European Journal of Operational Research 202 (2010) 153–163

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Stochastics and Statistics

Combining integer programming and the randomization method to schedule employees

Armann Ingolfsson [a,*], Fernanda Campello [a], Xudong Wu [b], Edgar Cabral [c]

[a] School of Business, University of Alberta, Edmonton, Alberta, Canada T6G 2R6
[b] Matisse Networks Inc., Mountain View, CA, USA
[c] E-Opt Ltda., Campinas, Brazil

ARTICLE INFO

Article history:
Received 21 May 2008
Accepted 28 April 2009
Available online 7 May 2009

Keywords:
Service operations management
Employee scheduling
Staffing requirements
Nonstationary queues
Randomization method
Integer programming

ABSTRACT

We describe a method to find low cost shift schedules with a time-varying service level that is always above a specified minimum. Most previous approaches used a two-step procedure: (1) determine staffing requirements and (2) find a minimum cost schedule that provides the required staffing in every period. Approximations in the first step sometimes cause the two-step approach to find infeasible or suboptimal solutions. Our method iterates between a schedule evaluator and a schedule generator. The schedule evaluator calculates transient service levels using the randomization method and identifies infeasible intervals, where the service level is lower than desired. The schedule generator solves a series of integer programs to produce improved schedules, by adding constraints for every infeasible interval, in an attempt to eliminate infeasibility without eliminating the optimal solution. We present computational results for several test problems and discuss factors that make our approach more likely to outperform previous approaches.

1. Introduction and literature review

The need to schedule employees is ubiquitous in the service sector. Bank branches, restaurants, retail stores, and airline check-in areas are but a few examples of organizations that need to schedule employees to match the demand for services—which is typically random and varies over time—to the supply of employees providing services. Call centers are perhaps the largest sector in need of employee scheduling. Modern call centers are complex organizations, both technologically and operationally. As this sector grows and matures, and as technology advances, there are increasing opportunities to employ models to improve operations (Gans et al., 2003). Labor is typically the largest cost for a call center (60–70% of total cost according to Gans et al., 2003) and therefore efficient employee scheduling provides substantial opportunities for productivity improvements.

Models for employee scheduling have a long history in the operations research literature. Edie's (1954) classic study of traffic delays at tollbooths used a combination of empirical analysis and formulas for stationary queueing systems to generate the staffing requirements needed to ensure a specified level of service. Soon after, Dantzig (1954), referring to Edie's work, showed how a linear integer program could find shift schedules that provide enough staffing to meet specified requirements in each planning period—such as the ones developed by Edie—at minimum cost.

A typical sequence of steps in scheduling employees is (Buffa et al., 1976):

Step 1: Forecast demand,
Step 2: Convert demand forecasts into staffing requirements,
Step 3: Schedule shifts optimally, and
Step 4: Rostering: Assign employees to shifts.

Current practice (e.g., Fukunaga et al., 2002) and most research on employee scheduling has followed this approach. Edie's paper demonstrated one way of performing *Step 2.* Dantzig's tour scheduling model addressed *Step 3. Steps 1 and 4* are important, but outside the scope of this paper.

The papers by Edie and Dantzig were among the first in two distinct streams of research: one on how to set staffing requirements and the other on how to optimally schedule employees subject to staffing requirements.

Step 2 often uses formulas for stationary $M/M/s$ queueing systems to determine the smallest number of servers (employees) needed to provide a specified level of service (often expressed as the percent of customers who experience queue delay of less than some threshold time). Green et al. (2001) termed this the SIPP (stationary independent period by period) approach. A stream of research in queueing theory has developed better methods to

* Corresponding author. Tel.: +1 780 492 7982; fax: +1 780 492 3325.
E-mail addresses: Armann.ingolfsson@ualberta.ca (A. Ingolfsson), campello@ualberta.ca (F. Campello), xudong_wu@yahoo.com (X. Wu), edgaracabral@gmail.com (E. Cabral).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.04.026

determine employee requirements (for example, see Jennings et al., 1996, Green et al., 2007, and Feldman et al., 2008).

Research on shift scheduling has developed efficient algorithms for special cases (e.g., Bartholdi et al., 1980), heuristics (e.g., Brusco and Jacobs, 1993), and reformulations to allow larger problems to be solved to optimality (e.g., Aykin, 1996).

We will refer to performing *Steps 2 and 3* once in sequence as the "approximate approach." The main simplifying assumption in the approximate approach is that the staffing requirement for a period can be determined independent of staffing in prior periods. The extent to which this assumption is valid determines whether it is reasonable to decouple *Steps 2 and 3*. Kolesar et al. (1975) demonstrated that this approximation is not always warranted. More recently, Green et al. (2001, 2003) conducted extensive experiments to investigate the *reliability* of the SIPP approach, which they defined as the number of half-hours during which the desired service level falls below a desired minimum. They demonstrated that the SIPP approach is, unfortunately, unreliable in many situations. They explored various ways of modifying the SIPP approach while retaining the simplicity of calculations with $M/M/s$ queueing formulas. The most promising of these heuristics was the *lag max* approach, which replaces the average arrival rate over a planning period with the maximum of the arrival rate function over that planning period, shifted forward by one average service time. The lag max approach extends the range of situations where SIPP generates reliable staffing requirements considerably, as Green et al. (2001, 2003) showed. When the approximate approach is justified (see Green et al., 2001, for guidelines), then it should be used, because it is simpler and faster than the approach we will describe. Our focus is on situations where the approximate approach (using either SIPP or lag max to generate staffing requirements) has been demonstrated to be unreliable. However, our approach can also result in cost savings in situations where the approximate approach is reliable, as we demonstrate.

Ingolfsson et al. (2002) described an approach to integrating *Steps 2 and 3*. This article presents an improved implementation of that method, involving two algorithmic components: a schedule generator and a schedule evaluator. The schedule generator searches for good schedules using exact or heuristic optimization. The schedule evaluator estimates the cost and service level of a schedule. In Ingolfsson et al. (2002), the schedule generator used a genetic algorithm, and the schedule evaluator used numeric integration of the forward differential equations for an $M(t)/M/s(t)$ system to evaluate the service level. In this paper, we use an integer programming heuristic to generate schedules and we use the randomization method (Grassmann, 1977) to compute service levels. These algorithmic improvements result in a substantial reduction in computation time, which has allowed us to perform computational experiments to generate insight into when decoupling *Steps 2 and 3* is justified and when it is not. While the method does not guarantee optimality, it provides a good feasible solution and a lower bound on the minimum cost.

We define the service level at time t as the probability that the virtual waiting time is less than a maximum acceptable waiting time τ. Because we solve the forward differential equations, we can compute instantaneous service levels for as many time points as desired, and we define our optimization problem in terms of instantaneous service levels. In related research that uses simulation, as well as in practice, service levels are typically defined as averages over some time period, such as an hour. It is straightforward to modify our approach to conform with such definitions.

Thompson (1997) and Atlason et al. (2004, 2008) described other approaches to integrating *Steps 2 and 3*. Thompson generated staffing requirements using stationary $M/M/s$ formulas, with a heuristic adjustment (described in Thompson, 1993) for transient effects. As discussed in Ingolfsson et al. (2007), this heuristic is similar to Green et al.'s lag max approach and, therefore, shares its limitations. Thompson used slack and surplus decision variables for deviations above or below the requirement for each planning period, with coefficients derived from $M/M/s$ formulas to quantify the impact of these deviations on the service level. He solved the resulting model using Brusco and Jacobs' (1993) simulated annealing heuristic. Atlason et al. (2004) iterate between simulation (to evaluate service levels) and integer programming, with constraints being added to the integer program at each iteration based on approximate subgradients of the service level (estimated using simulation) as a function of staffing in each planning period. Atlason et al. (2008) improved on the approach in Atlason et al. (2004), using "pseudogradients" rather than subgradients. Our approach adds constraints at each iteration as well, but our constraints do not require evaluation of subgradients or pseudogradients of the service level. We compare the performance of our approach to that of Atlason et al. (2008) as part of our computational experiments and we discuss these related approaches further in the last section.

The paper is organized as follows. Section 2 presents an example to illustrate the main issues, Section 3 states the problem formally, Section 4 reviews the randomization method, Section 5 describes an initial parameter estimation procedure, Section 6 presents our integer programming heuristic, Section 7 outlines our computational experiments and results, and Section 8 concludes with observations on how our approach performs and how it could be generalized. An online supplement contains appendices with supplementary material and additional computational results.

2. Example

We will use the following example to illustrate potential shortcomings of performing *Steps 2 and 3* sequentially and how these shortcomings can be addressed. A service system is open 12 hours each day and has a sinusoidally varying arrival rate with two daily peaks (Fig. 1). The planning period (the shortest time interval over which staffing is constant) is 15 minutes. For *Step 2,* we approximate the time-varying arrival rate by its average over each 5-minute interval and we use $M/M/s$ queueing formulas (with a service rate of 2 customers per hour) to determine, for each planning period, the smallest number of servers needed to ensure that at least 80% of customers do not have to wait before commencing service (this is the SIPP approach).

Shifts are four, six, or eight hours long and can start at the beginning of any planning period that allows the shift to end before the facility closes (Appendix A describes the 243 possible shifts). Fig. 2 (upper panel) shows the SIPP staffing requirements and the number of scheduled servers that minimizes the number of server-hours (by solving an integer program), while satisfying the staffing requirements. The lower panel shows the transient ser-

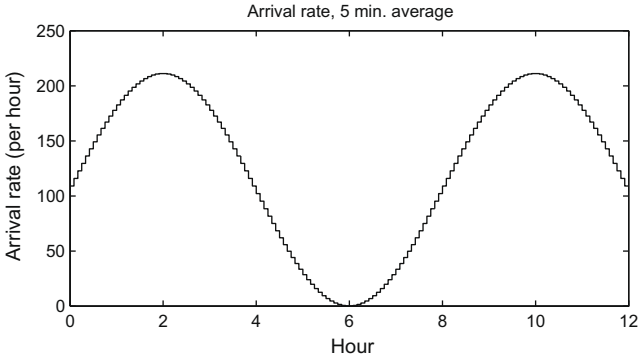

Fig. 1. Arrival rate for example.