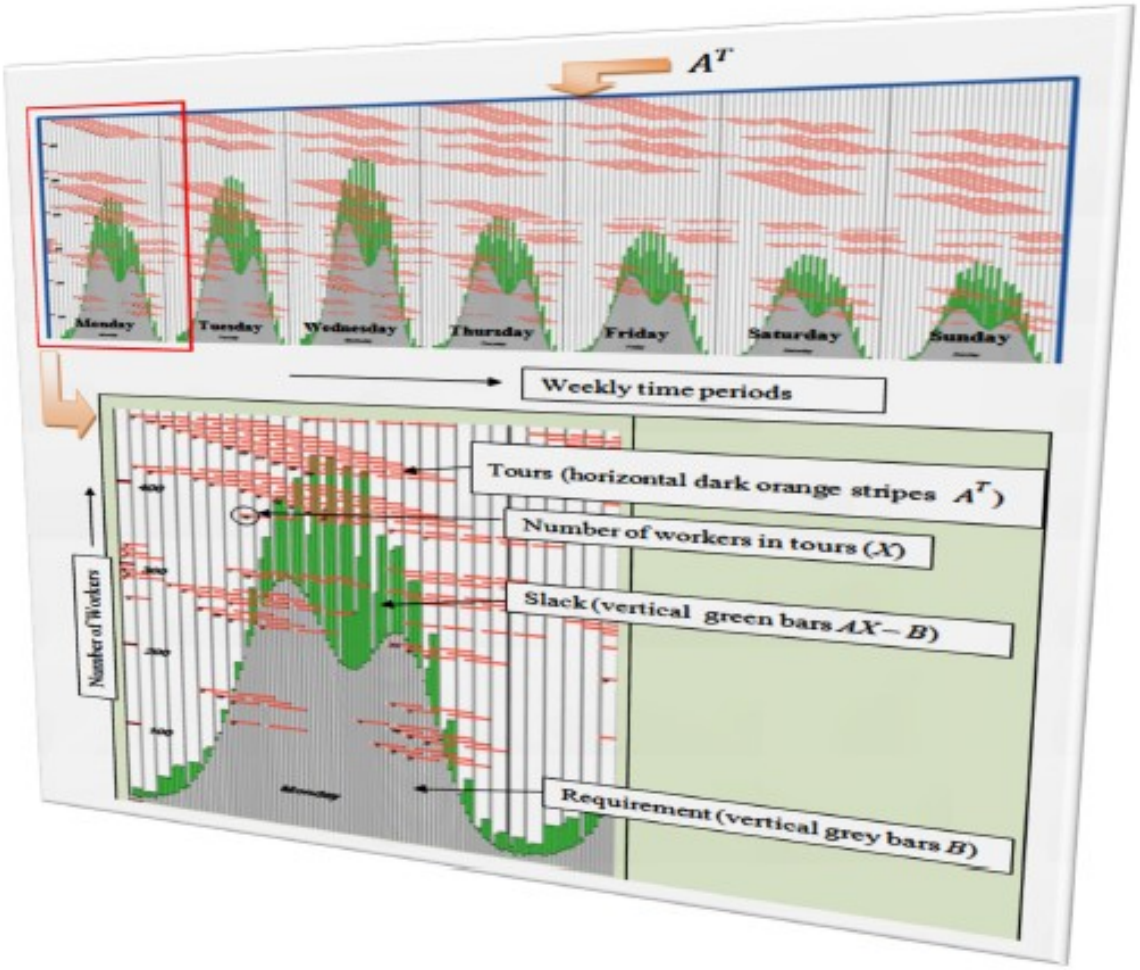
Dantzig's Workforce Scheduling Model

A spreadsheet solution for call centers using public solvers



Manjini Saminathan PhD Bahiru Kassahun PhD Arun K. Singh PhD

Dantzig's Workforce Scheduling Model

A spreadsheet solution for call centers using public solvers

A practical guide for call center analysts to implement Dantzig's seminal workforce scheduling model to generate, evaluate, and optimize service level performance of weekly agent schedules using Microsoft Excel 2007, C#.NET components and publicly available solvers, algorithms, and hardware.

Manjini Saminathan PhD Bahiru Kassahun PhD Arun Singh PhD

ISBN
LCCN

Printed in the United States of America

Preface

One of the major challenges that call centers face today is to operate the center in such a way that its weekly inter- and intra-day variations in the projected call load can be handled at a prescribed service level to its customers with available agent resources at the least possible cost. This can be accomplished by scheduling their agents appropriately, so that enough agents are assigned in each time period during the week to handle the offered load without jeopardizing service quality. A number of commercial software vendors make such a scheduling tool available as part of an overall proprietary workforce management system. A great number of researchers have published on this subject in scholarly journals. These public resources do not often make software tools available to a curious analyst who may have the need to experiment with the published models and algorithms in order to understand potential benefits, if any, that can be derived for his/her center. Also, once a schedule has been generated, a simple to use software tool is not readily available to run the center in a virtual mode using a stochastic simulator. Such an environment will allow the schedule to be exercised prior to deployment using random call arrivals depicting the incoming call load. The effects on service level under expected load and/or unexpected surge conditions can be understood at any time interval during the week. A potential outcome would be to derive a quantitative estimate of the cost and resource requirements needed to provide a given level of service to the customer.

This book bridges the gap that exists today between potentially good published models in the scheduling arena and a great number of software tools to solve such models, all available in the public domain. In particular, the seminal model proposed in 1954 by Prof. Dantzig [1], who invented Linear Programming and the Simplex algorithm seven years earlier, will be exclusively used to illustrate scheduling of agents in a call center. This original model to schedule workers subjected to varying workload requirements at different time periods during a given day laid the groundwork for a number of published call center scheduling models [7, 18, 19, 20, 21, 22, 26, 40] in the literature. In our opinion, this classic model in its original form provides most capabilities needed by a user to design a research tool and conduct a call center scheduling and optimization analysis.

A number of software tools are available in the public domain to solve Dantzig's model. In particular, we will focus on five of the solvers, namely, GLPK [2], lp_solve [3], COIN-OR Cbc [41] and NEOS(FEASPUMP and SCIP) [4] that implement Dantzig's algorithm in some shape or form to obtain a solution of the model in which some or all of the variables can be defined to be integer and/or binary. In our experience, these solvers are capable of handling models that can be encountered in scheduling large call centers employing thousands of agents within a reasonable amount of time using inexpensive and powerful desktop/laptop available in the market today. As of this writing the NEOS sys-

tem even allows the user to submit the model over the internet to be solved on a publicly available hardware cluster and to retrieve the result on line or on demand. The capabilities of these solvers to obtain a schedule for a relatively simple case versus a difficult one involving binary decision variables to implement "either-or" constraints will be discussed. Practical considerations may dictate that the tours in a schedule be staffed at or above a minimum number and sometimes may impose an upper bound on the total number of weekly tours. These added requirements can be handled by using auxiliary on-off decision variables in the model and make the job of obtaining a schedule hard. Therefore, the emphasis in this book is on obtaining a near optimal schedule rather than a globally optimal one which may take prohibitively long to yield a solution, sometimes with an unacceptable transient service level performance. A moderate amount of inefficiency introduced in a near optimal solution sometimes provides a beneficial agents buffer that dampens the swings in the transient service level from falling too far below a specified target in a system in which the call arrivals exhibit a stochastic behavior. A side effect of lower efficiency is increased cost.

This book provides code snippets that explain the application and generation of Dantzig's model for weekly scheduling of agents in a call center and its solution using APIs (application programming interfaces) exposed by the five solvers. The authors exclusively use Microsoft's C#.NET language for all illustration and explain how to P/Invoke (Platform Invoke) methods in the solver modules written in a non-managed code environment. The authors have chosen Microsoft's VSTO to develop an Excel 2007 front end to provide input parameters, conduct a call center scheduling scenario using C# components executing the solver modules in a background thread, and display the schedule as tables and charts. This familiar environment, available on most desktops, offers capabilities beyond a browser needed by today's knowledge worker. However, the components developed can be reused to develop Windows, web, console, or service oriented applications. The approach is to provide the snippets in a step by step manner through the various phases of Dantzig's model generation and solution leading up to an end to end code for a complete console and Excel 2007 implementations.

The framework for a stochastic tool using Monte Carlo methods to simulate the service level performance of a weekly schedule for a given call load is also illustrated. The subject of stochastic simulation is widely available in the literature and will not be covered. However, the results of simulating the generated schedule to estimate transient service levels have been discussed quite extensively. Generating a weekly schedule is a deterministic activity, i.e. the schedule dictates the exact number of workers needed in each tour along with relief breaks. Once the schedule is in place, the random nature of arriving call load handled by the workers in the schedule can produce varying degrees of service levels with an expected value in each planning period. Hence, it may be an arduous task to micro manage the deterministic placement of shifts and relief breaks and find a globally optimal schedule in the context of a stochastic call arrival activity. So, the emphasis in this book is on generating a near optimal solution of Dantzig's model in a relatively short

period of time, estimating its service level performance against the projected call load using a stochastic simulator, and regenerating the schedule, if necessary, by tuning controllable parameters so that a specified target level is met within a given tolerance. In our opinion, understanding a schedule's service level is a very important element of its overall characteristics. Several scenarios to illustrate the effect of scheduling parameters on cost, performance, and resource requirements will demonstrate the effectiveness of the model.

Adjusting the requirement vector in Dantzig's model to account for period to period dependence of call load using stochastic simulation is illustrated. The compensated requirement attenuates swings in transient service level and wait time to a greater degree compared to Stationary Independent Period by Period (SIPP) [33, 34, 35] or Erlang-C requirement. An initial attempt at an alternative formulation of Dantzig's model in the frequency domain is provided. This formulation using Fourier transform of both the requirement and tour vectors to retain only components in the low frequency range considerably reduces the model size compared to its traditional time domain counterpart. A schedule generated using this approach is provided and it compares well with that obtained using its time domain equivalent.

This practical guide is intended for technically advanced call center analysts who constantly strive to find ways to improve the performance of their centers. Knowledge of call center terminology and a working knowledge in Microsoft Excel 2007, C#, Visual Studio 2005, object oriented design, and basic linear and complex number algebra is assumed. Knowledge of Dantzig's simplex algorithm, though useful, is not required. This is implicit in the solver modules. This book's emphasis is on understanding and formulating Dantzig's model and generating schedules quickly. For introductory purposes, the solution of the model can be treated as a black box in which the solver libraries encapsulate all the functionalities needed to generate a schedule. This book facilitates the basic mechanics involved in 1) coding necessary to generate Dantzig's model matrices A, B, and C, 2) invoking the appropriate APIs in the modules that transfer the inputs to the solver engine, 3) generating a near optimal solution, 4) estimating transient service level performance of the schedule using a stochastic simulator, 5) tuning adjustable parameters, if necessary, to regenerate a schedule that meets a target level and 6) presenting the results. Once these basics are understood, the user can experiment with advanced features of the modules to optimize the solution process. *Code snippets in this book for a sample case may serve as a guide for an analyst to expand and build his/her own analysis tool. They were developed and tested using Visual Studio 2005*[1] *on a Wintel desktop and laptop running Windows XP Professional or Vista ultimate. The reader is free to copy and modify the code as he/she sees fit and is responsible for installing all prerequisites. The*

[1] Visual Studio 2005, Excel 2007, C#.NET, VB.NET, LINQ, VSTO, Windows XP Professional, and Vista Ultimate are products of Microsoft Corporation, Redmond, Washington.

authors provide no warranty of any kind. It is the sole responsibility of the user for the code's behavior in his/her machine and adherence to software licensing terms of the solver modules. The authors try to be concise in the presentation of the problem formulation and implementation. An in-depth view of the model, algorithm, and APIs beyond what is covered here can be obtained from the references cited. We welcome comments.

Manjini Saminathan, Bahiru Kassahun and Arun Singh November 2008
E-mail: author@sittananda.com

To our spouses and children

Table of Contents

A practical guide for call center analysts to implement Dantzig's seminal workforce scheduling model to generate, evaluate, and optimize service level performance of weekly agent schedules using Excel 2007, C#.NET components and publically available solvers, algorithms, and hardware. Book features:

- Introduction to Dantzig's seminal workforce scheduling model: Minimize C(transpose)X such that AX >= B, X >= 0. Formulation of tour (A),requirement (B), and objective (C) matrices for weekly tour scheduling of agents

- Model solution using public solvers (GLPK, lp_solve, and NEOS Feasibility Pump and SCIP) and local/public hardware

- Scheduling with upper bound on total number of weekly tours, each tour having lower and upper bounds on number of agents. C# code snippets to create console and Excel 2007/VSTO applications

- Monte Carlo stochastic simulation to estimate schedule service level and ASA. Effects of controllable parameters on schedule quality

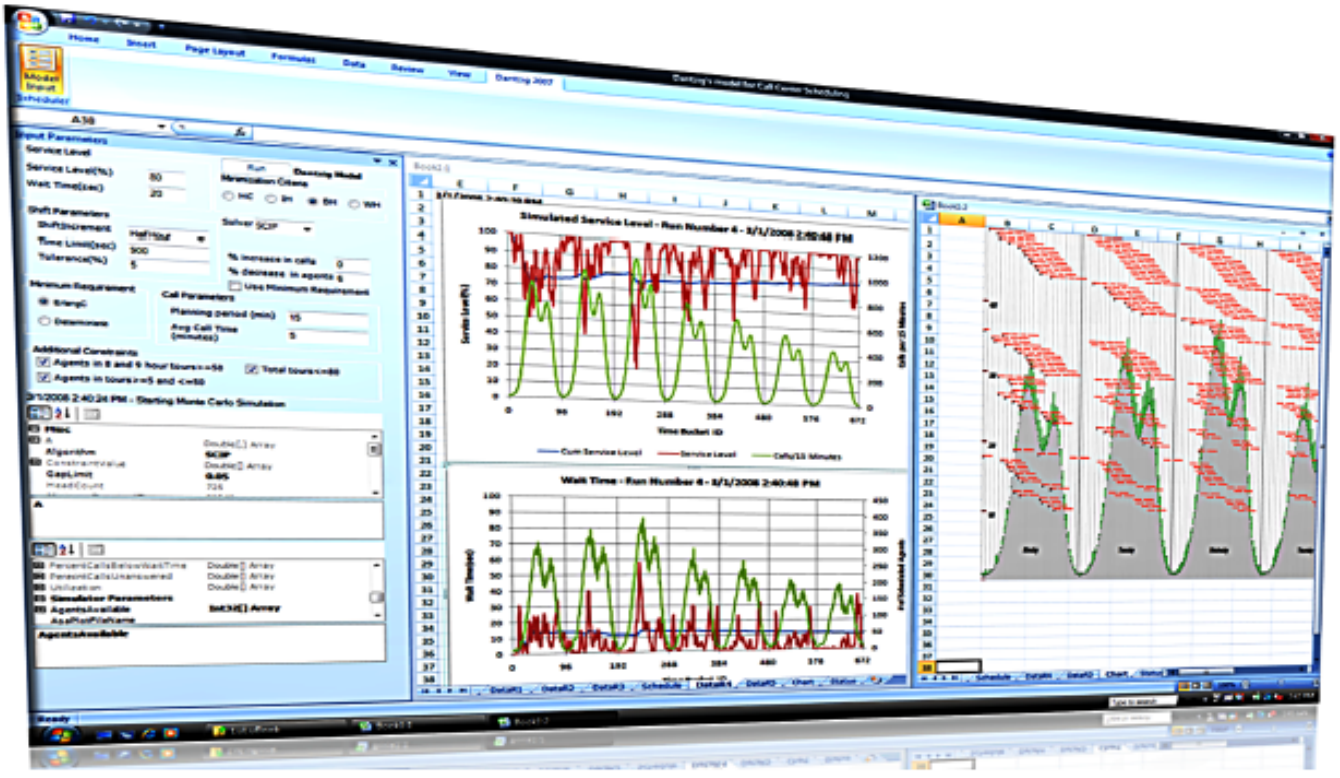
- Near optimal scheduling to meet a target service level for a given call load. Adjust SIPP (Erlang-C) requirement for period dependence to dampen "roller coaster" or "bull whip" effect in service level performance

- A novel frequency domain reformulation of Dantzig's time domain model to reduce size by 75% by retaining only low frequency Fourier components of the requirement (B) and tour (A) vectors

Manjini Saminathan is a software architect and consultant who currently develops engineering applications using the wealth of publically available models, algorithms and software libraries. He has more than two decades of experience in Bell Labs and Lucent in modeling, simulation and optimization of IC manufacturing processes and system assembly operations. He holds a PhD in Chemical Engineering from WPI, Worcester, Massachusetts.

Bahiru Kassahun's expertise is in modeling and optimizing thermal and polymeric processes in semiconductor and optical fiber manufacturing. His current focus is in Finite Element modeling of flow processes. He has more than two decades of experience in Bell Labs and Lucent and holds a PhD in Engineering Mechanics from ASU, Tempe, Arizona.

Arun K. Singh has a wide variety of experience in system engineering, combustion and engineering analysis. He has over 20 years of experience in Bell Labs in design and specification of high-speed optical networks. His focus is in queuing theory and discrete event simulation. He holds a PhD in Mechanical and Aerospace engineering from Rutgers University, New Brunswick, New Jersey.



ID: 2116200
www.lulu.com