portfolio
6. *Prepare Banker and Client Meeting:* information on client history, products and commercial opportunities is retrieved from data sources and sent to the banker
7. *Prepare 'Know Your Client' Revision:* the periodic mandatory check on client background, situation, risk and product purchase is prepared
8. *Prepare 'New Client Take On' Compliance Check:* the mandatory check on client background and the origin of assets before approval can take place, is prepared

9. *Process Client Request:* client requests with an administrative character, such as a printed statement of account balances, are being processed
10. *Start the Deceased Client Process:* start the process for deceased clients
11. *Change Authorizations:* change authorizations for clients, when for example a mother wants her daughter to take care of her bank account

In line with the property of a SSC stating that it services internal customers, agents of BS initially have no direct contact with clients of the bank. The so called ‘business’, or ’‘first layer’, consisting of the PB, APB, IM and AIM, are the employees of the local offices with direct client interaction. The majority of the mentioned tasks arise during interaction the first layer has with the client. During this interaction, the clients personal situation is discussed and advice is given. The administrational consequences of decisions made after the interactions will be passed on as tasks to BS. In addition to tasks coming from client interaction, some of the tasks arise from internal systems issuing planned or ad hoc compliance, risk and quality related tasks.

As agents working for Business Support are physically separated from employees of the local offices they serve, tasks are passed on digitally, using the banks Client Relationship Management (CRM) system. Tasks are placed in a worklist in this system, visible to all BS agents and their manager. Tasks can be picked up by agents, removing them from queue and when finished, are being sent back to the submitter.

In the current transformation plan, the topic of task assignment is still uncovered. Capacity calculations have been made, based on estimations of numbers of tasks sent to BS, average processing times of these tasks and the availability of agents, but the assignment of tasks is designated as one of the roles of the manager of BS, who can use a dashboard with limited worklist and performance data and can apply general management principles to influence performance. At this moment, a service level agreement (SLA) of four days is desired by regional management and the managers of local offices, meaning a task sent to BS should be finished within four days.

Since the success of the transformation and the implementation of BS depends on the ability of BS to serve the local offices in a timely manner, it is not enough to just check if the estimated total workload matches the overall availability. As the number of tasks sent to BS may differ for different periods of time, the experience and rapidity of agents may vary and requests with urgency may intervene with business as usual, it is needed to find a task assignment method that yields the fastest lead times and the highest service level, as defined as the percentage of tasks completed on or before the deadline.

From the point of view of the organization, the research problem can be summarized as:

“Improve the future task performance of Business Support by finding an applicable task assignment method, such that the shortest possible lead times are achieved, with the highest possible service level, in order to satisfy the demands of the commercial employees of local offices”

1.4.3. Scope

This thesis gives a simplified display of the mentioned business transformation and the organizational design. PBNL exists for over 300 years and has organically and inorganically transformed in the organization it is today. This history and the fact that the client base consists of high demanding clients with specific needs, both lead to frequent customizations of the generally applicable processes and procedures, making the organization occasionally less uniform as described in this thesis. With respect to the thesis, the following scope applies:

1. **The research only focuses on "PBNL Branches"**

In addition to the 23 earlier mentioned local offices, PBNL serves four different groups clients, segmented on deviating properties with respect to size of wealth, occupation and location. The so-called 'Special Segments' will also be served by a Business Support, but due to deviations in content of their business processes, the Business Support Special Segments will be out of scope.

2. **The research only focuses on the Client Support function of BS**

As stated before, BS will contain two kinds of functions, Client Support and Process Support. The research only focuses on the tasks of the function of Client Support. On one hand because this function embodies a vast majority (>80%) of the total activities BS, and on the other hand because the tasks of the Process Support function are less transactional and therefore less suitable for a simulation-based study and are not directly linked to the performance of BS as perceived by the employees of the local offices it serve.

1.5. Methodology and Structure

The research process of finding and testing task assignment methods within the business context of a shared service center makes the research qualify as a *design science* study, as defined by Hevner et al. (2004), since it approaches the problem of performance optimization with tested task assignment methods as artifacts. This section explains the methodology and structure of the research project. First, the different research deliverables are mentioned chronologically. Then, it is mentioned how these deliverables answer the research questions. Thirdly, the structure of the thesis is explained. Lastly, in Table 1, a combined summary of this structure is given. The way the different deliverables relate to the cycles of Hevner et. al. (2004) is also shown in this table.

After having interviews, making observations and reading company documents, the problem & company description (1) is made. By performing a literature review, an initial set of task assignment methods (2) is retrieved from literature. As the different task assignments are compared in a simulation, a conceptual simulation model (3) is created, based on theory on discrete event simulation. This conceptual model is made to determine the data requirements (4), which represent the input needed for the simulation and therefore the desired output for a data analysis. Aiming on retrieving this desired output, a data analysis, with data from the *Customer Relationship Management* (CRM) System, data from employee journals, observations and interviews lead to the simulation parameters (5), of task arrival behavior and agent behavior.

With the conceptual model and environment parameters, retrieved with the data analysis, the simulation model (6) is programmed in Python. This model is then validated and verified (7) by comparing the generated output with the real-world performance. Next, based on a requirement analysis of the implementation plans of PBNL, the initial selection of task assignment methods is narrowed down to a selection of assignment methods (8). The selected methods are then put into the simulation model and their performance is measured (9). This performance is then used to draw conclusions and write implementation recommendations for PBNL (10).

The first research question, asking which task distribution methods are given in recent, relevant literature, is answered by the literature review. The second question, asking which task distribution methods are applicable on transactional shared service centers, is answered by the requirement analysis, leading to the selection of assignment methods. The third question, asking which properties of processes and employees are necessary to perform an accurate simulation, is answered by the conceptual model, leading to the desired data analysis output and thus simulation environment parameters. The last question, on the impact of task distribution methods on the service level and agent utilization of a shared service center, is answered by the method evaluation.

After this first chapter, the second chapter gives the results of the literature review. The simulation model is described in the third chapter, first by explaining the conceptual model, then by explaining the data analysis performed to find the parameters and lastly by the validation and verification. The task assignment methods are discussed in the fourth chapter. The requirements, based on the situation of PBNL, are shared, followed by descriptions of the five selected methods. In the fifth chapter, the simulation results of eight different simulation setups are shared. In the sixth chapter, the simulation results are translated into implementation recommendations for PBNL. The conclusion of this thesis is given in the seventh, last chapter.

Table 1
Structure of thesis, in relation with steps of the Design Cycle (Hevner et. al., 2004) and the chapters

Deliverable	Activity	Input	Question	Phase	Chapter
1. Problem & Company Description	Company & Problem Analysis	Interviews, observations, company documents		Problem Investigation	Chapter 1
2. Initial set of assignment methods	Literature Review	Papers & journals	Which task distribution methods are given in recent, relevant literature?	Treatment Design	Chapter 2
3. Conceptual Framework	Model Design	Discrete Event Simulation Theory		Treatment Validation	Chapter 3
4. Data Requirements	Analysis Design	3. Conceptual Framework	Which properties of processes and employees are necessary to perform an accurate simulation?	Treatment Design	Chapter 3
5. Simulation Parameters	Data Analysis	CRM-Data, Employee Journals, Observations & Interviews		Problem Investigation & Treatment Validation	Chapter 3
6. Simulation Model	Model Development	3. Conceptual Framework & 5. Simulation Parameters		Treatment Validation	Chapter 3
7. Validation & Verification	Validation & Verification	Simulation Output & Data Analysis Output		Treatment Validation	Chapter 3
8. Selected assignment methods	Requirement Analysis	2. Initial set of assignment methods & Interviews	Which task distribution methods are applicable on shared service centers?	Treatment Design	Chapter 4
9. Method Performance	Model Evaluation	6. Simulation Model & 8. Selected assignment methods	What is the impact of assignment methods on the service level and agent utilization of a shared service center?	Treatment Validation	Chapter 5
10. Implementation Plan	Performance Evaluation	9. Method Performance, Interviews, Observations		Treatment Validation	Chapter 6

2. Theoretical Background

This chapter presents a literature study on task assignment methods. In the first section, the search protocol used to find, judge and select relevant literature is being explained. In the second section, the results of an analysis of the selected literature are shared. The last section briefly summarizes the literature review.

2.1. Search Protocol

The research question being addressed in this literature review is as follows:

Which task assignment methods exist that optimize agent-based business processes?

To answer the research question, literature is be retrieved from available sources. To find fitting articles, the research question has to be translated into a search instruction, which includes specific search terms. For the determination of search terms, the research question in divided into three different elements:

1) Business processes
2) Task assignment
3) Agent

Search engines are selected to find relevant literature covering the terms. To ensure a complete overview, the search engine selection is based on the number of available articles and the set of relevant topics covered by the engine. The selected search engines are: ACM Digital Library, Springer Journals, IEEE Xplore Digital Library and Web of Science.

Since the combined literature, together forming the initial set of search results contains articles that match the search terms, but don't offer information contributing to the research question. Selection criteria and quality assessments applied. The article has to be published in a journal or conference proceeding, written in English and full text can be accessed online. Also, the article describes or compares at least one task assignment method and the by the article mentioned task assignment method is applicable on a context of real-world agents performing time-consuming activities.

The mentioned assignment method does not aim at minimizing agent travel time or distance and it doesn't include variables describing agent or task locations. Lastly, the by the article mentioned task assignment method allows for an automated execution. This means methods in which agents' or other actors' involvement or input is needed for every single task assignment are excluded.

Table 2 gives the results of the amount of papers retrieved by applying the search terms, strategy and criteria as described. Of the 212 articles presented after using the earlier specified search queries, 42 unique articles described a task assignment method. From these 42, a final set of 18 articles describes contains the task assignment methods that meets the final criteria.

The list of articles in the final set are given in Appendix A. The full list of retrieved articles per search engine are given in Appendix B.

Table 2
Search results with number of journals, before and after applying search criteria

Engine	ACM	Springer Journals	IEEE Xplore Digital	Web of Science
	95	93	13	11
Initial set		**212**		
Criteria				
English, full-text, journals or conference papers and duplicates removed		166		
Containing task assignment method		42		
Non-spatial and applicable on context		21		
Automated execution possible		18		
Final set		18		

2.2. Analysis

This section provides an analysis of the articles found with the search protocol as described in past chapter. From the articles, 20 different task assignment methods and algorithms have been derived. These methods and their properties are shown in Table 5.

Firstly, two classification methods for task assignment methods are explained. Then, a novel way of classifying methods, based on properties of the task assignment methods that were found with the search protocol, will be given. This will be followed by a framework for the design of the task assignment business process based. Lastly, an overview and explanation on found methods and their properties is shared.

2.1.1. Existing classifications and taxonomy

With a wide variety of available techniques to assign tasks to agents, literature also provides with a way to classify the methods. In this section, the two methods presented in the retrieved literature will be explained.

Five Categories by Zhang, Colins & Shi

Zhang et. al. (2012) states task assignment methods can be placed in five categories that differ on two axes: centralized versus decentralized and auction driven versus logic driven.

Fully centralized approaches
In a fully centralized approach, a central allocator assigns tasks to a team, based on a model of the team and its members that's available to the allocator and enables it to verify the impact of the assignment with the underlying goal.

Centralized auctions
In a centralized auction, task agents place bids on tasks that are auctioned by a central agent. Based on the bids and configuration of the central auctioneer, tasks are assigned. Since information on, for example, costs, utility or lead times is given by the task agents, it is not necessary for the central auctioneer to use a model or have system information.

Distributed auctions
Distributed auctions follow the same logic as centralized auctions, but without a central auctioneering agent. Through internal coordination, the task of collecting the submitted bids and applying the predetermined rules or auction to assign the task will be performed by one of the task agents itself.

Fully distributed approaches
Within fully distributed approaches, no communication between agents is required. Each agent choses its own actions, based on its observations of the environment. Predetermined rules guide agents in their decision whether to pick up a task or not, and should also prevent contradictory behavior.

Hybrid approaches
In hybrid approaches, earlier mentioned approaches are combined. As an example, in a distributed auction, centralized coordination can be activated in certain circumstances.

Gerkey and Matarić's Taxonomy

Gerkey and Matarić (2004) present a taxonomy for the 'allocation of robots'-problem, which is a synonym for the assignment of agents to tasks. Their taxonomy contains three axes:

Single task (ST) versus Multi task (MT)
This axis makes a distinction between agents that can work on one task at a time and agents than can execute multiple tasks at the same time.

Single robot (SR) versus Multi robot (MR)
This axis describes the task type. Tasks can be performed by a single agent, but some tasks may require multiple agents.

Instantaneous assignment (IA) versus time-extended assignment (TA)
Instantaneous assignment of tasks occurs when the applied logic does not include plans or a planning for future assignments. In an time-extend assignment, future assignments are taken into account and a schedule may be generated.

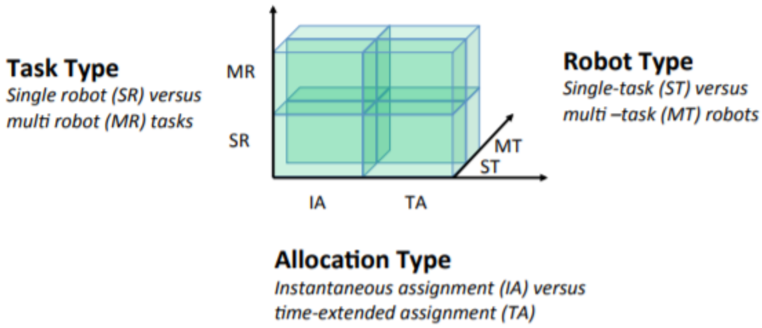

Figure 3: Visual representation of the Gerkey and Matarić's Taxonomy (Korsah et. al., 2013)

The three different axes lead, as is shown in Figure 3, to a classification set of eight different problems, denoted by the combination of the earlier mentioned abbreviation. So for example, an assignment problem for tasks that have to be performed by a single agent, who are only able to work on one task at a time and that are not able to have a planning or schedule, qualifies as a ST-SR-IA problem.

2.2.1. Output-based assignment classification

Next to the two different ways to classify the assignment methods as described in the previous section, this section discusses the way the found assignment methods differ in terms of given output and proposes this as a different way of classifying them.

The task assignment methods, which themselves will be presented in the last section of this chapter, vary on multiple properties. As an addition to earlier mentioned differences, the following distinction in output type is proposed:

Task selecting

Methods with a 'task selecting' output type continuously provide with a new task that must be assigned, chosen from a set of available, to be assigned tasks. Differences between agents may be considered, as long as they are part of a group and it is known for each task in the set of to be assigned tasks to which groups the to be assigned task can be allocated. Examples of task selecting methods are *First in, first out, Earlier Due Date* and *Shortest Processing Time*.

Agent selecting

In contrast to the task selecting assignment method, the 'agent selecting' output type takes a task as starting point and selects an agent to whom the task will be assigned. In agent selecting methods, the optimization method excludes the order in which tasks are performed as variable and thus solely achieves its aim by assigning the agent most fitting to the to be achieved goal. Examples of agent selecting methods are *Competency-driven Dynamic Resource Management, Least Loaded First* and the *BLE Assignment Algorithm*

Set matching
Where the task selecting and agent selecting methods both find a solution for a single, to be assigned task or agent, the set matching method combines both methods by considering a set of tasks and set of agents to deliver a set of assigned tasks to agents. This can be done by involving all pending tasks and idle agents in the computation. Additionally, set matching methods can also be applied to a set of planned or expected tasks. Examples of set matching methods are *Workload Balancing, SMARTCROWD* and the *Hungarian Algorithm.*

2.2.2. Framework for the design of the task assignment business process

With the two existing classification ways and the previously mentioned third possibility, this section aggregates the different classification methods and transforms their axes as design variables in a comprehensive task assignment method design variable framework. This framework is shown in Table 3.

Table 3
Framework for the Design of a Task Assignment Business Process

Design Variable			
		Objective	
Goal	*Financial*	*Non-financial*	
Outcome	*Task selecting*	*Agent selecting*	*Set matching*
		Environment	
Agent type	*Single task*	*Multi task*	
Task type	*Single agent*	*Multi agent*	
		Decision making	
Moment	*Instant*	*Time-extended*	
Method	*Bidding-based*	*Logic-based*	
Location	*Central*	*Decentralized*	

The design variables *Agent Type, Task Type* and *Decision Making Moment* are originating from Gerkey and Matarić's Taxonomy (2004). The design variables of *Decision Making Method* and *Location* are originating from the five Categories by Zhang, Colins & Shi (2012). The design variable *Outcome* is originating from the output-based classification method as described in section 4.2.

The design variable *Goal* is derived from the entire set of found methods. Having the optimization of business performance as common denominator, the way the assignment methods achieve this optimization varies. As shown by Venkatraman & Ramanujam (1986), measurement of business performance often takes place via financial indicators as profit, growth and costs but also via operational, nonfinancial indicators as quality and effectiveness.

The Framework, as shown in Table 3, represents available options for different elements of the design of the process of task assignment. It addresses the possible ways to configure the task assignment process. After analyzing the business context and selecting the desired or given options in the framework, the resulting set of design decisions furtherly guides the selection of a suitable task assignment method, since it scopes the available methods.

2.2.3. Found methods and their properties

In this section, the found task assignment methods are explained. The methods are shown in Table 4. The first two columns show the source of literature and the name of the found method. For each method, it is shown how the method qualifies for the earlier mentioned framework and its design variables in the last seven columns. The third and fourth column give additional information on the method, found in the literature. The columns *Goal* and *Input* will be explained furtherly.

Goal
As mentioned before, a difference between financial and nonfinancial goals is recognizable within the found methods. The *Hungarian Algorithm* as described by Kamrani et. al. (2010) and Springer et. al. (2010) aims at reducing costs, while the *MURDOCH Assignment* (Gerkey & Mataric, 2004) aims at optimizing fitness, defined as the extent to which the agent is suitable for the specific task.

Literature also gives other goals of assignment methods. Talib et. al. (2010) aim at improving overall performance after determining which factors contribute to the desired performance. Ha et. al. (2006) aim at increasing the utilization rate, defined as the fraction of time an agent is working on a task. Roy et. al. (2015) focus on optimizing task quality, specified by certain quality metrics, such as timeliness and amount of errors. Fu & Tari (2003) aim at minimizing the number of missed deadlines.

Input
Each method requires a specific input in order to perform its set of rules or actions. The repeatedly as necessary appointed set of tasks can come with additional requirements. The importance of the task can be needed (Zhao & Cao, 2007), as well as their priority, time in queue (Combi & Pozzi, 2006) and size (Fu & Tari, 2003).

For the set of agents, also frequently mentioned as needed input, additional information is mentioned as well, for example on their competencies (Talib et. al., 2010), skills (Roy et. al., 2015) and roles and responsibilities (Ha et. al., 2006).

Certain methods combine the set of tasks and set of agents as input and require information on a specified cost c_{ij} for i agents and j tasks (Springer et. al., 2010) or utility u_{ij} for i agents and j tasks (Gerkey & Mataric, 2004)

The methods shown in Table 4 form the result of the of literature review. Later in this thesis, based on the situation of PBNL, this table is used to choose the task assignment methods that will be tested in the simulation.

Table 4
Task Assignment Methods retrieved from literature

Source	Method	Procedure	Input	Objective		Environment		Decision		
				Goal	Outcome	Agent Type	Task Type	Moment	Method	Location
Zhao & Cao, 2007	Bidding Algorithm	Task agent request bids, resource agents submit proposal. Assignment based on highest global benefit.	Task importance, workload, compatibleness degree, bidding deadline	Provide global benefit	Agent selecting	Single task	Single agent	Time-extended	Bidding	Central
Kamrani et. al., 2010 & Sringer et. al., 2010	Hungarian algorithm	Set of fixed row operations on an agent task cost matrix	Cost(i,j) representing the cost of assigning task i to agent j	Reduce costs	Set matching	Single task	Single agent	Instant	Logic	Central
Talib et. al., 2010 & Talib et. al., 2011	Competency-driven Dynamic Resource Management	Dynamically allocation of employees based of their history of achieved business success	Goal, success demand, success supply, agent competencies	Achieve highest level of performance	Agent selecting	Single task / Multi task	Single agent / Multi agent	Instant	Logic	Central
Combi & Pozzi, 2006 & Ha, Bae, Park & Kang, 2006	First in, First out	Task that entered task queue first, gets assigned first	Time in queue	Assign every task	Task selecting	Single task / Multi task	Single agent	Instant	Logic	Central
Combi & Pozzi, 2006	Temporal Scheduler	Sort tasks based on deadline and duration, assignment based on role, availability, workload and completion times.	Agent availability, task duration, list of resources, list of (prioritized) tasks	Assign tasks, considering constraints	Agent selecting	Single task / Multi task	Single agent	Time-extended	Logic	Central
Pereira, Varajao & Uahi, 2020	Personal characteristics Work Distribution	Assign the most suitable agent to the job, based on personal information (Big 5, MBTI, Raisec)	RAISEC-profiles for jobs and agents	Maximize task agent fit	Agent selecting	Single task	Single agent	Time-extended	Logic	Central
Ha, Bae, Park & Kang, 2006	Shortest Processing Times	Task that has the shortest processing time, gets assigned first	Processing time	Assign every task	Task selecting	Single task / Multi task	Single agent	Instant	Logic	Central
Ha et. al., 2006 & Turner, 2018 & Mitiche et. al., 2003	Earliest Due Date	Task that has the earliest due date, gets assigned first	Due date	Assign every task	Task selecting	Single task / Multi task	Single agent	Instant	Logic	Central
Ha et. al., 2006 & Hwang, Choi & Kim, 2011	Least Slack Time first	Task with the smallest slack time has the highest priority	Release time, deadline, execution time	Increase utilization rate	Task selecting	Single task / Multi task	Single agent	Instant	Logic	Central
Ha, Bae, Park & Kang, 2006	Workload Balancing	Analyse process behaviour stochastically, forecast agent workload, plan by linear program	Customer arrival rate, set of tasks, set of agents, execution probability, agent responsibilities, service rate	Balance workloads	Agent selecting	Single task	Single agent	Time-extended	Logic	Central
Ha, Bae & Kang, 2004	Stochastic Workload Balancing	Transform processes in queueing network models, calculate server utilization, balance workload	Customer arrival rate, set of tasks and agents, execution probability, agent responsibilities, service rate	Maximize overall process efficiency	Agent selecting	Single task	Single agent	Time-extended	Logic	Central

Source	Method	Procedure	Input	Objective		Environment		Decision		
				Goal	Outcome	Agent Type	Task Type	Moment	Method	Location
Gerkey & Mataric, 2004	BLE Assignment Algorithm	Find agent-task pair with highest utility and remove them from consideration	Utility(i,j), representing the utility of agent i for task j	Optimize utilization rate	Agent selecting	Single task	Single agent	Instant	Logic	Central
Gerkey & Mataric, 2004	MURDOCH assign-ment	Assign task to most fitting available agent	Fitness(i,j) representing the fitness of assigning task i to agent j	Optimize fitness	Agent selecting	Single task	Single agent	Instant	Logic	Central
Dickerson et. al., 2018	Optimal non-adaptive algorithm	Assign task based on probability the agent-task combination yields the highest reward	Set of tasks, set of agents, value of task-agent combinations	Maximize profit	Agent selecting	Single task	Single agent	Instant	Logic	Central
Dickerson et. al., 2018	Greedy Algorithm	Adaptive algorithm, greedy assignment based on uniform weights and integral arrival rates.	Set of tasks, set of agents, value of task-agent combinations	Maximize profit	Agent selecting	Single task	Single agent	Instant	Logic	Central
Dickerson et. al., 2018	Adaptive algorithm	Assign task based on adaptive probability the agent-task combination yields the highest reward	Set of tasks, set of agents, value of task-agent combinations	Maximize profit	Agent selecting	Single task	Single agent	Instant	Logic	Central
Roy, Lykourentzo, Thirumuruganthan, Amer & Das, 2015	SMART-CROWD	All task-agent pairs are placed in an index, an overall value function is created, which is solved as a LP.	Set of agents, skills and tasks. Quality and cost thresholds, agents profiles, task values	Maximize task quality, minimize cost	Set matching	Single task	Single agent	Time-extended	Logic	Central
Fu & Tari, 2003	Least Loaded First - Priority	Task dispatcher assigns tasks to the server with the least flow time for that task, based on size and deadline priority	Task size, Deadline, Flow time, Slow down time	Minimize missed deadlines	Agent selecting	Single task / Multi task	Single agent	Instant	Logic	Central
Pihney, Doucette & Cohen, 2014	Distributed Multiagent Resource Allocation	Task and proxy agents interact in order to create a task agent plans with highest possible value	Tasks, task agents, resources, proxy agents	Increase global utility	Set matching	Single task / Multi task	Single agent	Instant	Bidding	Decentral
Zhang, Collins & Shi, 2012	Stochastic Clustering Auction	Based on stochastically moving and swapping tasks between clusters assigned to the agents	Set of agents, set of tasks, costs of tasks assigned to agents	Minimize total costs	Set matching	Single task	Single agent	Time-extended	Logic	Central
Mitiche, Boughaci & Gini, 2003	One Step Lookahead	Tasks allocated based upon near future consequences	Set of agents, set of tasks	Maximize possible assignments	Agent selecting	Single task	Single agent	Instant	Logic	Central

3. Simulation Model

This chapter discusses the model that is used to test different task assignment methods. Firstly, the conceptual model will be explained. Then, the results of a data analysis will be presented. For the first two sections, first an explanation on the applied methodology will be given, followed by the results. Based on the first two sections, the third section discusses the actual simulated model, its warm-up time, its validation and its verification.

3.1. Conceptual Model

Conceptual modeling is defined as *'the abstraction of a model from a real or proposed system'* and is recognized as the most vital part of a simulation study. A conceptual model is a simplified representation of a real situation, based on a problem, certain requirements and a definition of what is going to be simulated (Robinson, 2006). Robinsons (2013) framework of a conceptual model, shown in Figure 4, is applied for the simulation. The experimental factors form the input of the simulation. The simulation itself has content manipulating this input, together creating the output. The first paragraph of this section focuses on the input and content, the second paragraph focuses on the output.

In addition to the applied framework of Robinson (2013), it is important to state the simulation has the structure of a discrete-event simulation. With a discrete-event simulation, the modelled system operates as discrete sequence of events. The state of the system is being changed by events that occur at a particular moment in time. In the periods between these moments in time, the system does not change (Sharma, 2015).

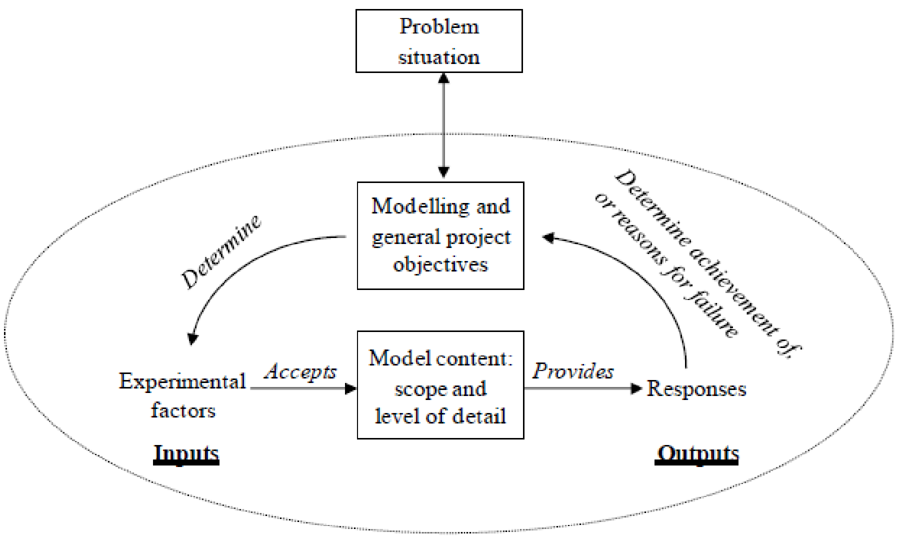

Figure 4: Framework of a conceptual model (Robinson, 2013)

3.1.1. Input and Content of Conceptual Model

The conceptual model for the discrete-event simulation used to test assignment methods in a shared service center is based on the structure and elements as described by Fishman (2001) and is shown in Figure 5. The elements of the conceptual model will be discussed briefly.

The model describes a service center as a system of tasks, agents and queues. The model is scoped by the organizational boundaries of the shared service center, meaning only behavior inside the SSC is modeled and the input the service center gets from its surrounding stakeholders is being represented mathematically. The number of task types entering the system and number of agents working on them correspond with reality, making the possible achievable level of detail matching the actual situation.

Tasks
All tasks that enter the shared service center are categorized in one of *N* task types, based on its nature and the steps required for the agent to process the task. In the case of PBNL, the different kind of tasks correspond with the list of activities as given in the first chapter, with '*Set up New Product*' and '*Change Client Data*' as two examples of those tasks. Each task type comes with a set of uniform attributes applicable to all tasks of the type that occur uniquely. An exhaustive list of properties and recorded data on tasks in shared later in this section, in Table 6. When arriving to the system, at t_a, each task is given a deadline t_d. This deadline equals the arrival time plus the service level sl_n of task *n*. The service level is the length of time the shared service has to finish tasks of type *n* in order to count as delivered on time. In the case of PBNL, the service level is recorded in the Service Level Agreement (SLA), between the shared service center and all combined local offices.

Besides the task type, categorizing the task on the set of actions needed to finish it, tasks are also categorized in a second way. In a shared service center, tasks can be brought to completion by an agent in one single action. These tasks are called single-step tasks (SST). There is also the possibility that when an agent starts working on a task, it can only be brought to a certain level of completion and depending on factors outside the influence of the agent or the shared service center, the task can be finished at a later moment in time. Within PBNL, for example, this occurs when a local banker sends a request to Business Support to prepare the *Know Your Client Revision,* the agent observes the annual figures of the customers enterprise is missing in the data and then has to request this with the client, leading to the task being finished half and being completed fully when the information has been received. Tasks that may be done in multiple actions, separated from moment in time, are defined as multi-step tasks (MST).

Arrival rate λ_n
With tasks as unique data entries in the simulation, their emergences are the events that makes the system change form. The emergence of tasks, and specifically the time between two following emergences of the same task type, takes place in a patterned behavior that when analyzed, can be described in the form of an arrival rate. Each task type *n* has its own arrival rate λ_n.

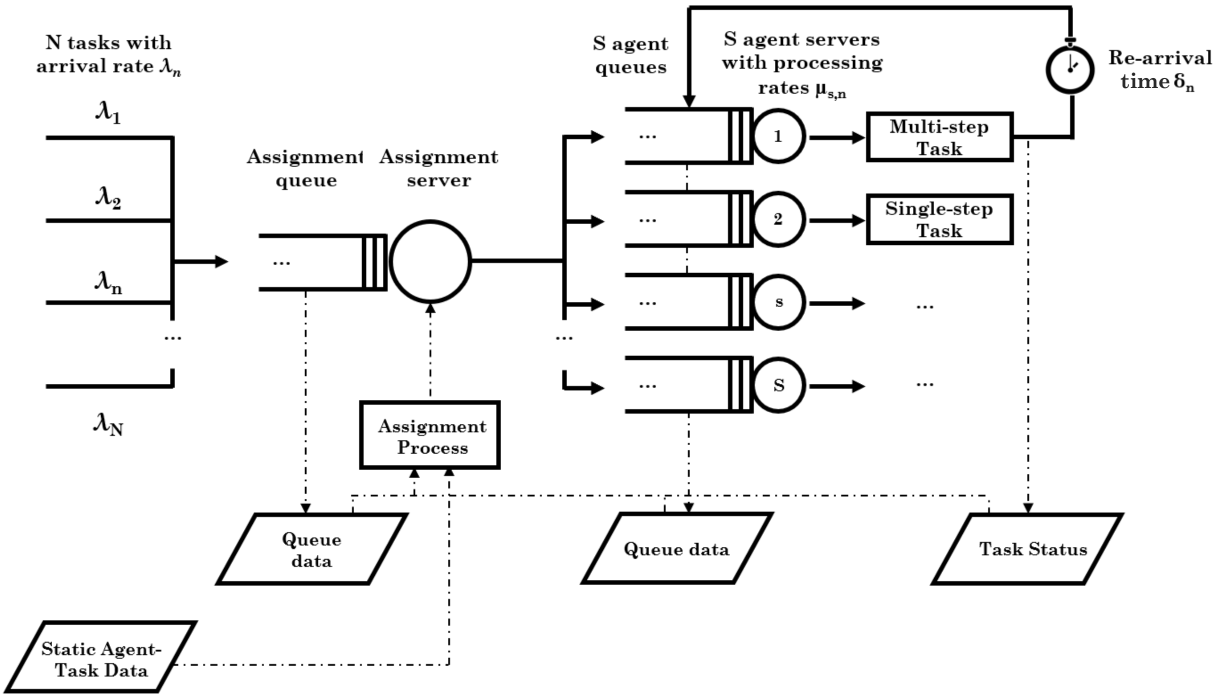

Figure 5: Conceptual model of the shared service center used in the simulation

Assignment queue
When tasks enter the system of the shared service center, they enter the assignment queue. Depending on the task assignment method, tasks stay in the assignment queue till an agent becomes idle, or tasks are placed in one of the agent queues directly.

Agent
Agents are responsible for handling tasks. Each unique agent *s* is part of the total set of agents S the shared service center has access to. With a fixed set of agent attributes, each agent may have individually different properties for these attributes. A more experienced agent may be placed in a higher salary scale and be of more cost and a less experienced agent may be less rapid in handling certain tasks. Agents can only work on one task at a time, assigned to them by the assignment server. When not working on a task, agents become idle, meaning they are available to start working on task the moment it gets assigned to them.

Assignment Server and Assignment Process
When tasks enter the system in the assignment queue, the assignment server decides when and where the task will go next. Just as an agent, who operates on task by taking actions to finish it, the server agent operates on tasks when they pick them from the assignment queue and assign them to an available agent or put them in one of the agents queues. The assignment server follows the logic of the applied task assignment method in its process of picking tasks from the queue and assigning them next.

Agent queues
Agent queues are queues filled with tasks that are already assigned to agents but have not yet been started by agents. Agent queues can be a design variable of a task assignment methods, as some methods place tasks in the queue of an agents at the moment they arrive in the system, and other tasks only assign to agents who have become idle.

Processing rate $\mu_{s,n}$
The process of an agent handling a task is represented in the model in the form of time passing by and the agent 'working on the task' being unavailable. An analysis on this period of time, also called the processing time t_p, allows for the processing behavior to be represented by a rate μ.

Since different handling times for different types of tasks may exists, but within the same group of agents, the processing time may also vary, the processing rate is modeled on task-agent level, by rate $\mu_{s,n}$, for agent s, working on task n.

Rearrival rate δ_n
As stated before, multi-step tasks are tasks which most often can not be completed in one single action by the agent and which are finished during a following action at a later moment in time. This may be the case when a colleague or client has to deliver information or take steps in a process. When the first step has been finished, completion of the task is out of influence of both the agent and the SSC, the agent becomes idle again. The task can be finished when the dependence on the other actor is raised. When this occurs, the task reenters the system. In this conceptual model, the task is placed in the queue of the agent who started working on it. In the situation of PBNL, it is declared as undesirable to have different agents working on the same task, such that status information on the task can be remembered by the first agent and writing and reading time can be saved. The rearrival rate, which is similar for each task but independent from the agent who initiated the task, is represented by rate δ_n for task n.

Queue data
As input to the assignment process of the assignment server, queue data is being recorded. When a task enters the assignment queue, it is recorded in the queue data, together with the moment it entered and the time it is queued. The same data is being recorded for the agent queues, enabling the assignment process to take into account the planned tasks and workload of each agents.

Task status
As shown in Table 6, multiple properties and attributes are being recorded for tasks that enter the system. The task status is mentioned separately, as it specifically records the status of multi-step tasks. When a multi-step task reenters the system, the assignment server takes into account whether a task is completely new or is already partly finished and should be put back in the queue of the agent who started working on it.

Static agent-task data
Additional information on an agent-task level is required by some task assignment methods and can be retrieved from the problem situation. An example of agent-task data, is cost $c_{n,s}$, representing the cost of task n being performed by agent s.

A summary of the input the conceptual model requires from the problem situation in order to successfully run a event-based simulation is displayed in Table 5.

Table 5
Summarized data input for conceptual model, for N task types and S agents

Task type	Sort	SL	λ	μ_1	μ_2	μ_s	μ_S	δ_n
1	SST, MST	sl_1	λ_1	$\mu_{1,1}$	$\mu_{2,1}$	$\mu_{s,1}$	$\mu_{S,1}$	δ_1 / -
2	SST, MST	sl_2	λ_2	$\mu_{1,2}$	$\mu_{2,2}$	$\mu_{s,2}$	$\mu_{S,2}$	δ_2 / -
…	…	…	…	…	…	…	…	…
N	SST, MST	sl_N	λ_4	$\mu_{1,N}$	$\mu_{2,N}$	$\mu_{s,N}$	$\mu_{S,N}$	δ_N/ -

3.1.2. Output of Conceptual Model

The simulation study focuses on performance, expressed in average lead times and service level, defined as the level of tasks performed on, or before the deadline. This goal shapes the way responses are being recorded.

Still following the approach of Fishman (2001), the simulation study executes the conceptual model a given number of times and records the performance metrics as responses, for each run. The overall simulation approach will be further discussed in the third section of this chapter.

During a single run of the simulation, information is recorded on task level and on system level, describing the performance of the run. The recorded information on task level is shown in Table 6. The metrics displaying the system performance during a single run of the simulation are shown in Table 7.

Table 6
During the simulation recorded task information

Task number	Unique ID
Task type	*1, 2, … N*
Arrival time	Timestamp of task entering system
Status	*1*: MST of SST task with only 1 action left *2:* Unstarted_MST
Assign time	Timestamp of task being assigned to agent or agent queue
Starting time	Timestamp of task being started for first time
Finish time	Timestamp of task being finished completely
Agent	Agent who processed task
Deadline	Arrival time plus service level
Total waiting time	Total time task waits in queue
Total processing time	Total time task is being processed by agent
Overdue time	Total time between deadline and actual finish time
<u>*For multi-step tasks:*</u>	
Starting time (*1st, 2nd)* step	Timestamp of starting with 1st or 2nd step of task
Ending time (*1st, 2nd)* step	Timestamp of ending with 1st or 2nd step of task
Waiting time (*1st, 2nd)* step	Waiting time before 1st or 2nd step of task
Processing time (*1st, 2nd)* step	Time 1st or 2nd step of task is being processed by agent
Rearrival moment	Time task reenters the system after the 1st step is finished

Table 7
Performance metrics recorded as response of the conceptual model

Tasks entered [1]	Number of tasks that entered the system
Tasks finished [1]	Tasks that have been fully finished
Tasks on time [1]	Tasks that have been finished before or on the deadline
Tasks overdue [1]	Tasks that have been finished after the deadline
Service level [1]	Tasks on time, divided by total tasks finished
Average waiting time [1]	Average time a task waits in a queue
Average processing [1]	Average time a task is being processed by an agent
Run time	Finishing time of last task minus the starting time of the first task in the simulated interval
Utilization rate [2]	Time spent processing tasks divided by total working time of agent s

[1] specified on system and task type level, [2] specified on system and on agent level

3.2. Data analysis and retrieval

With the conceptual framework as presented in previous section, the next step in in the simulation process is to fill it with such values it best represents the problems situation. The process of getting those values consists of three elements. Most importantly is the element of data analysis. By analyzing real data from the problem situation, existing patterns and behavior is retrieved and used for the simulation. On elements where data is not adequate or available, simulation input can be derived with the second element, consisting of observations and interviews with stakeholders. As third and last element, assumptions can be made to cover potentially remaining data requirements.

This section presents the method and results of the data analysis and retrieval that is performed to generate the needed input for the simulation. First, the methodology of the data analysis will be presented. Then, the process and results of the task arrival behavior analysis will be described. Lastly, the task processing analysis will be given.

3.2.1. Methodology of Data Analysis

Research activities, data sources and analyses performed to get the required information, are shown in Table 8. It is important to state that for PBNL, the concept of a shared service center does not yet exist. The data used to generate the model is mostly derived from actual operations and supplemented with figures from the implementation plan. As the granularity of used and generated data may vary, an explanation on applied extrapolations or generalizations is given in Figure 6 and will be explained later in the section

As is shown in Table 8, the handling rate $\mu_{s,n}$,for agent s on task n is retrieved by an analysis of journals of employees and complemented by data from held workshops. This gives actual handling rates for the four employees who held by a journal for four weeks and gives a range of possible handling rates that can be used to make estimations for the total group of agents of BS.

Table 8
Research activities, data sources and analyses performed for simulation input

	Task type	$\mu_{s,n}$	λ_n	δ_n
1	*Set up New Client*	*Journal Analysis & Workshops*	*Arrival analysis: pilot data*	*n/a*
2	*Set up New Product:*			
3	*Change Client Data*			
4	*Change Banking Product or Contract*			
5	*Change Investment Product or Contract*			
6	*Prepare Banker and Client Meeting*		*Client Meeting Target*	*Interviews & Observations*
7	*Prepare 'Know Your Client' Revision*		*Client Compliance Calendar*	
8	*Prepare 'New Client Compliance Check*		*Arrival analysis: pilot data*	*n/a*
9	*Process Client Request*			
10	*Start the Deceased Client Process*			
11	*Change Authorizations*			

For the arrival behavior λ_n of task *n,* pilot data is combined with the client meeting target and the client compliance calendar. During a period of eight months, four agents piloted the way of working of Business Support, meaning their work got sent to them by a worklist in the CRM-system. Data from the pilot is analyzed to get information on the arrival behavior of all tasks, with the exception from tasks *6* and *7*.

As stated before, tasks can occur in a reactive way, when they initiate during client contact. Tasks 6 and 7 occur in a proactive way. For this reason, the number of times task *6*, the 'preparation of a client meeting', must be prepared is derived from the total commercial target of client meetings for all employees of the region BS serves. The number of times task *7*, the preparation of a 'Know your Client compliance revision' must be prepared is derived from the internal compliance calendar. For this reasons, their arrival behavior is excluded from the pilot data.

The rearrival rates δ_n for tasks *n,* only applicable to multi-step tasks *6* and *7* are derived from interviews with bankers, assistants and managers.

The applied data resources do not match or represent the number of agents working for BS or the number of clients it will serve. For this purpose, the data is being extrapolated and generalized in a way that is displayed in Figure 6. For the simulation, a fictional Business Support will be constructed, averaging the number of clients and agents of the four to be implemented Business Supports.

The data from the pilot contains arrival behavior of tasks coming from 890 clients. The average number of clients served by a region equals 8900. This ratio is used to transpose the arrival behavior of the tasks. This means, the number of activities happening per period of time is multiplied with 10.

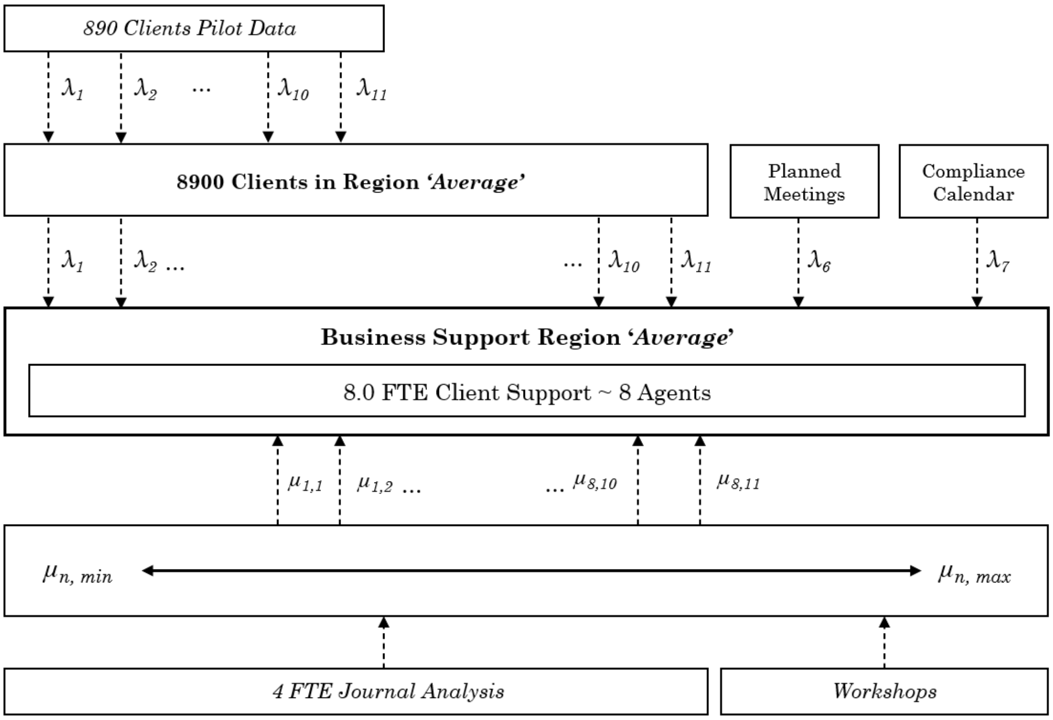

Figure 6: Conceptual representation of applied extrapolation and generalization of data resources for the construction of an average Business Support

The arrival behavior of task 6, preparation of client meetings, correlates with the number of clients of the region, as they are all aimed to meet at least once a year. The meetings are distributed uniformly over the year. The arrival behavior of task 7 correlates with the number of clients that have investments managed by the bank and thus have to collaborate with the yearly Know Your Client process. This arrival pattern follows an uniform distribution.

The processing rates $\mu_{s,n}$ of by agents S are per task determined, following a uniform distribution that is derived on the journals of 4 agents and on workshops held with other employees. The range of the distribution is determined by the minimal processing rate and the maximum processing rate. This range is then used to create semirandom fictional processing rates for four additional agents.

3.2.2. Results of Arrival Behavior Analysis

Conform the methodology described in previous section, this section shows the outcome of the analysis of arrival behavior, performed on pilot data and general information. The entire analysis can be found in Appendix 3.

For a period of eight months, four client facing employees sent tasks to four assistants using a worklist in the CRM-system, piloting the way the new Business Support will receive its tasks. During this period, 872 tasks have been placed in the worklist and have been finished by the employees.

Each task has been categorized to one of the 11 task type categories as described earlier. For each type, the average interarrival time calculated, except for tasks 6 and 7, of which the arrival behavior will be calculated differently. Histograms showing the distribution of interarrival times of each task are shown in Figure 7.

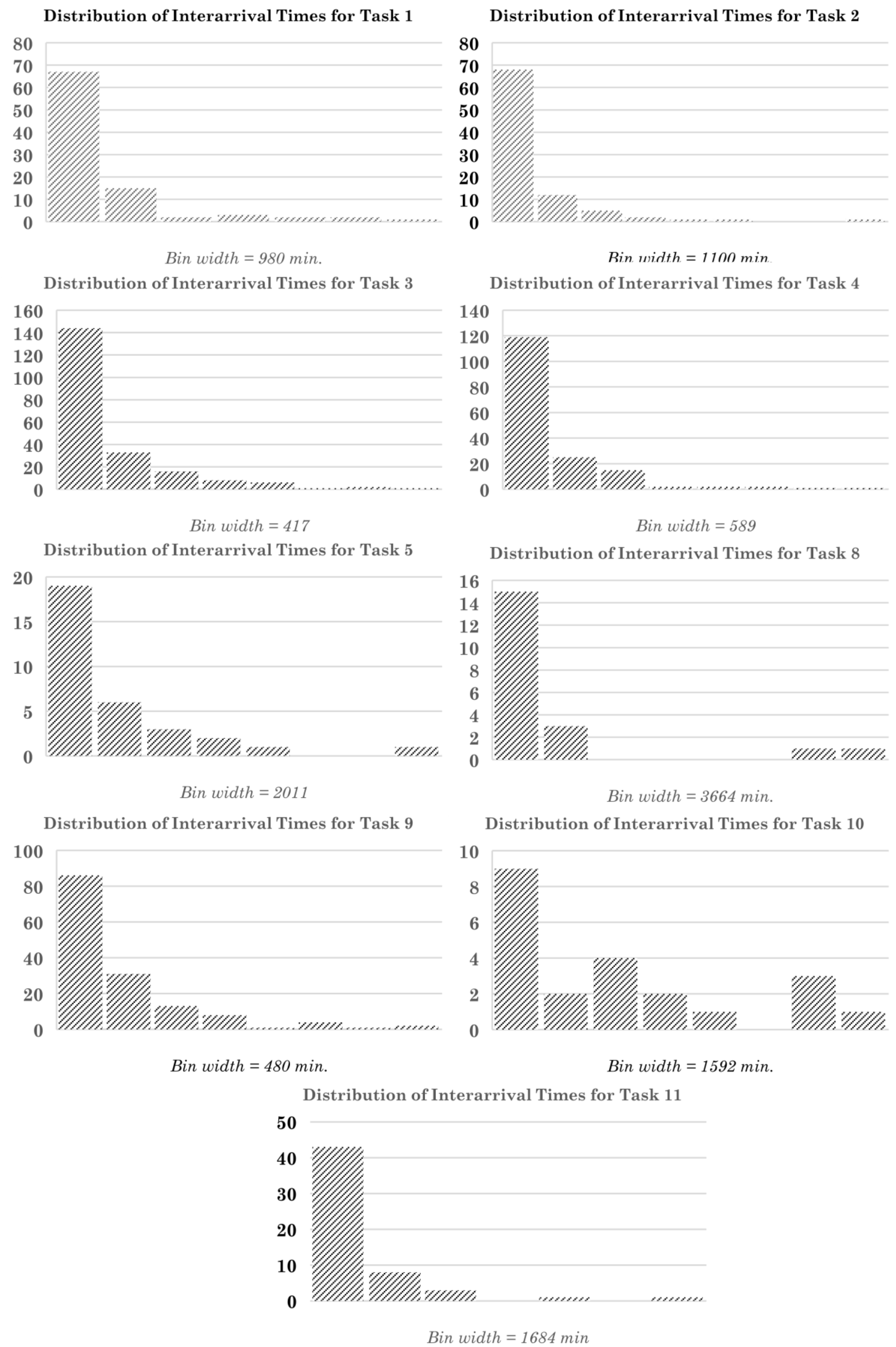

Figure 7: Histograms of the distributions of interarrival times T_{ia} for tasks in Table 10.

The arrival behavior visible in the histograms of Figure 7 appear to follow an exponential distribution. To test whether this is the case, chi-square goodness-of-fit tests are performed, for all distributions. As shown in Appendix 3, for all found arrival rates, the exponential distribution stays below the $\alpha = 0.05$ critical value of χ^2 is 16.81, meaning all tests are passed.

The tested rate λ_n however, applies to the behavior during the pilot, which does not match the behavior of the actual Business Support, since the pilot lasted 8 months and served only 890 clients, while a full BS serves 8900. By transforming the number of occurrences and thus interarrival times to a period of 12 month, and a client base of 8900, the actual expected number of arrivals for region 'Average' is given in the column *#regi*o, and the expected arrival rate for region 'Average' is given by the last column.

As stated before, all tasks except 6 and 7 are of a reactive character, meaning they occur due to an externally arisen need. Tasks 6 and 7 however, occur proactively, on initiative of the bank, and planned. The arrival behavior is based on the internal compliance calendar and the planned number of client meetings. Input of both sources has been transformed into an arrival rate, in a way that is shown in Appendix 3.

Combined with the already determined arrival rates, the final set of arrival rates is determined and shown in Table 9. The arrival rates correspond with a group of 8900 clients. Each implemented Business Support will serve a region with slight changes in client number. The arrival behavior for each regional Business Support will therefore be different.

Table 9
Arrival rates as result of the arrival behaviour analysis

Task	λ
1. Set up New Client	0,01101
2. Set up New Product	0,01042
3. Change Client Data	0,02588
4. Change Banking Product or Contract	0,02023
5. Change Investment Product or Contract	0,00389
6. Prepare Banker and Client Meeting	0,04529
7. Prepare 'Know Your Client' Revision	0,02989
8. Prepare 'New Client Compliance Check	0,00236
9. Process Client Request	0,01692
10. Start the Deceased Client Process	0,00269
11. Change Authorizations	0,00665

3.2.3. Results of Agent Behavior Analysis

To get expected values for processing times, necessary to simulate Business Support, an analysis on current processing times is performed and workshops are held. This section describes the results of the analysis and shows the results of a calculation to extrapolate the analysis agents to a full set of agents. The entire analysis and extrapolation is given in Appendix 4. This section also explains how the earlier mentioned rearrival rates are determined, and how agent availability is controlled.

For a period of four weeks, four assistants kept a journal. After finishing a task, employees noted how much time they spent working on the task, and what the content of the task was. Combined, a total of 1,777 journal entries are recorded. For each employee, the average processing time of each task type is calculated. During workshops, held with assistants, bankers, process consultants, and directors, estimates on processing times were also obtained.

With the average processing times and workshop information, for each task, a range is made using the minimum and maximum values. For each task, the processing time per agent is compared to the average of all four. This way, the relative speed per task per agent is calculated. The average, minimum and maximum relative speeds are used to semi-randomly fill the gaps and create processing rates for another four fictional agents. The entire process, shown in Appendix 4, leads to a processing rate μ for each agent and each task. The rates are shown in Table 10.

Table 10
Processing rates for all tasks and agents

	Agent							
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1. Set up New Client	0,027	0,039	0,042	0,040	0,032	0,037	0,029	0,034
2. Set up New Product	0,100	0,067	0,111	0,100	0,096	0,092	0,083	0,092
3. Change Client Data	0,043	0,053	0,045	0,067	0,054	0,045	0,056	0,039
4. Change Banking Product or Contract	0,056	0,063	0,037	0,063	0,063	0,040	0,056	0,043
5. Change Investment Product or Contract	0,062	0,033	0,042	0,067	0,042	0,060	0,046	0,048
6. Prepare Banker and Client Meeting	0,030	0,036	0,028	0,030	0,035	0,032	0,030	0,037
7. Prepare 'Know Your Client' Revision	0,021	0,020	0,024	0,024	0,023	0,023	0,022	0,022
8. Prepare 'New Client Compliance Check	0,100	0,100	0,100	0,100	0,100	0,100	0,100	0,100
9. Process Client Request	0,067	0,111	0,067	0,053	0,067	0,099	0,070	0,070
10. Start the Deceased Client Process	0,040	0,029	0,029	0,036	0,038	0,033	0,035	0,032
11. Change Authorizations	0,200	0,085	0,100	0,059	0,104	0,089	0,093	0,092

Multi-step Tasks and their Rearrival rates

As described in the conceptual model, task 6 and task 7 are defined as multi-step tasks. This means the agent starting the task, finishes a first part, but then has to wait for an external reaction before the task can be finished.

Task 6 and task 7 are mult-step tasks. For the preparation of a client meeting and the preparation of a NCTO-revision, input of respectively the banker and the client is needed. Based on information from workshops, average rearrival time for these tasks equal 2 working days. The processing rates for tasks 6 and 7 as shown in Table 13 apply for the tasks as a whole. Based on workshop information, the work is approximately split in two equal parts, before and after the rearrival time. This means the processing rate used in the simulation is double the rate as shown in Table 13.

Agent availability

As described earlier, the simulated BS employs 8 agents, corresponding with 8FTE. Agents, however, are not constantly available to work on tasks. Agents may take days off, can be ill or may have training. Managers of PBNL have indicated employees are actually working for 80% of the contracted time. This is processed in the model in the form of a schedule. Each agent is unavailable for one day of the week, creating a maximum availability of 80%.

Senior management has also indicated of the time at work, agents spend 90% of the time actually working, and spend 10% on tasks as chatting, getting coffee, helping colleagues and running errands. This random unavailability is included in the model in the form of an unavailability arrival rate λ_u and duration rate μ_u. With agents being at work for 32 hours (80%) of the 40 hours of the week, the weekly unavailability per agent is 3,2 hours, as this is 10% of 32. With an assumed unavailability duration of 15 minutes, occurring every 200 minutes, the weekly unavailability per agent exactly matches 3,2 with an applied unavailability arrival rate λ_u of 0,005 and unavailability duration rate of 0,0625.

3.3. Simulated model

This section gives information on the simulation model, which is the conceptual model as described in the first section, filled with the data from the second section and transferred into code in programming language Python. The pseudocode of the applied models can be found in Appendix F. This section first explains the applied warm-up time for simulation and then explains how the simulated model is verified and validated.

3.3.1. Warm-up

The design of the simulation is being concluded by the decision on simulation parameters warm-up time, number of runs and duration per run. A warm-up time is needed as queues are empty in the beginning of the simulation and this impact on performance should be excluded from the measurements. To determine the required warm-up time, the model is run, for a simulated timespan of 6 months, as shown in Figure 8. Based on the numbers in the queue and the average time tasks spent waiting before being processed, it is visually determined that a warm-up of two weeks, (4800 m) is a safe starting point.

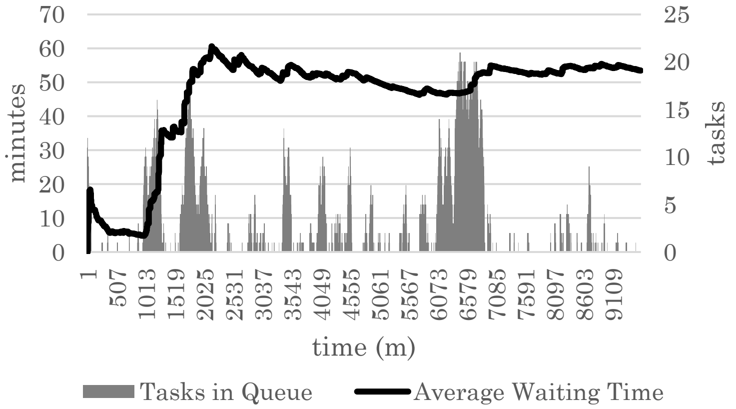

Figure 8: Que Length and Average Waiting Time for a FIFO-simulated Business Support Simulation, used to determine the needed warm-up time

3.3.2. Validation & Verification

With verification of the model, it is checked whether the model actually simulates the behavior as specified by the requirements and design specifications. With validation of the model, it is checked whether the model simulates the behavior occurring in the real world (Sharma, 2015).

The model is verified using a statistical analysis. By verifying the model, it is checked whether the number of generated tasks and the average processing times of the simulation output correspond with the data from which the simulation parameters are derived. This verification follows the approach of Hoad & Robinson (2010), which means the simulation is run multiple times and for each run, a new average value is generated, based on the new run and all preceding ones. For each run, and thus for each new average value, a t-test is performed, verifying whether the difference between the actual average, from data, and the simulated average concur enough to fall within the significance level. This approach does just not verify the model for a single run, it also find a needed amount of runs to get an average value that after a number of runs will continue to fall within the significance interval. The entire t-test can be found in Appendix G. After 25 runs, the t-test generated t-level for generated tasks (1.693) stays below the critical value (1.71) for 14 consecutive runs, and the t-level for processing time (1.378) stays below the critical value (1.71) for 5 runs. This means, the model is verified with $\alpha = 0.05$ and the number of required simulation runs amounts 25.

The validation of the model has taken place by comparing the performance during the pilot with the performance during a run of the model, with a FIFO assignment method. From the pilot data and the model data, lead times, calculated as time between entering and leaving the system, are measured per task. The average lead time during the pilot equals 679 minutes. It should be noted all data entries exceeding two weeks (15%) were unconsidered, as the worklist during the pilot was not used the same way it will be used after the implementation. Unfinishable tasks were not closed in the CRM system, tasks were left open deliberately for monitoring reasons and tasks were kept in the worklist for weeks or months without action being taken to close or finish them. The average lead time during the simulated run equals 510 minutes. The histograms for both lead times are shown in Figure 9. While the real-world performance exceeds the simulated behavior with 33%, it should be noted the research goal is to compare task assignment methods, and with lead times in the same size order, the model can do this in a valid way.

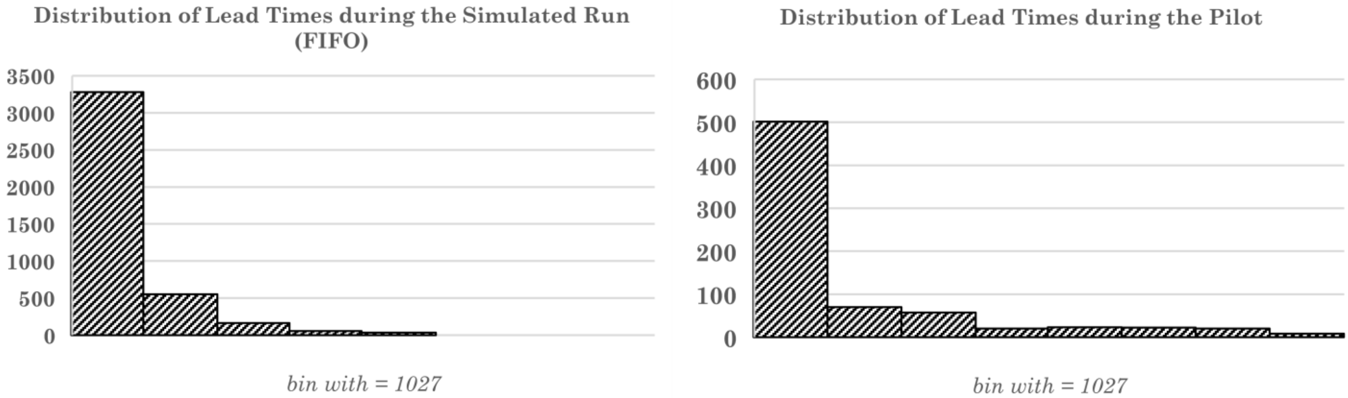

Figure 9: Distributions of lead times per task, for a simulated run (FIFO) and for the tasks during the pilot

4. Methods

This chapter discusses the selected task assignment methods that are being tested in the simulation, are compared with each other in the next chapter and can be used in shared service centers to assign tasks to agents. First, the requirements from the environment of the case study are being shared in the first section. Then, the selected methods are introduced, by giving a short description of the applied heuristic and the practical employability within the situation of PBNL.

4.1. Requirements

Considering the design cycle approach of this thesis, it is important to specify artifact requirements in advance, so only task assignment methods are tested and suggested that could be implemented in the real situation. The analysis of requirements is structured in two ways. Firstly, there are two possible scenarios for implementation. The analysis is performed for both scenarios. Secondly, the *Business Strategy, Context and Process (B-SCP) requirements analysis framework* by Bleistein et. al. (2005) is applied to the situation. Before addressing the elements of the framework for each of the two scenario's, the framework and the scenarios will be explained shortly.

In the first scenario, PBNL applies a Workflow Management System (WfMS), delivered by *Workflow Management* (WFM). WFM is a department within the banking group, part of the 2,000 FTE call center. WFM offers the software the call center uses for scheduling, assigning and measuring the work of their call center employees to other business units within the banking group. If PBNL engages with WFM, software will be implemented that organizes the schedules of BS agents, measures their performance and, in a way yet to be determined, assigns work to the BS agents. In the second scenario, PBNL does not make use of the services of WFM. Besides the CRM-system, in which the jobs are placed in a worklist, then no additional software is used to make schedules, measure performance or assign work.

The earlier mentioned B-SCP framework ensures new IT is aligned with the business strategy, context and processes. The framework, shown in Figure 10, consists of six elements: business strategy, business context, business processes, system goals and functions, system context and system processes. When following the structure and elements of the framework when analyzing the requirements of IT, organizations are enabled to cover all relevant and influencing actors and deliver the desired success.

While the scope of this thesis is finding a task assignment method, and not a new IT system, all task assignment methods eventually find their way into the organization by IT, especially in the case of the first scenario, when a WFMS. For this reason, the framework is applied concisely, with a primary purpose to ensure no business factors are overlooked in the requirement analysis.

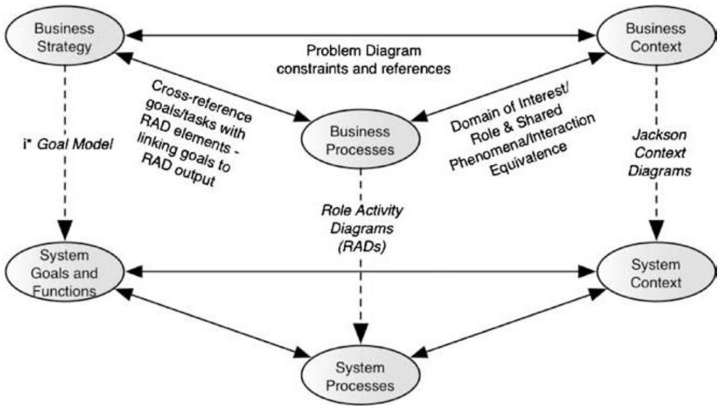

Figure 10: Business Strategy, Context and Processes Framework (Bleistein et. Al., 2005)

Business Strategy
Scenario I + II: As extensively covered in the first chapter, the strategy of the business is to deliver customers a high quality service with minimal use of resources, meaning an optimal balance has to be found between processing times and the amount of agents needed, leading to the first requirement (1) of Table 11.

Business Context
Scenario I + II: the business context is described in the first chapter. From a context point of view, it is important the task assignment method is applicable to a business environment in which manual, administrative, non-geographic, time-consuming transactional tasks are performed, leading to the second (2) requirement in Table 11.

Business Processes
Scenario I + II: on a process level, it is important to state tasks are sent once, and digitally. No interaction or further information from the task submitter is desired when the task is assigned (3). In the case of a Single Step Task (SST), tasks are processed within one action. In the case of a Multi Step Task (MST), tasks are processed in multiple non-consecutive steps (4). To prevent time spent on handovers, tasks are performed by one agent only (5).

System Goals and Functions
Scenario I: If a WfMS is applied, automated and moderately complex algorithms can be deployed, based on task information as start time, deadline, average processing times, and updated agent performance information on processing times per agent (6).
Scenario II: If no WfMS is applied, no automated assignment method can be deployed, meaning the task assignment method must be applied by agents themselves, or by manual work of the manager (8).

System Context
Scenario I: the task assignment method is applied within the WfMS (7)
Scenario II: the task assignment method is applied based on CRM observations (8)

System Context
Scenario I: the task assignment process takes place within the WfMS (7)
Scenario II: the task assignment method takes place manually, meaning the manager or employees must have the skills and knowledge to perform the method (9)

Table 11
Results of the requirement analysis

	General Requirements
R1	the method can be used to find a maximum service level with minimal deployment of agents
R2	the method is applicable with administrative, non-geographic, time-consuming and transactional tasks
R3	the task should be assigned without further interaction with the submitter
R4	the method is applicable for single step tasks and multiple step tasks
R5	tasks are assigned to one agent only
	Additional requirements of the first scenario
R6	the task assignment method can be performed automatically, based on task information as start time, deadline, average processing time and on updated agent information as average processing time
R7	the task assignment method must be compatible with the possibilities and limitations of the WfMS
	Additional requirements of the second scenario
R8	the task assignment method can be performed manually by agents or the manager, based on CRM accessible information as start time and deadline
R9	the manual task assignment method must be executable for employees with no or limited knowledge on mathematical operations management

4.2. Selected Methods

In this section, five task assignment methods are explained. All five methods meet the general requirements as described in last section. For each method, the heuristic will be explained and the practical applicability for PBNL and the two scenarios will be given.

4.2.1. First in, First out

The 'First in, first out' (FIFO) algorithm, in the literature review mentioned by Ha et. al. (2006), Combi & Pozzi (2006) and by Fisherman (2001) described as one of the most basic, yet widely applied task assignments follows the exact logic as it name suggest. Tasks with the oldest queue entrance moment will be assigned first when an agent comes available. The FIFO method is a task-selecting method, following the definition given in the literature review, meaning it only focuses on selecting a task from the queue on the moment an agent is ready to pick up a new task.

A benefit of FIFO is the fact it only requires a single data point, which is the moment the task (re)entered the queue. The FIFO method meets all 9 earlier defined requirements. PBNL's CRM system automatically records the moment a task is created and placed in the worklist, enabling both agents and manager to visually decide which task should be performed next, making a costly WfMS redundant, from an assignment point of view.

4.2.2. Earliest Deadline First

The 'Earliest Deadline First' (EDF) algorithm, in the literature review described by Turner (2018) and further explained in Fishmans (2001) research, is another basic task assignment method. Tasks with the earlier due date, or deadline, are assigned first when an agent comes available. Just like FIFO, the EDF method is task-selecting, meaning it only selects a task from the queue on the moment an agent is ready to pick up a new task.

As EDF also only requires a single data point, which is the due date, it also meets all requirements. When creating a new task for Business Support, the submitting bankers, investment managers and assistants fill in multiple fields in the form within the CRM-systems. One of those fields is the due date. Based on the Service Level Agreement (SLA), which includes the number of days before BS should have completed the task, the employee can fill in the due date. At this moment, the implementation plans include a SLA of 4 days to finish the task after sending it to BS.

EDF is compatible with a WfMS but is also uncomplex for agents and the BS manager to apply manually, without needing a WfMS. However, it should be noted from business observations and interviews, it is found to be not uncommon for bankers and investment managers to seek and break boundaries when it comes to verbal agreements. If the deadline is not set automatically to the number of days given by the Service Level Agreement, the concern exists employees fill in shorter deadlines in order to get their own tasks performed by BS sooner.

4.2.3. Workload Balancing

The task assignment method of 'Workload Balancing' (WldB) is proposed by Ha et. al. (2004) and Ha et. al. (2006). As this methods assigns tasks in a more extensive way compared to the first two, the applied heuristic will be explained. The goal of WldB is to optimize the balance between workloads by varying task assignment probabilities in a linear program. The method uses task arrival rate λ, T as set of tasks, A as set of agents, $R \subseteq T \times A$ as set of responsibilities on an agent for a task, $\mu_{t,a}$ as the average processing rate of agent a on task t, where (t, a) $\in R$

For each agent, the workload is computed with:

$$ld_a = \sum_{t \in T_a} \frac{\lambda_t \cdot p_{t,a}}{\mu_{t,a}}, \qquad \forall a \in A$$

where $p_{t,a}$ is the probability task t gets assigned to agent a. These probabilities, which during the course of the task assignment period determine which agents get which task, are calculated via the following Linear Program:

$$\begin{aligned} & \min \max\{ld_a\}, && \forall a \in A \\ \text{s.t.} \quad & \sum_{a \in A_t} p_{t,a} = 1 \ , && \forall\, t \in T \\ & 0 \le p_{t,a} \le 1 \ , && \forall\, (t,a) \in R \end{aligned}$$

With other word, this method tries to balance the percentage of time all agents are working on tasks, by analyzing the arrival and processing behavior and assigning tasks based on assignment probabilities, which can be updated constantly or periodically.

WldB is an agent-selecting method, as it selects an agent for a to be assigned task. Where previous two assignment methods did not include agent queues, as all tasks were selected from the assignment queue, WldB comes with agent queues, as the assignment method includes assignment probabilities but no agent availability.

The method of Workload Balancing meets the five general requirements and the two requirements of the WfMS scenario but does not meet the requirements of the second scenario. Agents of BS cannot apply the heuristic manually and while the manager of BS could assign tasks supported by an assignment probability tool, it would still need to adjust the calculations when personnel changes occur, needing information on arrival and processing behavior that with just having the CRM-system, is not available directly.

The for the case study calculated task assignment probabilities are given in Appendix H.

4.2.4. Worklist Balancing

The assignment method of 'Worklist Balancing' (WlsB) is not derived from articles of the literature review but is proposed by this thesis as an adaption to Workload Balancing. Just as with WldB, the WlsB method placed tasks in queues of agents the moment they enter the assignment queue. This makes WlsB an agent-selecting algorithm. The decision on which agent is selected for the task is not based on assignment probabilities, but on the length of agent queues. When a task enters the queue and the assignment method starts, the lengths of all agents' queues are measured in number tasks. The agent with the least number of tasks in its queue gets the new task. When multiple agents have the same number of tasks in its queue, the task gets assigned randomly amongst those agents.

In contrast to WldB, where the required mathematical calculations prevent the agents and manager to perform the method themselves, the WlsB meets all 9 requirements and is thus suitable for implementation without WfMS. In the CRM-system, tasks sent to BS can get a so-called owner. By manually assigning new tasks to agents with the least assigned tasks, the manager is able to apply this method.

4.2.5. Shortest Processing Time First

The assignment method of "Shortest Processing Time First" (SPT) is mentioned in the literature review by Ha et. al. (2006) and explored in depth by Rose (2001). The SPT method, as a task-selecting algorithm, selects the task with the shortest processing time when an agent comes available and is ready to pick up a new task.

The SPT method can use averaged task information, meaning processing times are not specified at an agent level, but can also consider individual processing times for each task, for each agent. In case of the former, the method meets all 9 requirements, as it is possible for all agents and managers to have access to average processing times and based on this, pick the task from the worklist which has the lowest average processing times. In the case of the second, when individual processing times are used, and especially when they are updated as agents work and new performance is measured, it does not longer meet requirements 8 and 9. This means the method can only be executed by the WfMS.

5. Evaluation

This chapter discusses the results of the simulations and thus the impact of different task assignment methods on the performance of PBNL's Business Support. In the first section, the task assignment methods are compared in the simulation that with all information described in earlier chapters tries to approach the situation of PBNL. In the second section, the influence of an increased workload on the outcome of the assignment methods is explored. In the third section, the impact of influencing factors, such as high priority tasks and scheduled unavailability is explored. This chapter is concluded by a check to verify whether BS could work with less agents than planned. Appendix I shows underlying data for all tables and graphs shown in this chapter.

5.1. Methods compared within the PBNL situation

This section discusses the results of the simulation based on the situation at PBNL. As shown in Table 12, the simulation applies the arrival and processing rates as discussed in the chapter on data analysis.

Table 12
Test setup for Simulation 1 'PBNL'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

Before the simulation results are analyzed, first it checked whether the simulation successfully processes all tasks and if there is enough production capacity. This is done by plotting the length of the queue(s) of a random run of the simulation against time. As is visible in Figure 11, the total queue length fluctuates strongly for the WlsB and WldB method but shows no continuing increase, meaning all tasks are eventually processed.

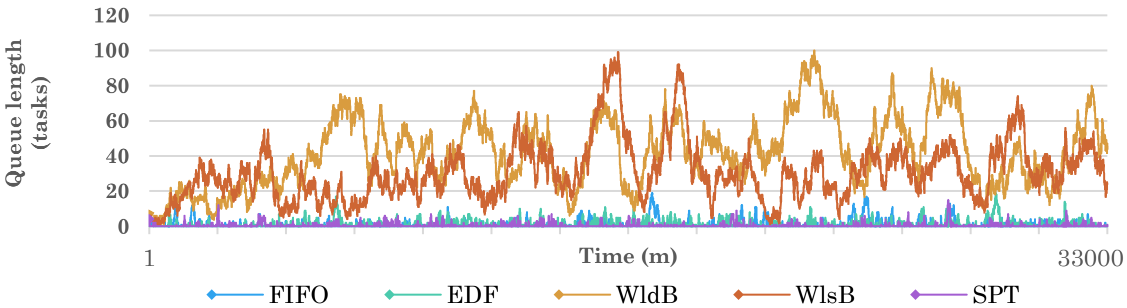

Figure 11: Length of assignment queue(s), expressed in number of tasks, for Simulation 1 'PBNL'

The difference in queue length between WlsB and WldB on the one hand, and FIFO, EDF and SPT on the other is caused by the fact that the first group is agent-selecting and the second group is task-selecting. In agent-selecting tasks, tasks are placed in queues of agents, who are unavailable for one day per week, leading to increased queues. In task-selecting methods, tasks are only assigned to available agents.

Figure 12 shows the results of the first simulation. The service level shows the ratio of tasks completed before the deadline of 4 days. The utilization rate shows average ratio of time spent working on tasks, compared with to the total time spent working, which excludes the planned day off. The waiting time represents the average time a task spends in a queue, the processing time equals the average time a task is being processed by an agent and the processing & waiting time represents these two times combined. The lead time shows the difference in time between entering the system for the first time and leaving it for the last time. This value is higher than the combined processing and waiting time due to the rearrival time of multi-step tasks. All graphs show the average value.

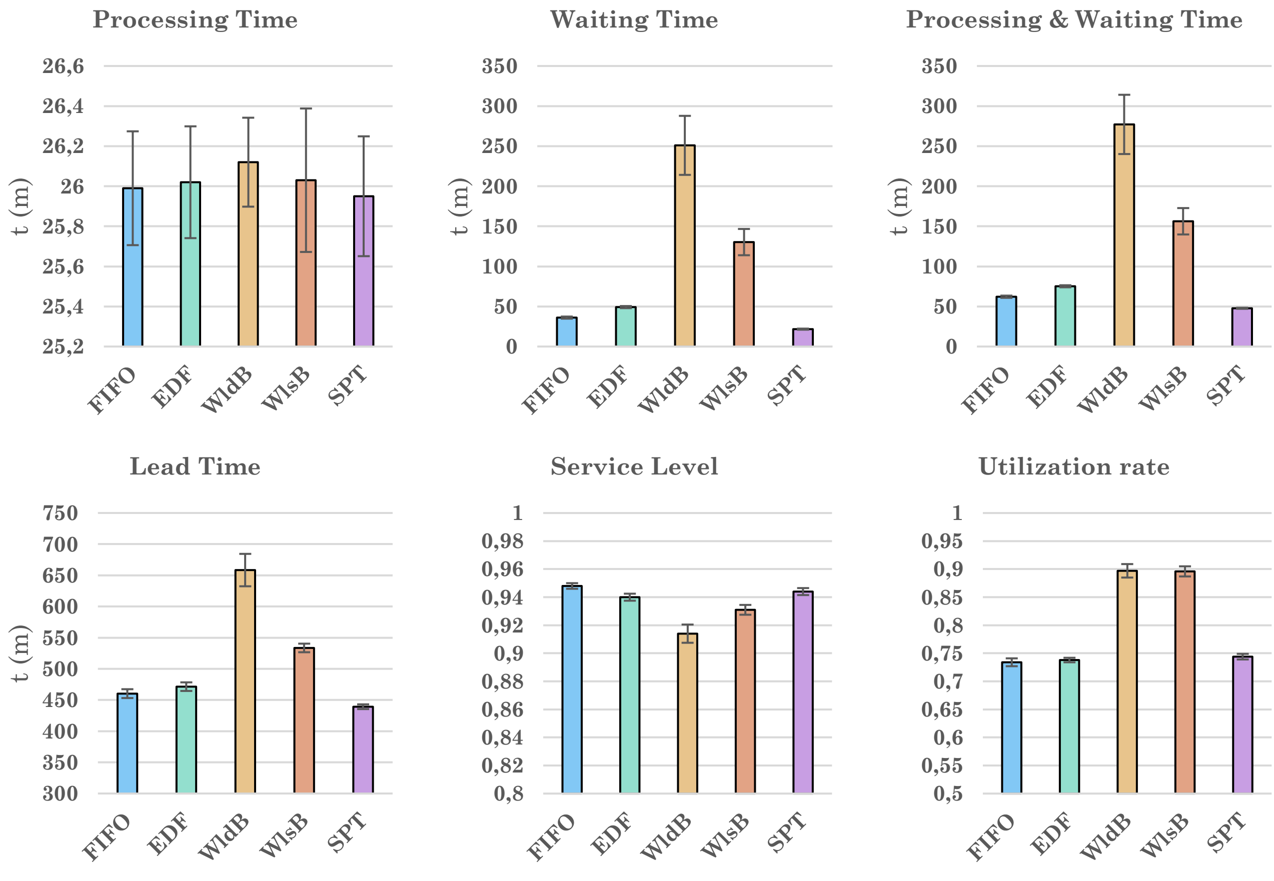

Figure 12: Results of Simulation1 'PBNL', showing average processing, waiting and lead times, together with service level and utilization rate. The I-shaped figures on each bar represent the standard deviation.

As shown in Figure 12, there is again a significant difference between the task and agent-selecting methods. Methods that assign tasks to agents' queues (WldB, WlsB) wait longer in agent queues, since the agent they are assigned to is unavailable to pick up tasks for one day per week. This longer average waiting times lead to increased lead times and eventually to the lowest service levels amongst all methods, meaning they more often exceed the deadline given by the SLA. Balanced worklists or workloads do not result in less utilized agents on average, as agent queues build up during their unavailability.

Within the group of task-selecting agents, FIFO and SPT result in lower processing times, waiting times and thus lead times, compared to EDF. While EDF focuses on finishing tasks with the earliest deadline, FIFO and SPT both result in a higher service level. This apparent contradiction is further investigated in the third section, where the influence of MST's on the ability of EDF to score highest on service level is tested.

Concluding the simulated implementation of BS at PBNL, following the data retrieved parameters, task-selecting methods result in a higher service level, less utilized agents and lower processing and waiting times than agent-selecting ones. Within the task-selecting methods, FIFO and SPT outperform EDF, both resulting in an utilization rate of 0.74, a service level of 0.94 and average lead times of 413 and 419 minutes, respectively.

5.2. Methods compared with an increased workload

In this section, the impact of an increased workload is further analyzed. An increased workload is of interest as, from a business perspective, it could be beneficial to have agents working on tasks for a larger share of their availability. This section explores the impact of an increased workload on the performance and differences of assignment methods. This is done in two steps.

In the first step, the workload is increased with 20%, to verify what happens with the utilization of agents in the agent-selecting methods, as their utilization rate already amounts 0,89 in the non-increased situation.

The workload is increased with 20% by increasing each arrival rate with 20%, resulting in 20% more tasks within the same simulated amount of time. The applied test setup is shown in Table 13.

Table 13
Test setup for Simulation 2 'PBNL 20% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.2
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

As expected, the 20% increase in workload causes the simulation to become unstable for the agent-selecting methods, as can be derived from the check on number of tasks in queues and the utilization rates, shown in Figure 13. When tasks are assigned via an agent-selecting method, they are placed in individual agents' worklists. For two days a week, one agent is unavailable and for three days a week, two agents are unavailable.

This procedure, in which 22.5% of the arrived tasks are placed in queues of agents that are unavailable to pick them up on the same day, leads to a situation in which queues increase more rapid than agents can handle. With an increased workload of 20%, agents over-receive an average of 8.16 (WlsB) and 7.38 (WldB) tasks per week.

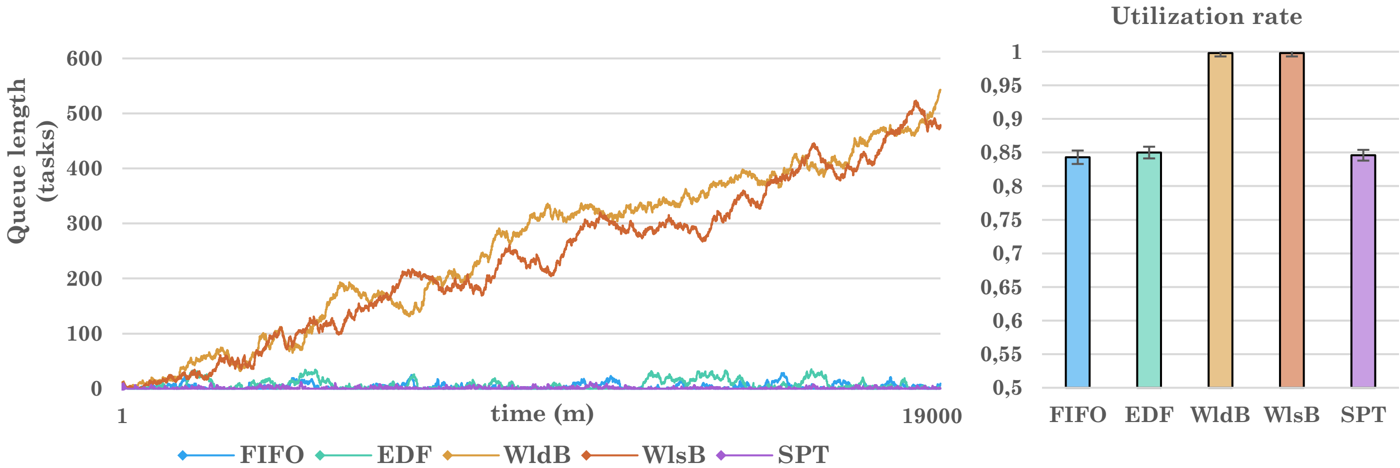

Figure 13: Selection of results of Simulation 2 'PBNL 20% Increased', showing an unstable increase in number of queued tasks on the left and fully utilized agents on the right.

As Figure 13 also shows the remaining, task-selecting methods, to reach a utilization rate not higher than 0.85, the experiment to verify assignment behavior with an increased workload continuous with the three remaining methods, but with a further increase in workload. As can been seen in Table 14, the second version of the increased workload simulation is performed with an increase of 40%, compared to the situation retrieved from the data study. The check on stability is given in Figure 14.

Table 14
Test setup for Simulation 3 'PBNL 40% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.4
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

As displayed in Figure 14, the queue length shows a significant deviation compared to the queue lengths for this selection of methods during the first two simulations. For all three methods, however, there is no continuous increase in queue length visible, meaning the stimulated workload does not exceeds the overall available production capacity.

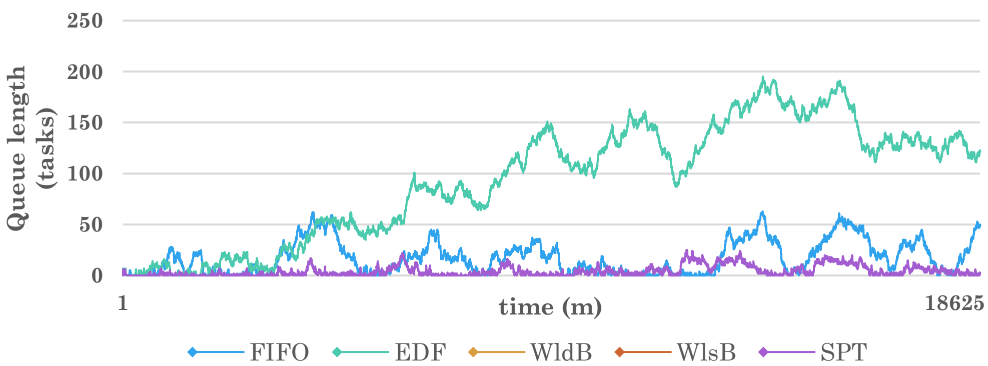

Figure 14: Length of assignment queue, expressed in number of tasks, for Simulation 3 'PBNL 40% Increased'

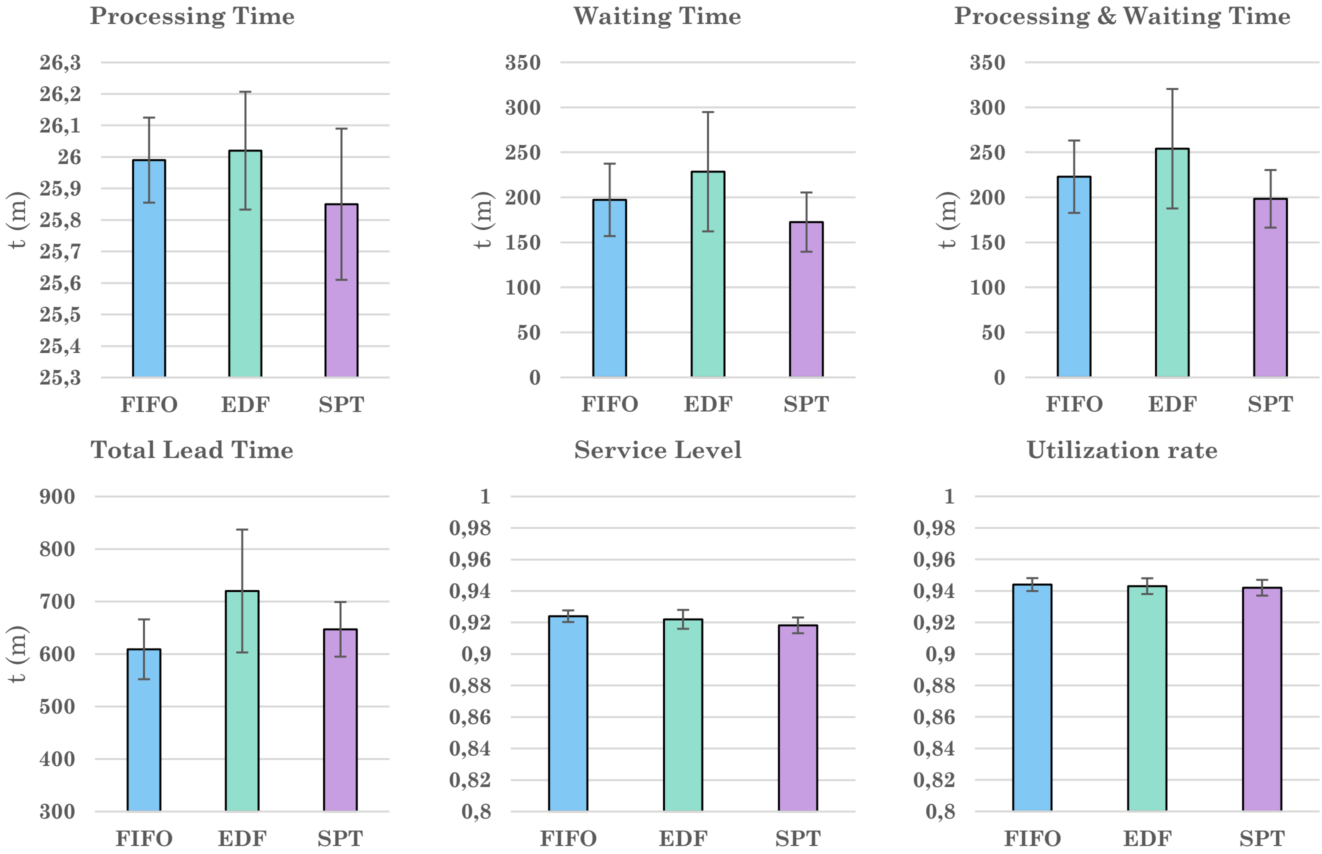

Figure 15: Results of Simulation 3 'PBNL 40% Increased'

When looking at the results of the simulation with the original arrival rates being increased with 40%, shown in Figure 15, it becomes clear the difference in service level between the three methods is negligible, and in average, has decreased from 0.94 in the first simulation to 0.92 in the 40% increased simulation. The observation that an increase of 40% in workload with an SLA of four days only results in the limited decrease of 0.02 in service level, leads to a further exploration of the impact of the number of days in the SLA on the achieved service levels, also plotted against the number of available agents, in the fourth paragraph of this chapter.

Looking further at the results, it shows the focus of EDF on picking tasks with the soonest deadline, results in higher waiting times and thus higher total lead times. This is caused by the existence of multi-step tasks, which rearrive after a given period of time and have an earlier deadline than tasks that arrived during this rearrival period. When these MST tasks are finished earlier than new tasks, the total average waiting time increases.

Concluding the simulation of an increased workload, it is stated that agent-selecting methods cannot cope with an increase of 20%. Task selecting methods can handle an increase of 40% at the price of increased agent utilization rates, of 0.944 (FIFO), 0.943 (EDF) and 0.942 (SPT). With an agreed deadline of 4 days, the service level shows a relatively small decrease of 0.02 when workload is increased with 40%. Average lead times increase due to longer waiting times. The average lead time for FIFO increases to 609, which closely approaches the value of 610 that was retrieved from the real-world validation in previous chapter.

5.3. Impact of influencing factors

The results of the compared and tested task assignment methods in previous two sections are subject to influencing behavior of specific factors of the situation at PBNL. In this section, adaptations to the test setup are made to verify whether the earlier drawn conclusions stand ground or lead to new insights. First, the impact of the scheduled unavailability of agents on the difference between task and agent selecting methods is analyzed. Then, the impact of the rearrival time on the achieved service level is explained. Lastly, the impact of the existence of tasks with shorter deadlines is explored

5.3.1. Continuous availability

To verify whether the relatively big difference between the task selecting and agent selecting methods, as demonstrated in the first simulation, is caused by the 20% scheduled agent unavailability, the methods are tested in a setup with 100% availability. To compensate the reduced workload by the increase of available agents, the workload is increased workload with 20%, following the test setup as shown in Table 15. With an overall workload similar to the first simulation, the system is found to be stable.

Table 15
Test setup for Simulation 4 'PBNL Continuous available, 20% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.2
Processing rates	Based on data analysis PBNL
Agent behaviour	100% availability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

As shown in Figure 16, removing the scheduled unavailability leads to a significant drop in agent utilization for the agent selecting assignment methods. The waiting times stay higher, due to the existence of agent queues in which tasks spend a longer time waiting than in the single assignment queue that directly leads to an available agent, for the task selecting methods. With a SLA of 4 days, the service level for agent selecting tasks reaches the same level (0.94) as for the task selecting ones.

Figure 16: Results of Simulation 4 'PBNL Continuous available, 20% increased'

5.3.2. Delayed rearrivals

Multi step tasks leave the system after the first half of work is performed and reenter the system after a client or colleague has finished their share of the required work. Task 6, preparation of client meetings, and task 7, preparation of the compliance revision are both MST's for which the rearrival time has been estimated in interviews at 2 days on average, and which has been distributed exponentially. This section explores what happens with the service level of Business Support when this rearrival time deviates. Table 16 shows the test setup. For six different average rearrival times, the service level is determined for each assignment method. Tasks arrive according to the actual situation from PBNL, as tested in the first Simulation, and the same follows for the processing rates and the 20% scheduled unavailability.

Table 16
Test setup for Simulation 5 'PBNL with Delayed Rearrivals'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability
SLA	4 days for all tasks
Rearrival time	Varying between 1 and 3.5
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

Figure 17 shows the results of the test. The percentage of tasks completed before the deadline of four days is plotted against the simulated average rearrival time. All methods show a decrease in service level when tasks reenter the system later. This is caused by the simple fact less time is left to complete the task before the deadline is over. The decrease per method varies in steepness. WldB and WlsB, both coming from a lower starting service level due to the reasons shared in the first section, show the steepest decline and drop a 12.7% (WldB) and 12.2% (WlsB), due to their inability to assign urgent tasks directly to idle agents. For EDF and SPT, the service levels decrease with 11.0% and 9.62% respectively and with FIFO, the service level decreases with 5.6%, still delivering 94% of tasks on time. Where it would seem logical EDF would outperform SPT, as it aims at picking up tasks with the earliest deadline, the method makes adaptions in the task order with tasks that already have missed their deadline, creating a possible domino effect. Not making any adaptions at all and finishing based on time spent in queue remains to be the most effective.

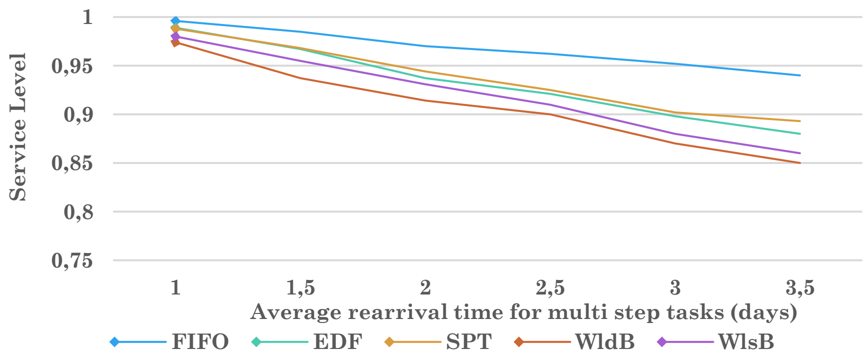

Figure 17: Compared Service Levels for each task assignment method with varying rearrival times for multi-step tasks. Results of Simulation 5 'PBNL with Delayed Rearrivals'

5.3.3. Urgent tasks

Bankers and investment managers at PBNL are found to be highly autonomous and creative in finding ways to get support by the parts of the organization they interact with. In concrete terms, this means senior management has expressed the expectation bankers and investment managers will try to get tasks done by BS with priority, with a quicker deadline than given in the SLA. If inside the CRM-system, PBNL does not automatically fill in and lock the deadline field, the requesting colleagues will fill in their own deadlines. Even if the deadline is set automatically, the possibility still exists colleagues from the local offices will have contact with BS agents via other canals to express their priority.

For this reason, the possibility to send urgent tasks, in a regulated manner, is tested in the simulation. The two tasks with the highest possibility of being sent as an urgent task, being task 6: preparation of client meeting and task 9: processing client request, are given a SLA of 1 day. The influence this has on the overall service level is tested with the sixth simulation, of which the set up is shown in Table 17.

Table 17
Test setup for Simulation 6 'PBNL with Urgent Tasks'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability
SLA	1 day for tasks 6 & 9, 4 days for other tasks
Rearrival time	2
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

As visible in Figure 18, each task assignment method drops in service level. Again, FIFO remains to be performing the best. EDF does not achieve a higher SLA, also because of the domino effect of sequenced overdue tasks. Task 6 is dominating the number of overdue tasks, caused by its nature as MST. Task 7, a MST with its original deadline of 4 days , is impacted heavier by the reduced SLA of the two tasks than task 9, while task 9 has its deadline decreased to 1 day. This again underlines the conclusions that the rearrival time has a significant influence on the service levels.

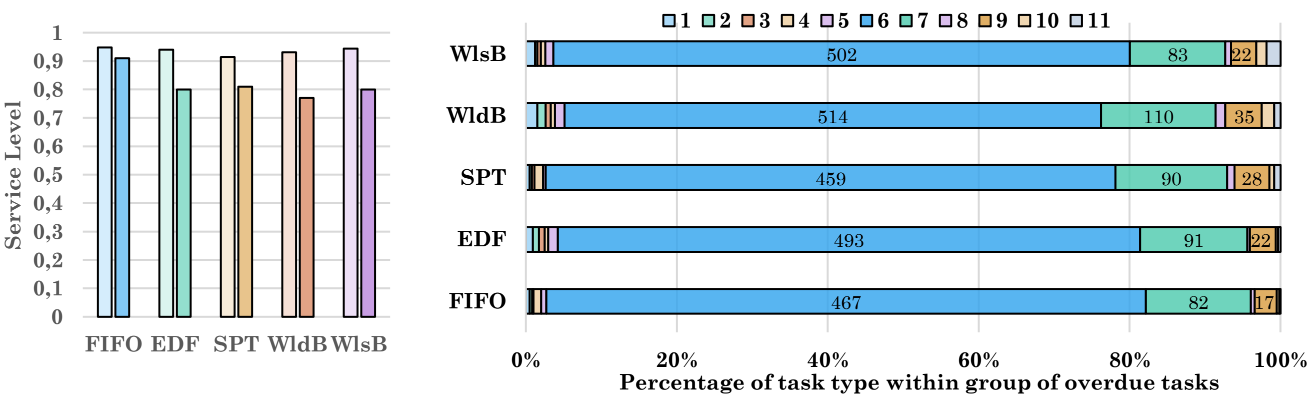

Figure 18: Left: service level in original situation (light) compared to service level when task 6 and 9 have a SLA of 1 days. Right: composition of overdue tasks, with percentage of overdue task type within entire group.

5.4. Random Unavailability & Reduced Number of Agents

The conclusions of the first two sections of this chapter show that when the task-selecting methods are applied, no maximum agent utilizations rates are found. Since the goal of PBNL is to deploy as few resources as needed to achieve the service level, this section explores whether it would be possible for PBNL to operate BS with less agents.

In order to make accurate recommendations on real world agents, the deviation between the real-world average lead times and the lead times found with the earlier found simulations is addressed first. Two modifications are made to the test setup. First, the workload is increased with 20%, the same way as explained in the second section of this chapter. The deviation between the two lead times, however, can be caused by other factors as well. As the possibility exists this deviation is caused by agents being occupied with non-work-related matters, as chatting, getting coffee and going to the toilet, a random unscheduled unavailability is added to the simulation. With an average of four daily breaks of 10-15 minutes, agents become randomly unavailable in the simulation for a total of 8% of their time, additional to the scheduled 20% of unavailability in the form of a weekly day off. When combined, this leads to a new test setup, shown in Table 18.

Table 18
Test setup for Simulation 7 'PBNL 20% Increased, 8% Random Unavailability – 8 agents'

Arrival rates	Rates from PBNL data analysis, increased with 20%
Processing rates	Based on data analysis PBNL
Agent behaviour	20% scheduled and 8% random unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8 agents
Included methods	FIFO, EDF, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

The results of this new approximation of the PBNL situation are shared in Figure 19. As shown, the lead times for all three methods approach the value of 610 minutes found in the validation better than previous simulations. With arrival rates being increased with 20% and with random unavailability of 8%, an average service level of 0.92 is achieved for FIFO and EDF, and 0.91 for SPT, with an utilization rate of 0.95.

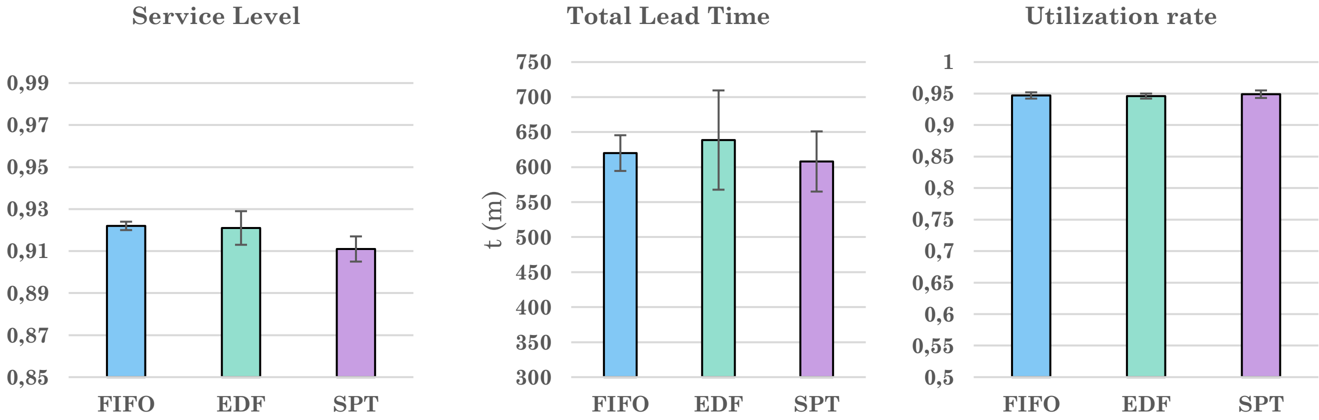

Figure 19: Results for Simulation 7 'PBNL 20% Increased, 8% Random Unavailability – 8 agents'

By adding the random unavailability to approach the validated lead time as much as possible, not only the validated time is better simulated, the agent utilization also approaches 1, indicating PBNL has determined the right number of agents for BS. To test what happens when the number of agents is decreased, the number of agents is changed to 7 and the simulation is run again, with parameters as shown in Table 19.

Table 19
Test setup for Simulation 8 'PBNL 20% Increased, 8% Random Unavailability – 7 agents'

Arrival rates	Rates from PBNL data analysis, increased with 20%
Processing rates	Based on data analysis PBNL
Agent behaviour	20% scheduled and 8% random unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	7 agents
Included methods	FIFO, EDF, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

As shown in Figure 20, with 7 agents, the simulation becomes unstable and generates more tasks than agents can process. A difference is notable between FIFO and EDF on one hand, with a continuous increase, and SPT, with a less rapid and more periodically occurring increase. With randomly unavailable agents and an increased workload, the SPT method is able to process more tasks than FIFO and EDF, due to its ability to have agents working on the jobs they can do fastest, and thus having them performing more tasks in the same period of time. For all three methods, the service level keeps decreasing over time.

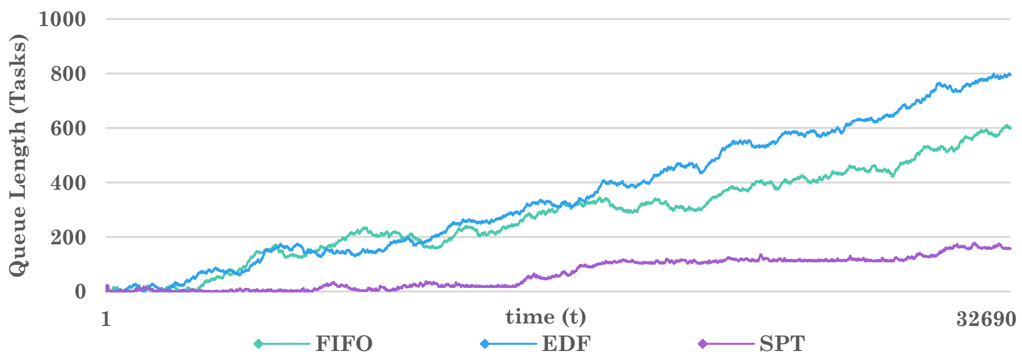

Figure 20: Results for Simulation 8 'PBNL 20% Increased, 8% Random Unavailability – 7 agents'

As a conclusion to the test whether PBNL could deploy less agents and keep a high service level, it is found that when the simulation best matches the actual situation, by increasing the workload and adding random unavailability, eight agents can maintain a high service level without significant differences between the methods. The decrease of a single agent, however, makes the system unstable and results, for all three task selecting methods, in an inflow of new tasks that is too high for the remaining seven agents to process. Figure 20 shows that when the backlog becomes increasingly longer, SPT is the most effective method in resisting the increase. Service levels are no longer being met, but with SPT tasks are finished by the agents who can do them the quickest, making it an effective method for crisis management.

6. Implementation

With the different task assignment methods being evaluated in the previous chapter, this chapter focuses on the practical recommendations PBNL can use for the implementation of Business Support. First, an overview of the five task assignment methods with their evaluation outcome and level of satisfaction to earlier mentioned requirements is given in Table 20. Then, based on findings from the evaluation, practical recommendations are shared.

Table 20
Overview of Task Assignment Methods score on requirements and simulation evaluation, where S represents Service level and U represents Utilization Rate

	FIFO		EDF		WLDB		WLSB		SPT	
General Requirements										
1. Can be used to find minimal deployment of agents	√		√		√		√		√	
2, Applicable with administrative, non-geographic, time-consuming and transactional tasks	√		√		√		√		√	
3. Can assign without interaction with submitter	√		√		√		√		√	
4. Applicable for SST and MST	√		√		√		√		√	
5. Assigns to one agent only	√		√		√		√		√	
Scenario 1 'WfMS'										
6. Assignment can be performed automatically, based on task information as start time, deadline and average processing time	√		√		√		√		√	
7. Compatible with the possibilities and limitations of the WfMS	√		√		√		√		√	
Scenario 2 "No WfMS"										
8. Can be performed manually by agents or the manager, based on CRM accessible information	√		√				√			
9. Executable for employees with no or limited knowledge on mathematical operations management	√		√				√		√	
Evaluation results	S	U	S	U	S	U	S	U	S	U
1. PBNL	0,95	0,73	0,94	0,74	0,91	0,90	0,93	0,90	0,94	0,74
2. PBNL 20% Increased	0,94	0,84	0,93	0,85	-	-	-	-	0,94	0,85
3. PBNL 40% Increased	0,92	0,94	0,92	0,94	-	-	-	-	0,92	0,94
4. PBNL Continuous Available, 20% Increased	0,94	0,67	0,94	0,68	0,94	0,68	0,94	0,68	0,94	0,68
6. PBNL with Urgent Tasks	0,91	0,73	0,8	0,74	0,81	0,90	0,77	0,90	0,80	0,74
7. PBNL 20% Increased, Random Unavailable - 8 Agents	0,92	0,95	0,92	0,95	-	-	-	-	0,91	0,95

Task Selecting versus Agent Selecting Method
Whether PBNL chooses to go for the first scenario with a Workflow Management System, or for the second scenario, with just a manager and agents themselves available to assign tasks, agents at BS will not work continuously. As shown in the simulation results, the agent-selecting methods are far from optimal when agents have scheduled unavailability. For this reason, the first recommendation for PBNL is to apply a task-selecting method.

FIFO, EDF or SPT
Being compared with each other under varying circumstances, FIFO, EDF and SPT show little differences on the defined performance indicators service level and utilization rate. The relatively spacious deadline of 4 days in the SLA covers the differences between the task assignment methods with respect to service level. Looking at time instead of service level, with FIFO, in nearly all test setups, combined processing and waiting times and total lead times are the lowest. Whether PBNL decides to go for the first or second scenario, the second recommendation is to apply FIFO as task assignment method.

Assigning FIFO with a WfMS
The implementation of FIFO in PBNL's Workflow Management System is not found to be complex from a practical point of view. When PBNL decides to hire the services of the Workflow Management team, it automatically gets task trafficking software. The assignment policy of this software can be set to FIFO. Agents report their status to this software and when they have finished a task, they get a new task, based on the policy.

Assigning FIFO without a WfMS
The implementation of FIFO without a WfMS is possible and relies on a correct use of the CRM-system. When agents have finished a task, they open the general worklist of BS in the CRM-system and select the task with the oldest date of entering the queue, and put the task on their own name. In this scenario, it is recommended the CRM-fields containing the temporal data are locked and generated automatically when a task is opened, to prevent the task sending colleagues from creating false urgent requests.

Minimize the Rearrival Time
From the evaluation, it is found the rearrival time for MST's has a bigger impact on the service level than a workload increase of 40%. This is partly caused by the fact that the workload from the data analysis is simply not demanding the full capacity of the available agents. However, with a SLA of 4 days and a rearrival time of 2 days, the performance of BS can be increased when PBNL takes effort to minimize the amount of tasks that cannot be finished in a single action, and thus appear as a MST. It is recommended during and after the implementation, Business Supports performance is further optimized by analyzing the tasks that cannot be finished in a single action and make adaptions to their corresponding business processes.

Number of Agents Deployed
The last simulation, which approaches the real-world situation the best of all simulations, shows the implementation plans of BS contain the right number of agents. It is not recommended to start the implementation with less agents. However, since agents are going to work on a smaller set of tasks with a higher frequency than in the current situation, it is possible this increase in experience will lead to shorter processing times. For this reason, it is recommended to repeat this study after 6 months, with a new data analysis, to explore the possibility of scaling down the agent deployment.

7. Conclusion

This chapter concludes this thesis. First, a concluding summary is given. Then, the limitations are shared. Thirdly, the scientific contribution is given. The recommendations for the host of the study are given lastly.

7.1. Concluding Summary

The research goal of this thesis is to deliver recommendations on task distribution, suitable for implementation within a shared service center, by applying currently existing distribution methods in a data-based simulation and assessing their impact on service level and agent utilization. Service level is defined as the percentage of tasks finished before the deadline of 4 days and agent utilization is expressed in percentage of time spent working on tasks, by the available agents.

This goal is achieved by firstly answering the question which task distribution methods are given in recent, relevant literature and are applicable on transactional shared service centers with a focus on administrative processes. Then, properties of processes and employees necessary to perform an accurate simulation are defined and retrieved from a data study at research host PBNL. Lastly, the impact of task distribution methods on the service level and agent utilization of a shared service center is being tested with a simulation.

From literature, 21 task assignment methods are found and analyzed. Task assignment methods can be placed in three categories. First, there are task-selecting methods, which find a suitable task when an agent becomes idle. Then, there agent-selecting methods, which find a suitable agent for a task that enter the system. Lastly, there are set-matching methods, which, with a set of (future) tasks and available agents, create a distribution.

Within the context of a shared service center, with Business Support at PBNL as an example, tasks arise random and unplanned. A selection of five task assignment methods is made. The three task-selecting methods are FIFO, which selects the task that entered the queue first, EDF, which selects the task with the soonest deadline and SPT, which selects the task the to-be-assigned agent can perform the quickest. The two agent-selecting methods are Worklist Balancing, which assigns tasks to the agent with the least amount of tasks in its queue and Workload Balancing, which assigns tasks according to predetermined assignment probabilities, that are calculated in a linear program and aim to balance the overall workload of agents.

By collecting data on the arrival behavior of tasks and the processing behavior of agents, arrival rates and processing rates are modelled in a queueing system. When transformed into code, this model allows for the comparison of assignment methods.

The simulation is verified and validated. The initial validation shows the parameters retrieved by the data analysis lead to an average lead time 33% smaller than observed in the real world. For this reason, multiple simulations are run, so the assignment methods are compared in varying circumstances.

Agent-selecting methods are not suitable for shared service centers in which agents are not available for 100% of the working hours. The existence of individual agents' queues leads to a significant increase in waiting times, which lead to a lower service level. The fact that agents start working with a filled queue, after a day off, also causes a higher overall utilization rate. As shown in Figure 21, with FIFO, the highest service level is found, also having lowest utilization rate.

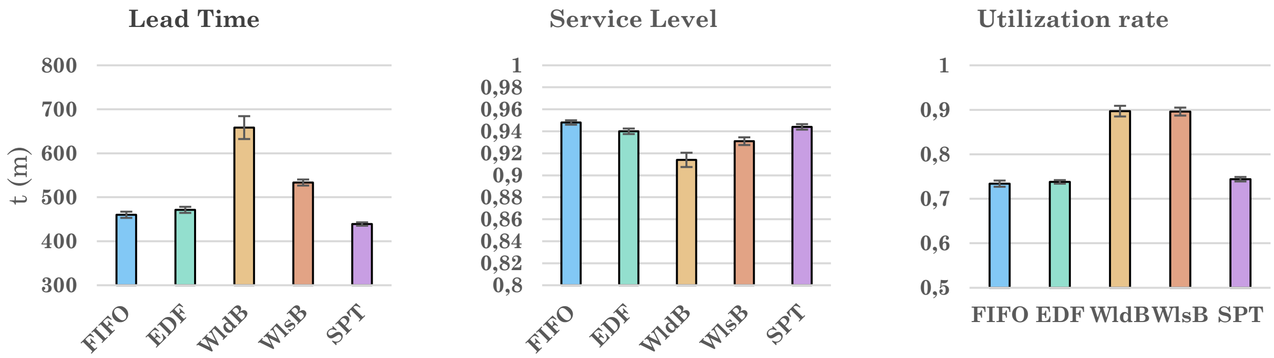

Figure 21: Selection of Results for the Simulation with Data Retrieved parameters (1)

The real-world situation is approached by altering the simulation parameters in two ways. First, the workload is increased with 20%. Then, random unavailability is added by making agents spend 8% of their time on other things than tasks. As shown in Figure 22, agents then are utilized up to 95% of their time, achieving a service level of 92% with FIFO scoring best again.

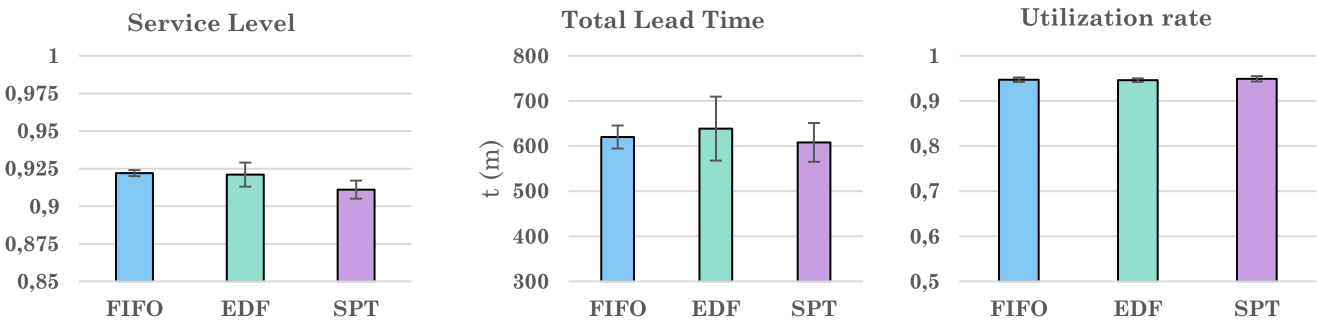

Figure 22: Selection of Results for Simulation most approaching the validation

As is visible in Figure 21 and 22, the difference between the task-selecting methods are relatively small and their significance appears to be questionable. However, in the simulations where urgent tasks are introduced or multi-step tasks have a higher rearrival time, the lead by FIFO increases. EDF, sequencing on deadline, is unable to deal with tasks that are already overdue and therefore periodically creates a domino effect of overdue tasks. SPT is unable to detect the tasks that are waiting the longest. When the total workload overburdens the processing capacity however, SPT is best in minimizing the continuous growth of the queued tasks. Furthermore, it is not found to be possible for PBNL to deploy less agents without losing the ability to keep up with the number of tasks arriving, for now.

7.2. Limitations

PBNL's Business Support, as the shared service center subject to the research, is not implemented yet. The simulation environment is created with arrival and processing rates derived from the pre-implementation situation. The actual behavior after implementation can deviate, due to changes between the pre and post situation. Where increased and decreased workloads showed no significant changes in the order of ranking of task assignment methods, this possible deviation mostly influences the conclusion whether Business Support can work with less agents or not.

Five task assignment methods are tested in this simulation. The selection is based on the requirements and on practical applicability within the context and time available for a master thesis. This thesis cannot rule out better performing task assignment methods exist.

7.3. Contributions

This thesis contributes in several ways. First, by providing with a novel classification for task assignment methods is proposed, with task-selecting, agent-selecting and set matching methods. This classification guides the selection of a task assignment method, based on the business context. Secondly, by giving a novel framework for the design of task assignment business process is proposed. This framework also guides the selection of a task assignment method, as it describes different design variables and scopes the possible methods. Lastly, this thesis gives a not in literature found comparison of task assignment methods FIFO, EDF, SPT and Workload Balancing, yielding the finding that within several test setups, FIFO outperforms the other three. This last contribution is applicable in a wider range and can be used by other organizations that want to implement a task assignment method with a similar context or characteristics.

7.4. Recommendations

From the business point of view of PBNL, several recommendations are given. First, it is recommended to not implement an agent-selecting method, as agents will be scheduled unavailable. Secondly, whether a WfMS is implemented or task assignment is performed by the team manager and the agents themselves, it is recommended choose for the FIFO method on moments the workload is under control. On moments in which task supply transcends the processing capacity and the goal is to eliminate the assignment queue as fast as possible, the third recommendation is to switch to SPT.

Fourthly, it is recommended to focus on preventing MST's as much as possible, as the length of the rearrival time has a strong influence on the overall capacity. BS should focus on getting tasks in such a way, they can be completed in a single action. PBNL management and performance consultants should actively monitor this and adjust and improve the processes leading to incoming MST's.

Lastly, it is recommended to start working with current planned number of agents and monitor the arrival and processing rates periodically, as the possibility exists agents become quicker on tasks and the possibility to deploy less agents still arises, months after implementation.

References

ACM Digital Library. (2020). *About the ACM Digital Library*. Retrieved September 23, 2020, from https://dl.acm.org/about

Basu Roy, S., Lykourentzou, I., Thirumuruganathan, S., Amer, S. & Das, G. (2015). Assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal, 24, 467–491*

Bleistein, S., Cox, K., Verner, J. & Phalp, K. (2005). B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. *Information and Software Technology, 48, 846-868*

Combi, C. & Pozzi, G. (2006). Task Scheduling for a Temporal Workflow Management System. *Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06), Budapest*, *61-68*

Dickerson, J., Sankararaman, K., Srinivasan, A. & Xu, P. (2018). Assigning Tasks to Workers based on Historical Data: Online Task Assignment with Two-sided Arrivals. *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18)*, *Richland*, *318–326*

Elsevier (2020. *ScienceDirect*. Retrieved September, 23, 2020, from https://www.elsevier.com/solutions/sciencedirect

Ernst, A., Jiang, H., Krishnamoorthy, M. & Sier, D. (2004). An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research, 127, 21-144*

Fu, B. & Tari, Z. (2003). A Dynamic Load Distribution Strategy for Systems Under High Task Variation and Heavy Traffic. *ACM Symposium on Applied Computing, 1031-1037*

Gerkey B. & Matarić, M. (2004) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research, 23(9), 939-954*

Giordani, S., Lujak, M. & Martinelli, F. (2010). A Distributed Algorithm for the Multi-Robot Task Allocation Problem. *23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, Cordoba, 721-730*

Ha, B., Bae, J., Park, Y., & Kang, S. (2006). Development of process execution rules for workload balancing on agents. *Data & Knowledge Engineering, 56, 64-84.*

Ha, B., Bae, J. & Kang, S. (2004). Workload Balancing on Agents for Business Process Efficiency Based on Stochastic Model. *Business Process Management. BPM 2004. Lecture Notes in Computer Science, 3080, 195-210*

IEEE Xplore Digital Library (2020). *About IEEE Xplore*. Retrieved September 23, 2020 from https://innovate.ieee.org/about-the-ieee-xplore-digital-library/

John, I. & Bhatnagar, S. (2019). Efficient Budget Allocation and Task Assignment in Crowdsourcing. *ACM India Joint International Conference on Data Science and Management of Data, New York, 318–321*

Kamrani, F., Ayani, R. & Karimson, A. (2010). Optimizing a Business Process Model by Using Simulation. *2010 IEEE Workshop on Principles of Advanced and Distributed Simulation*, *Atlanta, 1-8*

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*

Korsah G., Stentz A. & Dias M. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research, 32(12), 1495-1512*

Koubarakis M. & Plexousakis D. (2002). A formal framework for business process modelling and design. *Information Systems, 27, 299-319*

Mitiche, H., Boughaci, D. & Gini, M. (2015). Efficient heuristics for a time-extended multi-robot task allocation problem. *2015 First International Conference on New Technologies of Information and Communication (NTIC), Mila, 1-6*

Pereira, J., Varajão, J. & Uahi, R. (in press). A new approach for improving work distribution in business processes supported by BPMS. *Business Process Management Journal*. Retrieved from https://doi.org/10.1108/BPMJ-05-2019-0184

Pinhey, G., Doucette, J. & Cohen, R. (2014). Distributed multiagent resource allocation with adaptive preemption for dynamic tasks. *2014 international conference on Autonomous agents and multi-agent systems, Richland, 1441–1442*

Robinson, S. (2006). Conceptual modeling for simulation: issues and research requirements. *Proceedings of the 2006 Winter Simulation Conference, 792-800*

Robinson, S. (2013). Conceptual modeling for simulation. *Proceedings of the 2013 Winter Simulation Conference, 377-388*

Rose, O. (2001). The shortest processing time first dispatch rule and some variants in semiconductor manufacturing. *Proceedings of the 2001 Winter Simulation Conference, 1220 – 1224*

Sako, M. (2010). Outsourcing versus shared services. *Communications of the ACM, 53(7), 27-29*

Sharma, P. (2015). Discrete-Event Simulation. *International journal of scientific & technology research, 4(4), 136-140*

Springer. (2020). *Springer Link.* Retrieved September 23, 2020, from https://link.springer.com/

Talib, R., Volz, B. & Jablonski, S. (2011). Agent Assignment for Process Management: Competency-Driven Dynamic Resource Management Methodology. *2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver*, *480-487*

Thesaurus. (2020). *Cambridge Dictionary.* Retrieved September 21, 2020, from https://dictionary.cambridge.org/dictionary/english/

Turner, J. (2018). Distributed Task Allocation Optimisation Techniques. *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18), Richland, 1786–1787*

Venkatraman, N. & Ramanujam, V. (1986). Measurement of Business Performance in Strategy Research: A Comparison of Approaches. *The Academy of Management Review, 11(4), 801-814*

Walsh, P., Mcgregor-Lowndes, M. & Newton, C.J. (2008). Shared services: lessons from thepublic and private sectors for the nonprofit sector. *Australian Journal of Public Administration, 67, 200*

Wang, H. (2007). Performance analysis for shared services. *Communications of the IIMA, 7(2), 61-68*

Wang, S. & Wang, H. (2007) Shared services beyond sourcing the back offices: organizational design. *Human Systems Management, 26, 281-290*

Zhang, K., Collins, E. & Shi, D. (2010). Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. *ACM Transactions on Autonomous and Adaptive Systems, 7(2), 21*

Zhao, H. & Cao, J. (2007). A business process simulation environment based on workflow and multi-agent. *2007 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 1777-1781*

Appendix A – Final set of papers and journals

Zhao, H. & Cao, J. (2007). A business process simulation environment based on workflow and multi-agent. *2007 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 1777-1781*

Kamrani, F., Ayani, R. & Karimson, A. (2010). Optimizing a Business Process Model by Using Simulation. *2010 IEEE Workshop on Principles of Advanced and Distributed Simulation*, *Atlanta, 1-8*

Talib, R., Volz, B. & Jablonski, S. (2011). Agent Assignment for Process Management: Competency-Driven Dynamic Resource Management Methodology. *2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver*, *480-487*

Combi, C. & Pozzi, G. (2006). Task Scheduling for a Temporal Workflow Management System. *Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06), Budapest*, *61-68*

Pereira, J., Varajão, J. & Uahi, R. (in press). A new approach for improving work distribution in business processes supported by BPMS. *Business Process Management Journal*. Retrieved from https://doi.org/10.1108/BPMJ-05-2019-0184

Ha, B., Bae, J., Park, Y., & Kang, S. (2006). Development of process execution rules for workload balancing on agents. *Data & Knowledge Engineering, 56, 64-84.*

Ha, B., Bae, J. & Kang, S. (2004). Workload Balancing on Agents for Business Process Efficiency Based on Stochastic Model. *Business Process Management. BPM 2004. Lecture Notes in Computer Science*, *3080, 195-210*

Gerkey B. & Matarić, M. (2004) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *The International Journal of Robotics Research, 23(9), 939-954*

Dickerson, J., Sankararaman, K., Srinivasan, A. & Xu, P. (2018). Assigning Tasks to Workers based on Historical Data: Online Task Assignment with Two-sided Arrivals. *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18), Richland*, *318–326*

Basu Roy, S., Lykourentzou, I., Thirumuruganathan, S., Amer, S. & Das, G. (2015). Assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal, 24, 467–491*

John, I. & Bhatnagar, S. (2019). Efficient Budget Allocation and Task Assignment in Crowdsourcing. *ACM India Joint International Conference on Data Science and Management of Data, New York, 318–321*

Turner, J. (2018). Distributed Task Allocation Optimisation Techniques. *17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18), Richland, 1786–1787*

Fu, B. & Tari, Z. (2003). A Dynamic Load Distribution Strategy for Systems Under High Task Variation and Heavy Traffic. *ACM Symposium on Applied Computing, 1031-1037*

Giordani, S., Lujak, M. & Martinelli, F. (2010). A Distributed Algorithm for the Multi-Robot Task Allocation Problem. *23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, Cordoba, 721-730*

Pinhey, G., Doucette, J. & Cohen, R. (2014). Distributed multiagent resource allocation with adaptive preemption for dynamic tasks. *2014 international conference on Autonomous agents and multi-agent systems, Richland, 1*441–1442

Zhang, K., Collins, E. & Shi, D. (2010). Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. *ACM Transactions on Autonomous and Adaptive Systems, 7(2), 21*

Mitiche, H., Boughaci, D. & Gini, M. (2015). Efficient heuristics for a time-extended multi-robot task allocation problem. *2015 First International Conference on New Technologies of Information and Communication (NTIC), Mila, 1-6*

Appendix B – Initial search results

Table 1

Literature retrieved in initial search

Engine	Year	Author(s)	Title	Publication name
IEEE	2007	Zhao & Cao	A business process simulation environment based on workflow and multi-agent	2007 IEEE International Conference on Industrial Engineering and Engineering Management
IEEE	2010	Kamrani, Ayani & Karimson	Optimizing a Business Process Model by Using Simulation	2010 IEEE Workshop on Principles of Advanced and Distributed Simulation
IEEE	2011	Talib, Volz & Jablonski	Agent Assignment for Process Management Competency-Driven Dynamic Resource Management Methodology	2011 IEEE 11th International Conference on Data Mining Workshops
IEEE	2015	Aravind & Sandeep	Workflow signature for business process domain A new solution using IBMKD	2015 Global Conference on Communication Technologies (GCCT)
IEEE	2017	Nath & Niyogi	Design and verification of a collaborative task execution procedure using bpmn modeler	2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)
IEEE	2010	Li, Peng, & Liu	*Research on flexible process modeling based on agent organization*	2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)
IEEE	2010	Talib, Volz & Jablonski	Agent Assignment for Process Management Agent Performance Evaluation Framework	2010 IEEE International Conference on Data Mining Workshops
IEEE	2007	Wang, Wang & Liu	Dynamic Supply Chain Integration through Intelligent Agents	2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)
IEEE	2006	Combi & Pozzi	Task Scheduling for a Temporal Workflow Management System	Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06)
IEEE	2009	Zu, Liu & Bai	A Role and Task-Based Workflow Dynamic Authorization Modeling and Enforcement Mechanism	2009 First International Conference on Information Science and Engineering
IEEE	2006	Wang, Wu, Zheng & Jia	Agent based Load Balancing Model for Service based Grid Applications	2006 International Conference on Computational Intelligence and Security
IEEE	2011	Singh, Divakaran & Gonsalves	Taking rural BPO to new heights An ACM for distributed and secure document sharing	2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS)
IEEE	1998	Marazakis, Papadakis & Nikolaou	Management of work sessions in dynamic open environments	Proceedings Ninth International Workshop on Database and Expert Systems Applications (Cat. No.98EX130)
Web of Science	2020	Pereira, Varajao & Uahi	A new approach for improving work distribution in business processes supported by BPMS	Business Process Management Journal
Web of Science	2019	Jemel, Azzouna & Ghedira	RPMInter-work: a multi-agent approach for planning the task-role assignments in inter-organisational workflow	Enterprise Information Systems
Web of Science	2011	Singh, Divakaran & Gonsalves	Taking Rural BPO to New Heights: An ACM for Distributed and Secure Document Sharing	2011 IEEE 5th International Conference on Advanced Networks and Telecommunication Systems (Ants)
Web of Science	2011	Talib, Volz & Jablonski	Agent Assignment for Process Management: Goal Modeling for Continuous Resource Management	Business Process Management Workshops

Engine	Year	Author(s)	Title	Publication name
Web of Science	2009	Wang & Fang	Design of a Supply Chain Focused Enterprise Architecture	2009 Ieee 16th International Conference On Industrial Engineering And Engineering Management, Vols 1 And 2, Proceedings
Web of Science	2007	Zhao & Cao	A business process simulation environment based on workflow and multi-agent	2007 Ieee International Conference On Industrial Engineering And Engineering Management, Vols 1-4
Web of Science	2006	Ha, Bae, Park & Kang	Development of process execution rules for workload balancing on agents	Data & Knowledge Engineering
Web of Science	2006	Combi & Pozzi	Task scheduling for a temporal workflow management system	Time 2006: Thirteenth International Symposium On Temporal Representation And Reasoning, Proceedings
Web of Science	2004	Ha, Bae & Kang	Workload balancing on agents for business process efficiency based on stochastic model	Business Process Management
Web of Science	2004	Gerkey & Mataric	A formal analysis and taxonomy of task allocation in multi-robot systems	
Web of Science	1999	Murthy, Akkiraju, Goodwin et. al.	Enhancing the decision making process for paper mill schedulers	Tappi Finishing & Converting Conference And Trade Fair
ACM	2018	Dickerson, Sankararaman, Srinivasan & Xu	Assigning Tasks to Workers based on Historical Data: Online Task Assignment with Two-sided Arrivals	Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems
ACM	2009	Dahlem & Harrison	Globally Optimal Multi-agent Reinforcement Learning Parameters in Distributed Task Assignment	Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02
ACM	2013	Deb	Collaborative task assignment on tabletop computer	Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces
ACM	2014	Raravi & Nélis	Task Assignment Algorithms for Heterogeneous Multiprocessors	ACM Trans. Embed. Comput. Syst.
ACM	2020	Chen, Cheng, Chen, Lin & Shahabi	Fair task assignment in spatial crowdsourcing	VLDB Endowment
ACM	2007	Zhu & Patek	A distributed, utility-based architecture for task assignment in tactical WSNS	SIGBED Rev.
ACM	2018	Hönig, Kiesel, Tinka, Durham & Ayanian	Conflict-Based Search with Optimal Task Assignment	Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems
ACM	2015	Roy, Lykourentzou, Thirumuruganthan, Amer & Das	Task assignment optimization in knowledge-intensive crowdsourcing	The VLDB Journal
ACM	2006	Abrams, Chen, Guibas, Liu & Zhao	Kinetically stable task assignment for networks of microservers	Proceedings of the 5th international conference on Information processing in sensor networks
ACM	2018	Gong, Huang & Zhang	Task assignment for Eco-friendly Mobile Crowdsensing	Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services
ACM	2017	Badenes, Redondo & Corcho	Distributing Text Mining tasks with <i>librAIry</i>	Proceedings of the 2017 ACM Symposium on Document Engineering
ACM	2019	Lu & Zhou	Optimal Task Assignment Based on Incentive Design in the Virtual Enterprise: a Principal-agent Perspective	Proceedings of the 2019 10th International Conference on E-business, Management and Economics

Engine	Year	Author(s)	Title	Publication name
ACM	2013	Pippin, Christensen & Weiss	Performance based task assignment in multi-robot patrolling	Proceedings of the 28th Annual ACM Symposium on Applied Computing
ACM	2014	Mancuso, Bini & Pannocchia	Optimal Priority Assignment to Control Tasks	ACM Trans. Embed. Comput. Syst.
ACM	2016	Celis, Reddy, Singh & Vaya	Assignment Techniques for Crowdsourcing Sensitive Tasks	Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing
ACM	2019	John & Bhatnagar	Efficient Budget Allocation and Task Assignment in Crowdsourcing	Proceedings of the ACM India Joint International Conference on Data Science and Management of Data
ACM	2017	Liu, Liu, Zhu, Zhang, Fang & Peng	Multiple Assignment in Task Allocation of Communication Base Stations	Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing
ACM	2020	Miao. Kang, Ma, Liu & Chen	Quality-aware Online Task Assignment in Mobile Crowdsourcing	ACM Trans. Sen. Netw.
ACM	2020	Kobayashi, Wakabayashi & Morishima	Quality-aware Dynamic Task Assignment in Human+AI Crowd	Companion Proceedings of the Web Conference 2020
ACM	2018	Cheng, Jian & Chen	An experimental evaluation of task assignment in spatial crowdsourcing	Proc. VLDB Endow. N11 P 1428-1440
ACM	1992	Awerbuch, Kutten & Peleg	Competitive distributed job scheduling (extended abstract)	Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing
ACM	2012	Ciprés, Millán & Errasti	Hybrid method for task schedulling in a distribution center	Proceedings of the Winter Simulation Conference
ACM	2012	Ciprés, Millán & Errasti	Hybrid method for task schedulling in a distribution center	Proceedings of the Winter Simulation Conference
ACM	2018	Turner	Distributed Task Allocation Optimisation Techniques	Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems
ACM	1983	Hudak	Distributed task and memory management	Proceedings of the second annual ACM symposium on Principles of distributed computing
ACM	2007	de Weerdt, Zhang & Klos	Distributed task allocation in social networks	Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems
ACM	2009	Dasgupta, Nino, Garrett et. al.	A multiobjective evolutionary algorithm for the task based sailor assignment problem	Proceedings of the 11th Annual conference on Genetic and evolutionary computation
ACM	2019	Hattab, Ucar, Higuchi et. al.	Optimized Assignment of Computational Tasks in Vehicular Micro Clouds	Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking
ACM	2005	Ortiz, Vincent & Morisset	Task inference and distributed task management in the Centibots robotic system	Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems
ACM	2008	Bobroff, Dasgupta, Fong et. al.	A distributed job scheduling and flow management system	SIGOPS Oper. Syst. Rev. N1
ACM	2018	Turner, Meng, Schaefer & Soltoggio	Fast consensus for fully distributed multi-agent task allocation	Proceedings of the 33rd Annual ACM Symposium on Applied Computing
ACM	2004	Hill & Alford	A distributed task environment for teaching artificial intelligence with agents	Proceedings of the 35th SIGCSE technical symposium on Computer science education
ACM	2004	Hill & Alford	A distributed task environment for teaching artificial intelligence with agents	SIGCSE Bull.

Engine	Year	Author(s)	Title	Publication name
ACM	2006	Dasgupta & Hoeing	Market-Based Distributed Task Selection in Multi-agent Swarms	Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology
ACM	2009	Moallem & Ludwig	Using artificial life techniques for distributed grid job scheduling	Proceedings of the 2009 ACM symposium on Applied Computing
ACM	2000	Luo & Jha	Power-conscious joint scheduling of periodic task graphs and aperiodic tasks in distributed real-time embedded systems	Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design
ACM	2002	Sander, Peleschschuk & Grosz	A scalable, distributed algorithm for efficient task allocation	Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3
ACM	2019	Peccerillo & Bartolini	Task-DAG Support in Single-Source PHAST Library: Enabling Flexible Assignment of Tasks to CPUs and GPUs in Heterogeneous Architectures	Proceedings of the 10th International Workshop on Programming Models and Applications for Multicores and Manycores
ACM	2018	Wei, Wei, Li, Zhuang & Ye	Topology-Aware Task Allocation for Distributed Stream Processing with Latency Guarantee	Proceedings of the 2nd International Conference on Advances in Image Processing
ACM	1988	Kirshnan, Volz & Theriault	Implementation of task types in distributed Ada	Proceedings of the second international workshop on Real-time Ada issues
ACM	1992	Ahmad & Ghafoor	Fault-tolerant task management and load re-distribution on massively parallel hypercube systems	Proceedings of the 1992 ACM/IEEE conference on Supercomputing
ACM	1982	Hudak & Keller	Garbage collection and task deletion in distributed applicative processing systems	Proceedings of the 1982 ACM symposium on LISP and functional programming
ACM	2018	Turner, Meng, Schaefer & Soltoggio	Distributed Strategy Adaptation with a Prediction Function in Multi-Agent Task Allocation	Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems
ACM	1979	Jones & Schwans	TASK forces: Distributed software for solving problems of substantial size	Proceedings of the 4th international conference on Software engineering
ACM	2013	Lin	Context-aware task allocation for distributed agile team	Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering
ACM	2003	Fu & Tari	A dynamic load distribution strategy for systems under high task variation and heavy traffic	Proceedings of the 2003 ACM symposium on Applied computing
ACM	2007	Krainin, An & Lesser	An Application of Automated Negotiation to Distributed Task Allocation	Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology
ACM	2015	von Rosen, Meissner & Hedrich	Semiautomatic implementation of a bioinspired reliable analog task distribution architecture for multiple analog cores	Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition
ACM	2015	Cheung, Southwell, Hou & Huang	Distributed Time-Sensitive Task Selection in Mobile Crowdsensing	Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing
ACM	2003	Goldberg, Cicirello, Dias et. al.	Task allocation using a distributed market-based planning mechanism	Proceedings of the second international joint conference on Autonomous agents and multiagent systems
ACM	2017	Cox & Rossetti	Simulation modeling of alternative staffing and task prioritization in manual post-distribution cross docking facilities	Proceedings of the 2017 Winter Simulation Conference

Engine	Year	Author(s)	Title	Publication name
ACM	1995	McCann & Zahorjan	Scheduling memory constrained jobs on distributed memory parallel computers	Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems
ACM	1995	McCann & Zahorjan	Scheduling memory constrained jobs on distributed memory parallel computers	SIGMETRICS Perform. Eval. Rev.
ACM	2017	Xie, Baytas, Lin & Zhou	Privacy-Preserving Distributed Multi-Task Learning with Asynchronous Updates	Proceedings of the 23rd ACM SIGKDD International Conference
ACM	2015	Kyrkou, Theocharides, Panayiotou & Polycarpou	Distributed adaptive task allocation for energy conservation in camera sensor networks	Proceedings of the 9th International Conference on Distributed Smart Cameras
ACM	2010	Ge, Malani & Qiu	Distributed task migration for thermal management in many-core systems	Proceedings of the 47th Design Automation Conference
ACM	2007	Briquet, Dalem, Jodogne & de Marneffe	Scheduling data-intensive bags of tasks in P2P grids with bittorrent-enabled data distribution	Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks
ACM	2010	Sringer, Giordani, Lujak & Martinelli	A distributed algorithm for the multi-robot task allocation problem	Proceedings of the 23rd international conference on Industrial engineering and other applications of applied intelligent systems - Volume Part I
ACM	2019	Peccerillo & Bartolini	Single-source Library for Enabling Seamless Assignment of Data-parallel Task-DAGs to CPUs and GPUs in Heterogeneous Architectures	Proceedings of the 10th and 8th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms
ACM	2014	Pihney, Doucette & Cohen	Distributed multiagent resource allocation with adaptive preemption for dynamic tasks	Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems
ACM	2013	Trueba, Prieto & Bellas	Distributed embodied evolution for collective tasks: parametric analysis of a canonical algorithm	Proceedings of the 15th annual conference companion on Genetic and evolutionary computation
ACM	2006	González, Vuong & Leung	A mobile code platform for distributed task control in wireless sensor networks	Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access
ACM	2015	Kaiser, Heller, Bourgeouis & Fey	Higher-level parallelization for local and distributed asynchronous task-based programming	Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware
ACM	2016	Schlagkamp, Ferreira da Silva, Allcock et. al.	Consecutive Job Submission Behavior at Mira Supercomputer	Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing
ACM	2020	Shivakumar, Bositty, Peters & Pei	Real-Time Interruption Management System for Efficient Distributed Collaboration in Multi-tasking Environments	Proc. ACM Hum.-Comput. Interact.
ACM	2015	Claes, Robbel, Oliehoek et. al.	Effective Approximations for Multi-Robot Coordination in Spatially Distributed Tasks	Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems
ACM	2009	Cowley, Hsieh & Taylor	Development of top-down analysis of distributed assembly tasks	Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems

Engine	Year	Author(s)	Title	Publication name
ACM	2019	Albani, Hönig & Ayanian	Summary: Distributed Task Assignment and Path Planning with Limited Communication for Robot Teams	Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems
ACM	2005	Wei, Fedak & Cappello	Scheduling Independent Tasks Sharing Large Data Distributed with BitTorrent	Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing
ACM	2019	Tan, Jin, Feng et. al.	Scheduling of Distributed Collaborative Tasks on NDN based MANET	Proceedings of the ACM SIGCOMM 2019 Workshop on Mobile AirGround Edge Computing, Systems, Networks, and Applications
ACM	2012	Zhang, Collins & Shi	Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction	ACM Trans. Auton. Adapt. Syst.
ACM	2015	Maini & Sujit	Distributed task servicing using multiple robots with human-in-the-loop under limited communication range	Proceedings of the 30th Annual ACM Symposium on Applied Computing
ACM	2004	Springer, Feng & Cai	MCCF: a distributed grid job workflow execution framework	Proceedings of the Second international conference on Parallel and Distributed Processing and Applications
ACM	2017	Rosa & Rocha	Exportation to the cloud of distributed robotic tasks implemented in ROS	Proceedings of the Symposium on Applied Computing
ACM	1989	Royce	Reliable, reusable Ada components for constructing large, distributed multi-task networks: networks architecture services (NAS)	Proceedings of the conference on Tri-Ada '89: Ada technology in context: application, development, and deployment
ACM	2008	Shrinivas & Naughton	Issues in applying data mining to grid job failure detection and diagnosis	Proceedings of the 17th international symposium on High performance distributed computing
ACM	2018	Kantaros & Zavlanos	Distributed optimal control synthesis for multi-robot systems under global temporal tasks	Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems
ACM	2020	Zhang & Go	Anomaly detection for NILM task with Apache Flink	Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems
ACM	2008	Barlow, Oh & Smit	Evolving cooperative control on sparsely distributed tasks for UAV teams without global communication	Proceedings of the 10th annual conference on Genetic and evolutionary computation
ACM	2004	Czumaj & Ronen	On the expected payment of mechanisms for task allocation	Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing
ACM	2019	Suh, Woo, Kim & Park	A sim2real framework enabling decentralized agents to execute MADDPG tasks	Proceedings of the Workshop on Distributed Infrastructures for Deep Learning
ACM	2015	De Rango, Palmieri, Yang & Marano	Bio-inspired exploring and recruiting tasks in a team of distributed robots over mined regions	Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems
ACM	2014	Prisacari, Rodriguez, Heidelberger et. al.	Efficient task placement and routing of nearest neighbor exchanges in dragonfly networks	Proceedings of the 23rd international symposium on High-performance parallel and distributed computing
ACM	2003	Xi, Cao & Berman	Supply chain simulation: distributed supply chain simulation using a generic job running framework	Proceedings of the 35th conference on Winter simulation: driving innovation
ACM	2011	Caniou, Charrier & Desprez	Evaluation of reallocation heuristics for moldable tasks in computational grids	Proceedings of the Ninth Australasian Symposium on Parallel and Distributed Computing - Volume 118
ACM	2010	Emek, Halldórsson & Mansour	Online set packing and competitive scheduling of multi-part tasks	Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing

Engine	Year	Author(s)	Title	Publication name
ACM	2009	Khalid, Anthony & Nilsson	Enabling and optimizing pilot jobs using xen based virtual machines for the HPC grid applications	Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing
ACM	2010	Luckow, Lacinski & Jha	SAGA BigJob: An Extensible and Interoperable Pilot-Job Abstraction for Distributed Applications and Systems	Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing
ACM	2013	Rajachandrasekar, Moody, Mhoror & Panda	A 1 PB/s file system to checkpoint three million MPI tasks	Proceedings of the 22nd international symposium on High-performance parallel and distributed computing
ACM	2013	Rajachandrasekar, Moody, Mhoror & Panda	A 1 PB/s file system to checkpoint three million MPI tasks	Proceedings of the 22nd international symposium on High-performance parallel and distributed computing
ACM	2003	Batalin & Sukhatme	Poster abstract: sensor network as a distributed manager for multi-robot task allocation	Proceedings of the 1st international conference on Embedded networked sensor systems
ACM	2016	Yeung, Madria, Linderman & Milligan	Routing and scheduling of spatio-temporal tasks for optimizing airborne sensor system utilization	Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems
ACM	2018	Gholkar, Mueller, Rountree & Marathe	PShifter: feedback-based dynamic power shifting within HPC jobs for performance	Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing
ACM	2009	Costa, Brasileiro, Filho & Sousa	OddCI: on-demand distributed computing infrastructure	Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers
ACM	2006	Fraigniaud, Ilcinkas & Pelc	Oracle size: a new measure of difficulty for communication tasks	Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing
Springer Link	2003	Shepherdson, Lee & Mihailescu	mPower â€” A Component-Based Development Framework for Multi-Agent Systems to Support Business Processes	BT Technology Journal
Springer Link	2007	Shepherdson, Lee & Mihailescu	mPower â€” a component-based development framework for multi-agent systems to support business processes	BT Technology Journal
Springer Link	2000	Wang & Zhong	The distributed workflow management system â€” Flow agent	Journal of Computer Science and Technology
Springer Link	2011	Delias, Doulamis & Matsatsinis	What agents can do in workflow management systems	Artificial Intelligence Review
Springer Link	2000	Arpinar, Dogac & Tatbul	An open electronic marketplace through agent-based workflows: MOPPET	International Journal on Digital Libraries
Springer Link	2020	Shyalika, Silva & Karunananda	Reinforcement Learning in Dynamic Task Scheduling: A Review	SN Computer Science
Springer Link	2019	Mirkov, Bakic & Djapic	RFID technology in the function of generating flexible robotic sequences of the FMC	Journal of the Brazilian Society of Mechanical Sciences and Engineering
Springer Link	2006	Sugumaran, Dietrich & Kirn	Supporting mass customization with agent-based coordination	Information Systems and e-Business Management
Springer Link	2006	Maheswaran, Pearce, Bowring et. al.	Privacy Loss in Distributed Constraint Reasoning: A Quantitative Framework for Analysis and its Applications	Autonomous Agents and Multi-Agent Systems

Engine	Year	Author(s)	Title	Publication name
Springer Link	2003	Mitiche, Boughaci & Gini	Efficient Heuristcs for a time-extend multi-robot task allocation problem	Information Systems and e-Business Management
Springer Link	2006	Felner, Shoshani, Altshuler et. al.	Multi-agent Physical A* with Large Pheromones	Autonomous Agents and Multi-Agent Systems
Springer Link	1999	Shen & Norrie	Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey	Knowledge and Information Systems
Springer Link	2001	Li, Lei, Ying et. al.	Geo-Agents: Design and implement	Wuhan University Journal of Natural Sciences
Springer Link	2011	Barkat, Ruhul & Lokan	Handling equality constraints with agent-based memetic algorithms	Memetic Computing
Springer Link	1999	Mountzia & Rodosek	Using the Concept of Intelligent Agents in Fault Management of Distributed Services	Journal of Network and Systems Management
Springer Link	2014	Gilliland & Kim	When do incentives work in channels of distribution?	Journal of the Academy of Marketing Science
Springer Link	2004	Huang & Lai	Knowledge management system: an agent-based approach	Knowledge Management Research & Practice
Springer Link	2019	Kanno, Koike & Furuta	Human-centered modeling framework of multiple interdependency in urban systems for simulation of post-disaster recovery processes	Cognition, Technology & Work
Springer Link	2001	Silva, Romão, Deugo & da Silva	Towards a Reference Model for Surveying Mobile Agent Systems	Autonomous Agents and Multi-Agent Systems
Springer Link	2007	van der Zee	Developing participative simulation models-framing decomposition principles for joint understanding	Journal of Simulation
Springer Link	2019	Janiesch, Fischer, Winkelmann & Nentwich	Specifying autonomy in the Internet of Things: the autonomy model and notation	Information Systems and e-Business Management
Springer Link	2003	Ströbel & Weinhardt	The Montreal Taxonomy for Electronic Negotiations	Group Decision and Negotiation
Springer Link	2003	Vassileva, McCalla & Greer	Multi-Agent Multi-User Modeling in I-Help	User Modeling and User-Adapted Interaction
Springer Link	2010	Hübner, Boissier, Kitio & Ricci	Instrumenting multi-agent organisations with organisational artifacts and agents	Autonomous Agents and Multi-Agent Systems
Springer Link	2002	Szirbik	A Negotiation Enabling Agent Based Infrastructure: Composition and Behavior	Information Systems Frontiers
Springer Link	2018	Kemchi, Zitouni & Djoudi	AMACE: agent based multi-criterions adaptation in cloud environment	Human-centric Computing and Information Sciences
Springer Link	2006	Riley & Veloso	Coach planning with opponent models for distributed execution	Autonomous Agents and Multi-Agent Systems
Springer Link	1998	Moukas & Maes	Amalthaea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW	Autonomous Agents and Multi-Agent Systems
Springer Link	2019	Porter, Brock, Estabrooks et. al.	SIPsmartER delivered through rural local health districts: adoption and implementation outcomes	BMC Public Health
Springer Link	2001	Batten	Complex landscapes of spatial interaction	The Annals of Regional Science

Engine	Year	Author(s)	Title	Publication name
Springer Link	2006	Nahm & Ishikawa	An Internet-based integrated product design environment. Part II: its applications to concurrent engineering design	The International Journal of Advanced Manufacturing Technology
Springer Link	2004	Ha, Bae & Kang	Workload Balancing on Agents for Business Process Efficiency Based on Stochastic Model	Business Process Management
Springer Link	2009	Jones & Barber	Combining Job and Team Selection Heuristics	Coordination, Organizations, Institutions and Norms in Agent Systems IV
Springer Link	2002	Müller, Bauer & Berger	Software Agents for Electronic Business: Opportunities and Challenges	Multi-Agent Systems and Applications II
Springer Link	2006	Yan, Weiping, Haicheng et. al.	Towards an Agent and Knowledge Enacted Dynamic Workflow Management System for Intelligent Manufacturing Grid	Cooperative Design, Visualization, and Engineering
Springer Link	2010	Millán-Ruiz & Hidalgo	A Memetic Algorithm for Workforce Distribution in Dynamic Multi-Skill Call Centres	Evolutionary Computation in Combinatorial Optimization
Springer Link	2006	Ha, Reijers, Bae & Bae	An Approximate Analysis of Expected Cycle Time in Business Process Execution	Business Process Management Workshops
Springer Link	2018	Ivaschenko, Lednev, Diyazitdinova & Sitnikov	Agent-Based Outsourcing Solution for Agency Service Management	Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016
Springer Link	2019	Nodine	Communication and Coordination Support for Mobile, Transient and Distributed Agent Applications	Innovative Concepts for Agent-Based Systems
Springer Link	2004	Schillo, Fischer, Fley et. al.	FORM - A Sociologically Founded Framework for Designing Self-Organization of Multiagent Systems	Regulated Agent-Based Social Systems
Springer Link	2005	Yamamoto	Agent Server Technology for Managing Millions of Agents	Massively Multi-Agent Systems I
Springer Link	2010	Jurasovic, Kusek & Jezic	Team Formation and Optimization for Service Provisioning	Agent and Multi-Agent Systems: Technologies and Applications
Springer Link	2004	Kim, Yang & Lee	A Study on the Specification for e-Business Agent Oriented Component Based Development	Software Engineering Research and Applications
Springer Link	2021	Sunder, Vig, Chatterjee & Shroff	Prosocial or Selfish? Agents with Different Behaviors for Contract Negotiation Using Reinforcement Learning	Advances in Automated Negotiations
Springer Link	2012	Ivanovic, Vidakovic, Mitrovic & Budimac	Evolution of Extensible Java EE-Based Agent Framework	Agent and Multi-Agent Systems. Technologies and Applications
Springer Link	2005	Gorodetsky, Karsaev, Samoylov et. al.	Multi Agent System Development Kit	Software Agent-Based Applications, Platforms and Development Kits
Springer Link	2015	Jarrah, Zeigler, Xu & Zhang	A Multi-agent Simulation Framework to Support Agent Interactions under Different Domains	Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1
Springer Link	1999	Sullivan, Glass, Grosz & Kraus	Intention Reconciliation in the Context of Teamwork: An Initial Empirical Investigation	Cooperative Information Agents III
Springer Link	2002	Lingnau, Matthes & Drobnik	Supporting Mobility through Computer Networks	Networks

Engine	Year	Author(s)	Title	Publication name
Springer Link	2008	Pham, Harland & Winikoff	Modeling Agents' Choices in Temporal Linear Logic	Declarative Agent Languages and Technologies V
Springer Link	2001	Liu	An Adaptive Agent Society for Environmental Scanning through the Internet	Intelligent Agents: Specification, Modeling, and Applications
Springer Link	2013	Lützenberger, Küster, Konnerth et. al.	A Multi-agent Approach to Professional Software Engineering	Engineering Multi-Agent Systems
Springer Link	2015	Junior, de Freitas Filho & Silveira	E-HIPS: An Extention of the Framework HIPS for Stagger of Distributed Process in Production Systems Based on Multiagent Systems and Memetic Algorithms	Advances in Artificial Intelligence and Soft Computing
Springer Link	2001	Camarinha-Matos, Afsarmanesh & Rabelo	Supporting Agility in Virtual Enterprises	E-Business and Virtual Enterprises
Springer Link	2001	Bergenti, Poggi & Rimassa	Enabling FIPA Agents on Small Devices	Cooperative Information Agents V
Springer Link	2006	Sombattheera & Ghose	Supporting Dynamic Supply Networks with Agent-Based Coalitions	Advances in Applied Artificial Intelligence

Appendix C – Analysis of Arrival Behavior

Conform the methodology described in previous section, this section shows the outcome of the analysis of arrival behavior, performed on pilot data and general information.

For a period of eight months, four client facing employees sent tasks to four assistants using a worklist in the CRM-system, piloting the way the new Business Support will receive its tasks. During this period, 872 tasks have been placed in the worklist and have been finished by the employees.

Each task has been categorized to one of the 11 task type categories as described earlier, providing with a total of J occurrences of task n during the pilot period. For each type n, a total of $J - 1$ interarrival times has been calculated by subtracting the moment instance $j + 1$ entered the list from the moment instance j entered the list. The data input used for this calculation is shown in Table 9. As is shown in Table 9, the interarrival time T_{ia}, in minutes, is only recorded during working hours.

Table 9
Display of the data input used for pilot data analysis on arrival behaviour

Instance	Type	Start	T_{ia}
1	11. Change authorizations	23/11/2020 16:26	
2	11. Change authorizations	19/11/2020 14:34	1072
3	11. Change authorizations	18/11/2020 17:29	334
4	11. Change authorizations	18/11/2020 16:25	64
5	11. Change authorizations	17/11/2020 14:31	594
6	11. Change authorizations	15/11/2020 15:27	811
7	11. Change authorizations	15/11/2020 15:25	0
8	11. Change authorizations	15/11/2020 15:22	0
9	11. Change authorizations	15/11/2020 15:01	0
10	11. Change authorizations	12/11/2020 16:28	512
11	11. Change authorizations	09/11/2020 14:51	1537
…		…	…
J		…	…

For all tasks, excluding tasks 6 and 7 of which arrival behaviors will be calculated differently, the number of instances and interarrival times during the pilot have been determined. The results are shown in Table 10. Next to the number of instances, the minimum, maximum, averaged, and median interarrival time is given.

Table 10

Summary of task arrival numbers and interarrival times during pilot (minutes)

Task	J	$T_{ia,min}$	$T_{ia,min}$	$T_{ia,avg}$	$T_{ia,med}$
1. Set up New Client	93	0	5900	909	380
2. Set up New Product	91	0	9144	960	501
3. Change Client Data	226	0	3334	386	202
4. Change Banking Product or Contract	179	0	4701	494	259
5. Change Investment Product or Contract	34	1	14718	2571	1323
6. Prepare Banker and Client Meeting					
7. Prepare 'Know Your Client' Revision					
8. Prepare 'New Client Compliance Check	21	4	29283	4237	1603
9. Process Client Request	147	0	3742	591	305
10. Start the Deceased Client Process	24	7	12810	3720	3129
11. Change Authorizations	57	0	13473	1503	628

As arrival behavior in queueing systems can often be described by rate λ, corresponding to an exponential distribution, the next step in the data analysis is the verification of the distribution.

Histograms showing the distribution of interarrival times of each task are shown in Figure 7. Each bin has a bin size expressed in minutes. This means, for the first task, the first bin contains interarrival times between 0 and 980 minutes, the second bin contains times between 980 and 2060, and so on. The number of interarrival times in each bin is represented by the height of the bar and corresponds with the numbers of the y-axis.

The arrival behavior visible in the histograms of Figure 7 appear to follow an exponential distribution. To test whether this is the case, a chi-square goodness-of-fit test is performed. The following hypothesis is tested:

$$H_0: \quad T_{ia,n} \sim f_n(x) = \lambda e^{-\lambda x} \quad for\ n \in N$$

$$H_1: \quad T_{ia,n} \ncong f_n(x) = \lambda e^{-\lambda x} \quad for\ n \in N$$

By applying the chi-square test, the data is divided over k bins. The test statistic χ^2 is calculated as follows:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

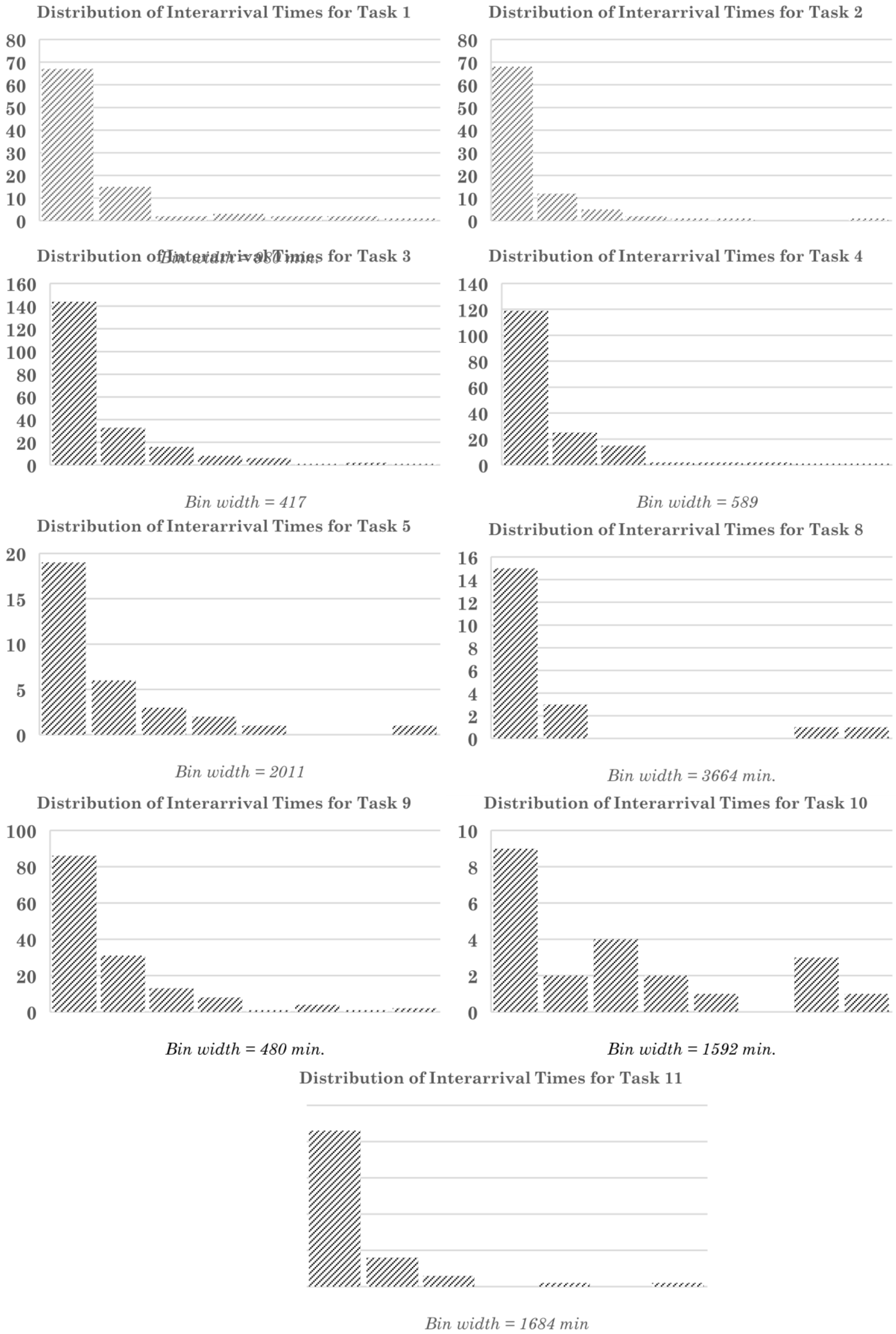

Figure 7: Histograms of the distributions of interarrival times T_{ia} for tasks in Table 10.

The in the dataset observed frequency in bin i is described with O_i. The expected frequency is described with E_i and calculated as:

$$E_i = N(F(Y_u) - F(Y_l))$$

where F corresponds to the tested distribution function, and Y describes the upper u and lower l limits for class i. The test follows a chi-square distribution with $k - c - 1$ degrees of freedom, where c corresponds with the number of estimated parameters and k with the number of bins.

An example of this chi-square test, for task 10, is given in Table 11. With six degrees of freedom, the critical value for $\alpha = 0.01$ equals 16.81. As χ^2_{10} is 11.45 and does not exceed the critical value, the null hypothesis for task 10 can not be rejected.

Table 11
Chi-square test

Class	Class intervals		O_i	E_i	χ^2
1	0	418	5	2,75	1,84
2	418	1318	4	2,75	0,57
3	1318	2789	1	2,75	1,11
4	2789	4958	6	2,75	3,84
5	4958	8028	2	2,75	0,20
6	8028	12366	3	2,75	0,02
7	12366	18874	1	2,75	1,11
8	18874	∞	0	2,75	2,75
			22	**22**	**11,45**

$k = 8$, $s = 1$, k-s-$1 = 6$, $\lambda = 0.00027$, $p = 0.0125$, $n = 22$

The results for the test on all tasks are given in Table 12. As can be seen in Table 12, none of the χ^2 values exceed the critical value of 16.81, This means for each task the exponential distribution has been proven and follows rate λ_n as given in the table. The described rate λ_n however, applies to the behavior during the pilot, which limits both timespan and client group. Extrapolating the number of occurrences and thus interarrival times for a period of 12 months, and a client base of 8900, the actual expected arrival rate for region 'Average', is given by the last column.

Table 12
Results of chi-square test for all tasks and expected arrival rates

Task	**$\#_{pilot}$**	**Tia**	**λ**	**$\chi 2$**	**$\#_{exp}$**	**λ_{exp}**
1. Set up New Client	93	909	0,01101	8,52	1299	0,0103
2. Set up New Product	91	960	0,01042	13,47	1229	0,0098
3. Change Client Data	226	386	0,02588	16,5	3054	0,0243
4. Change Banking Product or Contract	179	494	0,02023	15,56	2387	0,0190
5. Change Investment Product or Contract	34	2571	0,00389	7,50	459	0,0036
6. Prepare Banker and Client Meeting			0,04529			
7. Prepare 'Know Your Client' Revision			0,02989			
8. Prepare 'New Client Compliance Check	21	4237	0,00236	16,00	278	0,0022
9. Process Client Request	147	591	0,01692	11,59	1996	0,0159
10. Start the Deceased Client Process	24	3720	0,00269	11,45	317	0,0025
11. Change Authorizations	57	1503	0,00665	4,57	785	0,0062

As stated before, all tasks except 6 and 7 are of a reactive character, meaning they occur due to an externally arisen need. Tasks 6 and 7 however, occur proactively and are mostly planned.

Task 6, the preparation of client meetings, will be sent to the Business Support 'Average' for a maximum number of 5,344 times a year. This number corresponds to 64% of the 8.350 regional clients, since all clients are aimed to be met at least once a year but 36% of client meetings will be prepared by commercial assistants working on the local offices and therefore not by Business Support. While being planned and distributed uniformly over the year, the arrival rate is transformed in an exponential arrival with interarrival time:

$$\lambda_6 = \frac{1}{t_{ia,6}} = \frac{1}{\frac{yearly\ working\ minutes}{number\ of\ instances}} = \frac{1}{\frac{125760}{5969}} = 0.045$$

The same applies for task 7, the preparation of the Know Your Customer Client Revision. This task is sent to Business Support once a year, for 42% of the regional clients. This leads to an arrival rate of λ = 0.030. With this information, the list of expected arrival rates can be complemented. Table 13 shows the final results of the arrival rates that will be used in the simulation.

Table 13
Arrival rates as result of the arrival behaviour analysis

Task	**λ**
1. Set up New Client	0,01101
2. Set up New Product	0,01042
3. Change Client Data	0,02588
4. Change Banking Product or Contract	0,02023
5. Change Investment Product or Contract	0,00389
6. Prepare Banker and Client Meeting	0,04529
7. Prepare 'Know Your Client' Revision	0,02989
8. Prepare 'New Client Compliance Check	0,00236
9. Process Client Request	0,01692
10. Start the Deceased Client Process	0,00269
11. Change Authorizations	0,00665

Appendix D – Analysis of Processing Times

For a period of four weeks, four assistants kept a journal. After finishing a task, employees noted how much time they spent working on the task, and what the content of the task was. Combined, a total of 1,777 journal entries are recorded. 22% of the recorded activities categorize as one of the task types that Business Support will carry out in the future. For each employee, the average processing time of each task type is calculated. During workshops, held with assistants, bankers, process consultants, and directors, estimates on processing times were also obtained. The result of both efforts is shown in Table 14. For the empty cells, employees did not record any activities of this type during the period of journaling.

Table 14
Average processing times, in minutes, of tasks by agents, and estimates from workshops

	Employee				**Workshop**
Task	**1**	**2**	**3**	**4**	
1. Set up New Client	37	-	24	25	30
2. Set up New Product	10	15	9	10	20
3. Change Client Data	23	19	22	15	15
4. Change Banking Product or Contract	18	16	27	16	20
5. Change Investment Product or Contract	-	30	24	15	20
6. Prepare Banker and Client Meeting	-	-	-	-	30
7. Prepare 'Know Your Client' Revision	-	-	-	-	45
8. Prepare 'New Client Compliance Check	-	-	10	10	60
9. Process Client Request	15	9	15	19	30
10. Start the Deceased Client Process	25	-	34	n/a	90
11. Change Authorizations	5	-	10	17	25

To get insight in mutual differences between agents, the average processing time of each task by an agent, is divided by with the average processing time of that task. This calculation first yields the speed ratio of that task by that agent, and when averaged, yields the agent speed ratio. The result of this comparison is shown in Table 15.

The speed ratios are used to create four fictional agents and to fill the gaps for the existing agents. First, four new average agent speed ratios are semi-randomly chosen. With values 0.97. 0.99, 1.01 and 1.03, these values fit in the existing range of average agent speed ratios. Then, for each agent, task speed ratios are semi-randomly chosen. When averaged, the task speed ratio's match the earlier determined agent speed ratio. All task speed ratios fit in the range of existing task speed ratios. The same is done for empty values of the existing agents.

The results of this semi-random task and agent speed radio generation is shown in Table 16. The task speed ratios are multiplied with average task processing times, to obtain new processing times, for each agent, for each task. The workshop times are used for missing processing times of task 6 and 7. Generated processing times that exceeding the range of existing processing times are limited with the maximum or minimum. The result of this is shown in Table 17.

Table 15
Task and agent speed ratios for four agents

	Relative speed per agent						
Task	**1**	**2**	**3**	**4**	**Min**	**Avg**	**Max**
1. Set up New Client	1,29		0,84	0,87	0,84	1,00	1,29
2. Set up New Product	0,91	1,36	0,82	0,91	0,82	1,00	1,36
3. Change Client Data	1,16	0,96	1,11	0,76	0,76	1,00	1,16
4. Change Banking Product or Contract	0,94	0,83	1,40	0,83	0,83	1,00	1,40
5. Change Investment Product or Contract		1,30	1,04	0,65	0,65	1,00	1,30
6. Prepare Banker and Client Meeting							
7. Prepare 'Know Your Client' Revision							
8. Prepare 'New Client Compliance Check			1,00	1,00	1,00	1,00	1,00
9. Process Client Request	1,03	0,62	1,03	1,31	0,62	1,00	1,31
10. Start the Deceased Client Process	0,85		1,15		0,85	1,00	1,15
11. Change Authorizations	0,47		0,94	1,59	0,47	1,00	1,59
Agent speed ratio	**0,95**	**1,02**	**1,04**	**0,99**	**0,95**	**1,00**	**1,04**

Table 16
Filled and generated task and agent speed ratios

	Agent							
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1. Set up New Client	1,29	0,89	0,84	0,87	1,10	0,95	1,20	1,02
2. Set up New Product	0,91	1,36	0,82	0,91	0,95	0,99	1,10	0,99
3. Change Client Data	1,16	0,96	1,11	0,76	0,93	1,13	0,90	1,30
4. Change Banking Product or Contract	0,94	0,83	1,40	0,83	0,83	1,30	0,93	1,20
5. Change Investment Product or Contract	0,70	1,30	1,04	0,65	1,03	0,73	0,95	0,90
6. Prepare Banker and Client Meeting	1,10	0,93	1,20	1,10	0,95	1,03	1,10	0,90
7. Prepare 'Know Your Client' Revision	1,05	1,10	0,94	0,92	0,96	0,97	1,02	0,99
8. Prepare 'New Client Compliance Check	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9. Process Client Request	1,03	0,62	1,03	1,31	1,03	0,70	0,98	0,99
10. Start the Deceased Client Process	0,85	1,15	1,15	0,94	0,90	1,02	0,97	1,05
11. Change Authorizations	0,47	1,10	0,94	1,59	0,90	1,05	1,01	1,02
Agent speed ratio	**0,95**	**1,02**	**1,04**	**0,99**	**0,97**	**0,99**	**1,01**	**1,03**

Table 17
Analysis based processing times for fictional agents

	Agent							
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1. Set up New Client	37	26	24	25	32	27	34	29
2. Set up New Product	10	15	9	10	10	11	12	11
3. Change Client Data	23	19	22	15	18	22	18	26
4. Change Banking Product or Contract	18	16	27	16	16	25	18	23
5. Change Investment Product or Contract	16	30	24	15	24	17	22	21
6. Prepare Banker and Client Meeting	33	28	36	33	29	31	33	27
7. Prepare 'Know Your Client' Revision	47	50	42	41	43	44	46	45
8. Prepare 'New Client Compliance Check	10	10	10	10	10	10	10	10
9. Process Client Request	15	9	15	19	15	10	14	14
10. Start the Deceased Client Process	25	34	34	28	27	30	29	31
11. Change Authorizations	5	12	10	17	10	11	11	11

The last step of the analysis and generation of processing times is transforming the processing times to processing rates. Unlike the previous section, in which the exponential distribution of interarrival times was proven, this proof is not generated by the analysis of processing times. For this reason, it's assumed the average processing times follow an exponential distribution. This way, the processing rate μ is found with $\mu = 1 / tp_{rocess}$. The final set of processing rates used for the simulation is shown in Table 18.

Table 18
Processing rates for all tasks and agents

	Agent							
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1. Set up New Client	0,027	0,039	0,042	0,040	0,032	0,037	0,029	0,034
2. Set up New Product	0,100	0,067	0,111	0,100	0,096	0,092	0,083	0,092
3. Change Client Data	0,043	0,053	0,045	0,067	0,054	0,045	0,056	0,039
4. Change Banking Product or Contract	0,056	0,063	0,037	0,063	0,063	0,040	0,056	0,043
5. Change Investment Product or Contract	0,062	0,033	0,042	0,067	0,042	0,060	0,046	0,048
6. Prepare Banker and Client Meeting	0,030	0,036	0,028	0,030	0,035	0,032	0,030	0,037
7. Prepare 'Know Your Client' Revision	0,021	0,020	0,024	0,024	0,023	0,023	0,022	0,022
8. Prepare 'New Client Compliance Check	0,100	0,100	0,100	0,100	0,100	0,100	0,100	0,100
9. Process Client Request	0,067	0,111	0,067	0,053	0,067	0,099	0,070	0,070
10. Start the Deceased Client Process	0,040	0,029	0,029	0,036	0,038	0,033	0,035	0,032
11. Change Authorizations	0,200	0,085	0,100	0,059	0,104	0,089	0,093	0,092

Appendix E – Stakeholder Analysis

This section briefly discusses stakeholders relevant to the research problem and process, for the purpose of displaying a complete overview of the situation of the organization. For each stakeholder, its role and goal is mentioned.

1. *Client:* purchase products, requests service, expects the high level of service the private bank aims to distinguish itself with
2. *Private banker:* primary point of contact for the client. Responsible for maintaining a good relationship with the client. Gives reactive and pro-active advice and develops commercial opportunities. Wants the best for its clients and expects the organization to enable and support him/her in doing so
3. *Assistant private banker:* secondary point of contact for the client. Supports the private banker with operational activities. Wants the best for its private bankers and clients and expects a fast and flawless handling of processes it initiates at other places within the bank
4. *Investment manager:* primary point of contact for clients that have invested assets with the bank. Gives advice on investing and manages clients invested assets. Wants the best for its clients and expects the organization to enable and support him/her in doing so
5. *Assistant investment manager:* secondary point of contact for the client. Supports the investment manager with operational activities. Wants the best for its investment managers and clients and expects a fast and flawless handling of processes it initiates at other places within the bank
6. *Local office director:* manages the private bankers and assistant private bankers of a local office. Oversees performance, is the driver of commerciality, is responsible for quality and is the spokesperson of his/her team within the wider organization. Wants satisfied clients, a satisfied staff and wants to meet all commercial and non-commercial targets.
7. *Director Investments:* manages the investment managers and assistant investment managers of a region. Oversees performance, is responsible for quality and is the spokesperson of his/her team within the wider organization. Wants satisfied clients, a satisfied staff and wants to meet all commercial and non-commercial targets.
8. *Regional director:* manages the local office directors and director investments of his/her region. Oversees performance, implements strategic decisions and serves as a link between headquarters and the region. Wants a flawless collaboration between his/her local offices and its regional BS.
9. *Manager of BS:* manages the employees of Business Support. Responsible for delivering the desired performance and for operational excellence of the activities. Wants to be able to process tasks to the satisfaction of the local offices it serves.

10. *Employee of BS (Client Support):* processes tasks from the tasklist in order to support the local offices. Wants to have clear describes tasks it can finish smoothly.
11. *Employee of BS (Process Support):* processes quality, risk and compliance related checks and tasks for the local offices of its region. Wants the work of other employees to be of high quality, since its their job to indicate and fix errors
12. *CSM Transformation Lead:* manages the transformation. Is the host and supervisor of the research. Wants the implementation to be a success and wants to provide the future managers of BS with concrete input to achieve high performance

Appendix F – Pseudocode Simulation

Algorithm 1 Pseudocode for task selecting model

Given S agents (a_1, a_2, …, a_S) and N tasks (t_1, t_2, …, t_N) and time t
Given interarrival times (λ_1, λ_2,…, λ_N) and processing times (μ_1, μ_2,…, μ_N)
Procedure Simulate(R, T, W)
Input Number of runs R, Run duration T, warm-up W
For run in R:
While t < T
    Advance Time()
    If t = W:
        Start Recording()
    Append Results()
Return Results
Procedure Advance Time()
Generate Tasks()
Check Agents()
t = t + 1
Procedure Generate Tasks()
For n in N:
If arrival_time_n < t
    Assignment_Queue.append($task_n$)
    arrival_time_n = arrival_time_n + random_variable(λ_n)
    Append Task Information (Arrival Time, Task Type)
Procedure Check Agents()
For a in A:
If release_time_a < t
    Get Task()
    agent_s_release_t = agent_s_release_t + random_variable($\mu_{s,n}$)
    Append Task Information (Processing Time, Agent, Waiting Time)
Procedure Get Task()
<insert Task Assignment Method>
Return Task Number for To Be Assigned Task

Algorithm 2 Pseudocode for task selecting model

Given S agents (a_1, a_2, …, a_S) and N tasks (t_1, t_2, …, t_N) and time t
Given interarrival times (λ_1, λ_2,…, λ_N) and processing times (μ_1, μ_2,…, μ_N)
Procedure Simulate(R, T, W)
Input Number of runs R, Run duration T, warm-up W
For run in R:
While t < T
    Advance Time()
    If *t = W:*
      Start Recording()
    Append Results()
Return Results
Procedure Advance Time()
Generate Tasks()
Check Assignment Queue()
Check Agent Queues()
t = t + 1
Procedure Generate Tasks()
For n in N:
If arrival_time_n < t
  Assignment_queue.append(task$_n$)
  arrival_time_n = arrival_time_n + random_variable(λ_n)
  Append Task Information (Arrival Time, Task Type)
Procedure Check Assignment Queue()
If Assignment_queue > 0:
  Get Agent()
  Agent_s_queue.append(task$_n$)
  Assignment_queue.delete(task$_n$)
  Append Task Information (Assignment Time, Agent)
Procedure Check Agent Queues()
For s in S:
If Agent_s_queue > 0 **and** agent_s_release_t < t:
  To Be Assigned Task = Agent_s_queue[0]
  agent_s_release_t = agent_s_release_t + random_variable($\mu_{s,n}$)
  Append Task Information (Processing Time, Waiting Time)
Procedure Get Agent()
<insert Task Assignment Method>
Return *to_be_assigned_agent_s*

Appendix G – Verification of Simulation

The model is verified using a statistical analysis. The model is verified on number of generated tasks by the system and the average processing times, as these number corresponded with respectively the simulated arrival behavior and agent behavior. Following the approach of Hoad & Robinson (2010), a for each run updated t-student test is applied.

The simulation was run 25 times, for 10 weeks, excluding two weeks of warming up. Using the arrival and processing rates as described in previous section, a population mean μ_t of 4238 is found for the number of generated tasks. The population mean μ_p of 25,31 is also retrieved from the agent behavior analysis.

The simulated runs are used to generate sample means X_t for the number of generated tasks and X_p for average processing time. Each run, a new $X_{t,n}$ *and* $X_{p,n}$ is calculated, based on new run n, and the results from preceding *[1, 2 …, n]*.

With the results of 25 runs, the t_n-scores per run n are calculated by:

$$t_n = \frac{X_n - \mu}{\frac{S}{\sqrt{n}}}$$

where S is the standard deviation and n is the number of runs. The results of this test are shown in Table 14. The critical t-value for the significance level $\alpha = 0.05$ is shown in the last column. Runs in which the found t-values exceed the critical value are marked As shown in Table 15, at the 25th run, the t-level for generated tasks is below the critical value for 14 consecutive runs, and the t-level for processing time is below the critical value for 5 runs. This calculation verifies the model as a representation of the model requirements and design specifications, and also provides with a number of required runs of 25.

Table 14
T-test for each simulated run, on number of generated tasks and processing time

n	$X_{t,n}$	$X_{p,n}$	$S_{t,n}$	$S_{p,n}$	$t_{t,n}$	$t_{p,t}$	$t_{\alpha=0,05}$
1	4314	25,29					6.31
2	4243	25,52	101,12	0,32	0,045	0,911	2.92
3	4253	25,35	73,64	0,36	0,282	0,217	2.35
4	4277	25,35	77,04	0,29	0,871	0,250	2.13
5	4280	25,32	67,21	0,26	1,262	0,081	2.02
6	4280	25,31	60,13	0,23	1,556	0,016	1.94
7	4284	25,27	55,92	0,24	**2,009**	0,472	1.90
8	4274	25,25	58,97	0,23	1,610	0,698	1.86
9	4273	25,30	55,18	0,26	1,811	0,122	1.83
10	4275	25,29	52,44	0,24	**2,140**	0,208	1.81
11	4273	25,28	50,20	0,23	**2,228**	0,445	1.80
12	4265	25,25	56,71	0,25	1,555	0,858	1.78
13	4260	25,23	56,37	0,24	1,376	1,137	1.77
14	4273	25,21	72,90	0,24	1,752	1,464	1.76
15	4269	25,19	72,08	0,26	1,623	**1,842**	1.75
16	4268	25,19	69,74	0,25	1,683	**1,917**	1.75
17	4263	25,21	70,77	0,25	1,423	1,594	1.74
18	4262	25,20	68,80	0,25	1,445	**1,867**	1.73
19	4257	25,19	69,92	0,25	1,178	**2,105**	1.73
20	4252	25,20	71,88	0,24	0,864	**2,039**	1.73
21	4255	25,22	70,86	0,26	1,046	1,528	1.72
22	4253	25,22	69,48	0,26	0,998	1,653	1.72
23	4255	25,24	68,31	0,26	1,149	1,364	1.71
24	4263	25,24	78,11	0,26	1,535	1,350	1.71
25	4265	25,24	76,87	0,25	1,693	1,378	1.71

Appendix H – Task Assignment Probabilities

The task assignment method ‘Workload Balancing’, discussed in the fourth chapter, calculates task assignment probabilities. The method uses task arrival rate λ, T as set of tasks, A as set of agents, $R \subseteq T \times A$ as set of responsibilities on an agent for a task, $\mu_{t,a}$ as the average processing rate of agent a on task t, where (t, a) ∈ R

For each agent, the workload is computed with:

$$ld_a = \sum_{for\ t(t,a) \in R} \frac{\lambda_t \cdot p_{t,a}}{\mu_{t,a}}$$

where $p_{t,a}$ is the probability task t gets assigned to agent a. These probabilities, which during the course of the task assignment period determine which agents get which task, are calculated via the following Linear Program:

$$\min \max\{ld_a\}$$
$$s.t.\ 0 < p < 1\ (t,a) \in R$$
$$\sum_{a\ \in A, d_{t,a}} p_{t,a} = t \quad t\ \in T$$

When applied to the data from the data study, the following task assignment probabilities are found:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**		**Sum**
1	0,119	0,127	0,124	0,126	0,128	0,126	0,124	0,125	1,0	1
2	0,126	0,125	0,124	0,125	0,126	0,125	0,124	0,125	1,0	1
3	0,118	0,128	0,120	0,129	0,130	0,125	0,125	0,124	1,0	1
4	0,129	0,126	0,119	0,127	0,128	0,124	0,124	0,124	1,0	1
5	0,126	0,125	0,124	0,125	0,126	0,125	0,125	0,125	1,0	1
6	0,115	0,132	0,109	0,134	0,137	0,125	0,121	0,126	1,0	1
7	0,112	0,131	0,115	0,134	0,137	0,125	0,122	0,124	1,0	1
8	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	1,0	1
9	0,123	0,126	0,123	0,126	0,127	0,126	0,124	0,125	1,0	1
10	0,126	0,125	0,124	0,125	0,126	0,125	0,125	0,125	1,0	1
11	0,128	0,125	0,124	0,124	0,125	0,125	0,124	0,125	1,0	1

Appendix I – Outcomes of Simulations

This appendix shows the results of the different simulations. First, the table with the simulation setup is shown. Then, the results from the simulation are given.

Table 12
Test setup for Simulation 1 'PBNL'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

	FIFO	EDF	WldB	WlsB	SPT
			Service Level		
mean	0,948	0,940	0,914	0,931	0,944
0.5 s	0,002	0,0025	0,0065	0,0035	0,0025
			Processing Time		
mean	25,99	26,02	26,12	26,03	25,95
0.5 s	0,284	0,279	0,222	0,358	0,299
			Waiting Time		
mean	36,1	49,2	251	130,3	21,7
0.5 s	1,303	1,3185	36,8195	16,3825	0,6775
			Lead Time		
mean	460,2	471,4	658,4	533,5	439,2
0.5 s	7,1	6,9	26	6,9	3,8
			Avg Agent Occupancy		
mean	0,734	0,738	0,897	0,896	0,744
0.5 s	0,007	0,004	0,012	0,009	0,005
			Processing & Waiting		
mean	62,14	75,21	277,13	156,29	47,72
0.5 s	1,405	1,2795	36,957	16,4415	0,6635

Table 13
Test setup for Simulation 2 'PBNL 20% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.2
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

	FIFO	EDF	WldB	WlsB	SPT
			Service Level		
mean	0,942	0,943			0,944
0.5 s	0,0019	0,0031			0,0025
			Processing Time		
mean	25,99	26,21			25,98
0.5 s	0,3395	0,179			0,299
			Waiting Time		
mean	51,94	63,87			52,97
0.5 s	3,216	4,214			3,012
			Lead Time		
mean	430,48	426,34			430,4
0.5 s	5,297	7,558			8,399
			Avg Agent Occupancy		
mean	0,843	0,85	0,998	0,998	0,846
0.5 s	0,01	0,0087	0,005	0,005	0,008
			Combined Processing Waiting		
mean	77,936	90,08			78,95
0.5 s	3,49	4,32			3,82

Table 14
Test setup for Simulation 3 'PBNL 40% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.4
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability, scheduled unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

	FIFO	EDF	SPT
			Service Level
mean	0,924	0,922	0,9182
0.5 s	0,0037	0,006	0,005
			Processing Time
mean	25,99	26,02	25,85
0.5 s	0,135	0,187	0,24
			Waiting Time
mean	172	228	159
0.5 s	40,224	66,26	32,91
			Lead Time
mean	609	720	647
0.5 s	57	117	52
			Avg Agent Occupancy
mean	0,944	0,943	0,942
0.5 s	0,0041	0,005	0,005
			Combined Processing Waiting
mean	197	501	185
0.5 s	40,2	66,36	32

Table 15

Test setup for Simulation 4 'PBNL Continuous available, 20% Increased'

Arrival rates	PBNL arrival rates multiplied with 1.2
Processing rates	Based on data analysis PBNL
Agent behaviour	100% availability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

	FIFO	EDF	WldB	WlsB	SPT
			Service Level		
mean	0,943	0,943	0,937	0,938	0,94
0.5 s	0,0015	0,002	0,002	0,002	0,0012
			Processing Time		
mean	25,91	25,92	25,92	25,88	25,88
0.5 s	0,17	0,16	0,14	0,2	0,299
			Waiting Time		
mean	10,52	17,59	78,65	57,19	13,5
0.5 s	0,23	0,34	6,38	4,89	0,6775
			Lead Time		
mean	430,44	431,19	446,9	449,5	434,7
0.5 s	4,25	6,05	4,5	6,24	5,5
			Avg Agent Occupancy		
mean	0,677	0,683	0,678	0,68	0,683
0.5 s	0,004	0,007	0,012	0,007	0,005
			Time in System		
mean	36,4	43,5	104,57	83,06	39,5
0.5 s	0,398	0,48	6,45	5,02	0,79

Table 16

Test setup for Simulation 5 'PBNL with Delayed Rearrivals'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability
SLA	4 days for all tasks
Rearrival time	Varying between 1 and 3.5
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

Mean Service Level

# Days	FIFO	EDF	SPT	WldB	WlsB
1	0,996	0,989	0,988	0,974	0,98
1,5	0,985	0,967	0,968	0,937	0,955
2	0,97	0,937	0,944	0,914	0,931
2,5	0,962	0,921	0,925	0,90	0,91
3	0,952	0,898	0,902	0,87	0,88
3,5	0,94	0,88	0,893	0,85	0,86
	-5,62%	-11,02%	-9,62%	-12,73%	-12,24%

Table 17
Test setup for Simulation 6 'PBNL with Urgent Tasks'

Arrival rates	Based on data analysis PBNL
Processing rates	Based on data analysis PBNL
Agent behaviour	80% availability
SLA	1 day for tasks 6 & 9, 4 days for other tasks
Rearrival time	2
Number of agents	8
Included methods	FIFO, EDF, WldB, WlsB, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

Mean Service Level with adjusted deadlines

	FIFO	EDF	SPT	WldB	WlsB
Regular	0,948	0,94	0,914	0,931	0,944
1 day	0,91	0,8	0,81	0,77	0,8

Missed deadlines per task type

Task	FIFO	EDF	SPT	WldB	WlsB
1	3	6	3	11	8
2	2	5	2	8	2
3	1	5	2	5	3
4	6	3	7	4	4
5	4	8	2	9	7
6	467	493	459	514	502
7	82	91	90	110	83
8	3	2	6	9	5
9	17	22	28	35	22
10	2	2	4	12	9
11	1	2	5	6	12

Table 18
Test setup for Simulation 7 'PBNL 20% Increased, 8% Random Unavailability – 8 agents'

Arrival rates	Rates from PBNL data analysis, increased with 20%
Processing rates	Based on data analysis PBNL
Agent behaviour	20% scheduled and 8% random unavailability
SLA	4 days for all tasks
Rearrival time	2 days
Number of agents	8 agents
Included methods	FIFO, EDF, SPT
Number of runs	25
Run duration & warm-up	14 weeks, 2 weeks

	FIFO	EDF	SPT
		Service Level	
mean	0,922	0,921	0,911
0.5 s	0,002	0,008	0,006
		Lead Time	
mean	620	638,6	608
0.5 s	25,5	70,9	43
		Avg Agent Occupancy	
mean	0,947	0,946	0,949
0.5 s	0,005	0,004	0,006