Workforce Planning

From sales to schedules

Caroline Jagtenberg · Gerhard Post

Abstract We present a comprehensive workforce planning approach from both a functional and technical perspective. Our approach forecasts the demand for staff in the future and schedules the employees accordingly. We give a high level overview of the complete planning process, which is decomposed in several steps. Two crucial optimization steps are discussed in more detail: first, we analyze the set of employees and the flexibility in their contract hours. Their individual weekly workload is optimized to match the varying demand. This step can be viewed as capacity planning, and is done using linear programming. The second step we discuss in detail is how we design and assign shifts to staff according to elaborate shift structure rules. To this end we use variable neighborhood search. Our entire planning is done on a detailed level of 15-minute intervals. The method was commercially implemented and is currently used in practice. We finish with our ideas for extending this work.

Keywords Workload modeling · Workforce capacity allocation · Shift scheduling · Shift design

1 Introduction

Many companies face a varying demand for staff over time, both throughout the year and within a day. Our approach aims to have the right people working at the right time.

C. Jagtenberg
Houtsingel 5, 2719 EA Zoetermeer
Tel.: +3188 678 3265
E-mail: caroline.jagtenberg@ortec.com

G. Post
Houtsingel 5, 2719 EA Zoetermeer
Tel.: +3188 678 3265
E-mail: gerhard.post@ortec.com

Proceedings of the 11th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016) – Udine, Italy, August 23–26, 2016

First, we briefly describe how our solution leads to insights in the future staffing needs. This is useful since exact demand patterns are in practice often not clear to the user. Second, we discuss in depth how we optimize the workforce schedule to match this demand.

As we are dealing with an elaborate problem, it makes sense to decompose the solution in multiple steps. We consider demand modeling, capacity balancing and shift design.

2 Demand modeling

This section describes how we estimate the demand for staff in the future. We do this based on historical sales figures, imagine for example the sales of a product group in a store. Generally speaking, a larger turnover corresponds to a greater number of employees that have to be present.

Forecasting sales

Preferably we use historical sales data stretching over several years. This may be rather detailed, for example a data point every 15 minutes. First, we determine a so-called *day pattern* of how sales are typically distributed throughout the day. Then, the values are aggregated to day totals, which are more stable numbers to base our forecast on. To these day totals we apply several forecast methods including Holt-Winters and Exponential Smoothing. These forecasts can be combined to fit the validation data even better.

From sales to workload

In this step we translate the forecasted sales to a workload. Typically the work will consist of several activities, each requiring their own skill set. Our planning is made in 15 minute time slots. Although in personnel scheduling a minimum planning interval of 15 minutes is considered small (see, e.g., [1]) we allow for a reasonably long planning horizon: our method is currently used in practice to model demand and plan employees for half a year.

We allow sales to be transformed to workload in two different ways. The most straightforward is a linear relation between the sales and the number of employees. Alternatively, we allow for a service model approach. This method assumes the arrivals are Poisson distributed, and we approximate the number of employees required to obtain a desired service level according to a M/G/c queue. It should be noted that this approach perhaps makes more sense if one also has historical data containing the number of *customers*, rather than just sales figures.

We distinguish two different ways of modeling the staffing requirements:

Time specific staffing Some activities need to be executed at a specific time. For example, if customers want to buy goods, there should be an employee at the till at that precise moment.

Time interval staffing Some activities need to be executed *at some point*, but not necessarily at a specific time. From a functional perspective, the desired outcome is illustrated by the following example. Imagine a shop that gets busy during specific hours of the day. In this busy period, a few extra employees are needed behind the till. However, there may be labour rules - such as a minimum shift length - that prevent us from scheduling employees only during the short busy period. Those employees will fill the rest of their shift with other, less time sensitive activities until (at least) the minimum shift length is reached. Modeling staffing requirements with time intervals thus leads to more flexibility, which in turn helps to avoid overstaffing.

3 Capacity balancing

The main idea of this step is to utilize the flexibility of the work force. That is, employees may be somewhat flexible in the amount of hours they work per week: a labour contract typically specifies the *average* number of hours that should be worked per week, as well as a minimum and maximum number of hours per week[1]. The way we model this so-called capacity balancing is similar to [4].

Since contracts often specify a workload per week, it makes sense to approach the capacity balancing on a weekly basis. The goal of this step is to determine how many hours each employee should work in each week. Note that this results in a *target*, the exact number of hours an employee is eventually scheduled may differ due to rostering constraints.

We formulate an LP, in which the main objective is to minimize the maximum imbalance between planned capacity and demand, among all weeks. This prevents ending up with a terribly understaffed week, even if all other weeks are matched perfectly. Second, we aim to minimize the sum of the weekly imbalances. Thereto, these are added to the objective value, albeit with a lower weight. There are hard constraints ensuring that both the weekly maximum as well as the total number of worked hours - over the whole planning horizon - per employee are respected.

Note that, with the model as described above, there may be many solutions which result in the same objective value. To make a good decision among these solutions, we add two objectives regarding the happiness of employees. We observed that employees in practice do not appreciate a highly fluctuating workload. Thereto, we add two incentives to stay close to the average contract hours if possible. These both have much lower weights, representing their secondary importance. First of all, we aim to minimize the *maximum* deviation

[1] These contracts are common in several fields such as health care.

of each employee from his average contract hours among all weeks. Besides the maximum deviation, we also penalize each week's deviation (again, for each employee). This is modeled as a tiered penalty: for each employee, we try to stay within 0, 1, 2, 4, 8, or 16 hours of his preferred contract hours.

Although in our planning process employees may have specific skill sets, in this step skills are disregarded. From a worst-case point of view, this could lead to decisions that are far from optimal. However, in our experience, for practical cases this does not cause problems. Furthermore, note that our decision to penalize deviations from the desired contract hours per week also helps to ensure that there is a reasonable mix of all employees planned each week.

Practicalities As this workforce planning approach is used in practice, we should allow for practical issues to be taken into account. For example, employees may be absent, or have been given permission to work a certain number of hours in a certain week in the future. Similarly, there may exist realisations of worked hours in the past, that are relevant for the future roster. All these values may be entered by the user, and the capacity balancing is solved with these additional constraints. Note that it is possible that these extra constraints make the assignment of targets to certain employees infeasible. For those employees that lead to infeasibility, we simply plan the minimum for all remaining weeks (if too much was realized) or the maximum, if too little was realized. The remaining capacity can still be optimized according to the LP described above.

4 Shift design

Given the cover constraints that result from the demand modeling in Section 2, the question remains which employee should perform which activity - and when. The weekly targets per employee determined in Section 3 serve as a guide line. This section deals with how to schedule the work force while taking into account many practical constraints and wishes. Most of them would typically originate from labour laws or practical matters.

Under- and overstaffing We define our cover requirements to be *soft* constraints. Depending on the ratio of supply and demand, under- or overstaffing may be an unavoidable result. In our tool, the user may specify the relative importance of under- versus overstaffing. Herewith, we offer a flexible method that can be used under different scenarios. This information allows the algorithm to decide between, e.g., scheduling an employee that is only useful for a small part of his shift, or not schedule him at all. Furthermore, we allow the user to specify what to do with unavoidable overcapacity: these time slots may be positioned either at the beginning or end of a shift, depending on the practical matters. For example, a planner might reason that redundant staff is most useful at the end of their shift, so they can help clean or do administration. Alternatively, in a restaurant setting, it is more useful for a chef to show up early and start with some preparations, than to stay late after all customers have left.

Shift constraints We allow for various constraints on shifts and activities within shifts. The user can specify whether one or two breaks are required, depending on the shift length. The break lengths as well as their positions in the shift are also adjustable. All break constraints are modeled as hard constraints.

Shifts may follow one of several patterns, describing constraints on the begin and end times. Furthermore, the total shift length, and the length of a certain activity may be constrained. In particular, this allows the user to limit the length of a physically demanding activity. There may also be a maximum on the number of shifts one employee gets in a week.

Skills Activities may only be executed by employees with the right skills. We modeled this as a hard constraint. Note that our method does not explicitly prevent overqualified - and therefore perhaps more expensive - employees from being planned. However, if a certain skill is scarce, our solution will try to schedule skilled people only on those skill requiring activities. The line of thought is that the work force is a given - they have to be paid regardless, and the question we answer is how to best utilize this work force. This approach makes sense in the situations that we have often come across: the demand more or less matches the supply, or else there will typically be a shortage of staff rather than a surplus.

Note that employee preferences for holidays can be taken into account, in the sense that they can receive a lower target that week (e.g., 0 hours). Furthermore, unavailability on particular days can be specified on an individual level.

Solution method The set of shifts is optimized by an extensive metaheuristic framework. This method is an effective choice to handle the 'messy' real life constraints described above. By combining an efficient variable neighborhood search with multiple re-start heuristics, we are able to deliver good results for this complex rostering problem. In order not to over complicate matters, our implementation allows for at most 1 shift per employee per day.

Scalability Our solution method can be used to schedule a reasonably sized work force in great level of detail. Currently, users tend to optimize a schedule for departments up to 150 employees, planning 2 weeks at a time. Such an optimization can realistically be performed in 15 minutes.

5 Discussion / Future work

Our workforce planning approach stands out in several ways. First of all, it is a comprehensive approach that covers many aspects from demand modeling to actual rostering. Furthermore, it constructs shifts based on the particulars of individual employees, each with their own skills, availabilities etc. This approach is uncommon in personnel rostering literature. Precisely this disregard for personal preferences and attributes constitutes a gap between most literature and practice, as pointed out in Chapter 1 of [3].

Currently, our solution is designed to assign shifts to a fixed set of employees. For planners who want to make the most of their given workforce, this makes sense. However, in some industries the work force may be expanded using casual workers. Scheduling decisions on how to best combine these casual workers with the regular staff are then of crucial importance. This concept can also be found in, e.g., [2].

One might take this reasoning even further, and use our planning tool to determine which employees to hire or fire as permanent staff. We certainly see demand for this type of decision support in practice. Thereto, it would be valuable to make a roster using a large set of potential employees, but somehow minimize the number of people that actually have shifts assigned in the final solution. This further complicates the problem, and a straightforward adaptation of our algorithm is most likely not sufficiently scalable. As [1] points out, integrating decisions on hiring/firing with the regular personnel planning is one of the major areas for future research.

References

1. den Bergh, J.V., Belin, J., Bruecker, P.D., Demeulemeester, E., Boeck, L.D.: Personnel scheduling: A literature review. European Journal of Operational Research **226**(3), 367–385 (2013)
2. Hojati, M., Patil, A.S.: An integer linear programming-based heuristic for scheduling heterogeneous, part-time service employees. European Journal of Operational Research **209**(1), 37–50 (2011)
3. van der Veen, E.: Personnel preferences in personnel planning and scheduling. Ph.D. thesis, University of Twente, Enschede (2013)
4. van der Veen, E., Hans, E.W., Veltman, B., Berrevoets, L.M., Berden, H.J.: A case study of cost-efficient staffing under annualized hours. Health Care Management Science **18**(3), 279–288 (2014)