Long term planning of skill set assignment

A call center optimization problem

Author:
Jeroen Mens, 303049

Supervisor:
Dr. W. van den Heuvel

Co-reader:
Dr. D. Huisman

Master Thesis

24 May 2012

Abstract

Capacity planning is a hard problem these days. Employees with their capabilities must be assigned to the needed demand. In this thesis, a call center is taken as example. Currently an inefficient heuristic is running to determine the needed amount of agents to fulfill the demand. The needed demand is based on historical data and the service level the call center wants to reach. Based on this heuristic, a model is developed to get an overview of the long-term planning skill set assignment problem. According to this model several heuristics are developed. These heuristics are compared to the current heuristic. For small problems, we can also compare the solution and computation times to the optimal solution determined by the model solver. In the heuristics the employees or skills are sorted according to a rule that is different for every heuristic. The company wants the heuristic to be very fast and as good as possible.

The planning in the application is made for all 78 weeks ahead, with a rolling horizon. All weeks are independent of each other, so if there is demand left at the end of the week, this demand is not taken into account for the next week. In this thesis an extension is made to handle this left demand. We call this inventory demand. E-mail is an example of inventory demand, while we can not store calls to handle them next week.

To test the heuristics, data is simulated with different settings. Simulations with a different amount of agents, skills and weeks, both random and real case data are generated. All heuristics perform much better than the current heuristic both in quality and computation time. The best heuristic is offered to the organization to be implemented in the current application.

Table of Contents

1. Introduction

Organizations often deal with complex processes with a lot of data. These processes and data are mostly maintained in spreadsheets. Because the amount of data increases every year, organizations can not handle this amount of data. Anago offers solutions to these processes with the help of adaptive standard solutions for planning & budgeting and capacity planning.

Large organizations deal with different systems. Operational systems are used for the production processes, administrative systems, HR systems, financial systems, hour registrations and of course spreadsheet systems. Anago integrates with these systems, so the data can be used and exchanged. Workflows per scenario can also be defined. Authorizations per function and role make sure every user get access to the data he is authorized to see. The user can adapt only his own data.

Capacity planning becomes more and more important in businesses. Specialized employees are more difficult to get and processes must become more efficient. More work has to be done with less people.

Important factors on capacity planning are the management of skills, the available capacity planning, education planning and the assignment of supply and demand.

It is important to make a mid-term planning to answer questions like how much employees do we need next year. Or do I have the right amount of employees with the right skills. Do we have a problem in the period around Christmas?

Every organization is organized differently. A call center differs a lot compared to a hospital or a production environment. Anago consists of a series of templates, which can be adapted to the organization. Anago can take over the current way of planning or implement a different one.

One of the customers of Anago is an international Sales & Service Center whitin the aviation industry is active in the area of reservations, website support, loyalty and other services. As part of a large airliner, the company helps the customers and travel agents in nineteen European countries in eleven different languages. Every year, the expected budget, which consists of needed agents and skils, is determined for one and a half year ahead.

To determine this budget as good as possible, the customer makes use of a spreadsheet application. This application has grown in the recent years, because the budgets must be available in more detail. The company however was not capable to report these service levels on the correct level.

The use of spreadsheets leads to the following bottlenecks:

- Because of the size and the relationships between the different spreadsheets, the system often crashes.
- The application calculates only one service level for the complete department instead of multiple service levels for every target.
- The addition of a new budgetyear is very sensitive and takes a lot of time, because the formulas must be copied.
- The application was built on month level, while the short-term planning was calculated on week level.

Anago makes it possible to deal with these problems. The new application must be stable and fast, so heuristics have to be developed to guarantee the quality of the application.

This thesis consists of a literature overview in chapter 2. This is mostly literature about call centers in general. The specific skill set assignment problem is not reviewed in the literature yet.

In chapter 3, the specific problem is described. The company want to improve the skill set assignment is order to be more efficient and faster.

In chapter 4, 5 and 6 the current situation is described. Chapter 4 describes the current model. Chapter 5 the current heuristic and chapter 6 gives a data overview of the current situation.

In chapter 7 the current situation is reviewed and the improvements are stated.

Seven new heuristics are developed in chapter 8. First two general frameworks are explained and after that, the differences for each heuristic are described.

In chapter 9 and 10, the settings for the simulations are initialized and the results of the heuristics are analyzed.

The inventory model as extension on the current heuristic is introduced in chapter 11. Chapter 12 gives the results of the heuristics on the inventory model.

Finally the conclusion and some advices for further research are listed.

2. Literature overview

Call centers in general are reviewed a lot in the literature. Most papers are about a specific part of the total problem. The specific skill set assignment problem, which is just a small part of the call center problem isn't reviewed in the literature at all.

There are also papers which give a quick overview of the total call center problem (Gans, Koole, & Mandelbaum, 2003) and (Tien & Kamiyama, 1982). They reviewed several employee scheduling algorithms from a common framework. This framework consists of five stages, namely the determination of manpower requirements, total manpower requirement, recreation blocks, recreation/work schedule and shift schedule. For every stage they have one or more general models and algorithms. A cutting plane method for minimizing staffing costs satisfying an acceptable service level is one example. (Atlason, Epelman, & Henderson, 2004)

In the literature about call centers, most papers focus on the shift scheduling problem. In the shift scheduling problem the scheduler must determine which employees receive which shifts according to some constraints. A shift has a duration time, a start time and breaks. For example, during an evening shift, there must be a dining break of at least one hour. Some of the constraints for the shift scheduling problem are the available hours of the employee, minimum break duration in a shift and minimum time between two shifts of one employee. In this thesis, the focus is more on the step before the shift scheduling problem, namely the determination of the amount of employees needed to reach some service level every week, while taking into account the holidays and the different skills of the employees. This thesis will review the papers on shift scheduling because there may be some interesting heuristics in these papers we can use to find a heuristic for the skill set assignment problem.

Most linear programming models for shift scheduling are based on the set covering approach. (Thompson, 1990) (Caprara, Monaci, & Toth, 2003) In the last paper, they also describe some algorithms to solve this problem faster.

While the papers just mentioned are about single-skilled employees, there are also papers which taking multi-skilled employees into account, like the problem in this thesis. When dealing with multi-skilled employees, not only the staffing becomes more complex but also there must be a routing policy of the calls, with the goal to minimize the total staff. (Wallace & Whitt, 2005) A routing policy is the assignment of calls to the employees that can handle the specific type of call. For example, when a customer calls and only speaks Spanish, the call must be assigned to an employee that can speak Spanish too. But is it better to assign this call to the employee that can speak Spanish and English, or assign the call to the employee that can speak Spanish, English and Dutch? The scheduling staff problem can consists of different target functions. One example is the minimization of the total cost for assigning staff. Another goal is to maximize the surplus of staff, so they reach a service level as high as possible. The third goal is to reduce the variation of staff surplus over different scheduling periods. (Cai & Li, 2000) When dealing with shortage on one of the skills, it is possible to hire new employees, but it is also possible to train the current employees. To solve this problem optimally, you can use an integer linear programming model. (Brusco & Johns, 1998) The objective of the model is to minimize workforce staffing costs. Some papers studied the iterative cutting-plane algorithm on the single skilled employee problem (Atlason, Epelman,

& Henderson, 2004), and some papers do the same, but with multi-skilled employees. (Cezik & L'Ecuyer, 2008) Staffing decisions in order to minimize workforce costs can be solved with linear programming based heuristics and genetic algorithms. (Fowler, Wirojanagud, & Gel, 2008) Some papers present an algorithm for the employee assignment problem according to techniques from game theory. (Lagesse, 2006) Finally, the effect of cross-utilization on a workforce staff size is investigated. (Brusco, Johns, & Reed, 1998)

Furthermore, some models and heuristics are developed in the past to solve the staffing problem for large problems very fast. (Campbell & Diaby, 2002), (Bhulai, Koole, & Pot, 2008), (Wittrock, 1992), (Shen, Tzeng, & Liu, 2003), (Campbell, 1999)

There are a lot of different approaches to solve the different problems of the call center staffing problem. Some papers solve the whole model with only a single heuristic, while others divide the problem into different sub problems and try to solve these problems optimally. Because of the NP-hardness of some of the different sub problems, these can not be solved optimally. So for these problems heuristics have to be developed to reach optimality as close as possible.

Finally, a dissertation is written with the focus on employee skills. (Eitzen, 2002) The employee rosters are generated by taking into account fluctuating demand for employees, employee skills, working conditions, training and employee preferences. An integer programming solution is proposed to minimize the amount of under and overstaffing.

3. Problem description

The current application of Anago consists of a volume forecast based on linear regression, an availability planning and a long-term planning. We focus on the last part, namely the skill set assignment problem in the long-term planning engine.

The data that comes out of the volume forecast mainly consists of call data, e-mails, queues and agent information. ErlangX will compute the staffing level per skill set per period based on the offered calls, average handle time, service level, acceptable waiting time, variance, patience and the amount of lines. The staffing level will be used to compute the needed staffing level.

The application will partly assign agents to skill sets using an iterative mechanism using the following principles:

- The assignment of the agent will be based on the unfulfilled needed staffing level such that the assignment will be highest on the low volume skill set (since those will typically be the ones with critical availability)
- There could be entered a minimum FTE available to handle a specific skill set. If the result of the assignment (later on) will be impossible to fulfil this requirement, the shortage on the skill set will be equal to the shortage in # FTE. Note that this could well be happening while there is in practice a surplus on this skill set
- If a specific skill set has a very high priority the user can enter a minimum % which should be assigned if the agent is able to work on the skill set.
- An agent will only have a surplus if all his skill sets don't need any agents any longer. The surplus will spread over all his skill sets using the original proportion of the staffing levels.

Also the use of the application has become more efficient compared to the earlier solution.

- The budgeting has a rolling horizon of 78 weeks ahead instead of an extra budget year on a fixed moment in the year.
- The application is connected to a data warehouse, so manual copy of data is limited to the minimum.
- The long-term planning can be compared to the short-term planning, because they are both calculated on day and week level.
- Because the users are trained to use Anago, they can extend the application themselves with for example financial applications.

The problem has a few constraints:

- The amount of hours that are assigned to all skills must be smaller than or equal to the available hours per agent per week, because an agent cannot work more than his availability.
- It is only possible to assign hours to skills that an agent is capable of.
- The total availability of agents that are assigned to a skill must be larger than or equal to some amount of FTE. FTE is fulltime-equivalent, so if there is a demand of 40 hours and a minimum of 2 FTE for skill A and agents 1 and 2 are capable of this skill and work fulltime, we may not assign

all 40 hours to only 1 agent, but we need to assign for example 20 hours each to skill A. This is done to make sure the work is not done by a single agent, so if this agent is not present on a day, the skill is still executed.

- If an agent is capable of a skill, there must be a minimum percentage of his available hours assigned to this skill, to make sure an agent with multiple skills is not assigned to only one skill. The totals of these percentages cannot exceed one, because if it does, there is a possibility that an agent works more than his availability.
- If all demand is fulfilled, but there are still agents with available hours, we call this surplus. These surplus hours still have to be assigned to skills, but we can only assign these hours to the skills that can have surplus.

The engine gives the following output:

- The amount of hours that are assigned to an agent per skill per week, so it is easy to check for the agents at which skills they have to work in which weeks.
- The amount of surplus hours that are assigned to an agent per skill per week.
- The amount of FTE per skill per week.
- The amount of shortages per skill per week.

The company wants to add some constraints so the planning becomes more workable and better for the employees. By adding constraints the computation time of the algorithm probably becomes much larger. This is why new algorithms have to be developed. The goal is to develop a relatively good algorithm which performs in a relatively small amount of time.

So the research question is: Can we develop a heuristic for the long-term planning of skill set assignment which is better both in quality and computation time with the same constraints, inputs and outputs and maybe extend the problem?

4. Current model

The engine that is currently running in the application is a heuristic, which is explained later on in this thesis. The heuristic is programmed in the Anago software, so the first step is to understand the programming code step by step. According to the current problem, a mixed integer programming problem is developed to solve the problem for small sets exactly and test the robustness of the heuristics for larger problems further on in this paper. The model and heuristics are made as general as possible, so it can be used for other problems as well.

The skill set assignment problem consists of the following sets, parameters, variables and constraints.

Set of agents (index i)

The set of agents are all the employees in the database.

Set of skills (index j)

The set of skills are all skills that can be executed by the agents.

Set of time (index n)

The amount of time the forecast runs. The unit of time in this paper is one week, but it can also be used for days or months.

Fte (parameter Fte)

The amount of hours in a full time working period. If we take one week as time unit, the Fte is often equal to 40 hours.

Demand (parameter $c_{j,n}$)

The expected amount of hours needed per skill per time unit to fulfill the service level calculated in an earlier stage of the call center staffing problem.

Capability (parameter $a_{i,j,n}$)

The capability is one if agent i can execute skill j in time period n and zero otherwise. The capabilities can differ over the time periods because it is possible to train agents so they can execute other skills in time period n+1.

Supply (parameter $b_{i,n}$)

The supply is the amount of hours an agent is available per time unit. If the time is measured in weeks, the supply per agent can not be larger than the Fte, because this is the amount of hours of a full time working week.

Minimum amount of hours per skill (parameter $minskill_{j,n}$)

Minskill is the minimum percentage of contract hours an agent works on a skill if the agent is capable of skill j in time period n. So the amount of hours assigned to skill j from agent i must be larger or equal to:

$$a_{i,j,n} \cdot minskill_{j,n} \cdot b_{i,n}$$

Minimum amount of agents per skill (parameter $minagents_{j,n}$)

Minagents is the minimum amount of Fte assigned to a skill in time period n. The amount of Fte assigned to a skill in time period n is calculated as follows:

$$\sum_i \frac{(z_{i,j,n} \cdot b_{i,n})}{Fte},$$

where z is a binary variable which is one if agent i is used on skill j in week n and zero else.

Surplus handling (parameter $disskill_j$)

Disskill is a binary variable. If disskill is zero for a certain skill, this means that no surplus can be assigned to this skill. If disskill is equal to one for some skill, it is possible to assign surplus to this skill.

Inventory handling (parameter inv_j)

The inventory handling is a binary parameter. If inv is zero for a certain skill, the skill can not handle inventory. For example phone calls, you can not answer a week later, but e-mail you can.

Rareness (parameter rareness)

In some of the heuristics we make use of the rareness. We distinguish between agent rareness and skill rareness. In general, the rareness is a benchmark to rank skills or agents. It uses the supply of the agents, the demand of the skills and the capabilities of the agents.

Skill $rareness_{j,n}$

We can use skill rareness to rank the skills. The smaller the rareness, the more urgent is it to fulfill the demand of the skill. The formula for the skill rareness is:

$$rareness_{j,n} = \frac{\sum_i supply_{i,n} \cdot capability_{i,j,n}}{demand_{j,n}} \qquad (4.1)$$

The rareness can only be calculated if the demand of the skill is larger than zero. If the demand is equal to zero we assume the rareness is zero too. If the skill rareness is smaller then 1, the demand of the skill can not be fulfilled, so we have a shortage on this skill. The rareness can be updated every iteration, because once the demand of a skill is fulfilled, the supply changes too and this has effect on the rareness of the other skills.

Agent $rareness_{i,j,n}$

The agent rareness is a measure for the rareness of the agent. The smaller the rareness, the more urgent it is to assign the hours of the agent to the left demand of the skills. The definition of the agent rareness is somewhat more complicated. It uses the formula of the skill rareness:

$$agent\ rareness_{i,j,n} = rareness_{j,n} \cdot capability_{i,j,n} \qquad (4.2)$$

Then we can take for example the average of the agent rareness of all agents with rareness larger than zero:

$$agent\ rareness_{i,n} = \frac{\sum_j agent\ rareness_{i,j,n}}{\sum_j capability_{i,j,n}} \qquad (4.3)$$

Totskills (parameter $totskills_{i,n}$)

Totskills is the amount of skills per agent per time period. This variable is used in some of the heuristics.

$$totskills_{i,n} = \sum_j capability_{i,j,n} \quad (4.4)$$

Totagents (parameter $totagents_{j,n}$)

Totagents is the amount of agents per skill per time period. This variable is also used in some of the heuristics.

$$totagents_{j,n} = \sum_i capability_{i,j,n} \quad (4.5)$$

Minassignment (parameter $minassignment_{i,j,n}$)

Minassignment is the assignment according to parameter minskill.

$$minassignment_{i,j,n} = supply_{i,n} \cdot minskill_{j,n} \cdot capability_{i,j,n} \quad (4.6)$$

Assignment (variable $x_{i,j,n}$)

Assignment is the amount of hours agent i executes skill j in time period n.

Shortage (variable $sh_{j,n}$)

Shortage is the amount of demand that is left after the assignment is done. We try to minimize the amount of shortage, so we have to hire as less employees as possible and save costs. The shortage is always nonnegative.

$$shortage_{j,n} = demand_{j,n} - \sum_i assignment_{i,j,n} \quad (4.7)$$

Surplus (variable $su_{i,j,n}$)

The surplus consists of three different groups of surplus described below.

Constraint surplus

Constraint surplus is the total amount of hours of minassignment per skill per time unit that exceeds the demand per skill per time unit due to the minimum assignment constraint.

$$constraint\ surplus_{j,n} = max\{0, \sum_i minassignment_{i,j,n} - demand_{j,n}\} \quad (4.8)$$

Additional surplus

Additional surplus is the amount of hours per agent per time period that is left of supply and can not be assigned to any capable skill, because all the demand of these skills are fulfilled. This surplus is divided over the possible capabilities according to some rules explained in the heuristics.

$$adsurplus_{i,j} = \left(b_i - \sum_j assignhours_{i,j} - \sum_j surplus_{i,j}\right) \cdot w_{i,j} \quad (4.9)$$

Lost surplus

Lost surplus is the amount of hours that is lost due to the fact that some of the skills can not handle surplus. If an agent is only capable of skills that can not handle surplus and this agent has additional surplus, the hours are lost.

UseAgentSkill (variable $z_{i,j,n}$)

The variable UseAgentSkill is one if agent i has positive amount of hours on skill j in time period n. This variable is important to calculate the amount of minagents.

Formulation

The formulation of the mixed integer linear programming problem is as follows. The weeks are not dependent of each other, because in the formulation no inventory is added. So the problem can be solved for every week separately. Beneath the formulation we explain the different constraints and the objective function.

$$
\begin{array}{llllr}
 & \min \sum_{\mathrm{n}} \sum_{j} sh_{\mathrm{j,n}} & & & \text{(F1)} \\
\text{s.t.} & sh_{j,n} & = c_{j,n} - \sum_i x_{i,j,n} & \forall j,n & \text{(F2)} \\
 & \sum_j x_{i,j,n} & \leq b_{i,n} & \forall i,n & \text{(F3)} \\
 & z_{i,j,n}/1000 & \leq a_{i,j,n} \cdot x_{i,j,n} & \forall i,j,n & \text{(F4)} \\
 & x_{i,j,n} & \leq z_{i,j,n} \cdot b_{i,n} & \forall i,j,n & \text{(F5)} \\
 & \sum_i (z_{i,j,n} \cdot b_{i,n})/Fte & \geq minagents_{j,n} & \forall j,n & \text{(F6)} \\
 & a_{i,j,n} \cdot minskill_{j,n} \cdot b_{i,n} & \leq x_{i,j,n} & \forall i,j,n & \text{(F7)} \\
 & sh_{j,n}\ ,\ x_{i,j,n} & \geq 0 & \forall i,j,n & \text{(F8)} \\
 & z_{i,j,n} & \in \{0,1\} & \forall i,j,n & \text{(F9)}
\end{array}
$$

(F1) Minimize the total shortage over all skills and all periods.

(F2) Shortage is equal to the demand minus the sum over the agents of the assigned hours.

(F3) An agent can not work more than his availability.

(F4) An agent can only be used on a skill if the agent is capable of the skill and if there are hours assigned to this skill. Because we do not want to exclude the possibility that x is smaller than 1, we set the minimum amount to 1/1000 is z is 1.

(F5) An agent can not have more than his available hours assigned if this agent is used on a skill.

(F6) At least “minagents” full time working weeks are working on a specific skill.

(F7) If an agent is capable of a skill, the amount of hours executing this skill has to be larger or equal to “minskill” times his availability.

(F8) The shortage and the assignment has to be positive.

(F9) The variable z is binary.

5. Current heuristic

The current heuristic executes sixteen steps and iterates over all the agents and weeks without an order to solve the problem. The general idea is to first fulfill the skills with small amount of demand, so if there is a shortage, it is on the skills with large demand. This is done because they believe that it is much easier to train or hire agents on the large demand flows. An agent is often capable of multiple skills. The supply hours of the agent do not have to be assigned to only one skill, but this can be multiple skills. The first steps of the heuristic consist of searching some kind of rule to assign the supply of an agent to the different skills the agent is capable of. We have to adapt this rule in order to not exceed the left demand of the skills, because otherwise we get surplus, while we can use the agents on other skills. Once we found a proper ratio, we check if we fulfill the constraint of the minimum amount of hours. Also we determine a maximum amount for every skill, so we do not fulfill the skills we do not have enough FTE for.

Once we found a ratio that matches all constraints we determine the final assignment for the current agent by multiplying the ratios by the supply of the agent. Next we subtract the assignment from the left demand for every skill and calculate the surplus of the agent, if there is any. Finally we continue with the next agent and the new left demand.

Once the supply hours of all agents are assigned to the skills, we can determine the shortage according to the left demand for all skills. This is the heuristic for a single time unit. If we want to make a forecast for more time units, we have to execute the heuristic multiple times, so the time units are independent of each other. It is possible to end up with an infeasible solution where the constraint of the minagents or the constraint of the minskill is not met. When this is the case, the company is advised to hire new agents for the specific skills until both contraints are met or to lower the amount of minagents or minskills in order to get a feasible solution.

Now we continue by taking a closer look to all steps in the algorithm.

We first initialize some parameters because they are being used recursively.

- The initial left demand is equal to the demand of a skill.
  $left_{0,j} = c_j$
- The initial amount of minagents left is equal to the amount of minagents of a skill.
  $agentsleft_{0,j} = minagents_j$

For every agent i do:

1. Calculate the heuristic factor for every skill set j for agent i. The heuristic factor is a weight based on the amount of demand that is left for a skill and the capability of an agent. The more demand is left for a skill, the less weight has the factor. This is a choice made by the company to make sure the small flows are executed and if there is shortage, this shortage is assigned to the larger flows.

$$f_{i,j} = \begin{cases} \frac{a_{i,j}}{left_{i-1,j}} & if\ left_{i-1,j} > 0 \\ 0 & otherwise \end{cases}$$

2. In step 2, the heuristic factor is adapted to the current situation in the heuristic, to make sure the assignment does not exceed the demand. So if the assignment is done according to the heuristic factor, the following holds:

   $$exceed_{i,j} = \begin{cases} 1 & if\ \frac{f_{i,j}}{\sum_j f_{i,j}} \cdot b_i > left_{i-1,j} \\ 0 & otherwise \end{cases}$$

   If there is a skill that exceeds the demand, the heuristic factor is corrected according to:

   $$correct_{i,j} = \frac{exceed_{i,j} \cdot left_{i-1,j}}{b_i}$$

3. Now we calculate the assignment percentage per skill without taking into account the minimum assignment constraint. We correct and rescale the heuristic factors according to:

   - If an agent has an excess for every capable skill in step 2, then use the values of "correct" as "assign".
   - For the assignment of the skills that exceeds the demand, the assignment part is equal to:

     $$assign_{i,j} = \frac{agentsleft_{i-1,j}}{b_i}$$

   - Otherwise, the assignment part of the supply is equal to:

     $$assign_{i,j} = \frac{\left(1 - \sum_j correct_{i,j}\right) \cdot f_{i,j}}{\sum_j \left(\left(1 - exceed_{i,j}\right) \cdot f_{i,j}\right)}$$

4. Determine the minimum assignment per skill.

   $$minass_{i,j} = minskill_j \cdot a_{i,j}$$

5. Determine the maximum assignment per skill. The maximum assignment takes into account the amount of hours left for a skill and the amount of agents still needed for a skill.

   $$maxass_{i,j} = \frac{a_{i,j} \cdot \left(left_{i-1,j} - \max\ \left\{agentsleft_{i-1,j} - \frac{b_i}{Fte}, 0\right\}\right)}{b_i}$$

6. Correct the assignment percentage for the maximum assignment with the help of indicator function $I\left(minass_{i,j} = 0\right)$

   $$value1_{i,j} =$$

   $$\begin{cases} \max\ \left\{correct_{i,j}, assign_{i,j}\right\} & if\ assign_{i,j} > 0 \\ 0 & if\ \sum_j (I\left(minass_{i,j} = 0\right) \cdot assign_{i,j}) = 0 \\ \frac{correct_{i,j}}{\sum_j (I\left(minass_{i,j}=0\right) \cdot assign_{i,j})} \cdot \left(1 - \sum_j minass_{i,j}\right) & otherwise \end{cases}$$

   $$rescale_{i,j} = \begin{cases} maxass_{i,j} & if\ value1_{i,j} > maxass_{i,j} \\ value1_{i,j} & otherwise \end{cases}$$

7. Update the assignment percentage according to the rescale.

   $$fixed_{i,j} = \begin{cases} \mathrm{a_{i,j}} \cdot \max\ \left(rescale_{i,j}, minskill_j\right) & if\ minskill_j > 0 \\ rescale_{i,j} & if\ rescale_{i,j} < assign_{i,j} \\ 0 & otherwise \end{cases}$$

8. Calculate the final assignment percentages.

$$assignperc_{i,j} = \begin{cases} fixed_{i,j} & if\ fixed_{i,j} > 0 \\ 0 & if\ \sum_j (I(fixed_{i,j} = 0) \cdot rescale_{i,j}) = 0 \\ \frac{\left(1-\left(\sum_j (I(fixed_{i,j}>0) \cdot fixed_{i,j})\right)\right) \cdot rescale_{i,j}}{\sum_j (I(fixed_{i,j},=0) \cdot rescale_{i,j})} & otherwise \end{cases}$$

9. Calculate the assigned hours according to the assigned percentages and the amount of agents needed left.
   $$assignhours_{i,j} = \min\ \left\{assignperc_{i,j} \cdot b_i, left_{i-1,j} - \max\ \left\{agentsleft_{i-1,j} - \frac{b_i}{Fte}, 0\right\}\right\}$$

Now, the assignment is done. The following steps consist of calculations to update the parameters used in the first set of steps.

10. Calculate the amount of hours still needed after the assignment of the agent.
    $left_{i,j} = left_{i-1,j} - assignhours_{i,j}$
11. Calculate the number of agents needed after assigning agent i
    $$agentsleft_{i,j} = \max\ \left\{agentsleft_{i-1,j} - I\left(assignhours_{i,j} > 0\right) \cdot \frac{b_i}{Fte}, 0\right\}$$

The final steps consist of assigning the surplus of agent i to skill j.

12. Calculate the surplus due to the “minimum agents needed” constraints.
    $surplus_{i,j} = assignperc_{i,j} \cdot b_i - assignhours_{i,j}$
13. Calculate the static weights per skill set j for agent i

    $$w_{i,j} = \frac{a_{i,j} \cdot c_j}{\sum_j a_{i,j} \cdot c_j}$$

    Static weight is a weight based on the total demand per skill set and the capability of an agent. The static weight is only used to assign the surplus to the skills.

14. Calculate the additional surplus.
    $adsurplus_{i,j} = \left(b_i - \sum_j assignhours_{i,j} - \sum_j surplus_{i,j}\right) \cdot w_{i,j}$
15. Re-assign step 14 according to a binary variable that determines if a surplus can be assigned to a skill.
    $dissurplus_{i,j} = \left(surplus_{i,j} + adsurplus_{i,j}\right) \cdot disskill_j$
    $$finalsurplus_{i,j} = \sum_j \left(surplus_{i,j} + adsurplus_{i,j}\right) \cdot \frac{dissurplus_{i,j}}{\sum_j dissurplus_{i,j}}$$
16. If i is the last agent, then determine the final shortage according to the left demand of all skills, otherwise go to the next agent and return to step 1.

Output

- Surplus per skill: $\sum_i finalsurplus_{i,j}$
- Shortage per skill: $c_j - \sum_i assignhours_{i,j}$
- Assigned hours per skill: $\sum_i assignhours_{i,j}$

If m is the last agent, the shortage per skill is also equal to $left_{m,j}$.

We will demonstrate the current heuristic according to a small example.

Initialization:

Capable	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Demand	MinSkill	Minagents	Disskill
Skill 1	1	1	0	1	1	10	0	0	1
Skill 2	1	0	1	0	1	50	0,2	0	1
Skill 3	1	1	1	0	0	100	0	2	0
Supply	40	20	30	20	40				

Table 5.1: *Initialization of example*

The rows in the result table below consist of the different steps in the heuristic and the heuristic starts with agent 1, then agent 2, etc.

	Agent 1			Agent 2			Agent 3			Agent 4			Agent 5		
	Skill 1	Skill 2	Skill 3	Skill 1	Skill 2	Skill 3	Skill 1	Skill 2	Skill 3	Skill 1	Skill 2	Skill 3	Skill 1	Skill 2	Skill 3
Determine the assignment percentage															
factor	0,1	0,02	0,01	0	0	0,01	0	0,03	0,01	0	0	0	0	0,11	0
exceed	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
correct	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0,23	0
assign	0	0,5	0,25	0	0	1	0	0,7	0,3	0	0	0	0	0,23	0
minass	0	0,2	0	0	0	0	0	0,2	0	0	0	0	0	0,2	0
maxass	0,25	1,25	2,75	0	0	4,43	0	1	2,3	0	0	0	0	0,23	0
value1	0,8	0,5	0,25	0	0	1	0	0,7	0,3	0	0	0	0	0,23	0
rescale	0,25	0,5	0,25	0	0	1	0	0,7	0,3	0	0	0	0	0,23	0
fixed	0	0,5	0	0	0	0	0	0,7	0	0	0	0	0	0,23	0
assignperc	0,25	0,5	0,25	0	0	1	0	0,7	0,3	0	0	0	0	0,23	0
assignhours	10	20	10	0	0	20	0	21	9	0	0	0	0	9	0
Update the parameters															
left	0	30	90	0	30	70	0	9	61	0	9	61	0	0	61
agentsleft	0	0	1	0	0	0,5	0	0	0	0	0	0	0	0	0
Assign the surplus															
surplus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
weight	0,06	0,31	0,63	0,09	0	0,91	0	0,33	0,67	1	0	0	0,17	0,83	0
adsurplus	0	0	0	0	0	0	0	0	0	20	0	0	5,2	25,8	0
dissurplus	0	0	0	0	0	0	0	0	0	20	0	0	5,2	25,8	0
finalsurplus	0	0	0	0	0	0	0	0	0	20	0	0	5,2	25,8	0

Table 5.2: *Heuristic values of example*

Finally skill 1 has a surplus of 25.2, skill 2 has a surplus of 25.8 and skill 3 has a shortage of 61. In the optimal solution skill 1 has a surplus of 10, skill 2 has a surplus of 4 and skill 3 has a shortage of 24.

6. Data

The current application consists of 174 agents and 26 different skill sets. The agents can have different availability and capability per week. A single forecast consists of 78 weeks with different needed hours per skill set.

The different input parameters are:

- Availability per agent per week in hours (174 agents * 78 weeks = 13572 decimals)
- Capability per agent per skill per week (174 * 78 * 26 = 352872 booleans)
- Amount of hours needed per skill per week (26 * 78 = 2028 decimals)
- Minimum percentage assignment per skill per week (26 * 78 = 2028 decimals)
- Minimum FTE assignment per skill per week (26 * 78 = 2028 decimals)
- Hours per FTE per week (1 decimal)
- Surplus allowed per skill (26 booleans)

And the output variables are:

- Assigned hours per agent per skill per week (174 * 78 * 26 = 352872 decimals)
- Surplus hours per agent per skill per week (174 * 78 * 26 = 352872 decimals)
- Total assigned hours per skill per week (78 * 26 = 2028 decimals)
- Total assigned hours per agent per week (174 * 78 = 13572 decimals)

As can be seen, this is a very large problem with 372555 parameter cells and 721344 variable cells, so this is why we are going to develop some heuristics. Also these heuristics can be applied to an even larger problem. In the current heuristic, the weeks are independent of each other, so the problem can be solved for every week separately. When we consider every week separately, the problem becomes much smaller and is solvable with a MIP solver. But in the future, we expect the amount of parameters increases, so the problem becomes much larger. Also we want to consider the weeks dependent of each other, so we can not solve the problem optimally for every week separately.

In the following figures you can find a quick data analysis on the input data of a single forecast.

In figure 6.1 the amount of agents available per week and the total available hours of all these agents are displayed. These amounts vary over the weeks, because agents are hired and fired or are on a holiday for example. For the coming weeks, the amount of agents and hours are more certain then the amounts later on. These numbers are estimated based on the past.

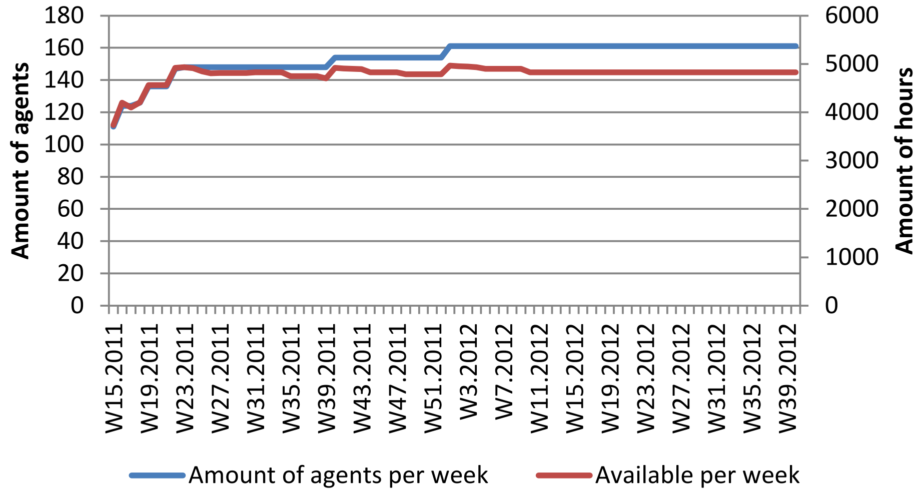

Figure 6.1: *Supply data per week*

Figure 6.2 shows the amount of hours per week needed. While this demand for all weeks is below 3000 hours, the supply is always above 3000 hours as can be seen in figure 6.1, so the total supply exceeds the total demand every week.

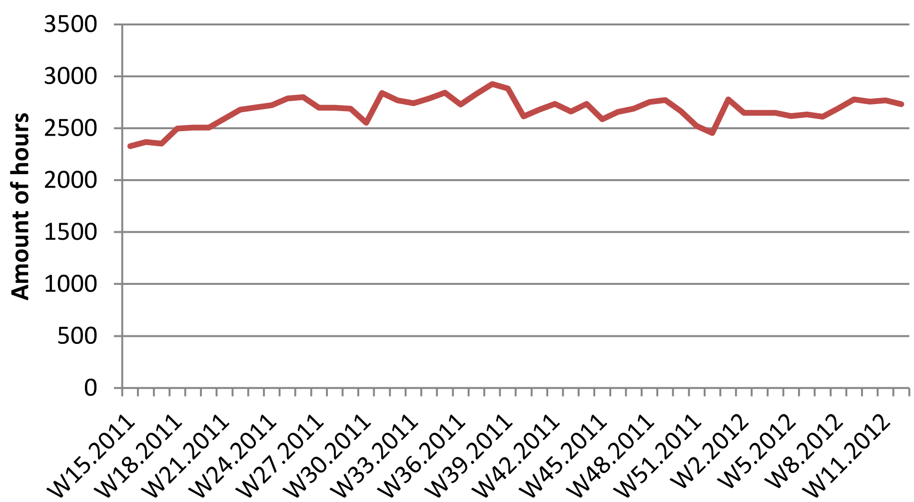

Figure 6.2: *Demand data per week*

Figure 6.3 shows a histogram with average working hours per week. There are 13 employees who aren't available at all 78 weeks, but most of the employees work between 31 and 33 hours a week.

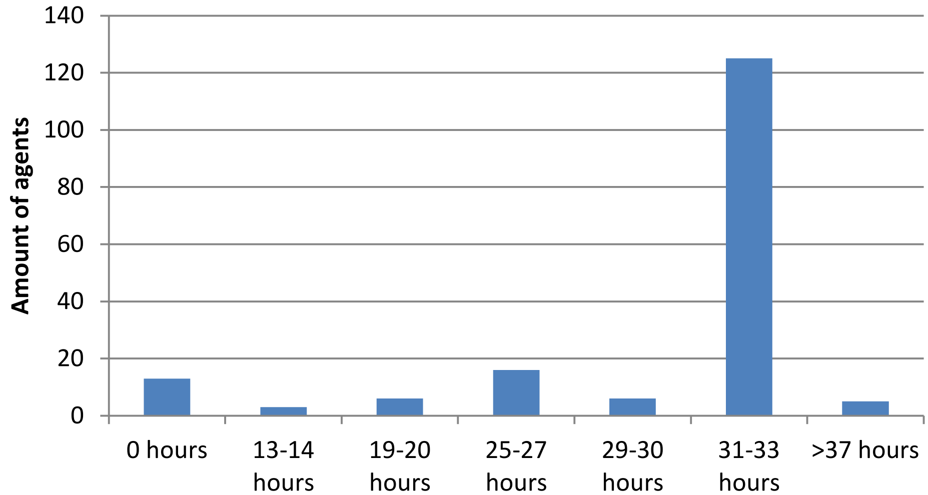

Figure 6.3: *Average amount of working hours per agent per week*

As can be seen, some agents have no available hours. Therefore we can delete them from the set of agents, so the problem becomes smaller.

The following graphs contain data about the skills. In figure 6.4 the average needed hours per week for every skill set is displayed and the amount of agents that are capable of that skill. Most agents have the English language skills and these are also the most needed.

At the moment of the data analysis the last six skills aren't used yet, so they have an average amount of zero hours needed and no agent can execute such a skill. Later on, these skills can be added to the heuristic.

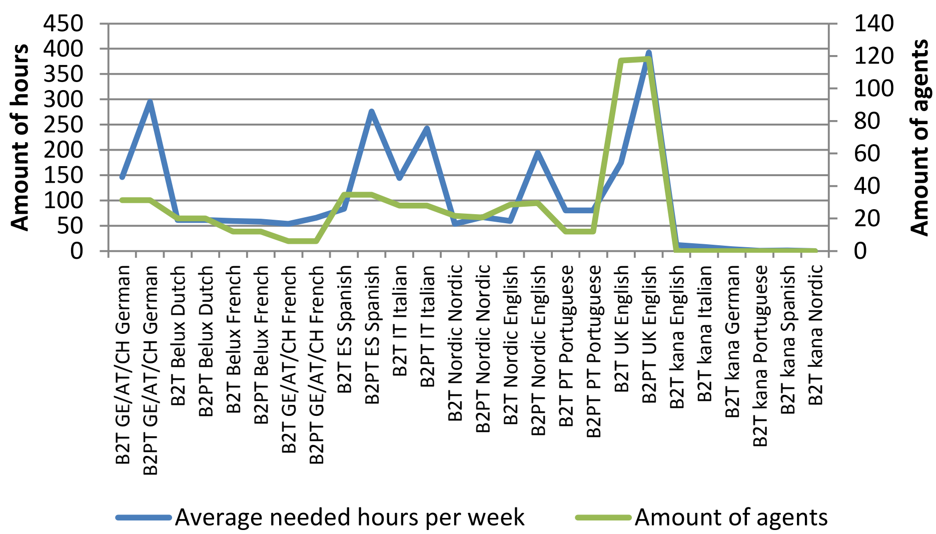

Figure 6.4: *Demand data per skill*

Figure 6.5 shows the amount of skills agents are capable of. Most agents have 4 skills, but there are also 24 agents who aren't capable of a single skill. These agents may have been fired or some employees are hired after the forecasting period. Almost all agents have skills of a multiple of 2. This is because there are 2 different skill sets per language, namely the B2T and the B2PT skill classes. Mostly agents have both the B2T and the B2PT skill of a language.

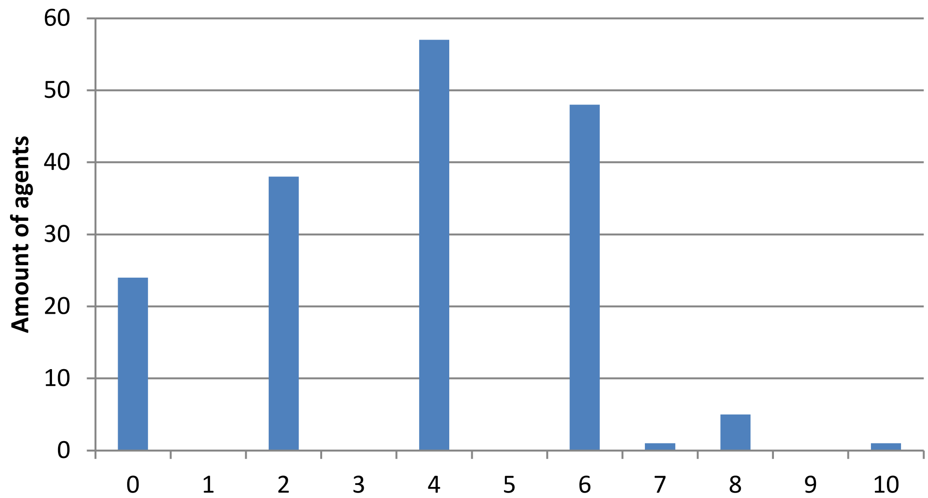

Figure 6.5: *Amount of skills per agent*

The last figure shows us the amount of other skills on average an agent of a specific skill set has. For example if an agent is capable of the first skill, he has on average 3 other skills he is capable of. This is important to determine the "rareness" of the skills.

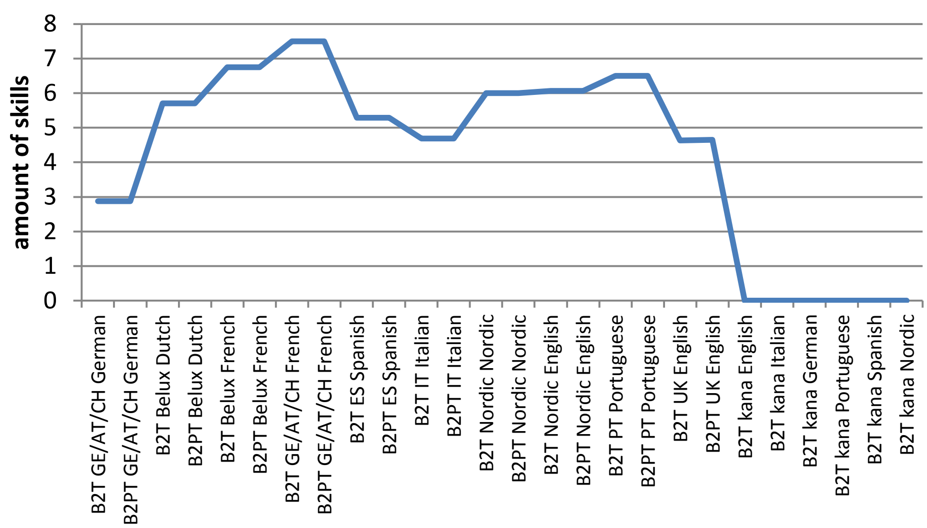

Figure 6.6: *Amount of other skills*

7. Goal and extensions

The current heuristic is very dependent of the order of the agents. This is why we want to improve the heuristic and develop some new heuristics. In the current engine, this order of the agents is random, so they haven't thought about this order. The assignment of available hours to a skill set is dependent on the capability of an agent and the left demand during the heuristic. The less left demand, the more the assignment is. The assignment of surplus to a skill is dependent on the initial demand per skill set. The larger this initial demand, the more surplus is assigned to that skill.

To improve the current heuristic we must think about some kind of order of the agents, like group agents with the same skills, or order agents according to the amount of skills they have. Also we have to improve the assignment percentages to the different skill sets and the assignment percentages of the surplus to the different skill sets to obtain a higher service level for example.

We can extend the current heuristic by adding inventory. If there is a shortage on a skill set you can take the shortage to the next week.

It is also possible to develop new heuristics. For example a fast and simple heuristic like a greedy heuristic and more complex heuristics, like taking into account the "rareness" of the skill sets and find the correlation between the skill sets. Also add a second step to the heuristic to improve the solution obtained in the first step.

To compare the heuristics we will run simulations with different real life scenarios. Also it is possible to solve the mixed integer linear programming problem for a small amount of weeks, amount of employees and amount of skill sets to compare the heuristics with the optimal solution.

8. Heuristics

We developed seven heuristics to solve large problems in a fast way. Heuristics 1, 3, 4, 5 and 6 are based on a selection on the skills and heuristics 2 and 7 are based on a selection on the agents.

The skills and agents with demand and supply are selected in a smart way. This is done different in the heuristics to determine the best solution. The other steps of the skill based heuristics are the same and the steps of the agent based heuristics are the same.

Finally we determine the shortages and surplus to come up with a solution.

First the general framework of the skill based heuristics is explained. Further on this chapter, the steps that are different are explained per heuristic.

Step 1:
First the minimum assignment is done. So we make sure we meet the constraints of the minimum percentage of the contract hours to a skill and the minimum amount of FTE to a skill. The first constraint is obvious. We just assign the minimum amount of hours of all agents to all skills, lower the supply of the agents and lower the demand of the skills.

$$minassignment_{i,j} = supply_i \cdot minskill_j \cdot capability_{i,j} \tag{8.1}$$

It is possible that the sum over the skills of “minassignment” is larger than the supply of a certain agent. If this is the case the combination of percentages of “minskill” is not feasible, an error appears and these percentages have to be adapted by the company. After the minassignment is done, lower the supply of the agents and lower the demand of the skills.

Step 2:
Select the skills that must have a minimum amount of agents and calculate the maximum amount of FTE for these skills according to:

$$maxFTE_j = \frac{\sum_i supply_i \cdot capability_{i,j}}{Fte} \tag{8.2}$$

The critical skills are the skills with a “minagents” almost equal to the maxFTE, so we sort these skills according to the difference between these two parameters in increasing order and go to step 3.

Step 3:
Sort the skills. This is different in each heuristic. After the selection, start with the first skill of the whole selection and go to step 4.

Step 4:
Sort the agents. This is different in each heuristic. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

Step 5:
If the amount of minagents of the current skill is larger than zero and the supply of the current agent is larger than 0, set the maxassignment equal to:

$$maxassignment_{i,j} = \frac{demand_j}{minagents_j} \cdot \frac{initialsupply_i}{Fte} \quad (8.3)$$

and determine the amount of hours of the current agent assigned to the current skill:

$$assignment_{i,j} = min\{supply_i, demand_j, maxassignment_{i,j}\} \quad (8.4)$$

and lower the minimum amount of agents:

$$minagents_j = minagents_j - \frac{initialsupply_{currentagent}}{Fte} \quad (8.5)$$

else do not take the maxassignment into account:

$$assignment_{i,j} = min\{supply_i, demand_j\} \quad (8.6)$$

Step 6:
Lower the supply of the current agent and lower the demand of the current skill according to the assignment in step 5.

If there are no more agents that can execute the current skill or if the demand of the current skill is 0 go to step 7. Else, go to the next agent and go back to step 5.

Step 7:
The demand that is left of the current skill is considered as shortage. The capabilities of the current skill are all set to 0 for all agents. If all skills are executed, go to step 8, otherwise go to the next skill and return to step 4.

Step 8:
The final assignment is the sum of the minimum assignment of step 1 and the assignment of step 5.

The supply of all agents that is left is considered as surplus. This surplus can be divided over the different skills that can handle surplus according to the size of the initial demand. If an agent is left with supply, but he has no skills that can handle surplus, the supply is considered as “lostsurplus”.

Next, the general framework of the agent based heuristics is explained. Further on this chapter, the steps that are different are explained per heuristic.

Step 1:
Do the minimum assignment according to formula 8.1, lower the supply of the agents and lower the demand of the skills.

Step 2/3:
Sort the agents. This is different in each heuristic.

Step 4:
Calculate the heuristic factors for all capable skills j. This is different in each heuristic.

Step 5:
Determine the assignment of hours of the current agents to every skill.

If the skill has a minimum amount of agents to fulfill, use formula 8.3 to calculate the maxassignment and the assignment can be calculated according to:

$$assignment_{i,j} = \min\{\text{demand}_\text{j}, \text{maxassignment}_{\text{i,j}}\} \quad (8.7)$$

else the assignment is equal to the demand of the skill.

Step 6:
If the assignment of step 5 is smaller or equal to $\frac{heuristic\ factor_j}{\sum_j heuristic\ factor_j} \cdot supply_i$ for some skill j, then the real assignment to this skill is equal to the assignment of step 5 and the heuristic factor of this skill becomes 0. Subtract the assignment from the supply and go back to step 4 if there is still a heuristic factor larger than 0.

Else go to step 7.

Step 7:
The real assignment of the skills for which the heuristic factor is larger than zero is equal to $\frac{heuristic\ factor_j}{\sum_j heuristic\ factor_j} \cdot supply_i$

We lower the minagents according to formula 8.5 and the demand of the skills by the assignment to the skills.

If not all agents are executed yet, continue with the next agent and go back to step 3.

Otherwise go to step 8.

Step 8:
The final assignment is the sum of the minimum assignment of step 1 and the real assignment of step 6 and 7.

The demand that is left is considered as shortage and the supply of all agents that is left is considered as surplus. This surplus can be divided over the different skills that can handle surplus according to the size of the initial demand. If an agent is left with supply, but he has no skills that can handle surplus, the supply is considered as “lostsurplus”.

Heuristic 1

The general idea of the first heuristic is to sort the skills according to their demand. Since we do not know if we first have to fill the skills with low demand or the skills with high demand we made two identical heuristics, but in heuristic 1a we first try to meet the demand of the skills with low demand and in heuristic 1b we first try to meet the demand of the skills with high demand. At the end we can determine which order works best.

Step 3:
Sort the skills that do not have a minimum amount of agents according to the demand in increasing (1a) or decreasing (1b) order and add this selection to the selection of step 2. Now start with the first skill of the whole selection and go to step 4.

Step 4:
Find the agents who are capable of the skill and sort them according to the total other skills the agent can execute in increasing order. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

We demonstrate this heuristic according to the same small example of the current heuristic:

First we do the minimum assignment:

	Agent 1	Agent 3	Agent 5
	Skill 2	Skill 2	Skill 2
Minassign	8	6	8
Supply	32	24	32
Demand	42	36	28

Table 8.1: *Minimum assignment of heuristic 1*

Then heuristic 1a is executed:

Sort skills	Skill 3			Skill 1				Skill 2		
Sort agents	Agent 2	Agent 3	Agent 1	Agent 2	Agent 4	Agent 1	Agent 5	Agent 1	Agent 3	Agent 5
Maxass	25	40	74,67							
Assign	20	24	32	0	10			0	0	28
Minagents	1,5	0,75	0							
Supply	0	0	0	0	10			0	0	4
Demand	80	56	24	10	0			28	28	0

Table 8.2: *Steps of heuristic 1a*

And heuristic 1b:

Sort skills	Skill 3			Skill 2			Skill 1			
Sort agents	Agent 2	Agent 3	Agent 1	Agent 3	Agent 1	Agent 5	Agent 1	Agent 2	Agent 4	Agent 5
Maxass	25	40	74,67							
Assign	20	24	32	0	0	28	0	0	10	
Minagents	1,5	0,75	0							
Supply	0	0	0	0	0	12	0	0	10	
Demand	80	56	24	28	28	0	10	10	0	

Table 8.3: *Steps of heuristic 1b*

Finally the hour assignment comes out with the shortage and surplus. This is the same for both heuristics:

Assignment	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	Shortage	Surplus
Skill 1		0		10			10,7
Skill 2	8		6		36		3,7
Skill 3	32	20	24			24	

Table 8.4: *Final assignment of hours, shortage and surplus for both heuristic 1a and 1b*

Heuristic 2

The general idea of heuristic 2 is to sort the agents according to their amount of capabilities. This is done because agents with a large amount of capabilities are more flexible and when assigning these at the end, we have a higher probability the agent can handle a skill whose demand is not zero yet. For the assignment of the hours of an agent to a skill we developed 2 methods. Heuristic 2a (formula 8.8) works the same as the current heuristic and calculates the heuristic factor according to the inverse of the left demand of a skill. Heuristic 2b (formula 8.9) calculates the heuristic factor according to the inverse of the rareness of the skill. The heuristic factor is calculated in step 4 for both methods. The other steps are the same for the two methods.

Step 2:
Sort the agents according to the amount of skills they have in increasing order and start with the agent with the least amount of skills.

Step 3:
If the current agent has no capabilities at all, go to the next agent and redo this step, else find the skills the agent is capable of with positive demand.

Step 4:
Calculate the heuristic factors for all capable skills j.

$$heuristic\ factor_j = \frac{1}{demand_j} \tag{8.8}$$

$$heuristic\ factor_j = 1 \Big/ \frac{\sum_i supply_i \cdot capability_{i,j}}{demand_j} \tag{8.9}$$

Heuristic 3

The general idea of this heuristic is to sort the skills according to the amount of agents that can execute the skill. At skills with a low amount of agents that can execute the skill, it is more difficult to meet the demand. Therefore we first assign the hours of agents to skills that have only a few agents.

Step 3:
Sort the skills that do not have a minimum amount of agents according to the amount of agents that can handle the skill and add this selection to the selection of step 2. Now start with the first skill of the whole selection and go to step 4.

Step 4:
Find the agents who are capable of the skill and sort them according to the total other skills the agent can execute in increasing order. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

Heuristic 4

The general idea of this heuristic is to sort the skills according to the rareness of the skill. The rareness is defined as the total supply to a skill divided by the demand of the skill (see formula 8.10). At skills with a low rareness, it is more difficult to meet the demand. Therefore we first assign the hours of agents to skills that have a small rareness. We first assign agents with less other skills.

Step 3:
Sort the skills that still have demand left according to the rareness.

$$rareness_j = \frac{\sum_i supply_i \cdot capability_{i,j}}{demand_j} \qquad (8.10)$$

The current skill is the skill with the smallest rareness, but larger than 0. Go to step 4.

Step 4:
Find the agents who are capable of the skill and sort them according to the total other skills the agent can execute in increasing order. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

Heuristic 5

The general idea of this heuristic is to sort the skills according to the rareness of the skill. We first assign the hours of agents to skills that have a small rareness, just like in heuristic 4, but now we assign the hours of agents in a different way. At the assignment of the hours of agents to the skills we look at the rareness of the other skills the agents are capable of.

Step 3:
Sort the skills that still have demand left according to formula 8.10. The current skill is the skill with the smallest rareness, but larger than 0. Go to step 4.

Step 4:
Find the agents who are capable of the skill and calculate the agent rareness of these agents. For every agent, the smallest agent rareness and larger than zero corresponds to the current skill, so we focus on the second smallest rareness of the agent and the amount of skills with positive demand the agent can execute. The smaller this rareness, and the larger the amount of skills the agents has, the more valuable the agent is. Therefore we take the ratio between the rareness and the amount of skills and sort them in decreasing order. If the agent is only capable of executing the current skill, the rareness is set to a very large number, so the agent is set at first. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

Heuristic 6

The general idea of this heuristic is again to sort the skills according to the rareness of the skill. This heuristic is more or less the same as heuristic 5, but in this heuristic we update the rareness after every assignment of hours of agents to skills. In heuristic 5 we only updated the rareness after the demand of a skill was fulfilled.

Step 3:
Sort the skills that still have demand left according to formula 8.10. The current skill is the skill with the smallest rareness, but larger than 0. If the rareness of all skills is zero go to step 8, otherwise go to step 4.

Step 4:
Find the agents who are capable of the skill and calculate the agent rareness of these agents. For every agent, the smallest agent rareness and larger than zero corresponds to the current skill, so we focus on the second smallest rareness of the agent and the amount of skills with positive demand the agent can execute. The smaller this rareness, and the larger the amount of skills the agents has, the more valuable the agent is. Therefore we take the ratio between the rareness and the amount of skills and sort them in decreasing order. If the agent is only capable of executing the current skill, the rareness is set to a very large number, so the agent is set at first. If there are no agents who are capable of the skill go to step 7, otherwise start with the first agent and go to step 5.

Heuristic 7

The general idea of this heuristic is again to sort the agents according to their total agent rareness. The smaller the total agent rareness means the agent is capable of rare skills or has not many skills he can execute.

Step 2:
Calculate the agent rareness per agent with positive supply per skill with positive demand and take the sum over the skills. If for all agents this number is equal to zero, go to step 8, otherwise go to step 3.

Step 3:
Sort the agents with positive total rareness in increasing order, and select the agent with the smallest total rareness.

Step 4:
Calculate the heuristic factors for all capable skills j according to formula 8.9.

$$heuristic\ factor_j = 1 \Big/ \frac{\sum_i supply_i \cdot capability_{i,j}}{demand_j} \qquad (8.9)$$

9. Data generation

To obtain some results on the performance of the heuristics we make use of simulations. To gather as much information as possible about the heuristics, the input parameters are different in the simulations. While this problem is time independent, we can solve the heuristic per week and determine the amount of optimal weeks per simulation per heuristic. It is also important to calculate not only the amount of optimal weeks, but also calculate the difference of the amount of shortage compared to the optimal solution of the mixed integer linear programming problem. When the problem becomes too large, we can compare the heuristics to each other.

In total we made use of seventeen different settings for the simulations of which six can be solved by the mixed integer linear programming problem. The first simulation has average demand and no minimum amount of agents. Simulations 2 until 4 differ in average demand size and some random skills must have a minimum amount of agents determined random. Simulation 5 and 6 consists of real data as used by the company, where simulation 6 is the real amount of demand times 1.5. This is because in the future the demand of the skills might increase. Simulations 7 until 11 consist of a lot more agents and skills and differ in average demand size. Furthermore, simulation 10 also has some skills that can not handle surplus and simulation 11 has some skills that can not handle surplus and some skills have minskills. Simulations 12 until 14 have normal amount of agents and skills, but the capabilities are dependent on the size of the demand. The last three simulations are the same, but this time with a larger amount of agents and skills. The exact settings for the simulations are listed in the table below.

Simulation	# Agents	# Skills	# Weeks	P(Capability)	Demand	Supply	P(minagents)	# minagents	P(minskill)	# minskills	P(disskill)
1	100	25	100	1/6	U(0, 250)	U(20, 40)	0	0	0	0	1
2	100	25	100	1/6	U(0, 250)	U(20, 40)	1/10	U(0, max)	0	0	1
3	100	25	100	1/6	U(0, 200)	U(20, 40)	1/10	U(0, max)	0	0	1
4	100	25	100	1/6	U(0, 300)	U(20, 40)	1/10	U(0, max)	0	0	1
5	174	26	78	real	real	real	real	real	0	0	1
6	174	26	78	real	real*1.5	real	real	real	0	0	1
7	500	100	78	1/6	U(0, 300)	U(20, 40)	1/10	U(0, max)	0	0	1
8	500	100	78	1/6	U(0, 200)	U(20, 40)	1/10	U(0, max)	0	0	1
9	500	100	78	1/6	U(0, 400)	U(20, 40)	1/10	U(0, max)	0	0	1
10	500	100	78	1/6	U(0, 300)	U(20, 40)	1/10	U(0, max)	0	0	9/10
11	500	100	78	1/6	U(0, 300)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
12	174	26	78	1/6 DD	U(0, 300)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
13	174	26	78	1/6 DD	U(0, 450)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
14	174	26	78	1/6 DD	U(0, 500)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
15	500	100	78	1/6 DD	U(0, 300)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
16	500	100	78	1/6 DD	U(0, 250)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
17	500	100	78	1/6 DD	U(0, 350)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10

Table 9.1: *Simulation settings*

Simulations 1-4 are small examples with different demand sizes which can be solved by the mathematical program too.

The probability that an agent is capable of a specific skill is listed in the column “P(Capability)”. For the last 6 simulations, the amount of agents that are capable of a skill is dependent of the demand of the skill. The larger the demand, the larger the probability an agent is capable of the skill. The demand, supply, amount of minagents and the amount of minskills are generated according to the uniform distribution.

10. Computational results

With the settings from last section we can test the developed heuristics and compare them to the (random) current heuristic with no order of agents at all and the optimal solution. Not only the optimality is important, but also the time to solve the problem is important. If there exists an optimal solution that takes an hour to solve and there exists a worse solution that takes only a few seconds, a tradeoff has to be made.

Simulation		Heuristic 1a	Heuristic 1b	Heuristic 2a	Heuristic 2b	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Heuristic 7	Current	MIP
1	Average shortage	249,64	251,62	322,95	248,47	253,74	244,58	243,85	244,10	266,42	471,97	243,76
	average different	5,88	7,86	79,18	4,71	9,98	0,82	0,09	0,33	22,66	228,21	0,00
	# weeks optimal	78	73	11	73	70	94	95	73	28	0	100
	av. diff. nonopt.	26,71	29,10	88,97	17,44	33,26	13,70	1,73	1,24	31,47	228,21	0
	time(sec)	0,24	0,26	0,53	0,75	0,15	0,25	0,65	1,84	3,32	3,98	17
2	Average shortage	307,54	310,56	373,25	306,44	313,42	302,29	302,04	301,82	324,01	510,26	294,91
	average different	12,63	15,65	78,34	11,53	18,51	7,38	7,13	6,91	29,10	215,35	0,00
	# weeks optimal	61	53	9	50	56	72	71	51	32	1	100
	av. diff. nonopt.	32,40	33,29	86,09	23,05	42,06	26,35	24,57	14,10	42,80	217,53	0
	time(sec)	0,23	0,28	0,53	0,8	0,16	0,25	0,57	1,65	3,36	3,78	11,82
3	Average shortage	8,26	9,27	36,65	6,93	8,68	5,14	5,14	5,15	12,65	116,10	4,70
	average different	3,56	4,57	31,95	2,23	3,98	0,44	0,44	0,45	7,95	111,40	0,00
	# weeks optimal	88	84	49	78	88	96	97	88	71	12	100
	av. diff. nonopt.	29,68	28,57	62,65	10,12	33,19	10,94	14,54	3,78	27,42	126,59	0
	time(sec)	0,24	0,27	0,54	0,72	0,16	0,25	0,62	1,78	3,27	3,77	11,23
4	Average shortage	798,02	797,48	848,83	797,31	802,08	795,32	795,32	795,41	807,87	979,29	786,49
	average different	11,53	10,99	62,34	10,82	15,59	8,83	8,83	8,92	21,38	192,80	0,00
	# weeks optimal	51	51	16	47	43	55	55	45	33	0	100
	av. diff. nonopt.	23,54	22,44	74,21	20,42	27,34	19,62	19,62	16,22	31,92	192,80	0
	time(sec)	0,22	0,27	0,53	0,8	0,17	0,25	0,59	1,47	3,45	3,74	12,98
5	Average shortage	15,86	15,78	15,89	15,78	15,78	15,78	15,78	15,83	18,87	40,27	15,78
	average different	0,08	0,00	0,11	0,00	0,00	0,00	0,00	0,05	3,09	24,48	0,00
	# weeks optimal	77	78	75	78	78	78	78	76	51	28	78
	av. diff. nonopt.	5,93	0,00	2,78	0,00	0,00	0,00	0,00	1,95	8,94	38,20	0
	time(sec)	0,21	0,24	0,52	0,62	0,16	0,17	0,43	1,87	2,44	6,08	9,53
6	Average shortage	69,04	135,95	125,76	70,28	46,08	50,00	49,95	46,64	111,86	143,85	45,83
	average different	23,21	90,12	79,93	24,45	0,25	4,17	4,12	0,81	66,03	98,01	0,00
	# weeks optimal	35	29	29	31	75	69	69	52	30	28	78
	av. diff. nonopt.	42,09	143,46	127,23	40,57	6,46	36,12	35,70	2,43	107,30	152,90	0
	time(sec)	0,22	0,2	0,51	0,75	0,16	0,18	0,43	1,9	3,04	6,11	9,53
7	Average shortage	399,82	400,19	438,73	401,06	399,64	399,30	399,27	399,98	399,36	536,12	
	average different	0,57	0,93	39,48	1,80	0,38	0,05	0,01	0,72	0,11	136,86	
	# weeks optimal	76	75	34	64	77	77	77	41	70	12	
	av. diff. nonopt.	22,06	24,30	69,98	10,05	29,94	3,68	0,75	1,52	1,06	161,75	
	time(sec)	3,27	2,95	3,38	9,73	2,7	4,28	13,14	57,6	141,75	169,63	
8	Average shortage	0	0	0	0	0	0	0	0	0	0	
	average different	0	0	0	0	0	0	0	0	0	0	
	# weeks optimal	78	78	78	78	78	78	78	78	78	78	
	av. diff. nonopt.	0	0	0	0	0	0	0	0	0	0	
	time(sec)	3,02	2,31	3,25	6,99	2,18	3,89	13,45	47,73	111,17	161,55	
9	Average shortage	5040,25	5040,25	5041,00	5040,25	5040,25	5040,25	5040,25	5040,64	5040,25	5092,20	
	average different	0,00	0,00	0,75	0,00	0,00	0,00	0,00	0,39	0,00	51,95	
	# weeks optimal	73	74	72	74	74	75	74	39	73	1	
	av. diff. nonopt.	0,01	0,01	9,78	0,01	0,01	0,01	0,01	0,78	0,01	52,62	
	time(sec)	2,95	2,57	3,3	8,54	2,71	3,47	9,87	53	133,37	164,73	
10	Average shortage	386,15	386,49	427,21	388,26	387,48	385,69	385,85	386,40	386,00	541,26	
	average different	0,51	0,85	41,56	2,61	1,83	0,04	0,20	0,76	0,35	155,61	
	# weeks optimal	75	74	37	61	74	76	74	37	69	12	
	av. diff. nonopt.	13,13	16,50	79,07	11,98	35,75	1,73	3,98	1,44	3,05	183,90	
	time(sec)	3	2,78	3,59	9,65	2,45	4,58	13,93	60,41	138,7	175,4	
11	Average shortage	534,78	534,91	573,07	535,68	536,44	537,63	537,31	535,89	534,97	886,70	
	average different	0,00	0,14	38,29	0,91	1,67	2,86	2,53	1,12	0,20	351,92	
	# weeks optimal	77	77	33	69	73	70	70	29	72	3	
	av. diff. nonopt.	0,01	10,55	66,37	7,85	26,04	27,88	24,69	1,78	2,57	366,00	
	time(sec)	2,85	2,59	3,73	9,05	2,43	3,96	12,55	56,93	147,19	178,48	
12												
	average different	0,62	2,63	0,51	0,10	0,31	1,62	2,13	0,31	0,34	1,06	
	# weeks optimal	76	64	72	75	77	72	68	76	72	67	
	av. diff. nonopt.	23,99	14,68	6,63	2,62	23,86	21,01	16,61	12,22	4,48	7,53	
	time(sec)	0,27	0,27	0,69	0,99	0,18	0,25	0,77	4,17	6,38	6,31	

13											
	average different	0,38	5,90	11,44	0,99	0,38	1,28	1,46	0,80	1,40	83,60
	# weeks optimal	76	61	60	70	76	72	70	47	67	4
	av. diff. nonopt.	14,74	27,09	49,58	9,64	14,74	16,63	14,20	2,02	9,95	88,12
	time(sec)	0,28	0,26	0,69	1,10	0,19	0,26	0,72	4,17	7,00	6,49
14											
	average different	1,45	1,73	3,79	1,95	1,45	1,97	1,97	1,75	1,95	57,59
	# weeks optimal	71	69	62	68	71	68	68	45	69	10
	av. diff. nonopt.	16,17	15,00	18,46	15,23	16,17	15,36	15,36	4,14	16,87	66,06
	time(sec)	0,28	0,27	0,73	1,13	0,19	0,26	0,74	4,10	7,04	6,78
15											
	average different	0,00	19,47	3,26	1,21	0,00	4,78	3,85	1,24	0,06	220,71
	# weeks optimal	75	33	67	51	75	54	56	20	68	8
	av. diff. nonopt.	0,01	33,75	23,14	3,50	0,01	15,54	13,66	1,67	0,45	245,93
	time(sec)	2,90	2,60	3,72	9,55	2,42	3,99	13,40	66,50	147,63	173,76
16											
	average different	0,00	3,25	0,00	0,04	0,00	3,90	4,08	0,03	0,00	0,03
	# weeks optimal	78	37	78	67	78	58	56	73	78	55
	av. diff. nonopt.	0,00	6,19	0,00	0,26	0,00	15,21	14,48	0,45	0,00	0,11
	time(sec)	2,55	2,18	3,08	7,46	2,17	3,45	11,80	56,61	120,50	152,38
17											
	average different	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,95	0,00	235,17
	# weeks optimal	74	75	75	74	72	77	75	18	75	0
	av. diff. nonopt.	0,01	0,01	0,01	0,01	0,01	0,01	0,01	1,23	0,01	235,17
	time(sec)	2,72	2,32	3,19	8,30	2,28	3,38	11,37	59,02	132,04	154,50
	av. diff. nonopt	250,48	404,93	764,96	172,75	288,85	223,78	199,92	66,96	288,29	2363,41

Table 10.1: *Simulation results*

All simulations consist of five characteristics. The first is the average amount of shortage per week. According to these amounts, the best solution is determined. For the other heuristics, the difference with the best solution is calculated and this is characteristic two. The third row is the amount of weeks, the solution was the best for the corresponding heuristic. The fourth characteristic is the same as the second, but this time only the average difference over the non-optimal weeks is calculated, so the optimal weeks are left outside this calculation. The last row consist of the total time over all weeks, the heuristic was running.

The bottom row is the general performance. For every heuristic, the differences are summed up, so the total difference over all simulations is calculated.

While analyzing the results, it is obvious that the current heuristic performs the worst. The total difference over all simulations is about 2363 hours, while the total difference of heuristic 6 is only 67 hours. The fastest heuristic is heuristic 3. Over all simulations, this heuristic only took 20 seconds, while the current heuristic took almost 1400 seconds. Heuristic 6, which has the least amount of shortage took 480 seconds to come to the solution. Heuristic 6 is often not the best, but it never is far from the best solution.

Now we will focus more on the real data simulations, namely simulation 5, with the real data, and simulation 6, with real data but with the demand multiplied by 1.5.

Simulation		Heuristic 1a	Heuristic 1b	Heuristic 2a	Heuristic 2b	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Heuristic 7	Current	MIP
5	Average shortage	15,86	15,78	15,89	15,78	15,78	15,78	15,78	15,83	18,87	40,27	15,78
	average different	0,08	0,00	0,11	0,00	0,00	0,00	0,00	0,05	3,09	24,48	0,00
	# weeks optimal	77	78	75	78	78	78	78	76	51	28	78
	av. diff. nonopt.	5,93	0,00	2,78	0,00	0,00	0,00	0,00	1,95	8,94	38,20	0
	time(sec)	0,21	0,24	0,52	0,62	0,16	0,17	0,43	1,87	2,44	6,08	9,53
6	Average shortage	69,04	135,95	125,76	70,28	46,08	50,00	49,95	46,64	111,86	143,85	45,83
	average different	23,21	90,12	79,93	24,45	0,25	4,17	4,12	0,81	66,03	98,01	0,00
	# weeks optimal	35	29	29	31	75	69	69	52	30	28	78
	av. diff. nonopt.	42,09	143,46	127,23	40,57	6,46	36,12	35,70	2,43	107,30	152,90	0
	time(sec)	0,22	0,2	0,51	0,75	0,16	0,18	0,43	1,9	3,04	6,11	9,53

Table 10.2: *Results on simulations 5 and 6 with real data*

In the first case, 5 heuristics finally came up with the optimal solution of 15.78 hours shortage. From these heuristics, heuristic 3 is the fastest. All simulation running times are very low, but when the problem is for example 100 times larger, it does matter if the heuristic takes 16 seconds or 611 seconds. Also, if the heuristics are used for what-if analysis, the running times must be as low as possible. The second simulation with larger demand gives a whole different result. In this case heuristic 3 is the best and the fastest. Only heuristic 6 performs better on the average difference to the optimal solution of non-optimal weeks, while heuristic 6 performs not very good on the first simulation. I did not expect this result, because heuristic 3 is not the most complex heuristic.

With this in mind, it is maybe profitable to do a research on the differences between the heuristics, dividing the heuristics in groups according to demand. Simulations 3, 5, 8, 12 and 16 are considered as low demand, while simulations 4, 9, 14 and 17 are simulations with high demand. The other simulations are in the last group with average demand.

Simulation		Heuristic 1a	Heuristic 1b	Heuristic 2a	Heuristic 2b	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Heuristic 7	Current
3	Average shortage	8,26	9,27	36,65	6,93	8,68	5,14	5,14	5,15	12,65	116,10
	average different	3,56	4,57	31,95	2,23	3,98	0,44	0,44	0,45	7,95	111,40
	# weeks optimal	88	84	49	78	88	96	97	88	71	12
	av. diff. nonopt.	29,68	28,57	62,65	10,12	33,19	10,94	14,54	3,78	27,42	126,59
	time(sec)	0,24	0,27	0,54	0,72	0,16	0,25	0,62	1,78	3,27	3,77
5	Average shortage	15,86	15,78	15,89	15,78	15,78	15,78	15,78	15,83	18,87	40,27
	average different	0,08	0,00	0,11	0,00	0,00	0,00	0,00	0,05	3,09	24,48
	# weeks optimal	77	78	75	78	78	78	78	76	51	28
	av. diff. nonopt.	5,93	0,00	2,78	0,00	0,00	0,00	0,00	1,95	8,94	38,20
	time(sec)	0,21	0,24	0,52	0,62	0,16	0,17	0,43	1,87	2,44	6,08
8	Average shortage	0	0	0	0	0	0	0	0	0	0
	average different	0	0	0	0	0	0	0	0	0	0
	# weeks optimal	78	78	78	78	78	78	78	78	78	78
	av. diff. nonopt.	0	0	0	0	0	0	0	0	0	0
	time(sec)	3,02	2,31	3,25	6,99	2,18	3,89	13,45	47,73	111,17	161,55
12											
	average different	0,62	2,63	0,51	0,10	0,31	1,62	2,13	0,31	0,34	1,06
	# weeks optimal	76	64	72	75	77	72	68	76	72	67
	av. diff. nonopt.	23,99	14,68	6,63	2,62	23,86	21,01	16,61	12,22	4,48	7,53
	time(sec)	0,27	0,27	0,69	0,99	0,18	0,25	0,77	4,17	6,38	6,31
16											
	average different	0,00	3,25	0,00	0,04	0,00	3,90	4,08	0,03	0,00	0,03
	# weeks optimal	78	37	78	67	78	58	56	73	78	55
	av. diff. nonopt.	0,00	6,19	0,00	0,26	0,00	15,21	14,48	0,45	0,00	0,11
	time(sec)	2,55	2,18	3,08	7,46	2,17	3,45	11,80	56,61	120,50	152,38
	av. diff. nonopt	59,60	49,44	72,06	13,01	57,05	47,15	45,63	18,39	40,84	172,43

Table 10.3: *Results on simulations with low demand*

In the simulations with low demand, heuristic 2b performs the best.

Simulation		Heuristic 1a	Heuristic 1b	Heuristic 2a	Heuristic 2b	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Heuristic 7	Current
4	Average shortage	798,02	797,48	848,83	797,31	802,08	795,32	795,32	795,41	807,87	979,29
	average different	11,53	10,99	62,34	10,82	15,59	8,83	8,83	8,92	21,38	192,80
	# weeks optimal	51	51	16	47	43	55	55	45	33	0
	av. diff. nonopt.	23,54	22,44	74,21	20,42	27,34	19,62	19,62	16,22	31,92	192,80
	time(sec)	0,22	0,27	0,53	0,8	0,17	0,25	0,59	1,47	3,45	3,74
9	Average shortage	5040,25	5040,25	5041,00	5040,25	5040,25	5040,25	5040,25	5040,64	5040,25	5092,20
	average different	0,00	0,00	0,75	0,00	0,00	0,00	0,00	0,39	0,00	51,95
	# weeks optimal	73	74	72	74	74	75	74	39	73	1
	av. diff. nonopt.	0,01	0,01	9,78	0,01	0,01	0,01	0,01	0,78	0,01	52,62
	time(sec)	2,95	2,57	3,3	8,54	2,71	3,47	9,87	53	133,37	164,73
14											
	average different	1,45	1,73	3,79	1,95	1,45	1,97	1,97	1,75	1,95	57,59
	# weeks optimal	71	69	62	68	71	68	68	45	69	10
	av. diff. nonopt.	16,17	15,00	18,46	15,23	16,17	15,36	15,36	4,14	16,87	66,06
	time(sec)	0,28	0,27	0,73	1,13	0,19	0,26	0,74	4,10	7,04	6,78
17											
	average different	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,95	0,00	235,17
	# weeks optimal	74	75	75	74	72	77	75	18	75	0
	av. diff. nonopt.	0,01	0,01	0,01	0,01	0,01	0,01	0,01	1,23	0,01	235,17
	time(sec)	2,72	2,32	3,19	8,30	2,28	3,38	11,37	59,02	132,04	154,50
	av. diff. nonopt.	39,73	37,46	102,46	35,67	43,54	35,00	35,00	22,37	48,80	546,64

Table 10.4: *Results on simulations with high demand*

In the simulations with high demand, heuristic 6 performs the best, but this heuristic is not very fast. Nevertheless, this heuristic is about 3 times as fast as the current heuristic.

11. Inventory model

The model with inventory can be applied to all heuristics and the mixed integer programming formulation. Some small adaptions have to be made to introduce this useful extension. The current application is made according to an inbound call center, so the skills consist of incoming e-mail and incoming calls. It is obvious that we can have inventory on e-mail, but we cannot have inventory on phone calls. This model also can be used for outbound call centers. In this case we can have inventory on both outgoing e-mails and outgoing calls.

We assume that when we have inventory at the end of week 1, these hours are executed at first in week 2 and after this execution, the demand of week 2 will be fulfilled. With this adaption of the model it is not possible to look at each week separately, but we have to consider all the weeks in the model, so the model becomes more complex.

We have to make some minor changes in the mixed integer programming formulation. The amount of shortage changes, because shortage can be stored and added to the demand of the next period.

The objective function becomes:

$$\min \sum_{\mathrm{n}} \sum_{j} \left(sh_{\mathrm{j,n}} \cdot \left(1 - \mathrm{inv_j}\right)\right) + \sum_{\mathrm{j}} \left(\mathrm{sh}_{\mathrm{j,n=last\ period}} \cdot \mathrm{inv_j}\right)$$

and the definition of the shortage becomes:

$$sh_{j,n} \quad = c_{j,n} + sh_{j,n-1} \cdot inv_j - \sum_i x_{i,j,n} \qquad\qquad \forall j, n$$

The first part of the objective function consists of the skills that cannot have inventory. Every week the shortage of these skills is equal to the left demand of the skills. The second part of the objective function consists of the skill that can have inventory. The left demand of these skills is added to the demand of next week, so only the left demand in the last week counts for the total shortage of the objective.

The definition of shortage is the demand of the current week and the inventory at the end of last week minus the amount of hours assigned in the current period to a specific skill. If the skill cannot handle inventory, the definition is just the demand minus the amount of hours assigned to the skill.

To handle the inventory as efficient as possible in the heuristics, we adapt the heuristic where we sort the skills according to the order of the skills. The skills that can handle inventory are assigned hours at last. The heuristics that are based on a selection of the agents, we adapt the heuristic according to the assignment of hours to the skills. The skills that can handle inventory are assigned less hours compared to the case when we do not consider inventory.

The changes made in the heuristics are listed below. In some heuristics there seems to be some random numbers. These numbers are determined by trial and error on some dataset. The numbers that performs best are used in the heuristics.

In heuristic 1, before we sort the skills according to their demand in increasing and decreasing order we divide the skills in two groups. The first group consists of the skills that cannot handle inventory. They are sorted in increasing (heuristic 1a) and decreasing (heuristic 1b) order. Furthermore the other skills are sorted in increasing and decreasing order.

In heuristic 2, the factor of step 4 is adapted for the skills that can handle inventory. Instead of 1 divided by the demand of the skill, the factor becomes for heuristic 2a, 1 divided by the demand + 5000. Heuristic 2b becomes 1 divided by the rareness * 8 instead of 1 divided by the rareness.

Heuristic 3 is adapted in the same way as heuristic 1. First the skills that cannot handle inventory are sorted according to the amount of agents that can execute the skill. At last, the other skills are sorted.

For heuristics 4, 5, 6 and 7 the definition of the rareness is different. For the skills that can handle inventory, the rareness is multiplied by 5 for heuristics 4 and 5. The rareness of heuristic 6 is multiplied by 10 and the rareness of heuristic 7 is multiplied by 10.000. In heuristic 7 also the heuristic factor is adapted. This factor is divided by 9 for the skills that can handle inventory.

With these adaptions to the heuristics and the same settings, the simulations are executed and discussed in the next chapter.

12. Results inventory model

Different from the last results, the results of the inventory model are time dependent. Because the shortage from last period is added to the demand of the current period, the problem becomes much more complex. The same settings are used to determine the data for the simulations. The only added parameter is a Boolean if the skill can contain inventory or not. The probability the parameter is 1 is set to 1/5.

This time we cannot calculate the amount of optimal weeks, because of the inventory, the result is only visible after the last week. So we have two characteristics to rate the heuristic; the amount of shortage at the end of the last period and the amount of time the heuristic took to solve the total problem.

Besides of the 17 simulations described in the seventh chapter, we added 6 more simulations with the settings as follows:

Simulation	# Agents	# Skills	# Weeks	P(Capability)	Demand	Supply	P(minagents)	# minagents	P(minskill)	# minskills	P(disskill)
18	1000	80	78	1/10 DD	U(0, 1000)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
19	1000	80	78	1/3 DD	U(0, 1000)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
20	1000	80	78	1/20 DD	U(0, 1000)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
21	1000	80	78	1/6 DD	U(0, 1000)	U(20, 40)	1/5	U(0, max)	1/10	U(0, 0.1)	9/10
22	50	150	78	1/6 DD	U(0, 15)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10
23	250	500	78	1/20 DD	U(0, 15)	U(20, 40)	1/10	U(0, max)	1/10	U(0, 0.1)	9/10

Table 12.1: *Settings of additional simulations*

Simulations 18 – 21 contain a lot of agents and differ in the probability an agent is capable of a skill. Simulations 22 and 23 contain a lot of skills. This is to determine if some heuristics perform better when there are a lot of agents.

The results are as follows:

Simulation		Heuristic 1a	Heuristic 1b	Heuristic 2a	Heuristic 2b	Heuristic 3	Heuristic 4	Heuristic 5	Heuristic 6	Heuristic 7	Current	MIP
1	Total shortage	17130,08	16780,31	21770,64	16863,78	16457,19	16068,01	15943,56	16110,73	16653,73	40997,5	15811,95
	Difference	1318,13	968,36	5958,69	1051,83	645,24	256,06	131,61	298,78	841,78	25185,55	
	Time(sec)	0,22	0,29	0,58	0,81	0,15	0,33	0,63	1,72	3,43	3,88	463,1
2	Total shortage	23583,78	23572,72	26567,79	23114,87	23311,43	23174,54	22985,57	23021,44	22972,67	42824,35	22372,32
	Difference	1211,46	1200,4	4195,47	742,55	939,11	802,22	613,25	649,12	600,35	20452,03	
	Time(sec)	0,23	0,3	0,54	0,8	0,16	0,25	0,56	1,38	3,37	3,64	454,2
3	Total shortage	449,37	79,85	997,72	37,51	0	0	0	35,59	244,7	9958,55	0
	Difference	449,37	79,85	997,72	37,51	0	0	0	35,59	244,7	9958,55	
	Time(sec)	0,25	0,3	0,56	0,73	0,16	0,25	0,6	1,75	3,22	3,68	247,14
4	Total shortage	79033,32	78998,32	81179,61	79053,35	79222,87	78944,74	78944,74	78950,13	79114,8	93098,15	78318,04
	Difference	715,28	680,28	2861,57	735,31	904,83	626,7	626,7	632,09	796,76	14780,11	
	Time(sec)	0,26	0,31	0,66	1,2	0,26	0,4	1,01	1,87	6,24	7,68	165,1
5	Total shortage	1236,9	1230,97	1239,31	1230,97	1230,97	1230,97	1230,97	1234,87	1472,24	3140,77	1230,97
	Difference	5,93	0	8,34	0	0	0	0	3,9	241,27	1909,8	
	Time(sec)	1,95	0,32	0,62	0,74	0,22	0,26	0,5	2,06	2,71	7,79	258,03
6	Total shortage	5385,05	10604,44	9809,53	5482	3594,45	3900,1	3896,36	3638,26	8725,42	11220,17	3575,06
	Difference	1809,99	7029,38	6234,47	1906,94	19,39	325,04	321,3	63,2	5150,36	7645,11	
	Time(sec)	0,24	0,26	0,57	0,75	0,16	0,17	0,43	1,97	3,17	6,16	140,87
7	Total shortage	11025,43	10420,21	10833,09	12582,65	8351,84	17231,38	17716,86	18060,28	11618,01	37759,24	
	Difference	2673,59	2068,37	2481,25	4230,81	0	8879,54	9365,02	9708,44	3266,17	29407,4	
	Time(sec)	3	2,74	3,73	9,84	2,6	4,45	13,48	61,63	137,3	172,83	
8	Total shortage	0	0	0	0	0	0	0	0	0	0,75	
	Difference	0	0	0	0	0	0	0	0	0	0,75	
	Time(sec)	3,05	2,66	3,47	7,86	2,52	4,43	14,11	52,97	112,55	170,73	
9	Total shortage	394779,07	394779,06	394779,06	394779,06	394779,07	394779,06	394779,06	394831,23	394779,06	398772,07	
	Difference	0,01	0	0	0	0,01	0	0	52,17	0	3993,01	
	Time(sec)	2,93	2,76	3,42	9,41	2,65	3,53	9,07	49,27	131,4	161,67	

10	Total shortage	3780,67	3734,39	3916,23	4263,36	3677,47	6835,78	6981,34	6807,29	3925,37	29872,12
	Difference	103,2	56,92	238,76	585,89	0	3158,31	3303,87	3129,82	247,9	26194,65
	Time(sec)	3,07	2,97	3,62	10,7	2,61	4,48	14,21	60,41	136,21	175,24
11	Total shortage	23598,06	21401,12	22246,72	20982,69	22063,28	22648,99	23195,18	22712,44	20726,16	61199,54
	Difference	2871,9	674,96	1520,56	256,53	1337,12	1922,83	2469,02	1986,28	0	40473,38
	Time(sec)	3,1	2,67	3,43	9,94	2,75	4,19	12,36	57,43	141,37	169,32
12	Total shortage	171,93	241,96	257,9	179,49	171,93	344,06	335,09	189,51	187,01	636,17
	Difference	0	70,03	85,97	7,56	0	172,13	163,16	17,58	15,08	464,24
	Time(sec)	0,31	0,28	0,69	0,98	0,21	0,27	0,75	4,17	6,15	6,31
13	Total shortage	57898,38	57903,99	61043,64	57936,38	57898,38	57982,55	57968,31	57957,67	57930,11	72211,09
	Difference	0	5,61	3145,26	38	0	84,17	69,93	59,29	31,73	14312,71
	Time(sec)	0,33	0,3	0,68	1,15	0,18	0,26	0,74	3,57	7,05	6,47
14	Total shortage	105570,52	105563,69	107533,58	105583,32	105566,9	105572,16	105581,54	105584,57	105567,59	115639,81
	Difference	6,83	0	1969,89	19,63	3,21	8,47	17,85	20,88	3,9	10076,12
	Time(sec)	0,32	0,3	0,67	1,14	0,18	0,26	0,73	3,13	6,87	6,35
15	Total shortage	18463,84	17312,28	17449,15	17185,82	17171,07	18406,93	18980,88	21803,42	16744,53	51947,1
	Difference	1719,31	567,75	704,62	441,29	426,54	1662,4	2236,35	5058,89	0	35202,57
	Time(sec)	2,96	2,63	3,39	9,77	2,71	4,12	12,77	66,19	139,55	166,56
16	Total shortage	80,54	325,98	80,54	81,48	80,54	497,74	446,5	83,66	82,63	107,94
	Difference	0	245,44	0	0,94	0	417,2	365,96	3,12	2,09	27,4
	Time(sec)	2,92	2,46	3,38	8,38	2,46	4,1	13,11	64,39	127,08	165,34
17	Total shortage	214762,54	214702,02	214762,22	214596,23	214627,27	215011	214993,22	214910,38	214596,23	237247,23
	Difference	166,31	105,79	165,99	0	31,04	414,77	396,99	314,15	0	22651
	Time(sec)	2,94	2,68	3,61	9,42	2,65	3,76	11,02	58,41	147,93	179,53
18	Total shortage	783373,09	783377,06	783894,19	783384,19	783373,09	783396,5	783402,56	783509,91	783382,02	798352,04
	Difference	0	3,97	521,1	11,1	0	23,41	29,47	136,82	8,93	14978,95
	Time(sec)	4,32	3,78	6,24	14,2	3,76	6,48	13,56	169,02	547,27	635,37
19	Total shortage	878043,45	879870,52	877986,16	877986,16	878043,45	882725,21	882710,33	879912,25	877986,16	923161,35
	Difference	57,29	1884,36	0	0	57,29	4739,05	4724,17	1926,09	0	45175,19
	Time(sec)	6,92	6,75	6,92	29,71	6,55	6,66	21,92	310,51	638,92	675,96
20	Total shortage	794095,15	794084,64	795520,32	794441,63	794090,59	794447,36	794439,26	794146,38	794428,84	823130,17
	Difference	10,51	0	1435,68	356,99	5,95	362,72	354,62	61,74	344,2	29045,53
	Time(sec)	3,16	2,83	5,63	9,41	2,75	5,29	8,89	100,2	538,83	620,73
21	Total shortage	798633,71	798645,03	798682,53	798608,33	798633,71	798662,08	798662,08	798844,97	798608,33	830710,31
	Difference	25,38	36,7	74,2	0	25,38	53,75	53,75	236,64	0	32101,98
	Time(sec)	4,74	4,42	6,42	17,58	4,25	5,56	13,74	146,39	589,89	639,31
22	Total shortage	432,03	965,69	468,39	431,65	429,95	434,28	438,12	433,24	432,72	782,94
	Difference	2,08	535,74	38,44	1,7	0	4,33	8,17	3,29	2,77	352,99
	Time(sec)	0,74	0,58	0,41	0,83	0,58	1,16	2,7	2,91	2,15	3,54
23	Total shortage	736,24	1042,22	746,84	736,24	736,24	739,7	740,42	738,5	736,89	813,05
	Difference	0	305,98	10,6	0	0	3,46	4,18	2,26	0,65	76,81
	Time(sec)	11,47	11,17	5,25	7,02	10,29	39,16	59,11	81,22	151,09	347,74
	Tot. diff.	13146,57	16519,89	32648,58	10424,58	4395,11	23916,56	25255,37	24400,14	11798,64	384465,83

Table 12.2: *Results on simulations with inventory*

As can be seen, heuristic 3 performs very well. Not only the amount of shortage is very low, but also the heuristic is very fast. When we add up all the best solutions of all simulations we have a total shortage a little less than 4.200.000 hours. The current, random heuristic is about 10% above this solution. The best heuristic, heuristic 3, is only 0,1% above the best solution. All heuristics are within 1% of the best solution, so all heuristics perform well.

Heuristic 3 solved all problems in 50.81 seconds, while the current heuristic took 72 minutes to solve all simulation problems. The MIP solver was only able to solve the first six problems. It took about 30 minutes to solve only these six simple problems (compared to the other simulations).

13. Conclusion

The current heuristic has no smart order of agents or skills. The hours are mostly assigned to the flows with low demand. With some easy changes in the heuristic, the problem can be solved better and faster. Also we extended the current problem by adding inventory.

In this thesis, we developed seven heuristics. The main focus in these heuristic was the order of skills or agents to assign the hours of the agent to the demand of a skill. In general we can conclude that it is very difficult to order agents in a smart way. An ordering of the skills has more effect and is in general faster, because you have to loop over the skills. Most of the times there are fewer skills than agents, so the heuristic is faster.

Two heuristics consists of an ordering of agents and five heuristics focus on the order of the skills. The heuristics are developed in such a way, that if there is a feasible solution, the heuristic finds one.

The heuristics are tested with real data and simulated data with different settings in demand, amount of agents and amount of skills. The results of the heuristics are compared to the current heuristic and the optimal solution to see how the heuristics perform. The real data consists of 174 agents, 26 skills and 78 weeks. The current heuristic is optimal in 28 of the 78 weeks and has an average exceed of shortage of 25 hours per week. The best heuristic is optimal in all weeks and takes only 0.16 seconds to solve the whole problem, while the current heuristic takes 6.08 seconds.

Now we multiply the demand of all skills with 1.5 and then solve the problem again to see if the new heuristic performs also well with larger demand. The current heuristic is again optimal in 28 of the 78 weeks, but this time with an average exceed of shortage of almost 100 hours per week. The new heuristic is optimal in 75 weeks with an average exceed of shortage of 0.25 hours per week. The amount of time to solve the problem is not changed for both heuristics compared to the case with normal demand.

Next we take a look at the problem with inventory. We assume that when we have inventory at the end of week 1, these hours are executed at first in week 2 and after the inventory hours, the demand of week 2 will be fulfilled. With this adaption of the model it is not possible to look at each week separately, but we have to consider all the weeks in the model, so the model becomes more complex.

We defined a different objective function and a different definition of the shortage. The heuristics are adapted and new simulations are defined. On average, heuristic 3 is the best heuristic. The results on this heuristic are only 0.1% above the best solution, while the current heuristic is 10% above the best solution. This may be not the optimal solution, because some of the simulations can not be solved optimally. Also the heuristic is much faster. The new developed heuristic solved all simulations in 50.81 seconds, while the current heuristic took about 72 minutes to solve all problems.

We can conclude that heuristic 3 performs the best. Heuristic 3 is based on a skill selection. The general idea of heuristic 3 was to order the skills according to the amount of agents that can execute the skill. We first assign the hours of agents to skills that have only a few agents, because for these skills it is more difficult to meet the demand.

The new heuristic is proposed to the customer of Anago and we hope it will be implemented. This heuristic mainly performs better on the real data simulations, so the customer's model will improve. Also the inventory extension of the model is proposed, so the planner can take into account that the employee can answer e-mails the next week, so the service level will increase.

14. Further research

We left a few extensions of the model for further research. For example, to improve the process of hiring and firing agents to save salary costs. In the current case, the planner can execute a what-if analysis what happens to the planning if an extra agent is added to the set of agents. This is also the case on training agents to increase the amount of skills the agent can execute.

Also we can add the priority of a skill set. If a skill set is very important a larger penalty is rewarded to a specific skill set. Furthermore we can add some flex-agents. This means for example that a flex-agent can work between twenty and forty hours a week, depending on the demand. They may be more expensive but it may be profitable at last to use them.

It may be useful to execute a sensitivity analysis. Now, the sensitivity analysis is done to see what happens when the demand differs from the demand as input. The sensitivity analysis is executed to run multiple different input scenarios and compare them to each other. In the future, a method can be developed to include the sensitivity analysis.

Another advice for further research is to focus more on the mixed integer programming problem. This problem only solves small problems and takes a lot of time. If some of the constraints can be relaxed, the solver may be able to solve larger problems in a smaller amount of time.

15. Bibliography

Atlason, J., Epelman, M. A., & Henderson, S. G. (2004). Call Center Staffing with Simulation and Cutting Plane Methods. *Annals of Operations Research, 127*, 333-358.

Atlason, J., Epelman, M. A., & Henderson, S. G. (2008, February). Optimizing Call Center Staffing using Simulation and Analytic Center Cutting Plane Methods. *Management Science, 54*(2), 295-309.

Avramidis, A. N., Chan, W., & L'Ecuyer, P. (2009). Staffing multi-skill call centers via search methods and a performance approximation. *IIE Transactions, 41*(6), 483-497.

Beasley, J., & Chu, P. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*(94), 392-404.

Bhulai, S., Koole, G., & Pot, A. (2008). Simple Methods for Shift Scheduling in Multiskill Call Centers. *Manufacturing & Service Operations Management, 10*(3), 411-420.

Brusco, M. J., & Johns, T. R. (1998). Staffing a Multiskilled Workforce with Varying Levels of Productivity: An analysis of Cross-training Policies. *Decision Sciences, 29*(2), 499-515.

Brusco, M. J., Johns, T. R., & Reed, J. H. (1998). Cross-utilization of a two-skilled workforce. *International Journal of Operations&amp Production Management, 18.6*, 555-564.

Cai, X., & Li, K. (2000). A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operations Research*(125), 359-369.

Campbell, G. M. (1999, May). Cross-Utilization of Workers Whose Capabilities Differ. *Management Science, 45*(5), 722-732.

Campbell, G. M., & Diaby, M. (2002). Development and evaluation of an assignment heuristic for allocating cross-trained workers. *European Journal of Operations Research, 138*, 9-20.

Caprara, A., Monaci, M., & Toth, P. (2003). Models and algorithms for a staff scheduling problem. *Math. Program.*(B 98), 445-476.

Cezik, M. T., & L'Ecuyer, P. (2008). Staffing Multiskill Call Centers via Linear Programming and Simulation. *Management Science*, 310-323.

Eitzen, G. E. (2002). *Integer programming methods for solving multi-skilled workforce optimization problems.* University of South Australia, School of Mathematics. Adelaide: School of Mathematics.

Fowler, J. W., Wirojanagud, P., & Gel, E. S. (2008). Heuristics for workforce planning with worker differences. *European Journal of Operations Research, 190*, 724-740.

Gans, N., Koole, G., & Mandelbaum, A. (2003, March). Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing and Service Operations Management, 5*, 79-141.

Lagesse, B. (2006). A Game-Theoretical Model for Task Assignment in Project Management. *IEEE International Conference* (pp. 678-680). Singapore: Management of Innovation and Technology.

Shen, M., Tzeng, G.-H., & Liu, D.-R. (2003). Multi-Criteria Task Assignment in Workflow Management Systems. *36th Annual Hawaii International Conference on System Sciences. 7*, p. 202b. Big Island: hiscc.

Thompson, G. M. (1990, August). Shift scheduling in services when employees have limited availability: An L.P. Approach. *Journal of operations management, 9*(3), 352-370.

Tien, J. M., & Kamiyama, A. (1982, July). On manpower scheduling algorithms. *SIAM review, 24*(3), 275-287.

Wallace, R. B., & Whitt, W. (2005). A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management, 7*, 276-294.

Wittrock, R. J. (1992, September). Operator Assignment and the Parametric Preflow Algorithm. *Management Science, 38*(9), 1354-1359.

www.anago.nl.