Appendix: Simple Methods for Shift Scheduling in Multi-Skill Call Centers

Sandjai Bhulai, Ger Koole & Auke Pot

Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Supplementary Material

In Section 1 we present a model for integrating the computation of the optimal staffing levels and the optimal set of shifts. Additional examples are provided in Section 2. Finally, Section 3 treats different extensions to the model that are relevant for the single-skill as well as the multi-skill methods. This section also describes an extension in which a constraint on the average service level during the day is taken into account.

1 Single-Skill Environment

The optimal shift schedule in a single-skill environment can be determined by the basic set-covering model as introduced in Dantzig [Dan54]. It is applied to call centers in Keith [Kei79]. The objective is to schedule the shifts such that conditions are satisfied on the minimal service level per period against minimal costs. The method consists of two steps: the calculation of staffing levels and shift scheduling. The staffing levels are chosen such that the minimal service level during each period is assured. In the second step, shifts are scheduled such that the staffing levels in each period are met.

There exist different methods to execute the first step. In call centers the staffing levels are often determined as follows. The working day is split in consecutive time intervals. Then, in each period call centers assume independence between the different periods and they assume stationary behaviour. Using standard formulas, like the Erlang-C formula, the staffing level is determined for each time interval. (For literature about the Erlang formula and the determination of staffing levels, we refer to Cooper [Coo81].) Most models from the literature have the limitation that assumptions are made (e.g., exponential distributions) to derive closed-form expressions for the relation between service level measures and staffing levels. These closed-form expressions have the advantage that computation times are short, however, the assumptions are unrealistic in practice.

The impact of assuming independence between the time intervals can be reduced. Thompson [Tho93] describes a method that corrects for the multi-period impact by means of adjusting the arrival rate appropriately in each period. Green *et al.* [GKS01] propose the lag-max algorithm that shifts the staffing levels that are determined in the first phase in the process of labor allocation forward in time. This results in service levels that are satisfied for almost all periods when taking transient effects into account.

The 2-step method is not optimal for certain service-level constraints, in particular conditions on the (weighted) average service level over a day. To obtain an optimal solution, one can integrate the determination of staffing levels and shift scheduling by constructing an integer programming model. For call centers with 1 skill, an appropriate integer programming model is presented in Thompson [Tho97]. It has a larger complexity than the model of [Dan54], but is computationally still tractable. Ingolfsson & Cabral [ICW02] worked on a similar subject. They present a heuristic method that takes transient effects into account.

The integer programming model from Thompson [Tho97] is easily extendable to the multi-skill environment. This requires a modification to the integer programming formulation. We discuss this modification to the model in case of one skill. The extension to multiple skills is straightforward. To formulate the model we need the following notation:

- $a_{k,t}$ is set to 1 if shift k overlaps time interval t and 0 otherwise;
- $\gamma_{s,t}$ denotes the expected number of customers that wait less than the AWT in the queue during interval t when we schedule s agents;
- $n_{s,t}$ is set to 1 if there are s agents scheduled during interval t, and 0 otherwise;
- x_k denotes the number of agents working in shift k;
- c_k is the cost associated with an agent working in shift k;
- $s \in \{0, 1, \ldots, S\}$ is the maximum number of agents scheduled during a period.

The integer programming model to compute the optimal schedule is

$$\begin{aligned}
&\min \textstyle\sum_{k\in\mathcal{K}} c_k x_k \\
&\text{subject to} \\
&\textstyle\sum_{k\in\mathcal{K}} a_{k,t}x_k = \sum_{s\in\mathcal{S}} n_{s,t}s, && t \in \mathcal{T}, \\
&\textstyle\sum_{s\in\mathcal{S},t\in\mathcal{T}} n_{s,t}\gamma_{s,t} \geq \alpha(\sum_{t\in\mathcal{T}} \lambda_t), \\
&\textstyle\sum_{s\in\mathcal{S}} n_{s,t} = 1, && t \in \mathcal{T}, \\
&x_k \geq 0 \text{ and integer}, && k \in \mathcal{K}, \\
&n_{s,t} \in \{\, 0, 1\}, && t \in \mathcal{T}.
\end{aligned}$$

This formulation has much similarity with the one from Atlason et al. [AEH04]. With respect to multi-skill call centers, the extended version of this model enables us to compute lower bounds for the criterion function.

	agent group g											
m	01	02	03	04	05	06	07	08	09	10	11	12
1	x		x	x	x		x	x	x		x	x
2			x			x	x	x			x	x
3		x		x		x	x		x	x	x	x
4					x					x		x
5								x	x	x	x	x

Table 1: Skills and policy of the first example

	time t													
m	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	440	440	440	440	440	440	440	440	440	440	440	440	440	440
2	140	240	340	440	540	490	440	440	490	540	440	340	240	140
3	440	340	440	340	440	340	440	340	440	340	440	340	440	340
4	340	390	440	490	540	540	515	490	465	440	415	390	365	340
5	340	365	390	415	440	465	490	515	540	515	490	465	440	415

Table 2: Arrival rates per hour of example with 5 skills

2 Examples

In this section we discuss numerical experiments. This section consists of two parts. Firstly, Section 2.1 treats a relatively large example. Secondly, Section 2.2 elaborates on an extreme example.

2.1 Large Example

This example concern a call center with 5 job types. The agent groups and the agent selection policy are similar to the ones used in Cezik & L'Ecuyer [CL06], which is depicted in Table 1. In contrast, an agent serves the longest waiting customer at a service completion. The service rates are $\mu_{m,g} = 12$ for all m and g. We require the overall service level to be at least 80%.

The arrival rates per hour are presented in Table 2. We apply the algorithm from Pot, Bhulai & Koole [PBK06]. The staffing costs are $\big(K^1(s_{t,1}), K^2(s_{t,2}), \dots\big) = (0.908, 0.954, 0.908, 1, 1, 1, 1, 1, 1, 0.954, 1, 1)$. Table 3 presents the staffing vectors that resulted from the staffing algorithm.

Shifts have a length of 5 or 6 hours and require 3 different sets of skills. The costs per shift type are:

- Skills 1 and 2: a cost of 4 and 5;
- Skills 3, 4, and 5: a cost of 4.5 and 5.5;

g	time t 1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	45	50	59	71	80	81	75	71	79	72	75	62	52	48
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	84	79	98	94	110	112	119	106	119	98	110	94	99	90
11	1	2	2	0	1	0	2	0	3	5	3	1	1	3
12	21	18	22	22	20	8	9	19	9	23	9	18	18	8

Table 3: Staffing vectors of example with 5 skills

- Skills 1, 2, 3, 4, and 5: a cost of 5 and 6,

for shifts of length 5 and 6, respectively.

Using the shift scheduling algorithm we obtain a solution with an objective value of 2389.5 in a fraction of a second.

2.2 Example with high variability in group sizes

In this section we give an example, with high variability in group sizes, of a shift scheduling problem for a call center with 3 skills, $S_1 = \{1\}$, $S_2 = \{2\}$, $S_3 = \{3\}$, $S_4 = \{1, 2\}$, $S_5 = \{2, 3\}$, $S_6 = \{1, 3\}$, and $S_7 = \{1, 2, 3\}$. The day consists of 14 periods, indexed from 1 to 14. We consider a limited number of shift types: Generalists work in shifts of length 6, have a cost of 6 and start in intervals 1–9. Cross-trained agents work in shifts with: skills 1 and 3, a length of 5, a cost of 5 and start in intervals 1–10, skills 2 and 3, a length of 6, a cost of 5.5 and start in intervals 1–9, and skills 1 and 2, a length of 5, a cost of 4.5 and start in intervals 1–10. Specialists of type 3 work in shifts of length 6 that start in intervals 1–6 and have a cost of 5. Specialists of type 2 work in shifts with a length of 6 that start in intervals 7–9, having a cost of 5, and they also work in shifts with a length of 5 starting in 1–3, and a cost of 4. Specialists of type 1 start in intervals 4–10 with shifts of length 5, and cost 4.

We composed staffing vectors manually for a day consisting of 14 intervals, see Table 4. The diversity of the agent types in each vector is higher than in reality; the number of specialists is relatively low and the composition of agents with 2 skills is diverse among the different intervals. The optimal solution is displayed in Table 5. The total number of shifts is 35, and the total number of idle periods is 16. The solution requires 8 generalists, 18 agents with 2 skills, and 9 specialists, including 16 shifts of length 6 and remainder of

	time t													
g	01	02	03	04	05	06	07	08	09	10	11	12	13	14
1	1	2	3	2	2	3	2	1	1	2	2	1	1	1
2	1	2	3	2	2	3	2	1	1	2	2	1	1	1
3	1	2	2	2	2	1	2	1	1	1	0	1	1	1
4	1	2	2	3	2	3	2	3	3	1	0	1	1	1
5	1	1	2	3	2	3	2	3	3	2	2	2	1	1
6	1	1	2	2	2	2	2	3	3	2	2	2	1	1
7	1	3	2	2	2	2	4	3	3	2	2	1	2	1

Table 4: Required group sizes $s_{t,g}$ (with 3 skills)

length 5.

As a comparison, we calculate the total number of agents in each interval and consider a single-skill call center. We solve the problem using Dantzig's formulation [Dan54] by using the online tool that is available at `http://www.math.vu.nl/~sapot/software/shift-scheduling`, which includes column generation. The optimal solution also requires 35 shifts, including 10 shifts of length 6. Thus, the total idle time is 6 periods less than the number that the multi-skill solution provides. Note that 6 is a only lower bound for the total idle time, because in the multi-skill case not all shift types have a length of 5 and 6; some have a length of 5, and others a length of 6.

3 Extensions

In this section we discuss a number of extensions. These additional features extend the model to more specific scenarios, and make the algorithm more useful in practice. The first extension explains how to reserve resources for emails and faxes. The second extension relaxes the service level conditions to decrease costs without much loss of performance. The third extension discusses the inclusion of bounds on the number of occurrences of shifts. This may be desired when assigning shifts or rosters to employees, such that the availability of employees can be taken into account more easily.

We illustrate the extensions by applying them to the single-skill shift-scheduling model. However, they can also be straightforwardly applied to the multi-skill integer programming models from Section 2.2 of [BKP06].

3.1 Emails and faxes

In many call centers nowadays, emails and faxes represent a substantial part of all traffic besides telephone calls. This has a positive impact on the productivity of these so-called 'contact centers'. The explanation is as follows. Since emails and faxes have lower service-level requirements than calls, emails and faxes can wait when many telephone calls are in the system or are expected to arrive. Hence, they are only handled in periods that are

	time t													
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
1	7	7	1	1	7	1								
2		7	1	6	7	1	1							
3		7	6	7	0	7	7							
4			7	7	0	7	7	5						
5			7	0	0	0	7	7						
6							7	7	7	7	7	3		
7								7	7	7	7	7	7	
8									7	0	0	0	7	7
9	6	6	6	6	6									
10					6	6	6	6	6					
11					0	6	6	6	6					
12								6	6	6	6	6		
13										6	6	6	6	6
14	5	5	5	5	5	5								
15			5	5	5	5	5	5						
16				5	0	5	5	5	5					
17									5	5	5	5	3	3
18									5	5	5	5	5	5
19	1	1	1	1	1									
20	2	1	2	4	1									
21	4	4	4	4	4									
22		4	4	4	4	2								
23					0	4	2	2	2					
24					0	4	2	4	4					
25					0	4	4	4	4					
26							4	4	4	2	1			
27										2	2	2	2	2
28										4	2	4	4	4
29	3	3	3	3	3	3								
30		3	3	3	3	0	3							
31					0	0	3	3	3	3				
32		2	2	2	2	2								
33		2	2	2	2	2								
34						1	1	1	1	1				
35										1	1	1	1	1

Table 5: Optimal shifts

not very busy. As a result the workload is more constant over the day. This flexibility of scheduling emails and faxes improves working schedules because idleness decreases.

We define q as the desired number of hours spent on handling emails and faxes. The decision variable $\hat{x}_t$ denotes the number of agents that work on emails and faxes at time t. The integer programming model then becomes

$$\min \sum_{k\in\mathcal{K}} c_k x_k$$

subject to

$$\begin{aligned} &\sum_{k\in\mathcal{K}} a_{k,t} x_k - \hat{x}_t \geq s_t, && t \in \mathcal{T}, \\ &\sum_{t\in\mathcal{T}} \hat{x}_t = q, && \\ &x_k, \hat{x}_t \geq 0 \text{ and integer}, && k \in \mathcal{K}, t \in \mathcal{T}. \end{aligned}$$

3.2 Penalties

By using the results of Keith [Kei79], the different models in this paper can be refined. It describes a method to weaken the constraints on the minimal personnel requirements.

Let the parameter $\bar{p}_t$ denote the cost per agent exceeding s_t during time interval t, and let $\tilde{p}_t$ denote the cost per agent falling short of s_t during time interval t. Define the decision variable $\bar{x}_t$ to denote the redundant number of agents during time interval t, and $\tilde{x}_t$ to denote the shortage of agents working in period t. The integer programming model then becomes

$$\min \sum_{k\in\mathcal{K}} c_k x_k + \sum_{t\in\mathcal{T}} (\bar{p}_t \bar{x}_t + \tilde{p}_t \tilde{x}_t)$$

subject to

$$\begin{aligned} &\sum_{k\in\mathcal{K}} a_{k,t} x_k - \bar{x}_t + \tilde{x}_t = s_t, && t \in \mathcal{T}, \\ &x_k, \bar{x}_t, \tilde{x}_t \geq 0 \text{ and integer}, && k \in \mathcal{K}, t \in \mathcal{T}. \end{aligned}$$

3.3 Lower and upper bounds

To take into account the availability of agents, a lower and upper bound can be specified for each shift. This is done as follows for N constraints. Let V_i denote the set of shift indices for which constraint i is applicable, $i \in \{1, 2, \ldots, N\}$. The corresponding shifts have a lower and upper bound on the total number of occurrences. Let parameter l_i be the lower bound on the number of shifts from set V_i, and u_i the upper bound on the number of shifts from set V_i. The integer programming model is then given by

$$\min \sum_{k \in \mathcal{K}} c_k x_k$$

subject to

$$\sum_{k \in \mathcal{K}} a_{k,t} x_k \geq s_t, \qquad t \in \mathcal{T},$$
$$l_i \leq \sum_{m \in S_i} x_m \leq u_i, \qquad V_i \subset \mathcal{K}, i = 1, \ldots, N,$$

$$x_k \geq 0 \text{ and integer}, \qquad k \in \mathcal{K}.$$

3.4 Global service-level constraints

The purpose of this section is to show that the algorithm from Section 2.2 of [BKP06] can be extended to handle a constraint on the average service level during the day. To this end, we describe a local-search method for determining staffing levels such that this type of constraint is satisfied. We illustrate that the extension behaves well, by means of a numerical experiment.

First we give two observations underlying the heuristic:

- Consider the integer programming model for the multi-skill call center. According to the model, each row corresponds to both a period and an agent group, which is denoted by the right-hand side of the model. It is well-known from duality theory that a dual variable indicates the relation between the right-hand-side and the value of the objective function. Thus, agents that contribute to the staffing level of a group with a high dual value, in a certain period, have a larger contribution to the objective value.

- With regard to service levels, not all adjustments to the staffing levels are equally preferred. For example, if we decrease $s_{t,g}$ by 1 and increase $s_{t',g'}$ by 1, we prefer to choose t, g, t', and g' such that the resulting overall service level is highest.

Algorithm

By combining both observations we construct the following algorithm:

Step 1: Solve the optimization problem from Section 3.1 of [BKP06]. This gives the initial solution.

Step 2: Select the m rows with the highest dual values and the m rows with the lowest dual values. Pick one row from each selection, which denotes a movement of an agent from one group to another and choose both rows in such a way that the decrease in service level is minimal. Repeat this step as long as the changes are significant. Otherwise go to step 3.

	time t													
g	01	02	03	04	05	06	07	08	09	10	11	12	13	14
1	2	5	8	7	8	10	6	8	9	6	5	5	3	2
2	3	5	6	6	7	8	5	6	7	8	5	5	3	2
3	0	0	1	3	2	0	5	2	0	3	0	0	2	1

Table 6: Group sizes $s_{t,g}$
(with 2 skills)

Step 3: If the overall service level exceeds the minimum requirement, select the m rows with the highest dual values and decrease one of these group sizes such that the decrease in service level is minimal. If the overall service level is not sufficient, select the m rows with the lowest dual values and increase the right-hand-size coefficient of one of the rows such that the service level is maximized. Repeat this step until the average service level is just above the bound. Otherwise stop or, if significant changes are made, go to step 2.

Numerical example

We apply this algorithm to the example from Section 3 of [BKP06]. Parameter m is set to 5. The initial solution s is determined by Table 2 of [BKP06], yielding a service level of 81% and an objective value of 167.

The outcome of the algorithm is presented in Table 6. This solution was obtained after 12 iterations. The corresponding schedule of shifts contains no idle periods, its objective value is 161, and the service level is on average 80% during the day.

References

[AEH04] J. Atlason, M.A. Epelman, and S.G. Henderson, *Call center staffing with simulation and cutting plane methods*, Annals of Operations Research **127** (2004), 333–358.

[BKP06] S. Bhulai, G.M. Koole, and S.A. Pot, *Simple methods for shift scheduling in multi-skill call centers*, Submitted, 2006.

[CL06] M.T. Cezik and P. L'Ecuyer, *Staffing multi-skill call centers via linear programming and simulation*, To appear in Management Science, 2006.

[Coo81] R.B. Cooper, *Introduction to Queueing Theory*, 2nd ed., North Holland, 1981.

[Dan54] G.B. Dantzig, *A comment on Edie's 'traffic delays at toll booths'*, Operations Research **2** (1954), no. 3, 339–341.

[GKS01] L.V. Green, P.J. Kolesar, and J. Soares, *Improving the SIPP approach for staffing service systems that have cyclic demand*, Operations Research **49** (2001), 549–564.

[ICW02] A. Ingolfsson, E. Cabral, and X. Wu, *Combining integer programming and the randomization method to schedule employees*, Technical report, School of Business, University of Alberta, Edmonton, Alberta, Canada, Preprint, 2002.

[Kei79] E.G. Keith, *Operator scheduling*, AIIE Transactions **11** (1979), no. 1, 37–41.

[PBK06] S.A. Pot, S. Bhulai, and G.M. Koole, *A simple staffing method for multi-skill call centers*, Submitted, 2006.

[Tho93] G.M. Thompson, *Accounting for the multi-period impact of service when determining employee requirements for labor scheduling*, Journal of Operations Management **11** (1993), no. 3, 269–287.

[Tho97] ______, *Labor staffing and scheduling models for controlling service levels*, Naval Research Logistics **44** (1997), no. 8, 719–740.