Simulation Foundations, Methods and Applications

Andreas Tolk
John Fowler
Guodong Shao
Enver Yücesan *Editors*

Advances in Modeling and Simulation

Seminal Research from 50 Years of Winter Simulation Conferences

Springer

Simulation Foundations, Methods and Applications

More information about this series at http://www.springer.com/series/10128

Andreas Tolk · John Fowler
Guodong Shao · Enver Yücesan
Editors

Advances in Modeling and Simulation

Seminal Research from 50 Years of Winter Simulation Conferences

Springer

Editors
Andreas Tolk
The MITRE Corporation
Hampton, VA
USA

John Fowler
Department of Supply Chain Management
Arizona State University
Tempe, AZ
USA

Guodong Shao
National Institute of Standards and Technology
Gaithersburg, MD
USA

Enver Yücesan
T&OM Area
INSEAD
Fontainebleau
France

ISSN 2195-2817 ISSN 2195-2825 (electronic)
Simulation Foundations, Methods and Applications
ISBN 978-3-319-64181-2 ISBN 978-3-319-64182-9 (eBook)
DOI 10.1007/978-3-319-64182-9

Library of Congress Control Number: 2017947460

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To all the Experts, Scholars, Practitioners, and Students of Modeling and Simulation and its related disciplines who contributed to the last 50 years of Winter Simulation Conferences;

To all the Mentors who sent their Protégés to learn about and contribute to our perpetually extending and evolving understanding of Modeling and Simulation;

And to all the Life Partners who loaned their loved ones to us for hours, days, and weeks of paper writing, peer reviewing, proceedings editing, and all the other countless activities that make the Winter Simulation Conference special!

Foreword

The Winter Simulation Conference is a special conference. It is not necessarily special because of the content, which is world-class, or the venues, which are outstanding. The Winter Simulation Conference is a special conference because it is undertaken every year by a group of volunteers who love to put on the world's leading simulation conference. In my career, I have worked for several organizations, both public and private, and I have been intimately involved in volunteer organizations in my communities and nationally, and I can say with utmost confidence that WSC is the best-run, best-organized, highest-quality labor of love that I am associated with. In my various conference roles since 1985, including Proceedings Editor, Program Chair and now the President of the Winter Simulation Conference Foundation, I have seen that commitment from hundreds of people who liberally give their time and professional expertise to make this conference a success. Everyone in the simulation community is indebted to this outstanding group of people.

This publication celebrates the 50th anniversary of the Winter Simulation Conference. The second Winter Simulation Conference, held in New York City in 1968, was the first to have a *Proceedings*. It was a purely academic conference with no tracks and 22 sessions. This year, the conference will have 26 tracks, including a vendor track and case studies track. The conference will also continue its strong Ph.D. Colloquium and practical workshops. In addition, there is the Modeling and Analysis of Semiconductor Manufacturing (MASM) Conference, multiple keynotes and the *Titans of Simulation* talks. WSC is now a strong and vibrant conference that is both theoretical and applied while serving a wide variety of interests.

This book looks back on 50 years of WSC and tracks the key simulation topics that the conference has covered. Each of the authors is part of the group that I mentioned above—people who love the conference and liberally give their time to make it a success. In addition, each person is a great researcher and an expert in their field. The topics, like the conference, are a mix of theory and practice. The authors look back at history and then imagine where simulation will take us. At the same time, we are looking forward, by featuring some of our "rising stars," presenting their research.

Where will the Winter Simulation Conference be in the future? That is hard to predict. In 2003, I was on a panel session titled, "The Future of The Winter Simulation Conference." As part of that panel, we were asked to predict where the conference would be in 10 years. As with most predictions 10 years out, I missed almost everything. But I did get one thing right—"I will meet my old(er) friends there." In 2013, I did meet my older friends there and I plan to do so for as long as I am able.

I am confident of this as well. This year, there will be a 25-year-old that will attend the conference for the first time. That person will have a passion for simulation that will grow over the years. That person and other like-minded people will join the group of committed volunteers of this conference and guarantee its quality for another generation. That is the future of the Winter Simulation Conference.

Here's to another 50 years!

Ricki G. Ingalls
President, Winter Simulation Conference Foundation
Diamond Head Associates, USA

Preface

This book is a contribution to celebrating the 50th anniversary of the Winter Simulation Conference. The editors are a group of volunteers who also serve currently on the Board of Directors. The chapters are contributed by long-time supporters and Titans of the simulation domain, augmented by rising stars who are starting to shape the community with their research contributions. As such, the chapters touch on where we are coming from, where we are, and where we will hopefully be going.

The book is organized into 16 invited chapters providing an overview of history, research, and applications. As such, it addresses scholars, students, and practitioners of the field who are interested not only in the state of the art, but also in the way this state was reached. These reflections are purposefully written from the personal viewpoints of experts who often were the driving power behind the efforts described in the chapter. The material can therefore be used for background research assignments in graduate and postgraduate education or simply to learn something more about such topics.

The topics addressed in this book are modeling activities, calibration, validation, and input model risk. The evolution of simulation modeling will be visited, and a history about Time Warp is presented. A tutorial on design and analysis of simulation experiments as well as simulation contributions to the Big Data challenge are covered. Research on Bayesian belief models, optimization under uncertainty, and parallel ranking and selection shows the close relation of operations research and simulation. From the application domains, overviews are presented on defense and security, social and behavioral simulation, as well as semiconductor manufacturing.

Finally, profiling research shows the influence of research presented at the conferences justifying the title of this book: Advances in Modeling and Simulation—Seminal Research from 50 Years of Winter Simulation Conferences.

Hampton, USA — Andreas Tolk
Tempe, USA — John Fowler
Gaithersburg, USA — Guodong Shao
Fontainebleau, France — Enver Yücesan
June 2017

Contents

Chapter 1
A Brief Introduction to the Winter Simulation Conference

Andreas Tolk, John Fowler, Guodong Shao and Enver Yücesan

Abstract The Winter Simulation Conference celebrates its 50th anniversary in December 2017. Over these years, it became one of the leading simulation conferences worldwide, with more than 600 annual participants and thousands of publications shaping the international simulation research. This chapter describes the history of the conference, presents some high-level statistics on attendance and publications, describes the current tracks of the conference, and presents the contributions of the various chapters of this book.

1.1 Introduction

In December 2017, the Winter Simulation Conference (WSC) celebrates its 50th anniversary. Among the special events planned to commemorate this milestone will be a track highlighting the history and fundamental contributions these conferences have made over the last decades to help in establishing the discipline of modeling and simulation. The interested reader is referred to the upcoming proceedings that

A. Tolk (✉)
The MITRE Corporation, Ste 200, 903 Enterprsie Pkwy, Hampton
VA 23666, USA
e-mail: atolk@mitre.org

J. Fowler
Department of Supply Chain Management, Arizona State University,
BA 422, Tempe, AZ 85287, USA
e-mail: john.fowler@asu.edu

G. Shao
National Institute of Standards and Technology, 100 Bureau Drive,
Gaithersburg, MD 20899, USA
e-mail: guodong.shao@nist.gov

E. Yücesan
T&OM Area, INSEAD, Blvd de Constance, 77305
Fontainebleau Cedex, France
e-mail: enver.yucesan@insead.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations,
Methods and Applications, DOI 10.1007/978-3-319-64182-9_1

promise to provide a significant contribution to capturing the history of WSC and its topics and accomplishments for the society.

This book is another contribution to this celebration, not only providing with its chapters an additional view at the seminal contributions of our titans, but also featuring our rising stars: honoring our past, but at the same time preparing for a great future. We invited a wide selection of authors who are well known in the WSC community as well as in the broader Modeling and Simulation (M&S) community. We also invited authors who have already shown that they will influence this community with their contributions, as the early work has already influenced other contributions to our conference.

1.2 The History of the Winter Simulation Conference

The early years of simulation are described by Goldsman et al. (2009, 2010), showing how several ideas culminated in the late sixties, leading to an expansion period of computer simulation related research within this timeframe. The Computer Simulation Archive at North Carolina State University (http://d.lib.ncsu.edu/computer-simulation/) chronicles many of the simulation pioneers and provides access to their collection of papers, leading to the blossoming of this new discipline.

The WSC started in this period of expanding research as the *Conference on the Applications of Simulation Using GPSS*[1] in 1967. After the first four years, the conference was officially referred to as the Winter Simulation Conference, the name under which it celebrates its 50th anniversary.

James R. Wilson is one of the leading veterans and titans of WSC and summarized the history in an article for *OR/MS Today,* the magazine for members of the Institute for Operations Research and the Management Sciences (Wilson 1996). His overview has been reused in front matter material summaries for WSC proceedings ever since; augmented by complementary information from Reitman (2008), it also helped in writing this chapter.

In the late 1940s, seminars started to be organized addressing the possibility of computer simulation. The General Purpose Simulation System, a simulation programming language used to build computer models for discrete-event simulations, started to create a common foundation for users of simulation. In the spring of 1967, the interest in a common conference at the national level was perceived to be strong enough to bring sponsors from the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), and a scientific users' group of IBM, called SHARE, together to organize the first conference. Motivated by the technical and financial success, a second conference was organized the following year. The sponsorship was broadened to include the Simulation Councils, Inc. (SCi) that later became the Society for Computer Simulation (SCS)

[1]General Purpose Simulation System.

and today is the Society for Modeling and Simulation. With the new sponsorship, the focus was also broadened as SCi did not focus on GPSS, but were at this time more interested in continuous systems simulation, which was the main focus of the Summer Simulation Conferences. The new scope attracted new attendees and increased the number of papers, leading to the first proceedings. The third conference included the American Institute of Industrial Engineers, the predecessor of today's Institute of Industrial and Systems Engineers (IISE), and The Institute of Management Sciences/College of Simulation and Gaming (TIMS/CSG), the predecessor of today's Institute for Operations Research and the Management Sciences (INFORMS) as new sponsors. In 1971, the name was officially changed to Winter Simulation Conference. Although no records survived from the 1971 conference, it is believed to have been the largest conference in the history of WSC. Due to growing administrative and organizational challenges resulting from the rapid growth, the conference planned for 1972 slipped into 1973. The Operations Research Society of America (ORSA) became a sponsor of the conference in 1974. However, by that time, the attendance numbers had already decreased since the ad hoc nature of the early days could not sustain a conference of this new magnitude. By 1975, the administrative structure imploded, leading to the necessity to cancel the 1975 WSC completely.

This could have easily been the end of the WSC, but a group of highly devoted simulation experts took the initiative to set the WSC back on solid tracks. Robert G. Sargent, Paul F. Roth, Harold J. Highland, and Thomas J. Schriber revived the idea of a national simulation conference by establishing a Board of Directors (BoD) and a set of bylaws to overcome the administrative shortcomings. They also brought the National Bureau of Standards (NBS), the predecessor of today's National Institute of Standards and Technology (NIST), into the circle of sponsors. With the BoD and rigorous guidelines ensuring academic excellence in place, the WSC started to grow again. Industry started to become interested in supporting an exhibition of emerging simulation tools in 1984. The American Statistical Association joined as a sponsor in 1985, bringing a new group of simulation methods and simulation software users to the conference. The academic rigor was strengthened by formalizing the peer-review process, requiring written referee reports for every contributed paper in 1990. With the transformation to a totally web-based system for submission, review, and final delivery of all articles in 1999, these processes were further enhanced. By constantly and consistently updating their bylaws, updating the scope of the conference, and utilizing latest technical developments, the WSC became a stable factor in the conference landscape not only for the M&S community, but for leading scientists in general.

Table 1.1 shows the location, leadership, and paid attendance for the WSC from 1967 to 2016, as published on the official website for the annual conference (http://www.wintersim.org). For the periods 1967–1968 and 1974–2016, the attendance figures shown in the table represent total paid attendance including students, but excluding exhibitor-only registrants. For the period 1969–1973, no surviving records of WSC attendance are held by any of the sponsoring societies. Consequently, attendance estimates were obtained by discussions with the Program

Chairs and General Chairs for this period. Although the figures for the years 1969–1973 are subject to some uncertainty, there is little doubt that in the period 1970–1971, attendance at the WSC was substantially higher than in all other years.

Table 1.1 uses the following abbreviations to denote the leadership roles: General Chair (GC), Program Chair (PC), Associate Program Chair (APC), Proceedings Editor (PE), Proceedings Co-Editor (PCE), and Associate Proceedings Editor (APE).

The attendance is plotted in Fig. 1.1.

Today, WSC is sponsored by a broad variety of professional organizations and societies, namely

- American Statistical Association (ASA)
- Arbeitsgemeinschaft Simulation (ASIM)
- Association for Computing Machinery: Special Interest Group on Simulation (ACM/SIGSIM)
- Institute for Operations Research and the Management Sciences: Simulation Society (INFORMS-SIM)
- Institute of Electrical and Electronics Engineers/Systems, Man, and Cybernetics Society (IEEE/SMCS)
- Institute of Industrial and Systems Engineers (IISE)
- National Institute of Standards and Technology (NIST)
- The Society for Modeling and Simulation International (SCS)

ACM/SIGSIM, IISE, INFORMS-SIM, and SCS, which are financial sponsors sharing equal fiduciary responsibility, also send two representatives each to form the voting members of the Board of Directors. The other organizations are technical co-sponsors of the conference. Each technical co-sponsor has a single representative on the Board of Directors; these representatives vote on all issues before the Board except those with financial implications. However, they participate in tiger teams to address new challenges, provide technical and organizational advice, and are the liaison to their societies and organizations.

WSC Proceedings are peer-reviewed contributions to the body of knowledge of M&S and archived by the digital libraries of ACM as well as IEEE. The Proceedings have also been indexed in the Thomas Reuters Institute for Scientific Information (ISI) Proceedings since 2000. All proceedings since 1968 have been digitized and can be accessed as PDF files via the program archives hosted by INFORMS and the digital libraries of ACM and IEEE.[2]

Academic rigor in the peer-review process was understood to be as important as the professional preparation of the proceedings and has been following a formalized process since 1990. Table 1.2 summarizes the numbers of submitted and accepted papers and the resulting acceptance rate since this information was tracked in 1994.

[2]The first conference did not produce proceedings, but Julian Reitman edited a special issue of the IEEE Transactions on Systems Science and Cybernetics (Volume SSC-4, Number 4, November 1968) that contained some of the papers presented at the 1967 conference.

Table 1.1 Location, leadership, and paid attendance of the winter simulation conferences

Conference, date, and location	Leadership	Attendance
Conference on the Applications of Simulation Using GPSS 13–14 November 1967 Hilton Hotel, New York, NY	Harold G. Hixson (GC) Julian Reitman (PC)	401
Second Conference on the Applications of Simulation 2–4 December 1968 Hotel Roosevelt, New York, NY	Julian Reitman (GC) Arnold Ockene (PC)	856
Third Conference on the Applications of Simulation 8–10 December 1969 International Hotel, Los Angeles, CA	Arnold Ockene (GC) Philip J. Kiviat (PC)	400
Fourth Conference on the Applications of Simulation 9–11 December 1970 Waldorf-Astoria Hotel, New York, NY	Philip J. Kiviat (GC) Michael Araten (PC)	1100
The 1971 Winter Simulation Conference 8–10 December 1971 Waldorf-Astoria Hotel, New York, NY	Michael Araten (GC) Joseph Sussman (PC)	1200
The 1973 Winter Simulation Conference 17–19 January 1973 St. Francis Hotel, San Francisco, CA	Joseph Sussman (GC) Austin C. Hoggatt (PC)	600
The 1974 Winter Simulation Conference 14–16 January 1974 Washington Hilton Hotel, Washington, DC	Michael F. Morris (GC) Harold Steinberg (PC) Harold J. Highland (PE)	463
The 1976 Bicentennial Winter Simulation Conference 6–8 December 1976 National Bureau of Standards, Gaithersburg, MD	Harold J. Highland (GC) Thomas J. Schriber (PC) Robert G. Sargent (APC)	306
The 1977 Winter Simulation Conference 5–7 December 1977 National Bureau of Standards, Gaithersburg, MD	Robert G. Sargent (GC) J. William Schmidt (PC) Harold J. Highland (PE)	465
The 1978 Winter Simulation Conference 4-6 December 1978 The Deauville Hotel, Miami Beach, FL	Larry G. Hull (GC) Norman R. Nielsen (PC) Harold J. Highland (PE)	388
The 1979 Winter Simulation Conference 3–5 December 1979 Holiday Inn Embarcadero, San Diego, CA	Mitchell G. Spiegel (GC) Robert Shannon (PC) Harold J. Highland (PE)	375
The 1980 Winter Simulation Conference 3–5 December 1980 Orlando Marriott, Orlando, FL	Paul Roth (GC) Tuncer I. Ören (PC, PE) Charles M. Shub (APC)	205
The 1981 Winter Simulation Conference 9–11 December 1981 Peachtree Plaza Hotel, Atlanta, GA	Claude M. Delfosse (GC) Charles M. Shub (PC) Tuncer I. Ören (PE)	267
The 1982 Winter Simulation Conference 6–8 December 1982 Holiday Inn Embarcadero, San Diego, CA	Yen W. Chao (GC) Orlando Madrigal (PC) Harold J. Highland (PE)	274
The 1983 Winter Simulation Conference 12–14 December 1983 Crystal Gateway Marriott Hotel, Arlington, VA	Jerry Banks (GC) Bruce W. Schmeiser (PC) Stephen D. Roberts (PE)	416
The 1984 Winter Simulation Conference 28–30 November 1984 Sheraton Dallas Hotel, Dallas, TX	Udo W. Pooch (GC) C. Dennis Pegden (PC) Sallie Sheppard (PE)	350
The 1985 Winter Simulation Conference 11–13 December 1985	Gerard C. Blais (GC) Susan L. Solomon (PC)	369

(continued)

Table 1.1 (continued)

Conference, date, and location	Leadership	Attendance
San Francisco Hilton and Tower, San Francisco, CA	Donald T. Gantz (PE)	
The 1986 Winter Simulation Conference 8–10 December 1986 Radisson Mark Plaza Hotel, Washington, DC	James O. Henriksen (GC) Stephen D. Roberts (PC) James R. Wilson (PE)	531
The 1987 Winter Simulation Conference 14–16 December 1987 The Ritz Carlton, Buckhead, Atlanta, GA	Hank Grant (GC) W. David Kelton (PC) Arne Thesen (PE)	475
The 1988 Winter Simulation Conference 12–14 December 1988 San Diego Marriott, San Diego, CA	Peter L. Haigh (GC) John C. Comfort (PC) Michael A. Abrams (PE)	535
The 1989 Winter Simulation Conference 4–6 December 1989 Capital Hilton Hotel, Washington, DC	Kenneth J. Musselman (GC) Philip Heidelberger (PC) Edward A. MacNair (PE)	619
The 1990 Winter Simulation Conference 9–12 December 1990 The Fairmont Hotel, New Orleans, LA	Randall P. Sadowski (GC) Richard E. Nance (PC) Osman Balci (PE)	517
The 1991 Winter Simulation Conference 8–11 December 1991 The Arizona Biltmore, Phoenix, AZ	W. David Kelton (GC) Gordon M. Clark (PC) Barry L. Nelson (PE)	540
The 1992 Winter Simulation Conference 13–16 December 1992 Crystal Gateway Marriott Hotel, Arlington, VA	Robert C. Crain (GC) James R. Wilson (PC) James J. Swain (PE) David Goldsman (APE)	734
The 1993 Winter Simulation Conference 12–15 December 1993 The Biltmore Hotel, Los Angeles, CA	Edward C. Russell (GC) William E. Biles (PC) Gerald W. Evans (PE) Mansooreh Mollaghasemi (APE)	572
The 1994 Winter Simulation Conference 11–14 December 1994 Walt Disney World Swan Hotel, Orlando, FL	Deborah A. Sadowski (GC) Andrew F. Seila (PC) Jeffrey D. Tew (PCE) S. Manivannan (PCE)	667
The 1995 Winter Simulation Conference 3–6 December 1995 Hyatt Regency Crystal City, Arlington, VA	William R. Lilegdon (GC) David Goldsman (PC) Christos Alexopoulos (PCE) Keebom Kang (PCE)	652
The 1996 Winter Simulation Conference 8–11 December 1996 Hotel Del Coronado, Coronado, CA	Daniel T. Brunner (GC) James J. Swain (PC) John M. Charnes (PCE) Douglas J. Morrice (PCE)	649
The 1997 Winter Simulation Conference 7–10 December 1997 Renaissance Waverly Hotel, Atlanta, GA	David H. Withers (GC) Barry L. Nelson (PC) Sigrún Andradóttir (PE) Kevin J. Healy (APE)	634
The 1998 Winter Simulation Conference 13–16 December 1998 Grand Hyatt Washington, Washington DC	John S. Carson (GC) Mani S. Manivannan (PC) D. J. Medeiros (PCE) Edward F. Watson (PCE)	778

(continued)

Table 1.1 (continued)

Conference, date, and location	Leadership	Attendance
The 1999 Winter Simulation Conference 5–8 December 1999 Squaw Peak, Phoenix, AZ	David T. Sturrock (GC) Gerald W. Evans (PC) P. A. Farrington (PCE) H. B. Nemhard (PCE)	671
The 2000 Winter Simulation Conference 10–13 December 2000 Wyndham Palace Resort & Spa, Orlando, FL	Paul A. Fishwick (GC) Keebom Kang (PC) Jeffrey A. Joines (PCE) Russell R. Barton (PCE)	710
The 2001 Winter Simulation Conference 9–12 December 2001 Crystal Gateway Marriott, Arlington, VA	Matt Rohrer(GC) Deb Medeiros (PC) Brett A. Peters (PCE) Jeffrey Smith (PCE)	496
The 2002 Winter Simulation Conference 8–11 December 2002 Manchester Grand Hyatt San Diego, San Diego, CA	Jane L. Snowdon (GC) John M. Charnes (PC) Enver Yücesan (PCE) Chun-Hung Chen (PCE)	550
The 2003 Winter Simulation Conference 7–10 December 2003 The Fairmont New Orleans, New Orleans, LA	David Ferrin (GC) Douglas J. Morrice (PC) Paul J. Sanchez (PCE) Stephen Chick (PCE)	566
The 2004 Winter Simulation Conference 5–8 December 2004 Washington Hilton and Towers Washington, D.C.	Jeff Smith (GC) Brett Peters (PC) Ricki G. Ingalls (PCE) Manuel D. Rossetti (PCE)	687
The 2005 Winter Simulation Conference 4–7 December 2005 Hilton in the Walt Disney World Resort, Orlando, FL	Frank "Brad" Armstrong (GC) Jeffrey A. Joines (PC) Natalie Steiger (PCE) Michael E. Kuhl (PCE)	700
The 2006 Winter Simulation Conference 3–6 December 2006 Portola Plaza Hotel, Monterey, CA	David Nicol (GC) Richard Fujimoto (PC) Barry Lawson (PCE) Jason Liu (PCE) Felipe Perrone (PCE) Fred Wieland (PCE)	658
The 2007 Winter Simulation Conference 9–12 December 2007 J.W. Marriott Hotel, Washington, D.C.	Jeff Tew (GC) Russell Barton (PC) John Fowler (APC) Shane Henderson (PCE) Bahar Biller (PCE) Ming-hua Hsieh (PCE) John Shortle (PCE)	736
The 2008 Winter Simulation Conference 7–10 December 2008 Hotel Intercontinental Miami, Miami, FL	Tom Jefferson (GC) John Fowler (PC) Ricki Ingalls (APC) Scott Mason (PCE) Ray Hill (PCE) Lars Moench (PCE) Oliver Rose (PCE)	654
The 2009 Winter Simulation Conference 13–16 December 2009 Hilton Austin Hotel, Austin, TX	Ann Dunkin(GC) Ricki Ingalls (PC) Enver Yücesan (APC) Manuel Rossetti(PCE) Ray Hill(PCE) Björn Johansson (APE)	598

(continued)

Table 1.1 (continued)

Conference, date, and location	Leadership	Attendance
The 2010 Winter Simulation Conference 5–8 December 2010 Marriott Waterfront Hotel, Baltimore, MD	Joseph Hugan (GC) Enver Yücesan (PC) Michael Fu (APC) Björn Johansson (PCE) Sanjay Jain (PCE) Jairo Montoya-Torres (PCE)	702
The 2011 Winter Simulation Conference 11–14 December 2011 Grand Arizona Resort Phoenix, AZ	Preston White(GC) Michael Fu (PC) Sanjay Jain (PCE) Roy Creasey (PCE) Jan Himmelspach (PCE)	709
The 2012 Winter Simulation Conference 9–12 December 2012 Intercontinental Hotel Berlin, Germany	Oliver Rose (GC) Adelinde Uhrmacher (PC) Markus Rabe (Local Chair) Christoph Laroque (PCE) Raghu Pasupathy (PCE) Jan Himmelspach (PCE)	701
The 2013 Winter Simulation Conference 8–11 December 2013 J.W. Marriott Hotel, Washington, D. C.	Ray Hill (GC) Michael Kuhl (PC) Raghu Pasupathy (PCE) Seong-He Kim (PCE) Andreas Tolk (PCE)	742
The 2014 Winter Simulation Conference 7–10 December 2014 Savannah Intl. Trade & Convention Ctr	Stephen J. Buckley (GC) John A. Miller (PC) Andreas Tolk (PCE) Levent Yilmaz (PCE) Saikou Y. Diallo (PCE) Ilya O. Ryzhov (PCE)	655
The 2015 Winter Simulation Conference 6–9 December 2015 Hyatt Regency Huntington Beach Resort and Spa, Huntington Beach, CA	Charles M. Macal (GC) Manuel D. Rossetti (PC) Levent Yilmaz (PCE) Il-Chul Moon (PCE) Wai Kin (Victor) Chan (PCE) Theresa Roeder (PCE)	657
The 2016 Winter Simulation Conference 11–14 December 2016 Crystal Gateway Marriott Hotel, Arlington, VA	Todd Huschka (GC) Stephen Chick (PC) Theresa Roeder (PCE) Peter Frazier (PCE) Robert Szechtman (PCE) Enlu Zhou (PCE)	628
The 50th Anniversary Winter Simulation Conference December 3–6 Red Rock Resort, Las Vegas, Nevada	Ernest H. Page (GC) Gabriel Wainer (PC) Victor Chan (PCE) Andrea D'Ambrogio (PCE) Gregory Zacharewicz (PCE) Navonil Mustafee (PCE)	

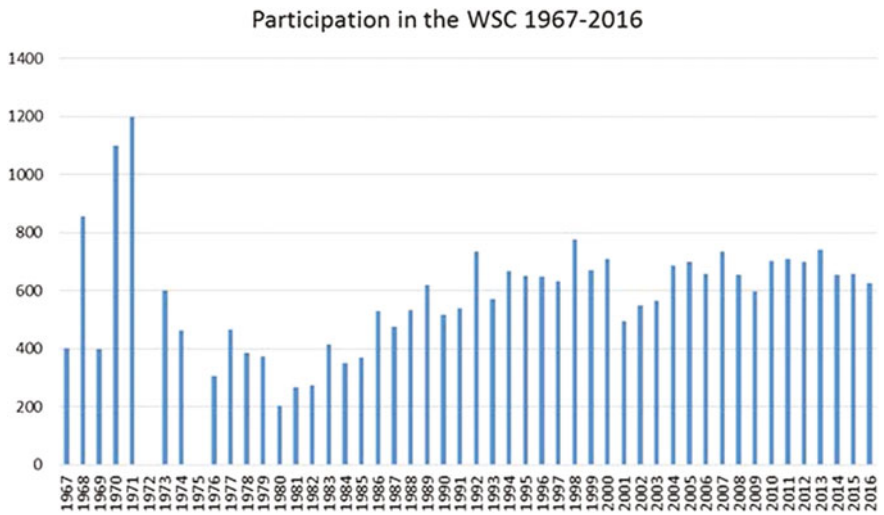


Fig. 1.1 WSC attendance over the years

Table 1.2 Submitted and accepted full papers (1994–2016)

Year	Submitted	Accepted	Rate (%)
WSC '94	209	100	48
WSC '95	183	122	67
WSC '96	187	128	68
WSC '97	191	121	63
WSC '98	216	164	76
WSC '99	206	139	67
WSC '00	227	158	70
WSC '01	155	111	72
WSC '02	185	166	90
WSC '03	189	128	68
WSC '04	171	144	84
WSC '05	316	209	66
WSC '06	252	177	70
WSC '07	244	152	62
WSC '08	304	249	82
WSC '09	256	137	54
WSC '10	281	184	65
WSC '11	270	203	75
WSC '12	384	189	49
WSC '14	320	205	64
WSC '15	296	202	68
WSC '16	254	174	69
Overall	5,296	3,562	67

WSC is established today as a leading academic event with strong industrial support and interest, and draws participants from all parts of the world. 2012 was the first time that the WSC was held outside of the United States, when it was hosted in Berlin, Germany. The success of this experiment encouraged the Board of Directors to plan for future conferences outside of the United States again, in order to reach and serve the members of the international M&S community better. The next event outside the United States is planned for 2018, when the WSC will be held in Gothenburg, Sweden.

1.3 Structure of Current WSC

WSC is held each year during the first or second week of December. The program runs from Sunday to Wednesday. The principle components of the programs are "tracks" which are organized by topical area and consist of some 90-min sessions. Established tracks with respective interest span over the full 3 days of the conference, filling all nine sessions (three on Monday, four on Tuesday, and two on Wednesday), while new ideas and special interest tracks can be limited to two sessions. The tracks provide the program chair with the structure and flexibility to address topics of all research domains. The following is a description of a selection of typical tracks featured at the current WSC.

- Two **Tutorial Tracks** are offered at the WSC, an introductory and advanced tutorial track. The Introductory tutorials track is oriented toward professionals in M&S interested in broadening or refreshing their knowledge of the field. Tutorials cover all areas including mathematical and statistical foundations, methods, application areas, and software tools. Advanced tutorials cover special topics or deepening the understanding beyond the introductory level. Typical advanced topics are domain specific applications, such as defense, healthcare, security, or transportation, but also topics considering the details of operational validation, dealing with uncertainty, or implementation details for simulation paradigm software solutions.
- The **Analysis Methodology** track is intended to cover a variety of empirical, computational, mathematical, and statistical techniques in the context of their application to simulations. Papers covering the construction and calibration of simulation inputs that either improve upon standard approaches or introduce new methods are encouraged. Similarly, papers covering the analysis of simulation output that aims to meaningfully interpret the information produced by simulations and allows modelers to make useful inferences regarding the simulated system are also included. Finally, the papers that deal with the efficiency, accuracy, and appropriateness of a simulation as a representative model of some actual system are also covered by the Analysis Methodology track. The focus of this track is to explore methods for obtaining better inputs, estimates or inferences using practical or novel approaches.

- The **Modeling Methodology** track is interested in methodological advances with respect to the theory and practice of modeling and simulation. These may include approaches to formal model development, data capture, model building, verification, validation, experimentation, and optimization. New modeling and simulation formalisms and extensions to current formalisms are welcome. Of special interest are formalisms that are able to integrate models described in different paradigms. Contributions to the advancement of the technology, standards, and the software used to support modeling are also welcome as are contributions featuring guiding or unifying frameworks, the development and application of formal methods, and lessons learned.
- The **Simulation Optimization** track focuses on algorithms that can be coupled with computer simulations to locate specific input parameter values for the simulation that maximize or minimize a simulation performance measure of interest. This track is interested in papers on both theoretical aspects of algorithm development and applied aspects of simulation optimization pertaining to computational performance and algorithm evaluation. New real-world applications of simulation optimization are also of interest with more desired areas including manufacturing, healthcare, military and homeland security, critical infrastructure systems, cybersecurity, network applications, communications, financial engineering, and energy systems.
- The **Agent-Based Simulation** (ABS) track is a relatively new track that is interested in theoretical, methodological, and applied research that involves synergistic interaction between simulation and agent technologies. Contributions to the ABS track are expected to use agent-based models of complex adaptive systems and self-organizing emergent phenomena with applications to fields such as biomedical sciences, business, engineering, environment, individual, group, organizational behavior, social systems, and intelligent transportation systems.
- **Hybrid Simulation methods** enable stakeholders to analyze and evaluate strategies for effective management of complex systems. It is therefore not surprising that an increasing number of studies have used techniques such as discrete-event simulation, Monte Carlo simulation, system dynamics, Markov chains, and agent-based simulation to make better and more informed decisions. However, such techniques have frequently been applied in isolation. The complexity of systems and their multi-faceted relationships may mean that the combined application of simulation methods will enable synergies across techniques and will provide greater insights into problem solving. The track focuses on combining techniques (e.g., discrete and continuous) and research demonstrating the need for hybrid simulation.
- The **Social and Behavioral Simulation** track features recent, principled work in the domain of computational social science using simulation methods. Computer simulation is increasingly being adopted as a technique for achieving results in the social sciences. Formalized models enable a generative approach to science that can identify what kinds of micro-level interactions are sufficient to produce the known macro-level patterns observed in real societies. Simulation

also allows social science researchers to explore out-of-equilibrium system behavior that is difficult to achieve with traditional analytical approaches.

- The **Healthcare Applications** track addresses an important area in which simulation can provide critical decision support for operational and strategic planning and decision-making that individual providers (doctors/nurses, clinics, urgent care centers, hospitals) face, as well as for policy issues that must be addressed by administering systems (e.g., hospitals, insurance companies, and governments). Traditionally, this track has been broad in focus, incorporating Discrete Event Simulation, System Dynamics, Agent-Based Simulation, and/or Monte Carlo simulations, with a variety of applications. A common thread is the use of simulation tools to provide insight into the decisions for improved healthcare outcomes. New modeling tools that address challenges with the conceptualization or implementation of healthcare systems, and general healthcare simulations are welcome.
- The **Manufacturing Applications** track is interested in research using simulation in industrial applications found in the automotive, aircraft and shipbuilding industries, among others. Simulation is a well-established model-based methodology for analyzing dynamical inter-dependencies in manufacturing systems. Manufacturing applications relate to the model-based analysis of (1) all production and logistics processes within a company or along a supply chain, and (2) all phases of a system life cycle, such as system acquisition, system design and planning, implementation, start of operation, ramp-up, as well as the operation itself. A contribution must describe the aims of investigation, the investigated system, the simulation model, the experimental plan, the simulation findings and any implementation results. Additionally, specific solutions for challenges such as system complexity, data collection and preparation, or verification and validation may be included.
- The **Logistics, Supply Chain Management, and Transportation** track is well established. The nature of highly dynamic and complex networks of supply, intralogistics, and distribution leads to decreasing transparency and increasing risk. Therefore, managers who are responsible for supply chain management and logistics require effective tools to provide credible analysis in this dynamic environment. In order to facilitate the discussion of the best applications of simulation in this area, this track includes papers in logistics simulation, supply chain simulation, and simulation for planning, analyzing, and improving logistics from the intralogistics view to global supply chains.
- **Modeling and Analysis of Semiconductor Manufacturing (MASM)** provides a forum for the exchange of ideas and industrial innovations between researchers and practitioners from around the world involved in modeling and analysis of complex high-tech manufacturing systems. The focus is the use of the entire suite of operations research and statistical tools and techniques, including but not limited to discrete-event simulation, aimed at improving semiconductor manufacturing operations.
- The **Military, Homeland Security, and Emergency Response Applications** track features papers that describe the application of simulation methods to

problems in the military, homeland security, and emergency response. Applications may be in any area related to military such as battlefield simulation, military logistics/transport, military manpower planning, unmanned systems, etc. Homeland and emergency response applications are sought in the protection of critical infrastructure, transportation security, bio-defense, and the phases of the emergency response lifecycle, i.e., preparation/training, response, recovery, and mitigation. Simulation applications that illustrate the relationships between the military and homeland security and emergency response are especially welcome.

- The **Simulation Education** track includes papers and panels from professionals in all disciplines including but not limited to engineering, sciences, arts, humanities, and social sciences to share experiments, lessons learned, projects, methods, tools, and case studies on how to train and educate students, scientists, and scholars at all levels and of all kinds to adopt and incorporate simulation in their work.
- WSC does not only provide an exhibition area, it also features a **Vendor** track that provides an opportunity for companies that market modeling and simulation technology and services to present their innovations and successful applications. The track is open only to companies that have paid for exhibit space at the conference. For each reserved booth, vendors get a 45-min time slot in the track. Vendors can choose to present a paper on conducting research with their products. These papers are subject to the standard WSC submission timeline and review process and appear in the archival proceedings. This track is complemented by vendor workshops that are conducted on Sundays.

The number and topic of tracks fluctuate from year to year, and the selection of topics featured in this chapter is neither complete nor exclusive.

In addition to these tracks, WSC features a PhD Colloquium that allows graduating doctoral students to present their work in a condensed form to their peers as well as experts in their chosen domain. A poster session allows the presentation of work in progress and offers a timely venue to present and discuss new modeling and simulation research through a forum encouraging graphical presentation, demonstration, and active engagement among WSC participants.

A specialty of the WSC since 2004 are the presentations of *Titans of Simulation*. These titan talks feature internationally recognized experts who provided widely disseminated technical contributions of highest impact. The following table shows the titans of simulations who presented at the WSC (Table 1.3).

Over the last decades, the WSC has become the premier international forum for disseminating recent advances in the field of system simulation, with the principal focus being discrete-event simulation and combined discrete-continuous simulation. In addition to a technical program of unsurpassed scope and high quality, WSC provides the central meeting place for simulation researchers, practitioners, and vendors working in all disciplines and in industrial, governmental, military, service, and academic sectors.

Table 1.3 Titans of simulation presentations

Year	Titans of simulation
2004	Phil Kiviat and Devadas Pillai
2005	Jack P. C. Kleijnen and C. Denis Pegden
2006	James O. Henriksen
2007	*Instead of titan talks, a series of landmark papers was presented by* David Goldsman, James O. Henriksen, Pierre L'Ecuyer, Barry L. Nelson, David H. Withers, and Nilay Tanik Argon
2008	Bruce Schmeiser and Mike Pidd
2009	Paul Fishwick and Thomas J. Schriber
2010	Philip Heidelberger and Paul Kleindorfer
2011	Lee W. Schruben and James R. Wilson
2012	Gianfranco Balbo
2013	Richard E. Nance and Barry L. Nelson
2014	John A. Swanson and Richard M. Fujimoto
2015	Averill M. Law and Pierre L'Ecuyer
2016	Edward H. Kaplan and Susan M. Sanchez
2017	Robert G. Sargent and Bernard P. Zeigler

As pointed out by Nance (2013), the WSC is the result of a remarkable collaborative effort that has been led entirely by volunteers for over four decades and that is based on a unique, longstanding cooperative arrangement among eight major professional organizations.

1.4 Contributions of the Chapters

When the editors invited chapter authors for this book, they had three objectives: to put the spotlight on a group of WSC veterans and titans who helped establishing the conference and ensured the high-quality standard of contributions; to feature a group of promising young contributors, as our "rising stars," who already made themselves a name within the WSC community, and who hopefully are going to continue to fuel the spirit of WSC in the future; and to showcase the many contributions that the WSC community has made to the broader modeling and simulation community as a complement to the History Track to be held at the 50th Winter Simulation Conference and to be archived in the WSC 2017 Proceedings.

Simulation is a powerful M&S approach as it enables the incorporation of arbitrary levels of complexity for depicting, analyzing, and optimizing the dynamic behavior of systems. Constructing valid and efficient models, however, is not a trivial task. In Chap. 2, Schruben addresses this challenge and emphasizes that the process of simulation modeling can be as valuable as the resulting computer model itself. Xu, in Chap. 3, discusses model calibration, the iterative process of comparing the outputs of a simulation model with the observed quantities in the real

system, and making changes to model input parameters accordingly to achieve an acceptable level of agreement between the simulation model and the real system. In Chap. 4, Szabo and Birdsey highlight the inherent challenges in validating emergent behaviors in large-scale complex systems with nonlinear interactions, as an invalid behavior can be indistinguishable from an emergent behavior that has never been seen before. They propose a two-step process of identification and comparison to alleviate this problem. Nelson and Song, in Chap. 5, discuss input model risk, the impact of not knowing the true, correct distributions of the basic stochastic processes that drive a computer simulation. The uncertainty in the input models that propagates to the estimates of the output performance measures must be properly quantified and adequately managed.

Valid models have to be implemented or instantiated in an efficient manner to harvest the full benefit of a simulation study. Pegden, in Chap. 6, traces the evolution of simulation languages, making this powerful approach accessible to a vast number of users in an equally vast number of fields. In Chap. 7, Jefferson and Fujimoto discuss the time warp algorithm to significantly increase execution speed.

The effectiveness of a simulation study can be enhanced further through rigorous experimental design. In Chap. 8, Kleijnen reviews the design and analysis of simulation experiments under various goals, including validation, prediction, sensitivity analysis, optimization (possibly robust), and risk or uncertainty analysis through metamodels. Sanchez and Sanchez further illustrate how data can be "farmed" to support decision-making in Chap. 9.

Simulation is often used as a tool as well as an objective of operations research methods. In Chap. 10, Ryzhov and Chen provide a comprehensive overview of Bayesian belief models to support simulation-based decision-making. In Chap. 11, Zhou and Wu highlight how input model uncertainty impacts simulation optimization and propose approaches to quantify and control the impact of input model risk on optimal decision-making. In Chap. 12, Hunter and Nelson discuss the deployment of valid ranking and selection algorithms in distributed cloud computing environments.

Modeling and simulation have always been applied in systems that are mission critical where alternatives must be carefully evaluated to avoid costly mistakes. In Chap. 13, Hill and Tolk review the history of military applications while Fowler and Mönch, in Chap. 14, discuss the applications of simulation in semiconductor manufacturing. An emerging application in social and behavioral modeling through agent-based simulation is discussed by Macal and Kaligotla in Chap. 15.

Over its 50-year history, the Winter Simulation Conference has made significant contributions to the broader M&S community. In Chap. 16, Mustafee and Fishwick process 8499 records over a 36-year horizon (from 1981 to 2016) to provide a big data perspective on the indisputable impact of the Conference.

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. The authors affiliations are provided for identification purposes only and are not intended to convey or imply concurrence with, or support for, the positions, opinions or viewpoints expressed by the authors.

References

Goldsman D, Nance RE, Wilson JR (2009) A Brief History of Simulation from 1777 to 1981. In: Winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 310–313

Goldsman D, Nance RE, Wilson JR (2010) A brief history of simulation revisited. In: Proceedings of the Winter Simulation Conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 567–574

Nance RE (2013) Simulation and software through 50 years. In: Titans Talk presented at the 2013 Winter Simulations Conference (WSC), Washington, DC, USA

Reitman J (2008) The origin of the winter simulation conferences. IEEE Ann Hist Comput 30 (4):98–102

Wilson JR (1996) Winter simulation conference: An inside look at the premier forum on simulation practice and theory. OR/MS Today 23, no. 4

Chapter 2
Model Is a Verb

Lee Schruben

Abstract This chapter is an update of a 2011 Winter Simulation Conference Titans speech with the same title. That talk fortunately was not recorded and never appeared in print. However, it provoked numerous discussions ranging from good-natured ridicule to thoughtful debate. The proposition of that talk and this chapter is that the *process* of simulation modeling can be as valuable as the resulting computer model itself. A focus on modeling, the verb, has practical consequences on the way we conduct and what we expect from simulation projects. This affects how we teach, practice, and conduct research on simulation.

2.1 Introduction

The word, model, can be a noun, two types of verbs, or even an adjective. As a noun: a model is one system used to study another. Models are how we structure our thinking and communicate our thoughts to others in every aspect of life (language is our model for human experience). In simulation, a model is a computer program created to study how we believe a system is or should be.

The verb, model, in simulation is *transitive* (meaning it has an object) usually called the real-world system. Since the system being studied might not yet exist, it is referred to here as the object system. Simulation is a particularly explicit form of modeling since, ultimately, we must communicate with each other through a computer. Learning to simulate is a valuable general thinking and communication skill.

When I ask colleagues about their recent simulation projects, they invariably tell me about a computer program. This usually involves showing me an animation or plots of some experiments run with their code, or even their code itself. My question was intended to be about *why* and *how* they modeled some system, and

L. Schruben (✉)
IEOR, University of California, 4131 Etcheverry Hall, Berkeley
CA 94720-1777, USA
e-mail: schruben@ieor.berkeley.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_2

what they learned. In particular how they structured their concept of the system before attempting to explain it to a computer. Simulation models are fascinating but can be intellectually fatal distractions from a deeper understanding of the modeling process.

People with a limited exposure to discrete event simulation might think of it as simply another type of computer programming. By viewing simulation as a highly disciplined, structured method of thinking broadens this perception about simulation. Communicating different system designs, operations, objectives, and perspectives become the focus of simulation rather than creating and running computer codes.

Background for this article is the philosophy in *The Art of Simulation* by Tocher (1967) and aphorisms by two famous statisticians:

"*All models are wrong, but some are useful.*"

- G. E. P. Box: (Box 1979)

and

"*The best time to design an experiment is after you've done it.*"

-Sir R. A. Fisher: as quoted in (Box 1993)

These two quotes will be updated for computer simulation.

2.2 The Steps in a Simulation Study

The textbook requisite *Steps in a Simulation Study* change substantially when describing the creation of a simulation model (noun), versus describing the activity of modeling (verb). Two flowcharts on the steps in conducting a simulation study from Banks et al. (2010) and Law (2014) are given in Figs. 2.1 and 2.2.

Both figures partition a simulation study into a sequence of three non-recurring main steps:

1. Collecting data and coding a model;
2. Experimenting; and
3. Implementing the study results.

The model (noun) is at the center of both studies in Figs. 2.1 and 2.2. The implication is that once the model is "finished", it is not revisited.

Simulation modeling (the verb) as depicted in Fig. 2.3 is a continuous interaction of learning and communication. Figure 2.3 evolved from decades of simulation consulting projects in a wide range of areas including agriculture and food production, animal and human healthcare, computer architecture and communications, epidemic control, construction, pharmaceuticals, banking, recreation, entertainment, evolutionary genetics (and unintentionally, religion), and hi-tech and low-tech production.

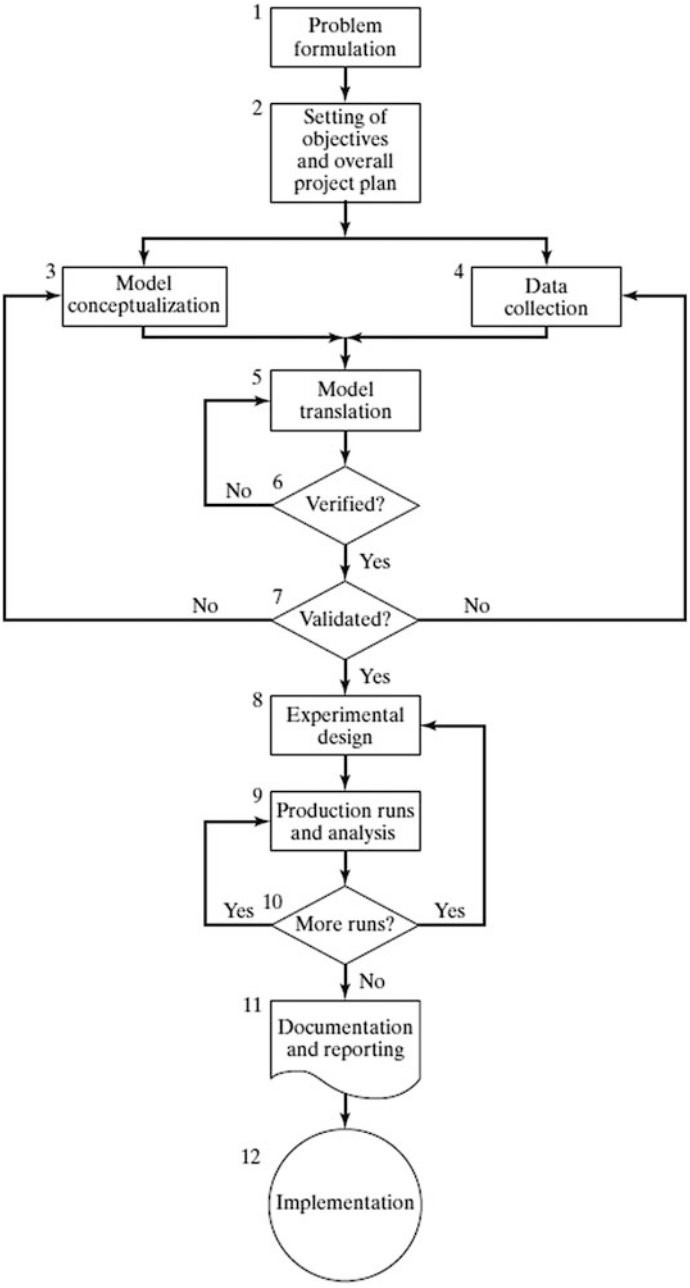


Fig. 2.1 Model (the noun) according to Banks et al. (2010)

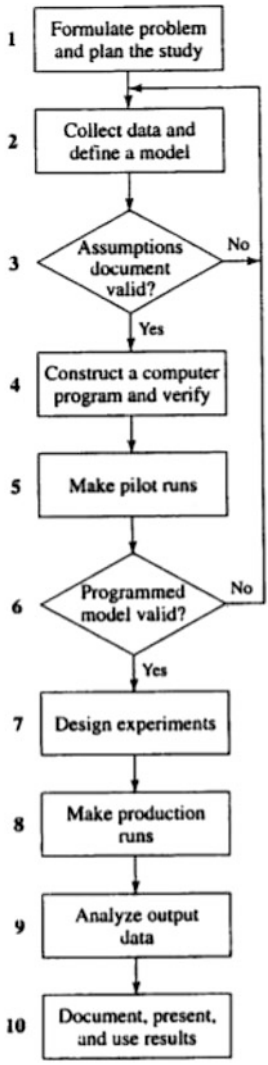


Fig. 2.2 Model (the noun) according to Law (2014)

The process shown in Fig. 2.3 has remained relatively stable for years.

Figures 2.1 and 2.2 present dramatically different views of simulation. The key difference is that the modeling process in Fig. 2.2 is iterative, comprised of two main repeated loops: *learning* about the object system and *communicating* using a simulation model. The model is a means to an end, but by no means an end.

In Fig. 2.3, a simulation project looks like it "Starts" but is never "Finished" (a lucrative business model for simulation consulting!). Actually, the project finishes whenever the test diamond nexus of the learning and communicating loops fails—

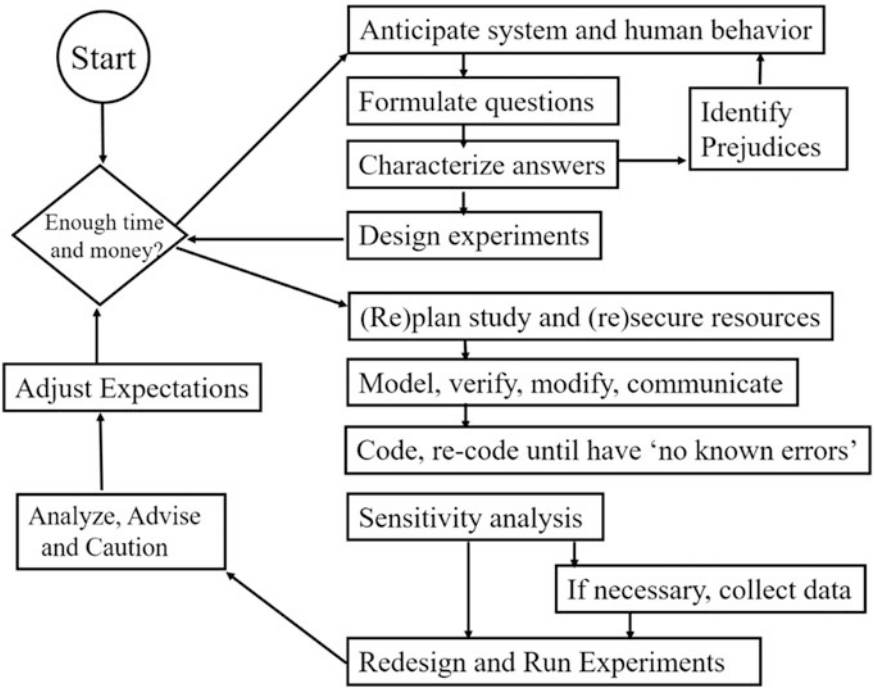


Fig. 2.3 Model (the verb) within modeling processes

the "false" branch from the test is not shown. The simulation project ends when there is not enough time and money to meet the adjusted expectations.[1]

The upper *learning cycle* in Fig. 2.3 includes anticipating behavior, formulating questions, characterizing the answers, identifying prejudices, and designing experiments. Formulating questions requires recognizing a well-posed question. For example, the simple question commonly answered by queueing and inventory theory: "what if demand doubles?" is too vague for a real system.[2] Demand doubling might happen if there are twice as many customers; the same number of customers each ordering twice as much; or in an infinite variety of other ways—all

[1] An aside from the WSC talk: The fundamental elements in life are time, money, and energy. Keeping these in balance is critical to happiness. If energy and time in youth are spent wisely, there might be enough money for later in life. There is usually an exchange rate from time to money (a salary). Money can sometimes be used to buy time; more precisely, someone else's time to save some of your own. There is never enough time so recognizing when you have enough money (for the time being) is critical. Energy comes in two complementary flavors; physical and emotional. Care for both or your time has no value.

[2] To the dismay of my colleagues in queueing theory, I have not found anyone who cared about average performance or stable queues. Most people are interested in extreme performance during periods of instability (like rush hours).

having different impacts on performance and profit. Each of these "demands" also might double half the time and halve the other half.

Formulating questions and characterizing answers iterate hand in hand in practice. The step of presenting a mock-up of possible "answers" (charts, tables, numbers, further questions) usually results in refining the questions. This can help avoid miscommunication in presenting the analysis when looping back to adjust expectations.

Identify prejudices early in a simulation project is important to success when the stakeholders disagree on key issues. Which is usually the case. If the stakeholders agreed on everything, they probably would not have commissioned a simulation study to provide an "*objective*" analysis. Knowing early in a study *who* is advocating *what* is immensely helpful in adjusting their expectations.

Contrary to the advice in Fig. 2.1, starting to design experiments before coding a simulation ensures a model that can run them. (Integrating models with experiments is discussed in Sect. 2.5.)

The lower *communications cycle* in Fig. 2.3, where a simulation model is developed, is similar to Fig. 2.1 with two notable exceptions. Like all software, a simulation's validity should never be certified or guaranteed. The best that can be achieved is the transient state of code with *no known errors*. Also, note that simulation input data is not necessarily collected. (More about data in the next section.)

2.3 Data

In Fig. 2.3, data is collected in the rare cases it might be necessary, and then only after a sensitivity analysis to see what data might actually matter. Lack of data is a common excuse for the failure of a simulation project. Skepticism about data's value in simulation is warranted. A simulation study is expensive and probably not undertaken unless a major change is being considered. Furthermore, the change is expected to have a significant impact on the system or its surroundings.

Object system data is available only if the system is operational. Most operational systems that are important enough to merit a simulation study have engineers whose jobs are to improve the system. Typically, they know the system's bottlenecks and may have already fixed these before the simulation discovers them. The object system is likely changing while its simulation model is being developed. The basic assumption that system data is stationary is always suspect.

Furthermore, even if the object system's data does not change during the study, it cannot be assumed that it will not change once the simulation study's results are implemented. If it assumed that a system or its surroundings will not respond when it is changed, why bother changing it?[3] The basic fallacy is assuming an open

[3]There is currently a research emphasis on simulation input uncertainty. Tools are needed to assess if the reactions to a change would make it a bad idea.

system is closed. Of course, some of the object system data might be unaffected by the change, but the system is likely to insensitive to such data.

In (Schruben 1991) five "Dastardlay D's of Data" were identified—a mnemonic in homage to the movie, Dodgeball—and constrain the length of this section. This list has been expanded to 16 problems with simulation input data. Data is:

1. *Dated*: Data is often pronounced "daytah" because it is dated. Live system data (as distinct from laboratory data) is probably outdated before it was recorded.
2. *Distorted*: Common examples are material move times that exclude (un)loading times; demand data excluding potential backorders.
3. *Dependent*: People like short lines better than long ones.
4. *Dynamic*: Friday nights in an Emergency Room are not like Tuesday mornings.
5. *Discrete*: Search: "discretization bias"
6. *Discouraging*: Studies find that flying is safer than driving and driving is safer than flying. Don't go anywhere if you can't walk!
7. *Distracting*: Search: "Prosecutor's Fallacy" and "Simpson's Paradox"
8. *Damaged*: Data might be collected (when?), recorded, (how?), aggregated, transcribed, and translated: a lot of possibilities for error.
9. *Deleted*: Used for cross-validation, audited, trimmed, censored, etc.
10. *Doctored*: Well-intentioned people attempt to clean it up.
11. *Deluged*: Too little information in too many numbers.
12. *Dogmatic*: My alternative facts are different from yours (see Fox News).
13. *Dangerous*: Racial or religious profiling, NSA call data, DNA data banks, epidemiology, personality testing, self-driving cars.
14. *Disclosed*: Voluntarily, by a self-selected population.
15. *Distorted*: Ratios, different scales, bounded by performance targets,
16. *Deceptive*: Any of the above D's might be intentional.

2.4 Modeling

There are many simulation modeling methodologies that provide system structure. These include Petri nets, event relationship graphs, process-interaction (transient flow) diagrams, activity cycles, etc. If the simulation modeler only knows one method (usually some commercial software), they cannot recognize when it might not be appropriate. If they know two methods, they can only see differences (there is no point of reference for "close"). Only when they know three methodologies can they see similarities. One must know about three different things to see similarities.[4] This is not only true about simulation modeling methodology but about life in general (politics, religion, cultures, parenting, etc.).

[4]The author teaches at least three different modeling paradigms in his courses loosely following (Choi and Kang 2013).

Modeling done in relative isolation from the object system (by, say, an outside consulting team) is loaded with pitfalls. A good way to reduce miscommunication is by using an interactive protocol (search for "Schruben–Turing Test" in (Kleijnen 1995) or (Robinson 2014).) Here simulation data and object system data are recorded in formats familiar to system subject matter experts (SME's). The outputs are shuffled and the SMEs are asked to identify which is real and which is simulated. If this is done early in a study, the SMEs can easily detect the simulated data. This is the point. Two miracles happen: 1. SMEs who were not previously interested in the simulation study are suddenly involved (they are in their element and having fun!) 2. The modelers learn how the SMEs can tell the real from the simulated data. If the difference is thought to be relevant, the model is enhanced. Another round is played and the increasingly interested SMEs have more and more difficulty identifying the real data from the simulated data. A simulation that incorporates SME's suggestions eventually becomes the SME's model! The model is then a bi-directional communication channel.[5]

For simulation (the verb) to be useful, it is not necessary that everyone agree the model is valid. If any two people have exactly the same "right" model, they cannot learn anything from each other (again true of life in general). Their models must be different ("wrong") to be of any value to each other. When simulation modeling is viewed as learning and communicating as in Fig. 2.2, Box's aphorism can be extended: "*All models are wrong, but some are useful,* ***because they are wrong***."

2.5 Experimenting

Experimenting with simulations is fundamentally different from other types of experiments. Simulation models have the powerful advantages of full observability and control. Anything can be observed and changed *while a simulation is running*. Neither observability nor control has yet to be exploited in commercial simulation languages. Indeed, simulation experimentation has been hampered by the view that models and experiments belong in different *frames*. Simulation languages have made spectacular advances in animation, but at the expense of experimental flexibility and analytical power.

Simulation textbooks typically describe building models, running experiments, analyzing outputs, and implementing results as distinct activities in a simulation project as in Fig. 2.1. However, simulation models can be tightly integrated with simulation experiments for more efficient analysis (Schruben 2010) For example, an array of simulation models running concurrently have demonstrated orders of magnitude greater analytical efficiency for a range of simulation optimization

[5]There are statistical measures to assess when SMEs are guessing such as the entropy in a bi-lateral communication channel (Schruben 1980). These tests are less important than the positive interaction that develops between SMEs and modelers during the protocol.

algorithms. These include randomized search, directional search, pattern search, and agent-based particle swarm optimization (Schruben 2013). In a simulation experiment, anything can be observed and changed at any time while the models are running (even changing time scales). A simulation model is not a "black box" at least to its creator, and treating it like one is a huge handicap. Fisher's quote can be updated to include simulation experiments as "*The best time to design an experiment is* ***while*** *you're running it.*"

Acknowledgements The author feels compelled to separate fact from opinion in a paper like this: it is all opinion. The view of simulation as a continuous learning and communication process has been influenced by all of the author's students, consulting clients, and the many colleagues he has had the pleasure of knowing for over 40 years.

References

Banks J, Carson JS, Nelson BL, Nicol DM (2010) Discrete-event systems simulation, 5th edn. Pearson

Box GEP (1979) Robustness in the strategy of scientific model building. In: Robustness in statistics. Academic Press, pp 201–236

Box GEP (1993) George's column. Qual Eng 5(2):321–330

Choi BK, Kang D (2013) Modeling and simulation of discrete event systems. Wiley

Kleijnen JPC (1995) Verification and validation of simulation models. Eur J Oper Res 82:145–162

Law AM (2014) Simulation modeling and analysis, 5th edn. McGraw-Hill

Robinson S (2014) Simulation: the practice of model development and use. Palgrave Macmillan

Schruben L (1980) Establishing the credibility of simulations. Simulation 34(3):101–105

Schruben L (1991) Sigma: a graphical simulation system. Academic Press

Schruben L (2010) Simulation modeling for analysis. ACM Trans Model Comput Simul 20(1):1–21

Schruben L (2013) Simulation modeling, experimenting, analysis, and implementation. In: Proceedings of the winter simulation conference, Washington, D.C

Tocher KD (1967) The art of simulation. English Universities Press

Chapter 3
Model Calibration

Jie Xu

Abstract Simulation model calibration refers to the iterative process of comparing the outputs of a simulation model with the observed quantities in the real system, and making changes to model input parameters accordingly to achieve an acceptable level of agreement between the simulation model and the real system. While calibration in a broader context may involve structural changes to the simulation model, this chapter focuses on the calibration of simulation model parameters that cannot be accurately estimated or specified for various reasons. When the simulation is time-consuming, has significant noise, and/or has a large number of parameters to calibrate, automatic and efficient calibration methods are critical to the success of any simulation-based analysis and optimization. This chapter discusses two main categories of general calibration methods: (1) direct calibration methods that search for the optimal calibration parameter that minimizes the difference between real system observations and simulation model outputs; and (2) Bayesian calibration methods that combine real system observations with prior knowledge to obtain a posterior distribution on the calibration parameters.

3.1 Introduction

Computer simulation models provide predictions of the outcome/behavior of complex systems/processes that are intractable to other forms of analysis. Because of the complexity of the systems/processes being modeled, simulation models often have a large number of input parameters. Generally speaking, there are two categories of input parameters: *variable input parameters* and *calibration input parameters* (Kennedy and O'Hagan 2001). Variable input parameters are assumed to be *known*, possibly with parametric uncertainty, e.g., when they are estimated from data. They may be *varied* during the calibration process, and describe alternative controls/designs for which the simulation model is used to predict the performance

J. Xu (✉)
George Mason University, 4400 University Drive, MS 4A6, Fairfax, VA 22030, USA
e-mail: jxu13@gmu.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_3

of the system. Calibration input parameters are assumed to be *unknown*, but remain *fixed* for all the alternative controls/designs for which the model is calibrated to simulate and predict. Most calibration input parameters are typically *physical* parameters that define the environment/situation within which the model simulates. There may also be *tuning* parameters that only exist in the simulation model and are used to control the behavior of the model. The output of the simulation model is a function of both variable and calibration input parameters.

The prediction accuracy of the simulation model depends on proper setting of the values of the calibration input parameters. Model calibration is the process to determine the values of the calibration input parameters such that the output of the computer simulation model provides the best fit to observed data from the real system. In the literature, there are two main classes of methods for model calibration. The first class of methods aim to minimize the difference between model outputs and the observations of real systems/processes. Typically, key system metrics are identified and an objective function is defined to measure the difference between the model output and the observed quantities. The calibration parameters are then iteratively adjusted manually or automatically by an algorithm to minimize the objective function until the stopping criteria are met. Such an approach has been widely used in both literature and industrial practice. We refer to this class of methods as *direct calibration* because it generally works directly with simulation model output and often uses direct search methods to adjust the calibration parameters, treating the simulation model as a black-box function. The second class of methods is *Bayesian calibration*. Bayesian calibration methods make use of models to describe how calibration parameters affect model output. The model can be a general one such as a Gaussian progress (Kennedy and O'Hagan 2001), or a problem-specific model such as the stochastic user equilibrium (SUE) model in a dynamic traffic assignment (DTA) simulator in Flötteröd et al. (2011).

The direct and Bayesian calibration methods each have their own merits and limitations. Direct calibration methods are intuitive, do not assume any knowledge on the model and/or calibration parameters, and are reasonably easy to understand and use. As a result, direct calibration methods have been widely used in practice across a large range of disciplines, albeit frequently in somewhat ad hoc ways and sometimes with manual adjustments of the calibration parameters. The recent advances in efficient simulation optimization algorithms and the access to low-cost massive computing power make direct calibration even more attractive. These algorithms provide an automatic and efficient way to calibrate a model, especially when there are a large number of parameters to calibrate and simulations are time-consuming and/or noisy. The popularity of direct calibration methods is also evident from the sizable number of proceedings papers in the past 10 years of the Winter Simulation Conference (WSC). Manual and automatic direct calibration methods were applied to calibrate agent-based (Johnson et al. 2009; Latek et al. 2013; Shi and Brooks 2007), systems dynamics (Aral et al. 2014), discrete-event simulation models (Henclewood et al. 2012; Vock et al. 2014), and general probability models used in Monte Carlo simulations (Matus et al. 2016).

Direct calibration is a "plug-in" approach. Once the calibration input parameters have been determined, they remain fixed and is taken as the "true value" in the ensuing simulation experiments and analyses. The uncertainty in the calibrated parameters and its impact on model predictions would then be left unaddressed. While central limit theorems can be established if convergent simulation optimization algorithms are used in calibration, these asymptotic properties are of very limited practical value. Sensitivity analysis is helpful, but does not provide a complete description of the uncertainty (Shi and Brooks 2007; Yuan et al. 2013). Another difficult issue is how to incorporate prior knowledge on calibration parameters. Prior knowledge might be used in the form of the initial values in the calibration process. However, this is a rather rigid way to use prior knowledge as it does not convey the confidence in the prior knowledge and the calibration process is basically free to move away from the initial value.

Bayesian calibration methods have merits and limitations that are complementary to direct calibration in many ways. Through the postulation of a prior distribution on calibration parameters, it facilitates the inclusion of domain knowledge into the calibration process, reflecting the confidence in the domain knowledge and updating it with evidence from data. The posterior distribution fully describes the uncertainty in the calibrated parameters and can then be used to calculate various metrics to quantify the uncertainty in the model predictions.

However, Bayesian calibration methods also come with their own limitations. Bayesian calibration relies on the use of a model to describe the relationship between calibration parameters and the model output. Unless one knows the precise structure of the model as in the case of the demand calibration of the DTA simulator in Flötteröd et al. (2011), the calibration process is subject to the additional model error. The model error may lead to significant biases in the estimate of calibration parameters and consequently large prediction errors. Second, the majority of Bayesian calibration methods make use of the Gaussian process as the model to relate calibration parameters to the model output. It is well-known that while Gaussian process is a very flexible framework, it does not scale up to the dimension of the calibration input parameter vector. Further, while homogeneous observation errors can be addressed under the Gaussian process framework, many stochastic simulation models have outputs with considerably different variances for different variable input parameter values. In addition, as shown in Han et al. (2009), Bayesian calibration may lead to large prediction errors without proper adjustments when applied to tuning parameters. Finally, from a practical point of view, Bayesian calibration methods require a considerable level of technical knowledge on Bayesian inference beyond the reach of many simulation modelers and analysts. In recent years, Bayesian calibration methods have made their ways to WSC proceedings (Frazier et al. 2009; Yuan and Ng 2013a). However, its widespread use by modelers remains to be seen.

The rest of the chapter is organized as follows. We first summarize the main notations and the models to describe the calibration process in Sect. 3.2. In Sect. 3.3, we present two cases of direct calibration using simulation optimization. Section 3.4 summarizes Bayesian calibration methods. A summary and a brief discussion on future research directions are given in Sect. 3.5.

3.2 Overview

We use $\mathbf{x} = (x_1, \ldots, x_{d_1})^T$ to denote the vector of variable input parameters of dimension d_1, and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{d_2})^T$ to denote the vector of calibration input parameters of dimension d_2. For stochastic simulations, we use ξ to represent the randomness inherent in the simulation. Formally, we describe the mapping of the variable and calibration input parameters to the expected model output as follows:

$$y(\mathbf{x}, \boldsymbol{\theta}) = \mathbb{E}\left[\eta(\mathbf{x}, \boldsymbol{\theta}; \xi)\right], \tag{3.1}$$

where the expectation is taken with respect to the probability measure for ξ. In general, we would not be able to directly observe y. Instead, we assume that we can run independent and identically distributed (IID) simulation replications and the estimate of the jth replication satisfies

$$y_j(\mathbf{x}, \boldsymbol{\theta}) = y(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon_j(\mathbf{x}, \boldsymbol{\theta}), \tag{3.2}$$

where the noise $\varepsilon_j(\mathbf{x}, \boldsymbol{\theta})$ is IID $N(0, \sigma^2(\mathbf{x}, \boldsymbol{\theta}))$. We can then use the sample average $\bar{y}$ to estimate y. We assume that the simulation model output y is scalar for simplicity throughout this chapter. When the output is a vector, for direct calibration, we can either take a multi-objective optimization approach, or use the weighted sum of the different elements in the output vector as the scalar output of the model to calibrate against. For Bayesian calibration under the Gaussian process framework proposed in Kennedy and O'Hagan (2001), the vector output can be handled similarly in the calibration process as in the scalar output case.

Model calibration is an iterative process. At the beginning of an iteration, we assume that we have the following information:

- The m observations of the real system response $\mathbf{z} = \{z_1, \ldots, z_m\}^T$ for m variable inputs $D_1 = \{\mathbf{x}_1^r, \ldots, \mathbf{x}_m^r\}$.
- The sample averages and sample variances of the simulation model outputs $\bar{\mathbf{y}} = \{\bar{y}_1, \ldots, \bar{y}_n\}^T$, $\mathbf{S}^2 = \{S_1^2, \ldots, S_n^2\}^T$ for n given variable and calibration inputs $D_2 = \{(\mathbf{x}_1^c, \boldsymbol{\theta}_1), \ldots, (\mathbf{x}_n^c, \boldsymbol{\theta}_n)\}$.

Let $\zeta(\mathbf{x})$ be the true response of the real system given input $\mathbf{x}$. For stochastic systems/processes, the response is also stochastic and we interpret $\zeta(\cdot)$ as a statistic of the stochastic response, e.g., mean, median, or quantile. Equation (3.3) describes the relationship between the true response of the real system and the observation

$$z_i = \zeta(\mathbf{x}_i^r) + e_i, \quad i = 1, \ldots, m. \tag{3.3}$$

In the above equation, e_i represents either the observation error or the inherent randomness in the response when the system is stochastic. It is possible to obtain multiple observations for the same variable input $\mathbf{x}_i, i = 1, \ldots, m$. Following most of

the literature, we assume that there is only one observation for each $\mathbf{x}_i, i = 1, \ldots, m$ unless explicitly indicated otherwise.

The true response of the real system and the simulation model output is modeled as

$$\zeta(\mathbf{x}) = \rho y(\mathbf{x}, \boldsymbol{\theta}^*) + \delta(\mathbf{x}). \tag{3.4}$$

Here $\boldsymbol{\theta}^*$ is the unknown true calibration input parameters and ρ is the scaling coefficient for the simulation model. The term $\delta(\mathbf{x})$ represents the bias of the simulation model, which is also referred to as model error or model inadequacy. Equation (3.4) provides just one way of modeling the relationship between the simulation model output and the true system response. This model was made popular by the work of Kennedy and O'Hagan (2001) and has been adopted in many subsequent Bayesian calibration studies.

It is important to point out that in model calibration, the "true" calibration input parameters $\boldsymbol{\theta}^*$ is the $\boldsymbol{\theta}$ that best fits the observations of the real system $\mathbf{z}$. For calibration input parameters that are physical quantities existing in the real system (see Sect. 3.4.2 for a discussion on the differences between physical parameters and tuning parameters in model calibration), $\boldsymbol{\theta}$ may not be the true values of these physical parameters.

An important difference between direct and Bayesian calibration is how "best-fitting" is measured. In direct calibration, we directly measure the difference between observations of the real system $\mathbf{z}$ and the model outputs on the same set of m variable inputs $\mathbf{x}_1^r, \ldots, \mathbf{x}_m^r$. We thus drop the superscript to simplify the notation in the rest of the chapter when we discuss direct calibration. We define an objective function $L(\boldsymbol{\theta}; \mathbf{z}, \mathbf{y}(\boldsymbol{\theta}))$ to measure the discrepancy between $\mathbf{z}$ and $\mathbf{y}(\boldsymbol{\theta})$ for a given $\boldsymbol{\theta}$. While we do not make the dependence of $\mathbf{y}$ on the variable inputs $\mathbf{x}_1, \ldots, \mathbf{x}_m$ explicit, it is understood that the outputs are with respect to these variable inputs. Direct model calibration is formulated as the following optimization problem:

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}; \mathbf{z}, \mathbf{y}(\boldsymbol{\theta})). \tag{3.5}$$

There are several choices of the functional form of $L(\cdot)$. We will review two calibration studies using two common choices in Sect. 3.3. The first example reviewed in Sect. 3.3.1 takes the form of sum of squared errors (SSE) (Yuan et al. 2013)

$$L(\boldsymbol{\theta}; \mathbf{z}, \mathbf{y}(\boldsymbol{\theta})) = \sum_{i=1}^{m} \left(z_i - y_i(\boldsymbol{\theta})\right)^2. \tag{3.6}$$

In Sect. 3.3.2, the model calibration problem uses the absolute difference between $\mathbf{z}$ and $\mathbf{y}(\boldsymbol{\theta})$

$$L(\boldsymbol{\theta}; \mathbf{z}, \mathbf{y}(\boldsymbol{\theta})) = \sum_{i=1}^{m} \left|z_i - y_i(\boldsymbol{\theta})\right|. \tag{3.7}$$

For stochastic simulations, we have to estimate $\mathbf{y}(\boldsymbol{\theta})$ and $L(\boldsymbol{\theta}; \mathbf{z}, \mathbf{y}(\boldsymbol{\theta}))$ by $\bar{\mathbf{y}}$. In this case, problem (3.5) becomes a simulation optimization, or optimization via simulation, problem. For real-world applications, the simulation model is often too complex to yield any structural information. When simulation runs are time-consuming and/or noise is significant, solving (3.5) is challenging and requires efficient simulation optimization algorithms that also converge to optimal solutions in the presence of simulation noise. In Sect. 3.3, we review two model calibration studies that used different simulation optimization algorithms to solve problem (3.5). For stochastic systems/processes, z_i may be a vector of multiple observations for $\mathbf{x}_i$. Model calibration then may involve comparing the probability distribution of the system response with the simulation model output distribution using statistical tests (Henderson et al. 2009) or statistical disparity measures (Vidyashankar and Xu 2015).

Compared to the intuitive formulation such as (3.6) and (3.7), Bayesian calibration is a much more complicated process. Bayesian calibration generally comprises the following steps

1. Specify a model describing the relationship between $\mathbf{z}, \zeta(\cdot), \eta(\cdot, \cdot)$, and $\mathbf{y}$. For instance, Kennedy and O'Hagan proposed to use (3.3) and (3.4), with an additional condition that $\eta(\cdot, \cdot)$ and $\delta(\cdot)$ are independent.
2. Postulate statistical surrogate models about $\zeta(\cdot), \eta(\cdot, \cdot)$, e.g., Gaussian processes, and specify the prior distributions of hyper-parameters.
3. Run simulations to obtain the data for the initial calibration iteration.
4. Estimate the hyper-parameters of the surrogate models and calculate the posterior distribution for $\boldsymbol{\theta}$.
5. If the stopping criteria are not met, determine the new experiment design point(s) to run simulations, and go back to Step 3.

In Sect. 3.4, we review the details in the above steps under the Gaussian process framework for Bayesian calibration proposed in Kennedy and O'Hagan (2001), which was extended to address homogeneous simulation noise in Yuan and Ng (2015).

3.3 Direct Calibration

Direct calibration is a simulation optimization problem as formulated in (3.5). Traditionally, simulation modelers and analysts attempted to solve (3.5) in rather ad hoc ways. It is a common practice to compare a small set of manually chosen calibration input parameters (Johnson et al. 2009; Henclewood et al. 2012; Aral et al. 2014), which are often defined on coarse grids spanning user-specified ranges of $\boldsymbol{\theta}$. Applications of meta-heuristics designed for deterministic black-box optimization problems, e.g., evolutionary strategies (Latek et al. 2013), genetic algorithms (Vock et al. 2014), Nelder–Mead simplex algorithm (Shi and Brooks 2007), were also frequently used to solve (3.5). The use of these meta-heuristics helped (partially) automate the model calibration process and allows the modeler/analyst to examine many

more values of $\boldsymbol{\theta}$ and obtain better calibration outcome. However, the applications of these meta-heuristics are fundamentally flawed as they do not fully account for the stochastic nature of the problem, and thus may often converge to false "optimal" calibration parameters. Readers can look for more information on the risks of applying deterministic optimization algorithms to solve stochastic simulation optimization problems in Xu et al. (2010, 2015), Hong et al. (2015), Fu (2015b). In this section, we review two studies that use simulation optimization algorithms to perform direct calibration in an automatic, efficient, and correct way. In addition to using intuitive objective functions like the ones described in (3.6) and (3.7), it is also straightforward to use other metrics that help compare the difference between the distributions of observed real system's responses and simulation model outputs (Henclewood et al. 2012; Vidyashankar and Xu 2015).

3.3.1 Direct Calibration Using Stochastic Approximation

When $\boldsymbol{\theta}$ is continuous and the objective function $L(\boldsymbol{\theta})$ is differentiable in $\boldsymbol{\theta}$, e.g., if we use SSE as in (3.6) and the model output y is differentiable in $\boldsymbol{\theta}$, well-known simulation optimization algorithms such as stochastic approximation (SA) (Kushner and Yin 2003) can be applied. Let $\mathbf{g}(\boldsymbol{\theta}) = \partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ be the gradient. If we know $\mathbf{g}(\cdot)$, we can solve problem (3.5) using Newton's method. However, in model calibration, most of the time we do not know $\mathbf{g}(\cdot)$ and instead need to estimate it. When simulations are stochastic, the estimate of $\mathbf{g}(\boldsymbol{\theta})$ is also stochastic and known as the stochastic gradient (Fu 2015a). Denote an estimate of the stochastic gradient as $\widehat{g}(\boldsymbol{\theta})$. Then SA searches for the optimal calibration input parameters through the following iterative equation

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k - a_k \widehat{\mathbf{g}}_k, \tag{3.8}$$

where k is the iteration counter and the deterministic gain sequence $\{a_k\}$ is chosen to satisfy $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$. In order for SA to converge, $\widehat{g}(\boldsymbol{\theta})$ should either be an unbiased estimator of $\mathbf{g}(\boldsymbol{\theta})$, or the bias goes to zero as $k \to \infty$. For more details on SA, readers can refer to Fu (2015a) and Kushner and Yin (2003). Except for problems where direct stochastic gradient estimates can be derived using methods such as infinitesimal perturbation analysis (IPA) (Fu 2015a), finite difference (FD) or simultaneous perturbation (SP) gradient estimators are often used in SA. The algorithms are consequently referred to as FDSA and SPSA. More information on stochastic gradient estimation can be found in Fu (2015a) and the references therein. The convergence conditions of SA applied to direct calibration were discussed in Yuan et al. (2013).

SA only converges to first order stationary points asymptotically. In model calibration, the objective function $L(\boldsymbol{\theta})$ may contain multiple local optima and when simulation runs are expensive, only a finite number of SA iterations would be executed. Therefore, in practice, multi-start is often recommended. Statistical ranking

and selection procedures (Xu et al. 2010) can then be used to select the best calibration parameter values found as the final result.

It is well-known that under some technical conditions, $\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*$ converges asymptotically to a normal random variable in distribution. Yuan et al. (2013) made use of the asymptotic distribution of $\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*$ to quantify the uncertainty in the estimate of $\boldsymbol{\theta}$ found by using FDSA and SPSA. Delta method can then be applied to quantify the uncertainty in the prediction made by the simulation model using the estimated calibration input.

Both FDSA and SPSA were used in Yuan et al. (2013) to calibrate a stochastic biological model simulating the mtDNA deletion accumulation in substantia nigra neurons and the death of neurons. The model can be used for studies on Parkinson's disease and was described in detail in Henderson et al. (2009). There are three parameters to be calibrated in the experiment: the mutation rate c_1, the degradation rate c_3, and the lethal threshold τ. Yuan et al. (2013) held $\tau = 0.962$ and applied FDSA and SPSA to calibrate $\theta_1 \doteq \log(c_1)$ and $\theta_2 \doteq \log(c_3)$. The variable input parameter in this example is the age of the patients, with $x_i = 19, 32, 42, 51, 56, 75, 81, 89$. The observed deletion accumulation z_i for each age x_i from the real process were collected using real-time polymerase chain reaction (RT-PCR) measurements. Ten simulation replications were run for each variable and calibration input and the sample averages were then taken as the simulation output $\bar{y}_i$ to compare with z_i.

FDSA/SPSA were started from four initial points. The choices of the initial points were based on the previous study in Henderson et al. (2009) that gave the 95% confidence intervals as $[-13.93, -6.87]$ for θ_1, and $[-4.53, -3.07]$ for θ_2. The combinations of the upper and lower limits of these two confidence intervals were chosen as the four initial points to start FDSA/SPSA. Ten runs of FDSA/SPSA were conducted from each initial point, with each run lasting 100 iterations. The objective function is SSE as given in (3.6). The mean of the ten runs from each initial point was taken as the parameter calibrated by FDSA/SPSA. In the experiment, FDSA/SPSA converged to similar results. Yuan et al. further validated the simulation model output using the calibrated parameters c_1 and c_3 to make predictions on the holdout samples of age 20, 44, 72, 77, and 91. The 95% confidence intervals that take into account both the uncertainty in the estimates of c_1 and c_3 and the noise in the stochastic simulations cover the observed measurements, and thus helped establish the validity of the calibration method and the calibrated simulation model.

3.3.2 Direct Calibration of a Drug Resistance Simulation Model in a Parasitology Study Using Discrete Optimization via Simulation

Frequent anthelmintic treatment has effectively controlled the pathogenic effects of strongyles living in the intestines of horses. However, drug resistance has been observed globally among the class of small strongyles (subfamily *Cyathostominae*).

The parasitology community has recommended alternative therapies that aim to reduce the intensity of anthelmintic treatment in an attempt to slow down the development of drug resistance. However, the reduction in anthelmintic treatment frequency and intensity has been found associated with the re-emergence of a highly pathogenic parasite *S. Vulgaris*, also known as the blood worm. There is a need to model the development of drug resistance in equine parasites and the re-emergence of *S. Vulgaris* to assist in the design of alternative sustainable anthelmintic regimens.

Xu et al. presented a stochastic simulation model in Xu et al. (2014), which makes use of a population genetics model and an epidemiological model to simulate the interactions among hosts, parasites, and therapy regimens. The conceptual diagram of the simulation is given in Fig. 3.1. The model was intended to be used to predict the development of drug resistance and re-emergence of *S. Vulgaris* under selective therapy, which is one of the most popular alternative anthelmintic therapy regimens. Instead of treating horses every 6 months, selective therapy stipulates that only horses with serious infections as determined by a specified fecal egg count (FEC) threshold should be treated.

Following common practice in the literature (Leathwick and Hosking 2009; Leathwick 2013; Xu et al. 2014) considered a simplified population genetics model with a single drug resistance gene **R** with two alleles. There are then three different genotypes: **SS**, **SR**, and **RR**. Here **S** represents the drug susceptible gene. It is assumed that worms of the **RR** genotype are fully drug resistant and worms of the **SR** genotype are partially drug resistant. The simulation model then keeps track of

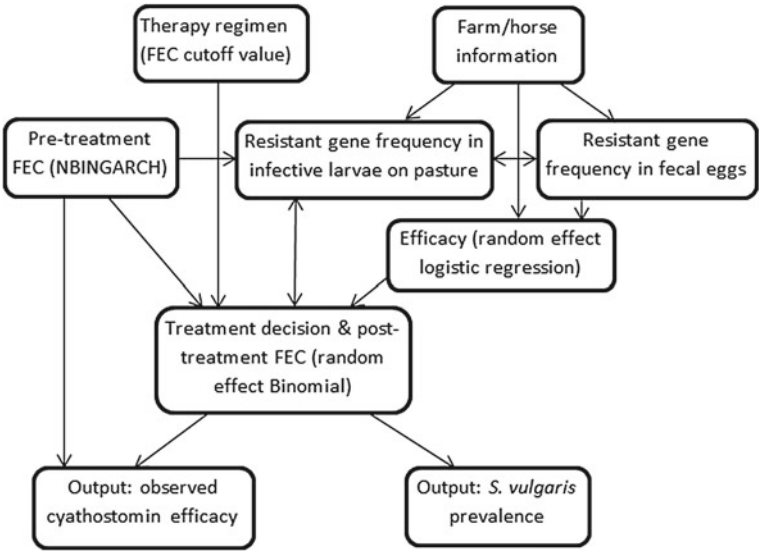


Fig. 3.1 Conceptual diagram of the simulation of the parasite-host-drug interaction (Xu et al. 2014)

the development of drug resistance via simulating the proportion of **R** genes as a function of time under drug selection pressure.

Let $F^0_{RR}, F^0_{SR}, F^0_{SS}$ be the fitness of these three genotypes during the evolution process when there is no anthelmintic treatment and thus drug selection pressure is zero. Let $F^1_{RR}, F^1_{SR}, F^1_{SS}$ be the fitness of these three genotypes under drug selection pressure. A common belief is the fitness of the resistant genotypes are lower than the susceptible genotypes, which is known as the biological cost of resistance (Leathwick 2013; Mackinnon 2005). Therefore, it is reasonable to assume that $0 < F^0_{RR} \leq F^0_{SR} < F^0_{SS} = 1$ under natural evolution, with $F^0_{SS} = 1$ being the benchmark fitness. For evolution with drug selection pressure, the fitness values were assumed to satisfy $0 < F^1_{SS} \leq F^1_{SR} < F^1_{RR} \leq 1$.

The simulation model has 36 input parameters, including the five fitness values of different genotypes described previously, and one variable input parameter that determines the FEC threshold that selective therapy uses to determine if a horse should receive anthelmintic treatment every 6 months. Xu et al. (2014) identified that these fitness values play a very important role in the development of drug resistance and thus should be properly set in simulation experiments. Unfortunately, there is no biological data set that can be used to estimate these fitness values. Furthermore, experiments to measure such fitness values would be both time and resources consuming, and estimates would also be highly noisy. Therefore, these fitness values were calibrated via direct calibration. Other the 30 input parameters were determined using a combination of reported values used in the literature, estimation by data from a 2008 Danish horse farm study (Nielsen et al. 2013), or expert opinions.

The consensus in the equine parasitology field is it took about $t_R = 30$ years to observed reduced drug efficacy on cyathostomins with a non-selective therapy regimen with horses treated every 2 months (Molento et al. 2012). In the calibration process, the objective is to calibrate $\boldsymbol{\theta} \doteq \{F^0_{RR}, F^0_{SR}, F^1_{RR}, F^1_{SR}, F^1_{SS}\}$ such that it took 180 treatments over a span of 30 years to observe the drug efficacy dropping below 95% for the first time according to the general equine parasitology context (Molento et al. 2012). In the model calibration process, each simulation ran for 40 simulated years with 240 treatments and the first drug resistance treatment was reported as an estimate of the simulation model output $y(\boldsymbol{\theta})$. If resistance is not observed within the 40 years simulated, the simulation model simply returns a large number. The calibration problem thus follows the formulation as described in (3.7)

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}) = \left| t_R - y(\boldsymbol{\theta}) \right|. \tag{3.9}$$

Notice here we only have one observation of the real system $t_R = 30$ years. The range $\boldsymbol{\Theta}$ was chosen based on previous drug resistance studies in the literature (Leathwick 2013) and satisfy $0.85 < F^0_{RR} < F^0_{SR} < F^0_{SS} = 1$, $0 < F^1_{SS} \leq 0.03$, $F^1_{SS} < F^1_{SR} \leq 0.15$, and $0.85 < F^1_{RR} \leq 1$. Without using an optimization-based direct calibration method, Leathwick (2013) only conducted "what-if" simulations with nine different combinations of fitness parameter values. In contrast, Xu et al. (2014) used a discrete optimization via simulation solver ISC (Xu et al. 2010, 2013; Hong

et al. 2010) to solve problem (3.9). While the fitness values are continuous, in reality, parasitologists would discretize the parameters. This is because the precision of any biological experiment to measure these parameters would be subject to substantial uncertainties and errors, making the precision to be a few digits at best. As a result, treating $\boldsymbol{\theta}$ as discrete-valued parameters is closer to reality.

The simulation budget was set to 2000 replications. ISC has the option to perform a final "clean-up" procedure to control δ, the error in the simulation estimate of the objective function value in (3.9). Xu et al. (2014) set $\delta = 1$, i.e., one would be able to distinguish with a probability of 0.95 the true parameter $\boldsymbol{\theta}^*$ from another parameter $\boldsymbol{\theta}$ if the difference in the drug resistance appearance time $|y(\boldsymbol{\theta}) - y(\boldsymbol{\theta}^*)| \geq 1$. The best $\boldsymbol{\theta}$ found by ISC was $\widehat{\boldsymbol{\theta}} = \{F^0_{SR} = 0.989, F^0_{RR} = 0.857, F^1_{SS} = 0.027, F^1_{0.045}, F^1_{RR} = 0.926\}$. The 95% t-confidence interval on $L(\widehat{\boldsymbol{\theta}})$ is $[8.66, 13.3]$. While ISC was not able to find a $\boldsymbol{\theta}$ that reduces $L(\cdot)$ to 0, it allowed one to systematically search through a very large parameter space $\boldsymbol{\Theta}$ with a very reasonable computation budget. The calibration outcome was thus deemed as of reasonably high quality.

3.4 Bayesian Calibration

Bayesian calibration uses a model to describe the relationship between the true system response $\zeta(\mathbf{x})$ and simulation model output $y(\mathbf{x}, \boldsymbol{\theta})$. The use of a model facilitates statistical inference and uncertainty quantification of $\boldsymbol{\theta}$. However, these merits come with limitations and challenges. The use of the model to relate $\zeta(\mathbf{x})$ and $y(\mathbf{x}, \boldsymbol{\theta})$ introduces additional modeling error. Estimation of the model parameters can be quite difficult and creates additional uncertainty. As shown in Sect. 3.4.2, if not properly handled, Bayesian calibration may perform quite poorly for tuning parameters. Handling heterogeneous simulation noise also remains a challenge. In Sect. 3.4.1, we first present the widely used Bayesian calibration framework using Gaussian process as the model between $\zeta(\mathbf{x})$ and $y(\mathbf{x}, \boldsymbol{\theta})$. We then review a Bayesian calibration approach for simultaneous calibration of physical and tuning parameters in Sect. 3.4.2.

3.4.1 A Gaussian Process-Based Bayesian Calibration Framework

In Kennedy and O'Hagan (2001), Kennedy and O'Hagan presented a Bayesian calibration framework using Gaussian process, which has been adopted in many later studies. The model takes the form of (3.4), and assumes that ρ is a constant. The prior information on $y(\cdot, \cdot)$ and $\delta(\cdot)$ is described by Gaussian processes:

$$y(\cdot, \cdot) \sim N\left[m_1(\cdot, \cdot), \sigma_1^2 \mathbf{R}_1\{(\cdot, \cdot), (\cdot, \cdot)\}\right], \quad \delta(\cdot) \sim N\left[m_2(\cdot), \sigma_2^2 \mathbf{R}_2(\cdot, \cdot)\right]. \tag{3.10}$$

The mean functions are assumed to have a linear form:

$$m_1(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{h}_1(\mathbf{x}, \boldsymbol{\theta})^T \boldsymbol{\beta}_1, \quad m_2(\mathbf{x}) = \mathbf{h}_2(\mathbf{x})^T \boldsymbol{\beta}_2, \tag{3.11}$$

where $\mathbf{h}_1(\cdot, \cdot)$ and $\mathbf{h}_2(\cdot)$ are vectors of p_1 and p_2 known functions respectively. The unknown coefficients $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are of dimension p_1 and p_2 and we denote the combined vector as $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T$. The correlation functions $\mathbf{R}_1$ and $\mathbf{R}_2$ are often postulated to be the Gaussian correlation function of the form

$$\mathbf{R}_1\{(\mathbf{x}_i, \boldsymbol{\theta}_i), (\mathbf{x}_j, \boldsymbol{\theta}_j)\} = \exp\left[-\Sigma_{l=1}^{d_1} \omega_{1_x,l}(x_{i,l} - x_{j,l})^2 - \Sigma_{l=1}^{d_2} \omega_{1_\theta,l}(\theta_{i,l} - \theta_{j,l})^2\right], \tag{3.12}$$

and

$$\mathbf{R}_2(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\Sigma_{l=1}^{d_1} \omega_{2,l}(x_{i,l} - x_{j,l})^2\right]. \tag{3.13}$$

We write $\boldsymbol{\omega}_1 = \{\omega_{1_x,1}, \ldots, \omega_{1_x,d_1}, \omega_{1_\theta,1}, \cdots, \omega_{1_\theta,d_2}\}^T$, $\boldsymbol{\omega}_2 = \{\omega_{2,1}, \ldots, \omega_{2,d_1}\}^T$ to denote all the unknown parameters in the correlation functions. Let $\boldsymbol{\omega} = \{\boldsymbol{\omega}_1^T, \boldsymbol{\omega}_2^T\}^T$. It is further assumed in Kennedy and O'Hagan (2001), Yuan and Ng (2015) that the stochastic simulation noise $\varepsilon_j(\mathbf{x}, \boldsymbol{\theta})$ in Eq. (3.2) is IID $N(0, \sigma_\varepsilon^2)$, and the observation noise for the real system response e_i in Eq. (3.3) is IID $N(0, \sigma_e^2)$. Further, $y(\cdot, \cdot)$, $\delta(\cdot)$, ε, and e are assumed to be mutually independent. Write $\boldsymbol{\sigma}^2 = \{\sigma_1^2, \sigma_2^2, \sigma_\varepsilon^2, \sigma_e^2\}$. Then the parameters to be estimated include $\boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}$.

Kennedy and O'Hagan (2001) proposed a hierarchical model to represent the prior information on $\boldsymbol{\beta}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \sigma_1^2, \sigma_2^2, \sigma_\varepsilon^2, \sigma_e^2$. Based on the mutual independence assumptions, the prior distribution can then be written as

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}) = p(\boldsymbol{\theta}) p(\boldsymbol{\beta}_1, \sigma_1^2) p(\boldsymbol{\beta}_2, \sigma_2^2) p(\boldsymbol{\omega}_1) p(\boldsymbol{\omega}_2) p(\sigma_\varepsilon^2) p(\sigma_e^2). \tag{3.14}$$

3.4.1.1 Deriving the Posterior Distribution of θ

Now given the observations of real system's responses $\mathbf{z}$ over variable inputs D_1 and the simulation model outputs $\bar{\mathbf{y}}$ over variable and calibration inputs D_2, the first step is to calculate the posterior distribution given data $\mathbf{d} = \{\mathbf{z}, \bar{\mathbf{y}}\}$

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}^2 | \mathbf{d}) \propto p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}) \phi(\mathbf{d} | \boldsymbol{\theta}; \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}), \tag{3.15}$$

where $\phi(\cdot; \cdot)$ is the density function of a multi-variate normal distribution whose mean and covariance functions can be represented in terms of $D_1, D_2, \mathbf{d}, \boldsymbol{\beta}, \boldsymbol{\omega}, \boldsymbol{\sigma}^2$.

The parameters $\boldsymbol{\beta}$ can be integrated out from (3.15). While the posterior distribution of $\boldsymbol{\theta}$ above provides the basis to quantify the uncertainty in calibration, it is challenging to conduct a full Bayesian analysis to obtain the posterior. Instead, two-stage procedures have been proposed to estimate the posterior in Kennedy and O'Hagan (2001), Yuan and Ng (2013b, 2015). Take the procedure in Yuan and Ng (2013b) as an example. The first step uses the simulation outputs $\bar{\mathbf{y}}$ to estimate $\boldsymbol{\omega}_1, \sigma_1^2, \sigma_\varepsilon^2$.

As pointed out in Kennedy and O'Hagan (2001), while $\mathbf{z}$ also contains some information that could be used to estimate $\boldsymbol{\omega}_1, \sigma_1^2, \sigma_\varepsilon^2$, the number of points in D_1 is often much smaller than the number of points in D_2, and $\mathbf{z}$ also depends on other parameters. Therefore, this simplification does not lose much information. In the second stage, $\boldsymbol{\omega}_1, \sigma_1^2, \sigma_\varepsilon^2$ are fixed and the full data $\mathbf{d}$ is used to estimate $\boldsymbol{\omega}_2, \sigma_2^2, \sigma_e^2$, using the Expectation-Maximization (EM) algorithm with $\boldsymbol{\theta}$ treated as latent parameters (Yuan and Ng 2013b). With the estimated $\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\sigma}}^2$, the conditional posterior distribution $p(\boldsymbol{\theta}|\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\sigma}}^2, \mathbf{d})$ can be used to quantify the uncertainty in prediction by integrating $\boldsymbol{\theta}$ over $p(\boldsymbol{\theta}|\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\sigma}}^2, \mathbf{d})$.

3.4.1.2 Initial Experiment Design

The data set $\mathbf{z}$ and D_1 are typically considered as given in model calibration and there is not much of a choice. For the simulation model output data set $\bar{\mathbf{y}}$, we generally can choose the set of variable and calibration inputs D_2 to run simulations. An intuitive consideration is the variable inputs in D_2 should include, or at least close to the variable inputs in D_1.

While the computer experiment literature has studied experimental design extensively (Santne et al. 2013), experimental design for Bayesian calibration faces new challenges because of the existence of calibration input parameters. Kennedy and O'Hagan took a heuristic approach (Kennedy and O'Hagan 2001). There are two general principles in choosing D_2. First, the design points should cover the range of variable inputs not only in the calibration process but also the range for predictions in the future. Second, the experiment design should cover the plausible range containing the true value $\boldsymbol{\theta}^*$.

In Kennedy and O'Hagan (2001), the initial experiment design was chosen to be the maxmin Latin hypercube design for the variable and calibration inputs. Such a heuristic design provides good coverage of the design space and also has some computational advantages. In Yuan and Ng (2015), a single Latin hypercube design was applied to both variable and calibration inputs, with an additional requirement of at least 10 design points per dimension according to a common recommendation in the Gaussian process literature (Loeppky et al. 2009). The Gaussian process model for $y(\cdot, \cdot)$ should also be validated and refined if necessary (Loeppky et al. 2009).

3.4.1.3 A General Sequential Bayesian Calibration Approach

Unlike traditional experimental design for computer experiment that focuses on obtaining good coverage of the design space for interpolation, Bayesian calibration needs to identify the optimal calibration input value $\boldsymbol{\theta}^*$. As a result, it would benefit from a sequential experimental design that adds points with an objective to improve calibration accuracy. Yuan and Ng (2013b, 2015), Yuan et al. (2013) studied this problem and proposed two criteria to guide the sequential Bayesian process. Notice

that the sequential Bayesian calibration approach is similar to direct calibration as new values of $\boldsymbol{\theta}$ are added and evaluated. The difference is how the new point is determined.

- Entropy Criterion. The entropy of $\boldsymbol{\theta}$ before adding a new design point is $H(\boldsymbol{\theta}) = -\int p(\boldsymbol{\theta}) \ln(p(\boldsymbol{\theta}))d\boldsymbol{\theta}$, where $p(\boldsymbol{\theta})$ is the prior density function of $\boldsymbol{\theta}$. Notice that the more uncertainty there is in the calibration estimate of $\boldsymbol{\theta}$, the larger the value of the entropy is. After conducting experiments on new design points D, the conditional entropy of $\boldsymbol{\theta}$ is then $H(\boldsymbol{\theta}|D) = -\int p(\boldsymbol{\theta}|D) \ln(p(\boldsymbol{\theta}|D))d\boldsymbol{\theta}$, where $p(\boldsymbol{\theta}|D)$ is the posterior density function of $\boldsymbol{\theta}$ after the experiment. Then the entropy criterion aims to maximize the entropy gain $\mathbb{E}[H(\boldsymbol{\theta}) - H(\boldsymbol{\theta}|D)]$, and equivalently to choose the new design point D that minimize $\mathbb{E}[H(\boldsymbol{\theta}|D)]$ since $H(\boldsymbol{\theta})$ is independent of the new design point D. For stochastic simulations, it is also possible to allocate the simulation budget to existing design points. Doing so will reduce the noise in the simulation estimate and help improve the accuracy in the parameter estimation, thus also improving the gain in entropy.
- Expected Integrated Mean Square Prediction Error Criterion (EIMSPE). Since the ultimate objective of model calibration is to improve prediction accuracy, it is quite intuitive to find a new design point D that would minimize EIMSPE. In comparison, the entropy criterion simply focuses on reducing the uncertainty in the estimate of $\boldsymbol{\theta}^*$ and does not explicitly consider the prediction uncertainty.

Yuan and Ng (2015) found out that neither the entropy criterion nor the EIMSPE criterion dominates the other. When the uncertainty in the estimate of $\boldsymbol{\theta}^*$ is very large, the entropy criterion tends to achieve better predictive performance. Otherwise, it would be better to use the EIMSPE criterion to improve predictive performance as it explicitly considers the prediction error. Based on these observations, it was then proposed to combine both criterion in the sequential Bayesian calibration process. To summarize, the idea is to use the entropy criterion in the beginning phase to quickly improve the quality of the calibration input parameter estimates. Once the uncertainty in the estimate is sufficiently small, EIMSPE can be used to further improve predictive performance.

3.4.1.4 Calibrating a Parkinson Disease Simulation

In Yuan and Ng (2015), Yuan and Ng applied the sequential Bayesian calibration approach using the combined entropy and EIMSPE sequential experimental design to calibrate a stochastic biological model of mtDNA population dynamics. The model can be used for assessing interventions designed to treat Parkinson's disease or predicting changes in the occurrences of Parkinson's disease in populations (Henderson et al. 2009). Notice this model was also used in Yuan et al. (2013) to demonstrate direct calibration using stochastic approximation as reviewed in Sect. 3.3.1. We only describe the difference in this section and please refer to Sect. 3.3.1 for details.

The initial experiment design includes 140 input sets generated using the Latin hypercube design for the calibration parameters using the normal priors, and the

variable input $x \in (1, 110)$, which is the age of the patient. For each input set, 10 replications were taken to estimate the simulation model output at each input set. With data from the initial experiments, the Gaussian process model was built and validated as a surrogate for the simulation model.

In the sequential experiment stage, 10 follow-up input sets were selected for simulations. The combined criterion successfully reduced the uncertainty in the estimates of the calibration input parameters to the level of the entropy criterion, as measured by the variances of the posterior distribution. At the same time, the combined criterion has the benefit of the EIMSPE criterion in terms of reduced prediction uncertainty.

3.4.2 *Simultaneous Bayesian Calibration of Physical and Tunable Parameters*

Depending on the nature of the calibration input parameters $\boldsymbol{\theta}$, they can be further categorized into *physical parameters* $\boldsymbol{\theta}_p$ and *tunning parameters* $\boldsymbol{\theta}_t$ (Han et al. 2009). Physical parameters have meanings in the real system, such as the fitness values of different genotypes under drug selection pressure in the biological model simulating the development of drug resistance in parasites described in Sect. 3.3.2.

In contrast, tuning parameters only exist in the computer simulation model and have no meanings in the real system. They are used to control the behavior of the simulation model and tune model output. For example, a vector of parameters representing bonuses and penalties used to encourage specific behaviors of the approximate dynamic programming model developed to make dispatching decisions for a large trucking company (Frazier et al. 2009). Tuning these parameters allows the analyst to match the model output with a series of historical performance metrics.

Direct calibration methods do not need to differentiate between tuning parameters and physical parameters. However, Bayesian calibration methods may not perform well when applied to both tuning and physical parameters. In Han et al. (2009), Bayesian calibration was applied to calibrate a tuning parameter and a physical parameter in a biomechanics simulation model for knee implants (Rawlinson et al. 2006). The tuning parameter controls load discretization and the physical parameter specifies the initial position. Using a near-uniform distribution as the priors for these two parameters, the posteriors were then calculated after running the simulations on 100 equally spaced values of each parameter. However, the posterior of load discretization is bimodal and the posterior of the initial position has a large variance. Therefore, both parameters were poorly calibrated.

In Han et al. (2009), a simultaneous Bayesian calibration of physical and tunable parameters was proposed to address the limitation of Bayesian calibration when applied to tuning parameters. The main idea is unlike physical parameters that can be reasonably described by a prior distribution and a posterior distribution updated using experiment data, tuning parameters do not really have any "uncertainty".

Their existence is for the sole purpose of minimizing the difference between the simulation model and the real system's response, as measured in forms described in (3.6) and (3.7). In other words, a direct calibration approach should be applied to determine $\boldsymbol{\theta}_t$ while Bayesian calibration can still be applied to $\boldsymbol{\theta}_p$ such that one can still benefit from Bayesian calibration's capabilities to quantify the uncertainty in model predictions.

The procedure in Han et al. (2009) works with Gaussian processes with constant mean functions. Following the notations in equation (3.11), this means $\boldsymbol{\beta}_1 = \beta_{1,0}$ and $\boldsymbol{\beta}_2 = \beta_{2,0}$, where $\beta_{1,0}$ and $\beta_{2,0}$ are two unknown parameters. The procedure has three steps:

Step 1 Generate a grid embedded within the plausible ranges of $\boldsymbol{\theta}_t$ and estimate the difference between observations of the real system and simulation outputs. For each $\boldsymbol{\theta}_t$ on the grid, do the following:

1. Determine the values of $\beta_{1,0}$ and $\beta_{2,0}$ using a non-Bayesian argument that interprets $\beta_{2,0}$ as the minimized difference between observations and simulation outputs over the feasible ranges of $\boldsymbol{\theta}_p$. This is done by using Latin hypercube design to select a set of $\boldsymbol{\theta}_p$ values, and then compute the difference between observations and simulation outputs for these $\boldsymbol{\theta}_p$ values, and set $\beta_{2,0}$ to the smallest among them. Then $\beta_{1,0}$ is simply the difference between the mean of the observations of the real system and $\beta_{2,0}$.
2. Generate Monte Carlo samples from the posterior of $\boldsymbol{\theta}_p, \boldsymbol{\omega}, \boldsymbol{\sigma}^2$.
3. For a set of variable inputs, compute the estimated discrepancy between real observations and simulation outputs using the Monte Carlo samples of $\boldsymbol{\theta}_p, \boldsymbol{\omega}, \boldsymbol{\sigma}^2$.

Step 2 Estimate the optimal tuning parameter values $\widehat{\boldsymbol{\theta}}_t^*$ by selecting the $\boldsymbol{\theta}_t$ that achieves the smallest discrepancy between observations of the real system and simulation outputs according to the estimates in Step 1.

Step 3 Holding $\widehat{\boldsymbol{\theta}}_t^*$ fixed, perform Bayesian calibration on $\boldsymbol{\theta}_p$ and obtain the posterior distribution.

When applied to the same biomechanics simulation model for implant knee, the variance posterior distribution of the physical parameter (initial position) was much lower than calibrating both the load discretization and initial position using the Bayesian calibration method. This represents a significant increase in calibration accuracy.

3.5 Summary

In this chapter, we review two main classes of methods for model calibration. Direct calibration has been the traditional approach towards model calibration. It is intuitive and easy to grasp by simulation modelers and analysts, and applies equally well

to both physical and tuning parameters. Traditionally, direct calibration was limited to manually selecting a small set of calibration inputs to compare due to a lack of computational power and methods to systematically perform calibration. In the past decades, productive research activities on simulation optimization have created efficient simulation optimization algorithms (Fu 2015b; Hong et al. 2015; Xu et al. 2016a) that largely alleviate this hurdle. These algorithms help make direct calibration more appealing, even when $\boldsymbol{\theta}$ is high-dimensional and may include a mix of continuous, integer, and categorical variables. It is equally important to point out that rigorous simulation optimization algorithms such as ISC properly handles the heterogeneous noise in stochastic simulations. However, direct calibration falls short of fully representing the uncertainty in the calibrated parameters and subsequently the uncertainty in model predictions.

Bayesian calibration offers a complementary set of merits and limitations. It provides a rigorous framework to integrate data with prior knowledge on calibration parameters, and fully accounts for the uncertainty in calibration parameter estimates and predictions. However, Bayesian calibration methods often require a considerable level of technical sophistication to be applied successfully. The use of Gaussian process also means Bayesian calibration in its current form is more appropriate for low-dimensional (e.g., 10) continuous calibration parameters, except when there is specific structural information to exploit as is the case of the stochastic user equilibrium model in a dynamic traffic assignment simulator in Flötteröd et al. (2011). While current Bayesian calibration methods model observational error and simulation noise, they are assumed to be homogeneous. However, it is well-known that stochastic simulations, especially discrete-event simulation models for queuing systems, often have variances that can be very different across the range of variable inputs of interest.

To conclude this chapter, we briefly discuss several directions of research that may produce new general and efficient model calibration methods in the future.

Efficient simulation optimization algorithms for direct calibration Direct calibration depends critically on the performance of the simulation optimization algorithm. While simulation optimization researchers have made big strides in the past decades with algorithms such as nested partition (Shi and Olafsson 2009), ISC (Xu et al. 2010), and R-SPLINE (Wang et al. 2013), there is still a great need for further computational efficiency improvement when simulation models are time-consuming to run. One promising strategy is to design simulation optimization algorithms that allocate limited simulation budget in an optimal way to minimize the impact of stochastic noise on the search. Preliminary results on random search (Zhu et al. 2016), cross entropy (He et al. 2010; Zhang et al. 2016), particle swarm optimization (Taghiyeh et al. 2016; Zhang et al. 2017), and nested partition (Chen et al. 2014) have shown the potential of such approaches. Another direction of work is to use statistical meta-models as a surrogate of the simulation model (Barton 2009; Xu 2012; Salemi et al. 2014). Then in the direct calibration process, simulation optimization algorithms can use predictions made by the

surrogate models to guide the search process, limiting the number of expensive simulations that need to be run during calibration.

Extending Bayesian calibration for stochastic model calibration One major limitation of Bayesian calibration is the assumption of homogeneous simulation noise. Recent works on stochastic kriging (Ankenman et al. 2010; Chen et al. 2012, 2013) extends Gaussian process to handle stochastic problems with heterogeneous variances, and may potentially extend the applicability of Bayesian calibration to more general stochastic simulation models. The stochastic kriging framework may also be extended in a similar way as in Han et al. (2009) to calibrate both physical and tuning parameters.

Multi-fidelity model calibration Both direct and Bayesian calibration methods may involve a large number of simulation model runs. For large-scale problems, a high-fidelity simulation model may be very time-consuming to run and thus the calibration process may face a prohibitive computation burden. Recent developments in multi-fidelity simulation optimization show that when low-fidelity models are available, it is possible to extract useful information from these approximate but computationally cheap models to achieve a significant increase in computational efficiency (Xu et al. 2014, 2016a, b). Therefore, multi-fidelity simulation optimization provides a promising path for direct calibration of time-consuming simulation models. The Bayesian calibration framework may also be extended to make use of information from low-fidelity models (Absi and Mahadevan 2016).

Acknowledgements This work has been supported in part by National Science Foundation under Award CMMI-1233376, CMMI-1462787, and ECCS-1462409 (jointly funded by United States Air Force Office of Scientific Research).

References

Absi GN, Mahadevan S (2016) Multi-fidelity approach to dynamics model calibration. Mech Syst Signal Proces 68:189–206

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. Oper Res 58:371–382

Aral KD, Chick SE, Grabosch A (2014) Primary preventive care model for type 2 diabetes: input calibration with response data. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 1399–1410

Barton RR (2009) Simulation optimization using metamodels. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 230–238

Chen X, Ankenman BE, Nelson BL (2012) The effects of common random numbers on stochastic kriging metamodels. ACM Trans Model Comput Simul 22:7

Chen X, Ankenman BE, Nelson BL (2013) Enhancing stochastic kriging metamodels with gradient estimators. Oper Res 61:512–528

Chen W, Gao S, Chen CH, Shi L (2014) An optimal sample allocation strategy for partition-based random search. IEEE Trans Autom Sci Eng 11:177–186

Flötteröd G, Bierlaire M, Nagel K (2011) Bayesian demand calibration for dynamic traffic simulations. Transp Sci 45:541–561

Frazier P, Powell WB, Simão HP (2009) Simulation model calibration with correlated knowledge-gradients. In: Proceedings of the 2009 winter simulation conference. IEEE, Piscataway, NJ, pp 339–351

Fu MC (2015a) Stochastic gradient estimation. In: Handbook of simulation optimization. Springer, New York, pp 105–147

Fu MC (2015b) Handbook of simulation optimization. Springer, New York

Han G, Santner TJ, Rawlinson JJ (2009) Simultaneous determination of tuning and calibration parameters for computer experiments. Technometrics 51:464–474

He D, Lee LH, Chen CH, Fu MC, Wasserkrug S (2010) Simulation optimization using the cross-entropy method with optimal computing budget allocation. ACM Trans Model Comput Simul 20:4

Henclewood D, Suh W, Rodgers M, Hunter M, Fujimoto R (2012) A case for real-time calibration of data-driven microscopic traffic simulation tools. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 1670–1681

Henderson DA, Boys RJ, Krishnan KJ, Lawless C, Wilkinson DJ (2009) Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons. J Am Stat Assoc 104:76–87

Hong LJ, Nelson BL, Xu J (2010) Speeding up COMPASS for high-dimensional discrete optimization via simulation. Oper Res Lett 38:550–555

Hong LJ, Nelson BL, Xu J (2015) Discrete optimization via simulation. In: Handbook of simulation optimization. Springer, New York, pp 9–44

Johnson RT, Lampe TA, Seichter S (2009) Calibration of an agent-based simulation model depicting a refugee camp scenario. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 1778–1786

Kennedy MC, O'Hagan A (2001) Bayesian calibration of computer models. J R Stat Soc Ser B 63:425–464

Kushner HJ, Yin G (2003) Stochastic approximation and recursive algorithms and applications. Springer, New York

Latek MM, Mussavi Rizi SM, Geller A (2013) Verification through calibration: an approach and a case study of a model of conflict in Syria. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 1649–1660

Leathwick DM (2013) Managing anthelmintic resistanceparasite fitness, drug use strategy and the potential for reversion towards susceptibility. Vet Parasitol 198:145–153

Leathwick DM, Hosking BC (2009) Managing anthelmintic resistance: modelling strategic use of a new anthelmintic class to slow the development of resistance to existing classes. NZ Vet J 57:203–207

Loeppky JL, Sacks J, Welch WJ (2009) Choosing the sample size of a computer experiment: a practical guide. Technometrics 51:366–376

Mackinnon MJ (2005) Drug resistance models for malaria. Acta Tropica 94:207–217

Matus O, Barrera J, Moreno E, Rubino G (2016) Calibrating a dependent failure model for computing reliabilities on telecommunication networks. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 490–500

Molento MB, Nielsen MK, Kaplan RM (2012) Resistance to avermectin/milbemycin anthelmintics in equine cyathostominscurrent situation. Vet Parasitol 185:16–24

Nielsen MK, Vidyashankar AN, Hanlon BM, Diao G, Petersen SL, Kaplan RM (2013) Hierarchical model for evaluating pyrantel efficacy against strongyle parasites in horses. Vet Parasitol 197:614–622

Rawlinson JJ, Furman BD, Li S, Wright TM, Bartel DL (2006) Retrieval, experimental, and computational assessment of the performance of total knee replacements. J Orthop Res 24:1384–1394

Salemi P, Nelson BL, Staum J (2014) Discrete optimization via simulation using Gaussian Markov random fields. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 3809–3820

Santner TJ, Williams BJ, Notz WI (2013) The design and analysis of computer experiments. Springer, New York

Shi D, Brooks RJ (2007) The range of predictions for calibrated agent-based simulation models. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 1198–1206

Shi L, Olafsson S (2009) Nested partitions method, theory and applications. Springer, New York

Taghiyeh S, Xu J (2016) A new particle swarm optimization algorithm for noisy optimization problems. Swarm Intell 10:161–192

Vidyashankar AN, Xu J (2015) Stochastic optimization using Hellinger distance. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 3702–3713

Vock S, Enz S, Cleophas C (2014) Genetic algorithms for calibrating airline revenue management simulations. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 264–275

Wang H, Pasupathy R, Schmeiser BW (2013) Integer-ordered simulation optimization using R-SPLINE: retrospective search with piecewise-linear interpolation and neighborhood enumeration. ACM Trans Model Comput Simul 23:17

Xu J (2012) Efficient discrete optimization via simulation using stochastic kriging. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 466–477

Xu J, Nelson BL, Hong, LJ (2010) Industrial strength COMPASS: a comprehensive algorithm and software for optimization via simulation. ACM Trans Model Comput Simul 20:3:1–3:29

Xu J, Nelson BL, Hong LJ (2013) An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. INFORMS J Comput 25:133–146

Xu J, Vidyashankar A, Nielsen, MK (2014) Drug resistance or re-emergence? Simulating equine parasites. ACM Trans Model Comput Simul 24:20:1–20:23

Xu J, Huang E, Chen CH, Lee, LH (2015) Simulation optimization: a review and exploration in the new era of cloud computing and big data. Asia-Pac J Oper Res 32:1650017:1–1650017:26

Xu J, Huang E, Hsieh L, Lee LH, Jia QS, Chen CH (2016a) Simulation optimization in the era of industrial 4.0 and the industrial internet. J Simul 10:310–320

Xu J, Zhang S, Huang E, Chen CH, Lee LH, Celik N (2016b) MO^2TOS: Multi-fidelity optimization with ordinal transformation and optimal sampling. Asia-Pac J Oper Res 33:1650017

Xu J, Zhang S, Huang E, Chen CH, Lee LH, Celik N (2014) Efficient multi-fidelity simulation optimization. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 3940–3951

Yuan J, Ng SH (2013a) An entropy based sequential calibration approach for stochastic computer models. In: Proceedings of the winter simulation conference. IEEE, Piscataway, NJ, pp 589–600

Yuan J, Ng SH (2013b) A sequential approach for stochastic computer model calibration and prediction. Reliab Eng Syst Saf 111:273–286

Yuan J, Ng SH, Tsui KL (2013) Calibration of stochastic computer models using stochastic approximation methods. IEEE Trans Autom Sci Eng 10:171–186

Yuan J, Ng SH (2015) Calibration, validation, and prediction in random simulation models: Gaussian process metamodels and a Bayesian integrated solution. ACM Trans Model Comput Simul 25:18:1–18:25

Zhang S, Lee LH, Chew EP, Xu J, Chen CH (2016) A simulation budget allocation procedure for enhancing the efficiency of optimal subset selection. IEEE Trans Autom Control 61:62–75

Zhang S, Xu J, Lee LH, Chew EP, Wong WP, Chen CH (2017) Optimal computing budget allocation for particle swarm optimization in stochastic optimization. IEEE Trans Evol Comput 21:206–219

Zhu C, Xu J, Chen CH, Lee LH, Hu JQ (2016) Balancing search and estimation in random search based stochastic simulation optimization. IEEE Trans Autom Control 61:3593–3598

Chapter 4
Validating Emergent Behavior in Complex Systems

Claudia Szabo and Lachlan Birdsey

Abstract Undesired or unexpected properties are frequent, as large-scale complex systems with nonlinear interactions are being designed and implemented to answer real-life scenarios. Identifying these behaviors as they happen as well as determining whether these behaviors are beneficial for the system is crucial to highlight potential faults or undesired side effects early in the development of a system, thus promising significant cost reductions. Beyond the inherent challenges in identifying these behaviors, the problem of validating the observed emergent behavior remains challenging, as this behavior is, by definition, not expected or envisaged by system designers. This chapter presents an overview of existing work for the automated detection of emergent behavior and discusses some potential solutions to the challenge of validating emergent behavior. Building on the idea of comparing an identified emergent behavior with previously seen behaviors, we propose a two-step process for validating emergent behavior. Our initial experiments using a Flock of Birds model show the promise of this approach but also highlight future avenues of research.

4.1 Introduction

Unexpected behavior that cannot be reduced to the behavior of the individual system components is common in complex systems, leading to in-depth analysis once the behaviors are observed (Davis 2005; Johnson 2006; Mogul 2006). As systems grow in the number and complexity of components, as well as the type of interactions and coupling between components, these *emergent behaviors* are becoming critical to analyze (Johnson 2006; Mogul 2006; Bedau 1997; Holland 1999). In other cases, a reverse process is undertaken, where a specific system of systems is designed and implemented with the specific aim of it exhibiting an emergent behavior. In recent years, advances in complexity theory and modeling and simulation has led to the

C. Szabo (✉) · L. Birdsey
The University of Adelaide, Adelaide, Australia
e-mail: claudia.szabo@adelaide.edu.au

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_4

analysis of a plethora of emergent properties examples, from flocks of birds, ant colonies, to the appearance of life and of traffic jams. In software systems, connection patterns have been observed in data extracted from social networks (Chi 2009) and trends often emerge in big data analytics (Fayyad and Uthurusamy 2002). More malign examples of emergent behavior include power supply variation in smart grids due to provider competition (Chan et al. 2010), the Ethernet capture effect in computer networks (Ramakrishnan and Yang 1994), and load balancer failures in a multi-tiered distributed system (Mogul 2006). As emergent properties may have undesired and unpredictable consequences (Mogul 2006; Ramakrishnan and Yang 1994; Floyd and Jacobson 1993), systems that exhibit such behaviors become less credible and difficult to manage.

Emergent properties have been studied since the 1970s (Bedau 1997; Holland 1999; Cilliers 1998; Gardner 1970; Seth 2008), and a number of methods for their identification, classification, and analysis exist (Seth 2008; Chen et al. 2007; Kubik 2003; Szabo and Teo 2012a; Brown and Goodrich 2014). Existing methods are usually employed on simplified examples such as the Flock of Birds model, where only three flight rules are implemented, as opposed to the myriad rules that affect flocking in real life. Approaches to identifying emergent behaviors can be categorized as either attempting to identify emergence as it happens, without prior knowledge, or as using a definition of an emergent property and trying to identify its root causes. In the first type of approaches (Kubik 2003; Szabo and Teo 2012a), formal methods or meta-models calculated from composed model states are used. A key challenge remains how to identify variables or attributes that describe the system components, or the *micro-level*, and the system as a whole, or the *macro-level*, and the relationships and dependencies between these two levels. Once these are defined, emergence can be specified as the set difference between macro-level and the micro-level, however these levels are extremely difficult to capture and computationally expensive to calculate. This does not happen when using a definition of a known or observed emergent property with the aim of identifying its cause, in terms of the states of system components and their interaction (Seth 2008; Chen et al. 2007). A key issue in these works is that a prior observation of an emergent property is required, and that emergent properties need to be defined in such a way that the macro-level can be reduced or traced back to the micro-level.

In addition to these challenges, most approaches (Seth 2008; Chen et al. 2007; Kubik 2003; Brown and Goodrich 2014) are demonstrated using simple models such as Flock of Birds or Predator–Prey, which have limiting assumptions and constraints when applied to more complex systems. For example, most approaches do not consider mobile agents (Kubik 2003), assume unfeasible a priori specifications and definitions of emergent properties (Szabo and Teo 2012a), or do not scale beyond models with a small number of agents (Teo et al. 2013). In the multi-agent systems community, approaches focus more on the engineering of systems to exhibit beneficial emergent behavior and less on its identification (Bernon et al. 2003; Jacyno et al. 2009; Salazar et al. 2011). However, approaches that engineer emergent behavior do not ensure that no other side effects occur as a consequence.

To the best of our knowledge, no works focus on the validation of emergent behavior, that is, on determining whether the observed emergent behavior is beneficial or harmful for the system, with some theoretical advancements (Szabo and Teo 2012b) existing in component-based simulations but without practical applications in real-life scenarios. The challenges of validating a previously identified emergent behavior surpass those of identifying the emergent behavior itself, as it is almost impossible, using traditional validation methods, to distinguish between invalid system outputs and the outputs generated by the emergent behavior itself. This challenge is even further exacerbated when considering bias (Tolk 2017), and what better place for bias to appear than when faced with results that seem impossible, such as those generated by emergent behavior.

In this chapter, we focus on the validation of emergent behavior as a two-step process. The first step focuses on identifying that the system exhibits emergent behavior and, if possible, on identifying the interactions between subcomponents that have caused the emergent behavior to appear. The second step focuses on identifying whether the emergent behavior is beneficial or desirable by comparing the identified behavior with behaviors that have been previously validated. We discuss several existing works that could be adapted to solve each step, outlining their limitations and highlighting several directions for future work.

4.2 Identifying Emergent Behavior

Complex systems are ubiquitous and encompass systems from every discipline, from biology (Odell 1998) to computer science (Odell 1998; Johnson 2006), an a tremendous body of work has been dedicated to better understand the various unexpected or undesired behaviors produced by these systems (Johnson 2006; Mogul 2006; Odell 1998; Fromm 2006). Holland (1999) and Bedau (1997) have pioneered the field by defining key terms such as *weak* and *strong* emergence as well as defining these terms using different degrees (Holland 2007). In this article, we consider *weak* emergence as being the *macro-level* behavior that is a result of *micro-level* component interactions, and *strong* emergence as the *macro-level* feedback or causation on the *micro-level*.

Bedau (1997) defines an emergent property as "a property of assemblage that could not be predicted by examining the components individually." The flocking of birds is a well-known example of emergent behavior in nature. Independent birds aggregate around an invisible center and fly at the same speed for flock creation. The birds come together to create something that would be entirely indiscernible by studying only one or two birds. Two well-known examples of systems that exhibit emergent behavior caused by interaction are the Fock of Birds model (Reynolds 1987), and the cellular automata Game of Life model (Gardner 1970). The former achieves its emergent properties through each bird flocking around a perceived flock center, while in the latter model the emergent properties are achieved by the patterns that are formed by the cells transitions between states. Studies that propose various

processes of detecting and identifying emergent behavior tend to use either one or both of these systems to prove the validity of their proposed approach (Chan et al. 2010; Seth 2008; Szabo and Teo 2013; Chan 2011a).

It is a challenging task to build the exact specification and implementation of a complex system, due to its inherent size, complexity, and nonlinear interactions between components. To address this, agent-based systems, and in particular multi-agent systems, are a useful formalism to model complex systems. The components present in a complex system can be modeled as agents that perform their respective actions and interactions. The agent-based models are then used in simulations to assist with research and analysis (Johnson 2006); in addition, multi-agent systems can be engineered to exhibit emergent properties (Fromm 2006; Savarimuthu et al. 2007). Several formalisms have been proposed to obtain or engineer emergent behavior, such as the DEVS extension proposed by Mittal (2013), Birdsey et al. (2016), but they have yet to be employed in practice. By creating models where emergence is an easily attainable product derived from agents interactions, users are relieved from having to model every aspect of the complex system under study. Multi-agent systems which have been designed to exhibit emergence are usually engineered to focus on self-organization and cooperation between agents. These systems generally rely on a system expert to identify the emergent behavior (Jacyno et al. 2009; Salazar et al. 2011; Savarimuthu et al. 2007). For example, human societies and the myriad ways that emergent properties can arise are generally modeled using this approach in order to study aspects such as norm emergence (Jacyno et al. 2009; Savarimuthu et al. 2007). However, even for systems that have been engineered to exhibit emergence it is imperative to ensure that no side effects occur and that other undesired properties to not appear and this remains a fundamental challenge.

Chan et al. (2010) highlights that agent-based simulation is the most suitable method for modeling systems containing unexpected or emergent behaviors, because it emphasizes that the actions and interactions between agents are the main causes for emergent behaviors. Several works support the use of agent-based modeling for studying emergent behaviors (Salazar et al. 2011; Fromm 2006; Anh 2012; Serugendo et al. 2006). In addition to the Flock of Birds and Game of Life models, Chan et al. (2010) show that other complex systems such as social networks and electricity markets, implemented within an agent-based simulation, can exhibit emergent properties, which can then be identified. The methods in Chan et al. (2010) for detecting emergence rely upon the presence of a system expert, who can identify the emergent behavior.

There is a plethora of methods for the detection of emergence, and existing work can be categorized as either *postmortem* or *live* analysis (Szabo and Teo 2012a). *Postmortem* analysis methods are applied after the system under study has finished executing, and use data that was recorded during the execution (Szabo and Teo 2013, 2012a). In contrast, *live* analysis methods are used while the system under study is executing (Szabo and Teo 2012a; Chan 2011a). Most existing works focus on *postmortem* analysis methods (Szabo and Teo 2013; Chen et al. 2009; Tang and Mao 2014). In addition to *postmortem* and *live* analysis, methods can be classified into three main types (Teo et al. 2013): *grammar-based* (Szabo and Teo 2013;

Kubik 2003), *event-based* (Chen et al. 2007), or *variable-based* (Seth 2008; Szabo and Teo 2013; Tang and Mao 2014).

Grammar-based methods attempt to identify emergence in agent-based systems by using two grammars, L_{WHOLE} and L_{PARTS}. Kubik (2003) defines that L_{WHOLE} describes the properties of the system as a whole and L_{PARTS} describes the properties obtained from the reunion of the parts, and in turn produces emergence as the difference between the two solutions. L_{WHOLE} and L_{PARTS} can be easily calculated as the sets of words that are constructed from the output of agent behavior descriptions. This method does not require a prior observation of the system in order to identify possible emergent properties or behaviors, which therefore makes it suitable for large-scale models where such observations are notoriously difficult (Teo et al. 2013). However, grammars require a formation of *words* and the process of formalizing grammar tokens from the system information becomes impractical as the system grows in complexity, resulting in computational issues, especially (Teo et al. 2013; Kubik 2003). Work attempting to alleviate these issues identifies *micro*-level properties and model interaction, and performs reconstructability analysis on this data (Szabo and Teo 2013) in a *postmortem* context.

Event-based methods define behavior as a series of both simple and complex events that changed the system state, as defined by Chen et al. (2007). Complex events are defined as compositions of simple, atomic events where a simple event is a change in state of specific variables over some nonnegative duration of time. These state changes are defined by a set of rules or constraints. A constraint could be a temporal, spatial, or component or variable constraints. First, a temporal constraint defines the temporal relationship between two events. Second, a spatial constraint defines the space within which an event should occur relative to another. Finally, component or variable constraints define the relationship between variables or components of the two events. Each emergent property is defined manually by a system expert as a complex event. It is the particular sequence of both complex and simple events in a system that lead to emergence occurring in the system. However, this method relies heavily on the system experts and their specific definitions of emergent behaviors. Furthermore, it can suffer from both agent and state space explosion making it unsuitable for large systems.

In *variable-based methods*, a specific variable or metric is chosen to describe emergence. Changes in the values of this variable signify the presence of emergent properties (Seth 2008). The key idea is that emergence most likely occurs as the system self-organizes and exhibits some kind of patterns or structures, thus resulting in lower entropy. (Mnif 2006) introduce emergence as the difference between the entropy at the beginning and at the end of the system run. A system is said to exhibit emergence if the entropy difference is positive, i.e., the entropy value decreases in the end. Despite its simplicity, Shannon entropy only considers a single system attribute with discrete values.

The center of mass of a bird flock could be used as an example of emergence in bird flocking behavior, as shown in Seth (2008). Seth's approach uses Granger causality to establish the relationships between a macro-variable and micro-variables and proposes the metric of G-emergence, a near-*live* analysis method. This has the

advantage of providing a process for emergence identification that is relatively easy to implement. However, the approach requires system expert knowledge as observations must be defined for each system. Szabo and Teo (2013) proposed the use of reconstructability analysis to determine which components interacted to cause a particular emergent property (defined through a set of variables). They identified the interactions that cause birds to flock (Reynolds 1987), the cells that cause the glider pattern in Conway's Game of Life (Gardner 1970), and the causes of traffic jams. However, their method is heavily dependent on the choice of the variable set that represents the micro- and macro-levels and requires the intervention of a system expert.

Variable-based methods from other fields, such as information theory and machine learning, have been adapted with the goal of emergence detection. Information theory approaches for detecting emergence have also been proposed by using such techniques as Shannon Entropy (Gershenson and Fernandez 2012; Prokopenko et al. 2009; Tang and Mao 2014) and variety (Holland 2007; Yaneer 2004). These have advantages over other *variable-based methods* in that they can process large amounts of data efficiently. Tang and Mao (2014) propose measures of relative entropy that depend on the main emergent property of a system under study. However, these methods require the input of a system expert because they rely on the emergent property of a system being classified along with a specific function to be defined for that particular property. Machine learning classification techniques have also been proposed as a way of detecting emergence. A variant of Bayesian classification (Brown and Goodrich 2014) has been used to successfully detect swarming and flocking behavior in biological systems such as the Flock of Birds model (Reynolds 1987). This approach involves identifying *key features* of an agent, such as how many neighbors an agent has, and uses this information to determine the likelihood that a random set of agents is exhibiting emergence. Other methods from machine learning have been utilized, such as conditional random fields, and hidden Markov models in Vail et al. (2007), but with the goal of activity recognition in domain specific contexts. Vail et al. used conditional random fields and hidden Markov models somewhat successfully to determine if agents were performing a particular distinct action based on their relational position to other agents. A substantial number of methods for detecting emergence, both in a *live* or *postmortem* capacity exist, however the problem of determining whether the emergent behavior is beneficial remains unsolved.

4.3 Validating Emergent Behavior

The validation of emergent behavior in a complex system is very challenging due to the inherent "unexpected" nature of emergent behavior: in this case, traditional methods of validation will fall short, as there are no means whereby anomalous results, impossible to be seen in the real system modeled, can be distinguished between the results caused by the manifestations of the emergent behavior or property. To address this, we propose to divide the validation step into two main steps. In the first step,

focused on *identifying emergent behavior*, techniques such as the ones discussed above could be used to identify that the system exhibits an emergent property. In addition, techniques such as reconstructability analysis (Szabo and Teo 2013) could be used to obtain further information about the causes of the identified emergent behavior, in terms of the system components involved in the interaction and the micro-property that caused the macro-emergent property to appear. Once the emergent property has been identified, the second step conducts a *behavior comparison* to determine if this current behavior is similar to some behaviors seen before. This method relies on the assumption that at the time of analysis, it will be understood by the system expert whether previously exhibited emergent behaviors are beneficial. We discuss these in the following.

4.3.1 Identifying Emergent Behavior

The identification of emergent behavior can be performed using any of the methods described above. In order for the behavior comparison to be effective, detailed information about the system must be recorded. This includes:

1. system states—all system states and their evolution over time must be recorded
2. interactions—both direct and indirect (see Chan 2011b) interactions must be recorded, including entity states before and after the interaction
3. emergent behavior causes—depending on the methods used in the behavior comparison, the causes of emergent behaviors, in terms of micro-properties, would have to be recorded. This would be the case if Bayesian classification were to be employed for comparison.

4.3.2 Behavior Comparison

The comparison between the model to be validated and a reference model has been proposed before for the validation of component-based simulations (Petty and Weisel 2003a, b; Szabo et al. 2009), where similar challenges are encountered. In these works, a formal representation of the model is compared with a reference model using graph-based techniques, however a significant challenge remains in specifying a reference model for comparison.

We build on this idea and the hypothesis that the evolution of component interactions over time is one of the causes of emergent behavior. We propose to capture the interaction between components over a time interval using an interaction graph formalism, and to compare between the interaction graphs of the current system under study, and a system that has been shown to exhibit emergent behavior, and for which emergent behavior has been previously validated.

4.3.2.1 Interaction Graph Formalism

We define an interaction graph to capture the interactions between the model entities over a given time interval T^s, where s is the size of the interval in time units and remains the same for a simulation run. The term *interactions* can capture all types of interactions, namely, information poll, communication, and indirect, as outlined by Chan (2011b). will capture all types of interactions as defined above. An interaction graph is a directed graph where each vertex represents a model entity, $e \in M$, and each arc represents a interaction between two entities, $e_i \rightarrow e_j$, and carries a weight w_{ij}. This is formally defined as:

$$IG_{T^s}(M) =< V_{T^s}, E_{T^s} > \tag{4.1}$$

$$V_{T^s} = \{e_i | a_i \in M, i = 1, \ldots, n\} \tag{4.2}$$

$$E_{T^s} = \{(e_i, e_j, w_{ij}) | e_i, e_j \in V_{T^s}, w_{ij} \in Z^+\} \tag{4.3}$$

The weight w_{ij} of the arc $e_i \rightarrow e_j$ is incremented every time an interaction between e_i and e_j takes place. Currently, this calculation increments the weight of the arc, but this can be refined to assign different weights to various interaction types, if, for example, more importance was assigned to indirect interaction as a cause of emergence (Szabo and Teo 2013).

4.3.2.2 Comparing Interaction Graphs

We discuss in the following several methods that could be used for comparing between two interaction graphs.

Comparison using Hausdorff Distance The Hausdorff distance (*HD*) is a metric that is used to determine the similarity between two graphs with respect to shape (Huttenlocher et al. 1993). For two interaction graphs $IG(A)$ and $IG(B)$, the Hausdorff distance is defined as:

$$HD(A, B) = \max\{h(A, B), h(B, A)\} \tag{4.4}$$

where

$$h(A, B) = \max_{a \in A}\{\min_{b \in B}\{d(a, b)\}\} \tag{4.5}$$

and d is the distance between vertices a and b, with $a \in A$ and $b \in B$, respectively. For points in a two-dimensional Euclidian space, the distance d could be calculated as the Euclidian distance between points $a(x_a, y_a)$ and $b(x_b, y_b)$ as

$d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$, where (x_a, y_a) and (x_b, y_b) are the Cartesian coordinates of points a and b, respectively. However, any distance can be used.

As an example, in a bird flocking model, the coordinates could be those of the birds. Intuitively, the Hausdorff distance between interaction graphs A and B measures how close A and B are to each other, where "close" means that every vertex in A is close to some vertex in B, using some distance function such as the two-dimensional Euclidian distance. When other distances are used, domain ontologies can be employed to further refine the meaning of similarity.

The Hausdorff distance focuses on the position of the agents and only considers agents that are interacting, i.e., are included in the interaction graph. This implies that this metric will not give useful results in cases where agents are stationary or agent interactions are sparse. In cases where the entities are stationary, their positions across all interaction graphs are unchanging, which would lead to results with constant values. If entity interactions are sparse, the interaction graphs would contain wildly different information at each interval as few agents would be captured. This could be remedied by careful selection of the snapshot interval size. Moreover, as the coordinate information is recorded at the end of the snapshot interval and the metric does not consider interaction beyond the presence of the nodes in the *IG*, the distance function ignores cases in which the emergent behavior happens in the middle of the interval. This further makes the metric dependent on the size of the snapshot interval. For example, if a flock formed at the start of a snapshot interval but was broken midway through the snapshot interval, the entity positions at the end of the snapshot would likely not represent a flock.

Comparison using Statistical Complexity Statistical complexity (*SC*) is a metric that determines the amount of information entropy, or complexity, in a particular graph. As the entropy of a system decreases, it can be shown that collective behavior increases (Tang and Mao 2014). As emergence can be construed as being a collective behavior, it makes sense to consider measures of entropy to capture emergence. The simplest way to determine statistical complexity is by using Shannon entropy (Gershenson and Fernandez 2012; Prokopenko et al. 2009), which is defined as:

$$SC_{SE} = -\sum_{i=0}^{n} p(a_i) log_{10} p(a_i) \tag{4.6}$$

where $p(a_i)$ is the probability mass function with respect to agent a_i, and n is the total number of agents. While generic probabilistic mass functions can be defined to cover all systems, more accurate results may be obtained by specifying it for each particular system. For example, for the Flock of Birds model the probability mass function, $p(a_i)$, could be defined as:

$$p(a_i) = \frac{\sum_{j=0}^{N} \{w_{jk} | j \vee k = i\}}{(n^2 - n) \times s \times typesOfInteractions} \tag{4.7}$$

where $w_{jk} \in E$, s is the snapshot size, n is the total number of agents in the snapshot, N is the total number of interactions that occurred in the snapshot, and *typesOfInteractions* is the number of interaction rules for the system. This particular $p(a_i)$ calculation represents the proportion of total interactions across the snapshot interval in which agent a_i is involved.

Since statistical complexity operates on a single interaction graph, a threshold or series of thresholds must be defined so as to allow for comparison between a candidate interaction graph and a reference interaction graph. A simple way of achieving this is by taking the difference of the result for the candidate interaction graph (IG) and the reference interaction graph (IG_e).

$$D(IG, IG_e) = SC_{SE}(IG_e) - SC_{SE}(IG) \tag{4.8}$$

By taking the difference, the change in information entropy between the candidate and reference interaction graphs becomes evident. If we were to use the individual $SC_{SE}(IG)$ result, it may not highlight an emergent behavior, as some systems do not show emergent behavior when this result tends to zero (Tang and Mao 2014).

G-Emergence G-Emergence is a metric that can provide a strong indication of the occurrence of emergent behavior. G-Emergence is defined in Seth (2008) as:

$$ge_{M|\mathbf{m}} = ga_{M|\mathbf{m}}\left(\frac{1}{N}\sum_{i=1}^{N} gc_{m_{i\to M}}\right) \tag{4.9}$$

A *macro*-variable M is *G-emergent* from a set of *micro*-variables $\mathbf{m}$ if and only if M is *G-autonomous* with respect to $\mathbf{m}$, and M is *G-caused* by $\mathbf{m}$. The larger $ge_{M|\mathbf{m}}$ is, the greater the presence of emergent behavior. This implies that G-emergence will tend to 0 if M is directly caused by $\mathbf{m}$ or if M is entirely independent. The definitions for G-autonomy and G-causality are also defined in Seth (2008).

G-Emergence allows for detection of emergence in a sequence of snapshots as opposed to a singular snapshot. While this would result in slightly delayed detection, if the snapshot interval is small enough the delay would be negligible.

Comparison using Bayesian Classification Bayesian classification (BC) is a metric that determines if a collective behavior is occurring, and has been shown to be moderately effective in identifying swarming in models such as Flock of Birds and schools of fish. Bayesian classification is defined by Brown and Goodrich (2014) as:

$$\hat{b}_{collective} = *arg\,max_b P(b)\prod_{i\in\mathbf{S}} P(d_i|type)P(w_i|type) \tag{4.10}$$

where b is a set of collective behaviors, $\mathbf{S}$ is the set of randomly selected agents, and $P(d|type)$ and $P(w|type)$ are the likelihoods of the chosen features. The set of collective behaviors is equivalent to a set of emergent behaviors that are exhibited by the model, such as flocking or cell structure patterns. This set of collective behaviors

must be defined by a system expert. As with any classification method, given sufficient training data, *BC* can be used to distinguish between various types of the same emergent behavior, e.g., birds behaving collectively by flocking closely together or by following each other in a torodial structure (Brown and Goodrich 2014).

One of the main distinctions from the previously discussed approaches is that Bayesian classification requires several factors to be established prior comparison. First, a set of collective behaviors needs to be identified. In the case of the flocking of birds, the only collective behavior is flocking. This can be easily done if the first validation step provides comprehensive information about the identified emergent behaviors, perhaps allowing some form of grouping. Second, at least one *key feature* of an entity also needs to be predetermined. For example, Brown and Goodrich (2014) use the angular velocity of an agent and the number of neighbors for detecting emergence in a swarming model. In Bayesian classification, the likelihoods of these key features are assumed to be conditionally independent, so it is not unreasonable to assert that any number of key features could be determined and used. This leads to a slight redefinition of Bayesian classification:

$$\hat{b}_{collective} = *arg\,max_b P(b) \prod_{i \in \mathbf{S}} P(F_{\alpha i}|type)P(F_{\beta i}|type)... \tag{4.11}$$

where the variables are the same as Eq. 4.10 except that $F_\alpha i$ and $F_\beta i$ are *key features* with respect to agent i.

The challenge in calculating Bayesian classification lies in defining the key features of a system and determining their likelihood prior to execution. Brown and Goodrich (2014) propose that the likelihoods of each feature be determined through a training phase.

Comparison using Similar Neighbors By adapting a popular clustering algorithm, k-means clustering, to focus on the characteristics of agents, we may reveal information that is associated with a particular type of emergent behavior. Similar Neighbors is defined as:

$$SN(T_k) = \sum_{i<N}^{i=0} \sum_{j<N}^{j=0} \{e_i | i \neq j, d(e_i, e_j) < \delta, c(e_i, e_j)\} \tag{4.12}$$

where N is the total number of entities, $d(e_i, e_j)$ is the distance between e_i and e_j, δ is a distance threshold, and $c(e_i, e_j)$ is a boolean characteristic comparison function that checks whether e_i and e_j possess the same characteristics.

4.3.3 Initial Experiments

We analyze the suitability of using interaction graphs and Hausdorff Distances to compare between two emergent properties in the Flock of Birds model. The Flock of

(a) Flock of Birds: $IG(B_{20})$ with minimum HD: 109.77

(b) Flock of Birds: Reference Graph: $IG_e(B_{20})$

Fig. 4.1 Comparison between $IG(B_{20})$ and $IG_e(B_{20})$

Birds model (Reynolds 1987) captures the motion of bird flocking and is a seminal example in the study of emergence. At the macro-level, a group of birds tends to form a flock, as shown in Fig. 4.1b. Flocks have aerodynamic advantages, obstacle avoidance capabilities and predator protection, regardless of the initial positions of the birds. At the micro- level, each bird obeys three simple rules (Reynolds 1987):

1. separation—steer to avoid crowding neighbors
2. alignment—steer toward average heading of neighbors
3. cohesion—steer toward average position of neighbors

We model this as a multi-agent system in which each bird is an agent that has the three movement rules defined above. Other bird attributes include initial position and initial velocities. In our experiments, the initial bird positions can be either fixed or assigned randomly at start up. Bird velocities are assigned randomly. The model parameters can also influence emergent behavior analysis. As such, we collect and analyze interaction graphs of Flock of Birds models with sizes of 20 and 50 birds, with fixed and randomly assigned position values, and randomly assigned velocity values.

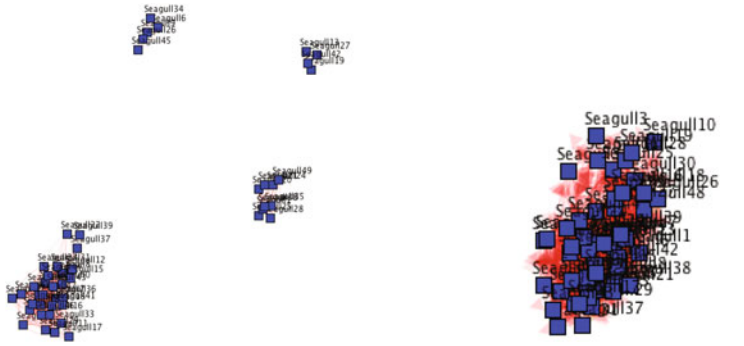
Table 4.1 presents the values of HD for a model of 20 birds, $IG(B_{20})$, and Fig. 4.1a is the companion interaction graph with the best result. The experiment compares interaction graphs at different time steps with $IG_e(B_{20})$, shown in Fig. 4.1b, which, for this experiment, is taken to be the interaction graph of a model with 20 birds that have the same starting positions. For both models, the initial velocities are randomly assigned. It is important to note here that a validation run in which $IG_e(B_{20})$ represents exactly the same model, i.e., the velocities have the same values, leads

Table 4.1 $HD(IG(B_{20}), IG_e(B_{20}))$: 20 birds, $s = 5$

Time interval	Min	Median	Mean	σ
T_{100}	109.77	161.62	165.62	31.34
T_{500}	110.49	173.93	203.15	84.86
T_{1000}	154.44	273.62	381.78	211.88

Table 4.2 $HD(IG(B_{50}), IG_e(B_{50}))$: 50 birds, $s = 5$

Time interval	Min	Median	Mean	σ
T_{100}	143.13	171.79	15.32	188.51
T_{500}	101.33	139.84	58.60	249.96
T_{1000}	86.84	215.83	88.08	346.39



(a) Flock of Birds: $IG(B_{50})$ with minimum HD: 86.84

(b) Flock of Birds: Reference Graph: $IG_e(B_{50})$

Fig. 4.2 Comparison between $IG(B_{50})$ and $IG_e(B_{50})$

to distance values of zero. The large variance shown in this particular experiment, and across many of the experiments, is caused by the initialization of bird velocities always being random. As the Flock of Birds experiments take place in an unbounded environment, it is highly likely that a small number of birds may be situated a large distance away from the flock therefore influencing the distance calculations.

Table 4.2 presents the result for a comparison where both the reference and candidate graphs contain 50 birds. Figure 4.2 presents both the candidate interaction graph with best result and the reference graph. Visual comparison between candidate graph with the best result and the reference clearly shows that emergence is present in multiple subsets of agents.

As it can be seen, we successfully detected emergence across several model configurations using a reference graph that contained only twenty birds. Despite the different model configurations ostensibly very similar semantically, they are very distinct from an automated emergence identification perspective with respect to numbers of birds, number of types and their position and velocity values.

We also found that graphs that were obtained at different time intervals could be compared successfully. In extreme cases, we theorize that the interval gap would be a problem but use of certain metrics may lessen this issue as well as how much information is present in the comparison and reference graphs. We also found that comparisons where the reference graph and candidate graph represented different numbers of agents did not influence the results significantly. In addition, we were

able to detect emergence with a reference graph of just twenty birds in a system that contained two types of birds and fifty birds overall.

These experiments show that this approach is promising, however several limitations exist. First, Hausdorff distance can only detect emergence that is a result of the agents' positions in space. Second, it only considers the positions recorded at the end of the time interval. Lastly, the Flocks of Birds model, while well studied in the literature, is still a simple, theoretical model. More detailed, realistic scenarios need to be considered, and all presented methods need to be studied in depth to further advance this work.

4.4 Conclusion

In this chapter, we highlight the inherent challenges in validating emergent behaviors, as an invalid behavior can be indistinguishable between an emergent behavior that has never been seen before. We propose a two-step process of identification and comparison to alleviate this. Our approach relies on an architecture that first identifies that an emergent, unexpected behavior has occurred in the system and retrieves and stores all information about this behavior. In the second step, the identified emergent behavior is compared with previously observed emergent behaviors using aninteraction graph formalism and several metrics, both existing and novel. This step builds upon a hypothesis that the evolution of component interaction is one of the causes of emergence. Our initial experiments show that this approach is promising, but more extensive experiments with various models under different conditions need to be performed.

References

Anh HT, Pereira LM, Santos FC (2012) The emergence of commitments and cooperation. In: van der Hoek W, Padgham L, Conitzer V, Winikoff M (eds) AAMAS. IFAAMAS, pp 559–566

Bedau M (1997) Weak emergence. Philos Perspect 11:375–399

Bernon C, Gleizes MP, Peyruqueou S, Picard G (2003) Adelfe: a methodology for adaptive multi-agent systems engineering. In: Engineering societies in the agents world III. Springer, pp 156–169

Birdsey L, Szabo C, Falkner K (2016) Casl: a declarative domain specific language for modeling complex adaptive systems. In: Winter simulation conference (WSC). IEEE, pp 1241–1252

Brown DS, Goodrich MA (2014) Limited bandwidth recognition of collective behaviors in bio-inspired swarms. In: Proceedings of the 13th international conference on autonomous agents and multiagent systems, pp 405–412

Chan WKV (2011a) Interaction metric of emergent behaviors in agent-based simulation. In: Proceedings of the winter simulation conference, Phoenix, USA, pp 357–368

Chan WKV (2011b) Interaction metric of emergent behaviors in agent-based simulation. In: Jain S, Jr RRC, Himmelspach J, White KP, Fu MC (eds) Winter simulation conference. WSC, pp 357–368

Chen C, Nagl SB, Clack CD (2007) Specifying, detecting and analysing emergent behaviours in multi-level agent-based simulations. In: Proceedings of the summer computer simulation conference

Chen CC, Nagl SB, Clack CD (2009) A formalism for multi-level emergent behaviours in designed component-based systems and agent-based simulations. Underst Complex Syst

Chan W, Son YS, Macal CM (2010) Simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. In: Proceedings of the winter simulation conference, pp 135–150

Chi L (2009) Transplating social capital to the online world: insights from two experimental studies. J Organ Comput Electr Comm 19:214–236

Cilliers P (1998) Complexity and postmodernism. Routledge

Davis P (2005) New paradigms and challenges. In: Proceedings of the winter simulation conference, Orlando, USA

Fayyad U, Uthurusamy R (2002) Evolving data into mining solutions for insights. ACM, Commun, p 45

Floyd S, Jacobson V (1993) The synchronization of periodic routing messages. In: Proceedings of special interest group on data communication, pp 33–44

Fromm J (2006) On engineering and emergence. arXiv preprint nlin/0601002

Gardner M (1970) The fantastic combinations of John Conway's new solitaire games. Math Games

Gershenson C, Fernandez N (2012) Complexity and information: measuring emergence, self-organization, and homeostatis at multiple scales. Complexity

Holland J (1999) Emergence, from chaos to order. Basic Books

Holland OT (2007) Taxonomy for the modeling and simulation of emergent behavior systems. In: Proceedings of the 2007 spring simulation multiconference, pp 28–35

Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal Mach Intell 15:850–863

Jacyno M, Bullock S, Luck M, Payne TR (2009) Emergent service provisioning and demand estimation through self-organizing agent communities. Proceedings of the international conference on autonomous agents and multiagent systems 1:481–488

Johnson CW (2006) What are emergent properties and how do they affect the engineering of complex systems? Reliability Engineering and System Safety 12:1475–1481

Kubik A (2003) Towards a Formalization of Emergence. Journal of Artificial Life 9:41–65

Mittal S (2013) Emergence in Stigmergic and Complex Adaptive Systems: A Formal Discrete Event Systems Perspective. Cognitive Systems Research 21:22–39

Mnif M, Müller-Schloer C (2006) Quantitative Emergence. In: IEEE Mountain Workshop on Adaptive and Learning Systems, pp 78–84, 24–26 July 2006. doi:10.1109/SMCALS.2006.250695

Mogul JC (2006) Emergent (mis)behavior vs. Complex Software Systems. In: Proceedings of the 1st ACM SIGOPS/eurosys european conference on computer systems, New York, USA, pp 293–304

Odell J (1998) Agents and emergence. Distrib Comput 51–53

Petty M, Weisel E (2003a) Basis for a theory of semantic composability. In: Proceedings of the spring simulation interoperability workshop, Orlando, USA

Petty M, Weisel EW (2003b) A composability lexicon. Proceedings of the spring simulation interoperability workshop. Orlando, USA, pp 181–187

Prokopenko M, Boschetti F, Ryan AJ (2009) An information-theoretic primer of complexity. Self-organization and emergence. Complexity 15:11–28

Ramakrishnan KK, Yang H (1994) The ethernet capture effect: analysis and solution. In: Proceedings of the IEEE local computer networks conference, Minneapolis, USA

Reynolds C (1987) Flocks, herds, and schools: a distributed behavioral model. In: Proceedings of ACM SIGGRAPH, pp 25–34

Salazar N, Rodriguez-Aguilar JA, Arcos JL, Peleteiro A, Burguillo-Rial JC (2011) Emerging cooperation on complex networks. In: Proceedings of the international conference on autonomous agents and multiagent systems, pp 669–676

Savarimuthu B, Purvis M, Cranefield S, Purvis M (2007) Mechanisms for norm emergence in multiagent societies. In: Proceedings of the 6th international joint conference on autonomous agents and multiagent systems, AAMAS'07, pp 173:1–173:3

Serugendo GDM, Gleizes MP, Karageorgos A (2006) Self-organisation and emergence in mas: an overview. Informatica (Slovenia) 30(1):45–54

Seth AK (2008) Measuring emergence via nonlinear granger causality. In: Proceedings of the eleventh international conference on the simulation and synthesis of living systems, pp 545–553

Szabo C, Teo Y (2012a) An integrated approach for the validation of emergence in component-based simulation models. In: Proceedings of the winter simulation conference, pp 2412–2423

Szabo C, Teo YM (2012b) An integrated approach for the validation of emergence in component-based simulation models. In: Proceedings of the winter simulation conference, p 242

Szabo C, Teo YM (2013) Post-mortem analysis of emergent behavior in complex simulation models. In: Proceedings of the 2013 ACM SIGSIM conference on principles of advanced discrete simulation. ACM, pp 241–252

Szabo C, Teo YM, See S (2009) A time-based formalism for the validation of semantic composability. In: Winter simulation conference, pp 1411–1422

Tang M, Mao X (2014) Information entropy-based metrics for measuring emergences in artificial societies, vol 16, pp 4583–4602. Multidisciplinary Digital Publishing Institute

Teo YM, Luong BL, Szabo C (2013) Formalization of emergence in multi-agent systems. In: Proceedings of the 2013 ACM SIGSIM conference on principles of advanced discrete simulation. ACM, pp 231–240

Tolk A (2017) Bias ex silico—observations on simulationist's regress. In: Proceedings of the spring simulation multi-conference, pp 314–322

Vail DL, Veloso MM, Lafferty JD (2007) Conditional random fields for activity recognition. In: Proceedings of the 6th international joint conference on autonomous agents and multiagent systems, AAMAS '07, pp 235:1–235:8

Yaneer B-Y (2004) A mathematical theory of strong emergence using multiscale variety. Complexity 9:15–24

Chapter 5
Input Model Risk

Eunhye Song and Barry L. Nelson

Abstract Input model risk, which is also called "input uncertainty," refers to the effect of not knowing the true, correct distributions of the basic stochastic processes that drive a computer simulation. Examples of input models include interarrival-time and service-time distributions in queueing models; bed occupancy and patient characteristic distributions in healthcare models; distributions for the values of underlying securities and assets in financial models; and time-to-failure and time-to-repair distributions in reliability models. When the input distributions are obtained by fitting to observed real-world data, then it is possible to quantify the input model risk, or to choose input models that hedge against this risk. In this chapter, we define input model risk and describe various proposals for addressing it that had their origins at the Winter Simulation Conference.

5.1 Introduction

The Winter Simulation Conference (WSC) has had a long and distinguished history of research in simulation model validation. Model validation is "the substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application" (Sargent et al. 2016). However, even a carefully validated model is an imperfect representation of reality, so decisions based on simulation models are always subject to some level of *model risk*, which we define loosely as the potential error that arises due to lack of fidelity of the simulation model. When simulation model risk can be quantified, then it is possible to reflect that risk in simulation-based decisions. Although a complete

E. Song (✉)
The Pennsylvania State University, 310 Leonhard Building,
University Park, PA 16802-4400, USA
e-mail: eunhyesong2016@u.northwestern.edu

B.L. Nelson
Northwestern University, 2145 Sheridan Road, Tech C210,
Evanston, IL 60208, USA
e-mail: nelsonb@northwestern.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations,
Methods and Applications, DOI 10.1007/978-3-319-64182-9_5

characterization of simulation model risk from all sources is not yet in sight, quantification of input model risk has been pioneered by WSC researchers since the 1990s. In this chapter, which is based on Song et al. (2014), we review the research contributions to input model risk that have occurred at WSC, and speculate about potential future advances.

Recall that the "stochastic" in stochastic simulation is a consequence of the input models that drive the simulation output. In the simplest case, the input models are fully specified univariate probability distributions from which the simulation generates independent and identically distributed (i.i.d.) random variates. More generally, the input models are stochastic processes described probabilistically (e.g., their joint distribution), rather than logically (customers are served in the order they arrive). The fidelity with which the input models capture the uncertainty in the system of interest directly affects the fidelity of the conclusions that can be drawn from the simulation experiment.

Input models may be specified a priori from process physics, culled from the experience of informed participants, or tailored to conform to real-world data. In this chapter we focus on input models that are "fit" to representative samples of real-world data. These input models are incomplete representations of reality because the data sample is necessarily finite, and this is the source of input model risk. Our interest is in quantifying or protecting against the uncertainty in the input models that propagates to the output performance measure estimates.

The chapter is organized as follows: After some background and notation in the next section, we review the process of input modeling in Sect. 5.3. Approaches for quantifying the variance induced by uncertain input models are presented in Sect. 5.4, followed by a brief overview of methods for hedging against input model risk in Sect. 5.5. The impact of input model risk on simulation optimization over a finite number of alternatives is described in Sect. 5.6 before offering conclusions. We almost exclusively cite papers presented at WSC as the key concepts and ideas typically first appeared there. For surveys on this topic that were presented at WSC, see Barton (2012), Henderson (2003), Lam (2016) and Song et al. (2014). See also Chap. 15 that addresses the impact of input uncertainty on simulation optimization with a focus on problems for which the decision variables are continuous-valued.

5.2 Background and Notation

Let $\mathbf{F}$ denote the collection of input models in the simulation (we will be more specific about this collection later). We represent the simulation output on replication j as

$$Y_j(\mathbf{F}) = \eta(\mathbf{F}) + \varepsilon_j(\mathbf{F})$$

where $\eta(\mathbf{F}) = \mathrm{E}[Y_j(\mathbf{F})]$ is the expected value of the simulation output random variable when we use input distributions $\mathbf{F}$, and $\varepsilon_j(\mathbf{F})$ is a mean 0 random variable representing the stochastic noise that is a consequence of the input models. Since the out-

put could be an average, an indicator variable, a sample variance, a sample quantile or any number of other quantities, this is a very general representation that emphasizes the dependence of the system performance measure $\eta(\mathbf{F})$ and the simulation output $Y_j(\mathbf{F})$ on the chosen input models $\mathbf{F}$.

To be concrete, in a queueing simulation we might have input models $\mathbf{F} = \{F_1, F_2\}$, where F_1 is a nonstationary Poisson arrival process, and F_2 is a service-time distribution of unknown family. The output $Y(\mathbf{F})$ is the average customer delay from 8 AM to 9 PM, and $\eta(\mathbf{F})$ its expected (mean) value, the quantity we are interested in estimating. To execute a simulation experiment, we need a specific choice or choices for $\mathbf{F}$, and the natural approach when we have real-world input data is to use a fitted distribution $\widehat{\mathbf{F}}$, which is a stand-in for the unknown true real-world distribution denoted by $\mathbf{F}^c$ (the "c" indicates "correct"). Therefore, the simulation results we generate are

$$Y_j(\widehat{\mathbf{F}}) = \eta(\widehat{\mathbf{F}}) + \varepsilon_j(\widehat{\mathbf{F}}).$$

Given the choice $\widehat{\mathbf{F}}$, there is a vast literature on simulation design and analysis that provides guidance on how many replications we need, and what estimator to use, to estimate $\eta(\widehat{\mathbf{F}})$. Clearly this analysis is *conditional* on $\widehat{\mathbf{F}}$, and therefore it ignores a potentially large source of model risk due to $\widehat{\mathbf{F}} \neq \mathbf{F}^c$.

The most basic form of the input model risk problem is to quantify how much uncertainty in the simulation-based estimator of $\eta(\mathbf{F}^c)$ *is caused by using* $\widehat{\mathbf{F}}$ *as an estimator of* $\mathbf{F}^c$. Notice that the goal is to estimate properties of the real-world system—denoted $\eta(\mathbf{F}^c)$ here—rather than properties of the simulation model, $\eta(\widehat{\mathbf{F}})$. Much of simulation output analysis has been tailored to the latter objective, not the former. Alternative input-model-risk formulations try to choose $\widehat{\mathbf{F}}$ to protect against downside consequences of decisions rather than to closely match the real-world data, and to optimize system performance when all of the alternatives are subject to input model risk (see also Chap. 15).

To set the stage, we start with a high-level discussion of input modeling itself.

5.3 Input Modeling

For ease of exposition, suppose (temporarily) that the system of interest has only one source of randomness represented by a univariate input distribution F^c. Generally F^c is unknown; however, we can observe a realization (real-world data) $\mathbf{x}_m = \{x_1, x_2, \ldots, x_m\}$ of the i.i.d. input process $X_1, X_2, \ldots, X_m \sim F^c$. We use capital letters such as X to denote a random variable, and lowercase letters such as x for a realization. The estimated distribution $\widehat{F}$ depends on any prior information we have about F^c and the observed data $\mathbf{x}_m$. Choosing $\widehat{F}$ is called *input modeling*.

The goal of the simulation experiment is to precisely estimate the true system performance measure, which is $\eta(F^c)$ in our set up. Therefore, estimating F^c precisely

is a sufficient, but not a necessary condition; to achieve our goal we need $\eta(\widehat{F})$ to be close to $\eta(F^c)$, not $\widehat{F} = F^c$. In some specialized situations, we know what specific properties of $\widehat{F}$ need to be close to the corresponding properties of F^c for $\eta(\widehat{F})$ to be close to $\eta(F^c)$—the steady-state mean delay in an M/G/1 queue depends only on the mean and variance of the service-time distribution, for instance—but this is certainly not possible in general because $\eta(\cdot)$ is implicit in the simulation logic and therefore unknown. As a consequence, the focus of input modeling is typically on the fidelity of $\widehat{F}$ with respect to F^c. The two main philosophies of input modeling are classical and Bayesian, and while certainly different, they share some common features when it comes to input model risk.

The classical approach is to assume that there exists a true distribution F^c, and the goal is to infer F^c from the real-world data using methods that have good performance when averaged over the possible samples $X_1, X_2, \ldots, X_m$. For instance, if F^c is believed to have a known distribution family but unknown parameters θ^c—that is, $F^c(x) = F(x \mid \theta^c)$ with F known—then the parameters may be estimated using maximum likelihood estimators (MLEs), least squares, generalized method of moments, or other methods. Typically, these estimators are asymptotically consistent for θ^c as $m \to \infty$. Prior information such as the physical basis of the input process may influence the choice of distribution family. There is commercial input modeling software that fits the parameters of a large collection of distribution families and also provides ways to assess the fit, including goodness-of-fit tests and likelihood-based rankings. Errors in fitting are characterized by the sampling distribution of the parameter estimator $\widehat{\theta}$; since this distribution is often hard to derive, approximations based on large-sample asymptotics or bootstrapping are often used. If there is not enough evidence that F^c has a certain parametric family, then an empirical cumulative distribution function (ecdf) can be employed, which converges to F^c asymptotically. Cheng (1994) presents a tutorial on input modeling with an eye toward quantifying input model risk.

Bayesians are less interested in discovering the true distribution F^c and more interested in the likelihood of possible choices given the data. Bayesian inference starts by capturing any knowledge available about F^c in the form of a prior probability distribution; this distribution may be on the family of distributions, the parameters of a specific family of distributions, or both. For instance, we might believe that F^c is one of the families in the set $\{\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_k\}$ with prior probabilities $\pi_1, \pi_2, \ldots, \pi_k$. More often, we assume that the family of F^c is known and provide a prior $\pi_{\Theta}(\theta)$ on the value of its parameter vector $\boldsymbol{\Theta}$, which is treated as a random vector.

After collecting the real-world data $\mathbf{x}_m$, the prior distribution $\pi_{\Theta}(\theta)$ is updated to a posterior distribution $p_{\Theta}(\theta \mid \mathbf{x}_m)$ according to Bayes' rule

$$p_{\Theta}(\theta \mid \mathbf{x}_m) \propto \mathscr{L}(\mathbf{x}_m \mid \theta)\pi_{\Theta}(\theta),$$

where $\mathscr{L}(\mathbf{x}_m \mid \theta)$ is the likelihood function of $\mathbf{x}_m$ given $\boldsymbol{\Theta} = \theta$. Notice that the posterior distribution is conditional on the particular realization $\mathbf{x}_m$ that was obtained. The posterior distribution $p_{\Theta}(\theta \mid \mathbf{x}_m)$ plays the role for Bayesians that the sampling

distribution of $\widehat{\theta}$ plays in the classical approach by quantifying the uncertainty about θ^c. Under some regularity conditions, the posterior distribution of Θ converges to a degenerate distribution at θ^c independent of the choice of the prior distribution as the sample size $m \to \infty$; this is the Bayesian concept of consistency. See Chick (1997, 1999) for Bayesian input modeling to facilitate evaluation of input model risk.

Returning to the main purpose of the simulation study—to make decisions based on an estimate of $\eta(F^c)$—both classical and Bayesian input modelers face a similar problem: actually executing a simulation experiment requires a specific choice of distribution and distribution parameters. When only the parameters are unknown, we often use the single "best" choice, such as the MLE in the classical paradigm or the maximum a posteriori (MAP) estimator for Bayesians, but a single choice assumes away the very real uncertainty about the input model. However, since $\eta(\cdot)$ is unknown, characterizing uncertainty about the input model (via a sampling or posterior distribution) is not enough since the goal is to measure the uncertainty in estimating $\eta(F^c)$. Therefore, to quantify input model risk, some way to propagate the uncertainty about $\widehat{F}$ to the estimate of $\eta(F^c)$ is required.

An alternative to *fitting* an input model $\widehat{F}$ that appears to best represent the real-world data from F^c and then to propagate the error, is to *search* for an input model $\tilde{F}$ that is *plausible* given the real-world data from F^c but leads to worst-case system performance among all plausible input models. This approach hedges against model risk rather than trying to quantify it, and directly emphasizes the output performance measure rather than the input model. The definition of "plausible" and the search within the plausible set are the challenges for this approach.

5.4 Quantification of Input Model Risk

We start with the goal of measuring the input model risk, when we fit $\widehat{\mathbf{F}}$ to estimate F^c and use it in the simulation.

5.4.1 A Measure of Input Model Risk

One difficulty in describing input uncertainty is that there are many types of input models that may arise in a computer simulation. These include univariate inputs such as customer service times; time series inputs such as the weekly demands for milk; multivariate inputs such as the age, income, gender, and occupation of a customer; and time-dependent inputs such as the arrival times of calls to a call center. For exposition, we assume that the input models for the simulation consist of L independent, univariate distributions $\mathbf{F} = \{F_1, F_2, \ldots F_L\}$ each of whose role is to model an i.i.d. input process. For instance, in a stationary queueing system with server failures we might have $\mathbf{F} = \{F_1, F_2, F_3, F_4\}$, the distributions of interarrival times, service times,

time until server failure, and server repair time, respectively. Let $Y(\mathbf{F})$ be a random variable that represents the generic output of the simulation—such as the average customer delay in the queueing system with failures—when the input distributions are $\mathbf{F}$.

We work with the premise that there are unknown, but "correct," real-world distributions $\mathbf{F}^c = \{F_1^c, F_2^c, \ldots, F_L^c\}$. From the ℓth distribution, we are able to observe m_ℓ i.i.d. real-world observations $X_{\ell 1}, X_{\ell 2}, \ldots, X_{\ell m_\ell}$ from which to "fit" F_ℓ^c, giving the collection of fitted input models $\widehat{\mathbf{F}} = \{\widehat{F}_1, \widehat{F}_2, \ldots, \widehat{F}_L\}$. We then run the simulation and obtain outputs $Y_1(\widehat{\mathbf{F}}), Y_2(\widehat{\mathbf{F}}), \ldots, Y_n(\widehat{\mathbf{F}})$ across n i.i.d. replications. *The goal of the simulation experiment is to estimate* $\eta(\mathbf{F}^c) = \mathrm{E}[Y(\mathbf{F}^c)]$. Again for ease of exposition, we assume that $\eta(\mathbf{F}^c)$ is estimated by the sample mean across n replications

$$\bar{Y}(\widehat{\mathbf{F}}) = \frac{1}{n}\sum_{j=1}^{n} Y_j(\widehat{\mathbf{F}}) = \eta(\widehat{\mathbf{F}}) + \frac{1}{n}\sum_{j=1}^{n} \varepsilon_j(\widehat{\mathbf{F}}).$$

The variance of the point estimator is

$$\begin{aligned}\mathrm{Var}[\bar{Y}(\widehat{\mathbf{F}})] &= \mathrm{Var}[\mathrm{E}(\bar{Y}(\widehat{\mathbf{F}}) \mid \widehat{\mathbf{F}})] + \mathrm{E}[\mathrm{Var}(\bar{Y}(\widehat{\mathbf{F}}) \mid \widehat{\mathbf{F}})] \\ &= \mathrm{Var}[\eta(\widehat{\mathbf{F}})] + \frac{1}{n}\mathrm{E}[\mathrm{Var}(\varepsilon_1(\widehat{\mathbf{F}}) \mid \widehat{\mathbf{F}})], \end{aligned} \tag{5.1}$$

where the outer variance and expectation on the right-hand side are with respect to the sampling distribution of $\widehat{\mathbf{F}}$ (classical) or the posterior distribution of $\widehat{\mathbf{F}}$ (Bayesian). Notice that the second term in (5.1) accounts for the inherent variability in the simulation given a choice of input distributions, and it can be driven to 0 by increasing the number of simulation replications n. The size of the first term, however, depends on a complex interaction between the real-world sample sizes $\mathbf{m} = \{m_1, m_2, \ldots, m_L\}$ and the structure of $\eta(\cdot)$.

One definition of input model risk is $\sigma_I^2 = \mathrm{Var}[\eta(\widehat{\mathbf{F}})]$, *the variance in the system mean due to having estimated* $\mathbf{F}^c$. Although this definition is straightforward, estimating σ_I^2 is not as it requires (a) the distribution of $\widehat{\mathbf{F}}$ and (b) the functional form of $\eta(\cdot)$, neither of which is known in general. This illustrates why assessing input model risk is difficult and motivates the methods that have been suggested at WSC and are reviewed below.

Conceptually, if we had B independent real-world samples of input data of size $\mathbf{m}$, yielding B independent fitted input distributions $\widehat{\mathbf{F}}_1, \widehat{\mathbf{F}}_2, \ldots, \widehat{\mathbf{F}}_B$, and we knew the true mean response functional $\eta(\cdot)$, then we could estimate σ_I^2 via the unbiased estimator

$$\widehat{\sigma}_I^2 = \frac{1}{B-1}\sum_{b=1}^{B}(\eta(\widehat{\mathbf{F}}_b) - \bar{\eta})^2, \text{ where } \bar{\eta} = \frac{1}{B}\sum_{b=1}^{B}\eta(\widehat{\mathbf{F}}_b).$$

However, if we actually had multiple real-world data sets then we would certainly pool them to obtain a better estimator of $\mathbf{F}^c$; thus, no matter how much data we have

it is best to treat it as one sample. In addition, $\eta(\cdot)$ is unknown or we would not be simulating.

It is worth mentioning that one other aspect of input uncertainty that is rarely noted, even in the input uncertainty literature, is bias. Specifically,

$$\mathrm{E}[\eta(\widehat{\mathbf{F}}) - \eta(\mathbf{F}^c)] \neq 0$$

in general. This is the case even when $\mathrm{E}[\widehat{\mathbf{F}}] = \mathbf{F}^c$ pointwise because $\eta(\cdot)$ is typically a nonlinear transformation. We might hope that $\eta(\widehat{\mathbf{F}}) \to \eta(\mathbf{F}^c)$ in some sense as the real-world sample sizes $m_\ell \to \infty$, $\ell = 1, 2, \ldots, L$, but this is the most we can hope. Thus, the ideal measure of input model risk is $\mathrm{MSE}[\eta(\widehat{\mathbf{F}})]$, which combines both variance and bias, but the published work focuses on variance.

5.4.2 *Variability of* $\widehat{\mathbf{F}}$

In the classical setting, the variability of $\widehat{\mathbf{F}}$ as an estimator of $\mathbf{F}^c$ is quantified by its sampling distribution; that is, we try to infer the distribution of possible $\widehat{\mathbf{F}}$'s that could be realized from the single sample of data that we actually have. There have been two primary approaches in the literature: large-sample asymptotic distributions as in Cheng (1994), and bootstrapping as in Ankenman and Nelson (2012), Barton and Schruben (1993, 2001), Cheng (2001), and Song and Nelson (2013).

To simplify the exposition, suppose again that there is only $L = 1$ input distribution, and we believe $F^c(x) = F(x \mid \boldsymbol{\theta}^c)$ is a parametric distribution with known family (e.g., Poisson), but unknown parameter vector $\boldsymbol{\theta}^c$. The family might be "known" because of process physics (e.g., an arrival process consisting of the superposition of many customers making independent but infrequent decisions about when to arrive tends to be Poisson), or from fitting a large number of possible parametric families and ranking their goodness of fit. For the parametric case, there is a well-developed theory for parameter estimators $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(X_1, X_2, \ldots, X_m)$ such as MLE. The sampling distributions for these estimators are known, at least asymptotically as $m \to \infty$. For instance, MLEs are often asymptotically normally distributed. Thus, for parametric distributions, input *model* uncertainty becomes input *parameter* uncertainty which is quantified by the sampling distribution of $\widehat{\boldsymbol{\theta}}$. Cheng (1994) (and follow-on papers) provides an expression for σ_I^2 by applying a first-order Taylor series approximation of $\eta(\widehat{\theta})$ to obtain

$$\sigma_I^2 = \mathrm{Var}[\eta(\widehat{\theta})] \approx \nabla\eta(\theta^c)^\top \mathrm{Var}[\widehat{\theta}]\nabla\eta(\theta^c) \tag{5.2}$$

where $\mathrm{Var}[\widehat{\theta}]$ is the variance–covariance matrix of $\widehat{\theta}$ and $\nabla\eta(\theta^c)$ is the gradient of $\eta(\cdot)$ evaluated at θ^c. The approximation (5.2) shows that input uncertainty depends both on how well the input model has been estimated, and how sensitive the system response is to the input model. The sampling distribution (or an approximation to it)

can provide an estimator of $\text{Var}[\widehat{\theta}]$, but that still leaves the question of how to estimate $\nabla\eta(\theta^c)$.

In bootstrapping, we assume that the fitted distribution $\widehat{F}$ is a good stand-in for F^c. An empirical cumulative distribution function (ecdf) is often used as the fitted distribution, although this is not required. Then instead of gathering B distinct real-world samples of input data, we simulate B bootstrap samples from the fitted distribution:

$$X_1^{\star(b)}, X_2^{\star(b)}, \ldots, X_m^{\star(b)} \overset{i.i.d.}{\sim} \widehat{F} \text{ for } b = 1, 2, \ldots, B.$$

Each bootstrap sample yields a fitted input distribution $\widehat{F}^{\star(b)}$. Therefore, an estimator of σ_I^2 is

$$\widehat{\sigma}_I^{2\star} = \frac{1}{B-1}\sum_{b=1}^{B}(\eta(\widehat{F}^{\star(b)}) - \bar{\eta}^{\star})^2$$

where $\bar{\eta}^{\star} = \sum_{b=1}^{B}\eta(\widehat{F}^{\star(b)})/B$. The validity of the bootstrap can be established as $m \to \infty$. However, this estimator requires a stand-in for $\eta(\cdot)$.

Bayesian input modeling typically emphasizes the choice of parameters for a parametric distribution, although as noted earlier it is possible to extend the definition of "parameter" to include the distribution family as well. The posterior distribution of the parameters $p_{\boldsymbol{\Theta}}(\boldsymbol{\theta} \mid \mathbf{x}_m)$ plays the role of the sampling distribution of $\widehat{\boldsymbol{\theta}}$ in the classical approach. However, in a Bayesian treatment there is no need for an asymptotic approximation because the posterior is valid for any quantity of real-world data. In fact, the posterior explicitly accounts for the quantity of data that is available: the more data there is, the more concentrated the posterior distribution will be.

From a Bayesian perspective, the corresponding overall measure of input uncertainty is $\sigma_I^2 = \text{Var}[\eta(\Theta)]$ where the variance is with respect to $\Theta \sim p_{\Theta}(\theta \mid \mathbf{x}_m)$. One difficulty in computing this quantity arises when the posterior distribution is not simple and can only be evaluated approximately via simulation methods, such as Markov chain Monte Carlo. And, of course, $\eta(\cdot)$ is not known. Biller and Gunes (2010) provide a Bayesian approach that allows for multivariate input models.

5.4.3 Propagating Input Model Uncertainty

The second reason that $\widehat{\sigma}_I^2$ is not realizable is that $\eta(\cdot)$ is unknown, and it is this mapping that propagates the input model uncertainty to the simulation performance measure uncertainty. Whether classical or Bayesian, there is really no choice but to estimate $\eta(\cdot)$ using simulation, and we can only estimate $\eta(\mathbf{F})$ given specific choices of $\mathbf{F}$. The key questions are, what should we estimate, and how should we expend the simulation effort to do it?

Direct simulation chooses a collection of distributions $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_B$, runs simulations at each choice, and estimates $\eta(\mathbf{F}_i)$ by the average of the simulation results $\bar{Y}(\mathbf{F}_i)$. The distributions $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_B$ are usually randomly chosen from the

sampling distribution of $\widehat{\theta}$ (classical) or the posterior distribution of $\boldsymbol{\Theta}$ (Bayesian) so as to represent the input model uncertainty. Barton and Schruben (1993) was the first WSC paper to suggest this.

1. For $b = 1$ to B
   a. If classical:
      i. Generate $\widehat{\theta}_b^\star$ from the sampling distribution of $\widehat{\theta}$ (e.g., using the asymptotic distribution or bootstrapping).
      ii. Using $\widehat{\theta}_b^\star$, run n replications to obtain the sample mean $\bar{Y}_b(\widehat{\theta}_b^\star)$.
   b. Else if Bayesian:
      i. Generate $\boldsymbol{\Theta}_b \sim p_{\boldsymbol{\Theta}}(\theta \mid \mathbf{z}_m)$
      1i. Using $\boldsymbol{\Theta}_b$, run n replications to obtain the sample mean $\bar{Y}_b(\boldsymbol{\Theta}_b)$.

   Next b
2. Use the data $\bar{Y}_b(\widehat{\theta}_b^\star)$ or $\bar{Y}_b(\boldsymbol{\Theta}_b)$ for $b = 1, 2, \ldots, B$ to estimate measures of input model risk.

Notice that in either case input uncertainty is confounded with stochastic simulation noise so they may need to be separated, which is the challenge. Yi et al. (2015) refine the direct method above by using a sequentially designed experiment rather than equal simulation replications on all bootstrap samples.

Metamodeling chooses distributions $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_d$ to cover a relevant design space, and then uses the simulation results to fit a metamodel $\widehat{\eta}(\cdot)$ to stand in for $\eta(\cdot)$. Barton et al. (2010) take a classical approach to input modeling and use Gaussian processes to build a metamodel at a chosen set of parameters, while Song and Nelson (2013) employ a linear regression on the mean and variance of each input distribution.

Given the metamodel, $\widehat{\sigma}_I^2$ or a number of other measures can be estimated inexpensively. The metamodel-assisted approach, both classical and Bayesian, has been extended to multivariate input distributions by Xie et al. (2014, 2015) and to nonstationary Poisson arrival processes by Morgan et al. (2016).

Gradients of $\eta(\boldsymbol{\theta})$ are needed if we use parametric input models and approximate $\text{Var}[\eta(\widehat{\boldsymbol{\theta}})]$ using a Taylor series expansion of $\eta(\widehat{\boldsymbol{\theta}})$ around $\boldsymbol{\theta}^c$ as in (5.2). Since $\eta(\boldsymbol{\theta})$ is unknown, its gradients are also unknown. There are several methods to estimate $\nabla\eta(\theta^c)$. The expansion in (5.2) facilitates propagating uncertainty about $\boldsymbol{\theta}$ (as represented by the sampling distribution of $\widehat{\boldsymbol{\theta}}$ or the posterior distribution of $\boldsymbol{\Theta}$) to the performance measure $\eta(\widehat{\boldsymbol{\theta}})$. Clearly this is another form of metamodeling.

5.4.4 Confidence and Credible Intervals

Interval estimates are also valuable. The typical simulation confidence interval (CI) guarantees $(1 - \alpha)100\%$ coverage of $\eta(\widehat{\mathbf{F}})$. Instead, an interval $[C_L, C_U]$ that guarantees

$$\Pr\{\eta(\mathbf{F}^c) \in [C_L, C_U]\} \approx 1 - \alpha$$

accounts for input model uncertainty, simulation error and bias. The corresponding Bayesian credible interval (CrI) would assert

$$\Pr\{\eta(\boldsymbol{\Theta}) \in [Q_L, Q_U] \mid \mathbf{x_m}\} = 1 - \alpha.$$

This interval contains $1 - \alpha$ of the probability content of the induced posterior distribution on $\eta(\boldsymbol{\Theta})$ when $\boldsymbol{\Theta} \sim p_{\boldsymbol{\Theta}}(\theta \mid \mathbf{x_m})$.

Many papers have proposed generating CIs or CrIs from direct simulation, as described above, using empirical quantiles of $\bar{Y}_b(\widehat{\theta}_b^\star)$ or $\bar{Y}_b(\boldsymbol{\Theta}_b)$, respectively, for $b = 1, 2, \ldots, B$. See Yi et al. (2015) for a direct bootstrapping approach, and Lam and Qian (2016) for a CI based on empirical likelihood.

A different sort of interval estimate is provided by Batarseh and Wang (2008): they use interval arithmetic to propagate likely intervals for each input distribution parameter individually to an interval on the output performance measure.

So far we have only considered the measures of input uncertainty as a function of $\widehat{\mathbf{F}}$, the collection of all input models used in the simulation. We next consider assessing which input models among $\widehat{\mathbf{F}}$ make the greatest contributions to input model risk, and from which would we most benefit from collecting more real-world data.

5.4.5 *Measures of Contribution and Sample-Size Sensitivity*

Contribution measures try to decompose σ_I^2 in a meaningful way. The ultimate goal is to identify from which distributions it would be most beneficial to reduce uncertainty further, or even to specify how to spend a budget for additional data collection.

Clearly, the contribution of ℓth input model $\widehat{F}_\ell$ depends on the real-world sample size m_ℓ; as m_ℓ becomes infinitely large, $\widehat{F}_\ell$ converges to F_ℓ^c (classical) or to a degenerate posterior distribution (Bayesian). Therefore, small m_ℓ causes bigger contribution. However, the contribution also depends on how sensitive $\eta(\cdot)$ is to F_ℓ. To make the point clear, suppose that F_1 is in fact a dummy distribution so that the functional $\eta(\cdot)$ does not depend on F_1. Then, no matter how small m_1 is, the contribution of $\widehat{F}_1$ should be 0. On the other hand, if $\eta(\cdot)$ is very sensitive to F_1, the uncertainty in $\widehat{F}_1$ is amplified as it is propagated to the uncertainty in $\eta(\widehat{\mathbf{F}})$.

Song and Nelson (2013) and Song et al. (2014) defined the contribution of $\widehat{F}_\ell$ to input uncertainty as

$$\mathsf{V}_\ell(m_\ell) \equiv \mathrm{Var}\left[\mathrm{E}[Y(F_1^c, F_2^c, \ldots, F_{\ell-1}^c, \widehat{F}_\ell, F_{\ell+1}^c, \ldots, F_L^c) \mid \widehat{F}_\ell]\right]. \tag{5.3}$$

In words, $\mathsf{V}_\ell(m_\ell)$ is the variability in the simulation's expected value when all of the true input distributions except F_ℓ^c are known and F_ℓ^c is estimated by $\widehat{F}_\ell$. Notice that

V_ℓ is a function of the sample size m_ℓ; the larger m_ℓ is, the smaller $\mathsf{V}_\ell(m_\ell)$ becomes as $\widehat{F}_\ell$ approaches F_ℓ^c.

Unfortunately, $\sum_{\ell=1}^{L} \mathsf{V}_\ell(m_\ell) \neq \sigma_I^2$ in general. Song and Nelson (2013) and Song et al. (2014) suggest looking instead at the relative contribution of the ℓth input model,

$$\frac{\mathsf{V}_\ell(m_\ell)}{\sum_{i=1}^{L} \mathsf{V}_i(m_i)}.$$

A reasonable heuristic to reduce σ_I^2 is to obtain more data for the distributions with the largest relative contribution. See Biller and Gunes (2010) and Morgan et al. (2016) for decompositions in multivariate-input-model settings.

Song and Nelson (2013, 2014) also propose a measure that does not decompose σ_I^2, but instead quantifies the impact of additional data for each input distribution: the sample-size sensitivity of $\mathrm{Var}[\bar{Y}(\widehat{\mathbf{F}})]$ with respect to ℓth input model. If we approximate m_ℓ as real valued then this is

$$\left.\frac{\partial \mathrm{Var}[\bar{Y}(\widehat{\mathbf{F}})]}{\partial m'_\ell}\right|_{m'_\ell = m_\ell}. \tag{5.4}$$

The sample-size sensitivity quantifies how much the estimator variance would be reduced by observing one more real-world sample from the ℓth input distribution given we already have m_ℓ observations. The input distributions with the largest (most negative) sensitivities are targets for more real-world data. Unfortunately, sample-size sensitivities are only local gradients, and therefore are not ideal for optimally allocating a substantial amount of additional effort. Freimer and Schruben (2002) attempt to go further.

If the budget for real-world data is effectively unlimited, then it makes sense to collect enough data so that $\mathrm{Var}[\bar{Y}(\widehat{\mathbf{F}})] \approx \mathrm{E}[\mathrm{Var}(\bar{Y}(\widehat{\mathbf{F}}) \mid \widehat{\mathbf{F}})]$; i.e., no input model risk. Freimer and Schruben (2002) achieve this objective by sequentially adding real-world data until the effect of input model uncertainty is statistically negligible relative to simulation output variability. To formalize this, they use a random-effects metamodel where the "random effects" are due to distribution parameter estimates obtained from bootstrapping the available data. In the case of $L = 2$ parameters, their model for the simulation output on replication h is

$$Y_{ijh} = \eta + A_i + C_j + AC_{ij} + \varepsilon_{ijh}, \quad \text{for } i,j = 1,2,\ldots,B,\ h = 1,2,\ldots,n \tag{5.5}$$

where A_i, C_j and AC_{ij} are the random effects of bootstrapped parameters $\theta_{1i}^\star$, $\theta_{2j}^\star$ and their interaction, respectively, B is the number of bootstrap samples, and ε_{ijh} is the stochastic variability of the simulation. They stop collecting real-world input data when the hypothesis that these three variance effects are 0 is no longer rejected.

5.5 Hedging Against Input Model Risk

Suppose that large values of $\eta(\mathbf{F})$ are problematic, for instance if $\eta(\mathbf{F})$ is the mean delay in queue or portfolio value at risk. Then one might prefer to run the simulation with input model $\tilde{\mathbf{F}}$, where

$$\tilde{\mathbf{F}} = \text{argmax}_{\mathbf{F}\in\mathsf{F}}\ \eta(\mathbf{F}) \tag{5.6}$$

and F is referred to as the *uncertainty* or *ambiguity* set that contains distributions plausible for $\mathbf{F}^c$ given the observed real-world data $\mathbf{x}_m$. This is an appealing approach because it protects against the downside risks that matter to the user. When F is designed to include plausible candidates for $\mathbf{F}^c$ with $1-\alpha$ probability, then we can form a confidence or credible interval for $\eta(\mathbf{F}^c)$ by

$$\left(\min_{\mathbf{F}\in\mathsf{F}}\ \eta(\mathbf{F}), \max_{\mathbf{F}\in\mathsf{F}}\ \eta(\mathbf{F})\right).$$

There have been a number of very recent WSC papers on this approach, including Ghosh and Lam (2015), Hu and Hong (2015), Lam and Ghosh (2013), Vidyashankar and Xu (2015) and Zhu and Zhou (2015). The challenges are defining the plausible set F in a way that is both practically meaningful, and that facilitates solving the optimization problem (5.6).

5.6 Simulation Optimization Under Input Model Risk

Thus far we have discussed quantifying input model risk of a single simulated system, but simulation is often used to select a system (or solution) with the best performance, e.g., cost-minimizing staff allocation in a service center, delay-minimizing bed assignment at a hospital, and profit-maximizing portfolio of assets. Needless to say, uncertain inputs models pose a risk of selecting a suboptimal real-world system because the performance of each feasible solution under the true distribution $\mathbf{F}^c$ is unknown.

Selecting the best performing system under input model risk requires quantifying the joint impact of input model risk on the simulation outputs of the systems we compare. Although the methods discussed in Sect. 5.4 can still be applied to measure the variability of $\widehat{\mathbf{F}}$ and propagate it to the simulation outputs of all systems, the optimization problem is more subtle. In this chapter, we limit our discussion to optimization problems with a finite number of discrete solutions, deferring discussion on continuous simulation optimization to Chap. 15.

To set up the discussion, suppose the goal of a simulation optimization is to find a solution that has the largest mean performance of a common output among k solutions. In other words, the objective is to find

$$i^c = \arg\max_{1 \le i \le k} \eta_i(\mathbf{F}_i^c),$$

where $\mathbf{F}_i^c$ represents the true input distributions for the ith solution (slightly abusing notation). Note that the true input distribution, $\mathbf{F}_i^c$, (and its estimator $\widehat{\mathbf{F}}_i$) may differ for each feasible solution i. Simulation optimization procedures typically try to select $\arg\max_{1 \le i \le k} \eta_i(\widehat{\mathbf{F}}_i)$, ignoring input model risk when $\mathbf{F}_i^c$ is modeled by $\widehat{\mathbf{F}}_i$.

Recently, there have been several papers published at WSC that tackle accounting for input model risk in simulation optimization, applying one of two different philosophical approaches to answer the question, "What does it means to be 'optimal' when there is input model risk?"

The first approach is to obtain the best statistical inference one can on the true optimal solution i^c given the finite amount of real-world input data. Corlu and Biller (2013) propose a subset selection procedure that finds a set of solutions whose output means are within $\delta > 0$ from i^c's. Using Bayesian posteriors on the input models, they propagate uncertainty in the parameters Θ to the k systems' simulation outputs using the delta method (Taylor series expansion).

Song et al. (2015) suggest an indifference-zone (IZ) ranking and selection procedure given IZ parameter $\delta > 0$ that guarantees a $1 - \alpha$ *average* probability of correctly selecting i^c over the sampling distribution of $\widehat{\mathbf{F}}_i$, where a mixed effects model is used to capture the joint and independent impact of $\widehat{\mathbf{F}}_i$ and simulation error to the simulation output. Under the assumption $\mathbf{F}_i^c = \mathbf{F}^c$ for all i, a version of their mixed effects model simplifies to the following representation of the output of the jth simulation replication of the ith system:

$$Y_{ij}(\widehat{\mathbf{F}}) = \eta_i(\mathbf{F}^c) + \gamma_i + \beta(\widehat{\mathbf{F}}) + (\eta\beta)_i(\widehat{\mathbf{F}}) + \varepsilon_{ij}(\widehat{\mathbf{F}}), \tag{5.7}$$

where $\gamma_i = \mathrm{E}[Y_{ij}(\widehat{\mathbf{F}})] - \eta_i(\mathbf{F}^c)$, $\beta(\widehat{\mathbf{F}}) \sim (0, \sigma_\beta^2), (\eta\beta)_i(\widehat{\mathbf{F}}) \sim \mathrm{N}(0, (k-1)/k\sigma_{\eta\beta}^2)$, and $\varepsilon_{ij}(\widehat{\mathbf{F}}) \sim \mathrm{N}(0, \sigma_i^2)$. In words, $\eta_i(\mathbf{F}^c) + \gamma_i$ represents the fixed effect of the ith system and $\beta(\widehat{\mathbf{F}})$ represents the random effect of $\widehat{\mathbf{F}}$. Their interaction effect is captured by $(\eta\beta)_i(\widehat{\mathbf{F}})$ with an additional assumption that $\sum_{i=1}^k (\eta\beta)_i(\widehat{\mathbf{F}}) = 0$, i.e., the interaction effects can be interpreted as fixed effects conditional on $\widehat{\mathbf{F}}$. Ignoring the finite sample biases, under Model (5.7)

$$\bar{Y}_i(\widehat{\mathbf{F}}) - \bar{Y}_{i'}(\widehat{\mathbf{F}}) = \eta_i(\mathbf{F}^c) - \eta_{i'}(\mathbf{F}^c) + (\eta\beta)_i(\widehat{\mathbf{F}}) - (\eta\beta)_{i'}(\widehat{\mathbf{F}}) + \bar{\varepsilon}_i(\widehat{\mathbf{F}}) - \bar{\varepsilon}_{i'}(\widehat{\mathbf{F}})$$

for $i \neq i^c$. Notice that the random effect, $\beta(\widehat{\mathbf{F}})$, cancels out as it only depends on $\widehat{\mathbf{F}}$, and therefore the input model risk is completely captured by the interaction effect terms, $(\eta\beta)_i(\widehat{\mathbf{F}}) - (\eta\beta)_{i'}(\widehat{\mathbf{F}})$. Hence, the larger $\mathrm{Var}[(\eta\beta)_i(\widehat{\mathbf{F}}) - (\eta\beta)_{i'}(\widehat{\mathbf{F}})]$ is, the more difficult it is to compare $\eta_i(\mathbf{F}^c)$ and $\eta_{i'}(\mathbf{F}^c)$. Consistent with this observation, Song et al. (2015) show that the average probability of correct selection defined as $\mathrm{E}\left[\Pr\{\bar{Y}_{i^c}(\widehat{\mathbf{F}}) \ge \bar{Y}_i(\widehat{\mathbf{F}}), \forall i \neq i^c | \widehat{\mathbf{F}}\}\right]$ of a single-stage ranking and selection procedure is a decreasing function of the interaction effect's variance, $\sigma_{\eta\beta}^2$.

Both Corlu and Biller (2013) and Song et al. (2015) note that δ in their procedures has a nonzero lower bound when there is input model risk below which it is not possible to statistically separate i^c from the rest of the systems if their mean performances are too similar. *Hence, input model risk may dominate the mean differences.* This lower bound is an increasing function of $1-\alpha$, which implies that we may either guarantee i^c with high probability assuming δ is large or provide a poor probability guarantee with small δ when input model risk is substantial.

In a newer subset selection procedure, Corlu and Biller (2015) drop δ and instead provide a set including $\bar{i} = \arg\max_{1\le i\le k} \mathrm{E}[\eta_i(\Theta^i)|\mathbf{x}_{\mathbf{m}_i}]$, where the expectation is with respect to the posterior distribution of input distribution parameters Θ^i given the real-world input data $\mathbf{x}_{\mathbf{m}_i}$ of the ith solution using direct simulation. Note that $\bar{i} \neq i^c$ in general given finite observations $\mathbf{x}_{\mathbf{m}_i}$.

The second approach finds a solution that best hedges input model risk, i.e., $\underline{i} = \arg\max_{1\le i\le k} \min_{\widehat{\mathbf{F}}\in \mathsf{F}_i} \eta_i(\widehat{\mathbf{F}}_i)$, which is closely related to distributionally robust optimization (DRO) (see Chap. 15 and Sect. 5.5).

Fan et al. (2013) focus on the case where $\mathbf{F}_i^c = \mathbf{F}^c$ for all i and create a two-layer ranking and selection procedure whose uncertainty set F is a discrete set of choices for $\widehat{\mathbf{F}}$ induced from data. In the first layer, this procedure selects the worst-case distribution for each solution i within F, while in the second layer it compares the k solution-worst-case-distribution pairs to find $\underline{i}$. Gao et al. (2016) propose a robust optimal computing budget allocation scheme for the same problem that maximizes a lower bound on the probability of correctly selecting $\underline{i}$. Both procedures may suffer from choosing a very conservative $\underline{i}$ that performs much worse than i^c under $\mathbf{F}^c$. To mitigate such conservatism, the uncertainty set F needs to be designed carefully.

5.7 Applications of Input Model Risk Quantification

Several papers have been presented at WSC that apply tools for quantifying, or optimizing in the presence of, input model risk for a specific application domain: water allocation (Love and Bayraksan 2013); assemble-to-order production systems (Akcay and Biller 2014); and contract-manufacturer selection in the pharmaceutical industry (Akcay and Martagan 2016). In addition, Song et al. (2014) report on an implementation of the methods for input uncertainty quantification, contribution and sample-size sensitivity from Song and Nelson (2013) in the simulation software Simio (http://www.simio.com).

5.8 Conclusions and Future Directions

In this chapter, we described input model risk and investigated different ways to measure it or protect against it. In the history of WSC, various methods have been developed to measure input model risk for both traditional and Bayesian input mod-

els. Alternatively, several papers propose to choose an input model that hedges the risk given the observations recognizing the resemblance between input model risk quantification and the DRO problem. There have been recent advances in simulation optimization under input model risk that either try to make the best inference on the true optimal solution under $\mathbf{F}^c$ given observations, or find the system that best hedges input model risk given a set of plausible input models.

Simulation-assisted decision-making under input model risk is a ubiquitous and practical problem in many applications. Thus, advancing the frontier of this research area has a significant potential impact on real-world problems. In the remainder of this section, we discuss some of the open research questions.

First, there are popular input models, such as nonparametric or time series, for which it is difficult to measure input model risk with the existing methods. For instance, it is difficult to propagate uncertainty in the inputs generated from the Markov Chain Monte Carlo (MCMC) method using the Taylor series approximation as we do not have a closed-form expression for the parameters of the limiting distribution fitted using the data. We may still use the direct simulation approach, however, it may require large simulation effort to obtain multiple batches of inputs via MCMC.

Allocating additional real-world data collection effort to decrease input model risk when there are several input processes is an important problem in simulation optimization. In Sect. 5.4.5, we discussed methods to reduce input model risk of a single simulated system. When there are several systems to compare, the allocation problem becomes more complex as we need to measure the impact of additional data collection from each source on the optimality guarantee.

In general, simulation optimization under input model risk suffers conservatism that cannot be overcome by simply increasing simulation effort, which is the biggest difference from the case with known $\mathbf{F}^c$. As discussed in Sect. 5.6, such conservatism may manifest as a low probability guarantee on the optimal solution, low quality of optimality (due to a large IZ parameter δ), or selecting a system that hedges input model risk well but performs poorly under the true distribution $\mathbf{F}^c$.

Song and Nelson (2017) claim that modeling how *differently* two solutions' simulation outputs are affected by the common $\widehat{\mathbf{F}}$ helps reduce the conservatism. They propose the input–output uncertainty comparisons (IOU-C) procedure, which provides a set of solutions whose mean performances are close to $\eta_{i^c}(\mathbf{F}^c)$. A key component to design the IOU-C procedure is the *common-input-data (CID) effect* modeled by the delta method:

$$\eta_i(\widehat{\boldsymbol{\theta}}) - \eta_{i'}(\widehat{\boldsymbol{\theta}}) \approx \eta_i(\boldsymbol{\theta}^c) - \eta_{i'}(\boldsymbol{\theta}^c) + \left(\nabla\eta_i(\widehat{\boldsymbol{\theta}}) - \nabla\eta_{i'}(\widehat{\boldsymbol{\theta}})\right)^\top (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^c).$$

Note that $\nabla\eta_i(\widehat{\boldsymbol{\theta}}) - \nabla\eta_{i'}(\widehat{\boldsymbol{\theta}})$ measures how differently system i and i' are affected by uncertainty in $\widehat{\boldsymbol{\theta}}$. Therefore, when $\nabla\eta_i(\widehat{\boldsymbol{\theta}}) \approx \nabla\eta_{i'}(\widehat{\boldsymbol{\theta}})$, the impact of input model risk on the comparison of $\eta_i(\boldsymbol{\theta}^c)$ and $\eta_{i'}(\boldsymbol{\theta}^c)$ is reduced. This is analogous to how positive correlation of simulation errors induced by common random numbers sharpens

the comparison between two simulation outputs. *Sharpening the comparisons under input model risk has an equivalent impact on the optimization procedure as collecting additional real-world data.*

In most discussions of input model risk in the literature, and certainly in this chapter, the scope of "input model" is restricted to the distribution of a basic stochastic process that drives a computer simulation. However, model risk from imperfect input models is only a small part of the overall model risk. Even when the input distributions of the simulation model are completely known, if the simulation logic incorrectly represents the events in the real-world system then decisions based on the simulation model may be faulty. Such model risk can be diminished by careful validation of the simulation model, however, we cannot be free from it. Therefore, quantifying logical model risk is important to understand the usefulness of the simulation model, yet it is much more difficult to carry out than quantifying input model risk. *Can this problem be addressed by changing the way we build a simulation model?* For instance, we may view the "simulation logic" itself as a statistical model that can be estimated from data. With an increasing abundance of data this is becoming possible.

For instance, suppose we have video footage of an emergency room (ER) that records the movements of the patients, staff, and equipment in the ER. The traditional simulation modeling approach is to first build a logical model that describes the events in the ER and then model the stochatic processes that trigger event transitions from the data. However, recent advances in machine learning enable us to "estimate" the events themselves as well as the related stochastic processes. As a result, we may be able to quantify the risk due to errors in the simulation logic (which is really just a statistical model in this case) in a similar way as we quantify input model risk.

Acknowledgements Portions of this chapter were previously published in Song et al. (2014), and the chapter is based upon work supported by the National Science Foundation under Grant No. CMMI-0900354.

References

Akcay A, Biller B (2014) Quantifying input uncertainty in an assemble-to-order system simulation with correlated input variables of mixed types. In: Proceedings of the 2014 winter simulation conference. IEEE, pp 2124–2135

Akcay A, Martagan T(2016) Stochastic simulation under input uncertainty for contract-manufacturer selection in pharmaceutical industry. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 2292–2303

Ankenman BE, Nelson BL (2012) A quick assessment of input uncertainty. In: Proceedings of the 2012 winter simulation conference. IEEE, pp 1–10

Barton RR (2012) Tutorial: input uncertainty in output analysis. In: Proceedings of the 2012 winter simulation conference. IEEE, pp 1–12

Barton RR, Nelson BL, Xie W (2010) A framework for input uncertainty analysis. In: Proceedings of the 2010 winter simulation conference. IEEE, pp 1189–1198

Barton RR, Schruben LW (1993) Uniform and bootstrap resampling of empirical distributions. In: Proceedings of the 1993 winter simulation conference. IEEE, pp 503–508
Barton RR, Schruben LW (2001) Resampling methods for input modeling. In: Proceedings of the 2001 winter simulation conference. IEEE, pp 372–378
Batarseh OG, Wang Y (2008) Reliable simulation with input uncertainties using an interval-based approach. In: Proceedings of the 2008 winter simulation conference. IEEE, pp 344–352
Biller B, Gunes C (2010) Capturing parameter uncertainty in simulations with correlated inputs. In: Proceedings of the 2010 winter simulation conference. IEEE, pp 1167–1177
Cheng, R.C.: Selecting input models. In: Proceedings of the 1994 winter simulation conference. IEEE, pp 184–191
Cheng RC (2001) Analysis of simulation experiments by bootstrap resampling. In: Proceedings of the 2001 winter simulation conference. IEEE, pp 179–186
Chick SE (1997) Bayesian analysis for simulation input and output. In: Proceedings of the 1997 winter simulation conference. IEEE, pp 253–260
Chick SE (1999) Steps to implement Bayesian input distribution selection. In: Proceedings of the 1999 winter simulation conference. IEEE, pp 317–324
Corlu C, Biller B (2013) A subset selection procedure under input parameter uncertainty. In: Proceedings of the 2013 winter simulation conference. IEEE, pp 463–473
Corlu CG, Biller B (2015) Subset selection for simulations accounting for input uncertainty. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 437–446
Fan W, Hong LJ, Zhang X (2013) Robust selection of the best. In: Proceedings of the 2013 winter simulation conference. IEEE, pp 868–876
Freimer M, Schruben L (2002) Simulation input analysis: collecting data and estimating parameters for input distributions. In: Proceedings of the 2002 winter simulation conference. IEEE, pp 393–399
Gao S, Xiao H, Zhou E, Chen W (2016) Optimal computing budget allocation with input uncertainty. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 839–846
Ghosh S, Lam H (2015) Mirror descent stochastic approximation for computing worst-case stochastic input models. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 425–436
Henderson SG (2003) Input model uncertainty: why do we care and what should we do about it? In: Proceedings of the 2003 winter simulation conference. IEEE, pp 90–100
Hu Z, Hong LJ (2015) Robust simulation of stochastic systems with input uncertainties modeled by statistical divergences. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 643–654
Lam H (2016) Advanced tutorial: Input uncertainty and robust analysis in stochastic simulation. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 178–192
Lam H, Ghosh S (2013) Iterative methods for robust estimation under bivariate distributional uncertainty. In: Proceedings of the 2013 winter simulation conference. IEEE, pp 193–204
Lam H, Qian H (2016) The empirical likelihood approach to simulation input uncertainty. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 791–802
Love D, Bayraksan G (2013) Two-stage likelihood robust linear program with application to water allocation under uncertainty. In: Proceedings of the 2013 winter simulation conference. IEEE, pp 77–88
Morgan, L.E., Titman, A.C., Worthington, D.J., Nelson, B.L.: Input uncertainty quantification for simulation models with piecewise-constant non-stationary Poisson arrival processes. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 370–381
Sargent RG, Goldsman DM, Yaacoub T (2016) A tutorial on the operational validation of simulation models. In: Proceedings of the 2016 winter simulation conference. IEEE, pp 163–177
Song E, Nelson BL (2013) A quicker assessment of input uncertainty. In: Proceedings of the 2013 winter simulation conference. IEEE, pp 474–485
Song E, Nelson BL (2017) Input-output uncertainty comparisons for optimization via simulation. Technical report, Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL USA

Song E, Nelson BL, Hong LJ (2015) Input uncertainty and indifference-zone ranking & selection. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 414–424

Song E, Nelson BL, Pegden CD (2014) Advanced tutorial: input uncertainty quantification. In: Proceedings of the 2014 winter simulation conference. IEEE, pp 162–176

Vidyashankar AN, Xu J (2015) Stochastic optimization using Hellinger distance. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 3702–3713

Xie W, Nelson BL, Barton RR (2014) Statistical uncertainty analysis for stochastic simulation with dependent input models. In: Proceedings of the 2014 winter simulation conference. IEEE, pp 674–685

Xie W, Sun H, Li C (2015) Quantifying statistical uncertainty for dependent input models with factor structure. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 667–678

Yi Y, Xie W, Zhou E (2015) A sequential experiment design for input uncertainty quantification in stochastic simulation. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 447–458

Zhu H, Zhou E (2015) Estimation of conditional value-at-risk for input uncertainty with budget allocation. In: Proceedings of the 2015 winter simulation conference. IEEE, pp 655–666

Chapter 6
The Evolution of Simulation Languages

C. Dennis Pegden

Abstract Since the introduction of the GSP simulation framework by Tocher (Proceedings of the second international conference on operations research, pp 50–68, 1960), simulation tools have continued to broaden in both scope and power. This chapter will review the key concepts and developments in simulation languages over the last half century.

6.1 Introduction

A simulation language executes a model of a system to dynamically act out the behavior of the real system over time. This is done by changing the value of state variables over simulated time. Simulation languages are generally categorized into two broad families: discrete and continuous. Discrete tools model systems is where the state of the system changes by discrete amounts at specific event times. Continuous tools model systems is where the state of the system changes continuously over time. The focus of this chapter is on simulation languages for modeling discrete systems. However, we will also discuss the role of continuous modeling for approximating large-scale discrete systems.

Simulation languages are multifaceted and involve features related to model definition, execution, animation, experimentation/analysis, as well as features related to applications in emulation and scheduling. In this review, we will discuss the evolution of simulation tools across this spectrum. We will begin by discussing progress with model definition based on the changing use or modeling world views over time.

C.D. Pegden (✉)
Simio LLC, Sewickley, PA, USA
e-mail: cdpegden@simio.com

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_6

6.2 Discrete Modeling World Views

A discrete simulation is a set of state variables and a mechanism for changing those variables at event times. A simulation modeling worldview provides the practitioner with a framework for defining a system in sufficient detail that it can be executed to simulate the behavior of the system. Unlike simple static descriptive tools such as Visio, IDEF, UML, etc., a simulation modeling worldview must precisely define the dynamic state transitions that occur over time. The worldview must provide a definitive set of rules for advancing time and changing the discrete state of the model.

Over the 50-year history of simulation, there have been three primary world views in wide use: event, process, and objects. These were all developed by the pioneers of simulation during the 1960s (Tocher 1960; Gordon 1961; Markowitz et al. 1962; Kiviat 1963; Dahl and Nygaard 1967). Although these world views have been significantly refined over the past 50-year period, the basic ideas for these three primary world views have not changed.

The evolution of modeling tools has focused on achieving a balance between ease of use and flexibility. The event world view provides the greatest flexibility, but is more difficult to use. The process view is easier to user, but at the price of decreased modeling flexibility. The object view is the easiest and most natural of all, but at a price of even less flexibility. The history of simulation language development has been focused on making the process and object views more flexible, while retaining their advantage with ease of use.

6.2.1 Event Orientation

In this modeling paradigm, the system is viewed as a series of instantaneous events that change the state of the system over time. The modeler defines the events in the system and models the state changes that take place when those events occur. An important concept is that simulated time cannot advance within an event. Modeling an activity that transpires over time requires separate events to define the start and end of the activity. This approach to modeling is very flexible and efficient, but is also a relatively abstract representation of the system. Thus, many people find modeling using an event orientation to be difficult.

We will illustrate the event worldview using a model of a service facility with a fixed capacity that defines the number of service operations that can be processed in parallel. When using the event worldview, we begin by asking the following questions:

1. How is the state of the system defined?
2. What events occur that can change the state of the system?
3. What is the logic within each event that defines the state transitions?

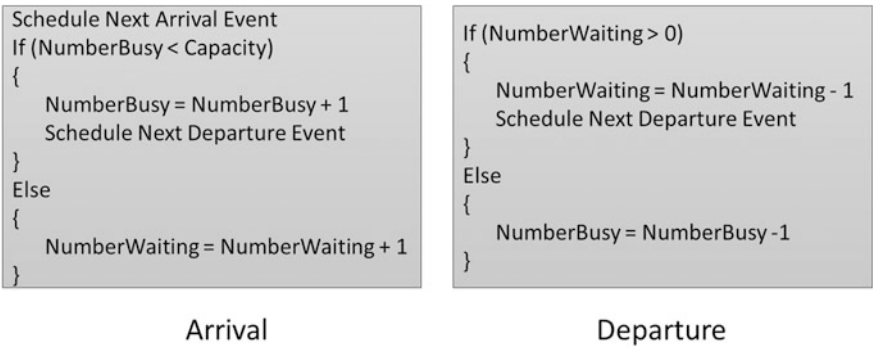


Fig. 6.1 Arrival and departure

In our system example, the state of the system can be defined by the number of busy servers, and the number of customers waiting to start service. This simple state description assumes that we do not distinguish between individual servers or customers (if we did, a more elaborate state description and transition logic would be required). The events that can change this state are a new customer arrival and a customer departure at the end of servicing. The state transition logic for each of these events is summarized above (Fig. 6.1).

Note that each arrival schedules the next arrival in the sequence. We then check to see if we have available capacity for this arrival and if so, we bump the number busy and schedule the departure event. Otherwise we bump the number waiting. In the departure event, we check if we have customers waiting and if so we decrement the number waiting and keep the number busy unchanged and schedule the departure event for the next customer. Otherwise we reduce the number of busy servers.

A modeler would typically define the logic for the state transitions for each event using a simulation programming language such as Simscript (Markowitz et al. 1962), or a general-purpose language such as FORTRAN, Java or C++ augmented with a library such as GASP (Kiviat 1963) to assist with common simulation functions such as sampling from distributions or scheduling events. Simulated time advances from event to event until the end of the simulation is reached (based on simulated time, number customers processed, etc.). Note that event-based tools provide none of the model logic, but simply a computationally efficient framework for defining that logic using a programming language.

The event world view was widely used during the first 20 years of simulation. These tools were favored by many because they were very flexible and could be used to efficiently model a wide range of complex systems. Note that during this period, efficiency was more important because computers had less memory and were significantly slower. However, event-based models were often difficult to understand and debug and required programming skills that limited their general use. Once the process modeling tools became as flexible and as computationally

efficient as the event tools, users quickly migrated to the process worldview because it was easier to learn and use. This shift was further accelerated when process tools (beginning with SIMAN) were made available on the new IBM personal computer. Although the event worldview is no longer used as the primary modeling approach in real applications, it remains important to understand. Many process and object-based simulation tools are internally implemented using this basic modeling approach, regardless of the worldview that they present to the user.

6.2.2 Process Orientation

In the 80s, the process orientation began displacing the event orientation as the dominant approach to discrete simulation modeling, and it held this dominant position for the next 20 years. In the process view, we describe the movement of passive entities through the system as a process described by a flowchart. The process flow is a series of process steps (e.g., seize, delay, release) that model the state changes that take place in the system. Note that each process step is an action that is typically named using a verb. Unlike a simple event that occurs at an instant in time, a process step may span time and execute as a sequence of events.

The internal logic for the early process-oriented tools was implemented based on activity scanning concepts introduced in the simulation tool GSP (Tocher 1960). The modeling view of GSP focused on activities and the conditions that start and end an activity. Although this modeling orientation was not widely used, the activity view shared much in common with the process orientation by focusing on activities that transpire over time. This tool incorporated a three-phase time advance mechanism involving bound and unbound activities, which was then adopted by the early process-oriented languages.

The process orientation dates to the early 1960s with the introduction by IBM of GPSS (Gordon 1961). This process-oriented simulation tool provided a more natural way to describe the system as compared to the event approach. The growth of GPSS was also fueled by the classic book *Simulation Using GPSS* (Schriber 1974). Despite its advantage in terms of ease of use, many practical issues with the original GPSS limited its use compared with the event approach. It did not become the dominant approach to modeling until improved versions of GPSS (e.g., GPSS/H Henriksen and Crane 1983) along with new process languages such as SLAM (Pegden and Pritsker 1979) and SIMAN (Pegden 1982) that addressed these practical limitations became widely used in the 80s.

The two key issues that the early process languages suffered were an integer clock and activity scanning logic, which resulted in slow execution times. The initial process languages would advance time by 1 time unit, do a scan of which process steps could start or stop, and then repeat the cycle. This activity scanning approach was slow for large models, and the integer clock also created many tied event times, where the order of execution would impact model logic. Some of the

early process languages were also interpretative languages, which further impacted their execution speed.

When modeling a system using the process worldview, we begin by asking two questions:

1. What are the entities (customers, work items, etc.) that move through the system?
2. What process steps are executed as each entity moves from step to step in the process?

One of the important advantages of the process orientation is that the process model is defined graphically in the form of a flowchart. Hence, the user does not have to be a programmer to be able to create a model of the system. The process flowchart for our simple server example is shown below (Fig. 6.2):

The service operation is modeled by a resource that has a capacity that can be seized and released by entities as they flow through the process. Entities representing customers arrive to the system, wait to seize the Server, delay by the processing time, release the Server, and then record the time they spend in the system. Learning to model with a process language involves learning the various process steps that are available, and then how to combine these steps together in the form of a flowchart to represent complex systems. Modern process languages have a wide range of process steps (>50) for manipulating fixed and moveable resources, transporters, conveyors, etc.

There are two basic types of time advances that may occur in a processing step. The first is a planned delay (e.g., wait for 2 min) that is represented by the delay step. The second is a conditional delay (e.g., wait until a resource becomes available, or wait until a tank is full of liquid). A conditional delay is illustrated by the seize step that waits (if necessary) until the server becomes available.

Comparing the process model to the event approach, we see that the model logic is much simpler to define and understand, and requires no programming skills. Hence once the initial implementation problems were addressed, the process approach quickly replaced the event approach as the dominant modeling world view.

Modern process languages use a next event time advance mechanism and real-time clock that make them computationally competitive with a pure event oriented modeling approach. The original activity scanning logic is replaced with internal event logic that efficiently implements conditional state changes. If activity scanning is provided, it is isolated in a specific process step (e.g., the SCAN step in SIMAN, which waits for an arbitrary condition to become true) that isolates and limits the computational burden of this approach to modeling.

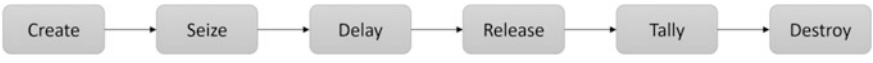


Fig. 6.2 Process flowchart

Another conceptual advance that occurred with process modeling was the introduction of hierarchical modeling tools which supported the idea of domain specific process libraries. The basic concept is to allow a user to define their own process step by combining existing process steps. The template feature in the widely used Arena modeling system is a good example of this capability.

6.2.3 Object Orientation

Although the process orientation greatly simplified the model building experience, an object orientation provides an alternative modeling paradigm that is more natural and in many cases easier to use. In an object-based approach to modeling, we create our model by placing objects into our model that represent the physical components of the system—e.g., a server, worker, forklift truck, conveyor, etc. These objects may be customized by specifying property values such as service time, travel speed, etc. For example, we model a factory by placing and describing through properties the workers, machines, conveyors, robots, and other objects that make up our factory. These physical components interact to produce a simulation of our system.

Note that where process steps are described by verbs (seize, delay, release), objects are described by nouns (machine, robot, worker). With the process orientation, the modeler describes the actions that take place in the system as entities move through processes. In the object orientation, the modeler simply describes the physical components in the system, and the behaviors and actions for these objects are already built into the objects. Hence, a worker object has predefined behaviors that allow it to interact with machines and other workers in the model.

The object model for our simple service system is shown below, where the system is modeled using a source, server, and sink object. Entities represented by dark gray triangles enter the model at the source, wait in the buffer to be processed by the server, and then depart the system at the sink (Fig. 6.3).

Building this model is both natural and straightforward—the user simply places these objects into their model and specifies any associated properties such as the arrival rate, buffer size, and server processing time. The underlying event logic is prebuilt into these objects.

It is difficult to envision a more natural way to build a model than by using a collection of prebuilt modeling components that mimic the components in the real

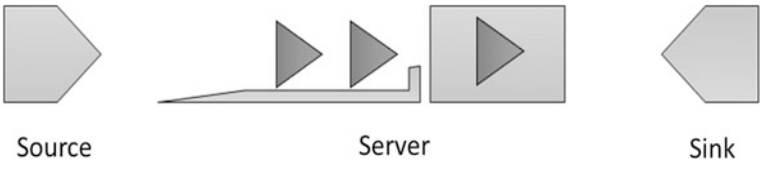


Fig. 6.3 A simple service system

system. The challenge with this approach is that if we want to model anything in the real world, we need an extensive library of objects to be able to capture the massive diversity of real objects that you may encounter. For example, it is not adequate to have a single object called robot as there are many different types of robots in the real world. The pursuit by simulation language developers of creating a practical simulation tool based on the object approach illustrates the challenge of having both flexibility and ease of use in the same simulation modeling tool.

Although the flexibility provided by the process orientation makes it still a widely used approach to simulation modeling, a growing number of successful simulation products have been introduced in the last 20 years based on the object orientation. Just as the second 20 years produced a shift from the event orientation to the process orientation, the past 20 years has seen a shift from the process orientation to the object orientation. The newer object-oriented tools have a rich set of object libraries that are focused in specific application areas such as manufacturing, supply chain, and health care. Some of these tools also allow users to create and customize their own object libraries for specific application areas.

The basic idea of being able to create custom objects as a formal concept was introduced by Ole-Johan Dahl and Kristen Nygaard of the Norwegian Computing Center in Oslo in the 1960s in Simula and Simula 67. Simula introduced the notion of classes of behavior (e.g., server, worker, robot), and instances thereof (objects, e.g., Fred and Drill), as part of an explicit modeling paradigm. A modeler can create an object class such as car, and then place multiple instances of that class into their model, and customize the behavior of each instance by setting property values. They also introduced the notion of subclassing objects. This powerful concept allows a user to create a new object class from an existing object class by inheriting, overriding, and extending the object class behavior. For example, a new object class named truck might be created from the object class named car by redefining some behaviors, and adding some new behaviors. The ability to create a new object class by starting with an existing object class that has some desired behaviors greatly simplifies the development of object libraries.

The ideas introduced by Simula provide the foundation for the recent advances made by simulation language designers to make the object-oriented approach to modeling both easy to use and flexible. Although these ideas were introduced as simulation modeling concepts, they have completely changed the design and implementation of programming tools in general. The ideas from Simula directly influenced many later programming languages, including Smalltalk, LISP, C++, Java, and C#. The object-oriented ideas introduced by Simula are not only the most significant development in simulation software, but perhaps the greatest advance in computer science in the last 50 years.

The key concepts introduced by Simula and now part of modern programming languages include the following:

- **Classes**: The abstract characteristic of a thing (object), including its characteristics (properties and states) and behaviors (things it can do).

- **Instances**: The actual object created from the class. The object holds its state and the state transition behavior is defined by the class.
- **Interface**: The objects interact in terms of passing values and messaging.
- **Subclassing**: Specialized versions of a class, which inherit attributes and behaviors from their parent class, which can then be modified and extended.

There are some differences of opinion as to what it means to be an object-oriented simulation tool. Some tools (e.g., Witness, Simul8) provide an object library as the basis of modeling, but do not support the object-oriented concepts for creating new object classes based on inheritance and subclassing. Other tools (e.g., AnyLogic, FlexSim, and Simio) fully support the object-oriented framework for creating new object classes.

Much of the innovative work in simulation language design is occurring in object-oriented simulation tools. These tools are getting more flexible while retaining their advantage in terms of ease of use, and therefore displacing the process-oriented tools. They also have a key advantage in terms of animation. In the case of the event and process orientations, the addition of animation is a two-step process, where the user first builds the logical model, and then creates the animation as a separate step, and then ties these two components together. In the object orientation, the predefined objects not only come with their associated properties, states, and behaviors, but also their associated 3D animations. This allows the user to rapidly build both the model logic and animation in a single step.

6.3 System Dynamics

System dynamics is a modeling approach developed at MIT during the late 50s by Jay Forrester. Although it is a continuous modeling approach, it is often used to approximate large-scale discrete systems. The basic idea is to model complex systems as a collection of stocks, flows, and feedback loops. A stock is represented as a tank that fills and empties and measures the level of a state variable such as the number customers that have purchased a product, or the number of people that have been exposed to a disease. A flow is represented as a valve that controls the rate of change of a stock (Fig. 6.4).

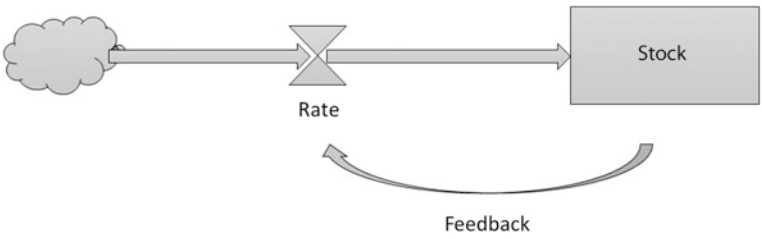


Fig. 6.4 A simple system dynamics model

Forrester published the first, and still classic, book in the field titled *Industrial Dynamics* (1961). One of the best-known system dynamics models is a model of the growth of world population, which was popularized in the bestselling 1972 book *Limits to Growth*. This model forecast the exponential growth of population and capital, with finite resources, leading to economic collapse under a wide variety of scenarios. The original model had five levels that measured world population, industrialization, pollution, food production, and resource depletion.

Although any continuous simulation tool can be used to simulate system dynamics models, a number of specific modeling tools have been developed, including Dynamo, PowerSim, and AnyLogic. Although system dynamics models are expressed as continuous systems, most of the applications involve modeling large-scale discrete systems with many entities. An alternative to System Dynamics for large-scale discrete systems is agent modeling, which we will discuss next.

6.4 Agent Modeling

The term agent-based modeling (ABM) has been widely used in the recent years. The basic concept of agent-based modeling is that a system is modeled by placing agents in the system and letting the system evolve from the interaction of those agents. Each agent is an autonomous entity which interacts with other entities in the system. The focus is on modeling agent behavior as opposed to system behavior. In a traditional process orientation, entities follow a sequence of process steps, which are defined from the top-down system perspective. In contrast, agent-based modeling defines the local behavior rules (often simple) of each entity from a bottom-up perspective. The system behavior is produced as the cumulative result of the agents interacting with each other. Example applications include crowds moving through an area, customers responding to new product introductions, or troops in combat (Fig. 6.5).

Although the state transition framework for the agents can be modeled using any world view, agent-based modeling is often implemented using an object-oriented simulation tool. Hence this is not a new discrete event world view, but rather a

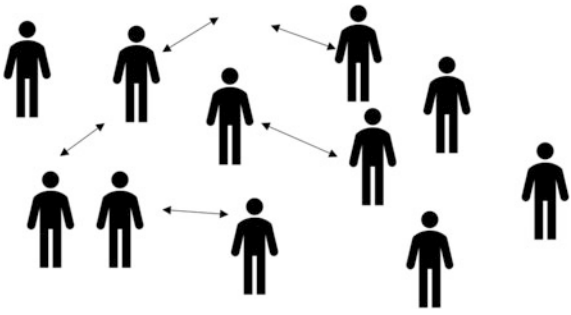


Fig. 6.5 Agents and interactions

group of applications that are often modeled with the object world view. Classes are used to define agent states and behaviors and instances (often large numbers) are placed in the model. The agents (objects) interact and the system state evolves over time. Some of the application areas in agent-based modeling present some unique challenges, particularly when large numbers of agents are involved (e.g., modeling the evacuation of fans from a stadium).

For simple applications of agent-based modeling, the full object-oriented framework with inheritance and subclassing is sometimes more powerful than is needed and a simple state diagram may be adequate for defining the class behavior for each agent. The concept of a state diagram was introduced by Shannon and Weaver in their (1949) book *The Mathematical Theory of Communication*. A state diagram defines a finite set of states that the agent can be in, along with the transition conditions that cause a transition between a pair of states. Each diagram typically defines behavior for an object of a given class, and the state transitions for the object instances of that class. Although there are several different variations for state diagrams in use, they all define states as nodes, and arcs for transitioning between states. For example, the following state diagram defines transitions between the states well, sick, and dead (Fig. 6.6).

Because of the increased capacity of computers to manage large numbers of agents, some models that were previously done as discrete approximations using a system dynamics approach are now better done using an agent-based modeling approach. This allows greater flexibility in the logic at the expense of a longer execution time.

One of the early agent-based models was the *Game of Life* developed by John Conway in the 1970s. The *Game of Life* is a two-dimensional model involving cellular growth that evolves using a fixed time step, where each cell has two states, alive, and dead. The state of a cell depends on the state of the neighbors of the previous time step. Conways's game demonstrated the basic concept of the emergence of complexity from simple rules.

Interest in agent-based modeling continued to grow and diversify in the 1990s with the appearance of various agent-based tools, particularly Swarm, NetLogo, Recursive Porous Agent Simulation Toolkit (Repast), and AnyLogic.

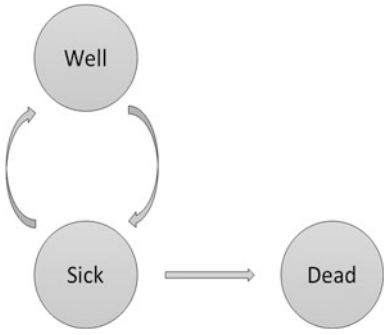


Fig. 6.6 State change example

6.5 Multi-paradigm Modeling

Most simulation modeling tools support multiple modeling paradigms. By providing alternative paradigms in the same tool, the user can select the modeling approach that best fits the problem at hand. GASP IV (Pritsker 1974) introduced the idea of combining discrete and continuous frameworks in the same model. Simulation Language for Alternative Modeling (SLAM) was one of the first widely used tools to promote the idea of mixing alternative modeling approaches (processes and events and continuous) in the same model. In the case of SLAM, the process/continuous modeling was used for basic modeling, and the events were used as the "backdoor" to provide added flexibility.

The modern object-oriented simulation tools also employ a multi-paradigm approach. These tools combine the ease and rapid modeling provided by objects with the flexibility added by incorporating user specified events and/or processes. For example, an object representing a server might have selected points in the object logic where the user can insert their own event logic or process logic. Event logic is typically incorporated into objects using either a programming language such as Java or C++, or a special scripting language that can be used in place of code. However, in either case event logic has one major restriction: simulated time cannot advance within the event. This severely restricts the type of logic that can be inserted into an object instance. For example, it is not possible to wait for a worker to become available within an event since this would require time to advance. Simio introduced a much more powerful approach which allows users to combine objects with processes. Since processes can span time they provide the user with considerable more power to extend the behavior of their objects. Hence, processes can be embedded within an object to wait for a specified time or specified condition (e.g., Fred is available). This approach combines the full power and flexibility of processes with the ease and rapid modeling capability of objects.

AnyLogic has combined agent-based modeling, system dynamics, and discrete event modeling into a single tool. A single AnyLogic model can combine all three modeling approaches to represent the system behavior.

6.6 Animation

For the first 30 years of simulation, the output was in the form of static reports. Beginning in the 80s, animation was introduced as a better way to understand the underlying behavior of the model. Animation has had a huge impact on model verification/validation and on communicating the results of a simulation model to the various stakeholders.

One of the early simulation animation systems was product called SEE WHY developed in the late 80s, which used rudimentary character-based animation. The Witness simulation software was developed from SEE WHY. The Cinema

animation system was a 2D vector-based animation system developed in the same time frame and was a real-time animation system for SIMAN models. The Cinema animation system and SIMAN modeling system were then integrated together into a single platform to create the widely used Arena product. Another early 2D animation system was Proof, developed by Jim Henriksen of Wolverine Software (1983).

In the 90s, 3D animation capabilities began to emerge. Some of the early systems included AutoMod (focused on material handling systems), Taylor ED (a precursor to FlexSim), and Simple++ (a precursor to PlantSim). Although 3D provides a clear animation improvement over the first generation 2D capabilities, these initial systems had more limited use due to the added complexity of working with a 3D model.

The combination of the object-oriented framework, an immersive 3D modeling environment, and improved 3D drawing features has proven to a very powerful combination for brining 3D animation into the mainstream of simulation modeling. Most modern simulation tools now provide 3D as a standard capability.

The next wave of animation for simulation is development of full virtual reality (VR) environments for displaying simulation animations. This allows a viewer to be immersed into the model, and walk through the system in a VR environment that mimics the real system. Several simulation systems (e.g., Simio, FlexSim, Emulate3D) already support VR environments for animation.

6.7 Model Verification/Validation

An important part of any simulation project is verification and validation of the simulation model logic. This process has been greatly improved over the years as animation features have continued to evolve. It is now much easier to have all the stakeholders involved with a simulation project, and play a role in verifying and validating the model operation.

When unexpected behavior is observed in the model, the next challenge is to find the problem in the model logic. The tools available for isolating logical problems in the model have continued to improve with each new generation of simulation products.

One important feature in the latest generation simulation products is filtered logic trace. Since the beginning of simulation tools, trace of model logic has always been a primary tool for tracking down issues with model logic. However, the problem with trace is that it is so verbose that it is sometimes difficult to find the appropriate information in the mass of trace output—particularly with the larger models that are often built today. Selective filtering of the trace output to a specific resource, entity, process, etc., makes it much easier to drill into the trace data and find the relevant information.

Another key feature that has been added to the latest generation simulation products is the ability to graphically add breakpoints and watches on state variables

for objects in the model. This makes it easy to interrupt the simulation at a specific point, and then query the values of the key state variables in the model.

The combination of enhanced animation, filtered trace, and graphical editing of breakpoints and watches has greatly improved the ability to track down and fix logical issues in the model.

6.8 Experimentation and Analysis

Although much attention has been given to the improvement of modeling features, much of the value of simulation comes from the effective use of the model for experimentation and analysis. This is a critical part of any simulation project that is often neglected.

During the first 40 years of simulation, experimentation received very little attention in simulation software design. As a result, experimentation was largely a manual effort that took a lot of time and effort. Although the academic world produced improved procedures for experimentation, these procedures were not incorporated into simulation software. Use of these procedures required statistical skills as well as significant time and effort. Since experimentation and analysis is done at the end of the project, time pressure often forced cuts in this area of the project. Hence, there was a gap between what could be done in theory, and what was done in practice.

During the last 20 years, there has been a greater focus from simulation software developers to incorporate analysis features into their products. As a result, it is now much easier to define and run experiments and do a proper analysis of the results. This makes it easier to make better decisions from the simulation modeling project. Modern simulation tools now contain automated procedures for selecting the best of a set of candidate system, or selecting a subset containing the best. They also contain specialized modules (e.g., OptQuest) for doing an optimal search of the decision space for the model.

Much of simulation analysis during the past 60 years has focused on using the model for estimating the mean value of key performance indicators (e.g., throughput, utilization, etc.). Confidence intervals on the mean are used to indicate the potential magnitude of the sampling error in the estimates based on the number of replications of the model. This sampling error can be made arbitrarily small by increasing the number of replications of the model. Although the confidence intervals provide a measure of sampling error, they provide no indication of the risk associated with the system due to the underlying variability in the system. Note that unlike sampling error, the underlying risk (variability) in the system is not reduced by more replications of the model. The risk is an underlying characteristic of the system that we would like to measure.

Although the expected value of a key performance indicator is important, in many applications knowing the variability and risk associated with that indicator is equally important. For example, knowing that the average daily throughput is 862

might not be as relevant as knowing that with 90% confidence the actual throughput on any given day is at least 820. Note that a confidence interval does not answer this important question. In the past simulation, tools provided no automated way to capture this risk information from the experimentation. However, with the recent introduction of MORE plots (Nelson 2008), this information is now readily available in some of the new simulation tools.

6.9 Simulation Execution

Simulations can be computationally intensive and place large demands on both computer memory and processor speed. As computer power has expanded, the constraints imposed by computer limitations have dramatically reduced. It is now possible to run large 3D animated simulation models on an inexpensive notebook computer. It is also now possible to run large-scale agent models that in the past could have only been done using a system dynamics approach.

Improvements in simulation execution are not only due to better hardware, but also advances in simulation software design. The latest event calendar designs are computationally scalable to support many scheduled events without a noticeable slowdown in execution speed. Most modern simulation software products now support 64-bit addressing, which means very large models can be executed. Some simulation products also make use of multiple cores and networked computers to simultaneously run multiple replications of a model. The combination of better computers and better software design allows larger models and larger associated experiments to be executed in significantly less time than before.

One significant recent development in simulation execution supports for running replications in the public cloud to allow for quick execution of massive simulation experiments involving 1000s of replications. This makes it possible to quickly scale up the necessary processor power for only the time required to complete the experiment. The cloud also makes it convenient for sharing models and model results across all the stake holders.

6.10 Simulation Applications

In the past, simulation was largely applied to facility design to reduce the risk of capital investments in new or existing facilities. While system design remains an important application area, simulation is rapidly expanding to additional application areas, including system commissioning and system operation.

As a design tool simulation is used to model and predict the behavior of either a new facility, or proposed changes to an existing facility, before making the capital investment in the real system. For example, by building a model of the factory we can simulate production to test out ideas and see the impact of critical design

decisions such as the adding new equipment or people, or changing the way we manufacture a specific product.

As a system commissioning tool simulation is used as a surrogate for the real system to test manufacturing execution system (MES) software and to provide operator training before the real system is in place. In this application, we run the model in real-time (or scaled real-time) emulation mode and use it as a direct substitute for the real system. This makes it possible to have both the MES software tested and the people trained and ready for startup once the equipment is in place. This can dramatically reduce the time and cost required to commission a new cell, line, or factory. Without the aid of simulation, new plant commissioning remains a costly and time-consuming process, largely due to the long lead times to bring the system to full operation. These long lead times occur at the point of maximum investment and zero return. Emulate3D is an example of a simulation tool designed specifically for this application area.

As a system operation tool simulation is used on an operational basis to schedule the actual operations within the system. For example, in a manufacturing application, we run the model to simulate the production of a specific set of jobs that we plan to produce in our facility during our next production cycle, and in so doing generate a production schedule for the plant. Unlike ERP/APS generated schedules, the simulation-generated schedule provides us with a detailed production plan that fully accounts for the complexities and constraints of the factory floor. In fact, we can watch a facility animation of our schedule being generated that shows us in detail how the work on the factory floor will flow through the facility over time. In addition, we can replicate the schedule generation process in stochastic model to generate risk measures for the resulting schedule. Simio is an example of a simulation tool designed specifically for this application area.

6.11 Summary

Since the early 1960s, simulation languages have continuously evolved and improved. Although the foundational modeling ideas were developed in the first decade, it took a half century to refine those ideas and create modern simulation tools that are both flexible and easy to use.

The latest developments in simulation language design have focused on leveraging the ease of use of the object-oriented approach while adding flexibility through improved library design and incorporating events and processes. Multi-paradigm modeling approaches blend the advantages of different modeling world views. These object-based languages also leverage the advances in 3D animation.

The advances in simulation software have also been paralleled by improved experimentation capabilities as well as significant advances in computational and animation hardware. Large animated 3D simulation models can now be run on inexpensive laptops and tablets, and large-scale experiments can be executed using

cloud computing. Finally, the role of simulation for improving business performance is expanding from system design to include system commissioning and system operation.

References

Dahl OJ, Nygaard K (1967) Simula 67. IFIP TC 2 Working conference on simulation languages, Oslo

Forrester J (1961) Industrial dynamics. The M.I.T. Press

Gordon G (1961) A general-purpose systems simulation program. In: Proceedings of the eastern joint computer conference, Washington, DC

Henriksen J, Crane R (1983) GPSS/H user's manual. Wolverine Software

Kiviat PJ (1963) GASP—A General Purpose Simulation Program. Applied Research Library, United States Steel Corporation

Markowitz H, Hausner B, Karr H (1962) SIMSCRIPT: a simulation programming language. Prentice Hall, Englewood Cliffs, NJ

Nelson B (2008) The MORE plot: displaying measures of risk and error from simulation output. In: Proceedings of the 2008 winter simulation conference

Pegden CD, Pritsker A (1979) SLAM: simulation language for alternative modeling. Simulation, pp 145–157

Pegden CD (1982) Introduction to SIMAN. Systems modeling

Pritsker A (1974) The GASP IV simulation language. Wiley Interscience

Schriber T (1974) Simulation using GPSS. Wiley

Shannon C, Weaver W (1949) The mathematical theory of communications. University of Illinois Press

Tocher KD, Owen DG (1960) The automatic programming of simulations. In: Proceedings of the second international conference on operations research, pp 50–68

Chapter 7
A Brief History of Time Warp

David Jefferson and Richard Fujimoto

Abstract This chapter is about the history of the Time Warp algorithm and optimistic approaches to parallel discrete event simulation. It concentrates on the early history from our personal perspective as active developers of the ideas over several decades.

7.1 Introduction

Time Warp is the name of an algorithm for doing discrete event simulation in parallel. It is an *optimistic* simulation mechanism, one that takes risks by performing *speculative* computation which, if subsequently determined to be correct, saves time, but if incorrect, must be rolled back. Time Warp can be used in any computation that uses a global temporal coordinate system for synchronization, but *discrete event simulation* using *simulation time*, is by far the most important example.

Because of its complete embrace of distributed rollback as the fundamental synchronization primitive instead of more conventional primitives such as locks, semaphores, or other process blocking constructs, Time Warp was considered a radical innovation when it first appeared. It is even today, after 35 years, virtually unique in that respect, but it has proved to be an elegant and powerful parallel algorithm, able to achieve excellent parallel performance at a scale of almost 2 million cores with 8 million threads and 250 million LPs so far (Barnes et al. 2013).

This chapter is a brief history of the development of Time Warp, largely concentrating on the 1980s and 1990s. It is necessarily incomplete, and is entirely from our personal perspectives. But this volume on the 50th Anniversary of WSC seems

D. Jefferson (✉)
Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, USA
e-mail: drjefferson@gmail.com

R. Fujimoto
Georgia Institute of Technology, Atlanta, USA

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_7

to be the perfect place in which to recall this early history. The first part of the chapter is by David Jefferson, and the second part by Richard Fujimoto. We hope that other authors will contribute their recollections as well.

7.2 The Early History of Time Warp: David Jefferson's Perspective

This first section provides the perspective of David Jefferson, describing the early years of Time Warp at RAND, JPL, and within the Jade projects.

7.3 Origin of Time Warp at RAND

The development of Time Warp began with a project at the RAND Corp. in Santa Monica in 1981. The Air Force was funding research to improve military simulations in two ways: first, to allow models to be specified in a quasi-natural language so that nonprogrammer generals might build their own models or scenarios, and second, to speed up simulations through parallelism. I was a young assistant professor at the University of Southern California with a background in parallel computation, which at that time was relatively rare, so they recruited me to work as a consultant on the project. I thought the first objective of natural language model building was unlikely to succeed, but I might be able to contribute toward the second of parallelizing discrete event simulations.

I had never read any literature on parallel discrete event simulation (PDES), of which there was not much, and I did not immediately do a literature search. I just started working on the problem from first principles. I was familiar with the sequential discrete event simulation algorithm based on a priority queue, and I thought that creating a parallel version of the algorithm would probably be straightforward. That, of course, turned out to be exceptionally optimistic because now, 35 years later, there are still basic issues, like load balancing or the relationship of PDES to the continuous simulation techniques used for numerical solution of ordinary and partial differential equations, that we are still trying to understand about PDES.

On my first consulting day at RAND, I recognized the core problem as synchronizing the interaction among many concurrently executing processes (called "logical processes", or LPs) sending timestamped event messages to one another. Every LP receives a stream of timestamped event messages sent by other LPs. The incoming messages at an LP do not generally arrive in increasing timestamp order, and in the general case there is no limit on how far out of order they are. Nonetheless, each LP must process event messages strictly in increasing timestamp order. If and only if all LPs do that (and have the same tie-breaking rule as well), the

resulting simulation is equivalent to that of the sequential algorithm, as required. This was similar (with some minor differences) to the way Chandy and Misra (1979, 1981), and independently Bryant (1977), had already framed the problem, though I did not know it yet. The most important conceptual difference was that they assumed event messages were transmitted within a static graph of order-preserving message channels between LPs, whereas I assumed any LP could send an event message to any other at any time, and without requiring order preservation.

I quickly realized some key facts about the synchronization problem in parallel discrete event simulations. Except in special cases, one cannot achieve any significant parallelism by trying to keep all the LPs of a simulation tightly synchronized in simulation time, or requiring that events in different LPs be executed in increasing simulation time order. Any such attempt to constrain the simulation would over-synchronize it and effectively sequentialize execution, no matter how many processors were used. Instead, a parallel simulator must allow some LPs to run ahead in simulation time while others lag behind, with no a priori bound on the time difference. Also, which LPs are ahead or behind must be able to change dynamically as the simulation progresses. There cannot be a single, globally shared standard for simulation time as there is in the sequential algorithm and in parallel time-stepped simulation algorithms. Instead, each LP would need its own simulation clock, and in any hypothetical instantaneous snapshot two such clocks would rarely, if ever, agree. These observations applied even in the context of shared memory parallelism, though we were interested in distributed algorithms.

I do not recall the exact moment of realization, but at some point it occurred to me to think in terms of *asynchronous parallel rollback* as a possible synchronization primitive instead of the classical primitives based on process blocking and resumption. In the late 1960s, rollback had been used in limited ways in sequential (but not parallel) debuggers (Balzer 1969). And rollback limited to the scope of a single transaction (transaction abortion) had recently been studied in the context of optimistic parallel database synchronization (Kung and Robinson 1981). Other work I was at least dimly aware of that may have had some influence on me included studies of parallel backtracking algorithms in such applications as parallel tree search, alpha–beta pruning in game trees, and branch-and bound optimization.

But while rollback was conceptually straightforward in the context of sequential computation or transaction abortion, how could it possibly work at all in the general case of an asynchronous parallel computation communicating by timestamped messages, let alone work efficiently? I did not know, but I felt forced in that direction because I had been able to construct simple artificial models with just two or three LPs where it seemed impossible to achieve enough parallelism without rollback. But in implementing rollback, how do you undo the fact that the process being rolled back may have sent out event messages that should be recalled, and the receivers of those messages may by now have sent secondary event messages that also should also be recalled? There would be a potentially large, dynamically- and asynchronously expanding tree of event messages that must all be recalled because of the original rollback, and that tree is growing, potentially exponentially, even

while the rollback is in progress! And in fact many parallel and uncoordinated rollbacks could be going on simultaneously in various parts of the simulation that would potentially interact and interfere with one another in unpredictable, nondeterministic ways. How was it possible to accomplish a clean rollback in the context of such a chaotic mess?

These considerations were all something of a disturbing surprise, and it concerned me enough that I checked with the project leader at RAND, Phil Klahr, to be sure that he was OK with me exploring what seemed like radical approaches to the problem of parallelizing discrete event simulation. To his great credit Klahr said that he did not care how I proposed to parallelize a simulation as long as the algorithm achieved speedup and got the correct results, i.e., the same results as would be produced by sequential execution. That freed me to think as far outside the box as I wanted.

7.4 Collaboration with Henry Sowizral

At this point, Klahr also made another crucial decision. He teamed me with a RAND researcher, Henry Sowizral. This turned out to be an extraordinarily fruitful partnership. After filling Henry in on my thinking, he immediately saw where it was going and we worked closely together thereafter. Over the next 9 weeks during my Friday consulting visits to RAND the two of us jointly developed most of the core ideas of what came to be known as Time Warp. Our research "method" was to walk outside along the Santa Monica Pier, the beach, and the palisades above, enjoying the sun and the sights, and brainstorming continuously for hours about PDES.

That collaboration, though lasting less than 2 years, was one of the most productive in my life. Henry and I were complementary and coequal. We adopted speculative execution with general distributed rollback as our computational paradigm, and then systematically rethought virtually every other issue in distributed computation with rollback in mind. This was, and still is, a radical departure from other paradigms of parallel computation. It allows the computation to execute speculatively *down completely incorrect computational paths* that were never intended or envisioned by the programmer. Eventually the incorrectness is detected by a causality violation signaled by the arrival of an event message with a timestamp in the simulation past of the receiving LP. That causes a cascade of actions in which all LPs in the distributed simulation that have been directly or indirectly affected by the original erroneous speculative event may be rolled back to times in their past when their dependencies on the original error began. Execution then proceeds forward in all of them again down a (more nearly) correct path, in a manner that might be described as many parallel backtracking "searches" for the correct forward paths.

The fact that this approach could be made to work at all was astonishing to us. It seemed totally counterintuitive that you could profit by (a) doing some distributed computation speculatively, which might be utterly wrong, while also paying substantial additional overhead to allow for possible rollback, and then (b) also

sometimes paying the cost of distributed rollback to undo the speculative computation when required, before (c) finally redoing the correct computation. But we became confident, without real proof yet, that the resulting parallel computation could be faster than an algorithm in which you carefully refrained from doing anything speculative at all, at least sometimes.

But we were still faced with a lot of difficulties. How could a rollback mechanism be implemented so that the speculative execution converged stably and deterministically on the correct execution? How could we guarantee that the simulation made forward progress instead of thrashing forever in rollback activity? If it was theoretically possible at all, how could it not be overwhelmed by state-saving or synchronization overhead? How could we reclaim memory and avoid filling it with data needed to support rollback? And how could this possibly scale well in a distributed-memory platform?

Henry and I kept struggling with these problems. We conceived of a mechanism whereby incorrect event messages could be "unsent" by sending special "cancellation" messages that tell the receiver to throw away the cancelled event message and to undo whatever further computation they may have done based on it. That was promising, because it would induce a tree of such cancellation messages and rollbacks in other LPs as necessary. But would this growing tree of cancellations ever converge? What if, after an LP sent a cancellation message, it had to roll back again to a time before it sent the cancellation? It seemed that we would then have to send out a second-order cancellation message to cancel the previous cancellation message, and third-order cancellations might be required to cancel erroneous second-order cancellations. Would not that mean we would need an infinite hierarchy of higher order cancellation types?

At this point an idea occurred to us that I consider the most beautiful one in the core of Time Warp. Cancelling a cancellation message could be made indistinguishable from resending the original event message, and in that case a third-order cancellation message was indistinguishable from a first-order cancellation. There was an even–odd parity at play, and we formalized it in terms of a message–anti-message duality. A regular event message would be considered "positive" and a cancellation message would be considered "negative", and they were symmetrical. Either one could cause a rollback, and whenever two messages that are identical except for sign were enqueued in the same message queue they would "annihilate" and just disappear. The notion of anti-messages and annihilation made asynchronous distributed rollback work cleanly, and it did so even in all the complex cases. It worked even though the timing was nondeterministic and messages were asynchronous and arbitrarily delayed, and even when messages were not delivered in FIFO order, and even when there were cycles in the communication graph, and when multiple, asynchronously interacting and mutually interfering rollbacks were simultaneously in progress, and when anti-messages were delivered before the messages they were supposed to cancel, and it even worked when *all* anti-messages were delivered systematically more slowly than positive messages! Furthermore, the anti-message mechanism scaled easily to an arbitrarily high degree of parallelism. It seemed miraculous that such a simple, elegant mechanism as timestamped

anti-messages could also be so powerful, scalable, and robust. (It still seems that way to me.) Henry and I repeatedly discovered that every issue created by the introduction of distributed rollback seemed to have an elegant, efficient, but often surprising solution, unlike anything we had seen in our computer science experience. That gave us confidence that we were on a very significant research path.

For such a brand-new and different kind of algorithm the correctness and performance properties of asynchronous distributed rollback really should be formally proved. I had substantial background in program verification, so I went through a private exercise of *trying* to prove both weak and strong correctness. I believe I could have done it but it would have required developing of great deal of new formal machinery which would have been a distraction from my main goals at the time. Still, the exercise of trying convinced me privately that there were no flaws for a simulation with a finite number of LPs. I do not think that even today anyone has published a formal proof of correctness of the full Time Warp algorithm (e.g., including the cancelback protocol), but Bagrodia et al. (1991) may have come closest. Of course many analytical and empirical papers have been published on the performance properties of Time Warp.

In those first few weeks, Henry and I considered many variations on Time Warp. The state restoration parts of the rollback mechanisms we considered were all based on saving snapshots of the state of an LP, but we considered several possibilities, including incremental and variable frequency state-saving. We also considered various message cancellation schemes and distinguished two of them, *lazy* and *aggressive cancellation*. We defined *GVT* (global virtual time), and articulated how it resolved commitment issues such as I/O, error handling, and termination detection, and at the same time allowed us to recycle memory using a technique we dubbed *fossil collection*, intentionally echoing the term *garbage collection*. We considered how an LP going down an incorrect execution path might commit a runtime error or get into an infinite loop, and yet even those problems could be cleanly handled by properly implemented rollback. We dealt with the semantics of handling ties in virtual time, i.e., two events at the same LP at the same simulation time, and invented the superposition concept in which the entire set of tying event messages are processed together in a single event execution. We spent an inordinate amount of time worrying about the problem of repeatability with the use of pseudorandom number generators (PRNGs) in the context of asynchronous rollback. For a long time, I had something of a blind spot and failed to realize the obvious, that when you just treat the PRNG seed and state as part of the *model state* that could be rolled back, rather than as part of the *simulator state*, then there is no problem at all.

For the most part, it is not possible to separate Henry's and my contributions to the early ideas in Time Warp. However, I specifically remember that Henry came up with the name "Time Warp". At first I did not like it at all because it sounded like science fiction to me and I thought such a significant algorithm deserved a more dignified and serious name. Eventually I caved, however, because other people liked it and because I had nothing better to offer. It turned out to be a wonderful choice because once people heard the details of the algorithm the catchy name

really helped them remember it. I contributed the term "antimessages" myself. People instantly got the allusion to particles and antiparticles in physics, and that also helped sell the unorthodox idea to the research community.

Henry and I did the first proof-of-concept implementation of Time Warp on a network of four Xerox Dolphin workstations used at RAND. We wrote in InterLisp because we could get something working interactively very quickly. Henry was at the keyboard with me looking over his shoulder kibitzing at every line. After we got the Time Warp code barely turning over we needed a benchmark simulation model to demonstrate that it worked. We chose to write a parallel event-driven version of the cellular automaton known as the Game of Life because it was easy, it could scale it to any size, it had plenty of parallelism available, and it could be trivially load balanced. By giving each LP responsibility for larger or smaller "chunks" of the two-dimensional region we could control the event granularity and the communication-to-computation ratio. That model also forced us to deal with event message ties because each cell in the Game of Life is updated only when it gets simultaneous (in simulation time) inputs from its eight neighbors. And finally, The Game of Life was simple and deterministic and easy validate, so we would know instantly if there was a problem.

The first few times we ran it we got no speedup at all compared to sequential execution, and in fact we measured a severe slowdown. To fix this we had to do our measurements after hours when we were not competing for cycles and network bandwidth with other users of the Dolphins. We had to turn off background computations and system services, such as various daemons, demand paging and lisp garbage collection. We also had to turn off our own debugging and trace instrumentation, which otherwise did file I/O for each event. Only then, when all those heavy performance drags and sources of performance noise were eliminated were we able to see the parallelism and measure any speedup. As I recall, we achieved almost a 2.5x speedup on our four-node network under optimal conditions, though no records of those initial runs survive.

Eventually the collaboration between Henry and myself ended as we began moving in different directions. Henry had not yet finished his PhD dissertation and had to get back to it, and thus wanted to hold off for some months before he could resume work on Time Warp, whereas I had a tenure clock running and students to engage, and wanted to take the research to the next level immediately. Also, inexplicably, RAND did not support our desire to apply for additional research funding to allow us to bring more people to the project. Fortunately for me, as a USC professor, I was not subject to RAND management, and could get my own funding. Before the collaboration ended Henry and I wrote the first paper on Time Warp in the form of a RAND Tech Report (Jefferson and Sowizral 1982). The title, "Fast Concurrent Simulation Using the Time Warp Method, Part I: Local Control", hints at a forthcoming Part II, describing global control (GVT, etc.), but we never got around to writing it.

7.5 Conservative Versus Optimistic Synchronization

After Henry and I had developed the core Time Warp methods and while we were still working together I finally began to study the published literature on PDES, mostly that by Mani Chandy and Jay Misra. They approached the synchronization of discrete event simulation by what I viewed as more conventional means, using algorithms based on process block-and-resume primitives rather than rollback. They introduced the now classic Null Message algorithm (Chandy and Misra 1979), a similar version of which had been independently invented earlier by Bryant (1977). Later they introduced another another algorithm based on a repeated cycle of running to global deadlock and breaking the deadlock (Chandy and Misra 1981).

With the Null Message algorithm (also known as CMB after the inventors) the authors were targeting simulations of queueing systems, dataflow architectures, computer networks, and other systems that are characterized by a static set of components and a static graph of interactions among them. When I finally studied their algorithm my reaction at the time was not favorable. The algorithm required not just a static graph topology, but one using strictly FIFO communication channels among the LPs (called "lines" in their paper), with the event messages sent down each channel required to be in increasing timestamp order. It allowed neither dynamic creation of new LPs nor of new channels between LPs. These seemed to me to be strong restrictions on the class of discrete models that the algorithm could be applied to. Henry and I had military combat simulations in mind, and it never occurred to us to assume a static graph of interactions. We needed a method that could simulate models that were much more dynamically malleable, permitting any LP to send an unexpected event message to any other. One could do that within the CMB paradigm only by assuming a complete graph of channels among the set of n LPs, and that would entail $O(n^2)$ channels, and in some cases $O(n^2)$ messages (mostly null) per unit of simulation time, which was clearly unscalable. We also wanted to allow dynamic creation and destruction of LPs at runtime, a capability that was incompatible with a static graph of LPs.

The CMB restriction that in each channel event messages must be transmitted in increasing timestamp order had other, less obvious consequences. An LP A with four successive events at times 10, 20, 30, and 40 might want to send one message per event to another LP B with timestamps 90, 80, 70, and 60 respectively, so that the event messages must be processed at the receiver in the reverse of the order in which they were sent. Chandy and Misra's version was based on Hoare's Communicating Sequential Processes and could only handle an inverted sequence of n messages from A to B if B were divided into n separate LPs and there were at n separate channels from A to each of them, with each message sent along a different channel. Since the number of channels had to be static, there was always a static limit on how long a sequence of inverted order messages could be sent from A to B. As a practical matter it is rare for an LP to want to send a long inverted sequence of event messages, but there is no theoretical reason to adopt any static restriction, and

no such restriction is needed by either by the sequential discrete event algorithm or by Time Warp.

Finally, I was also concerned that the Null Message algorithm required additional logic in the model code to decide when to send null messages and what timestamp to send on them. Null messages were required to assure good performance and to avoid deadlocks. This extra logic sometimes required fairly deep understanding of the mechanics of parallel simulation on the part of the model programmer, understanding that may not have been within his or her expertise. In some cases even though deadlock was avoided, it was sometimes only barely avoided, possibly requiring a large number of null messages to make a small amount of simulation progress. In current terminology, we would say that such models have very poor lookahead, and it is inherently difficult for them to achieve good performance with any kind of conservative synchronization, including CMB. Such additional logic is not required in either the sequential algorithm or in Time Warp.

These strong limitations led to my low opinion of the Null Message algorithm *at the time*. It seemed to me at best a special-purpose simulation algorithm, not a general discrete event simulator. In fairness, it was always presented as a network simulation algorithm, but many, if not most, readers thought that it applied much more broadly. For all its good qualities, I was then acutely aware at the time of the Null Message algorithm's limitations.

I hasten to add here that despite those limitations the Null Message algorithm has stood the test of time. There are many important "network" models that fit perfectly within its restrictions, and for them, it is often ideal. Even nonnetwork models can often be profitably shoehorned into network form so that it applies. In the intervening years I have implemented it, applied it in real applications, and taught it several times myself, and I have grown to appreciate it much more.

But in the early 1980s, with the Null Message algorithm's limitations paramount in my mind, I developed something of a competitive attitude. I foresaw great difficulty in getting an exotic rollback-based algorithm like Time Warp to be taken seriously when there seemed to be much simpler, more approachable and more understandable algorithms based on conventional synchronization.

Around this time I took a tour of universities, lecturing about Time Warp, and I visited U. T. Austin where I met Chandy and Misra for the first time. The meeting was a little awkward as I recall. In giving my usual lecture on Time Warp I would have compared it to their published algorithms, and probably argued to the audience that Time Warp was a superior approach. That may not have been the most politic thing to do at their own university. For the first time upon meeting them I realized that they were both quite distinguished and senior to me and thus deserved some deference. I think they may have been mildly irritated at my presentation, though they were totally professional about it. As I recall, it was in that first conversation with them that the terms "optimistic" and "conservative" were adopted to apply to speculative, rollback-based methods versus unspeculative, process-blocking methods. I had been using the term "optimistic" to describe Time Warp-like methods, by analogy with optimistic transaction synchronization, and the natural contrasting

term I had been using was "pessimistic" to describe nonspeculative methods. Chandy and Misra, understandably, did not like their algorithms characterized as "pessimistic", and proposed, if I recall correctly, that we perhaps refer to the two approaches as "liberal" and "conservative". I was not happy with the overtones those two terms evoked. However that conversation went, and it is very fuzzy to me now, by the end we agreed that each of us would choose our own term to describe the methods we had developed. I chose "optimistic", and they chose "conservative". That is how that asymmetric pair of terms came to be adopted.

For maybe two decades or longer a rivalry of sorts continued between proponents of optimistic and conservative methods. Each camp had its advocates, who pointed out the strengths of their methods and the weaknesses of the others. These positions were often inherited by the next generation or two of graduate students as well. I recall one of the more dramatic moments in the competition was an occasion when a prominent professor boomed to a conference audience that "Conservative methods are doomed! Doomed!". Another moment in the opposite direction was publication by Nicol and Liu of a significant paper entitled "The Dark Side of Risk (What your mother never told you about Time Warp)" (Nicol and Liu 1997). We all laugh about the controversy now, but for a long while the positions were passionately held on both sides. I was partly responsible for this rivalry, but I was certainly not alone. In my defense, I felt that I had a very uphill battle to make the case that a radically different, even bizarre, algorithm such as Time Warp, should even be considered for PDES or any other application. Most researchers initially assumed that conservative PDES synchronization, which was invented earlier and was easier to understand and implement, was the natural and reasonable approach. I thus thought it was necessary not only to demonstrate the advantages of optimistic methods, but also to articulate what I thought of as the inherent limitations of conservative methods.

The rivalry slowly faded. Thankfully it never became personal. Echoes of it still crop up indirectly on panels and over drinks at conferences, especially among old timers of my generation. But today there is widespread recognition and understanding of the merits and limitations of both conservative and optimistic synchronization methods. My own position has now matured as well. I now believe that when and where there is good lookahead information that is easily computed, conservative methods will generally dominate optimistic methods. When there is not, optimistic methods will dominate. In complex models where some parts have good lookahead and others do not, a hybrid synchronization system is called for.

7.6 The Virtual Time Paper

In 1983, I wrote a new paper, "Virtual Time" (Jefferson 1985), based on Time Warp. My collaboration with Henry had ended, and for a lot of reasons it was not going to be possible for us to write a jointly authored paper. Thus, he was not a coauthor, though arguably he should have been. Henry, as a RAND employee, was

bound by their slow and ponderous rules for submitting anything for publication, whereas with a tenure clock running I could not afford the year-long submission delays he would be subject to. Instead, to justify my sole authorship, I included as much new material of my own as possible beyond what we had already published. The new paper framed the Time Warp algorithm as a *general purpose* synchronization protocol, useful for an array of applications beyond just simulation, such as database concurrency control (Jefferson and Witkowski 1984; Jefferson and Motro 1986). I still believe that it has potentially widespread value in parallel applications with complex synchronization, fault recovery, and load balancing requirements. The paper also presented an extended analogy between rollback-based synchronization and demand paging implementations of virtual memory, in which a rollback ("time fault") is considered analogous to a page fault ("space fault"). This analogy is what inspired the title, "Virtual Time". That paper became very well known, winning the Most Original Paper award at an international conference in 1983, and becoming over the years one of the most frequently cited papers in computer science, and the catalyst for much further research on optimistic synchronization.

7.7 Research with My Students at the University of Southern California

After my collaboration with Henry ended, there was still a huge amount of research to do to flesh out the mechanisms and provide useful theoretical underpinnings for Time Warp. Fortunately, I had funding and graduate students at the University of Southern California that enabled us to make further progress.

Critical path lower bound on time performance: One question we addressed was just how much parallelism is available in a parallel discrete event simulation, and how much of that parallelism Time Warp can capture. With my student Orna Berry, now a distinguished scientist and entrepreneur in Israel, we used a *critical path* approach to define the amount of parallelism available in a simulation model or, equivalently, to define a lower bound on the time it takes to execute a simulation in parallel. We were able to prove that no conservative algorithm could execute in less time than the length of its critical path. Time Warp was also bound by the same critical path length if it used *aggressive* cancellation but, in an extremely unexpected result, Berry proved in her dissertation that Time Warp with *lazy* cancellation could sometimes actually execute faster than the critical path lower bound (Berry and Jefferson 1985; Jefferson and Reiher 1991). Even now, after 30 years, this result is not widely appreciated, and its consequences are still relatively unexplored.

Flow control: Another issue we had not considered in the original Time Warp research was message flow control. (I was not even aware of flow control as a generic distributed systems issue until I started teaching operating system courses at

USC.) In the original work at RAND there was no mechanism to prevent fast-sending producer LPs from filling up the memory of slow-processing consumer LPs with queued-up event messages, fatally choking the simulation. Classical "windowing" flow control algorithms do not apply in Time Warp (or any other general PDES simulator) because there are no message "channels" and message streams from multiple senders are merged at the receiver, because unexpected messages from *new* senders can arrive any time without warning, and because the order in which messages must be processed at the receiver (timestamp order) is not generally the same as the order in which they are sent, nor the order in which they arrive. Eventually, Darren West at Jade Simulations in Calgary, and I, working partially independently, came up with a very elegant solution, one based on the idea that a receiving LP whose incoming message queue was excessively long could send high-timestamped messages back to their sending LPs to make room for incoming messages with lower timestamps, in a protocol we called *cancelback*. When the sending LP receives a sentback message, it rolls back to before it sent the message, and executes forward again and resends the message later (Jefferson 1990).

The idea of sending a message backward from receiver to sender is another highly unorthodox feature of Time Warp that cannot work in most parallel computation paradigms because they lack the ability to roll back. Sending messages backward is a communication idea that is nicely symmetric to the computational idea of rollback, and meshes perfectly with it. That observation and others led to greater emphasis on elegance and symmetry in future presentations of the Time Warp algorithm.

Unfortunately, even though Time Warp has been implemented many times in the last 30 years, most implementations leave out the cancelback protocol. This leaves them open to unpredictable and unrepeatable runtime failures due to memory exhaustion in an LP. That kind of nondeterministic failure behavior is essentially a Heisenbug, and when it occurs people can waste a huge amount of time trying to figure out what the problem is or work around it. Fortunately, experience so far shows this does not happen very often, probably because the models we are interested in tend to be approximately well balanced, or because some kind of active throttling of optimism is used. But I expect that as the complexity of models increases, especially with federated and multiscale models, this memory management hazard will also increase in urgency. We should consider the cancelback protocol as a fundamental part of any Time Warp implementation.

Global memory management and memory bounds: The consideration of flow control led to a larger concern of more global memory management, and the question of the minimal memory requirements for a Time Warp simulation. It was obvious that to get good performance a Time Warp simulation would normally require at least several times more memory than an equivalent sequential simulation because it had to store both a sequence of snapshots of each LP's state *and*

previously processed event messages *and* anti-messages for every message the LP sent, and it had to retain them at least as far back as GVT. There was a danger that the memory requirements would grow without bound, and the simulation would be unstable for that reason. We needed a theory for how to manage Time Warp memory, and as part of that we wanted to know the *minimal* amount of memory required for a Time Warp simulation to complete (though more memory was always better).

My student, Anat Gafni, proved in her dissertation that Time Warp, using both the cancelback and fossil collection protocols together, could be guaranteed to complete a simulation if given no more than about twice the memory that a sequential execution of the same simulation would require (Gafni 1985). We later improved that result by a factor of two, so that in theory Time Warp could complete in about the *same* amount of memory as a sequential execution when running on a shared memory (or virtual shared memory) platform (Jefferson 1990). Of course, the runtime performance would be terrible when Time Warp is constrained to run with memory near the minimum, but the result made the point that Time Warp could be space optimal. Surprisingly, I was also able to prove that asynchronous conservative methods were far from space optimal in general. I constructed artificial models that, with unfortunate timing, could require many times the memory of the sequential execution, although that was not typical.

Symmetry: The success of the cancelback protocol, in which message sendback was a direct analog to computational rollback, and the success of the analogy between virtual time and virtual memory, led me to adopt *symmetry* as an explicit goal in the Time Warp algorithm. Symmetry helped me present the algorithm more compactly and convincingly, and it led me to search for other places in the algorithm that had near symmetries and to correct them to make them perfect. Over the years I came to recognize a lot of symmetries, including message–anti-message symmetry, state–message symmetry, forward–backward time symmetry, forward–backward message transmission symmetry (cancelback), virtual memory–virtual time symmetry, and others. In the end, the bedrock reason I personally remained inspired by optimistic simulation was fundamentally aesthetic. I did not see similar symmetries in conservative methods, so I did not find them so compelling.

These results from the 1980s gave us confidence that optimistic methods, unorthodox as they were, should be taken seriously, both practically and theoretically, and that conservative methods had some genuine limitations. But the real tests would have to come with a serious parallel implementation of Time Warp and performance studies using both benchmark models and realistic models. We needed to demonstrate that Time Warp, with its heavy overheads and its poorly understood dynamical behavior, could nonetheless achieve real speedup from parallelism on real applications, and could be competitive with conservative algorithms in at least some useful application areas.

7.8 The Time Warp Operating System at Jet Propulsion Laboratory

On a visit to Caltech I had the opportunity to give a series of two talks. Since it is hard to get people to come to a second talk after a week's interregnum I tried a dramatic trick to attract them back. I described the PDES synchronization problem in the first lecture and convinced the audience that the problem, as I framed it, was essentially impossible to solve. I left a cliffhanger, promising to resolve it a week later in the second talk. When that day came, I had a full room and presented Time Warp, cleanly solving all the apparent problems. It was the introduction of rollback, which I had not mentioned in the first talk, that made the problem as I had framed it soluble. Those two lectures turned out to be especially important subsequently because the audience was full of physics-oriented people from the Jet Propulsion Laboratory (JPL) and Caltech who appreciated the analogies between Time Warp and physics (symmetries, anti-messages, etc.). Soon afterward when we started a multiyear Time Warp development effort at the Jet Propulsion Laboratory, some of the core members of that team, especially Brian Beckman, had been present in that audience, and others who were there became supportive of the project.

By 1984 I had moved to UCLA and begun the relationship with JPL that lasted for 7 years. A coincidence of three circumstances made this possible. First the Army, a sponsor at JPL, was interested in speeding up combat models and looked to parallel discrete event simulation as a key enabler. Second, researchers at Caltech and JPL were designing and building a parallel computer with a new architecture, the Caltech hypercube, and were looking for projects that would make constructive use of it and demonstrate its value. And finally, because my collaboration at RAND had ended, I was available and eager to work on the project and was invited to lead it.

The Caltech Hypercube was a 32-node distributed-memory cluster with Intel 80286/87 processors connected by 128 KB/s communication channels in a 5-D hypercube topology. Each node had 256 KB of RAM. It was an ideal machine on which to build the first serious implementation of Time Warp. At that time 32 nodes constituted a very large parallel computer. It was much larger, and more tightly coupled, than the four-node network Henry and I had used—large enough that we would be able to demonstrate significant parallelism and do useful scaling studies. The fact that the Hypercube did not have shared memory was in my view an advantage. I did not want to be tempted to use shared memory as a performance crutch in any way, since any such dependence would leave doubt about the scalability of Time Warp on platforms larger than the practical limit of shared memory.

The Caltech Hypercube had no operating system per se that we could use. This was before the first release of Linux, and long before it became almost a de facto standard OS for cluster machines. The only system software it had was what today we would call a two-sided, synchronous, order-preserving message system, like a primitive MPI that was intended to support what we would today call the Single Program Multiple Data (SPMD) parallel programming model. But we could not use

that because Time Warp needed a one-sided, asynchronous, interrupting message system, and we did not need order preservation. Thus, we had to build our own messaging layer, and that layer rested essentially on the "bare metal" of the hypercube. Even this I did not view as a handicap because it fit with my view that Time Warp should be thought of not just as a *simulator*, but as a special-purpose *operating system*. The project was therefore named *TWOS*—the Time Warp Operating System (Jefferson et al. 1985, 1987; Wieland et al. 1989).

I think of Time Warp more as an operating system than as an application because a full implementation requires the same software components as an operating system for a parallel machine, but with alternative, virtual time- and rollback-friendly algorithms in place of the classical ones.

- Time Warp needs a process scheduler with a lowest virtual-time-first discipline rather than any variation on round-robin.
- Its primary synchronization is based on asynchronous interrupting messages and general distributed rollback, not on locks, semaphores or process blocking.
- It needs timestamp-order priority queues with anti-message annihilation, rather than FIFO message queues.
- Its message flow control and global storage management need to be based on fossil collection and cancelback rather than garbage collection and windowing.
- It needs daemons for distribution of GVT (global virtual time), using what today would be called asynchronous all-to-all reduction.
- It needs special normal and abnormal termination detection and error handling, based on GVT.
- It needs special I/O commitment, also keyed to GVT. (It would be nice to also have a file system that supports rollback, but no one has ever built one.)
- It needs custom instrumentation to measure quantities such as events executed, events rolled back, message annihilations, message cancelbacks, and various other performance metrics unique to Time Warp that have no analog in conventional operating systems.

Later in its development, Time Warp also needed special mechanisms for rollback-friendly LP creation and destruction, dynamic LP migration to support load balancing (Reiher and Jefferson 1990), and advanced dynamic message routing to deliver messages to migrating target processes (Ravi and Jefferson 1988).

Despite these considerations, today everyone implements Time Warp as a runtime system *on top of an OS*, rather than as an OS itself. But however practical that decision is, we should be aware that there are performance costs to it. It entails two levels of scheduling, two levels each of synchronization, message queuing, memory management, error handling, and termination detection, as well as reliance on a level of polling for incoming messages.

The TWOS project at JPL was fairly large, with a full-time development staff of anywhere from 8 to 12 at any one time over the 7 years. Two people wrote the core Time Warp algorithms, initially Brian Beckman and later Peter Reiher, both of whom did brilliant work. The project would never have succeeded without them.

Phil Hontalas had the highly technical task of writing the low-level asynchronous messaging system complete with routing and interrupt handling, etc., and then porting it twice to subsequent parallel machines. Mike DiLoreto was a master debugger and performance specialist jack-of-all trades. John Wedell built a high performance sequential simulator that was semantically identical to the TWOS simulator and used for performance comparisons. He kept improving its performance, which challenged the team working on TWOS since its performance was evaluated relative to the performance of Wedell's sequential engine. Fred Wieland built the main parallel benchmark simulations and wargame models for the Army. Steve Bellenot did R&D on GVT algorithms. Van Warren was our graphics specialist. Others, including Leo Blume, Joe Ruffles, Kathy Sturdevant, Larry Hawley, Abe Feinberg, Pierre Laroche, John Spagnuolo, Todd Litwin and two fine interns, Maria Ebling and Matthew Presley, contributed benchmarks, instrumentation, documentation, ran performance studies, etc. Jack Tupman and Herb Younger were our managers interfacing with the Army sponsors, JPL upper management and the proprietors of the Hypercube, and husbanding the finances.

After the first few years, JPL built a new, much more powerful hypercube: the JPL Mark III. It had 64 nodes arranged in a 6-D hypercube, and was based on the Motorola 68020/68881 processor/coprocessor pair with a comparatively whopping 4 MB of RAM per node! Still later our project purchased a BBN Butterfly GP1000, a shared memory machine with 112 nodes, also with 68020/68881 processors and 4 MB per node. We ported TWOS to each of these machines in succession, which allowed us to run still larger models, achieve higher degrees of parallelism, and demonstrate the portability and scalability of Time Warp.

The following graphs, which are re-scans of the original plastic transparencies I used in 1988 in various presentations, illustrate the performance of TWOS about midway through the 7-year project. Figure 7.1 shows a strong scaling study of a military ground combat model called STB88 that the Army commissioned us to build as a benchmark. It had 380 LPs and was run for 326,997 events. The graph shows speedup of the combat model as a function of the number of nodes used on the Mark III Hypercube. The speedup is relative to the performance of our fast sequential simulator, not to Time Warp executing on one node. That is important because if we had chosen the performance of our algorithm on one node as a basis for comparison, which parallel computation researchers often did in those days, it would have yielded much higher but quite artificial speedup values, since a one-node Time Warp execution would pay high and unnecessary overheads for support of rollback and would consequently run much slower than our sequential simulator. The curve in Fig. 7.1 shows approximately linear scaling, from a speedup of about 2.5 on 4 nodes to about 16.5 on 32 nodes. The data in the documentation box shows that the sequential simulation took 1.4 h, while the time on 32 nodes under Time Warp was 5.3 min. The initials of Fred P. Wieland in the last line of the box show that he performed the runs. The sponsor was very happy, and so were we—we put this graph on a T-shirt!

The next two figures show a performance study a few months later of TWOS running essentially the same combat model as in Fig. 7.1, but this time done on the

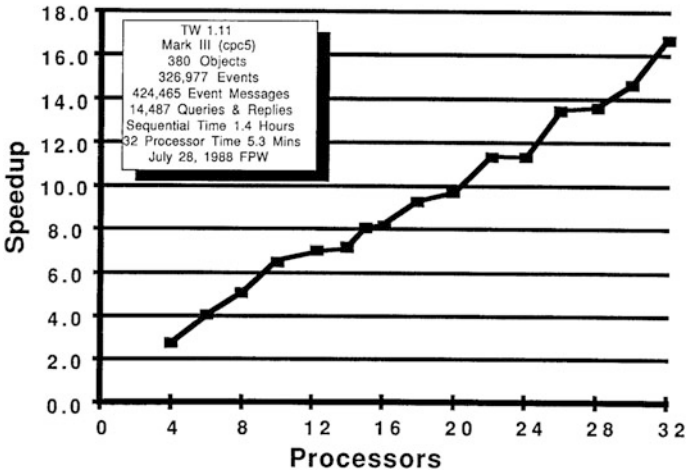


Fig. 7.1 Strong scaling study of TWOS relative to sequential execution of an Army combat model (STB88) running on the JPL Mark III Hypercube in July, 1988

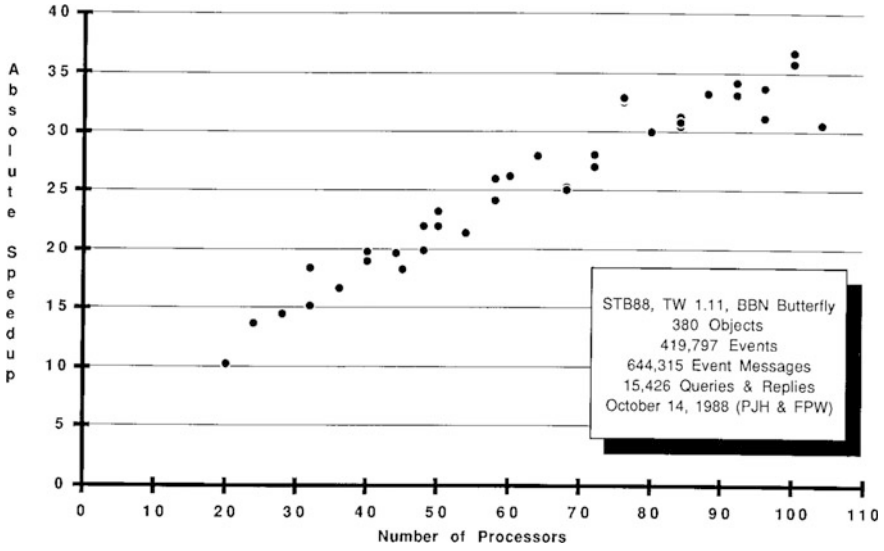


Fig. 7.2 Strong scaling study of TWOS relative to sequential execution of an Army combat model STB88 running on the BBN Butterfly in October 1988

BBN Butterfly by Fred Wieland and Phil Hontalas. Figure 7.2 is another plot of speedup as a function of the number of nodes applied. In this case, we reached a speedup factor of over 35x using 100 nodes and with an approximately linear speedup almost to full scale. We did some runs two or three times, and you can see up to 10% variation in performance even with the exact same configuration

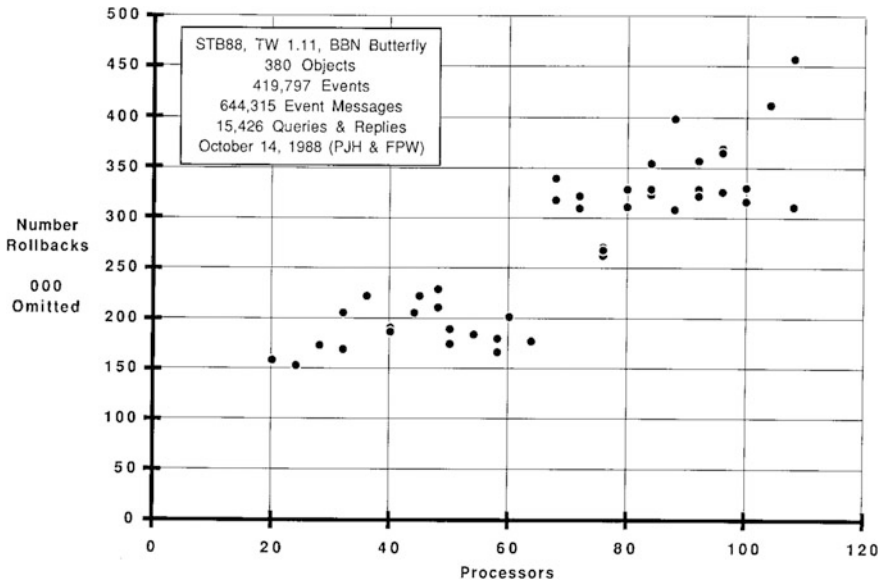


Fig. 7.3 Rollbacks as a function of number of nodes, using the same runs as in Fig. 7.2

measured twice. The variation shown between runs on different numbers of nodes is larger, and is primarily because the combat model was irregular in structure and the load was not always as well balanced at one scale as it was at another.

Figure 7.3 shows a plot of the number of rollbacks measured in the exact same runs plotted in Fig. 7.2. (It may not appear that way because in both graphs there are cases where multiple measurements at the same scale were so close that points coincide.) Figure 7.3 illustrates the fact that in Time Warp the number of rollbacks generally increases as a model is spread over more and more processors. In fact when running over 100 nodes, there were over 400,000 rollbacks when the total number of committed events was 419,797. Although a huge percentage of the events were rolled back and executed more than once, the speedup as shown in Fig. 7.2 continues to increase, at least up to the scale reached in this study.

This result is characteristic of Time Warp and very surprising to people new to it. How can it be that the speedup continues to increase even though the number of rollbacks does also? It is because the great majority of those rollbacks are for events that are way off the critical path of the computation, and thus they do not slow down the global progress of the simulation. While there is such a thing as excessive optimism and too many rollbacks, this graph makes it clear that in tuning a Time Warp simulation one must not just naïvely make it a goal to reduce the number of rollbacks.

7.9 Jade Simulations

During roughly the same years as the JPL project, Brian Unger at the University of Calgary started a company, Jade Simulations, that had the goal of commercializing Time Warp. I was a Board member and advisor. Jade built a completely new distributed implementation of Time Warp that ran well on both networks and clusters. Unfortunately Jade never successfully found a market, partly I think because the relentless acceleration of sequential computation due to Moore's Law allowed people who needed higher performance simulations to just wait a couple of years, and partly because Jade was a Canadian company that was not permitted to compete for U.S. military simulation business. But I also think that Jade was just ahead of its time, when parallel computers were still very expensive. It would take several more years before any market matured sufficiently to make commercialization viable.

7.10 Subsequent Years

The most significant new PDES idea in many years came in 1999 when Richard Fujimoto and his students, Kalyan Perumalla and Chris Carothers, introduced the idea of reversible computation (Carothers et al. 1999). This was a dramatically different and more efficient approach to implementing rollback, and leads some tantalizing programming theory and programming language ideas as well. At the time I had left the field of PDES, and only learned about it several years later when I reconnected. But then I was deeply impressed. Eventually, when I found myself in position to work on it again at Lawrence Livermore National Laboratory, Markus Schordan, at my instigation, built a compiler that can take literally any program written in C++ and create reverse code for it (Schordan et al. 2015; Schordan 2016). We hope this development may make reversible computation the standard way of accomplishing rollback and remove most of the last remaining software engineering barriers to the widespread adoption of optimistic synchronization.

Time Warp has been implemented many times in the years since the RAND, JPL, and Jade projects. Richard Fujimoto, as he describes in the next section of this chapter, created Georgia Tech Time Warp (GTW), originally as a platform through which to study major improvements that could be made to Time Warp in a shared memory environment. He was the first person to do fair comparison studies between conservative and optimistic methods on the same problem and to verify that in many cases (but not all) the optimistic methods could indeed outperform conservative methods. Later his students developed their own implementations of Time Warp, derived to some extent from their experience with GTW. Kalyan Perumalla, now at Oak Ridge National Laboratory, developed the μsik simulator, which he still uses for advanced studies of PDES (Perumalla 2005). And Chris

Carothers, now at RPI, developed the ROSS simulator, which is open source and available to all researchers who wish to work with it (Carothers et al. 2002).

In at least one respect ROSS is now the state of the art in Time Warp implementations. It holds the world record to date (2017) for the largest scale and fastest discrete event simulations ever executed. In 2013 my colleagues Peter Barnes, Chris Carothers and Justin LaPre and I ran a series of benchmark runs on the *Sequoia* supercomputer at Lawrence Livermore National Laboratory, which at that time was the second fastest computer in the world. On the standard PHold benchmark with 251 million LPs we achieved a sustained 5.04 billion events per second using almost 2 million cores and almost 8 million hardware threads (Barnes et al. 2013). Those runs proved that the Time Warp algorithm can scale to gigantic degrees of parallelism.

In the 35 years since Henry Sowizral and I built the first primitive four-node implementation, Time Warp has achieved over a million-fold increase in demonstrated parallelism. While I anticipate a temporary pause in this peak performance increase as the architectures of supercomputers are changing in the current era in ways that do not benefit Time Warp or PDES, the future is nonetheless long and it is very hard to imagine what the next 35 years might bring.

7.11 My Adventures in Time Warp: Perspectives by Richard Fujimoto

The second section of this chapter provides the perspectives of Richard Fujimoto.

7.12 Beginnings

I first became interested in parallel discrete event simulation (PDES) when I was a doctoral student at the University of California in Berkeley (1978–1983). As an undergraduate student at the University of Illinois I became interested in computer architecture, and embarked on a doctoral dissertation at Berkeley looking at high speed switches to interconnect microprocessors, which at that time were just becoming powerful enough to be interesting, to create parallel computers. In order to evaluate our ideas about switches we needed parallel applications to generate realistic message traffic. It immediately became clear to me then that the discrete event simulator I had developed to evaluate our switches would be a perfect benchmark program. This of course led to consideration of the synchronization problem, and ideas akin to those developed by Chandy, Misra, and Bryant (published a few years earlier, but unknown to me) came to mind, but my main interest was in hardware design, so I did not embark on any serious studies of the problem while I was a graduate student.

It was not until years later that I renewed my interest in this area. I then came across Chandy and Misra's and other's papers concerning conservative synchronization, and Jefferson's work on Time Warp. At the time, the initial algorithms were in the published literature, but no one knew which approach was better, and under what circumstances one approach might dominate the other. My attention focused on doing a serious comparison of these algorithms. This work launched a career-long exploration of parallel and distributed simulation techniques.

7.13 Conservative Versus Optimistic Performance

Foremost in my thinking was the need to have a very efficient sequential simulation to determine the speedup obtained using parallel computing. This led to an exploration of the literature in event list implementations for sequential simulations. A paper by Doug Jones comparing different priority queues for discrete event simulations caught my attention (Jones 1986). This comparison, and others that had appeared at the time, were all based on something called the HOLD model, which seemed to be the standard benchmark for evaluating sequential event list performance. I developed a parallel version of the HOLD model, first described in (Fujimoto 1988), which later become known as PHOLD (Parallel HOLD). I used PHOLD in my initial comparisons of the Chandy/Misra/Bryant null message and deadlock detection and recover algorithms, and later Time Warp (Fujimoto 1990). PHOLD continues to be used to this day to evaluate parallel simulation performance.

There were a couple of central conclusions that came from these comparisons that drove much of my work that followed. First, these results showed conclusively that both Time Warp and the conservative algorithms could achieve excellent speedup relative to efficient sequential implementations using state-of-the-art event list data structures. Prior to that time much of the work had reported speedups of conservative methods, but often compared the parallel implementation against a sequential execution of the parallel algorithm, or did not use a particularly efficient sequential implementation. For example, I recall some results reported super-linear speedup. But on closer examination, this work compared performance against a sequential simulator using a linear list implementation of the event list, an approach that becomes very inefficient for large numbers of pending events, leading to inflated parallel performance. There was a dearth of empirical results reporting Time Warp performance at the time, though measurements of the Time Warp Operating System (TWOS) for realistic applications would soon appear.

A second conclusion from this (and other) work was that conservative algorithms relied on exploiting knowledge of the simulation application in order to extract good "lookahead" information, essential to obtaining efficient parallel execution. Briefly, lookahead is a guarantee made by the simulator that any new events it schedules are at least a certain amount of simulation time into the future. The greater the lookahead, the better. Intimate knowledge of the application is

required in order to make such a guarantee. This suggested significant drawbacks with conservative synchronization algorithms. It meant that the application itself had to possess good lookahead properties; not all applications possessed such properties. For example, if two entities in the simulation could interact in a small amount of time, e.g., two radios in a simulation of a wireless network could instantaneously communicate, large lookahead may be difficult or impossible to obtain. Further, even if one could build the simulation to have good lookahead, if the simulation model had to be later modified, such modification might destroy these lookahead properties, defeating the parallelization approach. For example, in a queueing network simulation, if one added high priority jobs that preempted service from lower priority jobs, it greatly reduces the lookahead, and could lead to a dramatic reduction in performance, even if there were only a few high priority jobs in the system. As a result, conservative simulation applications were prone to becoming brittle in that if details of the simulation model changed, the original parallelization approach might not yield acceptable performance, or it might not even run at all.

It was quite apparent to me early on that two facts seemed inescapable. The first was that Time Warp stood the best chance of realizing a general purpose parallel discrete event simulation engine over which a variety of applications could be developed. This, of course, seemed to be the most viable path for the technology to see real-world use and have widespread impact because it would enable exploitation of the technology in a wide variety of application domains without intimate knowledge of parallel processing or the synchronization mechanism. In much the same way the developers of sequential simulation models did not need to be concerned with the priority queue data structure that was used. Expecting domain experts, who would be the ones who developed the simulation models, to be experts in parallel computation and PDES seemed a stretch. The reliance of conservative methods on intimate knowledge of the application called for domain experts to have much more expertise in PDES than I thought realistic.

A second observation was that there were significant challenges that needed to be addressed for Time Warp to yield acceptable performance. Two hurdles seemed obvious. The first problem was one that everyone immediately understands as soon as they learn about Time Warp, namely that the computation could spend most of its time rolling back events and recovering from errors rather than completing computations for the simulation model. This problem appeared to be solvable, however, and approaches to addressing this problem began to appear soon after Time Warp had been invented. A second problem that seemed somewhat more difficult was the need for state-saving in order to allow the computation to be rolled back. This could consume much time and memory. Both of these problems were important and interesting, but the state-saving problem was the one that captured my attention first. With my background and interest in computer architecture, I envisioned this was something that could be addressed with hardware.

7.14 The Rollback Chip, Virtual Time Machine, and Reverse Execution

As an undergraduate I had been enamored with clever techniques to manipulate memory addresses in digital circuits to implement cache and virtual memory systems. It seemed natural to use such techniques to implement state-saving in hardware for Time Warp. One only needed to intercept memory writes, and modify the memory address to preserve the original contents of the memory location. Some additional work was needed to implement memory reads, in order to ensure the correct version of memory was accessed. This was straightforward to accomplish in hardware by keeping track of which blocks of memory had been written. These ideas led to a kind of memory management circuit that we called the rollback chip (Fujimoto et al. 1992). Separately, I was approached by two electrical engineers who were keen to develop prototype hardware, and were able to get some funding for the same, resulting in an initial proof-of-concept prototype implementation of the system (Buzzell et al. 1990).

The next logical step beyond the rollback chip work was to extend hardware support beyond state-saving, to other aspects of Time Warp. Here, the conceptual model I found useful was the data dependence, or task graph, where nodes represent event computations, and links represent dependencies between events. Specifically, each event within a logical process (LP) depends on (or more precisely, *could* depend on) the preceding event in the LP with the next smaller time stamp. And if one event schedules a second, then there is obviously a dependence of the scheduled event on the one that scheduled it. Data dependence graphs had been around for some time and were widely used to study parallel computations. One of David Jefferson's students, Orna Berry, used this model to determine properties of the computation such as average parallelism and minimum possible execution time (Berry and Jefferson 1985). Task graphs also greatly influenced my work in developing Time Warp software, described later. I realized that a hardware representation of the task graph, stored in the shared memory of a multiprocessor system, could be used as the basis for a general parallel computer based on Time Warp.

I called the resulting parallel computer the Virtual Time Machine (VTM) (Fujimoto 1989a, b), giving a nod to Jefferson's original Time Warp paper entitled Virtual Time (Jefferson 1985). Admittedly, I also liked the sci-fi-ish nature of the name. A distinction between the VTM design and Time Warp is that it did away with logical processes, and used a shared memory rather than a message-based computation model. At the core of VTM was the space-time memory system, a memory system addressed by a time value in addition to a conventional memory address.

We envisioned the VTM to be a general purpose parallel computer using rollback as its core synchronization primitive that could be used broadly for parallel computation, not just simulations. After all, one could, in principle, take any sequential computation, divide its execution up into blocks of instructions, assign each block a time stamp reflecting its sequential order of execution, and the result

would be a discrete event simulation computation where each block represented an event. We envisioned the VTM could be used to automatically parallelize sequential computations, something compiler writers at the time had been struggling with for years.

This vision proved to be a bridge-too-far, however. We focused on the use of the VTM to automatically parallelize sequential discrete event simulations as a first step toward this bigger, broader objective. This was examined by Jya-Jang Tsai for his doctoral research, who had completed some of the early simulation studies of the rollback chip. But even in this friendlier domain, it was difficult to reduce the computation overheads to an acceptable level, and only a modest amount of success was achieved. The computer architecture world at the time was moving toward reduced-instruction-set-computers (RISC) so the trend was toward simpler rather than more complex, and the VTM approach did not fit well into this movement. At a deeper level, sequential programs not written for parallel execution often do not exhibit sufficient parallelism for an approach such as this to succeed; the code needs to be structured for parallel execution earlier on in the software development cycle. So, we did not continue to pursue the VTM hardware architecture, though the VTM ideas influenced much of our later work in distributed simulation.

For example, a software-based distributed simulation approach derived from the Virtual Time Machine and Time Warp was a technique we termed "ad hoc distributed simulations" (Fujimoto et al. 2007). The context for this work was in the use of distributed simulation online to manage operational systems. Here we repurposed the space-time memory used in the VTM to hold a projected future state of the system. Simulations computed forward, ahead of wall clock time to predict future system states, which were stored in the space-time memory. If measurements of the actual system deviated from predictions by more than a specified threshold, a Time Warp style rollback mechanism was used to automatically correct the predictions. Although the approach is applicable to a variety of systems, our work largely focused on evaluating this approach for traffic management applications, and more broadly, systems modeled as queueing networks (Huang et al. 2012).

Getting back to Time Warp, since the VTM hardware approach was never able to get much traction, we focused more on software-based ideas. In particular, the next chapter in our work in addressing Time Warp's state-saving problem was to move away with state-saving entirely, and instead employ a reverse execution method to undo incorrect computations. Chris Carothers and Kalyan Perumalla pursued much of this work, with Kalyan pursuing it for his doctoral research (Carothers et al. 1999). Both continued to work on reverse computation after completing their work at Georgia Tech. Actually, reverse computation did not entirely eliminate state-saving because some computations are inherently irreversible; state-saving was used as a fallback approach in situations where the inverse computation could not be created. Nevertheless, reverse computation can significantly reduce the memory footprint of Time Warp programs and greatly reduce the time required for state-saving. Much of our work in reverse computation focused on proving its viability for various simulation applications. In addition to standard benchmarks such as communication networks, some of this effort focused

on applying the technique to scientific computing applications such as modeling the earth's magnetosphere in collaboration with Homa Karimabadi and his group at Scibernet, a start-up company in California, and a graduate student named Yarong Tang (Tang et al. 2006).

Developing the code to invert a computation is not straightforward, and prone to error. To address this problem Kalyan Perumalla developed a reverse execution compiler to automatically instrument the forward execution and generate the reverse code for his PhD dissertation. This work was subsequently continued in collaboration with David Jefferson and others at Lawrence Livermore National Laboratory (LLNL) and Georgia Tech to create the backstroke compiler (Vulov et al. 2011; Schordan et al. 2015, 2016).

7.15 Simulation Ensembles

We never reached the point of creating a hardware realization of the Virtual Time Machine. But as mentioned earlier, the computation model it used formed the basis of other work. One line of research focused on accelerating the completion of multiple simulation runs, called ensemble simulations. Virtually all simulation studies require many runs. Often these runs are similar, and have many computations in common. Our focus was on performing these common computations only once, and sharing their results. As will be seen, some ideas from Time Warp can be used to achieve this goal.

We used task graphs not unlike those used in VTM extensively in this work. Specifically, Steve Ferenci developed something we called updateable simulations (Ferenci et al. 2002). The basic idea is to record the task graph for a computation. Then, to complete subsequent runs of simulations similar to the recorded one, use a VTM/Time Warp style rollback mechanism to "update" the simulation execution in accordance with differences in the new run.

A related idea, developed earlier by Maria Hybinette for her doctoral research, was to use an incremental cloning mechanism to compute a set of similar simulation runs (Hybinette and Fujimoto 2001). This work used the traditional PDES logical process model and Time Warp, although her work did not require the use of Time Warp. Motivated by the use of Time Warp to assess alternate possible futures for air traffic control simulations, the idea was to identify points in the future when decisions would need to be made, and then to replicate or clone the parallel simulation to concurrently explore these alternate futures. The central idea in Maria's work was to use an incremental LP replication method where LPs were replicated as needed as the computations of the different runs diverged. This enabled us to share the results that were common among the different replications.

7.16 Georgia Tech Time Warp

Let me "roll back" now to 1988 and come back to our software implementation of Time Warp. I recognized early on that there were several important challenges in realizing an efficient implementation. My early work focused on developing techniques to create a fast Time Warp system using a variety of methods developed by our research group. These techniques were incorporated into a software system we called Georgia Tech Time Warp (GTW).

My initial work focused on developing an efficient implementation of Time Warp on shared memory multiprocessors. Motivated by the task graph model, my first implementation focused on, in effect, storing the task graph in the shared memory of the multiprocessor system. In particular, when one event scheduled another, we simply stored a pointer from the scheduling event to the event being scheduled. This greatly simplified the Time Warp implementation because it eliminated the need for separate anti-messages, with the pointer effectively implementing an anti-message. It also eliminated the need to create a copy of each scheduled message, and avoided the need to search for the matching positive message when an anti-message was received. I called this technique direct cancellation (Fujimoto 1989b). Direct cancellation allowed one to rapidly track down and correct errors resulting from out of order executions. I believed this was important because while one used anti-messages to correct these computations, the wrong computations themselves were spreading, so it was important to track down the errors as quickly as possible to minimize the damage they caused. The resulting shared memory data structure used to implement Time Warp motivated the space-time memory used subsequently in the Virtual Time Machine work described earlier.

We developed several other innovations to create an efficient shared memory Time Warp system, described in (Das et al. 1994; Fujimoto 2000a, b). For example, working with PhD student Maria Hybinette, we developed an efficient shared memory algorithm for computing Global Virtual Time (GVT) (Fujimoto and Hybinette 1997). GVT is needed in Time Warp to determine a lower bound on the time stamp of future rollbacks, enabling reclamation of memory (fossil collection) and committing irrevocable operations such as I/O. We also developed an efficient, "lazy" approach to implement fossil collection that we called on-the-fly fossil collection. It avoided the need to search through lists to identify chunks of memory that needed to be reclaimed.

Sequential discrete event simulations often use a mechanism to "unschedule" previously scheduled events. This is necessary to model activities such as preemption where some "normal" activity is interrupted by some other event, e.g., in a queueing network, servicing a job might be preempted by the arrival of a higher priority job. We realized that anti-messages, which needed to be implemented in Time Warp for synchronization purposes, could be used to implement this unscheduling of events. The difference now is that the generation of anti-messages was triggered by the application program itself, not the underlying simulation

engine. A small wrinkle was that these application-invoked event cancellations could themselves be rolled back, but this was easily accomplished by sending a copy of the original event. The simplicity of the mechanism derived from the simplicity and symmetry of the original Time Warp algorithm. Working with my PhD student Samir Das we developed the algorithms and an implementation, and evaluated its performance. Later, we found that Greg Lomow working with Brian Unger at the University of Calgary independently developed a similar mechanism, but had not published the work because they were working on commercializing the effort in the context of a company called Jade Simulations that Brian was developing. We agreed to jointly publish the work (Lomow et al. 1991).

As we developed new innovations we incorporated them into the GTW Time Warp software (Das et al. 1994). Based on my initial implementation that focused on the direct cancellation mechanism, Samir Das did much of the work in furthering the development of the system. Various developments were incorporated into GTW in the years ahead, including extension to message passing and distributed computing architectures. GTW formed the basis for most of our experimental work in Time Warp such as the reverse execution work described earlier.

GTW was used for a variety of applications, but perhaps the most memorable was its use to create fast air traffic control simulations. In the summer of 1989, I had the pleasure of spending a 3-month period working with David Jefferson and his research group at the Jet Propulsion Laboratory. They were working on the Time Warp Operating System (TWOS) project. During this period, I met Fred Wieland who was one of the TWOS developers, focusing on implementing a combat simulation model. After TWOS, Fred went on to work on simulation applications in the aviation industry with the MITRE Corporation. Fred is a likable guy, and we had good discussions on Time Warp, parallel discrete event simulation, and aviation over many years that followed. Anyway, Fred was interested in creating a fast parallel simulator to model the U.S. aviation system, and learned about GTW. He embarked on developing an air traffic simulation called the Detailed Policy Assessment Tool (DPAT) on our Time Warp system (Mitre Corp. 1997). DPAT was deployed for air traffic analyses, making it at the time one of the few real-world deployments of Time Warp for real-world applications (Wieland 2001). The DPAT/GTW system subsequently formed a basis for Maria Hybinette's work in parallel simulation cloning, discussed earlier.

7.17 Analytic Models, Memory, and Load Management

In 1990, I began work with Ian Akyildiz, a colleague at Georgia Tech, on a project funded by the Ballistic Missile Defense Organization (BMDO), later renamed the Missile Defense Agency (MDA). BMDO was working on developing the technologies for the Strategic Defense Initiative more commonly known as the "Star Wars" program that aimed, among other things, to create the ability to intercept incoming ballistic missiles. Simulation was a key part of the program. The program

manager, Lou Lome, was particularly interested in developing PDES technology, and funded some of our work. This project was the first in a string of BMDO projects that funded our work on Time Warp. Our program was part of BMDO's basic research program aimed to develop the underlying technologies to create fast parallel and distributed simulations. Our basic research program ran another 10 years; I was later told we were the only group still funded by this program when BMDO decided to end its basic research program.

Ian had background in developing mathematical performance models. This led to a collaboration focusing on developing analytic models of Time Warp. With PhD student Anurag Gupta we developed a model to predict the performance of homogeneous Time Warp systems where all LPs behave similarly, as would be the case in, for example, many queueing network simulations (Gupta et al. 1991). Development of a model for something as complex as Time Warp required some approximations to be made to make the mathematics tractable, so we validated that the model gave accurate predictions by comparing results produced by the model with measurements of GTW.

We later expanded this collaboration to include Dick Serfozo, another expert in stochastic models and a Georgia Tech professor in the School of Industrial and Systems Engineering. Our interest here was in understanding Time Warp performance when one limited the amount of memory allocated to the parallel simulation. At the time, there was much interest in developing techniques to execute Time Warp in situations with limited memory. David Jefferson had recently published his work on the cancelback algorithm that enabled Time Warp to execute, albeit slowly, within a constant factor of the amount of memory needed for a sequential execution. We wanted to examine how Time Warp performance would change as additional memory was provided. One of Dick's PhD students, Liang Chen, did the heavy lifting in developing an analytic model for Time Warp with limited memory (Akyildiz et al. 1993). We found that Time Warp performance increases rapidly as more memory is added, then hits a knee where diminishing returns set in, and subsequent additional memory provide modest or no benefit. In fact, if too much memory is provided, performance can actually decline as Time Warp becomes overly optimistic, and rolls back more computation than desired. Like our earlier work, we validated the models by comparing performance predictions made by the model with experimental results of the cancelback algorithm which Samir Das added to GTW.

While the analytic modeling work provided a window into Time Warp performance, it really only analyzed Time Warp performance rather than improving it. Our subsequent efforts focused on putting this work to good use to design efficient Time Warp systems. Specifically, this work highlighted the fact that memory could be used as a way to "throttle" the computation to avoid overly optimistic execution (Das and Fujimoto 1997a). We realized that by monitoring the execution of the Time Warp program and limiting the amount of memory that was provided we could adaptively control the execution of Time Warp to maximize its performance. Specifically, the goal was to keep Time Warp operating near the knee of the performance–memory curve so that Time Warp reaped the performance benefits of

additional memory, but did not utilize additional memory as that provided only marginal benefit. It also prevented the computation from moving into overly optimistic modes of execution, something that could occur by giving it too much memory. Samir Das designed, implemented, and validated this approach by showing that his algorithm could dynamically control the amount of memory provided to move the execution to the knee of the performance–memory curve (Das and Fujimoto 1997b). While other researchers had proposed other mechanisms to prevent overly optimistic execution, Samir's work was distinguished by having analytic modeling work as a basis for understanding the throttling mechanism. Further, this work provided a way to automatically adapt the execution to continually optimize performance throughout the parallel execution.

Our interest in monitoring the Time Warp execution and development of adaptive mechanisms to optimize its behavior led to an exploration of load balancing techniques. Here, we were particularly interested in the "background" execution of Time Warp on a distributed computing platform that was shared with other computations. We realized that execution of Time Warp on shared platforms presented a challenging test case because LPs that had to compete with many other computations to get CPU cycles would advance in simulation time slowly compared to LPs that had fewer other competitors for CPU cycles. This would cause the latter to race ahead, leading to much rollback and an inefficient execution. PhD student Chris Carothers developed a dynamic load distribution algorithm for GTW that would control the Time Warp execution in such circumstances. He implemented the algorithm and showed that it would yield efficient execution of Time Warp programs in the presence of other computations (Carothers and Fujimoto 2000).

Independent of our research, the concept of executing codes on shared remote servers became increasingly popular in industry, and is now referred to as cloud computing. We continued to have interest in this area. Our initial work was motivated by a somewhat different paradigm, that used in the Search for Extra-Terrestrial Intelligence (SETI) project that farmed out computations to remote servers, e.g., otherwise idle workstations. Alfred Park developed a novel execution approach based on farming out computations and later collecting their results (Park and Fujimoto 2007, 2012). While much of his early work focused on conservatively synchronized codes, he also examined the execution of Time Warp in grid computing environments (Park and Fujimoto 2008). Other subsequent work looked at the implementation of Time Warp in cloud computing environments (Malik et al. 2010).

7.18 The High-Level Architecture

In the U.S., the Department of Defense (DoD) was one of the principle organizations interested in developing modeling and simulation technologies. Beginning in the 1980's with a highly successful Defense Advanced Research Projects Agency

(DARPA) project called SIMNET (Miller and Thorpe 1995), the hot topic in DoD concerned how to reuse existing simulation models and get them to interoperate using distributed computing platforms. By the 1980s simulators to train pilots and equipment operators had become common, and were increasing in realism and sophistication with advances in computer graphics. Around that time local area networks were being invented, raising the possibility of interconnecting these simulators to create a kind of virtual battlefield with many simulated platforms to train military personnel. This was the forerunner to multiplayer video games that are common today. The SIMNET project demonstrated that this was a feasible concept. Throughout the 1990s there was a great emphasis, and investment, in creating interoperable distributed simulations. For example, the Distributed Interactive Simulations (DIS) (IEEE Std 1278.2-1995 1995) standards were developed to facilitate interoperability. While SIMNET and DIS focused on training simulations, the desire for interoperability spread to simulation models used for analysis, the area where Time Warp focused. An effort focusing on integrating wargame simulations called the Aggregate Level Simulation Protocol (ALSP) had begun with this objective (Wilson and Weatherly 1994).

In the mid-1990s an ambitious effort began to, in effect, combine the lessons learned in DIS and ALSP to create a common modeling and simulation architecture spanning the entire DoD. This effort, called the High-Level Architecture (HLA) was being led by Judith Dahmann of the Defense Modeling and Simulation Organization (DMSO). Just as HLA was getting underway, I received a phone call from Richard Weatherly of the MITRE Corporation. Richard had led the ALSP effort, and now was tasked with developing prototype implementations of the new HLA standard that was being developed. I had not met Weatherly before, but he had read some of my papers on parallel discrete event simulation, and asked if I was interested in getting involved in the HLA effort. I immediately accepted and became the technical lead of a working group tasked with defining the time management services of the standard that were largely concerned with synchronization issues in distributed simulations.

How does HLA relate to Time Warp? HLA was about getting separately developed simulations to work together in distributed computing environments. As such, a central objective of the time management services was to get simulations utilizing different mechanisms for time advancement to mesh together and interoperate. This meant getting time stepped and event-driven simulations to be able to synchronize and coordinate their time advances. With parallel discrete event simulation technologies becoming mature and finding some use in the DoD, e.g., through the ballistic missile defense program mentioned earlier, we extended this object to include *parallel* simulators, including both ones synchronized using conservative synchronization algorithms as well as optimistic ones such as Time Warp. Developing an approach to integrate all of these mechanisms together, while still ensuring that a reasonably efficient implementation could be realized was a significant challenge. Our task was to define the application program interface (API) that would be used by these different simulations, both parallel and sequential.

The main observation that enabled us to define an API for integrating these simulations was to realize that there was actually much commonality between conservative and optimistic approaches. This had been realized earlier, as described in work such as Chandy and Sherman's space-time simulation approach (Chandy and Sherman 1989). For example, the key quantity that conservative simulation algorithms needed to compute was a lower bound on the timestamp (LBTS) of messages that might be later received by an LP, because once this had been computed, the LP would know that all events with time stamp less than its LBTS value could be safely processed without fear of later receiving an event with smaller timestamp. LBTS is a close cousin of Global Virtual Time (GVT) used in Time Warp to determine a lower bound on the timestamp of any future rollback; rollbacks are also caused by receiving a message or anti-message in an LP's simulated past. Thus, if the time management services provided a means of computing LBTS and returning this information to each LP, or federate in HLA terminology, such a service would support both conservative and optimistic simulations.

A second observation was that Time Warp could be viewed as a conservative execution, but with the ability to optimistically process messages with time stamp larger than the LBTS value. Our approach in the HLA was to start with a conservatively synchronized distributed simulation, and add the mechanisms needed to allow optimistic processing. At the heart of the HLA's conservative synchronization approach were two mechanisms: one that guaranteed that messages would be delivered to a federate in timestamp order, and a second that enabled a federate to advance its simulation time in a way that guaranteed it would not receive any messages in its simulated past. To support Time Warp, a service was needed to also deliver optimistic messages to each federate, i.e., messages where it was possible a smaller timestamp message might later be received. This led to the *Flush Queue Request* service that when invoked, delivered all incoming messages to the federate invoking the service, regardless of its timestamp relative to LBTS or other events that had been previously delivered. Once these optimistic events were delivered to the federate, it was free to process them optimistically, risking the need for rollback. Should rollback be later required, this (specifically state-saving) was something the federate would have to implement on its own, so the API did not have to deal with this concern. Finally, the other part that was needed was a way to implement Time Warp anti-messages. Here, we leveraged our prior work in application-defined event cancellation (Lomow et al. 1991), discussed earlier. We realized that event retraction, i.e., an application unscheduling a previously scheduled event was a desirable application feature, independent of the underlying synchronization approach, be it optimistic or conservative. This led to the creation of the *Retract Message* service that could be used by conservative federates to unschedule events, as well as optimistic federates to implement Time Warp anti-messages. The services were defined so that the retraction of previously scheduled messages was transparent to conservative federates receiving the retracted message in that the original message would not be delivered to the federate until it could be guaranteed that it would not be later retracted. Of course, this guarantee could not be preserved for optimistic federates that were using the *Flush Queue* service. In this case, if the

message being retracted had already been delivered to the federate, the retraction request was simply treated by the receiving federate as an anti-message, and processed according to Time Warp event processing rules. In this way, the HLA time management services could support both conservative and optimistic simulations executing within the same federated distributed simulation. In the end, all this came together relatively smoothly. I worked out many of the technical details on planes between Atlanta and DC, a place free from phone calls, emails (this was well before inflight wifi), and visitors and students coming to my door.

Technical issues aside, a perhaps bigger challenge in the HLA effort was to build consensus among the various stakeholders who had their own ideas of how things should be done. Many folks in the DoD M&S community were not familiar with PDES technologies, and had never heard of CMB or Time Warp. Fortunately, I had some help here. A piece of Jade Simulations, a company founded by Brian Unger at the University of Calgary, was bought by Science Applications International Corporation (SAIC), a major defense contractor with a large stake in defense M&S in general, and HLA in particular. Two of Brian's former students, Darrin West and Larry Mellon, were leading much of the HLA effort in SAIC, and were very well versed in PDES. Together with a small team representing various constituencies we managed to get consensus on the specifications.

The High-Level Architecture was approved as the standard architecture for all M&S in the U.S. Department of Defense in 1996, and was subsequently standardized by IEEE (IEEE Std 1516.2-2000 2000) and later updated (IEEE Std 1516.1-2010 2010). Although the HLA "mandate" that originally required all M&S programs in the U.S. Department of Defense to become compliant with HLA was later rescinded, I was pleased to see HLA come up over the years in many different application areas other than defense. In retrospect, I have the highest regard for many of the individuals involved in the HLA effort. Richard Weatherly and his team at MITRE proved to be very capable developers. I never envied their task of developing an implementation of the standard while the standard itself was being changed! Richard was open to new ideas, and willing to incorporate changes that seemed well motivated. There were substantial differences and plenty of heated arguments during the HLA development. Many of these conflicts came to Judith Dahmann. I also did not envy her task. I found Judy to be both personable as well as being a strong and capable leader. She deserves enormous credit in seeing the HLA effort through to completion. I will also remember Judith as having a subtle sense of humor. The HLA effort had periodic meetings of the Architecture Management Group (AMG) at the DMSO headquarters in Alexandria Virginia, that included perhaps a hundred or so key stakeholders across the DoD M&S community. At one meeting, just before Christmas, Judith delivered a present to each attendee—a screw, with, for good measure, an attached bolt. Her only comment at the meeting was that the interpretation was left entirely up to each of us!

7.19 Pivoting to Federated Simulations

Subsequent to the adoption and standardization of the HLA by IEEE, my research program made a deliberate shift toward the federated simulation approach used in HLA. One of the challenges back then in the PDES community that persists even today concerned creation of PDES codes. Despite substantial efforts, application development still requires a certain amount of sophistication in parallel computing in addition to model expertise. The HLA demonstrated a pathway to take sequential simulation code that had never been developed for parallel or distributed execution, and transform it into a form suitable for parallel execution. Even in the pre-HLA days I saw that developers in the DoD were creating distributed simulations by, in effect, federating a code with itself to create a distributed version. This was in fact easier than federating different simulations because many interoperability issues associated with interconnecting different codes such as use of different model abstractions and representations disappeared. Thus, this seemed to be a practical approach to easily creating parallel discrete event simulations.

In order to get self-federation to work, one needed high performance runtime infrastructure software to implement synchronization (time management) and communication services. While my work in the previous decade focused on the Georgia Tech Time Warp software that served as a platform for our research in PDES, I decided to make a deliberate shift, to focus on HLA-like federated simulations. I spent the 1997–98 academic year on leave from Georgia Tech at the Defense Evaluation Research Agency in Malvern, England, with the intent of developing a new software base to support future research in federated simulation systems. The target platform used in this work was a set of workstations interconnected by a fast, Myrinet switch. Using communications software called fast-messages developed at the University of Illinois, in a few weeks I developed the first version of the RTI-Kit software and demonstrated it using a simplified version of the HLA API to implement communications and time management services (Fujimoto and Hoare 1998). In my initial work, I conducted a number of benchmarking experiments that demonstrated that RTI-Kit could achieve high performance, and was thus suitable for implementing parallel discrete event simulations.

Over the years that followed, RTI-Kit evolved into the Federated Distributed Simulations Kit (FDK) that became a central software base used by our research group over the next decade. We demonstrated that one could take existing sequential simulation codes and create high performance parallel versions suitable for execution on supercomputers. Not all simulations were suitable for parallelization using this approach. For example, we found that some simulations making extensive use of global data structures that were accessed throughout the entire code could not be easily parallelized. However, well-structured codes could be parallelized.

Our most successful illustration of this approach was to create a high performance, parallel version of the NS2 communication network simulator (Fall 1999).

The original NS2 was developed with no consideration of parallel implementation. In an effort by PhD student George Riley whom I co-supervised with networking research Mostafa Ammar, we created a parallel version dubbed PDNS (parallel/distributed NS) (Riley et al. 2004). Through a DARPA-funded project focused on large scale network simulation we demonstrated the largest available discrete event network simulations of the day, executing on more than a thousand processors of a supercomputer at the Pittsburgh Supercomputing Center (Fujimoto et al. 2003).

Much of our work in federated parallel simulations focused on conservative synchronization techniques, because existing sequential codes were not coded to include rollback, and adding rollback mechanisms was not straightforward. However, the federated approach is applicable to Time Warp simulations, or codes that were designed to include rollback. We demonstrated the use of HLA time management services to create Time Warp simulations with a prototype that a student named Steve Ferenci developed (Ferenci et al. 2000).

7.20 In Conclusion: The Future

More than 35 years since Time Warp was invented research continues both in its application and the technology itself. The goal of a general purpose parallel discrete event simulation engine that application developers can readily use without knowledge of parallel processing techniques continues to be elusive and the technology has not fulfilled its potential to see widespread adoption in industry. However, modern research in Time Warp systems is being driven more by the changing world and new technology developments rather than classical PDES problems defined decades ago.

One direction for research in Time Warp is driven from below: new computing platforms on which Time Warp simulations execute. Cloud computing, massively parallel supercomputers, graphical processing units (GPUs), and mobile computing platforms all present new challenges for running Time Warp simulations. For example, cloud computing platforms raise challenges due to the shared nature of the platform, and substantial communication delays, as noted earlier. Massively parallel machines raise questions concerning scalability, especially for real-world applications that are often highly irregular and contain inherent bottlenecks. GPUs, and more broadly heterogeneous computing platforms utilize SIMD, data parallel modes of execution that are very different from the platforms on which Time Warp was originally invented. Mobile computing platforms where data-driven simulations are used to monitor and manage operational systems again present new unexplored challenges for Time Warp. These challenges are discussed in greater detail in Fujimoto (2016).

Another research challenge faced by data centers, supercomputers, and mobile computing platforms concerns the amount of power consumed by the simulation. For mobile computing platforms energy consumption affects battery life, so it is an

area of great concern. In supercomputers, power consumption has become a major issue, limiting the performance of supercomputing nodes, and incurring substantial costs in operating data centers. Little is known concerning the power and energy consumption properties of Time Warp, and distributed simulations in general. This represents another important line of inquiry for future research in Time Warp.

More than three decades after its creation, Time Warp continues to be an active area of study and investigation in the parallel and distributed simulation community. While much progress has been made, the world has changed, raising new research challenges and opportunities that did not exist when it was invented. While much has been learned concerning Time Warp and its application, it will likely remain at the center of much parallel and distributed simulation research into the foreseeable future.

References

Akyildiz IF, Chen L, Das SR, Fujimoto RM, Serfozo R (1993) The effect of memory capacity on time warp performance. J Parallel Distrib Comput 18(4):411–422

Bagrodia R, Chandy KM, Liao WT (1991) A unifying framework for distributed simulation. ACM Trans Model Comput Simul 1(4):348–385

Barnes Jr PD, Carothers C, Jefferson D, LaPre J (2013) Warp speed: executing time warp on 1,966,080 cores. In: ACM SIGSIM principles of advanced discrete simulation (PADS), Montreal, Quebec, Canada, May 2013

Balzer RM (1969) EXDAMS: extendable debugging and monitoring system. In: Proceedings AFIPS'69 spring joint computer conference, pp 567–580, May 14–16, 1969

Berry O, Jefferson D (1985) Critical path analysis of distributed simulation. In: Society for Computer Simulation Conference on Distributed Simulation, San Diego, January 24–26, 1985

Bryant R (1977) Simulation of packet communication architecture computer systems. MS thesis, MIT/LCS/TR-188, November 1977

Buzzell CA, Fujimoto RM, Robb MJ (1990) Modular VME rollback hardware for time warp. In: SCS distributed simulation conference, pp 153–156

Carothers CD, Bauer D, Pearce S (2002) ROSS: A high-performance, low-memory, modular time warp system. J Parallel Distrib Comput 62(11):1648–1669

Carothers CD, Fujimoto RM (2000) Efficient execution of time warp programs on heterogeneous, NOW platforms. IEEE Trans Parallel Distrib Syst **11**(3):299–317

Carothers CD, Perumalla K, Fujimoto RM (1999) Efficient optimistic parallel simulation using reverse computation. ACM Trans Model Comput Simul 9(3):224–253

Chandy KM, Misra J (1979) Distributed simulation: a case study in design and verification of distributed programs. IEEE Trans Softw Eng SE-5(5)

Chandy KM, Misra J (1981) Asynchronous distributed simulation via a sequence of parallel computations. In: CACM, vol 24, no 11, April 1981

Chandy KM, Sherman R (1989) Space, time, and simulation. Proceedings of the SCS multiconference on distributed simulation, SCS simulation series 21:53–57

Das S, Fujimoto RM, Panesar K, Allison D, Hybinette M (1994) GTW: a time warp system for shared memory multiprocessors. In: Proceedings of the 1994 winter simulation conference, pp 1332–1339

Das SR, Fujimoto RM (1997a) adaptive memory management and optimism control in time warp. ACM Trans Model Comput Simul 7(2):239–271

Das SR, Fujimoto RM (1997b) An empirical evaluation of performance-memory tradeoffs in time warp. IEEE Trans Parallel Distrib Syst 8(2):210–224
Fall K (1999) Network emulation in the VINT/NS simulator. In: Proceedings IEEE international symposium on computers and communications, pp 244—250
Ferenci S, Fujimoto R, Ammar MH, Perumalla K (2002) Updateable simulation of communication networks. In: 16th workshop on parallel and distributed simulation
Ferenci S, Perumalla K, Fujimoto R (2000) An approach to federating parallel simulators. In: 14th workshop on parallel and distributed simulation
Fujimoto R (2000) Parallel and distributed simulation systems. Wiley
Fujimoto R, Hunter M, Sirichoke J, Palekar M, Kim H-K, Suh W (2007) Ad hoc distributed simulations. Principles of advanced and distributed simulation. IEEE, San Diego, CA, pp 15–24
Fujimoto RM (1988) Performance measurements of distributed simulation strategies. Distrib Simul SCS. 10:14–20
Fujimoto RM (1989a) Time warp on a shared memory multiprocessor. Trans Soc Comput Simul 6 (3):211–239
Fujimoto RM (1989) The virtual time machine. In: International symposium on parallel algorithms and architectures, pp 199–208
Fujimoto RM (1990) Performance of time warp under synthetic workloads. Proc SCS Multiconf Distrib Simul 22:23–28
Fujimoto RM (2016) Research challenges in parallel and distributed simulation. ACM Trans Model Comput Simul 24(4)
Fujimoto RM, Hoare P (1998) HLA RTI performance in high speed LAN environments. In: Proceedings of the fall simulation interoperability workshop, Orlando, FL
Fujimoto RM, Hybinette M (1997) Computing global virtual time in shared memory multiprocessors. ACM Trans Model Comput Simul 7(4):425–446
Fujimoto RM, Perumalla KS, Park A, Wu H, Ammar M, Riley GF (2003) Large-scale network simulation—how big? How fast? Modeling, analysis and simulation of computer and telecommunication systems
Fujimoto RM, Tsai JJ, Gopalakrishnan GC (1992) Design and evaluation of the rollback chip: special purpose hardware for time warp. IEEE Trans Comput 41(1):68–82
Fujimoto RM (2000) Parallel and distributed simulation systems. Wiley (2000)
Gafni A (1985) Space management and cancellation mechanisms for time warp. PhD dissertation, Department of Computer Science, University of Southern California, TR-85–341, December 1985
Gupta A, Akyildiz IF, Fujimoto RM (1991) Performance analysis of time warp with multiple homogeneous processors. IEEE Trans Softw Eng 17(10):1013–1027
Huang Y-L, Alexopoulos C, Hunter M, Fujimoto RM (2012) Ad hoc distributed simulation methodology for open queueing networks. Trans Soc Model Simul Int 88(7)
Hybinette M, Fujimoto RM (2001) Cloning parallel simulations. ACM Trans Model Comput Simul 11(2):378–407
IEEE Std 1278.2-1995 (1995) IEEE standard for distributed interactive simulation—communication services and profiles. Institute of Electrical and Electronics Engineer Inc, New York, NY
IEEE Std 1516.1-2010 (2010) IEEE standard for modeling and simulation (M&S) high level architecture (HLA)—interface specification. Institute of Electrical and Electronic Engineers Inc, New York, NY
IEEE Std 1516.2-2000 (2000) IEEE standard for modeling and simulation (M&S) high level architecture (HLA)—object model template (OMT) specification. Institute of Electrical and Electronic Engineers Inc, New York, NY
Jefferson D, Sowizral H (1982) Fast concurrent simulation using the time warp method, Part I: Local control. RAND note N-1906-AF, The RAND corporation, Santa Monica, California, December 1982

Jefferson D, Witkowski A (1984) An approach to performance analysis of timestamp-oriented synchronization mechanisms. In: acm symposium on the principles of distributed computing, Vancouver, BC, August 1984

Jefferson D, Beckman B, Hughes S, Levy E, Litwin T, Spagnuolo J, Vavrus J, Wieland F, Zimmerman B (1985) Implementation of time warp on the caltech hypercube. Society for Computer Simulation Conference on Distributed Simulation, San Diego, January 24–26, 1985

Jefferson D (1985) Virtual time. ACM Trans Program Lang Syst (TOPLAS) 7(3):404–425

Jefferson D, Motro A (1986) The time warp mechanism for database concurrency control. In: Proceedings of the IEEE 2nd international conference on data engineering, Los Angeles, CA, February 4–6 1986

Jefferson D, Beckman B, Wieland F, Blume L, DiLoreto M, Hontalas P, Laroche P, Sturdevant K, Tupman J, Warren V, Wedel J, Younger H, Bellenot S (1987) Distributed simulation and the time warp operating system. In: 11th symposium on operating systems principles (SOSP), Austin, TX, November 1987

Jefferson D (1990) Virtual time II: the cancelback protocol for storage management in time warp. In: Proceedings of the ACM symposium on the principles of distributed computing, Quebec City, Quebec, August 1990

Jefferson D, Reiher P (1991) Supercritical speedup. In: ANSS'91 proceedings of the 24th annual symposium on simulation, 1991

Jones DW (1986) An empirical comparison of priority-queue and event-set implementations. Commun ACM 29(4):300–311

Kung HT, Robinson JT (1981) On optimistic methods for concurrency control. ACM Trans Database Syst 6(2):213–226 (1981)

Lomow G, Das SR, Fujimoto RM (1991) Mechanisms for user invoked retraction of events in time warp. ACM Trans Model Comput Simul 1(3):219–243

Malik AW, Park AJ, Fujimoto RM (2010) An optimistic parallel simulation protocol for cloud computing environments. SCS Model Simul Mag 1(4). Society for Modeling and Simulation International

Miller DC, Thorpe JA (1995) SIMNET: the advent of simulator networking. Proc IEEE 83 (8):1114–1123

Mitre Corp (1997) DPAT: detailed policy assessment tool, brochure, Center for Advanced Aviation System Development (CAASD)

Nicol D, Liu X (1997) The dark side of risk (what your mother never told you about time warp). In: Proceedings of 11th workshop on parallel and distributed simulation (PADS), Lockenhaus, Austria, June 1997

Park A, Fujimoto RM (2007) A scalable framework for parallel discrete event simulations on desktop grids. In: 8th IEEE/ACM International Conference On Grid Computing

Park A, Fujimoto RM (2008) Optimistic parallel simulation over public resource-computing infrastructures and desktop grids. In: Workshop on distributed simulations and real-time applications

Park A, Fujimoto RM (2012) Efficient master/worker parallel discrete event simulation on metacomputing systems. IEEE Trans Parallel Distrib Syst 23(5)

Perumalla KS (2005) μsik-a micro-kernel for parallel/distributed simulation systems. In: Workshop on principles of advanced and distributed simulation. IEEE, pp 59–68

Ravi TM, Jefferson D (1988) Message routing to migrating processes. In: Proceedings of the 1988 international conference on parallel processing (ICPP), August 15–18, 1988

Reiher P, Jefferson D (1990) Virtual time-based dynamic load management in the time warp operating system. In: Distributed simulation, Nicol D, Fujimoto R (eds), Simulation series, vol 22, no 2. Society for Computer Simulation, San Diego, January 1990

Riley G, Ammar M, Fujimoto RM, Park A, Perumalla K, Xu D (2004) A federated approach to distributed network simulation. ACM Trans Model Comput Simul 14(1):116–148

Schordan M, Jefferson D, Barnes Jr P, Oppelstrup T, Quinlan D (2015) Reverse code generation for parallel discrete event simulation. In: 7th conference on reversible computation, Grenoble, France, July 16–17, 2015

Schordan M, Oppelstrup T, Jefferson D, Barnes P, Quinlan D (2016) Automatic generation of reversible C++ code and its performance in a scalable kinetic Monte-Carlo application. In: SIGSIM PADS (Principles of advanced discrete simulation), Banff, Alberta, Canada, May 16–18, 2016

Tang Y, Perumalla KS, Fujimoto RM, Karimabadi H, Driscoll J, Omelchenko Y (2006) Optimistic simulations of physical systems using reverse computation. Simul: Trans Soc Model Simul Int 82(1):61–73

Vulov G, Hou C, Vuduc R, Quinlan D, Fujimoto RM, Jefferson D (2011) The backstroke framework for source level reverse computation applied to parallel discrete event simulation. In: Winter simulation conference

Wieland F (2001) Practical parallel simulation applied to aviation modeling. In: 15th workshop on parallel and distributed simulation, Lake Arrowhead, CA

Wieland F, Hawley L, Feinberg A, DiLoreto M, Blume L, Reiher P, Beckman B, Hontalas P, Bellenot S, Jefferson D (1989) Distributed combat simulation and time warp: the model and its performance. In: Distributed simulation, Unger B, Fujimoto R (eds) Simulation series, vol 21, no 2. Society for Computer Simulation, San Diego, 1989

Wilson AL, Weatherly RM (1994) The aggregate level simulation protocol: an evolving system. In: Proceedings of the 1994 winter simulation conference, pp 781–787

Chapter 8
Design and Analysis of Simulation Experiments: Tutorial

Jack P.C. Kleijnen

Abstract This tutorial reviews the design and analysis of simulation experiments. These experiments may have various goals: validation, prediction, sensitivity analysis, optimization (possibly robust), and risk or uncertainty analysis. These goals may be realized through metamodels. Two types of metamodels are the focus of this tutorial: (i) low-order polynomial regression, and (ii) Kriging (or Gaussian processes). The type of metamodel guides the design of the experiment; this design fixes the input combinations of the simulation model. However, before a regression or Kriging metamodel is applied, the many inputs of the underlying realistic simulation model should be screened; the tutorial focuses on sequential bifurcation. Optimization of the simulated system may use either a sequence of low-order polynomials—known as response surface methodology—or Kriging models fitted through sequential designs. Finally, "robust" optimization should account for uncertainty in simulation inputs. The tutorial includes references to earlier Winter Simulation Conference papers.

8.1 Introduction

Wagner (1975)'s famous textbook on *operations research/management science* (OR/MS) resorted to simulation "when all else fails" (p. 903) as the "method of last resort" (p. 907); nowadays, however, simulation is treated as a serious scientific method. Unfortunately, practitioners often treat their experiments with simulation models (or "simulators" or "computer codes") very unprofessionally. Nevertheless, the simulation community is making progress; see the *Winter Simulation Conference* (WSC) panel on the need for reproducible experiments in Uhrmacher et al. (2016), and also see the WSC paper on a survey of methodological aspects of simulation in Timm and Lorig (2015).

J.P. Kleijnen (✉)
Department of Management, TiSEM, Tilburg University, Room K 1113, 90153, 5000 Tilburg, Netherlands
e-mail: Kleijnen@TilburgUniversity.edu
URL: https://sites.google.com/site/kleijnenjackpc/

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_8

In this tutorial we focus on the statistical *design and analysis of simulation experiments* (DASE). We further focus on *random* (or stochastic) simulation, because the WSCs focus on this type of simulation; nevertheless, we shall occasionally discuss deterministic simulation. By definition, random simulation gives random output (response); examples are discrete-event dynamic systems, agent-based simulation, and stochastic differential equations (see Govindan 2016). Mathematically, these simulators give random outputs because they use *pseudorandom numbers* (PRNs) as an input.

We assume that a specific simulation model is given, albeit that this model may be changed before results of the DASE are implemented. To support the managers' decisions, the simulation analysts try different values for the inputs and the parameters of this simulation model. We claim that such numerical (computer) experiments may have one or more of the following *goals*: (i) verification and validation (V & V) of the simulation model; (ii) prediction of the response of the simulated system; (iii) sensitivity analysis (SA)—either global or local—of the simulation model; (iv) optimization of the simulated real system (SimOpt); (v) risk analysis (RA) of the simulated real system. We now discuss these goals (more details including references are given in Kleijnen (2015, p. 9); in this tutorial we introduce goal ii).

Sub (i): V & V may use many methods; see Sargent et al. (2016). We focus on the use of metamodels for V & V: do the estimated effects of the metamodel have signs that agree with prior expert-knowledge about the corresponding real system; two case studies are detailed in Kleijnen (1995) and Van Ham et al. (1992). In *calibration* we estimate the combination of simulation parameters that gives the simulation output that is closest to the observed real-world output, so calibration uses optimization; see Liu et al. (2017).

Sub (ii): Prediction is necessary if experimentation with the real system is too expensive, dangerous, etc. Obviously, we can use metamodels to predict the output for an input combination that has not yet been simulated. Such prediction may be necessary if simulation takes too long; an example is real-time, online decision-making.

Sub (iii): SA may be performed through "experimentation" with the simulation model; i.e., simulate different "values" or "levels" for the model's parameters, input values, and starting values of the inputs (we use the terminology in Zeigler et al. (2000); i.e., we must infer the value of a parameter, whereas we can directly observe the value of an input). For example, in a queueing simulation we may start with an "empty" (no waiting customers) simulated system, exponential interarrival times with a fixed arrival rate, and a fixed number of servers. In DASE we speak of *factors*, which have fixed values during one "run" of the simulation experiment.

When we experiment, we use a *metamodel*; i.e., an explicit and relatively simple approximation of the *input/output* (I/O) function implicitly defined by the underlying simulation model. We treat the simulation model as a *black box*; i.e., we observe only the I/O of the simulation model. Actually, simulation analysts may not realize that they are using a metamodel. For example, if we change only one factor at a time, then we assume that the factors do not interact. When we consider only two levels for each of our (say) k factors, then a 2^k design implies that we assume a metamodel

with interactions among these k factors—but without "pure" higher order effects such as purely quadratic or cubic effects. In this tutorial we shall discuss metamodels in much detail.

Sub (iv) and (v): We may apply RA for (a) a single combination of the simulation inputs, (b) a given set of combinations, (c) the best combination estimated through SimOpt. Alternative (b) is described in Kleijnen (2015, p. 221) using a first-order polynomial metamodel to estimate which combinations of uncertain inputs form the frontier that separates acceptable and unacceptable outputs. Alternative (c) is detailed in Kleijnen (2015, p. 275). However, we may also try to estimate the optimum while accounting for uncertainty in the parameters of the simulation; we call this approach *robust optimization* (RO). We distinguish two RO approaches; namely, the Taguchi approach and the mathematical programming (MP) approach. Taguchi has been popular in mechanical engineering, since several decades. In MP, RO is a recent important topic. We expect that both approaches will inspire OR/MS. For example, Taguchi inspired Dellino et al. (2009) and Dellino et al. (2010). An example of recent RO in MP is Bertsimas and Mišić (2017). A recent example of RO combining Taguchi and MP is Yanikoğlu et al. (2016). Approaches related to RO are also discussed in Song and Nelson (2017) and Zhou (2017).

The *goals of metamodels* immediately follow from the five goals of the underlying simulation models. Because Kleijnen (2015, p.9) does not discuss goal 2, we point out that a simulation may be computationally *slow* or *expensive*: i.e., it may take much computer time to obtain the response for a given combination of the simulation inputs. An example is the deterministic simulator of a car-crash model at Ford that required 36–160 h of computer time; see Simpson et al. (2004). Another example is the discrete-event simulation of a *rare* event with dramatic consequences (e.g., a nuclear accident); without adequate importance sampling the simulation needs very many replications. Obviously, a metamodel can quickly predict the output for a new input combination of the expensive simulation model. A case study including Kriging and neural-network metamodels within a decision support system for real-time failure assessment of a power-grid network (so this study is RA) is detailed in Rosen et al. (2015).

There are important differences between "traditional" *design of experiments* (DOE) and DASE. Kleijnen (2015, p. 12) mentions that DOE was developed for real-world (nonsimulated) experiments in agriculture, engineering, psychology, etc. In these experiments it is impractical to investigate "many" factors; i.e., ten factors seems a maximum. Moreover, it is hard to investigate factors with more than "a few" levels; five levels per factor seems the limit. In simulation, however, these restrictions do not apply; i.e., simulation models may have thousands of factors—each with many values. Consequently, a multitude of factor combinations may be simulated. Moreover, simulation is well-suited to "sequential" designs instead of "one shot" designs, because simulation experiments run on computers that typically produce output sequentially (apart from parallel computers; see Gramacy 2015), whereas agricultural experiments run during a single growing season. Altogether, many simulation analysts need a change of mindset.

Mathematically, we consider $w = f_{\text{sim}}(\mathbf{z}, \mathbf{r})$ where w denotes the simulation output (e.g., average waiting time), f_{sim} the I/O function of the simulation model, $\mathbf{z}$ the vector with the values of the k simulation inputs, and $\mathbf{r}$ the vector with PRNs used by the random simulation model; in deterministic simulation $\mathbf{r}$ vanishes. Usually, $\mathbf{z}$ is *standardized* (scaled) such that the resulting input $\mathbf{d}$ has elements either $-1 \leq d_j \leq 1$ or $0 \leq d_j \leq 1$ with $j = 1, \ldots, k$.

Note: An input of a queuing simulation may be the traffic rate—not the arrival rate $1/\mu_a$ and the service rate $1/\mu_s$ where μ denotes the mean (expected value)—so $z_1 = \mu_s/\mu_a$. Another input may be the priority rule (queueing discipline), which is a qualitative input with at least two levels; e.g., first-in-first-out and smallest-service-time-first. A qualitative input with more than two values needs special care; see Kleijnen (2015, pp. 69–71).

The function $w = f_{\text{sim}}(\mathbf{z}, \mathbf{r})$ may be approximated by $y = f_{\text{meta}}(\mathbf{x}) + e$ where y is the metamodel output, which may deviate from w so the approximation error e may have $\mu_e \neq 0$. If $\mu_e = 0$, then we call f_{meta} *adequate* (or valid). There are many types of metamodels; see Kleijnen (2015, p. 10). We, however, focus on two types: *low-order polynomials*, which are linear regression models, and *Kriging* or Gaussian process (GP) models.

Note: Mitchell and Morris (1992) introduced Kriging into the WSC community. Later on, Van Beers and Kleijnen (2004) surveyed Kriging for simulation. Biles et al. (2007) used Kriging in constrained simulation optimization. Kleijnen et al. (2012) introduced convex and monotonic bootstrapped Kriging.

We base this tutorial on Kleijnen (2017), which is a recent review based on Kleijnen (2015). We update that review; e.g., we add Binois et al. (2016) in Sect. 8.5. Furthermore, we emphasize publications in the WSC proceedings. Kleijnen (2015) includes hundreds of additional references, and many website addresses for software.

We organize this chapter as follows. Section 8.2 summarizes classic linear regression and DOE. Section 8.3 presents solutions for DASE if the classic statistical assumptions are violated in practice. Section 8.4 summarizes *sequential bifurcation* (SB) for the screening of a multitude of factors. Section 8.5 summarizes Kriging and its designs. Section 8.6 summarizes SimOpt using either low-order polynomials or Kriging, including RO.

8.2 Basic Linear Regression and Designs

Because we assume that the readers are familiar with the basics of linear regression, we limit our discussion of this regression to definitions of our mathematical symbols and terminology. We define the *linear regression* (meta)model $\mathbf{y} = \mathbf{X}_N\boldsymbol{\beta} + \mathbf{e}$ where $\mathbf{y}$ denotes the N-dimensional vector with the dependent (explained) variable (also see Sect. 8.1), N denotes the number of observations on the simulation response with $N = \sum_{i=1}^{n} m_i$, n the number of simulated input combinations, m_i the number of replications for combination i (obviously, deterministic simulation implies $m_i = 1$ so $N = n$), $\mathbf{X}_N$ is the $N \times q$ matrix of independent (explanatory)

regression variables—obviously, $\mathbf{X}_N$ has m_i identical rows, whereas $\mathbf{X}_n$ denotes the corresponding $n \times q$ matrix without any identical rows determined by the $n \times k$ design matrix $\mathbf{D}$ and the type of regression model (e.g., second-order polynomial), $\boldsymbol{\beta}$ denotes the q-dimensional vector with regression parameters (coefficients), and $\mathbf{e}$ denotes the N-dimensional vector with the residuals $\mathrm{E}(\mathbf{y}) - \mathrm{E}(\mathbf{w})$ where $\mathbf{w}$ denotes the N-dimensional vector with independent simulation outputs; this independence requires that random simulation does not use *common random numbers* (CRN). We define the q-dimensional row vector $\mathbf{x}_i = (x_{i;1}, \ldots, x_{i;q})$. Obviously, n denotes the number of *different* combinations of the k simulation inputs. We focus on a special type of linear regression; namely, a *second-order polynomial* with k simulation inputs, which has an intercept β_0, k first-order effects β_j $(j = 1, \ldots, k)$, $k(k-1)/2$ two-factor interactions (cross-products) $\beta_{j;j'}(j' > j)$, and k purely quadratic effects $\beta_{j;j}$. These *interactions* mean that the effect of an input depends on the values of one or more other inputs. A *purely quadratic* effect means that the marginal effect of the input is not constant, but diminishes or increases. This second-order polynomial is nonlinear in $\mathbf{x}$, and linear in β. We assume that interactions among three or more inputs are unimportant, because such interactions are hard to interpret and in practice are indeed often unimportant. Of course, we should check this assumption; i.e., we should "validate" the estimated metamodel (see the next section).

The *ordinary least squares* (OLS) estimator of $\boldsymbol{\beta}$ is $\hat{\beta} = (\mathbf{X}'_N\mathbf{X}_N)^{-1}\mathbf{X}'_N\mathbf{w}$, assuming the inverse of $\mathbf{X}'_N\mathbf{X}_N$ exists; e.g., $\hat{\beta}$ exists if $\mathbf{X}_N$ is *orthogonal*. If m_i is a positive integer constant (say) m, then we may replace $\mathbf{w}$ by $\overline{\mathbf{w}}$ with the n elements $\overline{w}_i = \sum_{r=1}^{m} w_{i;r}/m$ and replace $\mathbf{X}_N$ by $\mathbf{X}_n$. Moreover, $\hat{\beta}$ is the *maximum likelihood estimator* (MLE) if $\mathbf{e}$ is *white noise*, which means that $\mathbf{e}$ is *normally, independently, and identically distributed* (NIID) with zero mean and constant variance σ_e^2; i.e., $\mathbf{e} \sim \mathrm{N}_N(\mathbf{0}_N, \sigma_e^2\mathbf{I}_{N\times N})$ where N_N stands for N-variate (see subscript) normally (symbol: N) distributed, $\mathbf{0}_N$ for the N-variate vector with zeroes, and $\mathbf{I}_{N\times N}$ for the $N \times N$ identity matrix. If the metamodel is valid, then $\sigma_e^2 = \sigma_w^2$. White noise implies that $\hat{\beta}$ has the $q \times q$ covariance matrix $\boldsymbol{\Sigma}_{\hat{\beta}} = (\mathbf{X}'_N\mathbf{X}_N)^{-1}\sigma_w^2$. Because σ_w^2 is unknown, we estimate $\sigma_w^2 = \sigma_e^2$ through the *mean squared residuals* (MSR) $(\hat{\mathbf{y}} - \mathbf{w})'(\hat{\mathbf{y}} - \mathbf{w})/(N - q)$ with predictor $\hat{\mathbf{y}} = \mathbf{X}_N\hat{\beta}'$ and degrees of freedom (DOF) $N - q > 0$; this inequality is satisfied, even if $n = q$ but $m_i > 1$ for at least one value of i. This MSR gives the estimator $\widehat{\boldsymbol{\Sigma}}_{\hat{\beta}}$. This $\widehat{\boldsymbol{\Sigma}}_{\hat{\beta}}$ has a main diagonal with the elements $s^2(\widehat{\beta}_g)$ $(g = 1, \ldots, q)$, which give the square roots $s(\widehat{\beta}_g)$. *Confidence intervals* (CIs) and *tests* for the individual $\widehat{\beta}_g$ follow from the Student t-statistic with $N - q$ DOF: $t_{N-q} = (\widehat{\beta}_g - \beta_g)/s(\widehat{\beta}_g)$. Finally, $\hat{\mathbf{y}} = \mathbf{X}_N\hat{\beta}'$ implies $s^2(\hat{y}|\mathbf{x}_i) = \mathbf{x}'_i\widehat{\boldsymbol{\Sigma}}_{\hat{\beta}}\mathbf{x}_i$. This $s^2(\hat{y}|\mathbf{x}_i)$ is minimal at the center of the experimental area. So, if the goal of the metamodel is to predict the simulation output at a specific "new" point, then we should simulate N "old" points close to that new point (a similar guideline applies for Kriging; see Gramacy 2015).

We saw that $\boldsymbol{\Sigma}_{\hat{\beta}} = (\mathbf{X}'_N\mathbf{X}_N)^{-1}\sigma_w^2$, so we may select $\mathbf{X}_N$ such that we "optimize" $\boldsymbol{\Sigma}_{\hat{\beta}}$. Obviously, $\mathbf{X}_N$ is determined by $\mathbf{X}_n$, m_i, and the type of regression model (e.g., $\mathbf{X}_N$ may include x_j^2). DOE does not say much about the selection of m_i; typically, DOE assumes $m_i = 1$. If $m_i = m \geq 1$, then an orthogonal $\mathbf{X}_n$ implies an orthogonal

$\mathbf{X}_N$ (we may specify $\mathbf{X}_N$ as $\mathbf{X}_n$ "stapled" or "stacked" m times). We shall further discuss m_i in Sect. 8.3.3. In the next subsections we shall discuss the following special types of second-order polynomial regression model: (i) $\beta_{j;j'} = 0$ with $j \leq j'$; (ii) $\beta_{j;j} = 0$; we discuss (a) $\widehat{\beta}_j$ unbiased by $\beta_{j;j'}$ with $j < j'$, and (b) $\widehat{\beta}_j$ and $\widehat{\beta}_{j;j'}$ unbiased. These types of polynomials require designs of different *resolution* (abbreviated to R). Obviously, $\mathbf{X}_n$ is determined by $\mathbf{D}$. To select a specific $\mathbf{D}$ with $\mathbf{z}$ constrained over a given experimental area, we try to minimize $\text{Var}(\widehat{\beta}_g)$ $(g = 1, \ldots, q)$; other criteria are discussed in Kleijnen (2015, pp. 66–67). We can prove that this minimization requires an orthogonal $\mathbf{X}_N$, which gives $\mathbf{\Sigma}_{\hat{\beta}} = (N\mathbf{I})^{-1}\sigma_w^2 = \mathbf{I}\sigma_w^2/N$. This $\mathbf{\Sigma}_{\hat{\beta}}$ implies that the $\widehat{\beta}_j$ are statistically independent. Moreover, these $\widehat{\beta}_j$ have the same variance (σ_w^2/N). So we can *rank* the explanatory variables in order of importance, using either $\widehat{\beta}_g$ or t_{N-q} with $\beta_g = 0$ (so $t_{N-q} = \widehat{\beta}_g/s(\widehat{\beta}_g)$); obviously, we do not hypothesize that the intercept β_0 is zero. Usually, DOE assumes that $z_{i;j}$ is standardized such that $-1 \leq d_{i;j} \leq 1$.

8.2.1 R-III Designs for First-Order Polynomials

If a first-order polynomial is an adequate metamodel (so $\beta_{j;j'} = 0$ with $j \leq j'$), then *changing one factor at a time* gives unbiased $\widehat{\beta}_j$ $(j = 1, \ldots, k)$. However, these $\widehat{\beta}_j$ do not have minimum variances; a R-III design does minimize these variances because this design gives an orthogonal $\mathbf{X}_N$, as we shall see in this subsection. Furthermore, we can prove that $\text{Var}(\widehat{\beta}_j)$ is minimal if we simulate only two levels per factor, as far apart as the experimental area allows (in either a one-factor-at-a-time or a R-III design); see Kleijnen (2015, pp. 44–49).

Note: If all $\widehat{\beta}_g$ are independent, then the *full* regression model with q effects and the *reduced* model with nonsignificant effects eliminated have *identical* values for those estimated effects that occur in both models. If not all $\widehat{\beta}_g$ are independent, then so-called *backwards elimination* of nonsignificant effects changes the values of the remaining estimates.

R-III designs are also known as *Plackett-Burman* (PB) designs. A subclass are *fractional factorial two-level* designs, denoted by 2_{III}^{k-p} with integer p such that $0 \leq p < k$ and $2^{k-p} \geq 1 + k$: we first discuss 2_{III}^{k-p} designs. Any 2^{k-p} design is *balanced*; i.e., each input is simulated $n/2$ times at its lower value L_j and at its higher value H_j. Furthermore, a 2^{k-p} design gives an *orthogonal* $\mathbf{X}_n$. A R-III design may be *saturated*: $N = q$ (with $N = \sum_{i=1}^{n} m_i$, $n = 2^{k-p}$, and $q = 1 + k$). A saturated design implies that the MSR is undefined (because $N - q = 0$). To solve this problem, we may obtain replications for one or more combinations of the R-III design; e.g., the combination at the *center* of the experiment where $d_j = 0$ if d_j is quantitative and d_j is randomly selected as -1 or 1 if d_j is qualitative with two levels. A simple algorithm for constructing 2_{III}^{k-p} designs is given in Kleijnen (2015, pp. 53–54).

Obviously, 2^{k-p}_{III} designs have n equal to a power of 2. However, there are also PB designs with n a multiple of 4; e.g., $8 \leq k \leq 11$ implies $n = 12$ (whereas 2^{12-8}_{III} implies $n = 16$). If $8 \leq k < 11$, then we do not use $n - (k + 1)$ columns of $\mathbf{D}$. Actually, there are PB designs for $12 \leq n \leq 96$; for $12 \leq n \leq 36$ these designs are tabulated in Montgomery (2009, p. 326) and Myers et al. (2009, pp. 165). A PB design is balanced and gives an orthogonal $\mathbf{X}_n$.

8.2.2 R-IV Designs for First-Order Polynomials

A R-IV design gives $\widehat{\beta}_j$ unbiased by $\beta_{j;j'}$ with $j < j'$ (so $q = 1 + k + k(k-1)2$). Such a design uses the *foldover theorem*; i.e., augmenting a R-III design $\mathbf{D}$ with its *mirror* design $-\mathbf{D}$ gives a R-IV design (we shall also apply the foldover trick in Sect. 8.4 on screening). A R-IV design does not enable unbiased estimators of the $k(k-1)2$ *individual* two-factor interactions. For example, a 2^{7-4}_{III} design has $n_{III} = 8$ so $n_{IV} = 16$; furthermore, $q = 29$ so $n_{IV} < q$ and we cannot compute the OLS estimators of the 29 individual regression parameters.

8.2.3 R-V Designs for Two-Factor Interactions

A R-V design enables unbiased $\widehat{\beta}_j$ and $\widehat{\beta}_{j;j'}$ with $j < j'$. Obviously, we have $q = 1 + k + k(k-1)/2$. The DOE literature gives tables for generating 2^{k-p}_{V} designs. Unfortunately, these designs are not saturated at all; e.g., the 2^{8-2}_{V} design implies $n = 64 \gg q = 37$. We point out that *Rechtschaffner* designs do include saturated R-V designs; see Kleijnen (2015, pp. 62–63). We shall use R-V designs in the next subsection.

8.2.4 CCDs for Second-Degree Polynomials

A CCD or *central composite design* enables unbiased $\widehat{\beta}_j$ and $\widehat{\beta}_{j;j'}$ with $j \leq j'$. A CCD consists of three subdesigns: (i) a R-V design; (ii) the *central* combination $\mathbf{0}'_k$; (iii) the $2k$ *axial* combinations—which form a *star design*—with $d_j = c$ and $d_{j'} = 0$ where $j' \neq j$, and $d_j = -c$ and $d_{j'} = 0$. Obviously, $c \neq 1$ implies five values per input, whereas $c = 1$ implies three values per input. The usual choice of c is not 1. The "optimal" choice of c assumes white noise, which does not hold in practice so we do not detail this choice. Finally, if $c \leq 1$, then $-1 \leq d_{i;j} \leq 1$; else $-c \leq d_{i;j} \leq c$.

A CCD gives a *non-orthogonal* $\mathbf{X}_n$; e.g., any two columns corresponding with β_0, $\beta_{j;j}$, and $\beta_{j';j'}$ are not orthogonal. A CCD is rather inefficient (i.e., $n \gg q$); yet, CCDs are popular in DOE, especially in *response surface methodology* (RSM)

(see Sect. 8.6.1). For further discussion of CCDs and other types of designs for second-degree polynomials we refer to Kleijnen (2015, pp. 64–66), and Myers et al. (2009, pp. 296–317). A more efficient modified CCD is derived in Kleijnen and Shi (2017).

8.3 Classic Assumptions Versus Simulation Practice

The *classic* assumptions stipulate a single type of output (univariate output) and white noise. In simulation practice, however, the simulation model often has a *multivariate* output and no white noise—as we discuss now.

8.3.1 Multivariate Simulation Output

We assume that for r-variate ($r \geq 1$) simulation output we use r univariate linear regression metamodels $\mathbf{y}^{(l)} = \mathbf{X}_N^{(l)} \beta^{(l)} + \mathbf{e}^{(l)}$ with $l = 1, \ldots r$ where $\mathbf{y}^{(l)}$ corresponds with output type l; $\mathbf{X}_N^{(l)}$ is the $N \times q_l$ matrix for metamodel l; $\beta^{(l)}$ is the vector with the q_l regression parameters for metamodel l; and $\mathbf{e}^{(l)}$ is the N-dimensional vector with the residuals of metamodel l. More restrictively we assume that all r metamodels are polynomials of the same order (e.g., second-order), so $\mathbf{X}^{(l)} = \mathbf{X}$ and $q_l = q$. Obviously, $\mathbf{e}^{(l)}$ has variances that may vary with l (e.g., the variances differ for simulated inventory costs and service percentages), and $e_i^{(l)}$ and $e_i^{(l')}$ are not independent (they are different transformations of the same PRNs). Nevertheless, $\mathbf{X}^{(l)} = \mathbf{X}$ implies that the *best linear unbiased estimator* (BLUE) of $\beta^{(l)}$ is the OLS estimator computed per output: $\widehat{\beta}^{(l)} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}^{(l)}$. Furthermore, CIs and tests for the elements in $\widehat{\beta}^{(l)}$ use the classic formulas in the preceding section. We are not aware of any *general* designs for multivariate output. For further discussion of multivariate output we refer to Kleijnen (2015, pp. 85–88).

8.3.2 Nonnormal Simulation Output

The normality assumption often holds *asymptotically*; i.e., if the simulation run is long, then the sample average of the autocorrelated observations is nearly normal. Estimated quantiles, however, may be very nonnormal, especially in case of an "extreme" (e.g., 99%) quantile. The t-statistic (used in the CIs) is quite insensitive to nonnormality. Whether the actual simulation run is long enough to make the normality assumption hold, is always hard to know. Therefore it seems good practice to test whether the simulation output has a Gaussian *probability density function* (PDF). We may then use various *residual plots* and *goodness-of-fit statistics*; e.g.,

the chi-square statistic. We must then assume that the tested outputs are IID. We may therefore obtain "many" (say, 100) replications for a specific input combination (e.g., the base scenario). However, if the simulation is expensive, then these plots are too rough and these tests have no power.

Actually, the white-noise assumption concerns e, not w. Given $m_i \geq 1$ replications ($i = 1, \ldots, n$), we obtain $\overline{w}_i = \sum_{r=1}^{m_i} w_{i;r}/m_i$ and the corresponding $\widehat{\overline{e}}_i = \widehat{y}_i - \overline{w}_i$. For simplicity of presentation, we assume that m_i is a constant m. If $w_{i;r}$ has a constant variance σ_w^2, then $\overline{w}_i$ also has a constant variance; namely, $\sigma_{\overline{w}}^2 = \sigma_w^2/m$. Unfortunately, even if $\overline{w}_i$ has a constant variance $\sigma_{\overline{w}}^2$ and is independent of $\overline{w}_{i'}$ with $i \neq i'$ (no CRN), then $\boldsymbol{\Sigma}_{\widehat{\overline{\mathbf{e}}}} = [\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\sigma_{\overline{w}}^2$ so $\widehat{\overline{\mathbf{e}}}$ does not have IID components; so, the interpretation of the popular plot with estimated residuals is not straightforward.

We may apply *normalizing transformations*; e.g., $\log(w)$ may be more normally distributed than w. Unfortunately, the metamodel now explains the behavior of the transformed output—not the original output; also see Kleijnen (2015, p. 93).

A statistical method that does not assume normality is *distribution-free bootstrapping* or *nonparametric bootstrapping*. This bootstrapping may be used to examine (i) nonnormal distributions, or (ii) nonstandard statistics (e.g., R^2). We denote the *original observations* by w, and the *bootstrapped observations* by w^*. We assume that these w are IID; indeed, $w_{i;1}, \ldots, w_{i;m_i}$ are IID because the m_i replications use nonoverlapping PRN streams. We *resample—with replacement*—these m_i observations such that the original sample size m_i remains unchanged. We apply this resampling to each combination i. The resulting $w_{i;1}^*, \ldots, w_{i;m_i}^*$ give the average $\overline{w}_i^*$, which give the n-dimensional vector $\overline{\mathbf{w}}^*$. For simplicity's sake, we now assume $m_i = m > 1$, so the bootstrapped OLS estimator of β is $\widehat{\beta}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\overline{\mathbf{w}}^*$. To reduce sampling error, we select a *bootstrap sample size* (say) B, and repeat this resampling B times; e.g., B is 100 or 1,000. This B gives $\widehat{\beta}_b^*$ with $b = 1, \ldots, B$. For simplicity's sake, we focus on β_q (last element of β). To compute a two-sided $(1 - \alpha)$ CI, the *percentile method* computes the $\alpha/2$ quantile (or percentile) of the *empirical density function* (EDF) of $\widehat{\beta}_q^*$ obtained through sorting the B observations on $\widehat{\beta}_{q;b}^*$. This sorting gives the *order statistics*, denoted by the subscript $(\cdot)$ where—for notational simplicity—we assume that $B\alpha/2$ is integer so the estimated $\alpha/2$ quantile is $\widehat{\beta}_{q;(B\alpha/2)}^*$. Analogously we obtain $\widehat{\beta}_{q;(B[1-\alpha/2])}^*$. These two quantiles give a two-sided asymmetric $(1 - \alpha)$ CI: $\widehat{\beta}_{q;(B\alpha/2)}^* < \beta_q < \widehat{\beta}_{q;(B[1-\alpha/2])}^*$. We shall mention more bootstrap examples, in later sections.

8.3.3 Heterogeneous Variances of Simulation Outputs

In practice, $\text{Var}(w_i)$ changes as $\mathbf{x}_i$ changes ($i = 1, \ldots, n$). In some applications, however, we may hope that this variance heterogeneity is negligible. Unfortunately, $\text{Var}(w_i)$ is unknown so we must estimate it. The classic unbiased estimator is

$s^2(w_i) = \sum_{r=1}^{m_i}(w_{i;r} - \overline{w}_i)^2/(m_i - 1)$. This $s^2(w_i)$ itself has a high variance. To compare the n estimators $s^2(w_i)$, we can apply many tests; see Kleijnen (2015, p. 101).

If we either assume or find variance heterogeneity, then we may still use *OLS*. Actually, $\hat{\beta}$ is still *unbiased*, but $\mathbf{\Sigma}_{\hat{\beta}}$ becomes $(\mathbf{X}_n'\mathbf{X}_n)^{-1}\mathbf{X}_n'\mathbf{\Sigma}_{\overline{\mathbf{w}}}\mathbf{X}_n(\mathbf{X}_n'\mathbf{X}_n)^{-1}$ where for simplicity's sake we assume $m_i = m$ so $\mathbf{\Sigma}_{\overline{\mathbf{w}}}$ is the $n \times n$ diagonal matrix with the main-diagonal elements $\text{Var}(w_i)/m$.

The DOE literature ignores designs for heterogeneous output variances. We propose classic designs with m_i such that we obtain approximately constant $s^2(w_i)/m_i$ $(i = 1, \ldots, n)$. Initially we take a *pilot sample* of size $m_0 \geq 2$ for each combination, which gives (say) $s_i^2(m_0)$. Next we select a number of additional replications $\widehat{m}_i - m_0$ with

$$\widehat{m}_i = m_0 \times nint\left[\frac{s_i^2(m_0)}{min_i s_i^2(m_0)}\right] \tag{8.1}$$

where $nint[x]$ denotes the integer closest to x. Combining the $\widehat{m}_i$ replications of the two stages gives $\overline{w}_i$ and $s^2(w_i)$. This $\overline{w}_i$ gives $\widehat{\beta}$, while $s^2(w_i)$ gives the diagonal matrix $\widehat{\mathbf{\Sigma}}_{\overline{\mathbf{w}}}$ with main-diagonal elements $s_i^2(\widehat{m}_i)/\widehat{m}_i$. This $\widehat{\mathbf{\Sigma}}_{\overline{\mathbf{w}}}$ gives $\widehat{\mathbf{\Sigma}}_{\hat{\beta}}$, which—together with t_{m_0-1}—gives **a** CI for $\widehat{\beta}_j$.

Actually, (8.1) gives the *relative* number of replications $\widehat{m}_i/\widehat{m}_{i'}$. To select absolute numbers, we recommend the rule in Law (2015, p. 505) with number of replications $\widehat{m}$ and a relative estimation error r_{ee}:

$$\widehat{m} = min\left[r \geq m : \frac{t_{r-1;1-\alpha/2}\sqrt{s_i^2(m)/r}}{|\overline{w}(m)|} \leq \frac{r_{ee}}{1 + r_{ee}}\right]. \tag{8.2}$$

8.3.4 Common Random Numbers

CRN are often applied; actually, CRN are the default in software for discrete-event simulation. Given $m_i = m$, we can arrange the simulation output $w_{i;r}$ with $i = 1, \ldots, n$ and $r = 1, \ldots, m$ into a matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)$ with $\mathbf{w}_r = (w_{1;r}, \ldots, w_{n;r})'$. Obviously, CRN create correlation between $w_{i;r}$ and $w_{i';r}$. Moreover, different replications use nonoverlapping PRN streams so $w_{i;r}$ and $w_{i;r'}$ with $r \neq r'$—or the n-diagonal vectors $\mathbf{w}_r$ and $\mathbf{w}_{r'}$—are independent. CRN are used to reduce $\text{Var}(\widehat{\beta}_g)$ and $\text{Var}(\widehat{y})$; unfortunately, CRN increase the variance of the estimated intercept. For details on the effective usage of CRN we refer to Law (2015, pp. 592–604).

To compute $\widehat{\beta}$, we do not use $\mathbf{W}$; i.e., we use the vector $\mathbf{w}$ with $N = \sum_{i=1}^{n} m_i$ elements. To compute $\widehat{\mathbf{\Sigma}}_{\hat{\beta}}$ in case of CRN, we use the non-diagonal matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{w}}$. Unfortunately, this $\widehat{\mathbf{\Sigma}}_{\mathbf{w}}$ is *singular* if $m \leq n$; if $m > n$, then we may compute CIs for $\widehat{\beta}_j$ from t_{m-1}. An alternative method requires only $m > 1$, and computes $\hat{\beta}_r = (\mathbf{X}_n'\mathbf{X}_n)^{-1}\mathbf{X}_n'\mathbf{w}_r$

$(r = 1, \ldots, m)$. We again focus on a single element of this $\hat{\beta}_r$; namely, element g $(g = 1, \ldots, q)$. Obviously, $\widehat{\beta}_{g;r}$ and $\widehat{\beta}_{g;r'}$ with $r \neq r'$ and $r' = 1, \ldots, m$ are IID with variance $\text{Var}(\widehat{\beta}_g)$. The m replications give $\overline{\widehat{\beta}}_g = \sum_{r=1}^{m} \widehat{\beta}_{g;r}/m$ and $s^2(\overline{\widehat{\beta}}_g) = \sum_{r=1}^{m} (\widehat{\beta}_{g;r} - \overline{\widehat{\beta}}_g)^2/[m(m-1)]$; together they give $t_{m-1} = (\overline{\widehat{\beta}}_g - \beta_g)/s(\overline{\widehat{\beta}}_g)$.

Unfortunately, we cannot apply this alternative when estimating a *quantile* instead of a mean. We then recommend distribution-free bootstrapping; see Kleijnen (2015, p. 99, 110). Furthermore, m_i is not a constant if we select m_i such that $\overline{w}_i$ has the same—absolute or relative—width of the CI around $\overline{w}_i$; see again (8.2). We must then adjust the analysis; see Kleijnen (2015, p. 112).

8.3.5 *Validation of Metamodels*

We discuss various validation methods (which we may also use to compare first-order against second-order polynomials, or linear regression against Kriging metamodels). One method uses $R^2 = \sum_{i=1}^{n} (\widehat{y}_i - \overline{\overline{w}})^2 / \sum_{i=1}^{n} (\overline{w}_i - \overline{\overline{w}})^2 = 1 - \sum_{i=1}^{n} (\widehat{y}_i - \overline{w}_i)^2 / \sum_{i=1}^{n} (\overline{w}_i - \overline{\overline{w}})^2$ where $\overline{\overline{w}} = \sum_{i=1}^{n} \overline{w}_i/n$ and $m_i \geq 1$. If $n = q$ (saturated design), then $R^2 = 1$—even if $\text{E}(\widehat{e}_i) \neq 0$. If $n > q$ and q increases, then R^2 increases—whatever the size of $|\text{E}(\widehat{e}_i)|$ is; because of possible *overfitting*, we may therefore use the *adjusted* R^2: $R^2_{\text{adj}} = 1 - (1 - R^2)(n-1)/(n-q)$. Unfortunately, we do not know *critical values* for R^2 or R^2_{adj}. We might either use subjective lower thresholds, or estimate the distributions of these two statistics through distribution-free bootstrapping; see Kleijnen (2015, p. 114).

Actually, we prefer *cross-validation* over R^2 or R^2_{adj}. We again suppose that $m_i = m \geq 1$ so we may replace $\mathbf{w}$ by $\overline{\mathbf{w}}$ in OLS. Now we delete I/O combination i to obtain $(\mathbf{X}_{-i}, \overline{\mathbf{w}}_{-i})$ where we suppress the subscript n of $\mathbf{X}$. Next we compute $\widehat{\beta}_{-i} = (\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}\mathbf{X}'_{-i}\overline{\mathbf{w}}_{-i}$ $(i = 1, \ldots, n)$. This gives $\widehat{y}_{-i} = \mathbf{x}'_i\widehat{\beta}_{-i}$. We may "eyeball" the *scatterplot* with $(\overline{w}_i, \widehat{y}_{-i})$, and decide whether the metamodel is valid. Alternatively, we may compute the *Studentized prediction error*

$$t^{(i)}_{m-1} = \frac{\overline{w}_i - \widehat{y}_{-i}}{\sqrt{s^2(\overline{w}_i) + s^2(\widehat{y}_{-i})}} \tag{8.3}$$

where $s^2(\overline{w}_i) = s^2(w_i)/m$ and $s^2(\widehat{y}_{-i}) = \mathbf{x}'_i\widehat{\boldsymbol{\Sigma}}_{\widehat{\beta}_{-i}}\mathbf{x}_i$ with $\widehat{\boldsymbol{\Sigma}}_{\widehat{\beta}_{-i}} = s^2(\overline{w}_i)(\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}$. We reject the metamodel if $max_i|t^{(i)}_{m-1}| > t_{m-1;1-[\alpha/(2n)]}$ where we use the *Bonferroni inequality* so we replace $\alpha/2$ by $\alpha/(2n)$; i.e., we control the *experimentwise* or *familywise* type-I error rate α. Note that regression software uses a shortcut to avoid the n recomputations in cross-validation.

Cross-validation affects not only $\widehat{y}_{-i}$, but also $\widehat{\beta}_{-i}$ (see above). We may be interested not only in the predictive performance of the metamodel, but also in its *explanatory* performance; i.e., do the n estimates $\widehat{\beta}_{-i}$ remain stable?

Related to cross-validation are *diagnostic* statistics; e.g., the *prediction sum of squares* (PRESS): $[\sum_{i=1}^{n}(\widehat{y}_{-i} - w_i)^2/n]^{1/2}$. We may apply bootstrapping to estimate the distribution of the various validation statistics; see Kleijnen (2015, p. 120).

If the validation suggests am unacceptable fitting error e, then we may consider various *transformations*. For example, we may replace y and x_j by $\log(y)$ and $\log(x_j)$ $(j = 1, \ldots, k)$ so that the first-order polynomial approximates relative changes through k *elasticity coefficients*. If we assume that f_{sim} is monotonic, then we may replace w and x_j by their ranks: *rank regression*. In the preceding subsections, we also considered transformations that make w better satisfy the assumptions of normality and variance homogeneity; unfortunately, different objectifies of a transformation may conflict with each other.

In Sect. 8.2 we discussed designs for low-order polynomials. If such a design does not give a valid metamodel, then we do not recommend routinely adding higher order terms: these terms are hard to interpret. However, if the goal is better prediction, then we may add higher order terms; e.g., a 2^k design enables the estimation of the interactions among three or more inputs. However, adding more terms may lead to overfitting; see our comment on R^2_{adj}. A detailed example of fitting a polynomial in simulation optimization is presented in Barton (1997, Fig. 4). Adding more explanatory variables is called *stepwise regression*, whereas eliminating nonsignificant variables is called *backwards elimination*, which we briefly discussed in the Note in Sect. 8.2.1.

8.4 Factor Screening: Sequential Bifurcation

Factor screening—or briefly screening—searches for the really important inputs (or factors) among the many inputs that can be varied in a (simulation) experiment. It is realistic to assume *sparsity*, which means that only a few inputs among these many inputs are really important; the *Pareto* principle or 20–80 rule states that only a few inputs—say, 20%—are really important. Two examples of screening 281 and 92 inputs respectively show only 15 and 11 inputs to be really important; see Kleijnen (2015, p. 136). In practice, we should apply screening *before* we apply the designs and analyses of the preceding two sections; however, to explain screening methods, we need those two sections.

Four types of screening designs that use different mathematical assumptions, are summarized in Kleijnen (2015, pp. 137–139). We, however, focus on SB, because SB is very efficient and effective if its assumptions are satisfied. SB is sequential; i.e., SB selects the next input combination after analyzing the preceding I/O data. SB is customized; i.e., SB accounts for the specific simulation model. (Frazier et al. 2012 considers SB from a Bayesian dynamic programming perspective.)

8.4.1 Deterministic Simulation and First-Order Polynomials

To explain the basic idea of SB, we assume a deterministic simulation and a first-order polynomial metamodel. Moreover, we replace the standardized x_j by the original simulation input z_j ($j = 1, \ldots, k$), and β_j by (say) γ_j. Furthermore, we assume that the *sign* of γ_j is known; some reflection shows that we can then define the low and high bounds L_j and H_j of z_j such that $\gamma_j \geq 0$. Consequently we can rank the inputs such that the most important input has the largest γ_j. We call input j *important* if $\gamma_j > c_\gamma$ where $c_\gamma \geq 0$ is a threshold specified by the users.

In *step 1* of SB we aggregate the k inputs into *a single group*, and check whether that group is important. We therefore obtain the simulation outputs $w(\mathbf{z} = \mathbf{L})$ with $L = (L_1, \ldots, L_k)'$; we also obtain $w(\mathbf{z} = \mathbf{H})$ with $\mathbf{H} = (H_1, \ldots, H_k)'$. If $\gamma_j = 0$ held for all j, then our assumptions would imply $w(\mathbf{z} = \mathbf{L}) = w(\mathbf{z} = \mathbf{H})$. In practice, however, $\gamma_j > 0$ for one or more j, so $w(\mathbf{z} = \mathbf{L}) < w(\mathbf{z} = \mathbf{H})$. (If $0 \leq \gamma_j < c_\gamma$ for all j, then $w(\mathbf{z} = \mathbf{H}) - w(\mathbf{z} = \mathbf{L}) > c_\gamma$ may happen; this "false importance" will be discovered in the following steps, as we shall see.)

In *step 2* we *split* the group with k inputs into two subgroups: *bifurcation*. Let k_1 and k_2 denote the size of subgroup 1 and 2 respectively (obviously, $k_1 + k_2 = k$). Let $w_{[k_1]}$ (subscript within []) denote w when $z_1 = H_1, \ldots, z_{k_1} = H_{k_1}$ and $z_{k_1+1} = L_1, \ldots, z_k = L_k$. In this step we obtain $w_{[k_1]}$, and compare this $w_{[k_1]}$ with $w_{[0]} = w(\mathbf{z} = \mathbf{L})$ of step 1: if (but not "if and only if") $w_{[k_1]} - w_{[0]} < c_\gamma$, then none of the individual inputs within subgroup 1 is important and we *eliminate* this subgroup from further experimentation. We also compare $w_{[k_1]}$ with $w_{[k]} = w(\mathbf{z} = \mathbf{H})$ of step 1; $w_k - w_{k_1} > c_\gamma$ if at least one input is important and this input is a member of subgroup 2.

In the following steps we continue splitting important subgroups into (smaller) subgroups, and eliminate unimportant subgroups. In some steps we may find that both subgroups are important; we then experiment with two important subgroups *in parallel*. Finally we identify and estimate all *individual* inputs that are not in eliminated (unimportant) subgroups.

Obviously, $\gamma_j \geq 0$ ensures that first-order effects do not cancel each other within a group. In practice the users often do know the *signs* of γ_j; e.g., if some inputs are service speeds, then higher speeds decrease the mean waiting time. If in practice it is hard to specify the signs of a few specific inputs, then we should treat these inputs *individually* through one of the classic designs discussed in Sect. 8.2 (e.g., a R-IV design); this is safer than applying SB and assuming a negligible probability of cancelation within a subgroup.

The *efficiency* of SB—measured by the number of simulated input combinations—improves if we can label the individual inputs such that they are placed in *increasing* order of importance; this labeling implies that the important inputs are *clustered* within the same subgroup. The efficiency further improves when we place (e.g.) all service inputs in the same subgroup. Anyhow, subgroups of *equal* size are not necessarily optimal. Academic and practical examples of SB are given in Kleijnen (2015, pp. 136–172).

8.4.2 Random Simulation and Second-Order Polynomials

Now we assume a second-order polynomial metamodel. Moreover, we assume that if input j has $\gamma_j = 0$, then all its second-order effects $\gamma_{j;j}$ ($j \leq j'$) are also zero: so-called *heredity* assumption. SB then gives $\widehat{\gamma}_j$ unbiased by $\gamma_{j;j'}$—provided we simulate the *mirror inputs* of the original inputs in the sequential design; see again the *foldover* principle in Sect. 8.2.2. So let $w_{[-j]}$ denote the mirror output with $z_1 = L_1, \ldots, z_j = L_j$ and $z_{j+1} = H_{j+1}, \ldots, z_k = H_k$. We standardize z such that x is either -1 or 1 with first-order effect β (Sect. 8.4.1 uses z instead of x). Kleijnen (2015, p. 149) shows that an unbiased estimator of $\beta_{j'-j} = \sum_{h=j'}^{j} \beta_h$ (the endash in $j' - j$ denotes "through") is

$$\widehat{\beta}_{j'-j} = \frac{(w_{[j]} - w_{[-j]}) - (w_{[j'-1]} - w_{[-(j'-1)]})}{4}. \tag{8.4}$$

First we assume a *fixed* number of replications, m. Let $w_{[j;r]}$ with $r = 1, \ldots, m$ denote observation r on $w_{[j]}$. Using (8.4), we then obtain $\widehat{\beta}_{(j'-j);r}$, which gives the classic estimators of the mean and the variance of $\widehat{\beta}_{(j'-j)}$. These estimated mean and variance give t_{m-1}. SB uses this t_{m-1} in a one-sided test, because $\gamma_j > 0$ so $\beta_j > 0$.

Next we assume a *random* m. Whereas the preceding t-test assumes a favorite null hypothesis and a fixed m, the *sequential probability ratio test* (SPRT) in Wan et al. (2010) considers two comparable hypotheses and a random m in each step. This SPRT controls the type-I error probability through the whole procedure and holds the type-II error probability at each step. The users must select thresholds $c_{\gamma \mathrm{U}}$ and $c_{\gamma \mathrm{I}}$ such that $\beta_j \leq c_{\gamma \mathrm{U}}$ is *unimportant* and $\beta_j \geq c_{\gamma \mathrm{I}}$ is *important*; when $c_{\gamma \mathrm{U}} < \beta_j < c_{\gamma \mathrm{I}}$, the power is "reasonable". Statistical details of this SPRT and a Monte Carlo experiment are given in Kleijnen (2015, p. 154–159). (The robustness of several SPRTs is investigated in Kleijnen and Shi 2017.)

In practice, simulation models have *multiple response types*; see again Sect. 8.3.1. We can extend SB to *multiresponse SB* (MSB). Details including extensive Monte Carlo experiments, a case study concerning a logistic system in China, and the validation of the MSB assumptions are given in Kleijnen (2015, pp. 159–172) and Shi and Kleijnen (2017).

8.5 Kriging Metamodels and Their Designs

Kriging metamodels are fitted to simulation I/O data obtained for *global* experimental areas, which are larger than the *local* areas in low-order polynomial metamodels. Because we assume that many readers are not familiar with the basics of Kriging, we detail various types of Kriging. We use the same symbols as above, unless Kriging traditionally uses different symbols.

8.5.1 Ordinary Kriging in Deterministic Simulation

Ordinary Kriging (OK) is popular and successful in practical deterministic simulation. OK assumes $y(\mathbf{x}) = \mu + M(\mathbf{x})$ where μ is the constant mean $\mathrm{E}[y(\mathbf{x})]$ and $M(\mathbf{x})$ is a zero-mean Gaussian stationary process, which has covariances that depend only on the distance between the input combinations $\mathbf{x}$ and $\mathbf{x}'$. We call $M(\mathbf{x})$ the *extrinsic noise* (to be distinguished from "intrinsic" noise in stochastic simulation; see Sect. 8.5.3). Let $\mathbf{X}$ denote the $n \times k$ matrix with the n "old" combinations $\mathbf{x}_i$ $(i = 1, \ldots, n)$, where the original simulation inputs $\mathbf{z}_i$ are standardized to obtain $\mathbf{x}_i$ (unlike DOE, Kriging does not use the symbol $\mathbf{D}$ for the design matrix). The *best linear unbiased predictor* (BLUP) for the *new* combination $\mathbf{x}_0$ is $\hat{y}(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i w_i = \boldsymbol{\lambda}'\mathbf{w}$. This unbiased $\hat{y}(\mathbf{x}_0)$ implies that if $\mathbf{x}_0 = \mathbf{x}_i$, then $\hat{y}(\mathbf{x}_0)$ is an *exact interpolator*: $\hat{y}(\mathbf{x}_i) = w(\mathbf{x}_i)$. The "best" $\hat{y}(\mathbf{x}_0)$ minimizes the *mean squared error* (MSE), which equals $\mathrm{Var}[\hat{y}(\mathbf{x}_0)]$; see (8.7) below. Altogether, the *optimal* $\boldsymbol{\lambda}$ is

$$\boldsymbol{\lambda}_o' = [\boldsymbol{\sigma}_M(\mathbf{x}_0) + \mathbf{1}_n \frac{1 - \mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\sigma}(\mathbf{x}_0)}{\mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \mathbf{1}_n}]' \boldsymbol{\Sigma}_M^{-1} \tag{8.5}$$

where $\mathbf{1}_n$ denotes the n-dimensional vector with ones, $\boldsymbol{\Sigma}_M = (\sigma_{i;i'}) = (\mathrm{Cov}(y_i, y_{i'}))$ $(i, i' = 1, \ldots, n)$ denotes the $n \times n$ matrix with the covariances between the metamodel's old outputs y_i, and $\boldsymbol{\sigma}_M(\mathbf{x}_0) = (\sigma_{0;i}) = (\mathrm{Cov}(y_0, y_i))$ denotes the n-dimensional vector with the covariances between the metamodel's new output y_0 and y_i. The components of λ_o decrease with the *distance* between $\mathbf{x}_0$ and $\mathbf{x}_i$ (so $\boldsymbol{\lambda}$ is not a constant vector, whereas $\boldsymbol{\beta}$ in linear regression is). Substitution of λ_o into $\hat{y}(\mathbf{x}_0) = \boldsymbol{\lambda}'\mathbf{w}$ gives

$$\hat{y}(\mathbf{x}_0) = \mu + \boldsymbol{\sigma}_M(\mathbf{x}_0)' \boldsymbol{\Sigma}_M^{-1} (\mathbf{w} - \mu \mathbf{1}_n). \tag{8.6}$$

Obviously, $\hat{y}(\mathbf{x}_0)$ varies with $\boldsymbol{\sigma}_M(\mathbf{x}_0)$, whereas μ, $\boldsymbol{\Sigma}_M$, and $\mathbf{w}$ remain fixed.

Note: The *gradient* $\nabla(\hat{y})$ follows from (8.6); see Lophaven et al. (2002, Eq. 2.18). Sometimes we can also compute $\widehat{\nabla}(w)$ and estimate a better OK model; see Kleijnen (2015, pp. 183–184) and Ulaganathan et al. (2014).

Instead of the symbol $\mathrm{Var}(y_i) = \sigma_{i;i} = \sigma_i^2 = \sigma^2$ we use the classic Kriging symbol τ^2 in

$$\mathrm{Var}[\hat{y}(\mathbf{x}_0)] = \tau^2 - \boldsymbol{\sigma}_M(\mathbf{x}_0)' \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\sigma}_M(\mathbf{x}_0) + \frac{[1 - \mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \boldsymbol{\sigma}_M(\mathbf{x}_0)]^2}{\mathbf{1}_n' \boldsymbol{\Sigma}_M^{-1} \mathbf{1}_n}. \tag{8.7}$$

This equation implies $\mathrm{Var}[\hat{y}(\mathbf{x}_0)] = 0$ if $\mathbf{x}_0 = \mathbf{x}_i$. Experimental results suggest that $\mathrm{Var}[\hat{y}(\mathbf{x}_0)]$ has local maxima at $\mathbf{x}_0$ approximately halfway between old input combinations (also see Sect. 8.6.2). Kriging gives bad extrapolations compared with interpolations (linear regression also gives minimal $\mathrm{Var}[\hat{y}(\mathbf{x}_0)]$ when $\mathbf{x}_0 = \mathbf{0}$).

Sometimes it is convenient to switch from covariances to correlations; the correlation matrix $\mathbf{R} = (\rho_{i;i'})$ equals $\tau^{-2}\boldsymbol{\Sigma}_M$ and $\rho(\mathbf{x}_0)$ equals $\tau^{-2}\boldsymbol{\sigma}_M(\mathbf{x}_0)$. There are several types of correlation functions; see Kleijnen (2015, pp.185–186). Most popular is the *Gaussian correlation function*:

$$\rho(\mathbf{h}) = \prod_{j=1}^{k} \exp\left(-\theta_j h_j^2\right) = \exp\left(-\sum_{j=1}^{k} \theta_j h_j^2\right) \tag{8.8}$$

with distance vector $\mathbf{h} = (h_j)$ where $h_j = \left|x_{g;j} - x_{g';j}\right|$ and $g, g' = 0, 1, \ldots, n$.

Obviously, we need to estimate the *Kriging (hyper)parameters* $\psi = (\mu, \tau^2, \theta')'$ with $\theta = (\theta_j)$. The most popular criterion is *maximum likelihood* (ML) (but OLS and cross-validation are also used). The computation of the *ML estimator* (MLE) $\widehat{\psi}$ is challenging, so different $\widehat{\psi}$ may result from different software packages or from initializing the same package with different starting values; see Erickson et al. (2016).

Plugging $\widehat{\psi}$ into (8.6) gives $\widehat{y}(\mathbf{x}_0, \widehat{\psi})$. Obviously, $\widehat{y}(\mathbf{x}_0, \widehat{\psi})$ is a *nonlinear* predictor. Furthermore, we *plug* $\widehat{\psi}$ into (8.7) to obtain $s^2[\widehat{y}(\mathbf{x}_0, \widehat{\psi})]$. To obtain a symmetric $(1 - \alpha)$ CI for $w(\mathbf{x}_0)$, we use $z_{\alpha/2}$ (standard symbol for the $\alpha/2$ quantile of N(0, 1)) and get $\widehat{y}(\mathbf{x}_0, \widehat{\psi}) \pm z_{\alpha/2} s[\widehat{y}(\mathbf{x}_0, \widehat{\psi})]$. There is much *software* for Kriging; see the many publications and websites in Kleijnen (2015, p. 190).

8.5.2 Designs for Deterministic Simulation

There is an abundant literature on various design types for Kriging in deterministic simulation; e.g., orthogonal array, uniform, maximum entropy, minimax, maximin, integrated mean squared prediction error, and "optimal" designs; see Kleijnen (2015, p. 198). However, the most popular *space filling* design uses *Latin hypercube sampling* (LHS). LHS assumes that the metamodel is more complicated than a low-order polynomial, but it does not assume a specific type of metamodel. LHS divides the range of each input into n mutually exclusive and exhaustive intervals of equal probability, and samples each input without replacement. Whereas DOE makes n increase with k (e.g., $n = 2^{k-p}$), LHS does not impose such a relationship. Nevertheless, if n is "small" and k is "large", then LHS covers the input space so sparsely that the fitted Kriging model may be inadequate. A well-known rule-of-thumb for LHS in SA through Kriging is $n = 10k$. LHS may be further refined, leading to maximin LHS, nearly orthogonal LHS, etc.; see Kleijnen (2015, p. 202).

Instead of LHS with its *fixed sample* or *one-shot* design, we may use a *sequential* design that is *customized* for the given simulation model; i.e., we learn about f_{sim} as we collect I/O data (also see Sect. 8.4 on SB). In this subsection we discuss SA, and in Sect. 8.6.2 we shall discuss SimOpt. Using an initial (relatively small) LHS design, we obtain simulation I/O data and fit a Kriging model. Then we consider—but not yet simulate—$\mathbf{X}_{cand}$ which denotes a larger design matrix with *candidate* combinations selected through LHS, and we find the candidate with the highest $s^2\{\widehat{y}(\mathbf{x}, \widehat{\psi})\}$ with $\mathbf{x} \in \mathbf{X}_{cand}$. Next we use the selected candidate point as the input to be simulated, which gives additional I/O data. Then we refit the Kriging model to the augmented I/O data; we may reestimate ψ. We stop if either the Kriging metamodel satisfies the goal of SA or the computer budget is exhausted. Experiments show that this

sequential design selects relatively few combinations in subareas with an approximately linear f_{sim}. Details are given in Kleijnen (2015, pp. 203–206).

8.5.3 Kriging in Random Simulation

Ankenman et al. (2010) develops *stochastic Kriging* (SK), adding the *intrinsic noise* term $\varepsilon_r(\mathbf{x}_i)$ for replication r $(r = 1, \ldots, m_i)$ at combination $\mathbf{x}_i$ $(i = 1, \ldots, n)$. After averaging over the m_i replications, SK uses the formulas for OK but replaces $\mathbf{w}$ by $\overline{\mathbf{w}}$ and $M(\mathbf{x}_i)$ by $M(\mathbf{x}_i) + \overline{\varepsilon}(\mathbf{x}_i)$ where $\overline{\varepsilon}(\mathbf{x}_i) \in N(0, \text{Var}[\varepsilon_r(\mathbf{x}_i)]/m_i)$ and $\overline{\varepsilon}(\mathbf{x}_i)$ is independent of $M(\mathbf{x})$. If we do not use CRN, then $\mathbf{\Sigma}_{\overline{\varepsilon}}$ is diagonal (if we used CRN and selected $m_i = m$, then $\mathbf{\Sigma}_{\overline{\varepsilon}}$ would equal $\mathbf{\Sigma}_{\varepsilon}/m$; however, we assume no CRN in this subsection). To estimate $\text{Var}[\varepsilon(\mathbf{x}_i)]$, SK may use either $s^2(w_i)$ or another Kriging model for $\text{Var}[\varepsilon(\mathbf{x}_i)]$—besides the Kriging model for the mean $\text{E}[y_r(\mathbf{x}_i)]$; see Kleijnen (2015, p. 208). We use $\psi_{+\varepsilon}$ to denote ψ augmented with $\text{Var}[\varepsilon_r(\mathbf{x}_i)]$ (extra parameters of random simulation). Note that SK for a quantile instead of an average is discussed in Chen and Kim (2013).

Binois et al. (2016) gives an alternative—called *hetGP*—for SK. This alternative assumes $m_i \geq 1$, whereas SK assumes $m_i \gg 1$. SK maximizes a likelihood for ψ, but not the full likelihood; i.e., SK does not give the MLEs of the intrinsic variances, but uses the moments $s^2(w_i)$. So, $\widehat{\psi}_{+\varepsilon}$ in SK is biased. The alternative uses the full likelihood for $\psi_{+\varepsilon}$; i.e., whereas SK fits the Kriging models for the mean and the intrinsic variances independently, hetGP couples these models through a joint likelihood for $\psi_{+\varepsilon}$ that is optimized in one shot. This alternative requires computational time of the same order as SK does.

8.5.4 Monotonic Kriging

In practice we sometimes know that f_{sim} is *monotonic*; also see Sect. 8.4 on screening. The Kriging predictor $\hat{y}$, however, may be *wiggling* if the sample size is small. To make $\hat{y}$ monotonic, we may apply *distribution-free bootstrapping* with *acceptance/rejection*. This method rejects the Kriging model fitted in bootstrap sample b $(b = 1, \ldots, B)$ if this metamodel gives wiggling predictions. We think that the resulting SA is better understood and accepted by the users. Monotonic Kriging may give a smaller MSE and a CI with higher coverage and acceptable length. For details we refer to Kleijnen et al. (2012) and Kleijnen (2015, pp. 212–216).

We may apply this method to preserve other I/O characteristics; e.g., waiting times and variances have only *positive* values. Furthermore, we may apply this method to other types of metamodels; e.g., regression models.

8.5.5 Global Sensitivity Analysis

So far we focused on $\widehat{y}$, but now we measure how sensitive $\widehat{y}$ and w are to the individual inputs x_j ($j = 1, \ldots, k$) and their interactions. We assume that $x = (x_j)$ has a prespecified distribution, as in LHS; see Sect. 8.5.2. We apply *functional analysis of variance* (FANOVA), using variance-based indexes (originally proposed by Sobol). FANOVA decomposes σ_w^2 into fractions that refer to sets of inputs; e.g., FANOVA may estimate that 70% of σ_w^2 is caused by x_1, 20% by x_2, and 10% by the interaction between x_1 and x_2. In general, σ_w^2 equals the following sum of 2^{k-1} components:

$$\sigma_w^2 = \sum_{j=1}^{k} \sigma_j^2 + \sum_{j<j'}^{k} \sigma_{j;j'}^2 + \cdots + \sigma_{1;\ldots;k}^2 \tag{8.9}$$

with the main-effect variance $\sigma_j^2 = \text{Var}[\text{E}(w|x_j)]$, the two-factor interaction variance $\sigma_{j;j'}^2 = \text{Var}[\text{E}(w|x_j, x_{j'})] - \text{Var}[\text{E}(w|x_j)] - \text{Var}[\text{E}(w|x_{j'})]$, etc. This σ_j^2 gives the *first-order sensitivity index* or the *main-effect index* $\zeta_j = \sigma_j^2/\sigma_w^2$, which quantifies the effect of varying x_j alone—averaged over the variations in all the other $k-1$ inputs—where the denominator σ_w^2 standardizes ζ_j to provide a fractional contribution. Altogether the sum of the 2^{k-1} indexes is 1. In practice, we assume that only the ζ_j—and possibly the $\zeta_{j;j'}$—are important, and that they sum up to a fraction "close enough" to 1. To *estimate* these measures, we may use LHS and replace the simulation model by a Kriging metamodel; see Kleijnen (2015, p. 218).

8.5.6 Risk Analysis or Uncertainty Analysis

In RA (or UA) we typically estimate the *failure probability* $P(w > c_\text{f})$ with a given threshold value c_f; see the many applications in nuclear engineering, finance, water management, etc. This $P(w > c_\text{f})$ may be very small—so $w > c_\text{f}$ is called a *rare event*—but this event may have disastrous consequences. Spreadsheets are popular software for such RA. The uncertainty about the exact values of the simulation inputs $\mathbf{z}$ is called *subjective* or *epistemic*, whereas the "intrinsic" uncertainty in random simulation is called *objective* or *aleatory*.

SA and RA address different questions: "Which are the most important inputs in the simulation model?" and "What is the probability of a given event happening?" respectively. So, SA may identify those inputs for which the distribution in RA needs further refinement.

Methodologically, we use a Monte Carlo method to sample input combination $\mathbf{z}$ from its given distribution. Next we use this $\mathbf{z}$ as input into the given simulation model. We run the simulation model to transform $\mathbf{z}$ into w: so-called *propagation of uncertainty*. We repeat these steps "many" times to obtain the EDF of w. Finally, we

use this EDF to estimate $P(w > c_f)$. This method is also known as *nested simulation*; see Kleijnen (2015, p. 284).

In *expensive* simulation, we replace the simulation model by its *metamodel*. The true $P(w > c_f)$ may be better estimated through inexpensive sampling of many values from the metamodel, which is estimated from relatively few I/O values obtained from the expensive simulation model.

Instead of $P(w > c_f)$ we may estimate the *excursion set*, which is the set of input combinations that give outputs that exceed a given threshold. Obviously, the volume of the excursion set is closely related to $P(w > c_f)$. Details on RA though Kriging and regression metamodels are given in Kleijnen (2015, p. 221) and Song and Nelson (2017).

8.6 Simulation Optimization

The importance of optimization is demonstrated by the many WSC publications on this topic. The simplest optimization problem has no constraints for the inputs or the outputs, has no uncertain inputs, and concerns the expected value of a single output, $E(w)$. This $E(w)$ may represent the probability of a binary variable. However, $E(w)$ excludes quantiles and the mode of the output distribution. Furthermore, the simplest optimization problem assumes continuous inputs, excluding MRP and MCP (also see Sect. 8.1).

There are many optimization methods; also see Ryzhov and Chen (2017). We, however, focus on RSM (using linear regression) and Kriging. Jalali and Van Nieuwenhuyse 2015 claims that metamodel-based optimization is "relatively common" and that RSM is the most popular metamodel-based method, while Kriging is popular in theoretical publications. We add that in *expensive* simulation it is impractical to apply optimization methods such as the popular *OptQuest*. A single simulation run may be computationally inexpensive, but there may be extremely many input combinations. Furthermore, most simulation models have many inputs, which leads to the *curse of dimensionality*, so we should apply screening before optimization. Moreover, a single run is expensive if we want an accurate estimate of the steady-state performance of a queueing system with a high traffic rate. Finally, if we wish to estimate a small $E(w) = p$ (e.g., $p = 10^{-7}$), then we need extremely long simulation runs (unless we successfully apply importance sampling).

8.6.1 Response Surface Methodology

RSM uses a sequence of local experiments, and has gained a good track record; see Kleijnen (2015, p. 244), Law (2015, pp. 656–679), and Myers et al. (2009). We assume that before RSM is applied, we have identified the important inputs and their experimental area (RSM and screening may be combined; see Kleijnen 2015, p. 245).

Methodologically, RSM tries to minimize $\mathrm{E}(w|\mathbf{z})$. We start RSM in a given point; e.g., the combination currently used in practice. In the *neighborhood* of this point we fit a first-order polynomial, assuming white noise; however, RSM allows $\mathrm{Var}(w)$ to change in a next step. Unfortunately, there are no general guidelines for determining the appropriate *size* of the local area in each step. To fit a first-order polynomial, we use a *R-III design* (see Sect. 8.2.1). To quantify the *adequacy* of this estimated polynomial, we may compute R^2 or cross-validation statistics; see Sect. 8.3.5. In the following steps, we use the *gradient* implied by this first-order polynomial: $\nabla(\widehat{y}) = \widehat{\gamma}_{-0}$ where -0 means that the intercept $\widehat{\gamma}_0$ is removed from the $(k+1)$-dimensional vector with the estimated regression parameters $\widehat{\gamma}$. This $\nabla(\widehat{y})$ implies the *steepest descent* direction. We take a step in that direction, trying intuitively selected values for the *step size*. After a number of such steps, w will increase (instead of decrease) because the *local* first-order polynomial becomes inadequate. When such deterioration occurs, we simulate the $n > k$ combinations of the R-III design—but now centered around the best combination found so far. We reestimate the polynomial and in the resulting new steepest descent direction we again take several steps. Obviously, a *plane* (implied by a first-order polynomial) cannot adequately represent a *hill top* when searching to maximize w or— equivalently—minimize w. So, in the neighborhood of the latest estimated optimum we now fit a *second-order* polynomial, using a CCD (see Sect. 8.2.4). Next we use the derivatives of this polynomial to estimate the optimum; we may apply *canonical analysis* to examine the shape of the estimated optimal subregion: does this subregion give a unique minimum, a saddle point, or a ridge with stationary points? If time permits, then we may try to escape from a possible local optimum and *restart* the search from a different initial local area.

While applying RSM, we should not eliminate inputs with *nonsignificant* effects in a local first-order polynomial: these inputs may become significant in a next local area. The selection of the number of replications in RSM for random simulation is a moot issue; see Sect. 8.3.3 and the SPRT for SB in Sect. 8.4.2.

After discussing classic RSM, we summarize three extensions. (i) A *scale-independent* steepest-descent direction that accounts for $\mathbf{\Sigma}_{\widehat{\gamma}}$ is discussed in Kleijnen (2015, pp. 252–253). (ii) In practice, simulation models have *multiple* responses types (see again Sects. 8.3.1 and 8.4.2). For such situations the RSM literature offers several approaches, but we focus on *generalized RSM* (GRSM), which solves the following *constrained nonlinear random optimization problem*:

$$\begin{aligned} &\min_{\mathbf{z}} \mathrm{E}(w^{(1)}|\mathbf{z}) \\ &\mathrm{E}(w^{(l')}|\mathbf{z}) \geq c_{l'} \ (l' = 2, \ldots, r) \\ &L_j \leq z_j \leq H_j \text{ with } j = 1, \ldots, k. \end{aligned} \tag{8.10}$$

GRSM combines RSM and *interior point* methods from MP, avoiding creeping along the boundary of the feasible area that is determined by the constraints on the random outputs and the deterministic inputs. So, GRSM moves faster to the optimum, and is scale independent. For details we refer to Kleijnen (2015, pp. 253–258). (iii) Because the GRSM heuristic may miss the true optimum, we can test the first-order

necessary optimality or *Karush–Kuhn–Tucker* (KKT) conditions. To test these conditions, we may use *parametric bootstrapping* that samples w^* from the assumed distribution; namely, a multivariate normal) with parameters estimated from the original w. Details are given in Kleijnen (2015, pp. 259–266).

8.6.2 Kriging for Optimization

Kriging is used by *efficient global optimization* (EGO), which is a popular *sequential* method that balances *local* and *global* search; i.e., EGO balances *exploitation* and *exploration*. We present only the *basic* EGO variant for *deterministic* simulation; also see the classic EGO reference, Jones et al. (1998). There are many more variants for deterministic and random simulations, constrained optimization, multi-objective optimization including Pareto frontiers, the "excursion set" or "admissible set", robust optimization, estimation of a quantile, and Bayesian approaches; see Kleijnen (2015, pp. 267–269).

In this basic variant we find the best simulated "old" output: $f_{\min} = \min_i w(\mathbf{x}_i)$ $(i = 1, \ldots, n)$. To select a new combination $\mathbf{x}_0$, we consider $\widehat{y}(\mathbf{x}_0)$ and $s^2[\widehat{y}(\mathbf{x})]$ (we suppress $\widehat{\psi}$); e.g., if two new points $\mathbf{x}_0$ and $\mathbf{x}_0'$ have $\widehat{y}(\mathbf{x}_0) = \widehat{y}(\mathbf{x}_0')$ and $s^2[\widehat{y}(\mathbf{x}_0)] > s^2[\widehat{y}(\mathbf{x}_0')]$, then we explore $\mathbf{x}_0$ because $\mathbf{x}_0$ has a higher probability of improvement (lower w). We know that $s^2[\widehat{y}(\mathbf{x}_0)]$ increases as $\mathbf{x}_0$ lies farther away from $\mathbf{x}_i$; see (8.7). Actually, we estimate the maximum of the *expected improvement* (EI), which is reached if either $\widehat{y}(\mathbf{x}_0)$ is much smaller than $f_{\min}$ or $s^2[\widehat{y}(\mathbf{x}_0)]$ is relatively large so $\widehat{y}(\mathbf{x}_0)$ is relatively uncertain.

More precisely, we start with a pilot sample—typically selected through LHS—which results in $(\mathbf{X}, \mathbf{w})$, Next we find $f_{\min} = \min_{1 \le i \le n} w(\mathbf{x}_i)$. We also fit a Kriging metamodel $\widehat{y}(\mathbf{x})$. Together, this gives $\mathrm{EI}(\mathbf{x}) = \mathrm{E}\left[\max\left(f_{\min} - \widehat{y}(\mathbf{x}), 0\right)\right]$. Jones et al. (1998) derives the EI estimator

$$\widehat{\mathrm{EI}}(\mathbf{x}) = \left(f_{\min} - \widehat{y}(\mathbf{x})\right) \Phi\left(\frac{f_{\min} - \widehat{y}(\mathbf{x})}{s[\widehat{y}(\mathbf{x}_0)]}\right) + s[\widehat{y}(\mathbf{x}_0)]\phi\left(\frac{f_{\min} - \widehat{y}(\mathbf{x})}{s[\widehat{y}(\mathbf{x}_0)]}\right) \tag{8.11}$$

where Φ and ϕ denote the cumulative distribution function (CDF) and the PDF of N(0, 1). Using (8.11), we find the estimate of $\mathbf{x}$ that maximizes $\widehat{\mathrm{EI}}(\mathbf{x})$ (say) $\widehat{\mathbf{x}}_{opt}$ Actually, to find $\widehat{\mathbf{x}}_{opt}$, we may use either a global optimizer or a set of candidate points selected through LHS. Next we use this $\widehat{\mathbf{x}}_{opt}$ to obtain $w(\widehat{\mathbf{x}}_{opt})$. Then we fit a new Kriging model to the augmented I/O data. We update n, and return to (8.11)—until we satisfy a stopping criterion; e.g., $\widehat{\mathrm{EI}}(\widehat{\mathbf{x}}_{opt})$ is "close" to 0. Note that EGO is combined with bootstrapping in Kleijnen and Mehdad (2013).

Now we consider the constrained optimization problem in (8.10), augmented with constraints for $\mathbf{z}$ (e.g., budget constraints) and the constraint that z must belong to the set of nonnegative integers $\mathbf{N}$. To solve this problem, we might apply an EGO variant, as mentioned above. Alternatively, we may apply *Kriging and integer mathematical*

programming (KIMP), which combines (i) *sequentialized* designs to specify the next combination, like EGO does; (ii) *Kriging* to analyze the resulting I/O data, and obtain explicit functions for $\mathrm{E}(w^{(l)}|\mathbf{z})$ $(l = 1, \ldots, r)$, like EGO does; (iii) *integer nonlinear programming* (INLP) to estimate the optimal solution from these explicit Kriging models, unlike EGO. Experiments with KIMP and OptQuest suggest that KIMP requires fewer simulated combinations and gives better estimated optima.

8.6.3 Robust Optimization

The estimated optimum (see the preceding two subsections) may turn out to be inferior because it ignores uncertainties in the noncontrollable environmental variables; i.e., these uncertainties create a *risk* (also see RA in Sect. 8.5.6). *Taguchians* emphasize that in practice some inputs of a manufactured product (e.g., a car) are under complete control of the engineers (the car's design), whereas other inputs are not (the driver). Taguchians therefore distinguish between (i) *controllable* (decision) variables, and (ii) *noncontrollable* (environmental, noise) factors. Let the first k_{C} of the k simulated inputs be controllable, and the next k_{NC} inputs be noncontrollable. Let $\mathbf{z}_{\mathrm{C}}$ and $\mathbf{z}_{\mathrm{NC}}$ denote the vector with the k_{C} controllable and the k_{NC} noncontrollable inputs.

Methodologically, Taguchians assume a single output (say) w, focusing on its mean $\mathrm{E}(w)$ and its variance, which is caused by $\mathbf{z}_{\mathrm{NC}}$ so $\sigma^2(w|\mathbf{z}_{\mathrm{C}}) > 0$. These two outputs are combined into a *scalar loss function* such as the *signal-to-noise* or *mean-to-variance* ratio $\mathrm{E}(w)/\sigma^2(w|\mathbf{z}_{\mathrm{C}})$; see Myers et al. (2009, pp. 486–488). We, however, prefer to use $\mathrm{E}(w)$ and $\sigma(w|\mathbf{z}_{\mathrm{C}})$ (obviously, $\sigma(w|\mathbf{z}_{\mathrm{C}})$ has the same scale as $\mathrm{E}(w)$ has) *separately* so we can use *constrained optimization*; i.e., given an upper threshold c_σ for $\sigma(w|\mathbf{z}_{\mathrm{C}})$, we try to solve $\min_{\mathbf{z}_{\mathrm{C}}} \mathrm{E}(w|\mathbf{z}_{\mathrm{C}})$ such that $\sigma(w|\mathbf{z}_{\mathrm{C}}) \leq c_\sigma$. Constrained optimization is also discussed in Myers et al. (2009, p. 492).

The Taguchian worldview is successful in production engineering, but statisticians criticize the statistical techniques. Moreover—compared with real-life experiments—simulation experiments have more inputs, more input values, and more input combinations; see again Sect. 8.1. Myers et al. (2009, pp. 502–506) combines the Taguchian worldview with the statisticians' RSM. Whereas Myers et al. (2009) assumes that $\mathbf{z}_{\mathrm{NC}}$ has $\boldsymbol{\Sigma}_{\mathrm{NC}} = \sigma^2(w|\mathbf{z}_{\mathrm{C}})I$, we assume a general $\boldsymbol{\Sigma}_{\mathrm{NC}}$. Whereas Myers et al. (2009) superimposes contour plots for $\mathrm{E}(w|\mathbf{z}_{\mathrm{C}})$ and $\sigma^2(w|\mathbf{z}_{\mathrm{C}})$ to estimate the optimal $\mathbf{z}_{\mathrm{C}}$, we use MP. This MP, however, requires specification of the threshold c_σ. In practice, managers may find it hard to select a specific value for c_σ, so we may try different c_σ values and estimate the corresponding *Pareto-optimal* efficiency frontier. To estimate the variability of this frontier that results from the estimators of $\mathrm{E}(w|\mathbf{z}_{\mathrm{C}})$ and $\sigma(w|\mathbf{z}_{\mathrm{C}})$, we may use bootstrapping. Instead of RSM we may apply Kriging. Details are given in Kleijnen (2015, pp. 273–284).

Finally, we summarize *RO in MP*; see again Bertsimas and Mišić (2017). If MP ignores the uncertainty in the coefficients of the MP model, then the resulting so-called *nominal solution* may easily violate the constraints in the given model.

RO may give a slightly worse value for the goal variable, but RO increases the probability of satisfying the constraints; i.e., a robust solution is "immune" to variations of the variables within the so-called *uncertainty set U*. Yanikoğlu et al. (2016) derives a specific *U* for **p** where **p** denotes the unknown PDF of $\mathbf{z}_{NC}$ that is compatible with the given historical data on $\mathbf{z}_{NC}$. RO in MP develops a computationally tractable *robust counterpart* of the original problem. Compared with the output of the nominal solution, RO may give better worst-case and average outputs.

Acknowledgements I thank the editors for inviting me to contribute a chapter to this book, commemorating the 50th anniversary of the Winter Simulation Conferences.

References

Ankenman B, Nelson B, Staum J (2010) Stochastic Kriging for Simulation Metamodeling. Oper Res 58(2):371–382

Barton RR (1997) Design of experiments for fitting subsystem metamodels. In: Proceedings of the 1997 winter simulation conference, pp 303–310

Bertsimas D, Mišić VV (2017) Robust product line design. Oper Res 65(1):19–37

Biles BE, van Beers WCM, Kleijnen JPC, van Nieuwenhuyse I (2007) Kriging metamodels in constrained simulation optimization: an exploratory study. In: Proceedings of the 2007 winter simulation conference, pp 355–362

Binois M, Gramacy RB, Ludkovskiz M (2016) Practical heteroskedastic Gaussian process modeling for large simulation experiments. ArXiv, 17 Nov 2016

Chen X Kim K-K (2013) Building metamodels for quantile-based measures using sectioning. In: Proceedings of the 2013 winter simulation conference, pp 521–532

Dellino G, Kleijnen JPC, Meloni C (2009) Robust simulation-optimization using metamodels. In: Proceedings 2009 winter simulation conference, pp 540–550

Dellino G, Kleijnen JPC, Meloni C (2010) Parametric and distribution-free bootstrapping in robust simulation optimization. In: Proceedings 2010 winter simulation conference, pp 1283–1294

Erickson CB, Sanchez SM, Ankenman BE (2016) Comparison of Gaussian process modeling software. In: WSC 2016 proceedings, pp 3692–3693

Frazier PI, Jedynak B, Chen L (2012) Sequential screening: a Bayesian dynamic programming analysis. In: Proceedings 2012 winter simulation conference, Berlin, pp 555–566

Govindan E (2016) Yosida approximations of stochastic differential equations in infinite dimensions and applications. Springer

Gramacy RB (2015) laGP: large-scale spatial modeling via local approximate Gaussian processes in R. J Stat Softw (available as a vignette in the laGP package)

Jalali H, Van Nieuwenhuyse I (2015) Simulation optimization in inventory replenishment: a classification. IIE Trans 47(11):1217–1235

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13:455–492

Kleijnen JPC (1995) Case study: statistical validation of simulation models. Eur J Oper Res 87(1):21–34

Kleijnen JPC (2015) Design and analysis of simulation experiments, 2nd edn. Springer

Kleijnen JPC (2017) Regression and kriging metamodels with their experimental designs in simulation: a review. Eur J Oper Res 256:1–16

Kleijnen JPC, Mehdad E (2013) Conditional simulation for efficient global optimization. In: Proceedings 2013 winter simulation conference, pp 969–979

Kleijnen JPC, Mehdad E, van Beers WCM (2012) Convex and monotonic bootstrapped kriging. In: Proceedings of the 2012 winter simulation conference, pp 543–554
Kleijnen JPC, Shi W (2017) Sequential probability ratio tests: conservative and robust. In: CentER discussion paper, vol 2017-001. CentER, Center for economic research, Tilburg
Law AM (2015) Simulation modeling and analysis, 5th edn. McGraw-Hill, Boston
Liu Z, Rexachs D, Epelde F, Luque E (2017) A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity. Comput Ind Eng 103:300–309
Lophaven SN, Nielsen HB, Sondergaard J (2002) DACE: a matlab kriging toolbox, version 2.0. IMM Technical University of Denmark, Kongens Lyngby
Mitchell TJ, Morris MD (1992) The spatial correlation function approach to response surface estimation. In: Proceedings of the 2012 winter simulation conference, pp 565–571
Montgomery DC (2009) Design and analysis of experiments, 7th edn. Wiley, Hoboken, NJ
Myers RH, Montgomery DC, Anderson-Cook CM (2009) Response surface methodology: process and product optimization using designed experiments, 3rd edn. Wiley, New York
Rosen SL, Slater D, Beeker E, Guharay S, Jacyna G (2015) Critical infrastructure network analysis enabled by simulation metamodeling. In: Proceedings of the 2015 winter simulation conference, pp 2436–2447
Ryzhov IO, Chen Y (2017) Bayesian belief models in simulation-based optimization. In: Advances in modeling and simulation: seminal research from 50 years of winter simulation conferences, pp 181–217
Sargent RG, Goldsman DM, Yaacoub T (2016) A tutorial on the operational validation of simulation models. In: Proceedings of the 2016 winter simulation conference, pp 163–177
Shi W, Kleijnen JPC (2017) Testing the assumptions of sequential bifurcation in factor screening, vol 2017-06. Discussion paper, Center for economic research, Tilburg
Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang R-J (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Multidiscip Optim 27(5):302–313
Song E, Nelson BL (2017) Input model risk: where we stand and where we are going. In: Tolk A, Fowler J, Shao G, Yücesan E (eds) Advances in modeling and simulation: seminal research from 50 years of winter simulation conferences, pp 63–80
Timm IJ, Lorig F (2015) A survey on methodological aspects of computer simulation as research technique. In: Proceedings of the 2015 winter simulation conference, pp 2704–2715
Uhrmacher A, Brailsford S, Liu J, Rabe M, Tolk A (2016) Panel—reproducible research in discrete event simulation—a must or rather a maybe? In: Proceedings of the 2016 winter simulation conference, pp 1301–1315
Ulaganathan S, Couckuyt I, Dhaene T, Laermans E (2014) On the use of gradients in kriging surrogate models. In: Proceedings of the 2014 winter simulation conference, pp 2692–2701
Van Beers WCM, Kleijnen JPC (2004) Kriging interpolation in simulation: a survey. In: Proceedings of the 2004 winter simulation conference, pp 113–121
Van Ham G, Rotmans J, Kleijnen JPC (1992) Techniques for sensitivity analysis of simulation models: a case study of the CO2 greenhouse effect. Simulation 58(6):410–417
Wagner H (1975) Principles of operations research; with applications to managerial decisions, 2nd edn. Prentice-Hall Inc, Englewood Cliffs, New Jersey
Wan H, Ankenman BE, Nelson BL (2010) Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening. INFORMS J Comput 22(3):482–492
Yanikoğlu I, den Hertog D, Kleijnen JPC (2016) Robust dual-response optimization. IIE Trans Ind Eng Res Dev 48(3):298–312
Zeigler BP, Praehofer H, Kim TG (2000) Theory of modeling and simulation, 2nd edn. Academic, San Diego
Zhou E, Wu D (2017) Simulation optimization under input uncertainty. In: Tolk A, Fowler J, Shao G, Yücesan E (eds) Advances in modeling and simulation: seminal research from 50 years of winter simulation conferences, pp 219–247

Chapter 9
Better Big Data via Data Farming Experiments

Susan M. Sanchez and Paul J. Sánchez

Abstract The term 'big data' has become intertwined with 'data mining' in the minds of many people. Modern computing can generate massive amounts of data via simulation studies, but a key drawback to the data mining paradigm is that it relies on observational data and thus limits the types of insights that can be gained. We can do much better with 'data farming,' a metaphor that captures the notion of purposeful data generation from simulation models. Prospective designs of experiments can establish causal relationships, in contrast to data mining that can only find correlations. The use of large-scale designed experiments lets us grow simulation output efficiently and effectively, and is a game changer in terms of the power and flexibility it offers analysts and decision makers. When combined with modern simulation tools and cluster computing, it allows studies to focus on much broader questions and obtain much richer insights. In this chapter, we discuss the implications data farming has on model building, verification and validation, input modeling and data requirements, multi-criteria decision-making and tradeoff analysis, and a few ethical aspects of the decision process.

9.1 Background and Motivation

Over the past 50+ years, simulation has experienced rapid and continual growth. This growth has been fueled in part by the exponential increase in computing capabilities and availability (Moore 1965), but also by the manifold contributions of researchers and practitioners who have created better tools and techniques in diverse areas such as input modeling, random variate generation, output analysis, model paradigms and their associated implementations, model verification and validation, optimization, and ranking & selection, to name a few.

S.M. Sanchez (✉) · P.J. Sánchez
Naval Postgraduate School, Monterey, CA, USA
e-mail: ssanchez@nps.edu

P.J. Sánchez
e-mail: pjsanche@nps.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_9

These developments have provided new opportunities for simulation-based systems exploration. In its early years, simulation was often referred to as a "method of last resort" (Wagner 1969) because of its complexity, cost, and limitations. Given the current state of the art, it can now be argued quite strongly that simulation should often be considered a method of first resort (Lucas et al. 2015). The resulting shift in views and attitudes about how simulation can be used to support decision-making is reflected by an evolution in the types of questions we seek to address with our simulation models, from 'what is?' to 'what if?,' and more recently to 'what matters?, how?, and why?'

'What is?' questions can be answered when we view simulation as a way of gathering descriptive statistics about a single configuration of a system. Monte Carlo simulation falls into this category. For instance, if we are trying to determine the distribution of the output of an electronic circuit with randomly distributed component characteristics, or evaluate an intractable multidimensional integral for physics modeling, Monte Carlo simulation works quite well.

It is a short step from asking 'what is?' to asking 'what if?' For example, those who build models of a factory floor might be interested in finding out whether throughput would improve if a new piece of equipment or more workers are added. Similarly, those building models of emergency rooms might be interested in how much of a reduction in patient waiting time can be obtained if additional beds, lab testing personnel, or doctors are added to the system. Such manufacturing and healthcare applications have been mainstays at the Winter Simulation Conference (WSC) for decades. The software has changed, and the models themselves may have become more complicated, but these basic types of questions remain compelling.

'What is?' and 'what if' questions are answered primarily using descriptive statistics, but simulation can be even more powerful as we move to inferential statistics. By varying inputs in carefully chosen ways and exploring or building metamodels of the input/output relationships, we can answer 'what matters? how? and why?' within the context of our simulation model. We will discuss this theme further in Sects. 9.2.4 and 9.2.6.

9.2 A 'Think Big' Mindset

In a very real sense, the field of simulation has always been on the forefront of 'thinking big.' Simulation has extended the scope and magnitude of studies in all branches of science and engineering, since it can be used to model systems which are beyond the reach of other techniques due to issues of size or complexity. It can be used to model systems where time or safety issues prevent us from collecting real-world data. We advocate extending the 'think big' philosophy to the analysis process itself.

Most people have heard of data mining, and would tend to interpret a discussion about 'thinking big' in those terms. We prefer to frame the problem in terms of 'data farming,' which is a technique for prospectively generating or growing data by purposefully running simulation experiments at carefully selected combinations of

input settings (design points). This is a significant contrast with data mining, which attempts to retroactively find correlations within large data sets. We find that using real-world mining versus farming is a useful metaphor for understanding the difference between the two techniques (Sanchez 2010, 2015).

WSC's first introduction to the term 'data farming' was in 1999 (Horne 1999). The usage quickly evolved to include designed experiments, and so did the metaphor. Since 2000, the term has been used in over 40 WSC papers, describing a variety of applications (Lucas et al. 2007; Kang and McDonald 2011; Feldkamp et al. 2015), new design or analysis methodologies (Vieira et al. 2011; Hernandez et al. 2012; MacCalman et al. 2016), and tutorials or panel discussions (Sanchez and Wan 2015; Robinson et al. 2015). Other papers describe new approaches for dealing with large-scale experiments, even if they do not explicitly mention data farming (e.g., Méndez-Vázquez et al. (2013)).

Beginning in 2015, 'Data Farming 101' became a regular half-day pre-conference tutorial, and one of the Titan talks at WSC '16 was 'A data farmer's almanac' (Sanchez 2016). Data farming has made inroads into the way the Department of Defense and its allies use simulation. At the Naval Postgraduate School's SEED Center for Data Farming, promoting simulation experiments and efficient designs, over 180 students (active duty military or civilian defense analysts) have received graduate degrees applying data farming methods to a variety of problems in defense and homeland security (SEED Center for Data Farming 2015). Data farming 'Technical Activities' have been set up under the NATO umbrella for many years (Kallfass and Schlaak 2012; Seichter and Horne 2014; Huber and Kallfass 2015). The NATO MSG-088 team received the 2014 Scientific Achievement Award from NATO Science & Technology Organization (Horne et al. 2014).

Topics in big data have been sprinkled throughout WSC papers for several years. The 2013 Keynote Address (Bonabeau 2013) and 'Modeling and Simulation Grand Challenges' panel discussion (Taylor et al. 2013) both bring up big data, paving the way for a mini-track on 'Big Data and Decision Making' during 2014 and 2015. Interest in the interface between big data and simulation continues to increase, as evidenced in part by the 2017 I-Sim Research Workshop: Towards an Ecosystem of Simulation Models and Data.

By the end of this section, we intend to convince you that data farming provides a real revolution in the quality and quantity of insights that you can gain from your simulation models. Our bottom line is that 'thinking big' about your simulation models should mean thinking about large-scale experiments. Along the way, we will describe how big questions, big data inputs, and big data outputs tie into a cohesive whole.

9.2.1 Big Questions

Why is simulation such a powerful and important tool? Common benefits include cost, safety, flexibility, the ability to achieve greater realism in modeling assumptions, and the ability to model planned or prospective systems, but the primary goal

in all cases is to expand our knowledge of some system. We want to learn things—in other words, to answer questions—that can be addressed more easily, quickly, or safely in the virtual world than in the real world, or more realistically than by using assumption-laden analytical models.

There are big opportunities if we strive to answer big, complex questions. From the outset, this means we seek multiple insights from our simulation study, not just the answer to a single tightly focused question. In observational studies, this is the difference between (say) looking over data from people in two hospitals to see which hospital has a longer average patient stay, versus looking over data from people across the nation. In the latter case we might uncover variables that correlate not just with patient stays, but also with prevalence of various diseases, demographic information, patient histories, physician characteristics, and more. Delving into the data might yield a number of interesting insights that had not been previously considered. With so much information at your fingertips, it would seem very strange to restrict yourself to looking at only one aspect. Yet, this is what happens if we simulate a complex system and focus on a single response! We are ignoring the richness of all the other potentially interesting results, which might be explored either as end-of-run aggregates or as internal interactions or behaviors. Multiple performance measures may conflict with each other, which highlights the need to study them jointly, and the importance of trade-off analysis.

It can also be useful to consider a multi-model approach. Sometimes, it might be that two (or more) modeling platforms have different strengths and weaknesses in their representations of the system being modeled. For example, we might choose to model humanitarian assistance operations using both a process model that easily represents logistical operations, and an agent-based model that more easily models human behavior. Alternatively, we have seen analysts use several models operating at different fidelities, such as two-dimensional and three-dimensional models, in energy, wind tunnel simulations, or physics scenarios. If multiple models yield similar insights despite their differences, it provides stronger evidence that those insights are real and not artifacts of a particular simulation modeling paradigm (Page 2016, https://www.coursera.org/learn/model-thinking).

9.2.2 Big Data as Model Inputs

There is no universally accepted definition for the term ‘big data,’ although it is often characterized by ‘the 3 V’s’: volume, variety, and velocity (Laney 2001). Volume is the amount of data. Variety refers to the diverse characteristics and encoding of the data, and can be particularly challenging to deal with when pooling ill-structured data from multiple sources. Velocity refers to the speed at which new data arrives. With simulation experiments, big data can come into play in a variety of ways for model inputs.

A typical simulation model may have hundreds or thousands of embedded parameters and assumptions. Think of a simulation as a gray box that maps inputs to

outputs. We call this a gray box because it exhibits both black box and white box characteristics. Despite the complete visibility of the model implementation, it is sufficiently complex that the I/O mapping is not self-evident. Nonetheless, subject-matter experts may have partial knowledge about some aspects of the system. If we conceptualize each of the continuous inputs as a knob that can be turned, and each of the discrete or categorical inputs as a set of radio buttons that can be toggled, we very quickly find ourselves in a big data setting.

Sometimes it is clear that the input data requirements are large. For example, the U.S. Navy uses the Synthetic Theater Operations Research Model (STORM) simulation for campaign analysis (McDonald et al. 2014; Morgan et al. 2017).

> All told, STORM models typically require upwards of 40 MB of data spread over nearly 150 input files. These files contain many types of data in a variety of formats with a myriad of details about individuals, groups, and classes of entities, and decision rules that capture CONOPs and tactics. Acquiring, vetting, and verifying the data is a challenging task involving numerous organizations spread throughout the Department of Defense" (Morgan et al. 2017).

The input data in a scenario like this exhibit both volume and variety.

Velocity applies to simulation models that pull inputs directly from data sources in (near) real time. These simulations can fall into the realm of control systems, where the purpose of the simulation may be to provide guidance on how to respond to changes in the underlying system in (near) real time. Traffic simulations are one example (Henclewood et al. 2011; Suzumura and Kanezashi 2014), and this style of modeling may become more prevalent as time goes on and the availability of real-time data from sources such as GPS systems and self-driving cars continues to increase. Military or cybersecurity applications, air traffic control, fire service emergency cover, real-time control of machines and production processes, and health care are other situations of where this type of input may be available (see, e.g., Cheng 2007; Tan et al. 2012; Vahdatikhaki et al. 2013; Linares et al. 2016 for examples). This type of input can be further complicated by the reliability and integrity, or lack thereof, of data found in databases. Our experience is that databases are often repositories of what was recorded rather than what was true. Anybody who has worked with real-world data knows that it is often messy, incomplete, or erroneous. Simulation models based on raw input streams need automated ways of coping with potentially flawed data.

In other cases, even apparently simple models may have a large number of embedded parameters. For example, in an agent-based model we can vary the numbers of entities in different classes of agents. Agents may also have different starting locations, different propensities for moving, staying close, sharing, or interacting with other agents, and different ways of interacting with their environment. These change in time- and state-dependent manners, easily yielding hundreds or thousands of inputs that could be explored. Although these inputs can be viewed as distinct factors, it is sometimes productive to view them as a set of randomly generated effects from a parameterized distribution. For example, rather than specifying individual values for each agent's speed, consider generating the speeds for a set of agents from a distribution that requires only a few parameters.

Dealing with these big data sets can be challenging, and requires a modeling decision. We can fit parametric distributions to existing data, which may be a way of getting tail behavior not already present in the existing data. Alternatively, we can use empirical distributions by bootstrapping the existing data. Neither of these approaches was computationally practical for even moderately sized data sets in the early days of simulation, but both are viable options 50 years later.

All too often, people think about inputs as descriptive of a 'ground truth' that must be accurately estimated. This may be appropriate for computer models in science and engineering, such as models that might help predict the properties of new materials. However, as Heraclitus said approximately 2,500 years ago: "change is the only constant." Even if we have perfect knowledge of the parameterized distribution of customers' arrivals and service times are for a call center this day, week, or month, it will be different in the future. Assessing the sensitivity of the results over a range of 'reasonable' inputs may be more informative and useful than trying to improve our current estimates.

In cases where data are unavailable or suspect, or might change in the future, we can vary the inputs and seek to identify robust regions where the decisions are invariant despite uncertainties in the inputs. This approach applies equally whether the uncertainties are associated with individual inputs, parameters of input distributions, or the input distributions themselves. Chapter 5 in this volume describes techniques for characterizing or hedging against input risk (Song and Nelson 2017), but explicit input variation provides yet another alternative.

9.2.3 Big Data as Model Outputs

Not surprisingly, simulation output can quickly become big data. At a bare minimum, our model should produce summary information such as interval estimates for the primary responses of interest. Much better is to provide distributional results such as quantiles of the output distributions, or proportions of time that a performance measure exceeds certain thresholds or remains within certain bounds. If the output is comprised of raw outcomes, choices of response metrics can be deferred to the analysis stage, although this can be expensive in both storage and time requirements. The volume of output data is further increased with a 'big question' outlook, because there are inevitably numerous responses of interest. For example, the STORM campaign analysis model can take hours to complete and generates tens or hundreds of Gigabytes (GB) of output data in a mixture of dozens of output files and a database for a single replication (Morgan et al. 2017). An interesting feature of this study is that no single output measure, by itself, shows the quality of the policies under consideration. Instead, even for the small training scenario, there are over 20 'what it takes to win' metrics of interest that often conflict with one another.

A 'big output' approach can increase data volume to the point that the time required for printing detailed output files can be a bottleneck for completing a simulation study. Fortunately, we simulators have some helpful tricks up our collective

sleeves. Some simulations run in a small fraction of the time if detailed I/O, such as state-space logging, is turned off—having a toggle for this is useful. Alternatively, if storage space is limited it may be beneficial to print a pre-selected subset of the outputs or to 'pipe' some outputs through, e.g., a Kalman filter, to reduce the volume for preliminary analysis. Cluster computing, which needs to move data back and forth over a network, can reap significant speedups using these techniques. As long as you retain the random number seeds used to generate the runs, you have the capability to go back and regenerate the corresponding full output if needed. The use of Stochastic Information Packets (SIPs), a "concept of data structures for storing arrays of simulated realizations," may also be useful (Thibault 2014; Savage and Thibault 2015). Hardware choices have an impact: current solid-state drives (SSDs) have I/O speeds an order of magnitude faster than current hard disk drives (HDDs), but HDDs offer a higher density of storage at lower cost. Some might posit that data volume will become less of an issue in the future, as storage continues to become cheaper, I/O continues to become faster, and data scientists continue developing computationally efficient tools for combing through large data sets. We think otherwise—the history of computing shows that as hardware and software capabilities increase, we always stretch them to their limits. This is consistent with the idea that "… big data can be viewed as any data set that pushes against the limits of currently available technology" (Sanchez 2015).

9.2.4 Better Big Data via Large-Scale Experiments

Our premise is that while all simulation studies can be viewed as experiments, many simulation studies are not well-designed experiments. Our goal in this section is to delineate what makes a suitable design for large-scale simulation experiments. The opportunities inherent in data farming are even greater when we take this 'big' view.

Design of experiments (DOE) has been a regular topic at WSC, from the first tutorial in 1968 (Frank 1968) to recent tutorials (Law 2014; Sanchez and Wan 2015). It also appears in recent simulation textbooks (Law 2014; Kleijnen 2015) that provide comprehensive lists of of the numerous journal articles in this area. Several topics related to DOE for simulation experiments appear in Chap. 8 in this volume (Kleijnen 2017). However, with a data farming approach our goals and recommendations differ, as discussed in Sect. 9.2.6 and explained in more detail in (Sanchez et al. 2014; Sanchez and Wan 2015). Since 2015, we have also offered 'Data Farming 101' as a pre-conference activity to provide a detailed, hands-on introduction to the principles and practice of large-scale simulation experiments.

The basic principles of DOE are randomization, replication, and control. In real-world experiments, randomization guards against hidden sources of bias, replication is a brute-force means of increasing the discriminating power of statistical statements (e.g., tighter confidence intervals, or smaller p-values for hypothesis tests), and control can further improve the amount of information available from a limited set of data. For simulation experiments, the analyst has fewer restrictions and more

opportunities, so these concepts are used somewhat differently. Given our control over random number seeds and streams, "the results from a simulation experiment are perfectly repeatable, and randomization is not needed to guard against hidden or uncontrollable sources of bias," replication is used to obtain "multiple experimental units (runs or batches) to get a sense of the magnitude of the variability" associated with the responses, and control can be exerted by specifying a design for the experiment and, in some cases, leveraging common random numbers (Sanchez 2015).

With designed experiments we can establish cause-and-effect relationships, in contrast to observational studies where conclusions are limited to statements about correlation. This is the major benefit of designed experiments. A well-designed experiment will also allow us to estimate effects without confounding. For complex models of complex systems, these effects may go beyond main effects or simple polynomial terms, and thus require using designs with more than a few levels per factor.

The ability to gain insight about a multitude of cause-and-effect relationships underpins our premise that simulation big data is 'better' than observational big data. Consider this passage from a recent WSC tutorial:

> Why are so many either extolling or lamenting big data's focus on correlation rather than causation? I suggest it is because their view of big data is observational. This is not the case for those using simulation. Using the data farming metaphor, we grow the data for analysis, rather than mine existing data. However, just as a different mindset is needed for dealing with big observational data, so a different mindset is needed for generating and dealing with big simulation data (Sanchez 2015).

The paper then goes into a more detailed discussion of simulation-generated big data.

Even when the number of factors is small, good designs are needed to avoid confounding factor effects (i.e., being unable to separately estimate or identify them). Also, poorly chosen designs may prevent you from identifying main effects in the presence of interactions. When the number of factors is large, the need for good designs is paramount: a brute-force approach to evaluating all possible factor combinations is impossible, no matter how much computing power is available, or how few potential factor levels you choose to explore. A brute-force design that studies all combinations of 100 factors, each at only two levels, for a simulation that runs in a single second would take many times the current age of the universe using the world's most powerful computer. In contrast, studying that same system with modern designs would permit exploration of a simulation that takes 20 min to run at hundreds of levels for each of the 100 factors, and run to completion in less than a week on a standard laptop, or as little as 20 min on a moderately sized computing cluster.

Another pitfall to avoid is an unwarranted focus on designs that seek to reduce the number of design points by limiting both the number of factors and the number of levels per factor. Multi-core machines and computing clusters are now readily available, which can speed up the experiment completion time in a linear manner. However, because run completion times times can vary widely for different design points or replications, restricting their number may not lead to proportional reductions in the total time. Last but not least, given the amount of time already invested

in creating a model, there is no reason to place arbitrary limits on the time allotted for running an experiment. Instead, by running large-scale experiments, you can let your model work for you.

Suffice it to say that data farming experiments benefit from much larger designs than classical DOE. The following designs, which are freely available from the Naval Postgraduate School's SEED Center for Data Farming (SEED Center for Data Farming 2015), are our current 'go-to' recommendations.

- Nearly orthogonal-and-balanced (NOB) designs (Vieira Jr et al. 2013). An Excel spreadsheet for NOB designs (NOB_Mixed_512DP.xls) allows you to build a customized design with 512 design points that can explore up to 20 factors apiece for each of $\ell = 2, \ldots, 11$ levels per factor and up to 100 continuous-valued factors simultaneously. The resulting space-filling designs facilitate trade-off analysis.
- Very large resolution V fractional factorial (RFFF or rf_cubed) and central composite designs (CCDs) (Sanchez and Sanchez 2005). These are able to simultaneously estimate all main effects and two-way interactions without confounding for up to 120 factors. The CCDs efficiently extend the rf_cubed designs to permit orthogonal estimation of full second-order metamodels. Design generators for these are available at (SEED Center for Data Farming 2015) as portable cross-platform Ruby (https://www.ruby-lang.org/) scripts, and the method has also been implemented in the R package FrF2Large (https://rdrr.io/cran/FrF2/man/FrF2Large.html). The rf_cubed designs can be used for screening on both main-effects and two-way interactions, as well as for exploring binary factors.

Designs can also be combined in interesting and useful ways. For instance, if you keep factor ranges the same (or symmetric about a central baseline design point), you can concatenate two or more orthogonal designs and maintain orthogonality—an example would be adding an RFFF design to a NOB design to provide more sampling near the corners of the input factor space. If we classify and separate the factors into decision factors (controllable in the real world) and noise factors (uncontrollable in the real world), we can either construct a crossed design from separate designs for the two factor classes, or construct a single combined design. The choice may depend not only on the data requirements, but on the types of statistical and graphical methods that will be used to convey results to the decision makers. Yet another choice is to rotate or shuffle the mapping of raw space-filling design columns to factors to further increase their space-filling properties.

All of the large-scale designs described above can be used for both screening and metamodeling purposes, as we discuss in Sect. 9.2.6. We recommend a 'big view' of screening based on our experience that quadratic and two-way interactions are often among the most impactful effects. Space-filling designs such as the NOB provide even more analysis flexibility when exploring complicated response surfaces.

It is more important to use any good design than to use a particular design. While the NOB designs (and other designs based on Latin hypercube (LH) designs) we commonly use were developed with an explicit interest in reducing the maximum pairwise correlation among main effect estimates, other designs are available in various software packages, including the DOE menu in statistical software

JMP (https://www.jmp.com/), several R packages for creating maximin or space-filling LH designs (http://CRAN.R-project.org/), and custom design software (http://www.statease.com/dx10.html). Sequential approaches are also available, including sequential bifurcation (Kleijnen and Bettonvil 1997) and its extensions (Kleijnen 2017), also (Wan et al. 2006; Shen et al. 2009), as well as batch sequential methods (Duan et al. 2017; Loeppky et al. 2010). Several simulation packages now include an excursion or scenario window where the user can create or import a list of factor combinations (design points) to investigate.

Ultimately, we recommend using a well-chosen design that is big enough to handle the scope of your simulation study, and gives sufficient flexibility for a broad variety of analysis options. Keep in mind that what seems big today may seem run-of-the-mill tomorrow. Big is a moving target, and has been throughout the history of computing.

9.2.5 Bigger can be Easier: The Nuts and Bolts of Data Farming

No matter how elegant and beautiful a concept may seem, it is only useful if it can actually be implemented. Implementing large-scale designs requires some groundwork, but the benefits far exceed the effort involved. The overall approach we use was inspired by the architecture of UNIX operating systems, where complex tasks can be accomplished via composition of a sequence of simple atomic tools. Once the design has been selected and constructed, our goal is to automate, to the fullest possible extent, the process of performing model runs and collating the results into a format suitable for analysis with your favorite statistics package.

The biggest impediment to automation is usually the model itself. Ideally, your model should meet the following three requirements.

1. Factor settings should be controlled via external input mechanisms such as command-line arguments or input files (flat files, XML, or database).
2. The model should be able to run without a graphical user interface, i.e., it should not open windows or alerts of any type.
3. The model should be capable of running in a 'batch mode,' i.e., without human intervention of any sort.

While data farming can be and has been done with models that do not meet these requirements, it then becomes a manual process which is tedious, time consuming, error prone, and does not scale well to more than a few dozen runs of the model.

Whether your model conforms to those requirements or not, data farming involves the following steps.

1. Identify all input requirements for your model—the structure, order, and format of all factors, parameters, and run-control options.

2. Choose a suitable design, a 'base configuration' for your system, and appropriate ranges of variation for your factors.
3. For each design point in your design:
    a. modify the base design factor values to the settings of the current design point;
    b. execute the model with the current settings;
    c. (optional) post-process the output to extract suitable output measures, if needed;
    d. collate the design point specification with the output measures and append to prior run results.
4. Repeat step (3) for the desired number of replications.
5. Analyze output with your favorite statistical tool(s).

If your model is compliant with the requirements previously described, some simple programming or scripting can automate this process. Automation of all the tasks in step (3) is usually accomplished with script programming or wrapper functions of some sort. Scripting languages such as Ruby (https://www.ruby-lang.org/) or Python (https://www.python.org) are excellent choices for this—both offer extremely powerful string manipulation capabilities and the ability to dynamically construct and execute external commands to perform run control. Over the years, the SEED Center has created several Ruby scripts that, while not universally applicable, can handle a broad variety of data manipulation and run control tasks in an operating-system independent fashion. These are freely available at https://bitbucket.org/paul_j_sanchez/datafarmingrubyscripts.git.

Note that we advocate iterating through all design points before proceeding to replication. This is because the desirable mathematical properties of a design require data be sampled at *all* of the design points. If the runs of your model get interrupted for any reason, such as a power outage or because variations in actual execution time result in fewer runs than you had originally intended, this strategy will greatly improve your chances of being able to maximize the information yield from the existing output.

If you are working on a multi-core machine, or have access to cluster computing, the process can be sped up significantly by using suitable cluster control software. You will need to invest some time to learn how run control is handled, but with modern cluster infrastructure the speedup is nearly linear in the number of cores available. We have had excellent results using HTCondor (HTCondor 2016), which is an open source software framework freely available from the Computer Science department at the University of Wisconsin.

It is often the case that analysis yields interesting insights that warrant further exploration. Steps 2–5 can be repeated as often as desired to generate additional data if appropriate.

9.2.6 From Better Big Data to Better Insights

So, once we have this rich set of data, what do we do with it? Three potential goals for simulation studies identified in Sanchez et al. (2012) are:

1. Developing a basic understanding,
2. Finding robust decisions or policies, and
3. Comparing different alternatives.

In keeping with the 'think big' mindset, the first goal is quite broad. It can involve nonparametric techniques, such as partition trees, to screen a large number of potential factors and determine which ones have the largest impact on a particular response. It can involve metamodeling approaches that succinctly summarize how individual responses behave based on a relatively small number of factor terms. We remark that, in the data farming world, statistical significance does not mean practical importance—and this should be considered when you construct a metamodel. For example, it is possible to throw out terms that have very small p-values (e.g., 0.001), with little noticeable effect on R^2, if their associated t-values are orders of magnitude less than those of the dominant metamodel terms. This ability to move away from using a p-value cutoff as a strict sufficient condition for metamodel inclusion means that data farmers are less apt to mistakenly identify the types of spurious results that have been reported in other contexts (Vigen 2014, 2015; Carroll 2017). Other classic assumptions, such as normality and constant variance, may not hold. Fortunately, many techniques are robust to departures from these assumptions, particularly when large amounts of data are available.

Note that many of the commonly used metamodeling techniques, including stepwise regression, logistic and multinomial logistic regression, partition trees, and kriging (Gaussian process metamodeling) are possible only because of the ready availability of computing power. Other analysis alternatives, such as bootstrapping, cross-validation, maximum likelihood estimation, random forests, and neural networks, benefit from computing power as well.

We highly recommend the use of visual representations for your data. Even the results of a set of replications for a single design point can yield a variety of interesting insights. A diverse set of graphs appears in Morgan et al. (2017), who provide multi-color dashboards that display, for each replication, whether or not each of many user-defined success metrics are met. They also consider correlation plots of key metrics; present a variety of heatmaps that show conditions, events, and resource levels across time and replications; use cluster analysis to identify 'good' and 'bad' clusters; and construct trees that show how often different events fire that allow for more in-depth explorations of why the clusters differ.

Broad understandings of the response behavior are possible for visualizations involving data from all design points. Partition trees can quickly alert a nontechnical audience to good, intermediate, and bad sets of outcomes, particularly if a green-yellow-red coding is used; color or shape can also characterize an extra dimension in two- or three-dimensional scatter plots or surface plots (Sanchez 2017). Plots of

response variability against response means or medians can help the analyst identify robust configurations (Marlow et al. 2015). Small multiples or trellis plots can do an excellent job of revealing, at a glance, differences across replications, time, or response measures (Sanchez et al. 2012; Feldkamp et al. 2015). WSC papers with graphical representations of the results from large-scale experiments include (Sanchez and Lucas 2002; Sanchez 2015). Dynamic or interactive high-dimensional data visualizations may be particularly powerful for conveying complex relationships.

There is still an art in creating compelling graphical displays (Tufte 1986). Common diagnostic graphs, such as scatter plots, may not be informative for raw results when you have thousands or hundreds of thousands of realizations, so some graphical displays are best constructed using summary information (means, quantiles, or ranges) from the design points. The special structure of model-driven big data can help guide the analysis.

The second goal—finding robust decisions or policies—involves identifying robust solutions. We are strong advocates of robustness. Robust design is a system optimization and improvement process, originally developed for manufactured product design by Taguchi (1987), that explicitly captures both the mean and variability of a response in a loss function. It has been credited as a trigger for the "explosive use of design for product quality improvement in monitoring in industrial processes" (Hedayat et al. 1999). The first WSC paper discussing how to extend the Taguchi philosophy to simulation experiments appeared in 1991 (Ramberg et al. 1991), and since that time there have been dozens of application and methodology papers that reference the philosophy. A robust design approach separates factors into two groups: the decision factors that are controllable in the real world, and the noise factors that are not. By explicitly varying both sets of factors, a robust design approach allows the analyst to find solutions (combinations of decision factors) that work well across a variety of noise factor variation. The use of a loss function provides a measure of robustness that facilitates continuous improvement.

The third goal—comparing different alternatives—involves experiments on qualitatively-different systems, and is handled by ranking & selection (R&S) approaches. R&S has been a very fruitful area for simulation research and application over the last several decades. Many researchers in this field also embrace a 'big data' view (Nelson et al. 2001; Kim and Nelson 2006; Luo and Hong 2011; Vieira Jr 2014), and the development of procedures for very large numbers of alternatives is ongoing. Recent and ongoing work takes advantage of the computational power offered by cluster computing: see Hunter and Nelson (2017) for an extensive discussion and bibliography.

The final arbiter of whether a large-scale simulation experiment is successful is whether it facilitates good decision-making. The WSC archives and simulation journal articles are full of examples, where simulation made a tangible difference—whether that involved saving lives, saving time, saving money, saving the environment, or other important decisions. Large-scale experiments can greatly increase the nature and quantity of insights you obtain from your simulation model.

In summary, when we move from real-world observational big data to simulation-generated inferential big data, we can move beyond the '3 V's' to the '3 F's' of data farming: features, factors, and flexibility. Our 'big question' view means that we recognize that the response surfaces may have many interesting features. Our 'big input data' view acknowledges that there are a large number of factors that are worth exploring. As stated in Sanchez (2015), our 'big output data' view means that we need designs capable of providing the flexibility to "to answer many questions from our experiments, even if we don't know a priori all the questions that might be asked," and consequently, our designs should facilitate "a broad variety of meta-modeling, data mining, and graphical analysis tools" for analysis purposes, as well as flexibilities in how we store and retrieve output data.

9.3 Ramifications for Decision Making

> Our objective … is to change mindsets in support of those who are actually concerned about actually solving actual problems.
> —from "Changing the Paradigm: Simulation, Now a Method of First Resort" (Lucas et al. 2015)

In writing the 'First Resort' paper from which this quote comes, we and our co-authors spent many delightful hours arguing over specific words. This quote may sound jarring, but we wrote it that way on purpose. Replace or remove any 'actual' and you change our intended meaning. A 'big question' view means we are interested in actual problems, not just stylized problems. Simulation, by its nature, is a way of avoiding Type III errors (Mitroff and Featheringham 1974)—i.e., working on the wrong problems—because simulation values computational tractability over analytical tractability, and does not require unwarranted assumptions. Large-scale simulation experiments can provide useful, robust, actionable recommendations to decision makers.

9.3.1 Ethics in Simulation

> You must do something to make the world more beautiful.— Barbara Cooney (Cooney 1982)

All decision-making involves risk. There are many ways in which the decision process can go wrong. In simulation, some of these are technical issues, such as invalid or unjustified assumptions, inadequate or insufficient models, bugs in the implementation, application of incorrect or inappropriate analytic techniques, or conflating correlation with causation, to name a few. There is also an ethical component that must be considered. Selected professional codes of ethics, and their applicability and importance to simulators, appear in Tolk (2017). Coding or modeling errors can be honest mistakes, but if they are covered up the result is faulty

analysis that potentially leads to faulty conclusions. It is also unethical to intentionally manipulate the model or model inputs to attain some preconceived conclusion. Unfortunately, if someone is trying to do this, most simulation models are sufficiently complex that they provide many degrees of freedom for 'tuning' parameters in a single-scenario study.

These problems, whether intentional or inadvertent, are mitigated or even eliminated by large-scale experiments. It becomes harder for people to 'game' the system by turning knobs to get the output they want when parameter settings which are unrealistic, but yield targeted results for a single scenario, will usually result in implausible behaviors when observed across a broad variety of scenarios. Large-scale experimentation also helps identify bugs in a model that would otherwise go unnoticed, and may highlight inappropriate assumptions. We used to refer to this as 'accidental' verification and validation but now call it 'structured' instead. It has proven to be such an invaluable tool for identifying bugs in both model implementations and the modeling platforms themselves, that we now view the massive sensitivity analysis afforded by large-scale designed experiments as an integral part of the model development process. In the long run, we have found it is worthwhile to test early and test often.

Finally, we return to the idea of big questions. Earlier we discuss these in terms of the number and complexity of the responses. Even more important may be the questions themselves—ethics comes into play when we decide what problems to tackle. As Fig. 9.1 illustrates, 'thinking big' gives us opportunities for envisioning what could be and what should be, as we look at the many complex problems the world faces (Sanchez 2016). How can we have an impact on the big problems like climate change, global migration, defense and national security, poverty, and more? Simulation has a lot to offer in the dialog about these complex problems.

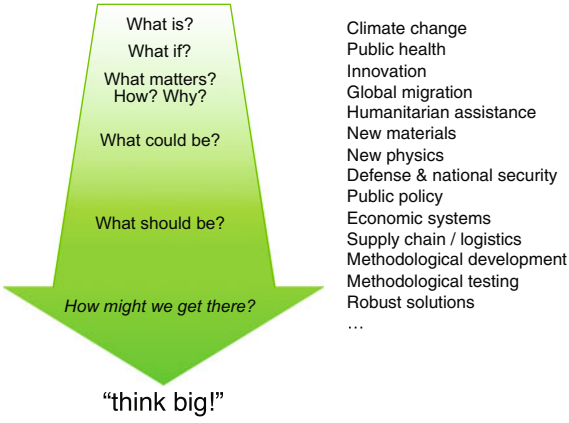


Fig. 9.1 A hierarchy of questions that we can ask of simulation models as we move to using large-scale designed experiments, along with a partial list of challenging application areas. *Source* Adapted from Sanchez (2016)

9.3.2 Leveling the Playing Field

Having large-scale experiments can help adjudicate differences in approaches that might be due to different assumptions on the part of modelers, analysts, and stakeholders. Large designed experiments can accommodate such differences as parameterizations, allowing the impact of differing assumptions to be explicitly explored. It may turn out that changes to the assumptions are damped by the system and have no tangible effects, which is useful knowledge for all parties involved. Alternatively, if different assumptions lead to meaningfully different results, this helps the modeling process by highlighting the importance of the assumptions. At a minimum, it should spur a worthwhile discussion about how the model development should proceed, and it might identify productive avenues for data collection to clarify the structure of the system.

9.4 Looking Forward

We finish as we started, by mentioning a few trends and opportunities in simulation applications, analysis methodology, and modeling methodology that we feel are ripe for further contributions by the simulation community.

On the application side, future simulation clients will continue to be faced with complex problems. Addressing these with a 'big thinking' simulation mindset helps us avoid Type III errors, and embrace computational tractability. Growing better big data via large-scale simulation experiments can lead to dramatically richer and more interesting insights. The expanding communities in data science and analytics, as well as the increased integration of computerized and computer-based decisions (such as self-driving cars or automated health diagnostics) also means the number of decision makers comfortable with simulation and simulation-driven decision making should only increase (Elmegreen et al. 2014). As the data science community develops new tools for characterizing, analyzing, and visualization large data sets, the simulation community can leverage these new approaches.

There is an abundance of opportunity in research on simulation methodology. If we view simulation studies as ongoing processes, and not end states, it is clear that methods that utilize unused computing cycles can be very beneficial. Some work has already been done on multi-objective simulation procedures, and open-ended adaptive sequential designs, but there is room for much more—particularly as researchers develop procedures that take advantage of parallel computing. Exploration and optimization techniques for adaptively updating metamodels would be of tremendous benefit for tackling big problems.

The ideas of continuous adaptation and improvement also have potential for producing major changes in how some simulation models and modeling platforms are used. Our models (or confederations of models) could draw on real-world data in real time, and use this information to evolve. Technologies such as listener event graph

objects (Buss and Sánchez 2002) demonstrate that it is possible to bridge between disparate models and data sources.

We believe that robust decision-making is an area that merits more research. A new R&S goal that explicitly seeks a "probably approximately correct selection" (Ma and Henderson 2017) aligns nicely with the robust design philosophy, and we think this is a fruitful area of exploration. We urge our simulation optimization colleagues to consider robustness as a potential optimization goal.

Smarter computational agents may also come into play on the analysis side. As we generate larger and larger model-driven data sets, we should consider a future where intelligent agents can search through the output to identify important factors and interesting features, learning as they go. A key question here is defining 'interesting' in a broader, richer way than simply 'maximum' or 'minimum.' The emerging areas of artificial intelligence and machine learning hint at great potential in this area.

We have shared some of our views about several opportunities for the simulation community. Fifty years ago there were no personal computers, no smart phones, and no email, internet, or social media. There were only a handful of simulation languages in their infancy. Could any of the first WSC attendees have envisioned our modern world? Probably not in their wildest dreams. Nevertheless, they laid the groundwork for our thriving community.

WSC is a very special conference. One reason it has been so influential is that it brings together academia, industry, and government. It fosters dialog and collaboration among people interested in modeling methodology, modeling languages, analysis methodology, and a wide variety of simulation applications. Conference contributors have a history of pushing the envelope of what is possible, and of nurturing young professionals. Our hope and expectation for the next 50 years is that the WSC community will remain vibrant, innovative, and supportive.

Acknowledgements Thanks to all our friends, students and other collaborators, and research sponsors over the years. Susan Sanchez was supported in part by the Office of Naval Research via NPS's Consortium for Robotics and Unmanned Systems Education and Research (CRUSER) initiative, and NPS's Naval Research Program. Paul Sánchez was supported in part by NPS's Naval Research Program.

References

URL https://rdrr.io/cran/FrF2/man/FrF2Large.html

URL http://CRAN.R-project.org/

Bonabeau E (2013) Big data and the bright future of simulation (the case of agent-based modeling). In: 2013 winter simulation conference keynote address

Buss AH, Sánchez PJ (2002) Building complex models with LEGOs (Listener event graph objects). In: Yucësan E, Chen C, Snowdon JL, Charnes J (eds) Proceedings of the 2002 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 732–737

Carroll AE (2017) Science needs a solution for the temptation of positive results. The Upshot, New York Times, May 29. https://www.nytimes.com/2017/05/29/upshot/science-needs-a-solution-for-the-temptation-of-positive-results.html?hpwŹrref=upshotŹaction=clickŹpgtype=HomepageŹmodule=well-regionŹregion=bottom-wellŹWT.nav=bottom-well

Cheng R (2007) Determining efficient simulation run lengths for real time decision making. In: Henderson SG, Biller B, Hsieh MH, Shortle J, Tew JD, Barton RR (eds) Proceedings of the 2007 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 138–149

Cooney B (1982) Miss Rumphius. Viking Books

Duan W, Ankenman BE, Sanchez SM, Sanchez PJ (2017) Sliced full factorial-based latin hypercube designs as a framework for a batch sequential design algorithm. Technometrics 59(1):11–22

Elmegreen BE, Sanchez SM, Szalay AS (2014) The future of computerized decision making. In: Tolk A, Diallo SY, Ryzhov O, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 943–949

SEED Center for Data Farming (2015) SEED Center. http://harvest.nps.edu. Accessed 2 July 2015

Feldkamp N, Bergmann S, Strassburger S (2015) Visual analytics of manufacturing simulation data. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 779–790

Frank AL (1968) The use of experimental design techniques in simulation. In: Reitman J, Ockene A (eds) Second conference on the applications of simulation, pp 11–12. URL http://informs-sim.org/wsc68papers/1968_0003.pdf

Hedayat AS, Sloane J, Stufken J (1999) Orthogonal arrays: theory and applications. Springer, New York

Henclewood D, Guin A, Guensler R, Hunter M, Fujimoto R (2011) Real-time data driven arterial simulation for performance measures estimation. In: Johansson B, Jain S, Montoya-Torres J, Hugan J, Yücesan E (eds) Proceedings of the 2010 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 2057–2069

Heraclitus: Fragment (circa 500 B.C.E.)

Hernandez AS, Lucas TW, Sanchez P (2012) Selecting random Latin hypercube dimensions and designs through estimation of maximum absolute pairwise correlation behavior. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Urmacher AM (eds) Proceedings of the 2012 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, New Jersey, pp 280–291

Horne GE (1999) Maneuver warfare distillations: essence not verisimilitude. In: Farrington PA, Nembhard HB, Sturrock DT, and Evans GW (eds), Proceedings of the 1999 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, New Jersey, pp 1147–1151

Horne GE Akesson B, Anderson S, Bottiger M, Britton M, Bruun R, ..., Zimmermann, A (2014) Data farming in support of NATO. Tech. Rep. TR-MSG-088. NATO Science & Technology Organization. URL https://www.cso.nato.int/Pubs/rdp.asp?RDP=STO-TR-MSG-088

HTCondor (2016). URL http://research.cs.wisc.edu/htcondor/

Huber D, Kallfass D (2015) Applying data farming for military operation planning in NATO MSG-124 using the interoperation of two simulations of different resolution. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 2535–2546

Hunter SR, Nelson BL (2017) Parallel ranking and selection. In: Tolk A, Fowler J, Shao G, Yücesan E(eds) Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences, Springer, pp 249–275

Kallfass D, Schlaak T (2012) NATO MSG-088 case study results to demonstrate the benefit of using data farming for military decision support. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Urmacher AM (eds) Proceedings of the 2012 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 2481–2492

Kang K, McDonald ML (2011) Impact of logistics on readiness and life cycle cost. In: Johansson B, Jain S, Montoya-Torres J, Hugan J, Yücesan E (eds) Proceedings of the 2012 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 1336–1346

Kim SH, Nelson BL (2006) Selecting the best system. In: Elsevier handbooks in operations research and management science: simulation, chap. 17, pp 501–534. Elsevier

Kleijnen JPC (2015) Design and analysis of simulation experiments, 2nd edn. Springer, New York

Kleijnen JPC (2017) Design and analysis of simulation experiments: tutorial. In: Tolk A, Fowler J, Shao G, Yücesan E (eds) Advances in Modeling and Simulation - Seminal Research from 50 Years of Winter Simulation Conferences, Springer, pp 138–155

Kleijnen JPC, Bettonvil B (1997) Searching for important factors in simulation models with many factors: sequential bifurcation. Eur J Oper Res 96(1):180–194

Laney D (2001) 3d data management: controlling data volume, velocity, and variety. In: Application delivery strategies, 949. META Group Inc

Law AM (2014) Simulation modeling and analysis, 5th edn. McGraw-Hill, New York

Linares MP, Montero L, Barceló J, Carmona C (2016) A simulation framework for real-time assessment of dynamic rid sharing demand responsive transportation models. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, USA, pp 2216–2227

Loeppky JL, Moore LM, Williams BJ (2010) Batch sequential designs for computer experiments. J Stat Plan Inference 140(6):1452–1464. doi:10.1016/j.jspi.2009.12.004. URL http://www.sciencedirect.com/science/article/pii/S0378375809003747

Lucas TW, Kelton WD, Sánchez PJ, Sanchez SM, Anderson BL (2015) Changing the paradigm: simulation, now a method of first resort. Naval Res Logist 62(4):293–303

Lucas TW, Sanchez SM, Martinez F, Sickinger LR, Roginski JW (2007) Defense and homeland security applications of multi-agent simulations. In: Henderson SG, Biller B, Hsieh MH, Shortle J, Tew JD, Barton RR (eds) Proceedings of the 2007 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 138–149

Luo J, Hong LJ (2011) Large-scale ranking and selection using cloud computing. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu M (eds) Proceedings of the 2011 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 4051–4061

Ma S, Henderson SG (2017) An efficient fully sequential selection procedure guaranteeing probably approximately correct selection. In: Chan WKV, D'Ambrogio A, Zacharewicz G, Mustafee N, Wainer G, Page E (eds) Proceedings of the 2017 winter simulation conference (forthcoming). Institute of Electrical and Electronic Engineers, Inc, Piscataway, NJ

MacCalman AD, Sanchez SM, McDonald ML, Goerger SR, Karl AT (2016) Tradespace analysis for multiple performance measures. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, USA, pp 3063–3074

Marlow DO, Sanchez SM, Sanchez PJ (2015) Testing aircraft fleet management policies using designed simulation experiments. In: Proceedings of the 21st international congress on modelling and simulation, pp 917–923. Modelling and Simulation Society of Australia and New Zealand Inc (MSSANZ)

Matsumoto Y Ruby programming language. https://www.ruby-lang.org/

McDonald ML, Upton SC, Seymour CN, Lucas TW, Sanchez SM, Sanchez PJ, Schramm HC, Smith JR (2014) Enhancing the analytic utility of the synthetic theater operations research model (STORM). In: Tolk A, Diallo SY, Ryzhov O, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 4136–4137

Méndez-Vázquez YM, Ramirez-Rojas KL, Cabrero-Rios M (2013) The search for experimental designs with tens of variables: preliminary results. In: Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 2654–2665

Mitroff II, Featheringham TR (1974) On systemic problem solving and the error of the third kind. Behav Sci 19(6):383–393

Moore GE (1965) Cramming more components onto integrated circuits. Electronics 38(8). URL https://drive.google.com/file/d/0By83v5TWkGjvQkpBcXJKT1I1TTA/view

Morgan BL, Schramm HC, Smith JR, Lucas TW, McDonald ML, Sanchez PJ, Sanchez SM, Upton SC (2017) Improving U.S. navy campaign analysis using big data. Interfaces (forthcoming)

Nelson BL, Swann J, Goldsman D, Song W (2001) Simple procedures for selecting the best simulated system when the number of alternatives is large. Operat Res 950–963

Page SE (2016) Many model thinking. In: Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, USA. URL http://dl.acm.org/citation.cfm?id=3042094.3042096

Ramberg JS, Sanchez SM, Sanchez PJ, Hollick LJ (1991) Designing simulation experiments: Taguchi methods and response surface metamodels. In: Nelson BL, Kelton WD, Clark GM (eds) Proceedings of the 1991 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 167–176

Robinson S, Arbez G, Birta L, Tolk A, Wagner G (2015) Conceptual modeling: definition, purpose and benefits. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 2812–2836

Sanchez PJ (2010) Data farming. In: Presentation, CENIC conference, Corporation for Education Network Initiatives in California. http://harvest.nps.edu/papers/DataFarming.mp4. Accessed 2 July 2015

Sanchez SM A data farmer's almanac. In: Titan talk at the 2016 winter simulation conference. URL https://www.informs.org/Resource-Center/Video-Library/INFORMS-Meetings-Videos/Presentations-from-Other-INFORMS-Conferences/Titan-Talk-A-Data-Farmer-s-Almanac

Sanchez SM (2015) Simulation experiments: better data, not just big data. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 800–811

Sanchez SM (2017) Data farming: reaping insights from simulation experiments. Chance (forthcoming)

Sanchez SM, Lucas TW (2002) Exploring the world of agent-based simulation: simple models, complex analyses. In: Yucësan E, Chen C, Snowdon JL, Charnes J (eds) Proceedings of the 2002 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 116–126

Sanchez SM, Lucas TW, Sanchez PJ, Nannini CJ, Wan H (2012) Designs for large-scale simulation experiments, with applications to defense and homeland security. In: K. Hinkelmann (ed) Design and analysis of experiments: special designs and applications, vol 3, 1st edn, chap 12. Wiley, New York pp 413–441

Sanchez SM, Sanchez PJ (2005) Very large fractional factorial and central composite designs. ACM Trans Model Comput Simul 15(4):362–377

Sanchez SM, Sanchez PJ, Wan H (2014) Simulation experiments: better insights by design. In: Proceedings of the 2014 summer simulation conference. The Society for Modeling & Simulation International, San Diego, California

Sanchez SM, Wan H (2015) Simulation experiments: better data, not just big data. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, New Jersey

SAS Corporation: Statistical Software | JMP Software from SAS. https://www.jmp.com/

Savage SL, Thibault JM (2015) Towards a simulation network or the medium is the monte carlo (with apologies to marshall mcluhan). In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 4126–4133

Seichter S, Horne G (2014) Data farming in support of NATO operations—methodology and proof-of-concept. In: Tolk A, Diallo SD, Ryzhov O, Yilmaz L, Buckley S, Miller JA (eds) Proceedings

of the 2014 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 2355–2363
Shen H, Wan H, Sanchez SM (2009) A hybrid method for simulation factor screening. Naval Res Logist 57:45–57
Song E, Nelson BL (2017) Input model risk. In: Tolk A, Fowler J, Shao G, Yücesan E (eds) Advances in modeling and simulation—seminal research from 50 Years of winter simulation conferences. Springer, pp 63–80
Stat-Ease: Design-Expert version 10 software. http://www.statease.com/dx10.html
Suzumura T, Kanezashi H (2014) Multi-modal traffic simulation platform on parallel and distributed systems. In: Tolk A, Diallo SD, Ryzhov O, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 769–780
Taguchi G (1987) System of experimental design, vol 1, 2. UNIPUB/Krauss International, White Plains, NY
Tan WC, Haas PJ, Mak RL, Kieliszewski CA, Selinger PG, Magio PP, Glissman S, Cefkn M, Li Y (2012) Splash: a platform for analysis and simulation of health. In: Proceedings of the 2nd ACM SIGHIT symposium on international health informatics. ACM, pp 543–552
Taylor SJD, Brailsford S, Chick SE, L'Ecuyer PL, Macal CM, Nelson BL (2013) Modeling and simulation grand challenges: an or/ms perspective. In: Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 1269–1282
Thibault JM (2014) Standard specification for stochastic information packets (SIPs) and stochastic library units with relationships preserved (SLURPs) version 2.0. URL http://probabilitymanagement.org/library/SIP-Standard-Version2.pdf
Tolk A (2017) Code of ethics. The profession of modeling and simulation: discipline. Ethics, education, vocation, societies, and economics. Wiley, Hoboken, NJ, pp 35–51
Tufte ER (1986) The visual display of quantitative information. Graphics Press, Cheshire, CT, USA
University of Michigan. URL https://www.coursera.org/learn/model-thinking
Vahdatikhaki F, Setayeshgar S, Hammad A (2013) Location-aware real-time simulation framework for earthmoving projects using automated machine guidance. In: Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 3086–3097
van Rossum G Python programming language. https://www.python.org
Vieira H, Sanchez SM, Kienitz KH, Belderrain MCN (2011) Improved efficient, nearly orthogonal, nearly balanced mixed designs. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu M (eds) Proceedings of the 2011 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 3605–3616
Vieira H Jr, Sanchez SM, Kienitz KHK, Belderrain MCN (2013) Efficient, nearly orthogonal-and-balanced, mixed designs: an effective way to conduct trade-off analyses via simulation. J Simul 7(4):264–275
Vieira H Jr, Sanchez SM, Sanchez PJ, Kienitz KHK, Belderrain MCN (2014) A restricted multinomial hybrid selection procedure. ACM Trans Model Comput Simul 2(10):1–10, 24
Vigen T (2014) Spurious correlations. http://www.tylervigen.com. Accessed 2 July 2015
Vigen T (2015) Spurious correlations. Hachette Book Group, New York, NY
Wagner H (1969) Principles of operations research with applications to managerial decisions. Prentice-Hall, Englewood Cliffs, NJ
Wan H, Ankenman B, Nelson B (2006) Controlled sequential bifurcation: a new factor-screening method for discrete-event simulation. Oper Res 54:743–755

Chapter 10
Bayesian Belief Models in Simulation-Based Decision-Making

Ilya O. Ryzhov and Ye Chen

Abstract We present an overview of Bayesian statistical models and their use in simulation-based optimization. Bayesian schemes are valuable for their ability to model our beliefs about an uncertain environment (for example, the unknown output distribution of a complex simulation), as well as the evolution of these beliefs over time as information is acquired through simulation. With this ability, we can make adaptive decisions that improve over time and anticipate the effect of new information before it is observed. We discuss two broad classes of such adaptive algorithms, show how they interface with the underlying Bayesian statistical models, and summarize the comparative advantages of each algorithmic approach. We also discuss how approximate Bayesian models can be designed to retain the advantages of adaptive learning in problems where the observations are censored or incomplete.

10.1 Introduction

Simulation-based optimization (Fu 2015) is the problem of optimizing a complex system under the condition that the performance of that system can only be observed by running simulation experiments. This type of optimization differs from classic mathematical programming in that the objective function cannot be expressed in closed form. Rather, we first choose a solution (a set of values for the decision variables), then acquire information (possibly with stochastic noise) about the performance of that solution. Information may be extremely noisy, or expensive to collect (for example, testing a single solution inside the simulator may require a large amount of machine time), and the set of feasible solutions may be very large or even

I.O. Ryzhov (✉)
Robert H. Smith School of Business, University of Maryland,
College Park, MD 20742, USA
e-mail: iryzhov@rhsmith.umd.edu

Y. Chen
Department of Mathematics, University of Maryland,
College Park, MD 20742, USA
e-mail: yechen@math.umd.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_10

uncountably infinite. Even if the number of possible solutions is relatively small, our simulation budget may be even smaller; in some cases, instead of running a simulator, we might even collect information from field experiments that incur economic costs. Some examples of applications include the following:

- *Clinical trials*. Clinical trials often involve a large number of potential alternatives (e.g., medical treatments) and a high cost of collecting information, which may be measured in dollars or time delay (Chick et al. 2015).
- *Revenue management*. Qu et al. (2013) considered the problem of pricing high-volume business-to-business contracts. The *demand curve* that governs buyer response to different prices is unknown and must be inferred based on buyer reactions to past prices.
- *Service operations*. Arlotto et al. (2010) studied a hiring and retention problem in which the employer must balance the cost of hiring a new worker (and the possibility that the new worker may not perform well) against the past performance of current employees.
- *Marketing*. Firms use experiments to identify optimal advertising strategies for different segments of the customer population; different strategies may have segment-specific effects (Han et al. 2013).
- *Manufacturing*. Simulation is widely used to compare candidate configurations for a production system, or the utilization of employees resulting from different assignments (April et al. 2006).

In all of these problems, not all of the feasible solutions are equally important. If we are given a very limited number of opportunities to collect information, we may prefer to spend them on a small number of "promising" decisions rather than attempting to learn about everything at once. This gives rise to the fundamental *exploration/exploitation* tradeoff: we have to choose between learning more about a decision that already appears to be good (exploitation), or we may risk spending part or all of the budget on a decision that we know nothing about (exploration). In the former case, we may fail to discover a decision that was much better than we believed; in the latter case, we might simply confirm that the uncertain decisions are not competitive after all. To make this tradeoff in a principled manner, we first require *statistical* models to represent our knowledge of the system and capture how this knowledge changes over time in response to new information; we then require an *optimization* framework in which these models of uncertainty may be used to quantify the value of learning about various decisions.

The simulation community has been the home of much of the seminal research in this area. Over the years, the WSC proceedings have featured many surveys and tutorials on simulation optimization: see Kim and Nelson (2007), Hong and Nelson (2009), Chau et al. (2014), or Jian and Henderson (2015) for examples. See also Fu et al. (2000) or Fu et al. (2014) for examples of panel discussions of the field that have taken place at the WSC. The goal of this chapter, however, is not to summarize all of this work, but rather to highlight one particular statistical approach—namely, the use of adaptive Bayesian models—that has consistently shown its usefulness in simulation optimization. This more specialized area has also been the subject of

multiple WSC surveys and tutorials in the past (Chick 2000, 2004, 2006a), but there have also been some significant theoretical and methodological breakthroughs in very recent years.

The main distinguishing feature of Bayesian models is that they directly quantify and integrate our uncertainty about our estimates of various problem parameters (for example, the means of the output distributions of our simulations). Every parameter is treated as a random variable, and the distribution of this random variable is a distribution of *belief*. In other words, the model allows that our estimates (based on past data) may be inaccurate and provides an explicit value for the likelihood that any other quantity (potentially one that is very different from our estimate) may be the "correct" one. Thus, the Bayesian model consists not only of an estimate but also of an entire distribution, although in many standard models this distribution may be compactly represented by a small number of parameters. This distribution allows us to make detailed probabilistic forecasts about unknown values: for example, we may calculate the probability that the "true" unknown value is significantly higher (or lower) than our estimate, or we may calculate expected costs (incurred, e.g., by conducting an experiment) over the distribution of belief.

Essentially, the dimensions of the exploration/exploitation tradeoff are directly built into the Bayesian model. Exploitation is handled by a point estimate of the value of a decision, such as a sample mean; in many cases this estimate is calculated in exactly the same way as in classical statistics. Exploration, however, is handled by the Bayesian concept of uncertainty, which now gives us a precise way to compare two decisions based on their respective likelihood of being substantially better than the one currently believed to be the best. For these reasons, researchers have developed an entire class of *optimization* algorithms that leverage the features unique to Bayesian models to guide the acquisition of information. There are three important dimensions for the study of these algorithms: theoretical, computational, and empirical.

In this chapter, we primarily focus on computational and theoretical aspects. Our first goal is to demonstrate the computational flexibility of the Bayesian approach. One of the most attractive aspects of Bayesian modeling (as well as Bayesian optimization) is its ability to generalize to various classes of statistical and decision problems. Section 10.2 presents an overview of three standard Bayesian models. The first of these is a simple textbook model for learning the unknown mean of a normal distribution based on noisy observations. Overall, this model is not too different from classical frequentist statistics. The second model, however, uses the Bayesian notion of "correlated beliefs" to learn about a *vector* of unknown values from *scalar* observations of individual components. Correlated beliefs have enormous practical potential, as they enable us to learn about large decision spaces from very small numbers of observations, and this particular form of correlation appears to be a purely Bayesian concept, with no natural analog in frequentist statistics. Finally, the third model shows how Bayesian concepts may be integrated into linear regression (the setting of many practical applications) with no substantial addition in computational or storage cost.

Section 10.3 gives examples of *decision* problems where these and other Bayesian models may be applied. While Sect. 10.2 focuses purely on statistical estimation, Sect. 10.3 places estimation in the context of *economic* problems where the goal is to identify an optimal decision from a feasible set. For illustrative purposes, we mainly focus on discrete simulation optimization (Hong et al. 2015), particularly on the widely studied ranking and selection problem (Chen et al. 2015), but we give an example of a regression-based optimization problem with a more sophisticated optimization aspect. Then, Sect. 10.4 introduces the concept of *Bayesian optimization*, which combines the uncertainty in the statistical model with the structure of the optimization problem to produce promising decisions for exploration. We cover two popular Bayesian methodologies, Thompson sampling (Russo and Van Roy 2014) and value of information (Chick 2006b), both of which can be generalized to many classes of decision models beyond ranking and selection. The guiding principles of each methodology are presented in the context of the examples in Sects. 10.2–10.3.

Our second goal in this chapter is to highlight several recent breakthroughs in Bayesian simulation optimization. Section 10.4.3 describes very recent advances in the theoretical analysis of Bayesian optimization for ranking and selection. Much of this material has appeared within the past 1–2 years and offers many new research directions. Section 10.5 describes approximate Bayesian inference, a methodology for statistical learning in problems where information is incomplete or censored; this methodology has exhibited great practical potential in recent years, and a full theoretical understanding is only now being developed.

This chapter is not an exhaustive survey; our goal is rather to present several fundamental concepts, hint at how these concepts generalize beyond their most basic forms, and showcase some more advanced topics. For example, in our discussion of decision models in Sect. 10.3, we sketch out several additional extensions with references, which may hopefully be a useful starting point for interested readers. Here, it may be useful to briefly discuss what is *not* covered in this chapter. Due to space considerations, we have chosen not to delve into the vast literature on Bayesian *global* optimization beyond a few references (Jones et al. 1998; Staum 2009). We are also not able to go into Bayesian input uncertainty (Xie et al. 2014), an interesting emerging area in the simulation literature. We do not extensively discuss simulation optimization outside the Bayesian context, leaving out the considerable literature on indifference-zone selection (Kim and Nelson 2001; Hong and Nelson 2005), optimal computing budget allocation (Chen et al. 2000; Chen and Lee 2010), or other recent frequentist methods (Fan et al. 2016); thus, we also do not conduct detailed comparisons between Bayesian and non-Bayesian methods, preferring instead to focus on an exposition of those characteristics of Bayesian methods that (in our opinion) make them valuable for simulation-based decision-making. Finally, the problems we have chosen to present resemble the types of problems most frequently encountered in the simulation community; thus, for example, we do not go into the literature on multi-armed bandits (Gittins et al. 2011) or design of experiments (Zhang and Qian 2013), whether those problems are Bayesian or not, although these areas have often considered very similar challenges.

10.2 Bayesian Learning

In this section, we summarize three Bayesian learning models and contrast them with classical frequentist statistics. There is no optimization in this section; decision problems are considered in Sect. 10.3. However, the majority of existing research in Bayesian simulation optimization relies on one of these standard statistical models. Derivations can be found in DeGroot (1970), Powell and Ryzhov (2012) or other standard references.

10.2.1 Learning the Mean of a Normal Distribution

Suppose that we are able to collect noisy samples $W^1, W^2, \ldots$ drawn independently from the *sampling distribution* $\mathcal{N}\left(\mu, \lambda^2\right)$. For simplicity, suppose λ is known; the only problem is to estimate the unknown mean μ. In traditional (frequentist) statistics, one would typically use the sample mean

$$\theta^n = \frac{1}{n}\sum_{m=1}^{n} W^m, \tag{10.1}$$

which in this setting has the distribution $\mathcal{N}\left(\mu, \frac{\lambda^2}{n}\right)$ and is known to have various desirable optimality properties (Bickel and Doksum 2015).

In Bayesian statistics, any unknown quantity (in this case, μ) is treated as a random variable. The distribution of μ has a different interpretation from the sampling distribution: it represents our *prior knowledge* about μ. For example, suppose that we impose the prior distribution $\mu \sim \mathcal{N}\left(\theta, \sigma^2\right)$, where θ, σ are constants that we choose. In doing so, we are saying that (1) we believe that μ is equal to θ; (2) we allow the possibility that μ is different (perhaps very different) from θ, but we believe that values closer to θ are more likely.

If we have no knowledge whatsoever about μ, we may pick an arbitrary value for θ while setting $\sigma = \infty$. However, in a practical application, it is more likely that we can at least give a rough range of values where μ is likely to fall. To help us choose the parameters of the prior distribution, we could use the standard interpretation that the interval $\theta \pm 2\sigma$ contains μ with 95% confidence.

With these Bayesian assumptions, the sampling distribution now requires a new interpretation: the statement $W \sim \mathcal{N}\left(\mu, \lambda^2\right)$ is *conditional*, i.e., given μ, the sample is conditionally normal with mean μ and variance λ^2. The Bayesian model thus gives us two probability densities:

$$P\left(W \in dw \mid \mu = x\right) = \frac{1}{\sqrt{2\pi\lambda^2}} e^{-\frac{(w-x)^2}{2\lambda^2}} dw, \tag{10.2}$$

$$P(\mu \in dx) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\theta)^2}{2\sigma^2}} dx. \tag{10.3}$$

We can now apply Bayes' rule to write

$$P(\mu \in dx \mid W = w) \propto P(\mu \in dx) \cdot P(W \in dw \mid \mu = x). \tag{10.4}$$

Using (10.2)–(10.3) and a lot of tedious algebra, we can derive an explicit expression for the right-hand side of (10.4) in terms of x. This expression characterizes the *posterior distribution* of μ given the event that $W = w$. Since the distribution of μ is an object of belief, (10.4) represents how our beliefs about μ have evolved as a result of a single observation.

In fact, for the model presented here, (10.4) turns out to be a normal density with parameters

$$\theta' = \frac{\sigma^{-2}\theta + \lambda^{-2}w}{\sigma^{-2} + \lambda^{-2}}, \tag{10.5}$$

$$\sigma' = \left(\sigma^{-2} + \lambda^{-2}\right)^{-\frac{1}{2}}. \tag{10.6}$$

It is easy to see that (10.5) is a weighted average of the prior estimate θ and the new observation w, with more weight placed on the observation when λ is small relative to σ, i.e., our confidence in our prior estimate is weak compared to the level of noise in the sampling distribution. Our uncertainty about μ, represented by σ', will decrease relative to the prior uncertainty σ; again, the reduction in uncertainty is greater if λ is smaller (that is, if the samples are more accurate).

The fact that (10.4) is a normal density is particularly useful for *sequential* learning. Since the posterior distribution is normal, we can now treat it as the prior and collect a second observation. The new posterior distribution will still be normal; we will just need to apply (10.5)–(10.6) a second time. In general, when we start with the prior distribution $\mu \sim \mathcal{N}\left(\theta^0, \left(\sigma^0\right)^2\right)$, the posterior distribution of μ, given $W^1, ..., W^n$ for any n, is normal with parameters θ^n, σ^n, which are updated recursively as

$$\theta^n = \frac{\left(\sigma^{n-1}\right)^{-2}\theta^{n-1} + \lambda^{-2}W^n}{\left(\sigma^{n-1}\right)^{-2} + \lambda^{-2}}, \tag{10.7}$$

$$\sigma^n = \left(\left(\sigma^{n-1}\right)^{-2} + \lambda^{-2}\right)^{-\frac{1}{2}}. \tag{10.8}$$

As a result, we never have to store the density of μ in memory; since it is always normal, we only need to store two parameters, which are trivial to update. This property, when the posterior belongs to the same distributional family as the prior, is called *conjugacy*. The overwhelming majority of research in Bayesian simulation optimization relies on simple conjugate models, usually involving normal distributions.

In the special case where $\sigma^0 = \infty$ (recall that this models the case where we have absolutely no prior knowledge about μ), (10.7) reduces to (10.1), and $\sigma^n = \frac{\lambda}{\sqrt{n}}$ is the usual frequentist variance of the sample mean. This fact is useful for intuitive understanding of the Bayesian model: it shows that our *estimation* of the unknown value proceeds in the same way that it always would. However, we now turn to a different setting where Bayesian beliefs offer more power with regard to estimation.

10.2.2 Learning with Correlated Beliefs

Suppose now that μ is an M-vector, i.e., we have M unknown values to learn rather than just one. In this case, we might impose a multivariate normal prior distribution $\mathcal{N}_M(\theta, \Sigma)$. The prior mean θ is now an M-vector as well; furthermore, we now have to specify an $M \times M$ matrix Σ. The diagonal entries of Σ, as before, represent our uncertainty about the individual unknown values μ_x, $x \in \{1, ..., M\}$. The off-diagonal entries represent "correlated beliefs." Essentially, we believe that there is some relationship between the unknown values; as a result, anything that we can learn about μ_x also provides information about other values μ_y for $y \neq x$.

We first give the technical details of the learning process and then discuss interpretation. Suppose that we observe $W_x \sim \mathcal{N}\left(\mu_x, \lambda^2\right)$, where λ is known as before. Note that W_x is a scalar observation: it only provides a sample of μ_x by itself. However, after applying Bayes' rule, one will find that the posterior distribution of μ given W_x is still multivariate normal with parameters

$$\theta' = \theta + \frac{W_x - \theta_x}{\lambda^2 + \Sigma_{xx}} \Sigma e_x, \tag{10.9}$$

$$\Sigma' = \Sigma - \frac{\Sigma e_x e_x^\top \Sigma}{\lambda^2 + \Sigma_{xx}}, \tag{10.10}$$

where e_x denotes a vector of zeroes with only the xth component equal to 1. The vector Σe_x is simply the xth column of Σ, containing the prior variance of μ_x and also all the prior covariances between μ_x and other unknown values. If some or all of these covariances are nonzero, (10.9) will update multiple components of θ, even though we have observed only one. More specifically, suppose that $W_x > \theta_x$; in this case, we should increase our beliefs about μ_x (as in the previous model), but we should also increase our beliefs about μ_y for any y that is positively correlated with x. Conversely, we should decrease θ_y if y is negatively correlated with x.

Unlike the model in Sect. 10.2.1, in which the posterior mean was more or less identical to the usual sample mean, the model here offers a new and different way of learning. There is no analog of this model in classical frequentist statistics, because there are no correlations in the *sampling* distribution: we collect independent, scalar observations of individual components of μ. Correlations in this model are purely an object of belief: $\Sigma_{xy} > 0$ implies that we believe μ_x and μ_y to be "similar" in some

way, while $\Sigma_{xy} < 0$ means that we believe these values to be "different." Depending on the covariance relationship, learning about μ_x will implicitly provide different types of information about μ_y.

Consider the following example. A delivery service wishes to determine its fleet size (total number of trucks) for the coming year. Managers believe that this number should be somewhere between 50 and 100 trucks. Given a fleet size $x \in \{50, ..., 100\}$, let μ_x be the average daily profit (revenue minus maintenance cost) earned by the fleet. For any fixed x, we can set up a set of vehicle routing problems (VRPs) with randomly generated demands. These problems can be solved to near-optimality and the reported profits can be averaged to obtain a single observation W_x. The noise level λ depends on the stochasticity of the demand distribution; potentially, there could be a great deal of variation between individual instances of the VRP. We may not have the time to exhaustively simulate sufficiently many observations to learn each μ_x with a high degree of accuracy.

However, correlated beliefs may help. The profits μ_{50} and μ_{51}, for 50 and 51 trucks respectively, are likely to be similar, or at least, it seems reasonable to suppose that there is more similarity between them than between μ_{50} and μ_{100}. Thus, we can impose the "distance-based" covariance structure

$$\Sigma_{xy} = K \exp^{-\alpha(x-y)^2},$$

where K and α are user-specified parameters. According to this choice of prior distribution, two fleet sizes are more likely to have "similar" profits if the fleet sizes themselves are similar. Thus, collecting one observation W_{50} should provide the most information about μ_{50}, a bit less about μ_{51}, μ_{52} etc., and very little about μ_{100}. This greatly increases the power of each individual observation: now, we may learn quite a bit about a particular μ_x without ever actually having observed component x directly.

There is a price to pay for this added power: we now have to store and update an $M \times M$ covariance matrix. This can be difficult: on one hand, correlated beliefs are the most valuable when the problem size is large relative to the number of observations, and yet the cost of storing and updating Σ grows quadratically in the problem size. Recent work (Salemi et al. 2014) has investigated Gaussian Markov models that can reduce this cost by enforcing a certain sparsity structure in Σ.

10.2.3 Learning in Linear Regression

The correlated Bayesian model from Sect. 10.2.2 also ties into a Bayesian version of linear regression. Suppose now that, rather than observing an individual component of μ, we collect a sample of the form

$$W = \mu^\top \phi + \varepsilon,$$

where $\phi \in \mathbb{R}^M$ is some fixed M-vector, and $\varepsilon \sim \mathcal{N}\left(0, \lambda^2\right)$ is an independent residual error. Supposing that $\mu \sim \mathcal{N}(\theta, \Sigma)$ as before, the posterior distribution of μ given W is again multivariate normal with parameters

$$\theta' = \theta + \frac{W - \theta^\top \phi}{\lambda^2 + \phi^\top \Sigma \phi} \Sigma \phi, \tag{10.11}$$

$$\Sigma' = \Sigma - \frac{\Sigma \phi \phi^\top \Sigma}{\lambda^2 + \phi^\top \Sigma \phi}. \tag{10.12}$$

Equations (10.11)–(10.12) are identical to (10.9)–(10.10) with e_x replaced by the vector ϕ. The regression model obviates the need for an in-depth characterization of the prior covariance structure; even if Σ is diagonal, (10.11)–(10.12) will quickly populate the off-diagonal entries with empirical covariances as long as ϕ has multiple nonzero entries.

In Bayesian simulation optimization, the regression model often helps with dimension reduction. For example, suppose that $\phi \in \{0, 1\}^M$ is a binary vector (representing the presence or absence of certain characteristics). Then, $\mu^\top \phi$ may have a different value for each possible ϕ. The problem could thus have 2^M distinct unknown values to be learned; attempting to model each value individually (as was done in Sect. 10.2.2) would quickly become computationally burdensome since we would then have to store a matrix of size $2^M \times 2^M$. However, the assumption of linear structure allows us to get by with an $M \times M$ covariance matrix.

By now, the reader may be concerned about the recurring assumption that the variance λ^2 of the sampling distribution is known. In practice, this quantity may be more difficult to estimate than the mean, and it is quite unlikely that we would know it exactly. In fact, it is possible to extend all of the models in this section to handle unknown variance, but due to space considerations, we only describe this extension for the regression model, as it is the most flexible and powerful of the three models we have discussed. In this extension (first studied in detail by Han et al. 2013, 2016), both μ and λ^2 are unknown, and the Bayesian model imposes a joint prior distribution on the pair $\left(\mu, \lambda^{-2}\right)$.

This joint distribution is characterized as follows. First, the marginal distribution of λ^{-2} is assumed to be *Gamma* (a, b); then, the conditional distribution of μ given λ is assumed to be multivariate normal with mean vector θ and covariance matrix $\lambda^2 \Sigma$. It can be shown that, under this model, the marginal distribution of μ is a multivariate Student's t-distribution (Kotz and Nadarajah 2004), in line with the well-known approach from classical statistics of using the t-distribution to model normal samples with unknown variance. Given an observation W, the posterior joint distribution of $\left(\mu, \lambda^{-2}\right)$ has the same structure (known as "multivariate normal-gamma") with the updated parameters

$$\theta' = \theta + \frac{W - \theta^\top \phi}{1 + \phi^\top \Sigma \phi} \Sigma \phi, \tag{10.13}$$

$$\Sigma' = \Sigma - \frac{\Sigma\phi\phi^\top\Sigma}{1+\phi^\top\Sigma\phi}, \tag{10.14}$$

$$a' = a + \frac{1}{2}, \tag{10.15}$$

$$b' = b + \frac{\left(W - \theta^\top\phi\right)^2}{2\left(1+\phi^\top\Sigma\phi\right)}. \tag{10.16}$$

Modeling the unknown residual variance is thus fairly straightforward, requiring only two additional scalar parameters. Equations (10.13)–(10.14) are identical to the well-known recursive least squares update (Powell 2011), reaffirming our intuition that the Bayesian model estimates the regression coefficients in largely the same way as the classical model.

There are other potentially useful Bayesian models with the conjugacy property. For example, DeGroot (1970) surveys a number of models in which the sampling distribution is non-normal; however, these models generally are not able to handle correlated beliefs as conveniently as normal models. It is also possible to extend the models in Sects. 10.2.2–10.2.3 to learn continuous functions rather than finite-dimensional vectors. Thus, we may let μ be such a function, and suppose that we can observe scalar values $\mu(x)$ for fixed x, either with or without stochastic noise. The Bayesian framework of Gaussian process regression (Rasmussen and Williams 2006), known by the name "stochastic kriging" in the simulation literature (Huang et al. 2006; Kleijnen 2009; Ankenman et al. 2010), can then be used to learn μ based on sequential scalar observations.

10.3 Decision Models

We now illustrate how learning may be integrated with optimization in the context of two decision models from the simulation literature. As is common in this literature, we consider "offline learning" problems, meaning that a simulation model is used to collect information for a period of time, and subsequently a single optimization problem is solved using all the information that has been acquired. The solution returned by this problem is then selected for "implementation" in some application outside the simulator.

Our intention here is to illustrate the main issues and tradeoffs that arise in simulation-based optimization, and so we do not present an exhaustive survey of decision models. To name some examples that are not covered: the simulation community (Fu et al. 2008; Staum 2009; Kim and Zhang 2010; Scott et al. 2010; Zhang et al. 2011) has put a great deal of effort into global optimization problems, also known as "derivative-free" or "black-box optimization" (Conn et al. 2009), in which the solution space is continuous. The computer science community typically focuses on "online learning," where statistical learning and optimization occur

simultaneously; the multi-armed bandit problem (Gittins et al. 2011) is perhaps the best-known example of an online problem. However, many of the issues discussed in this section also apply in these other settings.

10.3.1 Ranking and Selection

The ranking and selection (R&S) problem is quite simple to describe. Let μ_x be an unknown performance value for each $x \in \{1, ..., M\}$. The goal is to discover

$$x^* = \arg\max_x \mu_x.$$

The typical interpretation is that the index x denotes competing *alternatives*; for example, these may be differently calibrated simulation models, or simulation models of different processes, e.g., different assignments of agents to call centers. The expected performance μ_x of alternative x cannot be expressed in closed form, but may be estimated from simulation runs whose output is a sample of the form $W_x \sim \mathcal{N}\left(\mu_x, \lambda_x^2\right)$. We may use one of the models in Sects. 10.2.1–10.2.2 to update our beliefs about the various alternatives as new information comes in.

If we had the ability to run very large numbers of simulations for each x, this problem would not be very challenging: as we saw, the posterior mean of μ_x behaves like the sample mean, so the law of large numbers applies, meaning that eventually $\arg\max_x \theta_x^n$ will be identical to x^*. Suppose, however, that we do not have this ability, for example, because a single simulation run may take a week of machine time. Instead, we are limited to a small number N of simulation runs. Furthermore, only one alternative may be simulated at a time; thus, assigning a simulation run to alternative x means that we have given up the chance to learn something about $y \neq x$. This is the main tradeoff of the R&S problem: how many runs should be assigned to each x in order to identify x^* as efficiently (in some sense to be defined) as possible?

First, let us formalize the learning process using the setting of Sect. 10.2.1 as an example. For each x, we begin with the prior distribution $\mu_x \sim \mathcal{N}\left(\theta_x^0, \left(\sigma_x^0\right)^2\right)$. We assume that μ_x and μ_y are independent for any $x \neq y$; furthermore, we assume that all simulation runs produce independent output. From this it follows that simulating x will never provide any information about $y \neq x$. Simulations will be conducted one by one; denote by $\{x^n\}_{n=0}^{N-1}$ the sequence of alternatives to which these runs are assigned. Then, (10.7)–(10.8) become

$$\theta_x^n = \begin{cases} \frac{\left(\sigma_x^{n-1}\right)^{-2}\theta_x^{n-1} + \lambda_x^{-2} W_x^n}{\left(\sigma_x^{n-1}\right)^{-2} + \lambda_x^{-2}} & \text{if } x^{n-1} = x \\ \theta_x^{n-1} & \text{if } x^{n-1} \neq x, \end{cases} \tag{10.17}$$

and

$$\left(\sigma_x^n\right)^{-2} = \begin{cases} \left(\sigma_x^{n-1}\right)^{-2} + \lambda_x^{-2} & \text{if } x^{n-1} = x \\ \left(\sigma_x^{n-1}\right)^{-2} & \text{if } x^{n-1} \neq x. \end{cases} \tag{10.18}$$

This is quite intuitive: if the next run is not assigned to y, we should not update our beliefs about μ_y. Due to conjugacy, our beliefs after n simulations are completely characterized by the pair $K^n = (\theta^n, \sigma^n)$ of posterior means and variances. We may refer to K^n as the *state of knowledge* available to us at time n.

The main reason to learn in this sequential manner is because, in this way, we can use the outcomes of past simulations to guide the allocation of future ones. Thus, we assume that each x^n is chosen *adaptively* based on the state of knowledge K^n. More formally, we write $x^n = X^\pi(\theta^n, \sigma^n)$ where X^π is an *allocation rule* mapping a knowledge state (θ^n, σ^n) to an alternative $x^n \in \{1, ..., M\}$; the superscript π is used to distinguish between different allocation rules. Now, instead of choosing the sequence $\{x^n\}$, our task is to choose an allocation rule π. An example of a somewhat trivial allocation rule is the greedy policy

$$X^G(\theta^n, \sigma^n) = \arg\max_x \theta_x^n,$$

which simply ignores the posterior variances and assigns the next simulation to the alternative that seems to be the best based solely on the current posterior means. Note that even this simple rule is adaptive: we do not know exactly which alternative will be chosen at time n until θ^n actually becomes known. However, this rule will most likely be unsatisfactory since it over-relies on the posterior means, and does not account for our uncertainty (potential for error) about these estimates. One might then consider a simple modification, such as

$$X^{IE}(\theta^n, \sigma^n) = \arg\max_x \theta_x^n + \kappa \cdot \sigma_x^n.$$

This rule, known as "interval estimation" (Kaelbling 1993), places more value on alternatives with high posterior uncertainty, because these are the alternatives that are more likely to be much better than we think. Of course, this rule does not really alleviate the difficulty of trading off between the estimated means θ_x^n and the uncertainty; it is still up to us to choose the weight κ in the IE calculation.

In general, how do we know which rule to choose? The performance of an allocation rule depends on how well it helps us to discover x^*. However, the same rule may perform quite differently in two independent sets of simulation runs, due to the stochastic noise in the samples $\left\{W_{x^n}^{n+1}\right\}_{n=0}^{N-1}$. We may be sampling the "right" alternatives and getting "unlucky" samples that mislead us as to their real values. Furthermore, since the allocation rule itself is adaptive, unlucky values of (θ^n, σ^n) may actually misdirect the rule toward assigning more samples to poor alternatives. Finally, in the Bayesian setting, the optimal alternative x^* is itself a random variable. We require a

formal objective function that addresses these various issues; the standard choice in the literature is

$$\sup_{\pi} P^{\pi}\left(\arg\max_{x} \theta_x^N = x^*\right), \tag{10.19}$$

known as the "probability of correct selection" or PCS. Equation (10.19) seems simple, but there are subtleties that merit a detailed interpretation:

1. The event $\left\{\arg\max_x \theta_x^N = x^*\right\}$ occurs randomly, because both the sequence $\{x^n\}$ of simulated alternatives and the final posterior mean vector θ^N depend on the random simulation output, and also because x^* is a random variable (in the Bayesian model);
2. The allocation decisions x^n made by the rule π influence the distribution of θ^N, but have no other impact on our evaluation of the policy. Thus, there is nothing wrong with sampling extremely poor alternatives if, by doing this, the event that $\arg\max_x \theta_x^N = x^*$ becomes more likely;
3. The theoretically optimal policy is the one with the highest PCS. It is not possible to guarantee that we will always find x^*, and even the optimal policy may fail to do so. However, a better policy is more likely to find x^* on average.

A second possible objective, which often appears in the literature on Bayesian methods, is given by

$$\inf_{\pi} \mathbb{E}^{\pi}\left(\max_{x} \mu_x - \mu_{\arg\max_x \theta_x^N}\right). \tag{10.20}$$

The idea behind this objective (known as the "expected opportunity cost" or EOC) is that, even if we do not identify the best alternative, we may be satisfied with a suboptimal one whose value is close to the best. Such suboptimal alternatives should be valued more highly than others whose values are poor. However, recent work by Gao et al. (2017) has shown a form of asymptotic equivalence between (10.19) and (10.20), suggesting that the choice between the two may be largely cosmetic, at least from the point of view of theoretical analysis.

With this, the model is now complete, and we are free to focus on the problem of choosing an efficient allocation rule π. This problem is the main focus of the vast literature on R&S, and we will examine two prominent Bayesian approaches in Sect. 10.4.

10.3.2 Learning in a Linear Model

We briefly touch on one generalization of R&S that may be particularly useful in practical applications. The goal is once more to discover an optimal decision x^*, but this quantity is now redefined as

$$x^* = \arg\max_{x \in \mathscr{X}} \mu^\top x. \tag{10.21}$$

Again, μ is an M-vector of unknown values, to be learned using Bayesian statistics. The set $\mathscr{X}$ may be a polyhedron, in which case (10.21) is simply the optimal solution to a linear program with unknown objective coefficients (Ryzhov and Powell 2012). Alternately, $\mathscr{X}$ may be a large discrete set; for example, it may be the feasible region of an integer program (Han et al. 2016).

In both cases, the linear objective function is used to simplify the learning process and reduce computational and storage costs. For example, suppose that $\mathscr{X} = \{0,1\}^M$. Then, the set $\mathscr{X}$ of possible "alternatives" can be very large, but the size of the statistical model is quite small; recall from Sect. 10.2.3 that we only need to store and update an M-vector and an $M \times M$ matrix in order to completely characterize our beliefs at any given moment. Because our distribution of belief about μ is multivariate normal, this automatically induces a normal belief about the value of any feasible alternative. That is, if $\mu \sim \mathscr{N}_M(\theta, \Sigma)$, then for any fixed $x \in \mathscr{X}$, the value $\mu^\top x \sim \mathscr{N}\left(\theta^\top x, x^\top \Sigma x\right)$.

From here, we proceed similarly to Sect. 10.3.1. Suppose that each allocation decision x^n is a vector in $\mathscr{X}$, and our observation takes the form

$$W_{x^n}^{n+1} = \mu^\top x^n + \varepsilon^{n+1},$$

as in Sect. 10.2.3. We can then apply (10.11)–(10.12), or the unknown-variance version of this model in (10.13)–(10.16), to update our beliefs about μ after each observation; for example, in the known-variance case, we would store the state of knowledge $K^n = (\theta^n, \Sigma^n)$ and use (10.11)–(10.12) to update the posterior parameters recursively. The allocation rule π will choose x^n based on the most recent posterior parameters. To evaluate π, we may use a version of the EOC objective given by

$$\inf_\pi \mathbb{E}^\pi \left(\mu^\top \left(x^* - \arg\max_{x \in \mathscr{X}} \left(\theta^N\right)^\top x\right)\right).$$

As before, the goal of the learning process is to help us find a better solution to the linear optimization problem, and the main methodological challenge is to design a rule that can do this more efficiently.

We have chosen to single out the setting of learning in a linear model because, on one hand, it has received less attention in the simulation literature than traditional R&S, and on the other hand, it appears to possess great practical potential. Essentially this model allows us to integrate learning with *regression-based optimization*. Linear regression is perhaps the most widely used statistical technique in business analytics and other application domains, used to estimate the relationship between a set of response variables $W^1, W^2, \ldots$ and accompanying vectors $x^0, x^1, \ldots$ of explanatory features. These features, however, may actually be controllable by the decision-maker, as in the following examples:

1. *Pricing* (Qu et al. 2013). The response variable represents the sales for a product; the features are derived from the price, which can be adjusted by the seller.

2. *Clinical trials* (Negoescu et al. 2011). A clinician tests large numbers of molecular configurations for a new cancer treatment. The regression features describe the configuration.
3. *Nonprofit management* (Han et al. 2016). The performance of a direct-mail fundraising campaign depends on the design attributes of the mailings, which are chosen individually for each campaign.

Thus, the decision-maker uses the regression model to predict the performance of various decisions, and then *optimizes* the model to obtain a single "recommended" decision. If learning is not involved, this will simply be the greedy decision

$$X^G = \arg\max_{x \in \mathcal{X}} \left(\theta^N\right)^\top x,$$

where θ^N is the vector of regression coefficients estimated from N data points. This decision will subsequently be put into practice (for example, a seller adjusts the price displayed on the firm's website) and its performance is observed. This observation, together with the attributes of the decision, constitutes a new data point that feeds back into the regression model, potentially leading to a different recommended decision the next time around. The modeling framework described in this section allows learning to be directly integrated into the decision-making process: instead of the greedy decision, one is now free to choose a different strategy that accounts for the uncertainty and potential for improvement in the regression coefficients.

10.3.3 Modeling Extensions

To illustrate that the issues discussed in this section are not exclusively confined to the examples we have covered, we briefly survey some recent extensions of the R&S framework. In all of these cases, the learning process remains largely unchanged (relying on the models in Sect. 10.2), but the optimization side of the problem is considerably more challenging. Many of these extensions have not received a Bayesian treatment as yet, but they offer potential future directions for research in this area, and also illustrate the diversity of problems in the learning space.

- *Constrained R&S*. Strictly speaking, the model in Sect. 10.3.2 is one instance of R&S with constraints. More generally, however, the constraints may be nonlinear or even unknown (Gramacy and Lee 2011), requiring us to expend additional simulation runs in order to find out whether a particular decision is even feasible. In fact, feasibility determination may be the focus of the problem: instead of maximizing an objective function, we may simply wish to correctly identify the feasible region (Szechtman and Yücesan 2008, 2016; Batur and Kim 2010). In the simulation community, Andradóttir and Kim (2010), Xu et al. (2010), and Pasupathy et al. (2014) have all approached constrained R&S.

- *Comparison with a standard*. In this problem, alternatives are additionally compared against a fixed value (the "standard"); if no alternative is found to outperform the standard, then the standard is implemented instead. See Nelson and Goldsman (2001), Kim (2005), or Chen (2006) for examples of methodological approaches to this problem. In some cases, the goal may simply be to find the set of alternatives that outperform the standard, without ranking them (Singham and Szechtman 2016).
- *Subset selection*. This is a version of R&S in which the goal is to find alternatives with the S highest values, rather than just the best. Chen et al. (2008), Wang et al. (2011), and Zhang et al. (2016) have variously approached this problem.
- *Multiobjective optimization*. As one may expect, this is a version of R&S in which a single alternative may have multiple "values." The goal now becomes to find the Pareto frontier with respect to these objectives. Some approaches to this problem from past WSC proceedings include Lee et al. (2004), Ryu et al. (2009), Feldman et al. (2015), and Hunter and Feldman (2015).
- *Quantile R&S*. One may also wish to optimize a quantile of the sampling distribution, rather than the mean. See, e.g., Bekki et al. (2007) or Shin et al. (2016) in the WSC proceedings.

10.4 Bayesian Decision Procedures

We now turn to the problem of designing efficient rules for allocating simulations, primarily in the context of the R&S problem that was described in Sect. 10.3.1. In keeping with the theme of this chapter, we specifically focus on *Bayesian* approaches to this problem. Thus, we do not survey the vast literature on indifference-zone selection (Kim and Nelson 2001; Hong and Nelson 2005) or other non-Bayesian techniques (Fan et al. 2016), nor do we attempt to compare Bayesian and non-Bayesian approaches (however, extensive numerical comparisons may be found in Branke et al. 2007).

Instead, we present two Bayesian methodologies with the potential for *generality*, that is, the guiding principle of each methodology should be flexible enough to produce computationally efficient implementations in a wide variety of simulation-based optimization problems beyond just R&S. We show that, for the two methodologies shown here, this is at least true for the linear model from Sect. 10.3.2. We also explain how the distinguishing characteristics of the Bayesian model are utilized to create these methods, and discuss the state of the art in their theoretical analysis.

10.4.1 Thompson Sampling

Thompson sampling (also often called "posterior sampling") is an algorithmic approach that dates back to Thompson (1933), but has recently attracted a great deal

of renewed attention (Chapelle and Li 2011; Agrawal and Goyal 2012; Russo and Van Roy 2014). The main appeal of this algorithm is due to its computational efficiency and ease of implementation (for example, Berry 2004 mentions its usefulness in clinical trials), and also because its simplicity lends itself well to theoretical analysis (Russo and Van Roy 2016).

Returning to the R&S problem (Sect. 10.3.1), recall that the greedy rule $X^G(K^n) = \arg\max_x \theta_x^n$ is perhaps the simplest possible way to allocate the next simulation. This simplicity is exploited by Thompson sampling. First, for every x, we draw a single sample $\hat{\theta}_x^n \sim \mathcal{N}\left(\theta_x^n, \left(\sigma_x^n\right)^2\right)$ from the current posterior distribution of μ_x. The next allocation decision is then made via the rule

$$x^n = \arg\max_x \hat{\theta}_x^n.$$

In other words, Thompson sampling is almost identical to the greedy policy, the only difference being that we greedily maximize over a set of samples drawn from the respective posterior distributions, instead of maximizing over only the posterior means. However, this difference has a profound impact on performance. Essentially, by sampling from the posterior, we automatically force the greedy rule to conduct some experimentation. The likelihood that we will allocate the next sample to alternative x directly depends on the uncertainty σ_x^n; if θ_x^n is small, but σ_x^n is large, there is still a chance that x will have the largest value in the random sample. At the same time, since $\theta_x^n \to \mu_x$ and $\sigma_x^n \to 0$ by the law of large numbers (as we sample x more often), eventually the randomly generated values will be close to the true values, and the rule will favor those alternatives that are truly better.

Thompson sampling is extremely practical because it is very easy to generalize. Thus, in the linear model considered in Sect. 10.3.2, we need only draw a sample $\hat{\theta}^n \sim \mathcal{N}(\theta^n, \Sigma^n)$ from the current (multivariate) posterior distribution, and solve

$$x^n = \arg\max_{x \in \mathcal{X}} \left(\hat{\theta}^n\right)^\top x. \tag{10.22}$$

Since the random sample is drawn first, before x^n is calculated, (10.22) is simply a deterministic LP (or IP). The computational complexity of solving (10.22) is the same as for the greedy rule, which would essentially solve the same problem with the posterior mean vector θ^n.

In many problems, it is often quite straightforward to implement a greedy rule, meaning that Thompson sampling can also be applied. For example, Defourny et al. (2015) considered a regression-based optimization problem where the greedy rule solves the second-order cone program (SOCP)

$$x^n = \arg\max_{x \in \mathcal{X}} (\theta^n)^\top x - \sqrt{x^\top C x}$$

for some matrix C. Additional exploration can be induced by simply drawing a sample $\hat{\theta}^n$ from the posterior distribution of μ and plugging this into the SOCP instead

of the mean vector θ^n. Computationally, this is much simpler than attempting to probabilistically analyze the solution to the SOCP (for instance, to compute expectations, as will be discussed in Sect. 10.4.2). This simplicity makes the Thompson rule an attractive choice for benchmarking in more complex problems where more specialized rules may be used.

In the context of simulation-based optimization, Thompson sampling does have a potential drawback, in that in some cases it may not induce *enough* exploration. Since $\theta_x^n \to \mu_x$ and $\sigma_x^n \to 0$, eventually the Thompson sample will be drawn from a posterior distribution that is strongly concentrated around the true value. This may lead the rule to over-sample x^* and under-sample suboptimal alternatives. While this may seem like a benefit, it means that our ability to distinguish suboptimal alternatives from the optimal one (as quantified by PCS) will be impaired. To solve this difficulty, Russo (2017) proposed the following simple modification for R&S: we first draw a sample $\hat{\theta}^n$ from the posterior distribution of μ as before, but we only make the decision $x^n = \arg\max_x \hat{\theta}_x^n$ with some prespecified probability β. The rest of the time, we continue drawing samples $\tilde{\theta}^n$ from the posterior until $\arg\max_x \tilde{\theta}_x^n \neq \arg\max_x \hat{\theta}_x^n$. We then make the decision $x^n = \arg\max_x \tilde{\theta}_x^n$. This rule (called "top-two Thompson sampling") can be shown to possess more desirable asymptotic behavior in simulation optimization problems than does ordinary Thompson sampling.

To bring the discussion back to the theme of this chapter, we should note that Bayesian logic is built into Thompson sampling, i.e., this form of exploration would not be possible if we had not used a Bayesian model. Due to the structure of this model, although we never know the exact value of μ, we are always able to precisely describe the *distribution* of the unknown values. Since the Thompson sample is also generated from this distribution, one can think of $\arg\max_x \hat{\theta}_x^n$ as an unbiased sample of $x^* = \arg\max_x \mu$; in other words, the conditional probability (given K^n) that $x = \arg\max_x \hat{\theta}_x^n$ is the same as the conditional probability that $x = x^*$. Although this probability is quite difficult to calculate, sampling from the corresponding distribution is almost trivial. In this way, Thompson sampling explores seemingly suboptimal x based on the precise likelihood (under the Bayesian assumptions) that x may in fact be the best choice.

10.4.2 Value of Information

Like Thompson sampling, value of information procedures (VIPs) uses the Bayesian model to make probabilistic forecasts of the unknown values μ_x and calculate various measures of the potential for any given alternative x to be the best. VIP-like procedures date back at least to Kushner (1964), with Jones et al. (1998) being perhaps the best-known algorithmic work on the topic. Also like Thompson sampling, VIPs have shown the ability to generalize to many complex decision problems beyond just R&S, although they tend to require specialized derivations for each problem setting.

10.4.2.1 Expected Improvement

The expected improvement (EI) method of Jones et al. (1998), one of the first and most enduring VIPs, can be stated as follows. Consider R&S with independent normal beliefs, i.e., $\mu_x \sim \mathcal{N}\left(\theta_x^n, \left(\sigma_x^n\right)^2\right)$ conditionally given K^n and (10.17)–(10.18) are used to update the posterior parameters. Then, the $(n+1)$st simulation is allocated to the alternative

$$x^n = \arg\max_x v_x^{EI,n}, \tag{10.23}$$

where

$$v_x^{EI,n} = \mathbb{E}\left(\max\left\{\mu_x - \max_y \theta_y^n, 0\right\} \mid K^n\right). \tag{10.24}$$

Equation (10.24) represents the potential for alternative x to improve upon our current beliefs about the best value.

First, (10.24) depends on μ_x. In frequentist statistics, this quantity is unknown and we would not be able to use it in computations. In the Bayesian model, it is still unknown, but we now know its conditional distribution given K^n. Furthermore, if K^n is given, the quantity $\max_y \theta_y^n$ is known to us. Thus, computationally, (10.24) is an instance of the equation

$$h(a,b) = \mathbb{E}(\max\{a + bZ, 0\})$$

for arbitrary $a \in \mathbb{R}$, $b > 0$ and $Z \sim \mathcal{N}(0,1)$. This expectation is relatively simple and, in fact, can be computed in closed form, which will be given shortly. With regard to the intuition behind (10.24), essentially $\max_y \theta_y^n$ is our point estimate of the highest value (if we had to stop learning right now, this is what we would report as the value of the best alternative); we then consider the possibility that, for a particular x, the true value μ_x may turn out to be better than this estimate (i.e., "improve" over it) and then calculate the expected value of this improvement (if μ_x turns out to be worse, that does not count as improvement). Because μ_x has a different distribution for each x, (10.24) will yield a different number for each x. We then assign the next simulation to the alternative that is believed to have the most potential.

The closed-form solution of (10.24) is given by

$$v_x^{EI,n} = \sigma_x^n f\left(-\frac{\left|\theta_x^n - \max_y \theta_y^n\right|}{\sigma_x^n}\right), \tag{10.25}$$

where $f(z) = z\Phi(z) + \phi(z)$ and ϕ, Φ are the pdf and cdf of the standard normal distribution. The function f is highly nonlinear and nonconvex in the posterior parameters, but is quite easy to code, given the availability of very efficient approximations for the standard normal cdf in standard software packages. Overall, the computational formula (10.24) represents an explicit mathematical tradeoff between the

estimated value of x, represented by θ_x^n, and our uncertainty about this value, represented by σ_x^n. In fact, $\nu_x^{EI,n}$ is increasing in σ_x^n (thus, all else being equal, more uncertainty is more valuable) and decreasing in $\left|\theta_x^n - \max_y \theta_y^n\right|$ (thus, if x is believed to be the best or close to it, it is also valued more highly). Furthermore, it is always the case that $\nu_x^{EI,n} > 0$ unless $\sigma_x^n = 0$, indicating that there is always some nonzero potential for improvement as long as uncertainty remains.

One is not required to accept (10.24) as the only definition of "expected improvement." In fact, the literature has developed many other definitions, some of which may work better in various settings. One popular alternate rule is the "knowledge gradient" or KG method (extensively surveyed in Powell and Ryzhov 2012), which replaces $\nu_x^{EI,n}$ by

$$\begin{aligned} \nu_x^{KG,n} &= \mathbb{E}\left(\max\left\{\theta_x^{n+1} - \max_{y\neq x}\theta_y^n, 0\right\} \mid K^n, x^n = x\right) \\ &= \mathbb{E}\left(\max_y \theta_y^{n+1} - \max_y \theta_y^n \mid K^n, x^n = x\right). \end{aligned} \tag{10.26}$$

Equation (10.26) is nearly identical to (10.24), with two differences. First, instead of attempting to predict the true value μ_x, we only look ahead to the results of the next simulation, i.e., μ_x is replaced by θ_x^{n+1}. Since we are conditioning on K^n, we know θ_x^n but not θ_x^{n+1}; however, one can derive the conditional distribution of θ_x^{n+1} given K^n (known as the "predictive distribution" in Bayesian statistics). Recalling (10.17)–(10.18), we note that, since only one set of beliefs is updated after each simulation, we need to condition on $x^n = x$ in addition to K^n in (10.26); essentially, we are calculating the expected improvement that can be obtained if we have already committed to simulating x but have not yet observed the outcome W_x^{n+1} of the simulation. The second major difference between EI and KG is that, in (10.26), alternative x is being compared to $\max_{y\neq x}\theta_y^n$, an estimate of the "best of the rest" among all alternatives that are not x. As will be clear in Sect. 10.4.3, this minor difference actually has substantial implications for theoretical performance.

Just like the EI method, KG admits the closed-form solution

$$\nu_x^{EI,n} = \tilde{\sigma}_x^n f\left(-\frac{\left|\theta_x^n - \max_{y\neq x}\theta_y^n\right|}{\tilde{\sigma}_x^n}\right), \tag{10.27}$$

where

$$\left(\tilde{\sigma}_x^n\right)^2 = \left(\sigma_x^n\right)^2 - \left(\sigma_x^{n+1}\right)^2$$

represents the (deterministic) uncertainty reduction achieved by performing one additional simulation of alternative x. In terms of computational cost, (10.25) and (10.27) are virtually identical. In terms of motivation, it may be argued that (10.26) reflects the value of information more accurately since we assign simulations one at a time (in other words, the next decision will be made based on a new set of parameters).

A third variation is known as "complete expected improvement" or CEI, developed very recently by Salemi et al. (2014) in the WSC proceedings. In this version, $v_x^{EI,n}$ is replaced by

$$v_x^{CEI,n} = \mathbb{E}\left(\max\left\{\mu_x - \mu_{x_*^n}, 0\right\} \mid K^n\right), \tag{10.28}$$

where $x_*^n = \arg\max_x \theta_x^n$ is the alternative currently believed to be the best. The motivation for this modification is that (10.25) simply uses the point estimate $\max_y \theta_y^n$ to represent the best value, but doing so ignores the uncertainty inherent in that estimate. On the other hand, (10.28) considers this uncertainty, and only uses a point estimate of x^* rather than for the best value. Computationally, (10.28) yields the solution

$$v_x^{CEI,n} = \sqrt{\left(\sigma_x^n\right)^2 + \left(\sigma_{x_*^n}^n\right)} f\left(-\frac{\left|\theta_x^n - \max_y \theta_y^n\right|}{\sqrt{\left(\sigma_x^n\right)^2 + \left(\sigma_{x_*^n}^n\right)}}\right), \tag{10.29}$$

for $x \neq x^*$. Essentially, the only difference from (10.25) is that we combine the uncertainty from both x and x_*^n into the calculation. However, (10.29) is only valid if $x \neq x_*^n$; otherwise, (10.28) evaluates to *zero*, requiring us to introduce some additional logic to determine whether we should sample x_*^n.

There are still other variants, such as probability of improvement (Kushner 1964), generalized EI (Sasena et al. 2002), weighted EI (Sóbester et al. 2005) and the similarly named "weighted improvement" (Ji and Kim 2013), multipoints EI (Ginsbourger et al. 2010), etc. At some point, the sheer number of EI-type methods makes it difficult to decide between them and delve into the seemingly minor (but not necessarily minor in actual fact) distinctions between different types of EI formulas. We shed some light on this issue in Sect. 10.4.3 below, but first we discuss one of the advantages of the EI logic, namely its ability to generalize beyond basic R&S.

10.4.2.2 Learning with Correlated Beliefs

We continue to explore the R&S problem, but now we will relax the assumption of independent beliefs. Instead, we will use the model from Sect. 10.2.2. Recall that, in this model, our beliefs are represented by $K^n = (\theta^n, \Sigma^n)$, which are updated recursively via the equations

$$\theta^n = \theta^{n-1} + \frac{W_{x^{n-1}}^n - \theta_{x^{n-1}}^{n-1}}{\lambda^2 + \Sigma_{x^{n-1},x^{n-1}}^{n-1}} \Sigma^{n-1} e_{x^{n-1}}, \tag{10.30}$$

$$\Sigma^n = \Sigma^{n-1} - \frac{\Sigma^{n-1} e_{x^{n-1}} e_{x^{n-1}}^\top \Sigma^{n-1}}{\lambda^2 + \Sigma_{x^{n-1},x^{n-1}}^{n-1}}. \tag{10.31}$$

As we saw before, we still simulate one alternative at a time, but now the correlation structure allows us to learn about multiple alternatives (possibly all of them) from one simulation.

For our discussion here, we will focus on the KG rule specifically. Recall from (10.26) that this rule makes a forecast of the impact of the next simulation on the posterior mean. In this case, however, a single simulation of x may change *every* component of θ^n, i.e., if K^n but not K^{n+1} is given, θ^{n+1} is a random vector. Fortunately, the distribution of this random vector can be simplified; Frazier et al. (2009) showed that the *conditional* distribution of θ^{n+1}, given K^n and given $x^n = x$, can be written as

$$\theta^{n+1} \sim \theta^n + \frac{\Sigma^n e_x}{\sqrt{\lambda^2 + \Sigma^n_{xx}}} \cdot Z, \tag{10.32}$$

where $Z \sim \mathcal{N}(0,1)$. Without getting into the technical details, we note one intuitive fact about (10.32): although every component of θ^{n+1} may be different from θ^n, the randomness driving this change is due to a single scalar random variable Z. This reflects the structure of the updating Eq. (10.30), where only a scalar random observation is made at each time stage.

Applying (10.32) to the KG rule, we find that the value of information $\nu^{KG,n}_x$ is essentially an instance of the equation

$$h(\theta, s) = \mathbb{E}\left(\max_y \theta_y + s_y \cdot Z\right) \tag{10.33}$$

with the particular vectors

$$\theta = \theta^n, \qquad s = \frac{\Sigma^n_{x\cdot}}{\sqrt{\lambda^2 + \Sigma^n_{xx}}}$$

plugged in. Equation (10.33) is the expected value of a convex, piecewise linear function of $Z \sim \mathcal{N}(0,1)$. The calculation of the expected value hinges on our ability to determine the breakpoints of the piecewise linear function; since the maximum in (10.33) is over finitely many y, it follows that there exist values c_y such that $\max_y \theta_y + s_y Z$ is linear on the interval $Z \in (c_{y-1}, c_y)$. This is illustrated in Fig. 10.1, which shows $\max_y \theta_y + s_y z$, as a function of z, in an instance with four different y.

If we know the breakpoints c_y, it is straightforward to calculate the expected value of the linear function on each interval, leading to the solution

$$h(\theta, s) = \sum_y (s_{y+1} - s_y) f\left(-\left|c_y\right|\right), \tag{10.34}$$

where the slopes s_y and breakpoints c_y have been sorted in increasing order (as in Fig. 10.1). The function $f(z) = z\Phi(z) + \phi(z)$ is the same as in (10.27). The main

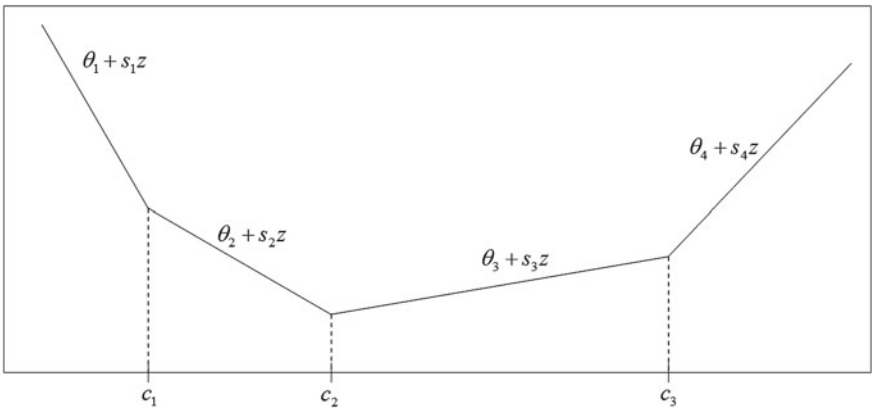


Fig. 10.1 Illustration of the piecewise linear function in the KG computation

computational challenge is then to find the breakpoints; this is a purely deterministic problem and can be solved using a numerical algorithm in Frazier et al. (2009).

There are two points to take away from this section. First, the value of information logic (represented by the KG method) is able to generalize to the setting of correlated beliefs. As a result, not only are we able to learn more quickly about the alternatives (because correlations are in our statistical model), but we can also consider these relationships between alternatives when determining what to sample next. Second, there is a cost that we have to pay for this ability: in order to extend the KG logic, we first have to go through a specialized algorithmic derivation, and then we have to perform additional computations (to find the breakpoints) in order to implement the procedure. These two points reoccur frequently in the literature on VIPs: the idea of expected improvement is flexible enough to admit a practical algorithm in a wide variety of problems, but this generally requires more work on the front- and back-end to derive and implement the algorithm. Some examples of such extensions include the following:

- *Unknown variance*. Chick and Inoue (2001) and Chick et al. (2010) develop EI-like criteria for problems where the sampling variance is unknown, and our uncertainty about it is modeled using a gamma distribution (as was briefly discussed in Sect. 10.2.3).
- *Linear regression*. Negoescu et al. (2011) and Han et al. (2016) considered VIPs in the setting of regression-based optimization.
- *Network problems*. Ryzhov and Powell (2011) proposed a VIP for learning in the context of network optimization (learning the length of the shortest path on a graph).
- *Global optimization*. Benassi et al. (2011) discussed how unknown variance may be handled in simulation-based optimization over a continuous space.

Overall, while Thompson sampling is generally much easier to implement, experimental work in the above papers suggests that various types of VIPs may perform quite well, particularly when the simulation budget is small. Furthermore, since the Thompson sample has to be randomly generated, the performance of VIPs may be subject to less random variability. Both approaches have their respective merits, both theoretical and practical.

10.4.3 Theoretical Analysis

In this section, we present a brief summary of recent breakthroughs in the theoretical study of Bayesian decision procedures. This material is of interest for two reasons. First, the sheer variety of algorithms, even within a single class (Thompson sampling or VIPs), makes it difficult to understand how they can be compared, or which one should actually be used; there is, of course, a large body of empirical work on this subject, but all of it is quite problem-dependent and does not suggest any clear conclusion. Theoretical guarantees help to shed light on this issue. Second, the in-depth theoretical study of these methods is very new, and the results presented here all appeared within the past 1–2 years. Thus, this is a rich and growing area with many opportunities for further work.

For this discussion, we confine ourselves (once again) to the R&S problem with independent normal beliefs, in which (10.17)–(10.18) are used for learning, and (10.19) is used to evaluate different allocation rules. In a seminal paper in the WSC proceedings, Glynn and Juneja (2004) developed a framework for characterizing the *theoretically optimal* allocation, which makes (10.19) converge to 1 at the fastest possible rate as the simulation budget $N \to \infty$. It turns out that the convergence rate of PCS depends on the asymptotic proportions of the budget assigned to the various alternatives. To formalize this idea, let

$$N_x^n = \sum_{m=0}^{n-1} 1_{\{x^n=x\}}$$

be the number of samples assigned to x up to time n (by convention, $N_x^0 = 0$). Suppose that the allocation rule π is chosen in such a way that the limit

$$p_x = \lim_{n\to\infty} \frac{N_x^n}{n}$$

exists with probability 1 *and is nonzero*, for every x. It then follows that the limit

$$\Gamma^p = -\lim_{n\to\infty} \frac{1}{n} P^\pi \left(\arg\max_x \theta_x^n \neq x^* \right)$$

also exists, implying that the probability of *incorrect* selection converges to zero at an exponential rate with exponent Γ^{π}. One can then characterize the particular limiting allocation $p \in \mathbb{R}^M$, satisfying $\sum_x p_x = 1, p_x > 0$, which maximizes Γ^p, thereby achieving the best possible rate of convergence. Under normality assumptions, this allocation satisfies the equations

$$\left(\frac{p_{x^*}}{\lambda_{x^*}}\right)^2 = \sum_{x \neq x^*} \left(\frac{p_x}{\lambda_x}\right)^2, \tag{10.35}$$

$$\frac{\left(\mu_x - \mu_{x^*}\right)^2}{\frac{\lambda_x^2}{p_x} + \frac{\lambda_{x^*}^2}{p_{x^*}}} = \frac{\left(\mu_y - \mu_{x^*}\right)^2}{\frac{\lambda_y^2}{p_y} + \frac{\lambda_{x^*}^2}{p_{x^*}}}, \qquad x, y \neq x^*. \tag{10.36}$$

It is worth noting that, if the solution happens to satisfy $p_{x^*} \gg \max_{x \neq x^*} p_x$, Eq. (10.36) is closely approximated by

$$\frac{p_x}{p_y} = \frac{\lambda_x^2 \left(\mu_y - \mu_{x^*}\right)^2}{\lambda_y^2 \left(\mu_x - \mu_{x^*}\right)^2}, \tag{10.37}$$

the ratio used to derive the well-known optimal computing budget allocation (OCBA) rule of Chen and Lee (2010).

The analysis in Glynn and Juneja (2004) is non-Bayesian, but this is not a major issue, as we could simply require the limiting proportion to satisfy (10.35)–(10.36) with probability 1, i.e., the optimality conditions must hold for every possible realization of the true values. One could also study a Bayesian criterion, such as (10.25), in a non-Bayesian setting; in that case one would simply look at the behavior of the computational formula and apply a frequentist asymptotic analysis.

We now return to the EI rule $x^n = \arg\max_x \nu_x^{EI,n}$, where $\nu_x^{EI,n}$ is given by (10.25). Although this rule has existed since at least Jones et al. (1998), it was not until Ryzhov (2015, 2016) that a detailed convergence rate analysis became available. The basic idea of the analysis is as follows. First, it is straightforward to show that $\nu_x^{EI,n} \to 0$, for all x, as $n \to \infty$. However, since the rule always assigns the next simulation to the alternative with the *largest* EI criterion, this suggests that $\nu_x^{EI,n}$ should decline to zero at the same rate for all x (if one criterion seems to be declining too quickly, we will stop sampling that alternative for some time). Therefore, the arguments inside the function f in (10.25) should behave similarly, that is,

$$\frac{\left|\theta_x^n - \max_z \theta_z^n\right|}{\sigma_x^n} \approx \frac{\left|\theta_y^n - \max_z \theta_z^n\right|}{\sigma_y^n} \tag{10.38}$$

for $x, y \neq x^*$ and n large. Recalling that $\theta^n \to \mu$ by the law of large numbers, and also that $\sigma_x^n \approx \frac{\lambda_x}{\sqrt{N_x^n}}$, (10.38) leads to the limiting result

$$\lim_{n\to\infty} \frac{N_x^n}{N_y^n} = \frac{\lambda_x^2 \left(\mu_y - \mu_{x^*}\right)^2}{\lambda_y^2 \left(\mu_x - \mu_{x^*}\right)^2}$$

for $x, y \neq x^*$. In words, the EI rule asymptotically achieves the OCBA ratios from (10.37).

The same type of "matching" argument can be applied (in a more rigorous fashion) to derive rate results for other variants of EI. Ryzhov (2016) presents such results for the KG criterion from (10.27), but neither this rule nor classic EI turn out to be optimal in the sense of (10.35)–(10.36). In that regard, the CEI criterion of (10.29), due to Salemi et al. (2014), is especially interesting. Applying the matching argument to CEI, we find that

$$\frac{\left|\theta_x^n - \theta_{x_*^n}^n\right|}{\sqrt{\left(\sigma_x^n\right)^2 + \left(\sigma_{x_*^n}^n\right)^2}} \approx \frac{\left|\theta_y^n - \theta_{x_*^n}^n\right|}{\sqrt{\left(\sigma_x^n\right)^2 + \left(\sigma_{x_*^n}^n\right)^2}}. \tag{10.39}$$

Recalling again that $\sigma_x^n \approx \frac{\lambda_x}{\sqrt{N_x^n}}$, we find that (10.39) looks remarkably similar to (10.36). In fact, we should be able to guarantee (10.36) asymptotically, provided that we can somehow ensure that p_{x^*} exists and is nonzero. To that end, Chen and Ryzhov (2017) proposes a slight modification of the CEI criterion, called "modified CEI" or mCEI. The modified procedure is shown in Fig. 10.2; it is easy to see that it is almost identical to CEI, but introduces additional logic to determine whether we should simulate alternative x_*^n, in a way that mimics the structure of condition (10.35). With this modification, Chen and Ryzhov (2017) shows that the procedure attains both (10.35) and (10.36) asymptotically.

In parallel, the Thompson sampling literature has developed in a similar way. Russo (2017) has also looked at Thompson sampling in the context of the rate-optimality framework of Glynn and Juneja (2004). The "top-two" version of Thompson sampling, which we discussed in Sect. 10.4.1, is likewise able to achieve (10.36) asymptotically, but encounters the same difficulty as CEI in satisfying condition

Let $n = 0$ and repeat the following:

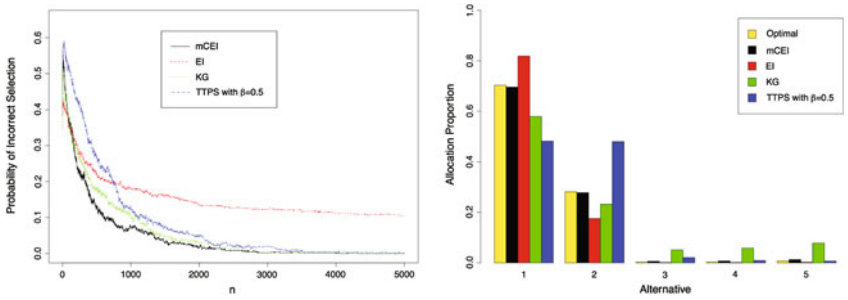
1: Check whether

$$\left(\frac{N_{x_*^n}^n}{\lambda_{x_*^n}}\right)^2 < \sum_{x \neq x_*^n} \left(\frac{N_x^n}{\lambda_x}\right)^2. \tag{40}$$

2: If (40) holds, assign $x^n = x_*^n$. If (40) does not hold, assign $x^n = \arg\max_{x \neq x_*^n} v_x^{CEI,n}$, where $v_x^{CEI,n}$ is given by (29).

3: Observe $W_{x^n}^{n+1}$, update posterior parameters, and increment n by 1.

Fig. 10.2 Modified CEI (mCEI) algorithm for R&S (Chen and Ryzhov 2017)



(a) Empirical probability of incorrect selection.

(b) Empirical allocations after 5000 simulations.

Fig. 10.3 Illustrative comparison of Bayesian decision procedures

(10.35), which governs the proportion of the budget allocated to the optimal alternative x^*. Essentially, the procedure resolves this difficulty by requiring the decision-maker to guess this proportion, in the form of a tunable parameter β. If β is chosen correctly (i.e., if $\beta = p_{x^*}$), the top-two rule will then achieve (10.36).

Figure 10.3 presents an illustrative example for a simple instance of R&S with just five alternatives (of which the first one is the best). Figure 10.3a shows the declining behavior of the probability of incorrect selection under the various Bayesian methods discussed in this chapter ("TTPS" is one of the top-two rules presented in Russo 2017), while Fig. 10.3b shows the empirical proportions of the budget that are assigned to the five alternatives after 5000 simulations, and compares these to the *optimal* proportions obtained by solving (10.35)–(10.36) by brute force for the given problem settings. All results are averaged over 500 macro-replications. We see that mCEI exhibits the best performance in terms of PCS, and furthermore, that the allocation produced by the mCEI rule is very close to optimal. The TTPS rule was run with $\beta = 0.5$; since this is not the correct proportion to assign to the first alternative, the rule experiences some difficulties.

This example is certainly not comprehensive, but it suggests that the theoretical differences between these rules do have an impact in terms of practical performance. Furthermore, the tuning issue (in the case of TTPS) is also important for practical performance. At the same time, it is reassuring that at least one Bayesian method is able to achieve theoretically optimal performance without the need for tuning. Results like this help to build a stronger mathematical foundation for the development and application of Bayesian decision support methods in simulation optimization.

10.5 Approximate Bayesian Inference

We briefly discuss approximate Bayesian inference, a very recent development in the literature. Recall that all of the Bayesian learning models presented in Sect. 10.2 possessed the property of *conjugacy*, which is essential for sequential learning because

it allows us to store and update our beliefs in an efficient way. Unfortunately, there are many problems of interest in which none of the existing conjugate models is suitable. Such problems arise whenever our ability to learn is censored or otherwise incomplete in some way. Section 10.5.1 presents one example of such a problem and discusses recent efforts to solve problems of this type. Section 10.5.2 discusses other applications culled from the literature.

10.5.1 Moment-Matching for Binary Censored Observations

Consider the following simple example. A doctor prescribes a dosage b of a drug to a patient. On average, the maximum dose that can be prescribed without unacceptable risk of side effects is denoted by μ. However, any given patient may have a higher or lower tolerance than this number. We assume that the patient's individual tolerance is represented by an independent random sample $W \sim \mathcal{N}\left(\mu, \lambda^2\right)$; for ease of presentation, we assume λ is known.

The doctor does not know μ, and imposes the prior distribution $\mu \sim \mathcal{N}\left(\theta, \sigma^2\right)$, where the prior parameters may be chosen based on past experience or published clinical studies. Now, if the doctor could observe W directly, we could simply apply (10.5)–(10.6) to update our beliefs. Unfortunately, the doctor cannot see W. Instead, the only observable information is the quantity

$$B = 1_{\{W<b\}},$$

which indicates whether or not the patient experienced side effects as a result of the prescribed dose b. The random variable B has a Bernoulli distribution, and the success probability depends on b. If we attempt to derive the posterior distribution of μ given B, using Bayes' rule, we will arrive at a complicated mixture density that does not resemble any standard family of distributions. It is thus not clear how we can update our beliefs, without attempting to somehow store the entire posterior density.

By itself, this may not be a big problem: statisticians routinely deal with non-conjugacy through the use of Markov chain Monte Carlo (MCMC) techniques (see Chick 1997 or Robert 2011 in the WSC proceedings), which have been the subject of extensive theoretical study (Asmussen and Glynn 2011). However, these techniques will become very cumbersome if we consider a *sequential* version of this problem, in which the doctor observes a sequence $B^1, B^2, \ldots$, where

$$B^n = 1_{\{W^n<b^{n-1}\}},$$

W^n is the individual tolerance of the nth patient, and b^{n-1} is the dose prescribed to that patient by the doctor. Ideally, we would like these prescription decisions to be adaptive. For example, supposing that the doctor wishes to limit the risk of side effects to a small probability α, the optimal dose might be

$$b^* = \mu + \lambda \Phi^{-1}(1-\alpha). \tag{10.41}$$

If we could maintain a posterior distribution of belief about μ, we might consider (for example) applying Thompson sampling to (10.41) and calculating doses that way. However, in order to do this, we require a tractable posterior distribution for μ. Attempting to derive this distribution in closed form will produce increasingly cumbersome mixture models, while applying MCMC will incur extremely high computational cost since we would have to implement a sequence of time-consuming MCMC algorithms.

Since we cannot update the beliefs exactly, we choose to update them *approximately* using the method of "moment-matching" (Opper 1998; Minka 2001). Assuming that $\mu \sim \mathcal{N}(\theta, \sigma^2)$ and that B is given with some fixed b, let $\tilde{\mu} \sim \mathcal{N}\left(\theta', \left(\sigma'\right)^2\right)$, where θ', σ' are chosen to satisfy the equations

$$\int_{\mathbb{R}} xP(\tilde{\mu} \in dx) = \int_{\mathbb{R}} xP(\mu \in dx \mid B), \tag{10.42}$$

$$\int_{\mathbb{R}} x^2 P(\tilde{\mu} \in dx) = \int_{\mathbb{R}} x^2 P(\mu \in dx \mid B). \tag{10.43}$$

In words, we create a completely artificial normal distribution, but choose the parameters of that normal distribution in such a way as to make its first and second moments equal those of the *exact* posterior density of μ given a single observation B. Assuming that these equations can be solved, we then simply discard the exact posterior, and move to the next stage of sampling under the assumption that our posterior beliefs about μ are exactly represented by the distribution $\mathcal{N}\left(\theta', \left(\sigma'\right)^2\right)$.

First, we demonstrate that the approach is computationally tractable. It can be shown (Chen and Ryzhov 2016) that (10.42)–(10.43) are solved by the update

$$\theta' = \theta - \sigma^2 \left(B \frac{1}{\sqrt{\lambda^2+\sigma^2}} \frac{\phi(q)}{\Phi(q)} - (1-B) \frac{1}{\sqrt{\lambda^2+\sigma^2}} \frac{\phi(q)}{1-\Phi(q)} \right), \tag{10.44}$$

$$\begin{aligned} \left(\sigma'\right)^2 = \sigma^2 \Bigg(& 1 - B \frac{\sigma^2}{\lambda^2+\sigma^2} \frac{q\phi(q)\Phi(q) + \phi^2(q)}{\Phi^2(q)} \\ & - (1-B) \frac{\sigma^2}{\lambda^2+\sigma^2} \frac{\phi^2(q) - q\phi(q)(1-\Phi(q))}{(1-\Phi(q))^2} \Bigg), \end{aligned} \tag{10.45}$$

where

$$q = \frac{b-\theta}{\sqrt{\lambda^2+\sigma^2}}.$$

Equations (10.44)–(10.45) require some algebra to derive, but they are trivial to code. We now have a learning model that can be implemented sequentially; we simply assume at every time stage that our distribution of belief is normal, and update the

parameters recursively as given above. This instantly gives back to us the ability to design and implement adaptive procedures for choosing b, whether based on Thompson sampling, value of information, or other approaches.

The main question of interest is whether such procedures have any hope of doing well. A very basic, but nonetheless difficult, question is whether $\theta^n \to \mu$, where $\{\theta^n\}_{n=0}^{\infty}$ is the sequence of estimators produced by recursively applying (10.44)–(10.45) based on the thresholds $b^0, b^1, \ldots$ and corresponding censored observations $B^1, B^2, \ldots$. Note that this property, i.e., the statistical consistency of the Bayesian estimator, is something that we took completely for granted in Sect. 10.2 due to the conjugacy property. In other words, this is a new challenge that arises when we move beyond conjugate models.

Intuitively, one might expect that the Bayesian estimator will *not* converge. Note that it is subject to two distinct sources of error. First, the binary observation B is already much less informative than the full observation W, and it is not immediately clear whether the censored observations are sufficient for learning the true value. Second, our approximation is developed with respect to a normal prior; however, if n is large, our current posterior distribution is already inaccurate, and so even the "exact" density $P(\mu \in dx \,|\, B)$ in (10.42)–(10.43) will in fact also have errors. Surprisingly, a simple numerical experiment with randomly generated threshold values $\{b^n\}_{n=0}^{\infty}$ shows that $\theta^n \to \mu$, as can be seen in Fig. 10.4.

Chen and Ryzhov (2016) provides the mathematical groundwork for this observation. By leveraging an analogy between (10.44) and (10.45) and the well-known field of gradient-based simulation optimization (Kim 2006), one can show that, under minor assumptions, θ^n will converge under any bounded sequence $\{b^n\}$. This surprising result shows that, far from being an ad-hoc approximation, approximate Bayesian inference is a powerful statistical methodology with strong practical potential in sequential learning problems where information is censored or incomplete. In fact, the practical literature on approximate Bayesian inference is quite a bit ahead of the theory on this topic, as will become clear in the next section.

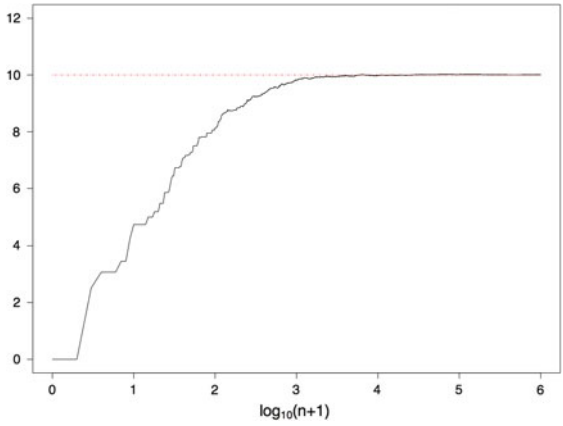


Fig. 10.4 Empirical convergence of the approximate Bayesian estimator to μ in an instance of the censored binary problem

10.5.2 Applications and Extensions

Approximate Bayesian inference is attracting increased attention mainly because of its computational power. Below, we briefly describe three additional applications where it has proven itself as an effective learning technique.

- *E-sports.* Possibly the best-known application of approximate Bayesian inference is the TrueSkill™ system developed and implemented (Herbrich et al. 2006; Dangauthier et al. 2007) by Microsoft's Xbox Live gaming service. In this application, large numbers of players simultaneously sign on to the service and request to play a competitive game. The system then has a short time in which to find a suitable opponent for each player. "Suitable" in this case means that the two opponents should be as evenly matched as possible, as players are likely to get discouraged and quit the game if they feel that they have no chance of winning. The system adopts an approximate Bayesian model in which each player x is assumed to have a "skill" value μ_x, a unitless and stylized quantity, and a normal prior is imposed upon μ by the system. Every time this player plays a game, the system assumes that a "performance" value $W \sim \mathcal{N}\left(\mu_x, \lambda^2\right)$, centered around the skill level, has been generated (modeling the fact that players sometimes perform better or worse than their average). However, this value also cannot be observed; instead, the game only allows win/loss outcomes. If player x wins against player y, this is interpreted by the system as the event that $W_x > W_y$. Thus, we are in the approximate Bayesian setting, and must use this binary setting to infer the values of μ_x and μ_y. The computational power of approximate Bayesian inference in this application was proved in practice; however, it was not until Chen and Ryzhov (2016) that the statistical consistency of the Bayesian procedure was established.
- *Market design.* When a financial market experiences a shock, traders may be unwilling to buy and sell assets because the value of these assets has become highly uncertain. In such cases, a market-maker may be used to encourage traders to trade. The market-maker sets bid and ask prices (and adjusts these prices over time) and traders react by buying (if their valuations are above the ask price), selling (if their valuations are below the bid price) or doing nothing (otherwise). The market-maker is never able to observe traders' valuations directly, but must infer them from the three types of signals. Das and Magdon-Ismail (2008) proposed an approximate Bayesian learning model, while Chakraborty et al. (2013) implemented it in a case study at Rensselaer Polytechnic University in which students traded mock assets whose value was related to their instructors' course evaluations.
- *R&S with unknown correlation structures.* Return to the R&S problem (Sect. 10.3.1) with correlated beliefs (Sect. 10.2.2). Although correlated beliefs can be powerful, a natural question to ask is how the practitioner might go about choosing the prior covariance matrix Σ. If there is no intuitive choice of correlation structure (such as the distance-based structure discussed previously), it is easy to argue that misspecified correlations may actually make it *more* difficult to estimate the unknown values (for example, if we believe two values to be similar, but

they are actually very different, the Bayesian update will completely misinterpret new information). Qu et al. (2012, 2015) and Zhang and Song (2015) considered a version of R&S in which the correlation structure was itself treated as random (using the Wishart matrix distribution). Unfortunately, there is no conjugate Bayesian model that can allow one to learn unknown correlation structures from scalar observations, as there is for the known-covariance case. On the other hand, approximate Bayesian inference can be leveraged to develop learning mechanisms that are computationally efficient and perform well empirically.

10.6 Closing Remarks

In closing this chapter, we would like to highlight a few common themes running through our discussion of Bayesian methods, which may be useful to consider for practitioners seeking to implement these methods or for researchers seeking to study them.

1. *Ease of updating*. Bayesian models are useful, not only because they model our uncertainty about the problem, but also because they model the *evolution* of uncertainty over time. Ideally, we would like to approach the problem in a sequential manner, so that each decision is always made using the most recent information. To do this, we require a model that is easy to store and update. Fortunately, conjugate models allow us to do this in a wide range of applications; when these models are not suitable, approximate Bayesian inference offers a principled learning approach.
2. *Computational generalizability*. The Bayesian methods that we have surveyed are based on general concepts, such as posterior sampling and expected value of information, that can lead to efficient algorithms in a wide variety of settings beyond just ranking and selection. For any of these settings, we may have to derive a specialized procedure based on the concept; however, the concept itself is very general and not beholden to any particular type of problem structure.
3. *Connections with frequentist approaches*. Interestingly, in some cases Bayesian methods turn out to exhibit parallels with other simulation optimization methodologies. Thus, as we have seen in Sect. 10.4.3, value of information procedures can be related to the optimal computing budget allocation class of methods, as well as to the theoretical optimality analysis of Glynn and Juneja (2004). It may be that, in adopting a Bayesian method, one is not necessarily "giving up" the chance to use a different approach; rather, the Bayesian method may simply be a different way of reaching the same result.
4. *Correlated beliefs*. At the same time, the Bayesian idea of "correlated beliefs" does seem to provide something that may not be easy to find in frequentist models—the ability to learn about large decision spaces from very small numbers of measurements. This ability is primarily useful when the measurement budget is very small, and one never has the opportunity to reach the asymptotic regime studied in the theory. However, there are modeling pitfalls having to do with the correct specification of the correlation structure.

References

Agrawal S, Goyal N (2012) Analysis of Thompson sampling for the multi-armed bandit problem. In: Mannor S, Srebro N, Williamson RC (eds) Proceedings of the 25th annual conference on learning theory, pp 39:1–39:26

Andradóttir S, Kim S-H (2010) Fully sequential procedures for comparing constrained systems via simulation. Naval Res Logist 57(5):403–421

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. Oper Res 58(2):371–382

April J, Better M, Glover F, Kelly J, Laguna M (2006) Enhancing business process management with simulation optimization. In: Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM (eds) Proceedings of the 2006 winter simulation conference, pp 642–649

Arlotto A, Gans N, Chick SE (2010) Optimal employee retention when inferring unknown learning curves. In: Johansson B, Jain S, Montoya-Torres J, Hugan J, Yücesan E (eds) Proceedings of the 2010 winter simulation conference, pp 1178–1188

Asmussen S, Glynn PW (2011) A new proof of convergence of MCMC via the ergodic theorem. Stat. Probab. Lett. 81(10):1482–1485

Batur D, Kim S-H (2010) Finding feasible systems in the presence of constraints on multiple performance measures. ACM Trans Model Comput Simul 20(3):13:1–13:29

Bekki JM, Fowler JW, Mackulak GT, Nelson BL (2007) Using quantiles in ranking and selection procedures. In: Henderson SG, Biller B, Hsieh M-H, Shortle J, Tew JD, Barton RR (eds) Proceedings of the 2007 winter simulation conference, pp 1722–1728

Benassi R, Bect J, Vazquez E (2011) Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion. In: Coello CA (ed) Learning and intelligent optimization, pp 176–190

Berry DA (2004) Bayesian statistics and the efficiency and ethics of clinical trials. Stat Sci 19(1):175–187

Bickel PJ, Doksum KA (2015) Mathematical statistics: basic ideas and selected topics, vol. I. CRC Press

Branke J, Chick SE, Schmidt C (2007) Selecting a selection procedure. Manag Sci 53(12):1916–1932

Chakraborty M, Das S, Lavoie A, Magdon-Ismail M, Naamad Y (2013) Instructor rating markets. In: Proceedings of the 27th AAAI conference on artificial intelligence, pp 159–165

Chapelle O, Li L (2011) An empirical evaluation of Thompson sampling. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ (eds) Advances in neural information processing systems, vol 24, pp 2249–2257

Chau M, Fu MC, Qu H, Ryzhov IO (2014) Simulation optimization: a tutorial overview and recent developments in gradient-based methods. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference, pp 21–35

Chen C-H, Lee LH (2010) Stochastic simulation optimization: an optimal computing budget allocation. World Scientific

Chen C-H, Chick SE, Lee LH, Pujowidianto NA (2015) Ranking and selection: efficient simulation budget allocation. In: Fu MC (ed) Handbook of simulation optimization. Springer, pp 45–80

Chen C-H, He D, Fu MC, Lee LH (2008) Efficient simulation budget allocation for selecting an optimal subset. INFORMS J Comput 20(4):579–595

Chen C-H, Lin J, Yücesan E, Chick SE (2000) Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Discret Event Dyn Syst 10(3):251–270

Chen EJ (2006) Comparison with a standard via all-pairwise comparisons. Discret Event Dyn Syst 16(3):385–403

Chen Y, Ryzhov IO (2016) Approximate Bayesian inference as a form of stochastic approximation: a new consistency theory with applications. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference, pp. 534–544

Chen Y, Ryzhov IO (2017) Rate-optimality of complete expected improvement. Submitted for publication

Chick SE (1997) Bayesian analysis for simulation input and output. In: Andradóttir S, Healy KJ, Withers DH, Nelson BL (eds) Proceedings of the 1997 winter simulation conference, pp 253–260

Chick SE (2000) Bayesian methods: Bayesian methods for simulation. In: Joines JA, Barton RR, Kang K, Fishwick PA (eds) Proceedings of the 2000 winter simulation conference, pp 109–118

Chick SE (2004) Bayesian methods for discrete event simulation. In: Ingalls R, Rossetti MD, Smith JS, Peters BA (eds) Proceedings of the 2004 winter simulation conference, pp 89–100

Chick SE (2006a) Bayesian ideas and discrete event simulation: why, what and how. In: Perrone, LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM (eds) Proceedings of the 2006 winter simulation conference, pp 96–106

Chick SE (2006b), Subjective probability and bayesian methodology. In: Henderson SG, Nelson BL (eds) Handbooks of operations research and management science, vol 13. Simulation. North-Holland Publishing, Amsterdam, pp 225–258

Chick SE, Inoue K (2001) New two-stage and sequential procedures for selecting the best simulated system. Oper Res 49(5):732–743

Chick SE, Branke J, Schmidt C (2010) Sequential sampling to myopically maximize the expected value of information. INFORMS J Comput 22(1):71–80

Chick SE, Forster M, Pertile P (2015) Optimal sequential sampling with delayed observations and unknown variance. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 3789–3800

Conn AR, Scheinberg K, Vicente LN (2009) Introduction to derivative-free optimization. SIAM

Dangauthier P, Herbrich R, Minka TP, Graepel T (2007) TrueSkill through time: revisiting the history of chess. In: Platt JC, Koller D, Singer Y, Roweis S (eds) Advances in neural information processing systems, Vol 20, pp 337–344

Das S, Magdon-Ismail M (2008) Adapting to a market shock: optimal sequential market-making. In: Koller D, Bengio Y, Schuurmans D, Bottou L, Culotta R (eds) Advances in neural information processing systems, vol 21, pp 361–368

Defourny B, Ryzhov IO, Powell WB (2015) Optimal information blending with measurements in the L2 sphere. Math Oper Res 40(4):1060–1088

DeGroot MH (1970) Optimal statistical decisions. Wiley

Fan W, Hong LJ, Nelson BL (2016) Indifference-zone-free selection of the best. Oper Res 64(6):1499–1514

Feldman G, Hunter SR, Pasupathy R (2015) Multi-objective simulation optimization on finite sets: optimal allocation via scalarization. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 3610–3621

Frazier PI, Powell WB, Dayanik S (2009) The knowledge-gradient policy for correlated normal rewards. INFORMS J Comput 21(4):599–613

Fu MC (2015) Handbook of simulation optimization. Springer

Fu MC, Andradóttir S, Carson JS, Glover F, Harrell CR, Ho Y-C, Kelly JP, Robinson SM (2000) Integrating optimization and simulation: research and practice. In: Joines JA, Barton RR, Kang K, Fishwick PA (eds) Proceedings of the 2000 winter simulation conference, pp 610–616

Fu MC, Bayraksan G, Henderson SG, Nelson BL, Powell WB, Ryzhov IO, Thengvall B (2014) Simulation optimization: a panel on the state of the art in research and practice. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference, pp 3696–3706

Fu MC, Chen C-H, Shi L (2008) Some topics for simulation optimization. In: Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T, Fowler JW (eds) Proceedings of the 2008 winter simulation conference, pp 27–38

Gao S, Chen W, Shi L (2017) A new budget allocation framework for the expected opportunity cost. Oper Res (to appear)

Ginsbourger D, Le Riche R, Carraro L (2010) Kriging is well-suited to parallelize optimization. In: Tenne Y, Goh C-K (eds) Computational intelligence in expensive optimization problems. Springer, pp 131–162

Gittins, JC, Glazebrook KD, Weber R (2011) Multi-armed bandit allocation indices, 2nd ed. John Wiley

Glynn PW, Juneja S (2004) A large deviations perspective on ordinal optimization. In: Ingalls R, Rossetti MD, Smith JS, Peters BA (eds) Proceedings of the 2004 winter simulation conference, pp 577–585

Gramacy RB, Lee HKH (2011) Optimization under unknown constraints. In: Bernardo JM, Bayarri MJ, Berger JO, Dawid AP, Heckerman D, Smith AFM, West M (eds) Bayesian statistics 9: proceedings of the ninth valencia international meeting, pp 229–256

Han B, Ryzhov IO, Defourny B (2013) Efficient learning of donor retention strategies for the American Red Cross. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference, pp 17–28

Han B, Ryzhov IO, Defourny B (2016) Optimal learning in linear regression with combinatorial feature selection. INFORMS J Comput 28(4):721–735

Herbrich R, Minka TP, Graepel T (2006) TrueSkill™: a Bayesian skill rating system. In: Schölkopf B, Platt JC, Hoffman T (eds) Advances in neural information processing systems, vol. 19, pp. 569–576

Hong LJ, Nelson BL (2005) The tradeoff between sampling and switching: new sequential procedures for indifference-zone selection. IIE Trans 37(7):623–634

Hong LJ, Nelson BL (2009) A brief introduction to optimization via simulation. In: Rosetti M, Hill R, Johansson B, Dunkin A, Ingalls R (eds) Proceedings of the 2009 winter simulation conference, pp 75–85

Hong LJ, Nelson BL, Xu J (2015) Discrete optimization via simulation. In: Fu MC (ed) Handbook of simulation optimization. Springer, pp 9–44

Huang D, Allen TT, Notz WI, Zeng N (2006) Global optimization of stochastic black-box systems via sequential kriging meta-models. J Glob Optim 34(3):441–466

Hunter SR, Feldman G (2015) Optimal sampling laws for bi-objective simulation optimization on finite sets. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 3749–3757

Ji Y, Kim S (2013) An adaptive radial basis function method using weighted improvement. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference, pp 957–968

Jian N, Henderson SG (2015) An introduction to simulation optimization. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 1780–1794

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492

Kaelbling LP (1993) Learning in Embedded Systems. MIT Press, Cambridge, MA

Kim S (2006) Gradient-based simulation optimization. In: Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM (eds) Proceedings of the 2006 winter simulation conference, pp 159–167

Kim S, Zhang D (2010) Convergence properties of direct search methods for stochastic optimization. In: Johansson B, Jain S, Montoya-Torres J, Hugan J, Yücesan E (eds) Proceedings of the 2010 winter simulation conference, pp 1003–1011

Kim S-H (2005) Comparison with a standard via fully sequential procedures. ACM Trans Model Comput Simul 15(2):155–174

Kim S-H, Nelson BL (2001) A fully sequential procedure for indifference-zone selection in simulation. ACM Trans Model Comput Simul 11(3):251–273

Kim S-H, Nelson BL (2007) Recent advances in ranking and selection. In: Henderson SG, Biller B, Hsieh M-H, Shortle J, Tew JD, Barton RR (eds) Proceedings of the 2007 winter simulation conference, pp 162–172

Kleijnen JPC (2009) Kriging metamodeling in simulation: a review. Eur J Oper Res 192(3):707–716
Kotz S, Nadarajah S (2004) Multivariate t-distributions and their applications. Cambridge University Press
Kushner HJ (1964) A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. J Basic Eng 86(1):97–106
Lee LH, Chew EP, Teng S, Goldsman D (2004) Optimal computing budget allocation for multi-objective simulation models. In: Ingalls R, Rossetti MD, Smith JS, Peters BA (eds) Proceedings of the 2004 winter simulation conference, pp 586–594
Minka TP (2001) Expectation propagation for approximate Bayesian inference. In: Proceedings of the 17th conference on uncertainty in artificial intelligence, pp 362–369
Negoescu DM, Frazier PI, Powell WB (2011) The knowledge-gradient algorithm for sequencing experiments in drug discovery. INFORMS J Comput 23(3):346–363
Nelson BL, Goldsman D (2001) Comparisons with a standard in simulation experiments. Manag Sci 47(3):449–463
Opper M (1998) A Bayesian approach to on-line learning. In: Saad D (ed) On-line learning in neural networks, pp 363–378
Pasupathy R, Hunter SR, Pujowidianto NA, Lee LH, Chen C-H (2014) Stochastically constrained ranking and selection via SCORE. ACM Trans Model Comput Simul **25**(1), 1:1–1:26
Powell WB (2011) Approximate dynamic programming: solving the curses of dimensionality, 2nd edn. Wiley, New York
Powell WB, Ryzhov IO (2012) Optimal learning. Wiley
Qu H, Ryzhov IO, Fu MC (2012) Ranking and selection with unknown correlation structures. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Uhrmacher AM (eds) Proceedings of the 2012 winter simulation conference, pp 144–155
Qu H, Ryzhov IO, Fu MC (2013) Learning logistic demand curves in business-to-business pricing. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference, pp 29–40
Qu H, Ryzhov IO, Fu MC, Ding Z (2015) Sequential selection with unknown correlation structures. Oper Res 63(4):931–948
Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT Press
Robert CP (2011) Simulation in statistics. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu MC (eds) Proceedings of the 2011 winter simulation conference, pp 108–119
Russo D (2017) Simple Bayesian algorithms for best arm identification. arXiv:1602.08448
Russo D, Van Roy B (2014) Learning to optimize via posterior sampling. Math Oper Res 39(4):1221–1243
Russo D, Van Roy B (2016) An information-theoretic analysis of Thompson sampling. J Mach Learn Res **17**, 68:1–68:30
Ryu J-H, Kim S, Wan H (2009) Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization. In: Rosetti MD, Hill RR, Johansson B, Dunkin A, Ingalls RG (eds) Proceedings of the 2009 winter simulation conference, pp 623–633
Ryzhov IO (2015) Expected improvement is equivalent to OCBA. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 3668–3677
Ryzhov IO (2016) On the convergence rates of expected improvement methods. Oper Res 64(6):1515–1528
Ryzhov IO, Powell WB (2011) Information collection on a graph. Oper Res 59(1):188–201
Ryzhov IO, Powell WB (2012) Information collection for linear programs with uncertain objective coefficients. SIAM J Optim 22(4):1344–1368
Salemi P, Nelson BL, Staum J (2014) Discrete optimization via simulation using Gaussian Markov random fields. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference, pp 3809–3820
Sasena MJ, Papalambros P, Goovaerts P (2002) Exploration of metamodeling sampling criteria for constrained global optimization. Eng Optim 34(3):263–278

Scott WR, Powell WB, Simão HP (2010) Calibrating simulation models using the knowledge gradient with continuous parameters. In: Johansson B, Jain S, Montoya-Torres J, Hugan J, Yücesan E (eds) Proceedings of the 2010 winter simulation conference, pp 1099–1109

Shin D, Broadie M, Zeevi A (2016) Tractable sampling strategies for quantile-based ordinal optimization. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference, pp 847–858

Singham DI, Szechtman R (2016) Multiple comparisons with a standard using false discovery rates. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference, pp 501–511

Sóbester A, Leary SJ, Keane AJ (2005) On the design of optimization strategies based on global response surface approximation models. J Glob Optim 33(1):31–59

Staum J (2009) Better simulation metamodeling: the why, what, and how of stochastic kriging. In: Rosetti MD, Hill RR, Johansson B, Dunkin A, Ingalls RG (eds) Proceedings of the 2009 winter simulation conference, pp 119–133

Szechtman R, Yücesan E (2008) A new perspective on feasibility determination. In: Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T, Fowler JW (eds) Proceedings of the 2008 winter simulation conference, pp 273–280

Szechtman R, Yücesan E (2016) A Bayesian approach to feasibility determination. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference, pp 782–790

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25(3–4):285–294

Wang Y, Luangkesorn L, Shuman LJ (2011) Best-subset selection procedure. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu MC (eds) Proceedings of the 2011 winter simulation conference, pp 4310–4318

Xie W, Nelson BL, Barton RR (2014) A Bayesian framework for quantifying uncertainty in stochastic simulation. Oper Res 62(6):1439–1452

Xu J, Nelson BL, Hong LJ (2010) Industrial strength COMPASS: a comprehensive algorithm and software for optimization via simulation. ACM Trans Model Comput Simul **20**(1), 3:1–3:29

Zhang Q, Qian PZG (2013) Designs for crossvalidating approximation models. Biometrika 100(4):997–1004

Zhang Q, Song Y (2015) Simulation selection for empirical model comparison. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference, pp 3777–3788

Zhang S, Chen P, Lee LH, Chew EP, Chen C-H (2011) Simulation optimization using the particle swarm optimization with optimal computing budget allocation. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu MC (eds) Proceedings of the 2011 winter simulation conference, pp 4298–4309

Zhang S, Lee LH, Chew EP, Xu J, Chen C-H (2016) A simulation budget allocation procedure for enhancing the efficiency of optimal subset selection. IEEE Trans Autom Control 61(1):62–75

Chapter 11
Simulation Optimization Under Input Model Uncertainty

Enlu Zhou and Di Wu

Abstract Stochastic simulation is driven by the input model, which is a collection of distributions that model the randomness in the system. Since the input model is often estimated from data of past observations, simulation is subject to the so-called input model uncertainty due to the finiteness of the data. As a result, optimizing the corresponding simulation model may lead to solutions that perform poorly under the true model. In the past, simulation optimization has been mostly studied under the assumption that the input model is accurate, thus only accounting for stochastic uncertainty in the system but ignoring input model uncertainty. In this chapter, we aim to answer two questions: (1) how to quantify the impact of input model uncertainty on the simulation optimization problem; (2) how to "optimize" a simulation model when taking into account input model uncertainty. To address the first question, we provide asymptotic results and confidence intervals on the optimality gap and performance of solutions. For the second question, we review a recently proposed framework of Bayesian risk optimization that captures trade-off between the expected performance and the variability of the actual performance. Many research questions still remain for simulation optimization under input model uncertainty, and will be discussed briefly at the end of this chapter.

11.1 Introduction

Complex systems arising in various areas such as supply chain, manufacturing, and service networks often require simulation and optimization techniques to evaluate the system performance and facilitate decision making. Such systems are characterized by complexity, nonlinearity, and stochasticity in system dynamics, and often resort to computer simulation for performance analysis and optimization. One crucial factor in successful simulation modeling is the so-called *input model*—a collection of probability distributions that model the system stochastic uncertainty. For

E. Zhou (✉) · D. Wu
Georgia Institute of Technology, Atlanta, Georgia
e-mail: enlu.zhou@isye.gatech.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_11

example, in a supply chain system, the random customer demands and order lead times need to be modeled by a collection of distributions, which are often estimated from observed data of past sales and order arrival times. Since there is only a finite amount of data available to the modeler, the input model is always subject to uncertainty, which is referred to as *input model uncertainty* or simply as *input uncertainty*. As a result, a typical stochastic simulation faces two types of uncertainties: the *stochastic uncertainty* of the system, and the input uncertainty of the system model. The performance estimation error due to stochastic uncertainty can be "simulated away", i.e., we can decrease the error by increasing the number of replications that we run the simulation model. In distinctive contrast, the input uncertainty is often uncontrollable, due to the limited amount of input data available and the difficulty in acquiring additional data.

Simulation optimization has been mostly studied under the assumption that the input model is accurate, and thus only accounts for stochastic uncertainty and ignores input uncertainty. One potential consequence of this approach is that if we optimize a model built on a small number of input data, the obtained optimal solution may have a poor performance under the true model. To see the risk of ignoring input uncertainty, let us consider an illustrative example of the newsvendor problem below.

11.1.1 Illustrative Example: Risk of Input Uncertainty

Consider the newsvendor problem with lost sales, where the goal is to choose an order amount to maximize the expected profit under stochastic demand. Suppose the order amount is denoted as x, demand is exponentially distributed with rate $\theta^c = 1/20$, and the unit price and unit cost are $p = 2$ and $c = 1$, respectively. The expected profit is given by

$$H(x, \theta^c) := p\mathbf{E}[\min(x, \xi)] - cx = \frac{p}{\theta^c}\left(1 - e^{-\theta^c x}\right) - cx$$

Note that since the closed-form expression of the expectation is known in this example, we will use this form directly for evaluation of the objective value instead of running simulation. We draw 1,000 sets of independent and identically distributed (i.i.d.) demand data, each dataset of size 20. Then, for each dataset we use maximum likelihood to get a point estimate $\hat{\theta}$ of θ^c, and we plug the estimate into the model to solve the *empirical optimization* problem $\max_x H(x, \hat{\theta})$. The histograms of optimal solutions and their corresponding true performances (under θ^c) are shown in Fig. 11.1 below. Although the average optimal solution is close to the true optimal solution 13.86, the extreme values could reach 5 and 25. Similarly, the extreme performance of solutions can be lower than 3, compared with the true optimal performance 6.14. In this example, a decision maker may suffer a great loss if s/he blindly assumes that the estimated input distribution is accurate. This simple example demonstrates the importance of taking into account input uncertainty when doing simulation optimization.

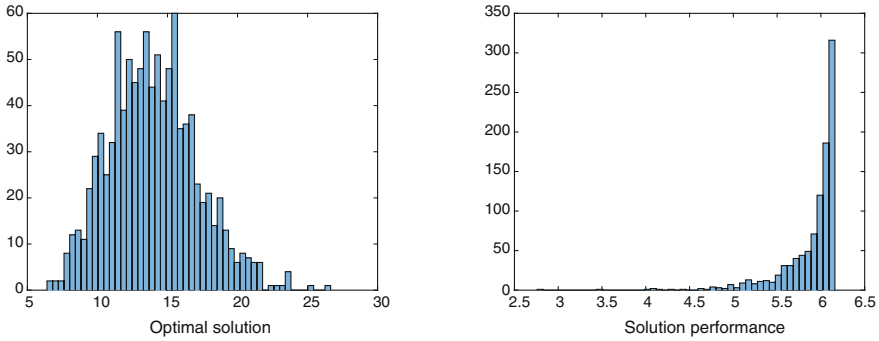


Fig. 11.1 Risk of input uncertainty in the newsvendor problem

11.2 Problem Setting and Fundamental Questions

In simulation optimization, one often wants to optimize the expected performance of a simulation model. Mathematically put, the following problem is of interest.

$$\min_{x \in \mathscr{X}} \mathbf{E}_{\xi \sim F^c}[h(x, \xi)], \tag{11.1}$$

where x is a decision variable that takes value in $\mathscr{X} \subset \mathbb{R}^d$, ξ is a random vector (r.v.) that takes value in $\mathbb{R}^m$, and $h : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ is a deterministic function capturing the system dynamics. Note that the form h is not known explicitly but instead is evaluated through simulation (hence the name "simulation optimization"). Here ξ represents the stochastic uncertainty in the system, and its true distribution is denoted by F^c.

In practice F^c is not known and often estimated from data. Suppose we have a given dataset consisting of n i.i.d. data samples from F^c, denoted by $\psi^n := (\xi_1, \ldots, \xi_n)$. Using the dataset ψ^n, an input model $\hat{F}$ is estimated and then used to drive the simulation. Without consideration of the estimation error in the input model, the typical simulation optimization setting solves the following *empirical optimization* problem:

$$\min_{x \in \mathscr{X}} \mathbf{E}_{\xi \sim \hat{F}}[h(x, \xi)]. \tag{11.2}$$

A solution to problem (11.2) may perform poorly for the true problem (11.1), as shown in the illustrative example above. This raises at least two fundamental questions:

- how to quantify the uncertainty in the performance of solutions to (11.2);
- how to take into account input uncertainty when doing simulation optimization.

We aim to answer these questions for a simplified setting, where the distribution of ξ has a parametric form. More specifically, we suppose F^c lives in a parametric family of distributions $\{F(\cdot; \theta), \theta \in \Theta\}$, with the true but unknown parameter value θ^c. For

technicality reasons, we assume θ^c is in the interior of Θ. The input uncertainty in this setting is the random error in the estimation of θ^c due to finiteness of data. For ease of presentation below, we define the function

$$H(x, \theta) := \mathbf{E}[h(x, \xi)], \quad \xi \sim F(\cdot; \theta), \tag{11.3}$$

where $\int |h(x, \xi)| dF < \infty$ for all $x \in \mathscr{X}, \theta \in \Theta$ so that H is well-defined. Problem (11.1) can then be rewritten as $\min_{x \in \mathscr{X}} H(x, \theta^c)$.

11.3 Relevant Literature

There has been an extensive study on how to quantify the impact of input uncertainty on simulation output (without optimization) in the presence of stochastic uncertainty. The methods proposed so far in the literature can be roughly grouped into three major categories. First is the frequentist methods that allow nonparametric input distributions and use direct or bootstrap resampling techniques to assess input uncertainty, such as Barton and Schruben (1993, 2001) and Cheng and Holloand (1997). Second is the Bayesian model averaging methods that assume a parametric model and use the posterior distribution of unknown parameters as the sampling distribution during the simulation process, such as Chick (2001) and Zouaoui and Wilson (2003, 2004). Third is the delta method: Cheng and Holloand (1997) uses the delta method to decompose the variance of simulation output into two parts respectively corresponding to stochastic uncertainty and input uncertainty; a robust sensitivity analysis approach developed by Lam (2016) can be viewed as the delta method with respect to a distributional perturbation. Recent advances in stochastic kriging Ankenman et al. (2010) also gives rise to the application of meta-model assisted methods to quantify input uncertainty, such as Barton et al. (2013) and Xie et al. (2014a, b). In addition, Song and Nelson (2015) proposes a method for quickly assessing the relative contribution of each input distribution to the overall effect of input uncertainty; Zhu and Zhou (2015) develops an approach to quantify the risk associated with input uncertainty in simulation output analysis. For a more comprehensive and detailed overview of input uncertainty quantification, please refer to Chap. 5 in this book Song and Nelson (2016).

Simulation optimization under input uncertainty, on the other hand, has not been studied as extensively. A few recent works in the simulation optimization literature have started to look at the case of a finite solution space, i.e., the ranking and selection (R&S) problem. Song et al. (2015) considers the situation where solutions have independent but different input distributions, and develops a procedure for selecting a subset of solutions whose performances are within a user-specified distance from the true optimal performance. Corlu and Biller (2013) considers a case of shared input distributions across solutions and explores the impact of input uncertainty on the indifference-zone (IZ) method; they show that a direct application of IZ selection may provide invalid correct-selection guarantee. Fan et al. (2013) presents a robust

IZ formulation that selects the best design with respect to the worst-case choices among a finite collection of possible input models. Song and Nelson (2016) develops an input–output uncertainty comparisons procedure that finds a set of likely optimal solutions with given probability guarantee. Gao et al. (2017) develops a robust optimal computing budget allocation (OCBA) procedure to find the best solution under the worst-case input distribution scenario. Despite these pioneering works on R&S with consideration of input uncertainty, the case of continuous solution spaces has rarely been studied for simulation optimization under input uncertainty.

Since the formulation of simulation optimization bears great similarity to stochastic optimization, perhaps the most related study in the literature of stochastic optimization is distributionally robust optimization (DRO). The idea of DRO is to optimize the worst case over a family of possible input distributions, where the crux is to construct a tractable ambiguity set with some probabilistic guarantees. An abundant literature is available on this topic, including but not limited to Bertsimas et al. (2011), Delage and Ye (2010), Bertsimas et al. (2013), Wiesemann et al. (2014), Esfahani and Kuhn (2015), Bayraksan and Love (2015), Ben-Tal et al. (2013), Jiang and Guan (2015), Popescu (2007), Zymler et al. (2013). One issue with DRO, as observed in Wang et al. (2016), is that if the ambiguity set is constructed in an inappropriate way, a solution optimizing the worst-case scenario may be so conservative that it performs poorly under scenarios that are much more likely in reality. Furthermore, unlike the setting in the DRO literature where the tractability (such as convexity) of the objective function is often a reasonable assumption, the objective function in simulation optimization can only be evaluated through simulation and often lacks such structural properties. Therefore, simulation optimization under input uncertainty needs a new formulation with more flexibility and requires different solution approaches.

11.4 Quantifying Uncertainty of Empirical Optimization

In this section, we aim to answer the first question raised in Sect. 11.2, i.e., how to quantify the impact of input uncertainty on the solution of the empirical optimization problem. Recall that we are considering a special case where F^c belongs to a parametric family. Since the true parameter θ^c is unknown, the maximum likelihood estimator (MLE), denoted by $\hat{\theta}_n$, can be used to estimate θ^c based on the input data ψ^n. We write $\hat{\theta}_n$ as $\hat{\theta}$ when there is no ambiguity. An empirical optimization problem given as follows is then solved as a surrogate of (11.1):

$$\min_{x \in \mathscr{X}} H(x, \hat{\theta}). \tag{11.4}$$

Unless we are very lucky, even the exact solution to (11.4) is only a suboptimal solution to (11.1). It is then of interest to study the true performance of the suboptimal

solution, i.e., how well it performs when it is plugged back into the true objective in (11.1), as well as the optimality gap, i.e., how far its performance deviates from the optimal performance.

One way to achieve the above goal is by the bootstrap resampling method: we bootstrap from ψ^n to get multiple $\hat{\theta}$'s; for each $\hat{\theta}$, we estimate the performance and optimality gap, from which empirical confidence intervals (CI), mean square error, etc., can be computed. Despite broad applicability of the bootstrap approach the computational cost of solving the optimization problem repeatedly may be prohibitively high. To avoid such high computational cost, we would like to exploit the problem structure using the delta method. It is assumed throughout this section that the asymptotic normality of MLE (see e.g., Lemann and Casella 1998) holds, i.e.,

$$\sqrt{n}(\hat{\theta} - \theta^c) \Rightarrow \mathcal{N}(0, [I(\theta^c)]^{-1}) \quad \text{as } n \to \infty, \tag{11.5}$$

where "⇒" denotes convergence in distribution and $I(\theta^c)$ is the Fisher information that a sample from $F(\cdot; \theta^c)$ carries about θ^c. We will show that if the problem possesses certain structure, then it takes *almost no additional computational cost* to characterize the optimality gap and to construct CIs for the true performance of solutions.

11.4.1 Asymptotic Distribution of the Optimality Gap

An important aspect of studying the asymptotics of the optimality gap is to understand the behavior of the solutions to the empirical optimization problem (11.4). Although it is clear that the solutions must depend on the estimate $\hat{\theta}$, we point out some technical difficulties associated with characterizing such dependence. Let us define

$$x_{\text{opt}}(\hat{\theta}) := \arg\min_x H(x, \hat{\theta}) \tag{11.6}$$

as the set of optimal solutions corresponding to a specific parameter $\hat{\theta}$. In general, $x_{\text{opt}}(\cdot)$ could be an ill-behaved set-valued function, which makes it challenging to analyze such quantities as $H(x_{\text{opt}}(\hat{\theta}), \theta^c)$. We plan to avoid the technicalities by restricting to a case where $x_{\text{opt}}(\cdot)$ is a single-valued differentiable function. This is made possible by the following lemma.

Lemma 11.1 (Implicit Function Theorem) *If $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuously differentiable on a neighborhood of $(\bar{x}, \bar{\theta})$, where $f(\bar{x}, \bar{\theta}) = 0$ and $\partial_x f(\bar{x}, \bar{\theta})$ is nonsingular, then there exists an open set U containing $\bar{\theta}$, an open set V containing $\bar{x}$, and a unique continuously differentiable function $x : U \to V$ such that $f(x(\theta), \theta) = 0$ for all $\theta \in U$.*

Lemma 11.1 provides the regularity conditions for the existence of a differentiable solution function of $\hat{\theta}$ on some neighborhood of θ^c. To see how this can facilitate

our analysis, note that the sensitivity of an optimal solution's true performance with respect to (w.r.t.) the estimate $\hat{\theta}$ can be decomposed into two parts: (i) the sensitivity of the objective function w.r.t. the decision variable; (ii) the sensitivity of the optimal solution w.r.t. the parameter. With the differentiability of the optimal solution, the latter is easily captured by the gradient. This motivates us to make the following assumption.

Assumption 11.1 There exists a solution $x^c \in x_{\text{opt}}(\theta^c)$ and an open neighborhood of (x^c, θ^c) on which $\partial_x H$ exists and is continuously differentiable. Moreover, $\partial_x^2 H(x^c, \theta^c)$ is nonsingular.

Assumption 11.1 is interpreted as follows. The fact that $\partial_x H$ is differentiable at (x^c, θ^c) indicates that $H(\cdot, \theta^c)$ is differentiable at x^c. Since x^c falls into an open neighborhood in $\mathscr{X}$, the first-order necessary optimality condition implies $\partial_x H(x^c, \theta^c) = 0$; the second-order necessary optimality condition and the nonsingularity assumption implies $\partial_x^2 H(x^c, \theta^c) \succ 0$. Now lemma 11.1 ensures that there exists an open set U containing θ^c, an open set V containing x^c, and some continuously differentiable function $x^* : U \to V$ satisfying $\partial_x H(x^*(\theta), \theta) = 0$ for all $\theta \in U$ and $x^*(\theta^c) = x^c$. There might be multiple solutions satisfying assumption 11.1, but theoretically we will focus on one particular solution, which is uniquely determined by the function x^*. The weak consistency of $\hat{\theta}$ guarantees that for n large enough, $\hat{\theta}$ will fall into U with a large probability. Therefore, when we let $n \to \infty$, it suffices to consider the case where $\hat{\theta} \in U$. Let

$$g(\theta) := H(x^*(\theta), \theta^c) \tag{11.7}$$

be the true performance of $x^*(\theta)$. The optimality gap is then given by $g(\theta) - g(\theta^c)$. It should be noted that even for an infinitesimal $\Delta\theta$, $x^*(\theta^c + \Delta\theta)$ may not be a global optimum of $H(x, \theta^c + \Delta\theta)$. However, it remains a local optimum due to the continuity of $\partial_x^2 H$. The following theorem will be useful in establishing the asymptotic distribution of the optimality gap.

Lemma 11.2 (Delta Theorem) *If a function $G : \mathbb{R}^m \to \mathbb{R}$ is differentiable at a point $\mu \in \mathbb{R}^m$, and $\{X_n\}$ is a sequence of random variables with $\sqrt{n}(X_n - \mu) \Rightarrow Z$ as $n \to \infty$, where Z is some random variable, then*

$$\sqrt{n}\left(G(X_n) - G(\mu)\right) \Rightarrow \nabla G(\mu)^\intercal Z, \quad \text{as } n \to \infty.$$

The delta theorem (see e.g., Casella and Berger 2002) essentially provides an approach for studying the limiting distribution of a function through Taylor series expansion. Therefore, differentiability is a necessary condition for using this method. Now we are ready to state the asymptotic result of the optimality gap.

Proposition 11.1 *Suppose Assumption 1 holds. Then*

$$\sqrt{n}\left(g(\hat{\theta}) - g(\theta^c)\right) \to 0 \quad \text{in probability as } n \to \infty;$$

furthermore, if g is twice differentiable at θ^c, then

$$n\left(g(\hat{\theta}) - g(\theta^c)\right) \Rightarrow \frac{1}{2} Z^\intercal \nabla^2 g(\theta^c) Z \quad \text{as } n \to \infty, \quad \text{where } Z \sim N(0, [I(\theta^c)]^{-1}). \tag{11.8}$$

Proof Note that g is a composition of $x^*(\cdot)$ and $H(\cdot, \theta^c)$. Assumption 11.1 implies that $H(\cdot, \theta^c)$ is differentiable at $x^*(\theta^c)$, and lemma 11.1 implies that $x^*(\cdot)$ is differentiable at θ^c, so by chain rule g is differentiable at θ^c. Furthermore,

$$\nabla g(\theta^c) = \underbrace{\partial_x H(x^*(\theta^c), \theta^c)}_{=0} \cdot \nabla x^*(\theta^c) = 0$$

by the first-order necessary condition. Thus, the first result follows immediately from Lemma 11.2. If g is twice differentiable at θ^c, we would have the second-order expansion

$$g(\hat{\theta}) - g(\theta^c) = \frac{1}{2}(\hat{\theta} - \theta^c)^\intercal \nabla^2 g(\theta^c)(\hat{\theta} - \theta^c) + e(\hat{\theta}) \|\hat{\theta} - \theta^c\|^2, \tag{11.9}$$

where $e : \Theta \to \mathbb{R}$ satisfies $e(\theta) \to 0$ if $\theta \to \theta^c$. Multiply both sides of (11.9) by n to get

$$n\left(g(\hat{\theta}) - g(\theta^c)\right) = \frac{1}{2}\left(\sqrt{n}(\hat{\theta} - \theta^c)\right)^\intercal \nabla^2 g(\theta^c)\left(\sqrt{n}(\hat{\theta} - \theta^c)\right) + e(\hat{\theta}) \|\sqrt{n}(\hat{\theta} - \theta^c)\|^2.$$

By continuous mapping theorem $\|\sqrt{n}(\hat{\theta} - \theta^c)\|^2 \Rightarrow \|Z\|^2$ as $n \to \infty$. Also, $\hat{\theta} \to \theta^c$ in probability implies that $e(\hat{\theta}) \to 0$ in probability and so does the remainder. We conclude that

$$n\left(g(\hat{\theta}) - g(\theta^c)\right) \Rightarrow \frac{1}{2} Z^\intercal \nabla^2 g(\theta^c) Z \quad \text{as } n \to \infty.$$

By definition, $g(\hat{\theta}) = H(x^*(\hat{\theta}), \theta^c) \geq H(x^*(\theta^c), \theta^c) = g(\theta^c)$ for all $\hat{\theta} \in \Theta$. If g is twice differentiable at θ^c, the second-order necessary condition indicates $\nabla^2 g(\theta^c) \succeq 0$, so (11.8) implies that $n(g(\hat{\theta}) - g(\theta^c))$ converges in distribution to a nonnegative random variable, which is consistent with that the optimality gap should be nonnegative. As a special case, if $\Theta \subset \mathbb{R}$ (11.8) can be rewritten as

$$2nI(\theta^c)\left(g(\hat{\theta}) - g(\theta^c)\right) \Rightarrow g''(\theta^c)\chi^2(1) \quad \text{as } n \to \infty.$$

Recall that $x^*(\theta)$ is defined as a solution to $\min_x H(x, \theta)$ when $\theta \in U$. By Proposition 11.1, $\hat{\theta}$ gets closer to θ^c, and the gap between $g(\hat{\theta})$, the true performance of an approximate solution $x^*(\hat{\theta})$, and $g(\theta^c)$, the true optimal performance, converges to 0 at a rate faster than $1/\sqrt{n}$ in probability. In fact, the rate is $O_p(1/n)$ by (11.8). Ideally, if $g(\hat{\theta})$ is available, we can construct the following asymptotically valid $100(1-\alpha)\%$

CI for $g(\theta^c)$.

$$\lim_{n\to\infty} \mathbb{P}\left(g(\hat{\theta}) - \frac{\gamma_{1-\frac{\alpha}{2}}}{2n} \le g(\theta^c) \le g(\hat{\theta}) - \frac{\gamma_{\frac{\alpha}{2}}}{2n} \right) = 1-\alpha, \tag{11.10}$$

where γ_β is the βth quantile of $Z^\intercal \nabla^2 g(\theta^c) Z$, i.e.

$$\gamma_\beta := \inf\left\{ y \mid \mathbf{P}\left(Z^\intercal \nabla^2 g(\theta^c) Z \le y \right) \ge \beta \right\}. \tag{11.11}$$

The CI in (11.10) is particularly appealing for its shrinking rate $1/n$. Nevertheless, we do not recommend using this CI for $g(\theta^c)$ or $g(\hat{\theta}) - g(\theta^c)$. The reason is that $g(\hat{\theta}) = H(x^*(\hat{\theta}), \theta^c)$ contains the unknown parameter θ^c, so in practice we have to use $H(x^*(\hat{\theta}), \hat{\theta})$ as an approximation. Unfortunately, our empirical experience in numerical experiments shows that such approximation usually introduces a bias that dominates the width of the CI, resulting in a poor coverage probability. Furthermore, even if g is twice differentiable, it is difficult to estimate $\nabla^2 g(\theta^c)$ through simulation. We will explain how to construct confidence intervals in more details in Sect. 11.4.

11.4.2 Asymptotic Distribution of the Performance of Solutions

The next goal is to study the true performance of a given deterministic solution $\hat{x}$, a random solution $x^*(\hat{\theta})$, and the optimal solution $x^*(\theta^c)$ (or simply written as x^*). Their relationship will also help us construct CIs for $g(\hat{\theta})$ and $g(\theta^c)$.

Assumption 11.2 $H(x, \cdot)$ is differentiable at θ^c for all $x \in \mathscr{X}$.

Assumption 11.2 is not too strong in practice, since a parametric family often has a density that is differentiable w.r.t its parameters. Suppose f is the density of F, then $H(x, \theta) = \int h(x, \xi) f(\xi; \theta) d\xi$. As long as h and f are well behaved so that we can interchange differentiation and integration, differentiability of H w.r.t. θ can be guaranteed. The following result follows directly from Lemma 11.2.

Proposition 11.2 *Suppose Assumptions 11.1 and 11.2 hold. Then for a fixed* $\hat{x} \in \mathscr{X}$,

$$\sqrt{n}\left(H(\hat{x}, \hat{\theta}) - H(\hat{x}, \theta^c) \right) \Rightarrow \partial_\theta H(\hat{x}, \theta^c)^\intercal Z, \quad Z \sim N(0, [I(\theta^c)]^{-1}), \tag{11.12}$$

as $n \to \infty$.

From (11.12) we have the following asymptotically valid $100(1-\alpha)\%$ CI for $H(\hat{x}, \theta^c)$.

$$\lim_{n\to\infty} \mathbb{P}\left(H(\hat{x}, \hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma_1}{\sqrt{n}} \le H(\hat{x}, \theta^c) \le H(\hat{x}, \hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma_1}{\sqrt{n}} \right) = 1-\alpha, \tag{11.13}$$

where

$$\sigma_1 := \sqrt{\partial_\theta H(\hat{x}, \theta^c)^\intercal [I(\theta^c)]^{-1} \partial_\theta H(\hat{x}, \theta^c)}, \tag{11.14}$$

and $z_{1-\alpha/2}$ is the $(1-\alpha/2)$th quantile of the standard normal distribution.

With appropriate regularity conditions, it is also possible to study $g(\hat{\theta})$, the true performance of a random approximate solution $x^*(\hat{\theta})$, where the randomness is induced by $\hat{\theta}$. Let

$$\hat{g}(\theta) := H(x^*(\theta), \hat{\theta}) \tag{11.15}$$

be the estimated performance of $x^*(\theta)$. Using this bridging function $\hat{g}$, the asymptotic distribution of $g(\hat{\theta})$ is characterized by the following proposition.

Proposition 11.3 *Suppose Assumptions 11.1 and 11.2 hold, and $\partial_\theta H(\cdot, \theta^c)$ is continuous at $x^*(\theta^c)$. Then*

$$\sqrt{n}\left(\hat{g}(\hat{\theta}) - g(\hat{\theta})\right) \Rightarrow \partial_\theta H(x^*(\theta^c), \theta^c)^\intercal Z, \quad Z \sim N(0, [I(\theta^c)]^{-1}), \tag{11.16}$$

as $n \to \infty$.

Proof Expand $H(x^*(\hat{\theta}), \cdot)$ around θ^c and multiply both sides by $\sqrt{n}$ to get

$$\begin{aligned}\sqrt{n}\left(\hat{g}(\hat{\theta}) - g(\hat{\theta})\right) &= \sqrt{n}\left(H(x^*(\hat{\theta}), \hat{\theta}) - H(x^*(\hat{\theta}), \theta^c)\right) \\ &= \partial_\theta H(x^*(\hat{\theta}), \theta^c)^\intercal [\sqrt{n}(\hat{\theta} - \theta^c)] + \sqrt{n} \cdot o(\|\hat{\theta} - \theta^c\|)\end{aligned}$$

Since $x^*(\cdot)$ is a continuous function and $\partial_\theta H(\cdot, \theta^c)$ is continuous at $x^*(\theta^c)$, the continuous mapping theorem implies that $x^*(\hat{\theta}) \to x^*(\theta^c)$ in probability and thus $\partial_\theta H(x^*(\hat{\theta}), \theta^c) \to \partial_\theta H(x^*(\theta^c), \theta^c)$ in probability. Similar to the proof of Proposition 11.1, the final result follows from that the remainder scaled by $\sqrt{n}$ converges to 0 in probability. □

As a result, we have the following asymptotic $100(1-\alpha)\%$ CI for $g(\hat{\theta})$.

$$\lim_{n\to\infty} \mathbf{P}\left(\hat{g}(\hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma_2}{\sqrt{n}} \le g(\hat{\theta}) \le \hat{g}(\hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma_2}{\sqrt{n}}\right) = 1 - \alpha, \tag{11.17}$$

where

$$\sigma_2 := \sqrt{\partial_\theta H(x^*(\theta^c), \theta^c)^\intercal [I(\theta^c)]^{-1} \partial_\theta H(x^*(\theta^c), \theta^c)}. \tag{11.18}$$

Lastly, the asymptotic distribution of a random solution's estimated performance $\hat{g}(\hat{\theta})$, is given by the next proposition.

Proposition 11.4 *Suppose Assumption 11.1 holds and H is differentiable at $(x^*(\theta^c), \theta^c)$. Then*

$$\sqrt{n}\left(\hat{g}(\hat{\theta}) - g(\theta^c)\right) \Rightarrow \partial_\theta H(x^*(\theta^c), \theta^c)^\intercal Z, \quad Z \sim N(0, [I(\theta^c)]^{-1}). \tag{11.19}$$

Proof Note that $\hat{g}(\hat{\theta}) - g(\theta^c) = H(x^*(\hat{\theta}), \hat{\theta}) - H(x^*(\theta^c), \theta^c)$, where the two terms in the difference can be viewed as the function $H(x^*(\theta), \theta) : \theta \mapsto H$ evaluated at $\hat{\theta}$ and θ^c, respectively. By chain rule and the first-order necessary condition,

$$\nabla_\theta H(x^*(\theta^c), \theta^c) = \underbrace{\partial_x H(x^*(\theta^c), \theta^c)}_{=0} \nabla x^*(\theta^c) + \partial_\theta H(x^*(\theta^c), \theta^c) = \partial_\theta H(x^*(\theta^c), \theta^c). \quad (11.20)$$

The rest follows from Lemma 11.2. □

Proposition 11.4 is a special case of a result in stochastic programming based on Hadamard directional differentiability (see Theorem 3.2 in Shapiro 1991). We remind the reader that

$$g(\cdot) := H(x^*(\cdot), \theta^c), \quad \hat{g}(\cdot) := H(x^*(\cdot), \hat{\theta}),$$

i.e., $g(\cdot)$ is the true performance of $x^*(\cdot)$, $\hat{g}(\cdot)$ is its estimated performance, and in particular $g(\theta^c)$ is the true optimal value. Now, we summarize from Propositions 11.2–11.4, the triangular relation between $\hat{g}(\hat{\theta})$, $g(\hat{\theta})$ and $g(\theta^c)$ as follows:

1. $\sqrt{n}\left(g(\hat{\theta}) - g(\theta^c)\right) \Rightarrow 0$
2. $\sqrt{n}\left(\hat{g}(\hat{\theta}) - g(\hat{\theta})\right) \Rightarrow \partial_\theta H(x^*(\theta^c), \theta^c)^\intercal Z$
3. $\sqrt{n}\left(\hat{g}(\hat{\theta}) - g(\theta^c)\right) \Rightarrow \partial_\theta H(x^*(\theta^c), \theta^c)^\intercal Z$

It can be seen that $(\hat{g}(\hat{\theta}) - g(\hat{\theta}))$ and $(\hat{g}(\hat{\theta}) - g(\theta^c))$ have the same asymptotic distribution because $(g(\hat{\theta}) - g(\theta^c))$ goes to 0 faster than the other two differences. This means the asymptotic CIs for $g(\hat{\theta})$ and $g(\theta^c)$ will be the same, i.e., we cannot distinguish $g(\hat{\theta})$ and $g(\theta^c)$ statistically when we only consider first-order Taylor expansion. In theory, we can expand H to higher orders to distinguish $g(\hat{\theta})$ and $g(\theta^c)$. However, not only would this require higher order differentiability on H, it is also hard to accurately estimate the higher order derivatives using simulation. Furthermore, the CIs for $g(\hat{\theta})$ and $g(\theta^c)$ should account for the fact that $g(\hat{\theta})$ is an upper bound of $g(\theta^c)$, which further complicates the issue. For these reasons, we acknowledge the restriction of the delta method and will not attempt to account for the optimality gap, i.e., we will use the same CIs for $g(\hat{\theta})$ and $g(\theta^c)$ in the numerical experiment.

Another restriction of the delta method is that it only relies on the local performance of a function, thus it can only characterize those performance measures that do not require global properties of the function. For example, the asymptotic distribution of $(\hat{g}(\hat{\theta}) - g(\hat{\theta}))$ depends only on the structure of g on a small neighborhood of θ^c. If we use the delta method to study the mean or the variance of $(\hat{g}(\hat{\theta}) - g(\hat{\theta}))$, without appropriate global assumptions on g (such as boundedness) and on $\hat{\theta}$ (such as uniform integrability), the remainder term in Taylor expansion may not converge in L^2 as $n \to \infty$.

11.4.3 Multiple Sources of Stochastic Uncertainty

In all previous sections, the stochastic uncertainty in the simulation model is summarized by a random vector ξ, where the input data for estimating ξ's distribution are essentially i.i.d. copies of ξ. In more general settings, ξ consists of several subvectors, where each one of them represents an independent source of stochastic uncertainty. For example, a queueing model may have $\xi = (\xi^1, \xi^2)$, where ξ^1 is the customer arrival time and ξ^2 is the service time. More generally, in this section we consider $\xi = (\xi^1, \xi^2, \ldots, \xi^l)$, where the cumulative distribution function (c.d.f.) of each ξ^i is given by $F_i(\cdot; \theta_i^c)$.

Assumption 11.3 The random vectors $\xi^1, \xi^2, \ldots, \xi^l$ are independent, and their datasets are also independent across different ξ^i's.

Assumption 11.3 implies that the input models are mutually independent across different sources of stochastic uncertainty. Now, the distribution of ξ is given by

$$F(\xi; \theta^c) := \prod_{i=1}^{l} F_i(\xi^i; \theta_i^c), \tag{11.21}$$

where $\theta^c := (\theta_1^c, \ldots, \theta_l^c)^\intercal$. In principle, we can still view ξ as a single source of stochastic uncertainty with distribution $F(\cdot; \theta^c)$. However, the delta method cannot be applied directly, because for any two sources of stochastic uncertainty, the sizes of their datasets can be different. Let $\hat{\theta}_{in_i}$ denote the MLE for θ_i^c when the data of ξ^i is of size n_i, and let $n = \sum_{i=1}^{l} n_i$ be the total size of the input data. Then, the MLE for θ^c is given by $\hat{\theta}_n := (\hat{\theta}_{1n_1}, \hat{\theta}_{2n_2}, \ldots, \hat{\theta}_{ln_l})^\intercal$. The classical delta method does not account for multiple n_i's, so in order to establish asymptotic distributions for $H(\hat{x}, \theta^c)$, $g(\hat{\theta})$ and $g(\theta^c)$, we need to assume that n_i's go to infinity in some pattern.

Assumption 11.4 $n_i = w_i n$, $\sum_{i=1}^{l} w_i = 1$, and $w_i > 0$ for all i.

With Assumption 11.4, we have $n_i \to \infty$ for all i as $n \to \infty$, ensuring the asymptotic normality for each $\hat{\theta}_{in_i}$. Of course, in practice it is very unlikely for n_i to increase in such a special pattern. But we can view the real data as a snapshot of the data collection process, where each n_i grows proportionally to n and the ratio is determined by n_i/n. Under Assumption 11.4, we have

$$\sqrt{n}(\hat{\theta}_n - \theta^c) = \sqrt{n}\begin{pmatrix} \hat{\theta}_{1n_1} - \theta_1^c \\ \hat{\theta}_{2n_2} - \theta_2^c \\ \vdots \\ \hat{\theta}_{ln_l} - \theta_l^c \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{w_1}}\sqrt{n_1}\left(\hat{\theta}_{1n_1} - \theta_1^c\right) \\ \vdots \\ \frac{1}{\sqrt{w_l}}\sqrt{n_l}\left(\hat{\theta}_{ln_l} - \theta_l^c\right) \end{pmatrix}, \tag{11.22}$$

where each component converges in distribution to $N(0, [I_i(\theta_i^c)]^{-1}/w_i)$ as $n \to \infty$. By the independence of $\hat{\theta}_{in_i}$'s, we can show (using characteristic function) that

$\sqrt{n}\left(\hat{\theta}_n - \theta^c\right) \Rightarrow \tilde{Z}$, where $\tilde{Z} \sim N(0, [\tilde{I}(\theta^c)]^{-1})$ and

$$\tilde{I}(\theta^c) := \begin{pmatrix} w_1 I_1(\theta_1^c) & & \\ & \ddots & \\ & & w_l I_l(\theta_l^c) \end{pmatrix} \tag{11.23}$$

is a block diagonal matrix. Since the asymptotic normality of $\hat{\theta}_n$ is maintained, Propositions 11.1–11.4 and the CIs remain valid as long as we substitute Z and I with $\tilde{Z}$ and $\tilde{I}$, respectively.

To gain a little more insight, we explicitly compute the CI for $H(\hat{x}, \theta^c)$ as an example. In particular, the variance of the asymptotic distribution is now given by

$$\tilde{\sigma}^2 := \partial_\theta H(\hat{x}, \theta^c)^\intercal [\tilde{I}(\theta^c)]^{-1} \partial_\theta H(\hat{x}, \theta^c), \tag{11.24}$$

which, by partitioning $\partial_\theta H(\hat{x}, \theta^c)$ as $(\partial_{\theta_1} H(\hat{x}, \theta^c), \ldots, \partial_{\theta_l} H(\hat{x}, \theta^c))^\intercal$, is equal to

$$\sum_{i=1}^{l} \frac{1}{w_i} \partial_{\theta_i} H(\hat{x}, \theta^c)^\intercal [I_i(\theta_i^c)]^{-1} \partial_{\theta_i} H(\hat{x}, \theta^c). \tag{11.25}$$

Let

$$\tilde{\sigma}_i^2 := \partial_{\theta_i} H(\hat{x}, \theta^c)^\intercal [I_i(\theta_i^c)]^{-1} \partial_{\theta_i} H(\hat{x}, \theta^c), \tag{11.26}$$

and define a function

$$H_i(\theta_i) := H(\hat{x}, \theta) \mid_{\theta=(\theta_1^c, \ldots, \theta_i, \ldots, \theta_l^c)}, \tag{11.27}$$

i.e., H_i is just H evaluated at $(\hat{x}, \theta)$ where each component of θ coincides with θ^c except for the ith component. Then $\tilde{\sigma}_i^2$ is exactly the variance of the asymptotic distribution of $\sqrt{n}\left(H_i(\theta_i) - H_i(\theta_i^c)\right)$, and it depends on two terms: (i) the Fisher information $I_i(\theta_i^c)$ that captures the uncertainty of the MLE; (ii) the partial derivative $\partial_{\theta_i} H(\hat{x}, \theta^c)$ that captures the sensitivity of the optimization problem to the estimation error of θ^c. Now the normalized asymptotic standard deviation is

$$\frac{\tilde{\sigma}}{\sqrt{n}} = \sqrt{\sum_{i=1}^{l} \frac{1}{w_i n} \tilde{\sigma}_i^2} = \sqrt{\sum_{i=1}^{l} \frac{1}{n_i} \tilde{\sigma}_i^2} \tag{11.28}$$

since $n_i = w_i n$. Thus, in a very loose sense, the normalized asymptotic variance $\tilde{\sigma}^2/n = \sum_{i=1}^{l} \tilde{\sigma}_i^2/n_i$, where each $\tilde{\sigma}_i^2/n_i$ can be viewed as the individual contribution of the ith input model to the total input uncertainty in simulation optimization. Note that for a fixed $\tilde{\sigma}_i$ the smaller n_i is, the larger the variance $\tilde{\sigma}_i^2/n_i$ becomes and hence a greater impact on $\tilde{\sigma}^2/n$. This agrees with our intuition that the total input uncertainty often can be attributed to those input models that are estimated from small datasets.

11.4.4 Constructing Confidence Intervals

In the derivation of the theoretical CIs, we do not consider the effect of stochastic uncertainty. This may not be an issue if we can run a large number of replications to average out the stochastic noise. However, if the simulation is expensive and we only have a limited computational budget, the stochastic uncertainty will dominate the input uncertainty, and ignoring the stochastic uncertainty will lead to a poor coverage probability for the CIs. Therefore, it is sometimes necessary to account for the stochastic uncertainty when constructing CIs.

For simplicity, we will illustrate the idea through the single source case, but it can be easily extended to the multiple source case. First of all, the CIs given by (11.13) and (11.17) both involve quantities that must be estimated. The CI for $H(\hat{x}, \theta^c)$ is

$$\left(H(\hat{x}, \hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma_1}{\sqrt{n}}, \quad H(\hat{x}, \hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma_1}{\sqrt{n}} \right), \tag{11.29}$$

where

$$\sigma_1 = \sqrt{\partial_\theta H(\hat{x}, \theta^c)^\intercal [I(\theta^c)]^{-1} \partial_\theta H(\hat{x}, \theta^c)}. \tag{11.30}$$

Since we do not know θ^c, we can only approximate $\partial_\theta H(\hat{x}, \theta^c)$ with $\partial_\theta H(\hat{x}, \hat{\theta})$, and approximate $I(\theta^c)$ with $I(\hat{\theta})$. Similarly, the CI for $g(\hat{\theta})$ and $g(\theta^c)$ is

$$\left(\hat{g}(\hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma_2}{\sqrt{n}}, \quad \hat{g}(\hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma_2}{\sqrt{n}} \right), \tag{11.31}$$

where

$$\sigma_2 = \sqrt{\partial_\theta H(x^*(\theta^c), \theta^c)^\intercal [I(\theta^c)]^{-1} \partial_\theta H(x^*(\theta^c), \theta^c)}. \tag{11.32}$$

In practice we will approximate $\hat{g}(\hat{\theta})$ with $H(\hat{x}, \hat{\theta})$, $\partial_\theta H(x^*(\theta^c), \theta^c)$ with $\partial_\theta H(\hat{x}, \hat{\theta})$, and $I(\theta^c)$ with $I(\hat{\theta})$. When $\hat{\theta}$ is close enough to θ^c such that $\hat{\theta} \in U$, if the problem has a unique global optimum, then $x^*(\hat{\theta})$ is exactly $\hat{x}$. Otherwise, there is no theoretical guarantee on the quality of such an approximation, and we do not recommend using this CI when the size of input data is too small (e.g., less than 20).

Notice that after approximation the CIs for $H(\hat{x}, \theta^c)$ and $g(\theta^c)$ both reduce to the same empirical CI. This inability to distinguish different CIs is due to the difficulty in estimating θ^c and $x^*(\theta^c)$. So in implementation the following CI is the only CI we can compute.

$$\left(H(\hat{x}, \hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}}, \quad H(\hat{x}, \hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}} \right), \tag{11.33}$$

where

$$\hat{\sigma} = \sqrt{\partial_\theta H(\hat{x}, \hat{\theta})^\intercal [I(\hat{\theta})]^{-1} \partial_\theta H(\hat{x}, \hat{\theta})} \tag{11.34}$$

To compute the CI (11.33), we can compute the Fisher information $I(\hat{\theta})$ analytically since we know the explicit form of F. Also, $H(\hat{x}, \hat{\theta})$ can be estimated by the sample mean

$$\hat{H}(\hat{x}, \hat{\theta}) = \frac{1}{N} \sum_{i=1}^{N} h(x, \xi_i), \tag{11.35}$$

where ξ_i's are i.i.d. samples drawn from $F(\cdot; \hat{\theta})$. However, $\hat{H}(\hat{x}, \hat{\theta})$ is subject to stochastic uncertainty since N is finite, which leads to an error possibly dominating the CI's half-width, and the coverage probability will be way below the target value. To account for this error, one way is to construct a CI for $H(\hat{x}, \hat{\theta})$ and then combine it with the original CI (11.33). The central limit theorem provides the following asymptotic $100(1-\alpha)\%$ CI for $H(\hat{x}, \hat{\theta})$.

$$\lim_{N\to\infty} \mathbf{P}\left(\hat{H}(\hat{x}, \hat{\theta}) - z_{1-\frac{\alpha}{2}} \frac{\sigma_h}{\sqrt{N}} \le H(\hat{x}, \hat{\theta}) \le \hat{H}(\hat{x}, \hat{\theta}) + z_{1-\frac{\alpha}{2}} \frac{\sigma_h}{\sqrt{N}} \right) = 1 - \alpha,$$

where

$$\sigma_h := \mathrm{Var}\left(h(x, \xi)\right), \quad \xi \sim F(\cdot; \hat{\theta})$$

and σ_h can be estimated via the sample variance. By Boole's inequality we have the following merged CI.

$$\left(\hat{H}(\hat{x}, \hat{\theta}) - z_{1-\frac{\alpha}{4}} \frac{\sigma_h}{\sqrt{N}} - z_{1-\frac{\alpha}{4}} \frac{\hat{\sigma}}{\sqrt{n}}, \quad \hat{H}(\hat{x}, \hat{\theta}) + z_{1-\frac{\alpha}{4}} \frac{\sigma_h}{\sqrt{N}} + z_{1-\frac{\alpha}{4}} \frac{\hat{\sigma}}{\sqrt{n}} \right) \tag{11.36}$$

Since (11.36) is the combination of two CIs, it is likely to result in over coverage. Now, we are left to compute $\partial_\theta H(\hat{x}, \hat{\theta})$. By definition,

$$\partial_\theta H(\hat{x}, \hat{\theta}) = \frac{\partial}{\partial \theta} \int h(\hat{x}, \xi) F_{\hat{\theta}}(d\xi),$$

where F_θ is short for $F(\cdot; \theta)$. In a general case where F may not have a density, we can use finite difference to estimate this partial derivative. When F has a density that is differentiable w.r.t. its parameters, we can use the likelihood ratio (LR) method to estimate $\partial_\theta H(\hat{x}, \hat{\theta})$, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N} h(x,\xi_i)\frac{\partial_\theta f(\xi_i;\theta)}{f(\xi_i;\theta)}, \quad \xi_i \sim^{\text{i.i.d.}} f(\cdot;\theta) \tag{11.37}$$

is an unbiased and strongly consistent estimator for $\partial_\theta H(x,\theta)$. We refer the reader to Rubinstein and Shapiro (1993) for extensive discussions about the conditions of applying LR and the associated performance guarantees. Compared with the finite difference method computing (11.37) requires no extra simulation. Indeed, the computation of CI involves estimating $H(\hat{x},\hat{\theta})$ using multiple replications, so we always have a stream of i.i.d. observations of h available. The only thing we need to compute is the likelihood ratio, which takes very little computational cost. Consequently, by taking advantage of the problem structure the likelihood ratio approach accomplishes the same goal more efficiently.

11.4.5 Numerical Illustration: Comparison of Confidence Intervals

To demonstrate the performance of the proposed CIs, we revisit the newsvendor problem mentioned in Sect. 11.1.1. Recall that the cost function is

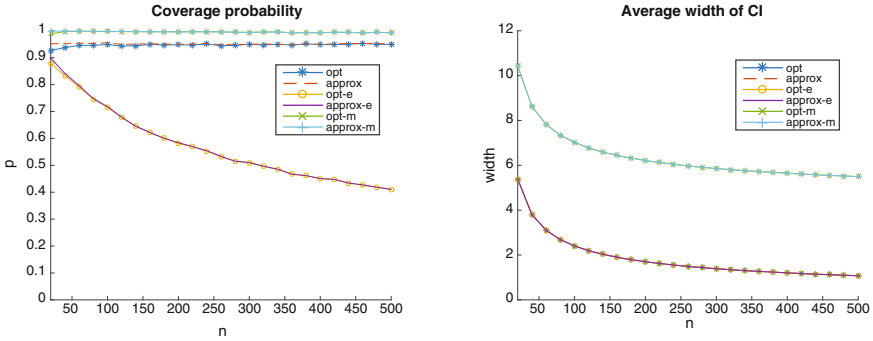
$$h(x,\xi) = cx - p\min(x,\xi),$$

where c is the unit cost, p is the unit price, and ξ is the demand. Following the same settings as in Sect. 11.1.1, we set $p = 2, c = 1$ and assume ξ is exponentially distributed with rate 20. For a given θ, the closed forms of the objective function and the optimal solution are given as follows.

$$H(x,\theta) = cx - \frac{p}{\theta}(1 - e^{-\theta x}), \quad x^*(\theta) = \frac{1}{\theta}\ln\left(\frac{p}{c}\right).$$

To compare the CIs we proposed, we use 10,000 replications to estimate their coverage probabilities and average CI widths. Figure. 11.2 summarizes the performance of the following CIs.

1. Opt CI: the CI for $g(\theta^c)$ (i.e., $H(x^*(\theta^c),\theta^c)$) given by (11.33);
2. Approx CI: the CI for $g(\hat{\theta})$ (i.e., $H(x^*(\hat{\theta}),\theta^c)$) given by (11.33);
3. Opt-e CI: implementing the opt CI using a small replication number $N = 100$, where 'e' stands for error;
4. Approx-e CI: implementing the approx CI using a small replication number N;
5. Opt-m CI: the CI for $g(\theta^c)$ given by (11.36), where 'm' stands for merge;
6. Approx-m CI: the CI for $g(\hat{\theta})$ given by (11.36);



(a) Coverage probability (b) Average CI width

Fig. 11.2 Performance of CIs

First of all, observe that the Opt CI attains the target probability, but the Approx CI is a little below the target. This is because we use the same CI for $g(\theta^c)$ and $g(\hat{\theta})$ while there is a difference between them. Second, the coverage probabilities of the Opt-e CI and the Approx-e CI decreases as n increases, because with only 100 replications the simulation incurs a large stochastic uncertainty that dominates the input uncertainty. Finally, the Opt-m CI and the Approx-m CI take into account the stochastic uncertainty and are able to avoid under coverage, though at the cost of larger widths and over coverage since we are merging two CIs via Boole's inequality in (11.36).

11.5 Simulation Optimization Under Input Uncertainty

In this section we attempt to address the second question raised in Sect. 11.2, i.e., how to "optimize" the simulation system when taking into account input uncertainty. As illustrated in the newvendor example, the optimal solution under the estimated input model may perform poorly for the actual system. Robust optimization tries to hedge against the extreme possible scenarios by optimizing the worst-case performance. In our setting, the distributionally robust optimization (DRO) approach optimizes the system under the worst-case input model scenario, i.e.,

$$\min_{x\in\mathscr{X}} \max_{\theta\in\tilde{\Theta}} H(x,\theta),$$

where $\tilde{\Theta}$ is an ambiguity set that includes possible input parameter values with certain probabilistic guarantee. The ideal choice of the ambiguity set should be the minimal set that achieves the target probabilistic guarantee, the ambiguity set should also be tractable given the objective function and the constraint set are tractable. Constructing such an ideal ambiguity set can be challenging or even impossible, and

there have been various ways to construct different ambiguity sets (see Sect. 11.3 and references therein). Despite the fact that DRO is a very popular approach, Hedging against the worst case might be conceptually too conservative, especially when the extreme case happens rarely in reality. Although the conservativeness can be alleviated by setting a smaller probabilistic guarantee for the ambiguity set, the robust optimization approach still looks at the worst case within the ambiguity set and overlooks the likelihood information we have about the possible scenarios. To accommodate more risk attitudes and take advantage of the likelihood information about the input uncertainty, a new framework of Bayesian risk optimization has been recently proposed by Zhou and Xie (2015) and Wu et al. (2016). In the following we will a give a detailed overview of the Bayesian risk optimization approach.

11.5.1 Bayesian Risk Optimization

Recall that we are given $\psi^n = (\xi_1, \ldots, \xi_n)$, which are n i.i.d. data from $F(\cdot; \theta^c)$. To estimate θ^c, we take a Bayesian viewpoint and assume that θ^c is a realization of a belief r.v. θ, whose distribution follows a chosen prior $\pi_0(\cdot)$. The Bayesian posterior distribution $\pi_n(\cdot)$ is given by

$$\pi_n(\theta) := p(\theta|\psi^n) \propto \pi_0(\theta)p(\psi^n|\theta),$$

where $p(\psi^n|\theta)$ is the likelihood of data ψ^n, and the notation $\propto$ denotes equivalence up to a normalization constant. The Bernstein–von Mises theorem (also known as the Bayesian central limit theorem, see Gelman et al. 2014) ensures that under some regularity conditions (notably that θ^c is an interior point of Θ), as the data size $n \to \infty$, the posterior distribution approaches normality with mean θ^c and variance $\{nI(\theta^c)\}^{-1}$, where $I(\theta^c)$ is the Fisher information at θ^c. It implies the posterior distribution will become more and more concentrated on the true parameter value θ^c as data size n increases. It is consistent with our intuition that the input uncertainty should decrease as we have more input data, and in the extreme case when we have an infinite amount of input data we should recover the true input distribution.

The posterior distribution $\pi_n(\cdot)$ characterizes the probabilistic structure over the parameter space Θ based on available data and the chosen prior. Let θ_n denote a r.v. that follows the posterior distribution π_n. The Bayesian risk formulation for simulation optimization is as follows:

$$\min_{x \in \mathscr{X}} \rho \left\{ H(x, \theta_n) \right\}, \tag{11.38}$$

where ρ is a risk measure that maps the random variable $H(x, \theta_n)$ (induced by the r.v. θ_n) to an extended real number, and thus measures the risk of $H(x, \theta_n)$ due to the uncertainty in θ_n. Generally speaking, a risk measure satisfies the following two conditions Hans and Schied (2002): (i) monotonicity: for two r.v.s X and Y, $X \leq Y$ almost surely (a.s.) implies $\rho(X) \leq \rho(Y)$; (ii) translation invariance: for a r.v. X and

a constant $a \in \mathbb{R}$, $\rho(X + a) = \rho(X) + a$. Numerous choices of risk measures can be applied to (11.38), and here we consider the following formulations.

(i) Mean/Mean-variance with weight parameter $c \geq 0$: $\min_{x \in \mathscr{X}} \mathbf{E}\left[H(x, \theta_n)\right] + c\text{Var}\left[H(x, \theta_n)\right]$.
(ii) VaR (Value-at-Risk) with risk level $\alpha \in (0, 1)$: $\min_{x \in \mathscr{X}} \text{VaR}_\alpha\left[H(x, \theta_n)\right]$.
(iii) CVaR (Conditional-Value-at-Risk) with risk level $\alpha \in (0, 1)$: $\min_{x \in \mathscr{X}} \text{CVaR}_\alpha\left[H(x, \theta_n)\right]$.

In particular, VaR and CVaR are two commonly used risk measures in financial industry for controlling large losses. For a r.v. X, $\text{VaR}_\alpha(X)$ is defined as the α-quantile of X, i.e., $\text{VaR}_\alpha(X) := \inf\{t : \mathbb{P}(X \leq t) \geq \alpha\}$, and CVaR is defined as the average large loss, i.e., $\text{CVaR}_\alpha(X) := \frac{1}{1-\alpha}\int_\alpha^1 \text{VaR}_r(X)dr$. When $\mathbb{P}(X = \text{VaR}_\alpha(X)) = 0$, CVaR can also be written as a conditional expectation $\text{CVaR}_\alpha(X) := \mathbf{E}[X \mid X \geq \text{VaR}_\alpha(X)]$. To avoid unnecessary technicalities, we assume that CVaR can be written in the form of conditional expectation. CVaR is a coherent risk measure Artzner et al. (1999) and enjoys some nice properties such as convexity. Though not coherent, mean-variance and VaR are also commonly-used risk measures. When the weight parameter $c = 0$, the mean-variance formulation reduces to the risk-neutral mean formulation.

We point out the connection between the proposed framework (11.38) and some existing formulations. The mean formulation parallels the Bayesian model averaging approach taken by Zouaoui and Wilson (2003) and Chick (2001) for performance evaluation, which averages out both input uncertainty caused by θ and stochastic uncertainty caused by ξ. The VaR formulation, when α set to be 100% and the posterior distribution is assumed to have a bounded support $\tilde{\Theta}$, reduces to DRO with the ambiguity set $\tilde{\Theta}$, i.e., $\min_{x \in \mathscr{X}} \text{VaR}_{100\%} H(x, \theta_n) = \min_{x \in \mathscr{X}} \max_{\theta \in \tilde{\Theta}} H(x, \theta)$. From this viewpoint, DRO can be viewed as a special case of the Bayesian risk framework by taking the risk measure ρ as the worst-case measure. When ρ is a coherent risk measure such as CVaR, the proposed formulation can also be recast as a DRO with an appropriately constructed ambiguity set. However, compared with constructing ambiguity sets in DRO, adjusting the risk level α of a chosen risk measure is probably a more intuitive and flexible way to accommodate the decision maker's risk preference.

11.5.1.1 Consistency of Objective Function and Optimal Solutions

To see the validity of the Bayesian risk optimization framework (11.38), we need to answer two questions: (i) does the objective function $\rho[H(x, \theta_n)]$ converge (in certain probabilistic sense) to the true objective function $H(x, \theta^c)$ as $n \to \infty$? (ii) does the set of minimizers converge (in certain probabilistic sense) to the minimizers of $H(x, \theta^c)$ as $n \to \infty$? Affirmative answers to these questions will give us assurance that the Bayesian risk formulations approach the true problem when the size of input data increases, and show that ideally when the input data size is infinity we will solve exactly the true problem. Zhou and Xie (2015) showed pointwise convergence of

$\rho[H(x,\theta_n)]$ to the true objective function $H(x,\theta^c)$ in probability ($F(\cdot;\theta^c)$) as $n \to \infty$. Wu et al. (2016) strengthened the result and showed the convergence is in the almost sure sense under slightly stronger regularity conditions, i.e., pointwise strong consistency of $\rho[H(x,\theta_n)]$ with the true objective function $H(x,\theta^c)$. This result guarantees the convergence for any given set of input data. Wu et al. (2016) further showed that the optimal solution(s) of (11.38) converge to the true optimal solutions. More specifically, let S_n be the set of optimal solutions of a risk formulation (11.38) when we have n i.i.d. input data, and let S be the set of optimal solutions to (11.1). If S_n and S are not singletons, the convergence of solutions will be in the mode of set convergence, which is defined in terms of the following "distance" between S_n and S: for two sets A and B in $\mathscr{X}$, define the deviation of A from B as $\mathbb{D}(A,B) = \sup_{x\in A} \text{dist}(x,B)$, where $\text{dist}(x,B) := \inf_{y\in B} \|x-y\|$. Then, (Wu et al., 2016) has the following convergence result under certain regularity conditions: $\mathbb{D}(S_n,S) \to 0$ a.s. ($F(\cdot;\theta^c)$) for almost all $\theta^c \in \Theta$ possibly except for a subset of measure 0 relative to π_0.

11.5.1.2 Asymptotic Normality and Insight of Bayesian Risk Optimization

Another theoretical result established in Wu et al. (2016) is the asymptotic normality of all the Bayesian risk optimization formulations under consideration. Specifically, for a risk measure ρ among the choices considered and a fixed x in $\mathscr{X}$, we want to know if the scaled difference in objective functions $\sqrt{n}\{\rho[H(x,\theta_n)] - H(x,\theta^c)\}$ converges in distribution to some r.v. depending on x. Moreover, we want to find out whether $\sqrt{n}\{\rho[H(x,\theta_n)] - H(x,\theta^c)\}$ converges weakly to a random function of x. These results, similar to the classical central limit theorem, give the asymptotic convergence rate and can be used to construct confidence intervals on the true performance of a given solution and in particular the optimal solution.

Roughly speaking, if H is a well-behaved function, the asymptotic normality of risk formulations is inherited from the asymptotic normality of the posterior distribution. Specifically, we have the following asymptotic normality results for the risk formulations under consideration (for technical details, we refer the reader to Wu et al. 2016):

(i) Mean / Mean-Variance: $\sqrt{n}\left\{\mathbf{E}[H(x,\theta_n)] + c\text{Var}[H(x,\theta_n)] - H(x,\theta^c)\right\} \Rightarrow \mathscr{N}\left(0,\sigma_x^2\right)$.

(ii) VaR: $\sqrt{n}\left\{\text{VaR}_\alpha[H(x,\theta_n)] - H(x,\theta^c)\right\} \Rightarrow \mathscr{N}(\sigma_x\Phi^{-1}(\alpha),\sigma_x^2)$.

(iii) CVaR: $\sqrt{n}\left\{\text{CVaR}_\alpha[H(x,\theta_n)] - H(x,\theta^c)\right\} \Rightarrow \mathscr{N}\left(\frac{\sigma_x}{1-\alpha}\phi(\Phi^{-1}(\alpha)),\sigma_x^2\right)$.

In (i)–(iii), $\mathscr{N}$ stands for a normal distribution, ϕ and Φ are the probability density function (p.d.f.) and c.d.f. of a standard normal distribution, and

$$\sigma_x^2 := \nabla_\theta H(x,\theta^c)^\intercal [I(\theta^c)]^{-1} \nabla_\theta H(x,\theta^c),$$

where $\nabla_\theta H(x, \theta^c)$ is the sensitivity of H at θ^c, and $I(\theta^c)$ is the Fisher information matrix that characterizes the information that ξ carries about θ^c. An immediate consequence of (i)–(iii) is that we can construct confidence intervals (CIs) for the true performance of solutions. More importantly, it reveals an important insight of the proposed framework: *the Bayesian risk formulations essentially capture the trade-off between the expected performance and the variability of the actual performance.* We give more details below.

Take the VaR formulation as an example. The asymptotic normality result above of the VaR formulation can be rewritten as

$$\text{VaR}_\alpha[H(x, \theta_n)] = H(x, \theta^c) + \frac{\nabla_\theta H(x, \theta^c)^\intercal Z}{\sqrt{n}} + \Phi^{-1}(\alpha)\frac{\sigma_x}{\sqrt{n}} + o_p\left(\frac{1}{\sqrt{n}}\right), \quad (11.39)$$

where Z is a standard normal r.v., $o_p(1/\sqrt{n})$ stands for a term whose product with $\sqrt{n}$ converges to 0 in probability ($F(\cdot; \theta^c)$) uniformly in x. The left-hand side of (11.39) is the risk formulation we are minimizing, and the right-hand-side can be viewed as the sum of the true objective $H(x, \theta^c)$ and some error terms. Compared with the mean formulation, which has

$$\mathbf{E}[H(x, \theta_n)] = H(x, \theta^c) + \frac{\nabla_\theta H(x, \theta^c)^\intercal Z}{\sqrt{n}} + o_p\left(\frac{1}{\sqrt{n}}\right),$$

we see that (11.39) has an extra deterministic bias term $\Phi^{-1}(\alpha)\sigma_x/\sqrt{n}$ that vanishes as $n \to \infty$. So the VaR formulation is approximately equivalent to a weighted sum of the posterior mean $\mathbf{E}[H(x, \theta_n)]$ and the bias term $\sigma_x/\sqrt{n}$, where the weight of the bias is $\Phi^{-1}(\alpha)$. Even though the bias diminishes as $n \to \infty$, it has an undeniable impact on the VaR formulation when n is small. In particular, if n is not too large (e.g., 20) and α is large (e.g., 95%), it is possible for the bias term to dominate $\mathbf{E}[H(x, \theta_n)]$ and we are close to solving $\min_{x \in \mathscr{X}} \sigma_x/\sqrt{n}$.

Similar decomposition for the CVaR and mean-variance formulations can be derived in the same way. In summary, we have the following decompositions of the Bayesian risk formulations.

(i) Mean / Mean-Variance: $\mathbf{E}[H(x, \theta_n)] + c\text{Var}[H(x, \theta_n)] = \mathbf{E}[H(x, \theta_n)] + c\frac{\sigma_x^2}{n} + o_p\left(\frac{1}{n}\right)$.

(ii) VaR: $\text{VaR}_\alpha[H(x, \theta_n)] = \mathbf{E}[H(x, \theta_n)] + \Phi^{-1}(\alpha)\frac{\sigma_x}{\sqrt{n}} + o_p\left(\frac{1}{\sqrt{n}}\right)$.

(iii) CVaR: $\text{CVaR}_\alpha\{H(x, \theta_n)\} = \mathbf{E}[H(x, \theta_n)] + \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\frac{\sigma_x}{\sqrt{n}} + o_p\left(\frac{1}{\sqrt{n}}\right)$.

Therefore, *the mean-variance, VaR, and CVaR formulations are approximately equivalent to a weighted sum of the mean formulation and the half-width of the true performance's CI* $\sigma_x/\sqrt{n}$ (or $(\sigma_x/\sqrt{n})^2$), i.e.,

$$\rho\{H(x,\theta_n)\} \approx \mathbf{E}[H(x,\theta_n)] + w\frac{\sigma_x}{\sqrt{n}} \text{ (or } \mathbf{E}[H(x,\theta_n)] + w\frac{\sigma_x^2}{n}),$$

where w, the weight of $\sigma_x/\sqrt{n}$, is controlled by α in the VaR and CVaR formulations, and the weight of $(\sigma_x/\sqrt{n})^2$ is controlled by the constant c in the mean-variance formulation. Interestingly, similar insight has also been observed for DRO by Gotoh et al. (2015), which shows that a large class of robust empirical optimization problems are essentially equivalent to a mean-variance formulation; More remotely related, Lam (2016) showed the robust sensitivity of an expectation with respect to the unknown distribution can be also decomposed as the mean plus a term depending on the standard deviation.

11.5.1.3 Illustrative Example: Insight of Bayesian Risk Optimization

Again, we use the newsvendor problem as an illustrative example. Under the same settings as in Sect. 11.4.5, the limiting standard deviation σ_x can be computed as

$$\sigma_x = \frac{p}{\theta^c}\left|1 - (1+\theta^c x)e^{-\theta x}\right|.$$

which has a unique minimizer 0 on $\mathscr{X} = \mathbb{R}^+$. This means that there is no input uncertainty when $x = 0$. Indeed, the expected loss is always 0 if we do not order anything, regardless of the demand distribution. Our goal is to examine how x_n^* respond to the changes of α and c when n is fixed at 20. We draw 1,000 i.i.d. datasets to estimate the distribution of x_n^*, and the risk formulations are solved using sample average approximation (SAA) Kleywegt et al. (2001). The histograms of x_n^* for the cases of mean-variance, VaR, and CVaR are displayed in Fig. 11.3.

Compared with the left plot of Figure 11.1, we can see that the optimal solutions x_n^* of the risk formulations all shift towards 0, i.e., the minimizer of σ_x. Moreover, as α and c increase, the more the average value of x_n^* shifts towards 0. This aligns with our insights that the mean-variance, VaR and CVaR formulations put more weight on $\sigma_x/\sqrt{n}$ when α and c are larger. However, the magnitude of shift is not very significant. The reason is that $\sigma_x/\sqrt{n}$ is dominated by $\mathbf{E}[H(x,\theta_n)]$, and we have to make the weight very large to see x_n^* approaching 0. Nevertheless, the VaR and CVaR formulations have a limited ability of increasing the weight of $\sigma_x/\sqrt{n}$ due to the light-tailedness of normal distribution. This turns out to be desirable in this example, because a solution close to 0 is too conservative: the main purpose of selling newspapers is to make profit rather than control loss. Therefore, if the goal is to achieve a balance between expected performance and the risk of actual performance, then more effort should be devoted to exploring the structure of the objective function and choosing an appropriate risk formulation as well as the associated parameters.

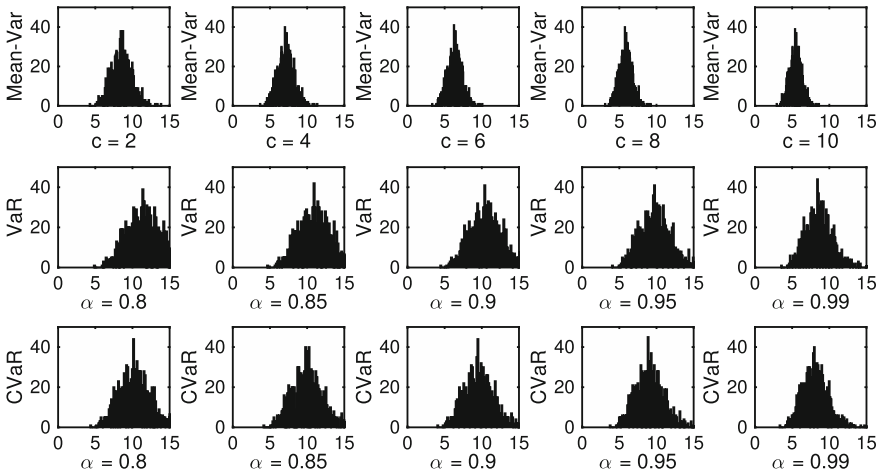


Fig. 11.3 Optimal solution x_n^* to Bayesian risk formulations with different α's or c's

11.6 Empirical Optimization Versus Bayesian Risk Optimization

The previous two sections respectively studied the empirical optimization approach and the Bayesian risk optimization approach to simulation optimization under input uncertainty, and we will compare them in this section. These two approaches deal with input uncertainty quite differently: the empirical optimization approach replaces the unknown input model with an estimated one, while the Bayesian risk optimization approach seeks to characterize and control the risk associated with input uncertainty using Bayesian posterior distributions. Although they both solve approximate versions of the underlying true problem and enjoy certain asymptotic performance guarantees, it is of more practical interest to compare and understand their behavior in the finite-sample case. More specifically, we would like to see how solutions acquired from solving these formulations perform when plugged back into the true objective function. To do so, we use two numerical examples to demonstrate the strengths and the limitations of each formulation. The first example is an M/M/1 queue control problem from Zhou and Xie (2015), and the other is the newsvendor problem we have been using in the previous sections. In the first example, we are asked to minimize the cost for an M/M/1 queue. The arrival rate is θ^c, and the decision variable is the service rate x. The objective function is

$$H(x, \theta^c) = \min\left\{\frac{1}{x-\theta^c} + cx, M\right\} \mathbb{1}_{\{x>\theta^c\}} + M\mathbb{1}_{\{x\le\theta^c\}}, \quad x > 0,$$

where c is the unit cost of service rate, M is a large positive number and $\mathbb{1}_{\{\cdot\}}$ denotes an indicator function. Note that $1/(x-\theta^c)$ is the steady-state average customer waiting time and M is an upper bound on the total cost. One can check that

$x^*(\theta^c) = \theta^c + 1/\sqrt{c}$ is the unique minimizer. We set $\theta^c = 10, c = 1$ and $M = 500$, so the true optimal solution is 11 and the optimal true objective value is 12. Every formulation is applied to this problem over 100 replications, where each replication uses an independent input dataset of size 20. The variance's weight is 0.01 in the mean-variance formulation. We plot the histograms of the solutions and their corresponding true performance in Fig. 11.4a, b, the widths of true performance's CIs in (c), and the objective graphs (based on a random dataset) in Fig. 11.4d. A number of observations are made as follows.

(i) The empirical optimization yields solutions that are closest to the true optimal solution 11, whereas all the risk formulations have solutions roughly centered around 20.
(ii) However, the true objective values of empirical optimal solutions either take small values (≤ 30) or go off to 500, in contrast to the risk formulations which mostly concentrate between 10 to 30. Also, all four risk formulations have much narrower CIs than empirical optimization. Indeed, since $\nabla H(x, \theta^c) = (x - \theta^c)^{-2}$, the farther a solution is to the right of 11, the smaller ∇H will be. This leads to a smaller CI width.
(iii) It can be seen from Fig. 11.4d that the true objective function is steep on the left of 11 and rather flat on the other side. Thus, the error in estimating θ^c may land an empirical optimal solution on the steep part and result in a big cost. The objective graphs of the risk formulations, though similar in shape to the true objective, attain their minima on the flat side of the true objective, lending robustness to their solutions against input uncertainty.

A main message from the M/M/1 example is that if the true objective function has special structures (e.g., different slopes around the optimal solution), and if the risk formulations "reshape" the objective in such a way that the resulting solutions concentrate on a relatively flat area, then one can expect more robustness compared with the empirical optimization approach. Due to the dependence on the problem structure, robustness in such sense is not guaranteed in practice. To see this, we apply the formulations to the newsvendor problem under the same settings as in previous sections. Based on 500 independent replications, the histograms of the solutions, their true objective values and the CI widths of the solutions' true performance are plotted in Fig. 11.5. We may see that

(i) Compared with the empirical and the mean formulation, the mean-variance, VaR, and CVaR formulations tend to produce solutions closer to 0, resulting in overall larger and more widely spread expected losses. As is explained in Sect. 11.5.1.3, this is because those three formulations aim to find a solution in between the optimal solution and the solution whose objective value is least sensitive to input uncertainty, where the latter, in this case, is 0 and is far from the optimum.
(ii) More insights about the performance of risk formulations can be developed from the objective graphs shown in Fig. 11.5c. It can be seen that the true objective is roughly symmetric around the optimum, and there is no drastic change of slopes. Without structural advantage, risk formulations fail to exhibit the same type of robustness as in the M/M/1 queue example.

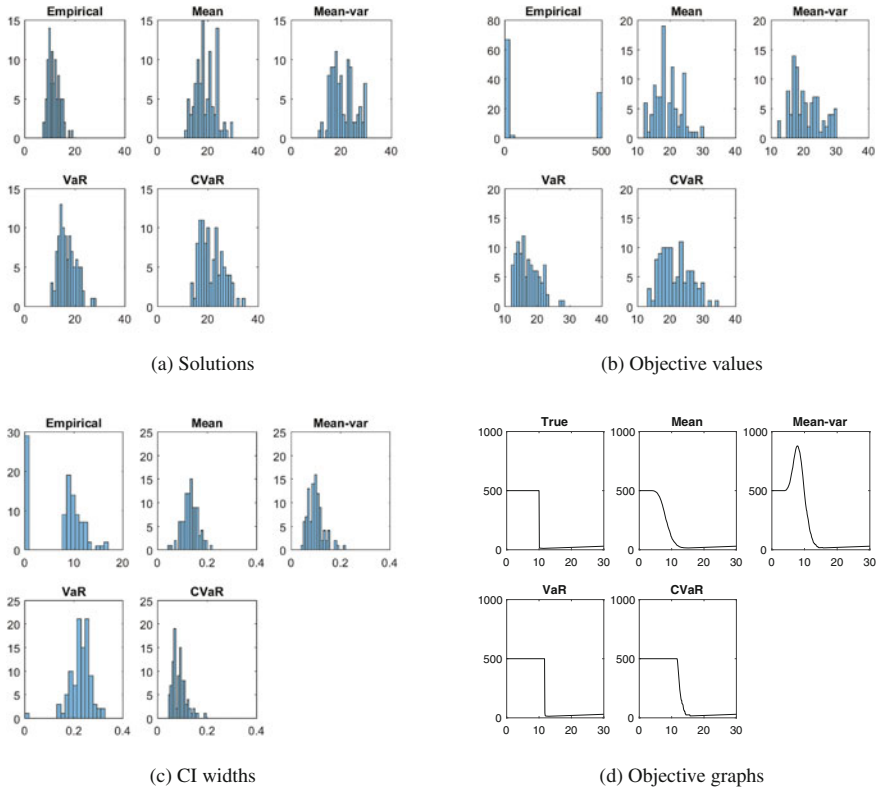
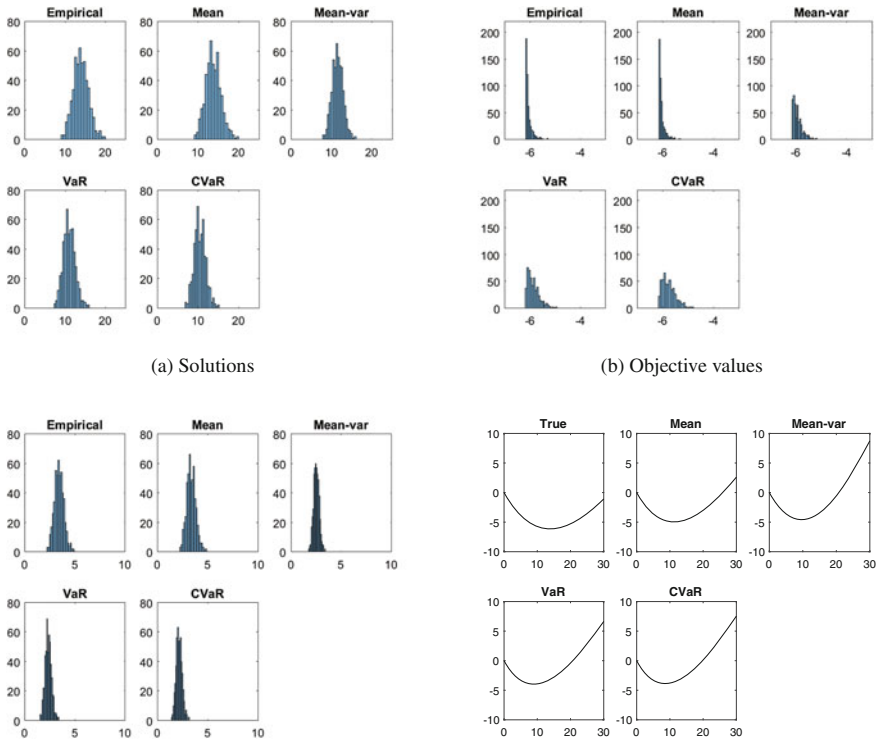


Fig. 11.4 M/M/1 queue problem

(iii) Nevertheless, if we compute the CIs for the solutions' true performance, it is obvious that empirical optimization has a much wider CI than the mean-variance, VaR, and CVaR formulations, which can be seen for both examples (Fig. 11.4c and Fig. 11.5c). This is not surprising since the CI width is proportional to $|\partial_\theta H(x, \theta^c)|$, and the further a solution is from 0, the larger this value becomes and the wider its CI gets.

We must therefore note that the Bayesian risk formulations, statistically speaking, do not necessarily produce solutions with near-optimal true performance. Rather, they account for the uncertainty about a solution's true performance, and use the CI width as a measure of such uncertainty. In particular, the Bayesian risk optimization approach tries to avoid such a scenario where a solution has temptingly good performance under the estimated input model, but has a wide CI and later turns out to perform badly under the true input model. In other words, *by possibly giving up some good true performance, we gain more confidence about the actual performance of a solution*. This is the robustness that the Bayesian risk optimiation approach can provide us.



(a) Solutions

(b) Objective values

(c) CI widths

(d) Objective graphs

Fig. 11.5 Newsvendor problem

11.7 Future Directions

This chapter studies how the input uncertainty affects the solution performance in simulation optimization, and reviews a Bayesian risk optimization approach that captures the trade-off between the expected performance and the variability of the actual performance. Our discussion suggests that even when there is input uncertainty, it is possible to quantify and take into account its impact during optimization. However, we have only touched the surface of this research topic, while many other related problems still remain open. We briefly discuss some of them in the following.

1. **Nonparametric input model**. This chapter focuses only on the case of parametric input models. It is possible to extend the results in this chapter to the nonparametric case, where the input distribution F lives in a general space. To use the Bayesian risk optimization approach for nonparametric input models, one might consider using the Dirichlet process as the nonparametric Bayesian model (Ferguson 1973) in order to have a tractable posterior distribution over a general support.

2. **Efficient algorithms for solving Bayesian risk formulations**. Although the Bayesian risk optimization approach in principle improves the robustness of a solution, efficiently solving the Bayesian risk formulations remains a numerical challenge. If the problem has differentiability or convexity, one possible approach is to use stochastic approximation (a.k.a. stochastic gradient descent), but VaR and CVaR formulations typically set the risk level α close to 1, which requires a large sample size to get a good estimate of the gradient. It is then of interest to explore the structure of the formulations to see if they can be converted into more tractable forms. If gradient information is not available, other methods such as model-based algorithms (see, e.g., Hu et al. (2012); Zhu et al. (2016)) may be more promising choices.
3. **Learning input model during optimization**. Input data are usually deemed expensive and it is often assumed that one can only work with the data s/he is given. In reality, however, there are many scenarios where additional input data can be collected, albeit at the expense of time or money. For example, a sales manager can collect more sales data to learn the demand's distribution better. Therefore, a decision maker needs to balance the input uncertainty and the stochastic uncertainty. On the one hand, "purchasing" more data may improve the input model, but there will be less budget to run simulation; on the other hand, too little input data results in a very inaccurate simulation model, and the error cannot be simulated away. This trade-off between learning and optimization may lead to interesting and practical algorithms for simulation optimization under input uncertainty. Some preliminary work has been done under a fixed budget setting see, e.g., Wu and Zhou (2017).

Acknowledgements The authors' research was partially supported by the National Science Foundation under Grant CAREER CMMI-1453934.

References

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. Oper Res 58(2):371–382

Artzner P, Delbaen F, Eber JM, Heath D (1999) Coherent measures of risk. Math. Finance 9(3):203–228

Barton RR, Nelson BL, Xie W (2013) Quantifying input uncertainty via simulation confidence intervals. INFORMS J Comput 26(1):74–87

Barton RR, Schruben LW (1993) Uniform and bootstrap resampling of empirical distributions. In: Evans GW, Mollaghasemi M, Russell MEC, Biles WE (eds) Proceedings of the 25th conference on Winter simulation. ACM, pp 503–508

Barton RR, Schruben LW (2001) Resampling methods for input modeling. In: Peters BA, Smith JS, Medeiros DJ, Rohrer MW (eds) Proceedings of the 2001 winter simulation conference, vol 1. Piscataway. New Jersey, Institute of Electrical and Electronics Engineers Inc., pp 372–378

Bayraksan G, Love DK (2015) Data-driven stochastic programming using phi-divergences. In: Tutorials in operations research, pp 1–19

Ben-Tal A, Den Hertog D, De Waegenaere A, Melenberg B, Rennen G (2013) Robust solutions of optimization problems affected by uncertain probabilities. Manag Sci 59(2):341–357

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. SIAM Rev 53(3):464–501

Bertsimas D, Gupta V, Kallus N (2013) Data-driven robust optimization. arXiv:1401.0212

Bhatnagar S, Prasad HL, Prashanth LA (2012) Stochastic recursive algorithms for optimization: simultaneous perturbation methods, vol 434. Springer

Casella G, Berger RL (2002) Statistical inference, vol 2. Duxbury Pacific Grove, CA

Cheng RCH, Holloand W (1997) Sensitivity of computer simulation experiments to errors in input data. J Stat Comput Simul 57(1–4):219–241

Chick S (2001) Input distribution selection for simulation experiments: accounting for input uncertainty. Oper Res 49(5):744–758

Corlu CG, Biller B (2013) A subset selection procedure under input parameter uncertainty. In: Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 463–473

Delage E, Ye Y (2010) Distributionally robust optimization under moment uncertainty with application to data-driven problems. Oper Res 58(3):595–612

Esfahani PM, Kuhn D (2015) Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. arXiv:1505.05116

Fan W, Hong JL, Zhang X (2013) Robust selection of the best. Pasupathy R, Kim SH, Tolk A, Hill R, Kuhl ME(eds) Proceedings of the 2013 winter simulation conference. Piscataway, New Jersey, Institute of Electrical and Electronics Engineers Inc, pp 868–876

Ferguson TS (1973) A bayesian analysis of some nonparametric problems. Ann Stat 1(2): 209–230

Gao S, Xiao H, Zhou E, Chen W (2017) Robust ranking and selection with optimal computing budget allocation. Automatica 81:30–36

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2014) Bayesian data analysis. Chapman & Hall

Gotoh J, Kim MJ, Lim A (2015) Robust empirical optimization is almost the same as mean-variance optimization. Available at SSRN 2827400

Hans F, Schied A (2002) Stochastic finance: an introduction in discrete time

Hu J, Wang Y, Zhou E, Fu MC, Marcus SI (2012) A survey of some model-based methods for global optimization. In: optimization, control, and applications of stochastic systems, Springer, pp 157–179

Jiang R, Guan Y (2015) Data-driven chance constrained stochastic program. Math Program, pp 1–37

Kleywegt A, Shapiro A, Homem de Mello T (2001) The sample average approximation method for stochastic discrete optimization. SIAM J Optim 12:479–502

Lam H (2016) Robust sensitivity analysis for stochastic systems. Math Oper Res 41(4):1248–1275

Lemann EL, Casella G (1998) Theory of point estimation. Springer, New York

Popescu I (2007) Robust mean-covariance solutions for stochastic optimization. Oper Res 55(1):98–112

Rubinstein RY , Shapiro A (1993) Discrete event systems: sensitivity analysis and stochastic optimization by the score function method. Wiley

Shapiro A (1991) Asymptotic analysis of stochastic programs. Ann Oper Res 30(1):169–186

Song E, Nelson B (2016) Input-output uncertainty comparisons for optimization via simulation. Oper Res (Submitted)

Song E, Nelson BL (2015) Quickly assessing contributions to input uncertainty. IIE Transactions 47(9):893–909

Song E, Nelson BL, Hong LJ (2015) Input uncertainty and indifference-zone ranking & selection. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, New Jersey, pp 414–424

Spall JC (1992) Multivariate stochastic approximation using simultaneous perturbation gradient approximation. IEEE Trans Autom Control 37:332–341
Wang Z, Glynn PW, Ye Y (2016) Likelihood robust optimization for data-driven problems. Comput Manag Sci 13(2):241–261
Wiesemann W, Kuhn D, Sim M (2014) Distributionally robust convex optimization. Oper Res 62(6):1358–1376
Wu D, Zhou E (2017) Ranking and selection under input uncertainty: a budget allocation formulation. In: winter simulation conference, (Submitted)
Wu D, Zhu H, Zhou E (2016) Asymptotics of bayesian risk formulations for data-driven stohastic optimization. SIAM J Optim, (Under review). arXiv:1609.08665
Xie W, Nelson BL, Barton RR (2014a) A bayesian framework for quantifying uncertainty in stochastic simulation. Oper Res 62(6):1439–1452
Xie W, Nelson BL, Barton RR (2014b) Statistical uncertainty analysis for stochastic simulation with dependent input models. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, New Jersey, pp 674–685
Zhou E, Xie W (2015) Simulation optimization when facing input uncertainty. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, New Jersey, pp 3714–3724
Zhu H, Hale J, Zhou E (2016) Optimizing conditional value-at-risk via gradient-based adaptive stochastic search. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, New Jersey, pp 726–737
Zhu H, Zhou E (2015) Risk quantification in stochastic simulation under input uncertainty. ACM Trans Model Comp Simul (Under review). arXiv:1507.06015
Zouaoui F, Wilson JR (2003) Accounting for parameter uncertainty in simulation input modeling. IIE Trans 35(9):781–792
Zouaoui F, Wilson JR (2004) Accounting for input-model and input-parameter uncertainties in simulation. IIE Trans 36(11):1135–1151
Zymler S, Kuhn D, Rustem B (2013) Distributionally robust joint chance constraints with second-order moment information. Math Program 137(1–2):167–198

Chapter 12
Parallel Ranking and Selection

Susan R. Hunter and Barry L. Nelson

Abstract The Winter Simulation Conference serves as the initial publication venue for many advances in ranking and selection (R&S), including the recently developed R&S procedures that exploit high-performance parallel computing. We formulate a new stylized model for representing parallel R&S procedures, and we provide an overview of existing R&S procedures under the stylized model. We also discuss why designing R&S procedures for a parallel computing platform is nontrivial and speculate on the future of parallel R&S procedures. In this chapter, "parallel computing" means multiple processors that can execute distinct simulations independently, rather than vector or array processors designed to speed up vector-matrix calculations.

12.1 Introduction

The term *Ranking and Selection* (*R&S*) broadly refers to solution methods developed to solve the R&S problem. The R&S problem is a stochastic optimization problem in which the decision-maker wishes to choose the "best" among a finite set of design points, or "systems," when the performance of each system can only be observed with error. The R&S problem can be considered a special case of the more general simulation optimization (SO) problem, which is a (usually) nonlinear optimization problem whose objectives and constraints, if present, can only be observed with error as output from a stochastic simulation (see, e.g., Pasupathy and Ghosh 2013; Fu 2015 for overviews). Among SO problems, the R&S problem is unique in the sense that it engenders interesting research questions only in the stochastic context: The deterministic black-box analog of the R&S problem is complete enumeration. In contrast, when the system performance measures are defined implicitly through a

S.R. Hunter (✉)
Purdue University, West Lafayette, IN, USA
e-mail: susanhunter@purdue.edu

B.L. Nelson
Northwestern University, Evanston, IL, USA
e-mail: nelsonb@northwestern.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_12

black-box stochastic simulation model, the decision-maker can only observe each system's performance by constructing an estimator whose precision depends on the simulation budget expended. In this context, interesting methodological questions arise. For example, two key questions are (a) how does one guarantee that at the end of simulating, the estimated best system is truly the best system with high probability and (b) how does one allocate a finite simulation budget across systems to efficiently identify the best system? For more than 60 years, researchers have sought answers to these questions, resulting in a large body of R&S literature. For overviews and entry points into this literature, see Bechhofer et al. (1995), Gupta and Panchapakesan (2002) for origins, and Goldsman and Nelson (1998), Kim and Nelson (2006b), and Branke et al. (2007) for the stochastic simulation perspective.

R&S was originally developed by the statistics community during the 1950s, 1960s, and 1970s (Bechhofer 1954; Paulson 1964; Fabian 1964; Dudewicz and Dalal 1957; Rinott 1978), but following its appearance at the Winter Simulation Conference (WSC), the nexus of research shifted from the statistics community to the stochastic simulation community sometime in the 1980s and 1990s. There were two key reasons for this shift: (a) optimizing a function embedded in a stochastic simulation was a natural goal for simulation practitioners, and early SO researchers borrowed existing methods from the statistics community and (b) efficiency in R&S often comes from sequential sampling and comparison of systems, and the barrier to sequential algorithms was far lower in computer experiments than in the industrial and biostatistics applications for which R&S was invented. Further, while the statistics community was often concerned with having procedures for different (non-normal) populations, the emphasis in the simulation community was on designing procedures for larger and larger numbers of alternatives, with normality being plausible due to averaging, e.g., via batch means (Schmeiser 1982).

During this shift, WSC becames a key venue joining the statistics and simulation communities. To the best of our knowledge, R&S first appeared at WSC in a 1976 session entitled, "Statistical Basis for Selection Among Alternatives," which contained the work of Turnquist and Sussman (1976) and Dudewicz (1976). R&S became an increasingly popular topic after Goldsman published a survey paper in the 1983 WSC *Proceedings* (Goldsman 1983). Since then, WSC has served as the initial publication venue for many key advances in the R&S literature, with the area still active at WSC 2016 (e.g., Dong and Zhu 2016).

The significant advances in computing power over the last 40 years, and particularly the recent proliferation of parallel computing platforms, is one reason why R&S is still an active research topic at WSC today.[1] Originally developed with serial computing platforms in mind, R&S procedures are now being redesigned for deployment as parallel procedures—a surprisingly nontrivial endeavor. Serial R&S procedures of the 1990s and early 2000s measure efficiency as the total number of

[1]In this paper "parallel computing platform" means multiple processors that can independently execute simulation experiments and communicate with each other via message passing or shared memory. We use the term "processors" to refer to cores or threads that can complete computing tasks, so the total number of processors is cores × (threads/core).

simulation observations required on a single processor, ensure efficiency by being fully sequential, and tend to work well when the number of systems is small. On a parallel platform, appropriate efficiency measures include processor utilization, wall-clock time, and monetary cost to rent processors. Fully sequential procedures may require bottleneck-inducing synchronization. Further, while today's computing power ensures we can solve small problems fast, it should also enable us to solve much bigger problems. Thus, parallel computing platforms require procedures that minimize new measures of efficiency, guard against the bottlenecks that can arise in a parallel setting and can handle a large number of systems. A handful of such parallel R&S procedures exist; all have made their debut at WSC.

Since R&S was originally designed to select among a small number of categorial or unordered alternatives, one may wonder, when do large R&S problems arise? First, large problems with categorical choices arise naturally in some applications, such as drug discovery and plant breeding. The respective goals in these applications are to find the best drug molecule among many potential drug molecules (Negoescu et al. 2011), and to find the best plant breeding pairs to produce a progeny population with desirable properties (Hunter and McClosky 2016). Second, problems with a very large but finite number of alternatives on an ordered space would seem to be most naturally solved by using algorithms that can exploit the spatial structure, such as R-SPLINE (Wang et al. 2013) or COMPASS (Hong and Nelson 2006; Xu et al. 2010). However, because of its simplicity and ability to provide a statistical guarantee that the selected system is truly the global best, R&S is often a go-to method for practitioners. For example, R&S can be applied to large problems that are created by considering all feasible combinations of a set of decision variables. Xu et al. (2010) describe a SO problem with 500^6 feasible solutions obtained by considering all 500 possible values of 6 order-up-to levels in a supply chain problem. A characteristic of such problems is that many (most) of the feasible solutions are substantially inferior to the better ones, and R&S procedures can exploit this tendency. Even when a search algorithm such as R-SPLINE or COMPASS is employed first, R&S can be used to provide a statistical guarantee as to which of the visited solutions is the best (Boesel et al. 2003).

In this chapter, we discuss the current state of the art in R&S for large problems solved on parallel computing platforms. We assume the reader is familiar with serial R&S procedures, at a broad level. In rethinking R&S procedures for parallel implementation, we provide a new stylized model for representing parallel R&S methods (Sect. 12.2), discuss mathematical and computational formulations of existing serial R&S procedures under the stylized model (Sects. 12.3 and 12.4, respectively), discuss design principles for efficiency and validity of parallel R&S procedures (Sect. 12.5), discuss existing parallel R&S procedures (Sect. 12.6), and speculate on the future of parallel R&S procedures (Sects. 12.7 and 12.8).

12.1.1 Problem Setting and Notational Conventions

R&S addresses the following SO problem: Let the true expected performances of the k competing systems be denoted

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_{k-1} \leq \mu_k,$$

where a larger mean is better, and let $\mathcal{S} = \{1, 2, \ldots, k\}$ denote the set of indices of all systems. We refer to system k, or any system tied with system k, as the best. Often, the best system is assumed to be unique, in which case $\mu_{k-1} < \mu_k$. Recall that we are unable to observe the true expected performances directly. Then suppose we are given a simulation oracle that can provide us with random variables $Y_{i1}, Y_{i2}, \ldots, Y_{in}$, where Y_{ir} is a random variable representing the performance of system i on the rth simulation replication, $r = 1, 2, \ldots, n$, $i \in \mathcal{S}$. For all systems $i \in \mathcal{S}$, we estimate the value of μ_i with a consistent estimator such as the sample mean $\bar{Y}_i(n) := \sum_{r=1}^{n} Y_{ir}/n$.

An *R&S procedure* is an algorithm that attempts to return the best system using only the estimators of the expected system performances. In this chapter, the estimators of the expected system performances after obtaining $n_i \geq 1$ simulation replications from each system $i \in \mathcal{S}$ are $\{\bar{Y}_i(n_i) : i \in \mathcal{S}\}$. We assume an R&S procedure returns the system with the largest estimated mean as the estimated best system, so that $\hat{K} := \text{argmax}_{i \in \{1,2,\ldots,k\}} \{\bar{Y}_i(n_i) : i \in \mathcal{S}\}$ is the estimated best system. (See, e.g., Peng et al. 2016, for an example, in which a system other than the one with the largest estimated mean is returned.) Since we may only assess each system with a finite computational budget, there is always a positive probability that an R&S procedure will return some system other than the best. Thus, R&S procedures are usually created to satisfy some form of mathematical or statistical objectives, which we discuss in Sect. 12.3.

12.1.2 Scope

We classify as R&S any procedures that include three key ingredients: (a) they are applied to a finite number of systems whose expected performance can only be observed with error as simulation output, (b) the procedure will simulate all of the systems and construct consistent estimators of their expected performance, and (c) the decision-maker wishes to select the best by comparing these systems to each other. While we consider only the single-objective problem formulation, we note that stochastically constrained and multiobjective versions of the R&S problem exist. For example, see Andradottir and Kim (2010), Pasupathy et al. (2014) for the stochastically constrained case and Lee et al. (2010), Feldman et al. (2015), Feldman and Hunter (2016), Hunter et al. (2017) for the multiobjective case.

R&S is closely related to some versions of best-arm identification in stochastic multi-armed bandit (MAB) problems; see Bubeck and Cesa-Bianchi (2012)

and Jamieson and Nowak (2014). However, there are differences: MAB is often concerned with online decision-making so as to accumulate the most reward, while R&S is always an offline optimization problem. Perhaps more critically, the key assessment of an MAB algorithm is its "big-O" computational complexity (convergence rate) when selecting the best, while R&S is concerned with finite-time performance, even when asymptotic methods are used in the analysis. As a result, MAB algorithms tend to be simple, have few distribution-specific assumptions, and their computational complexity is determined up to some unknown constants; as compared to R&S that tries to exploit specific distributions to gain efficiency and to assure validity even when unknown constants must be estimated. We focus exclusively on R&S.

12.2 A Stylized Computational Model for R&S

Throughout this chapter, we use a stylized computational model to facilitate our discussion of both serial and parallel R&S procedures. In this section, we define and discuss the stylized model.

We formulate a stylized computational model for parallel R&S procedures by breaking all simulation and calculation tasks that must be completed during an R&S procedure into *jobs*. All R&S procedures contain two primary tasks for processors to complete: (a) performing simulation replications, and (b) calculations completed after simulation replication output is obtained, such as comparing the performances of systems to each other to select the estimated best. Thus we define job j as the ordered list comprised of obtaining simulation replications and performing calculations,

$$J_j := \{(\mathcal{Q}_j, \boldsymbol{\Delta}_j, \mathcal{U}_j), (\mathcal{P}_j, \mathcal{C}_j)\},$$

where

- $\mathcal{Q}_j \subseteq \mathcal{S}$ a set containing the indices of systems to be simulated;
- $\boldsymbol{\Delta}_j = \{\Delta_{ij}\}$ specifies how many samples to take from each system $i \in \mathcal{Q}_j$;
- $\mathcal{U}_j$ is the assigned block of random numbers with which to perform the simulation replications;
- $\mathcal{P}_j$ is a list of jobs whose termination must precede the calculation $\mathcal{C}_j$, if any, and
- $\mathcal{C}_j$ is a list of non-simulation calculations or operations to perform.

We allow $(\mathcal{Q}_j, \boldsymbol{\Delta}_j, \mathcal{U}_j)$ or $(\mathcal{P}_j, \mathcal{C}_j)$ to be null, so that a job can consist of just simulations or just calculations. Since J_j is ordered, we assume that the simulation replications in $(\mathcal{Q}_j, \boldsymbol{\Delta}_j, \mathcal{U}_j)$ are completed before the calculation $\mathcal{C}_j$ begins. In the presence of only one processor, the list of jobs is usually created and performed dynamically by a single processor. In the presence of multiple processors, the list of jobs must be coordinated to preserve precedence requirements. For example, some simulation replications must be obtained from each system before their performances can be compared to each other.

When the number of processors $p \geq 2$, we broadly assume that parallel algorithms operate in what is known as a *master–worker* framework. In this framework, one *master* processor coordinates the activities of one or more *worker* processors. The workers execute jobs determined by the master, and report results back to the master. Communication may occur through shared memory or via message passing. A master–worker framework can also be implemented in multiple tiers, in which each worker acts as a master to, and coordinates the tasks of, one or more sub-workers. For simplicity and ease of exposition, we assume only one such tier for now.

Remark 12.1 We acknowledge the existence of various parallel computing architectures and frameworks for parallel algorithm design (see, e.g., Barlas and Kaufman 2015). We take a higher level approach that enables us to focus on broad R&S procedure design concepts, instead of the details related to the underlying parallel computing architecture.

When a master sends a job to a worker, we assume that all data required to do the job is also transferred, or is otherwise accessible by shared memory. Likewise, when a worker completes a task, we assume relevant data is transferred back to the master. Ensuring efficient data transfer is an important part of designing parallel algorithms; however for exposition, we suppress data transfer information in our framework. Thus while a worker's job may entail performing simulation replications and calculating statistics such as a sample mean, we do not explicitly denote whether the worker transfers just the sample mean back to the master, or the sample mean and all data used to compute the sample mean.

In the master–worker framework, we assume the (possibly dynamic) list of jobs

$$\mathcal{J} := \{J_j : 1 \leq j \leq M\},$$

is created and maintained by the master processor, where job $1 \leq M \leq \infty$ is some (possibly random) terminal job; $M = \infty$ denotes the list of jobs for a non-terminating algorithm. When a worker processor completes a job or becomes idle, it communicates any results back to the master processor and requests a new job. Henceforth, let $0 < T_j < \infty$ be the wall-clock time that job J_j finishes, so that

$$T_e(\mathcal{J}) = \max_{j=1,2,\ldots,M} T_j$$

is the (possibly random) ending time of the procedure.

Remark 12.2 We assume the master creates jobs that can be sent to the workers for execution. Some jobs, particularly jobs containing only calculations, may be executed by the master. Since only the master creates jobs, we do not consider the creation of jobs to be a job.

In modern computing environments, R&S procedures may be completed by purchasing processing power from a service. Since cores may often be purchased in increments such as 4, 8, 16, 48, or 64, with the price per hour varying by the type

of processing power provided, we formulate the general cost to purchase p processors for s time units as a function $c(p, s)$. For a total budget b, we require $c(p, s) \leq b$. Define the function $t(p, b)$ as the maximum amount of time we purchase on p processors, so that

$$t(p, b) := \max\{s : c(p, s) \leq b\}.$$

12.3 Mathematical Formulations of Existing R&S Procedures

Recall that because we cannot simulate every system infinitely often, upon termination, R&S procedures have some positive probability of selecting a system other than the true best. However, most R&S procedures are designed to control this error probability. In this section, we formulate the common goals of existing R&S procedures using the stylized model in Sect. 12.2.

First, we note that most R&S procedures are in some way concerned with the optimality gap between the true best system and the estimated best system, $\mu_k - \mu_{\hat{K}}$. We say that a *correct selection* (CS) event occurs if this optimality gap is zero, and $\mu_k = \mu_{\hat{K}}$. *An ideal R&S procedure would always deliver a CS for any computational budget* $n \geq k$. Since this ideal is impossible in the presence of noise, compromises are made, and the chosen compromise affects the nature of the procedure. R&S procedures may be classified by a number of different approaches and compromises, although these boundaries are not always sharp (see also Pasupathy and Ghosh 2013; Dong and Zhu 2016):

Fixed-precision versus fixed-budget guarantee Fixed-precision procedures execute until some form of guarantee holds, usually on the optimality gap between the selected and true best systems. Fixed-budget procedures attempt to allocate a fixed computational budget in a way that minimizes a loss function that penalizes an incorrect selection event.

Finite-sample versus asymptotic validity Finite-sample procedures provide some provable guarantee within a finite sample size, such as achieved probability of correct selection (PCS). Asymptotic validity procedures achieve guarantees only in some meaningful limit.

Frequentist versus Bayesian guarantee Frequentist probabilistic guarantees are averaged over (conceptually) repeated applications of the procedure. Bayesian probabilistic guarantees are conditioned on the data and averaged over the sources of parameter uncertainty.

In the next two sections, we discuss some of the standard compromises and approaches for creating R&S procedures. Later, we argue that the relevant compromises may be affected by the decision to implement the procedure in a parallel computing environment. In our discussion, we group procedures by whether they are fixed-precision or fixed-budget procedures, which often, but not always, determines the computational formulation of the R&S procedure, as we discuss in Sect. 12.4.

12.3.1 Mathematical Formulation of Fixed-Precision Guarantees

Ideally, fixed-precision R&S procedures are guaranteed to deliver the optimal solution with a pre-specified frequentist probability, which we denote by $1-\alpha$ for $1-\alpha \in (1/k, 1)$. This guarantee is called the probability of correct selection (PCS) guarantee, and is expressed as

$$\mathbb{P}\{\mu_{\hat{K}} = \mu_k\} \geq 1-\alpha.$$

If there are multiple optima, or several solutions with close performance, delivering this guarantee can be computationally infeasible. As a result, making one of the following additional compromises is typical.

- One can assume that the best is unique, and accept the possibility of substantial computation before termination, as in Fan et al. (2016).
- One can allow for a practically significant difference $\delta > 0$, also called an *indifference-zone* (*IZ*) parameter, and instead require

  $$\mathbb{P}\{\hat{K} = k \mid \mu_k - \mu_{k-1} \geq \delta\} \geq 1-\alpha.$$

  The IZ compromise has been widely adopted.
- One can be satisfied with returning a good solution with optimality gap no larger than a user-specified δ:

  $$\mathbb{P}\{\mu_k - \mu_{\hat{K}} \leq \delta\} \geq 1-\alpha.$$

  Some of the procedures that deliver a guaranteed PCS also deliver a guaranteed probability of good selection (PGS), but this is not always the case.
- One can be satisfied with

  $$\mathbb{P}\{\hat{K} \in [k, k-1, k-2, \ldots, k-m+1]\} \geq 1-\alpha.$$

  That is, one can be satisfied with selecting a top-m solution based on rank order, irrespective of the actual optimality gap. This is the compromise behind ordinal optimization (see, e.g., Chen and Lee 2010).
- One can be satisfied with a subset $\hat{\mathcal{S}} \subseteq \mathcal{S}$ such that

  $$\mathbb{P}\{k \in \hat{\mathcal{S}}\} \geq 1-\alpha.$$

  Subset procedures are closely related to multiple comparison procedures that provide simultaneous confidence intervals on some set of differences, and in particular to multiple comparisons with the best (MCB, Hsu 1984). Subset guarantees can often be delivered with weak assumptions, but the conclusion may also be weak if the subset is large. Subset procedures may be used within other R&S procedures

for *screening* or removing systems from the consideration that are estimated as inferior.

While all R&S procedures strive to be efficient, fixed-precision procedures require statistical guarantees to hold. Thus we formulate the objective of fixed-precision procedures by placing a hard constraint on the guarantee, but we wish to purchase processors p and create a job schedule $\mathcal{J}$ such that we minimize the expected (scaled) completion time of the procedure plus the (scaled) monetary cost of the R&S procedure. Under the stylized model in Sect. 12.2, we formulate this problem as

$$\text{minimize}_{p,\mathcal{J}} \quad \mathbb{E}[\beta_t T_e(\mathcal{J}) + \beta_c c(p, T_e(\mathcal{J}))] \quad \text{s.t.} \quad \mathbb{P}\{G\} \geq 1 - \alpha,$$

where $\beta_t \geq 0$ and $\beta_c \geq 0$ are scaling coefficients, and the event G denotes a "good event" upon termination of the procedure, in whatever form. For example, $G = (\hat{K} = k \mid \mu_k - \mu_{k-1} \geq \delta)$ for an IZ compromise, and $G = (k \in \hat{\mathcal{S}})$ for a subset selection compromise. Usually, $\beta_t \in \{0, 1\}$ and $\beta_c = 1 - \beta_t$, so that only expected wall-clock time or only expected cost is minimized, depending on the cost structure of the parallel computing environment. To ensure the probabilistic guarantee constraint is satisfied, we require purchasing as many processor hours as the procedure requires to terminate at time $T_e(\mathcal{J})$; thus, the monetary budget for purchasing processor hours should be $b = \infty$.

12.3.2 Mathematical Formulation of Fixed-Budget Guarantees

In contrast with fixed-precision procedures, in which the simulation budget is determined in part by the required precision, the goal of most fixed-budget procedures is to identify the best system efficiently under a fixed simulation budget. Thus most fixed-budget procedures provide an "efficiency guarantee," which we formulate as

$$\text{minimize}_{p,\mathcal{J}} \quad \mathbb{E}[\mathfrak{L}(G^c, \mathcal{J})] \quad \text{s.t.} \quad t(p, b) \leq t^*,$$

where the function $\mathfrak{L}$ is some type of loss function that depends on an undesirable event (which, loosely speaking, we denote as G^c) upon termination of the procedure, and t^* is a fixed limit on processor hours we purchase. This formulation implies that we wish to choose the processors and the job configuration to minimize the expected loss associated with an incorrect decision, subject to a hard budgetary constraint on the amount of processor hours. The budgetary constraint on the amount of processor hours differs from the traditional constraint on the total number of simulation replications; this formulation provides a more accurate way to measure cost in a parallel computing setting. Note that equivalently, we could formulate the constraint in terms of monetary cost instead of time.

Several prominent fixed-budget guarantee methods include those provided by OCBA (Optimal Computing Budget Allocation) (Chen et al. 2000), the Bayesian Expected Value of Information (EVI) (Chick et al. 2010) and Knowledge Gradient (KG) (Frazier et al. 2008) methods, and the frequentist SCORE (Sampling Criteria for Optimization using Rate Estimators) framework (Pasupathy et al. 2014), which generalizes the work of Glynn and Juneja (2004) and has a close relationship with OCBA and EVI (Ryzhov 2016).

12.3.3 Guarantees Require Standard Assumptions

Whether assuring a desired PCS or minimizing an expected loss, there is an underlying output distribution with respect to which the PCS or expected loss is evaluated. This underlying distribution may be derived from a strong assumption about the simulation output data, or hold asymptotically under weaker conditions. Establishing that these probability guarantees hold in small samples usually requires strong distribution assumptions. Asymptotic analysis (e.g., as $\delta \to 0$) can establish attainment in a large-sample sense. In either case, the actual distribution depends on both (a) the simulation model itself and (b) the sequence of jobs executed. Dependence on (b) is typically not a concern when there is only a single processor, but as discussed in Sect. 12.5, it is critical when jobs are executed in parallel.

To ensure the guarantees from the previous two sections hold, we define the standard output assumptions as follows. Recall that Y_{ir} is a random variable representing the performance of the ith system on the rth simulation replication, for each $r = 1, 2, \ldots$ and all $i \in \mathcal{S}$.

Definition 12.1 The *standard output assumptions* comprise the following:

1. (*Within*) for all systems $i \in \mathcal{S}$, the random variables Y_{ir}, $r = 1, 2, \ldots$ are i.i.d. normally distributed with finite variance, and
2. (*Between*) for all pairs of systems $i, i' \in \mathcal{S}$, the random variables Y_{ir} and $Y_{i'r'}$ are independent for all $r = 1, 2, \ldots$ and all $r' = 1, 2, \ldots$.

The validity of a serial R&S procedure can usually be established under the standard output assumptions. However, these assumptions may be overly stringent. We now provide several common relaxations to the standard output assumptions.

Within Relaxations for all systems $i \in \mathcal{S}$, the random variables Y_{ir}, $r = 1, 2, \ldots$, (a) are i.i.d. with finite variance; (b) are stationary with finite variance; or (c) appropriately standardized, satisfy a Functional Central Limit Theorem.

Between Relaxations for all pairs of systems $i, i' \in \mathcal{S}$, the random variables Y_{ir} and $Y_{i'r}$ are positively correlated for all $r = 1, 2, \ldots$, where the positive correlation is induced by the use of common random numbers (CRN).

CRN is a rule for assigning a set of jobs $\{J_j,\ j \in \mathcal{B}^{(b)}\}$ a "common" block of random numbers $\mathcal{U}_j = \mathcal{U}^{(b)}$ for all $j \in \mathcal{B}^{(b)}$, so that the blocks $b = 1, 2, \ldots$ exhaust

all jobs that require simulation replications in $\mathcal{J}$. The use of CRN across systems to induce a positive correlation and thereby reduce the variance of the difference $\bar{Y}_i(n) - \bar{Y}_{i'}(n)$ has long been a staple of R&S methods to improve statistical efficiency; see for instance Nelson (2013). Because CRN can, in fact, increase variance if simulation outputs are not appropriately paired with equal numbers of observations across systems, the use of CRN imposes an additional coordination problem when there are multiple processors. As a result, CRN has not yet been central to parallel R&S procedures. Therefore, we assume independent blocks of random numbers for each job from here on unless specifically indicated. For simplicity in our stylized model, whenever the blocks of random numbers are independent, we drop the the specification of $\mathcal{U}_j$ and instead write

$$J_j := \{(\mathcal{Q}_j, \Delta_j), (\mathcal{P}_j, \mathcal{C}_j)\}.$$

12.4 Computational Formulations of Existing Serial R&S Procedures

Once we have a mathematical formulation of the goals of the R&S procedure, we require a computational formulation of the procedure that can be implemented on one or more processors. To naïvely implement existing serial R&S procedures in a parallel computing setting, we require an assignment of jobs to processors such that the standard assumptions from the original serial procedure, in whatever form they exist, are still satisfied. The simplest way to accomplish this goal is to parallelize only the parts of the procedure that can be completed in an embarrassingly parallel fashion, and complete all other tasks in the original sequence. As is common in the parallel computing literature, we use the term *embarrassingly parallel* to refer to jobs that are trivially implemented in parallel and require no coordination or synchronization.

Before we provide naïve parallel computational formulations of existing serial fixed-precision and fixed-budget R&S procedures, we define the concepts of *coupled operations* and *stages*, which are concepts that assist with ordering jobs. First, recall that $\mathcal{C}_j$ is a list of calculations that are performed as part of a job j. Typical calculations that arise in R&S procedures include

- determining the sample mean and sample variance of the ith system, $\bar{Y}_i(n)$ and $S_i^2(n) := (n-1)^{-1} \sum_{r=1}^{n} (Y_{ir} - \bar{Y}_i(n))^2$, respectively,
- performing a pairwise comparison $\bar{Y}_i - \bar{Y}_{i'}$ for two systems i and i',
- determining the paired sample variance

$$S_{i,i'}^2(n) := (n-1)^{-1} \sum_{r=1}^{n} (Y_{ir} - Y_{i'r} - (\bar{Y}_i(n) - \bar{Y}_{i'}(n)))^2,$$

- updating a sample allocation rule $\mathfrak{R}$ in a fixed-budget procedure like OCBA, and
- updating a posterior distribution in a Bayesian R&S procedure.

Since operations like pairwise comparisons and calculating paired sample variances require the simulation output of two or more systems, we refer to these operations as *coupled*.

Definition 12.2 We define the following:

- A *coupled operation* or *coupling* is an operation or calculation in which the simulation output of two or more systems is required.
- A *fully coupled operation* or *full coupling* is an operation or calculation that requires the simulation output of all systems still in contention at that point in the procedure.

Thus coupled operations occur when the estimated system performances must be compared to each other, as in a pairwise comparison, or when some key quantity must be calculated that requires the compilation of simulation output from multiple systems. For example, calculating the estimated best system $\hat{K} = \text{argmax}_{i \in \mathcal{S}} \{\bar{Y}_i(n_i)\}$ is a fully coupled operation. Coupling is distinct from the concept of *synchronization* in parallel algorithms, since coupling is across systems, and synchronization is usually across processors. However, when there is a cost to switch from simulating one system to another, it may make sense to assign processors to simulate particular systems, in which case a coupled operation may require synchronization of simulation output across processors.

Since R&S procedures consist of simulations and comparisons, they are usually implemented in what are called *stages*. While the definition of the term "stage" has not always been consistent in the R&S literature, in the context of our stylized model, we define a stage as follows:

Definition 12.3 A *stage* is a portion of an R&S procedure that begins with the first simulation output obtained after initialization or after the last fully coupled operation, and ends when the next fully coupled operation terminates.

While the individual calculations required in the full coupling may be split into jobs that are carried out by multiple processors, when the final calculation of the full coupling is complete, then the stage is over. When variances are unknown, the minimum number of stages is two (Dudewicz and Dalal 1957); the variances are estimated in the first stage. Thus, the first stage almost always consists of obtaining $n_0 \geq 2$ observations from each system, and ends with a fully coupled calculation of key information for implementing the next stage.

In the computational formulations that follow, our goal is to demonstrate the coupling structure of naïve parallelization of each type of serial procedure. Thus, we provide only a straightforward formulation of jobs J_j. We acknowledge that many such formulations exist; some are more efficient than others.

12.4.1 Computational Formulation of Fixed-Precision Procedures

To create a computational formulation for fixed-precision procedures, we begin by formulating existing serial procedures using the stylized model described in Sect. 12.2. We provide a basic formulation of two prominent versions of fixed-precision procedures that have different coupling structures: *two-stage* procedures, which have exactly two stages with two full couplings, and *fully sequential* procedures, which have many stages and frequent full couplings.

The first stage of a two-stage procedure usually begins with obtaining $n_0 \geq 2$ simulation replications from each system, and ends with fully coupled operations that use rules to screen systems and to calculate second-stage sample sizes such that desired statistical guarantees hold. The screening and sampling rules, which we denote as rules $\mathfrak{R}$, are often functions of the user-specified parameters α and δ, and the variances of the system performances. The simulation replications in each of the two stages can be farmed out to worker processors in an embarrassingly parallel fashion, while the master completes all coupled operations. The full coupling at the end of each stage requires a full synchronization across all processors. A naïve two-stage fixed-precision procedure with an optional subset selection step is provided in Algorithm 1. Prominent two-stage procedures include Rinott (1978) and NSGS (Nelson et al. 2001). Ni et al. (2014) provide a parallel version of NSGS that is slightly different from Algorithm 1, called NSGS_p.

Algorithm 1 Naïve Two-Stage Fixed-Precision Procedure (Less Coupling)

procedure TWOSTAGE($\alpha, \delta, n_0, \mathfrak{R}_1, \mathfrak{R}_2$) ▷ Inputs: problem parameters $\alpha \in (1/k, 1)$ and $\delta \in (0, \infty)$, first-stage sample size $n_0 \geq 2$, (optional) rule $\mathfrak{R}_1$ for subset selection, and rule $\mathfrak{R}_2$ for second-stage sample size determination.
  Initialize: Set $\mathcal{Q} = \mathcal{S}$. Master creates jobs $J_i = \{(i, n_0), (\emptyset, \{\bar{Y}_i(n_0), S_i^2(n_0)\})\}$ for all $i \in \mathcal{Q}$.
  Stage 1: *(Simulate: Embarrassingly Parallel)* Workers complete jobs $\{J_i : i \in \mathcal{Q}\}$.
  (Subset Selection: Fully Coupled) Master eliminates inferior systems from $\mathcal{Q}$ using $\mathfrak{R}_1$. If $|\mathcal{Q}| = 1$, return the system in $\mathcal{Q}$ as the estimated best, $\hat{K}$. Otherwise, continue.
  (Calculation: Fully Coupled) Master determines second-stage sample sizes $N_{i,2}$ using $\mathfrak{R}_2$.
  For each $i \in \mathcal{Q}$, Master creates $J_{i,2} = \{(i, \max\{0, N_{i,2} - n_0\}), (\emptyset, \bar{Y}_i(N_{i,2}))\}$.
  Stage 2: *(Simulate: Embarrassingly Parallel)* Workers complete jobs $\{J_i : i \in \mathcal{Q}\}$.
  (Compare: Fully Coupled) Master returns $\hat{K} = \text{argmax}_{i \in \mathcal{Q}} \{\bar{Y}_i(N_{i,2})\}$.
end procedure ▷ Algorithm inspired by Pasupathy and Ghosh (2013, p. 127).

While two-stage procedures can be completed using mostly embarrassingly parallel computation with little synchronization, they are less efficient than fully sequential procedures in terms of the expected total number of simulation replications required. Fully sequential procedures gain sampling efficiency by frequent comparisons and screening. Arguably, fully sequential procedures have the maximum number of stages and hence a "maximal" coupling structure. In each Stage 2+, one simulation replication is obtained from each system still in contention, sample

means and paired variances are updated, and inferior systems are screened out. Since the number of simulation replications per system is equal across surviving systems in each stage of the procedure, that is, $n_i = n_{i'}$ for all $i, i' \in \mathcal{Q}$, some fully sequential procedures such as $\mathcal{KN}$ can be implemented with CRN, further enhancing efficiency. However, screening is an inherently coupled operation—especially when screening requires all pairwise comparisons between systems. Thus when adapting R&S procedures to a parallel computing platform, there exists a tension between sampling efficiency gained by frequent screening, and the potential inefficiency of attempting to perform frequent coupled screening operations across many processors. A generic, naïvely parallelized fully sequential procedure is provided in Algorithm 2. Prominent fully sequential procedures include $\mathcal{KN}$ (Kim and Nelson 2001).

Algorithm 2 Naïve Fully Sequential Fixed-Precision Procedure (More Coupling)

1: **procedure** FULLY SEQUENTIAL($\alpha, \delta, n_0, \mathfrak{R}_s$) ▷ Inputs: problem parameters $\alpha \in (1/k, 1)$ and $\delta \in (0, \infty)$, first-stage sample size $n_0 \geq 2$, a screening rule $\mathfrak{R}_s$.
2: **Initialize:** Set the total samples per system $n = n_0$ and the systems in contention $\mathcal{Q} = \mathcal{S}$. Master creates jobs $J_i = \{(i, n_0), (\emptyset, \{\bar{Y}_i(n_0)\})\}$ for all $i \in \mathcal{Q}$.
3: **Stage 1+:** *(Simulate: Embarrassingly Parallel)* Workers complete jobs $\{J_i : i \in \mathcal{Q}\}$.
4: *(Calculation: Coupled)* Master computes $S^2_{i,i'}(n)$ for all $i, i' \in \mathcal{Q}$, $i \neq i'$.
5: *(Screen: Fully Coupled)* Master uses rule $\mathfrak{R}_s$ on $(\bar{Y}_i(n), S^2_{i,i'}(n))$, $i, i' \in \mathcal{Q}$ to eliminate inferior systems from $\mathcal{Q}$. If $|\mathcal{Q}| = 1$, return the system in $\mathcal{Q}$ as the estimated best, $\hat{K}$. Otherwise, set $n = n + 1$. Master creates new jobs $J_{i,n} = \{(i, 1), (\emptyset, \bar{Y}_i(n))\}$ for each $i \in \mathcal{Q}$.
6: Go to the next stage, Step 3.
7: **end procedure** ▷ Algorithm inspired by Pasupathy and Ghosh (2013, p. 129).

12.4.2 Computational Formulation of Fixed-Budget Procedures

Fixed-budget procedures often take a similar computational structure, outlined in Algorithm 3. As in the fixed-precision procedures, the first stage begins by obtaining $n_0 \geq 2$ simulation replications from each system, and ends with a (usually) fully coupled operation that determines how to allocate the Δ simulation replications in the next stage, using a sampling rule $\mathfrak{R}$. This process of obtaining Δ simulation replications per stage and updating the sampling rule is repeated until the total simulation time has been exhausted. Since the frequency of the coupling and the number of stages is determined by the parameter Δ, these procedures tend to have a flexible coupling frequency. We note that some procedures are designed for myopic sampling, such that $\Delta = 1$, while other procedures are more flexible in the choice of Δ.

Algorithm 3 Fixed-Budget Procedure (Flexible Coupling Frequency)

procedure EFFICIENCY($n_0, \Delta, t^*, \mathfrak{R}$) ▷ Inputs: initial simulation budget $n_0 \geq 2$, stagewise simulation budget Δ, limit on total effort $t^* > 0$, and rule $\mathfrak{R}$ for stagewise allocation.
  Initialize: Set stage $\ell = 1$, sample sizes $n_{i,\ell} = n_0$ for all $i \in \mathbb{S}$, total effort $t_0 = 0$, and Master creates jobs $J_i = \{(i, n_0), (\emptyset, \{\bar{Y}_i(n_0), S_i^2(n_0)\})\}$ for all $i \in \mathbb{S}$.
  while $t_{\ell-1} \leq t^*$ **do**
    Stage ℓ: *(Simulate: Embarrassingly Parallel)* Workers complete jobs $\{J_i\}$.
    (Calculation: Fully Coupled) Master applies rule $\mathfrak{R}$ to statistical history to find next stage sample allocation $\{n_{i,\ell+1} : \sum_{i \in \mathbb{S}} n_{i,\ell+1} = \Delta\}$. Master creates jobs $J_i = \{(i, n_{i,\ell+1}), (\bar{Y}_i(n_{i,\ell+1}), S_i^2(n_{i,\ell+1}))\}$ for all $i \in \{i' : i' \in \mathbb{S}, n_{i',\ell+1} \geq 1\}$.
    Master updates total effort expended so far t_ℓ.
    Master sets $\ell = \ell + 1$.
  end while
  return $\hat{K} = \text{argmax}_{i \in \mathbb{S}} \{\bar{Y}_i(\sum n_{i,\ell})\}$.
end procedure ▷ Algorithm inspired by Pasupathy and Ghosh (2013, p. 132).

12.5 Parallelization: Efficiency and Validity

Having presented fairly straightforward parallel computational frameworks for existing serial R&S procedures that should preserve the standard assumptions from the original serial procedure, one may wonder, why not simply use these procedures? While such procedures surely can be implemented, they are unlikely to scale well to larger problem instances and to achieve the levels of speedup and efficiency we would like to see from a parallel R&S algorithm. The concepts of speedup and efficiency (or scalability) are defined in Barlas and Kaufman (2015) as follows. Suppose we are handed a parallel algorithm that requires t_p wall-clock time to be run on p identical processors in parallel, and t_s wall-clock time to be run on only one of the processors. Then the speedup is defined as the ratio of the sequential time to the parallel time, $speedup := t_s/t_p$. Given $p \geq 1$ processors, the *efficiency* is defined as the scaled speedup, $efficiency := s/p = t_s/(pt_p)$. Thus speedup gives a measure of how beneficial it is to execute the algorithm in parallel, while efficiency measures the utilization of the available processors. An Efficiency of 1 corresponds to *linear speedup*, in which case the speedup is equal to the number of processors, p. Embarrassingly parallel jobs that are appropriately load-balanced across cores tend to achieve almost linear speedup.

Since embarrassingly parallel implementations achieve almost linear speedup, it seems that two-stage procedures would perform the best in parallel. However, recall that two-stage procedures require more simulation replications, on average, than those that have more frequent coupled operations like screening. Thus it may benefit the procedure to introduce more frequent coupled operations. However, we have two potential forms of idleness that arise: (a) the master may be idle waiting for every simulation replication to complete before it performs any coupled operations—especially if simulation replication completion times are random—and (b) if screening and fully coupled operations take significant computational effort on the master, many worker processors must wait for the master to create new jobs.

Then, we may wish to design a procedure that does not require the processors to wait for each other. Unfortunately, *the standard assumptions are most easily assured by one-job-at-a-time execution.* Estimators can become biased if we do not wait for all parallelized simulation replications to complete; such bias was investigated by Heidelberger (1988) and Glynn and Heidelberger (1990, 1991). Further, serious violations of the standard assumptions can occur if jobs are executed in parallel but output data are used as available and without enforcing conformance with single-processor execution. These include the following, as described in Luo et al. (2015), Ni et al. (2013, 2017).

Random sample size The number of observations from system i when the nth overall observation is obtained may be random if job execution time is variable.

Not i.i.d. The observations n_i from system i may not be i.i.d. if the order in which jobs complete is not the order in which the jobs were dispatched, and there is a dependence between returned value and execution time.

Dependence across systems A difficult-to-characterize dependence across systems' outputs can be induced if elimination of system i by system i' frees processors that affect the number of observations obtained from other systems.

These issues suggest that we must employ output coordination strategies that ensure all calculations $(\mathcal{P}_j, \mathcal{C}_j)$ across jobs j are executed as they would be if there were only a single processor. However, this still leads to a potential degradation of efficiency and speedup from the two forms of idleness: master waiting for simulation replications, and workers waiting for the master's calculations.

Based on this analysis, efficient parallel R&S requires procedures with one or more of the following characteristics: (a) they implement careful load balancing to retain the standard assumptions without significant idling and overwhelming communication; or (b) they are valid under weaker assumptions than the standard ones; or (c) the procedure uses a combination of the strategies above. We discuss existing parallel R&S procedures of both types in the next section.

12.6 Existing Parallel Ranking and Selection Procedures

Existing parallel R&S procedures overcome some of the shortcomings of the naïve parallelization of existing serial R&S procedures. We now discuss the state of the art in both fixed-precision and fixed-budget parallel ranking and selection procedures.

12.6.1 Parallel Fixed-Precision Procedures

We describe four parallel fixed-precision procedures and formulate each procedure in terms of the types of jobs created and deployed to the workers. First, Luo and Hong (2011), Luo et al. (2015) extend the $\mathcal{KN}$ procedure (Kim and Nelson 2006a, 2001),

which provides a PCS guarantee, to the parallel setting in two distinct ways: a conservative vector-filling procedure (VFP) that strictly enforces the standard assumptions, and an aggressive asymptotic parallel selection (APS) procedure that is valid under weaker assumptions. Both algorithms resemble Algorithm 2 in that they use elimination at every stage; their key difference is in how they define the completion of a stage. Then, we discuss a simple divide-and-conquer approach by Chen (2005) for when the number of processors is small. Finally, a substantial extension of the divide-and-conquer approach is provided by the good selection procedure (GSP) of Ni et al. (2017), which provides a PGS guarantee.

Of these procedures, we highlight two procedures for their strategies related enhancing efficiency and maintaining validity. First, APS never allows the master to idle waiting for simulation replications, but maintains its validity under conditions weaker than the standard ones. Second, GSP maintains validity under the standard assumptions, but performs careful load balancing to maintain efficiency.

12.6.1.1 VFP: Vector-Filling Procedure

In VFP, the master creates/executes three types of jobs:

1. Initialization jobs:

$$\mathcal{J}_0 = \left[\{(1, n_0), (\emptyset)\}, \{(2, n_0), (\emptyset)\} \ldots, \{(k, n_0), (\emptyset)\}, \{(\emptyset), (\mathcal{P}_0, \mathcal{C}_0)\}\right]$$

   where the set $\mathcal{P}_0$ includes the k preceding simulation jobs, and the calculations $\mathcal{C}_0$ include computing the variance of all pairwise differences, making pairwise comparisons of the sample means of all k systems, and possibly eliminating some systems.
2. Round-robin simulation jobs: Conceptually, there is an infinite set of sets of simulation jobs

$$\mathcal{J}_\ell = \left[\{(1, 1), (\emptyset)\}, \{(2, 1), (\emptyset)\} \ldots, \{(k, 1), (\emptyset)\}\right], \ell = 1, 2, \ldots$$

   that obtain one additional replication from each system. However, if at stage ℓ' system i' is eliminated, then all $\{(i', 1), (\emptyset)\}$ jobs are eliminated from the unexecuted simulation job set. Upon completion of a simulation job, a worker pulls the next simulation job in the sequence to execute.
3. Elimination jobs: Stage ℓ is defined by an elimination job $\{(\emptyset), (\mathcal{P}_\ell, \mathcal{C}_\ell)\}$, where $\mathcal{P}_\ell$ contains all simulation jobs $\mathcal{J}_\ell$, and $\mathcal{C}_\ell$ performs pairwise comparisons of all systems that have not been eliminated at an earlier stage. The elimination jobs are executed by the master.

The VFP terminates when there is only one system that has not been eliminated. The term "vector filling" is appropriate because the VFP enforces the standard assumptions by associating each simulation output with its job set $\mathcal{J}_\ell$, and only performing a full coupling for stage ℓ when all jobs in $\mathcal{J}_\ell$ have completed. For this reason, outputs

from later job sets, say $\mathcal{J}_{\ell+1}$, that complete before jobs in $\mathcal{J}_\ell$ must be held in a vector for later elimination calculations.

12.6.1.2 APS: Asymptotic Parallel Selection

The APS procedure is superficially similar to the VFP, but a small change makes its computational profile quite different.

1. Initialization jobs:

$$\mathcal{J}_0 = \left[\{(1, n_0), (\emptyset)\}, \{(2, n_0), (\emptyset)\} \ldots, \{(k, n_0), (\emptyset)\}, \{(\emptyset), (\mathcal{P}_0, \mathcal{C}_0)\}\right]$$

   where the set $\mathcal{P}_0$ includes the k preceding simulation jobs, and the calculations $\mathcal{C}_0$ include computing the marginal variance of all k systems, making pairwise comparisons of the sample means of all k systems, and possibly eliminating some systems. (These initialization jobs are the same as those in VFP.)
2. Round-robin simulation jobs: Conceptually, there is an infinite set of sets of simulation jobs

$$\mathcal{J}_\ell = \left[\{(1, 1), (\emptyset)\}, \{(2, 1), (\emptyset)\} \ldots, \{(k, 1), (\emptyset)\}, \{(\text{phantom}, 0), (\emptyset)\}\right], \ell = 1, 2, \ldots$$

   that obtain one additional replication from each real system, and no replications from a "phantom" system. Again, if at stage ℓ' real system i' is eliminated, then all $\{(i', 1), (\emptyset)\}$ jobs are elminated from the unexecuted simulation job set. Upon completion of a simulation job, a worker pulls the next simulation job in the sequence to execute, which could be a phantom.
3. Elimination jobs: Stage ℓ is defined by an elimination job $\{(\emptyset), (\mathcal{P}_\ell, \mathcal{C}_\ell)\}$, where $\mathcal{P}_\ell$ is the ℓth phantom job. The calculation $\mathcal{C}_\ell$ updates marginal variances and performs pairwise comparisons of all systems that have not been eliminated at an earlier stage *using all available simulation output data.* The elimination jobs are executed by the master.

The APS procedure defines a stage as the completion of a phantom job, but otherwise makes no attempt to process simulation jobs in any order. Thus, it is aggressive in that the master never idles waiting for a particular real simulation job to complete, but it is subject to all of the violations of standard assumptions described in Sect. 12.5. The validity of APS is asymptotic, as $\delta \to 0$, with the key insight being that since there are $p < \infty$ processors the simulation jobs may only be out of order by an asymptotically negligible p jobs.

12.6.1.3 Simple Divide-and-Conquer

An early paper by Chen (2005) describes a simple approach that is sensible when the number of processors p is small; GSP below can be considered a substantial extension of this idea. There are two types of jobs:

1. Group R&S jobs: The k systems are divided as evenly as possible into p nonoverlapping groups of systems, say $\mathcal{G}_1, \mathcal{G}_2, \ldots \mathcal{G}_p$, and p jobs are formed

$$J_j = \{(\mathcal{G}_j, \Delta_j), (\mathcal{G}_j, \mathcal{C}_j)\},\ j = 1, 2, \ldots, p$$

   where each job j is a complete R&S procedure that returns a group-best selected system $\widehat{i}_j$ along with its accumulated output data.
2. Final R&S job: Let $\mathcal{Q} = \{\widehat{i}_1, \widehat{i}_2, \ldots, \widehat{i}_p\}$, the group bests. Then the final job is

$$J_{p+1} = \{(\mathcal{Q}, \Delta_{p+1}), (\mathcal{Q}, \mathcal{C}_{p+1})\},\ j = 1, 2, \ldots, p$$

   which performs a R&S procedure on the group-best systems $\mathcal{Q}$ starting with their previously accumulated data and $\mathcal{C}_{p+1}$ computes the sample means and selects the best.

Chen (2005) suggests some specific R&S procedures for each type of job, but the framework is flexible. The simplicity of this strategy is appealing, but it will lose effectiveness when $k \gg p$ so that the group R&S jobs themselves are challenging.

12.6.1.4 GSP: Good Selection Procedure

The GSP procedure of Ni et al. (2017) (also see Ni et al. 2013, 2014, 2015) provides a PGS guarantee, instead of the usual PCS guarantee, under the standard output assumptions. GSP exhibits good speedup and efficiency using careful load-balancing and reduced computation. Several key strategies of GSP include: (a) distributing screening tasks to the workers in a divide-and-conquer fashion to avoid overwhelming the master with screening calculations; (b) using only a reduced number of pairwise comparisons instead of completing all pairwise comparisons; and (c) carefully constructing load-balanced jobs of large-enough size to prevent overwhelming the master with communication. As a result, when implemented in a high-performance computing (HPC) environment in C with MPI, the master is idle most of the time. However in its idleness, the master usually is ready to communicate and can ensure the workers are *not* idle most of the time.

GSP has three stages and one "phase," which is a sequential portion of the algorithm containing multiple stages. GSP's stages and phases are: an optional load-balancing stage; an initialization stage with screening; a sequential phase that contains multiple stages and is somewhat similar in structure to Algorithm 2 after initialization; and a Rinott stage, similar in structure to Algorithm 2. The sequential

phase is intended to harness the efficiency of sequential screening to create a subset of contender systems likely to contain the best. In the Rinott stage, the appropriate sample sizes for the remaining systems are calculated, and the simulations for remaining competitive systems are completed in an embarrassingly parallel fashion.

In this section, we assume we have p worker processors, where the zeroth processor is the master. In each stage or phase, the master creates the following types of jobs:

1. Optional load-balancing jobs: The master randomly permutes the systems in $\mathcal{S}$ and assigns an approximately equal number of systems to groups $\mathcal{G}_0^1, \ldots, \mathcal{G}_0^p$, for each processor. Then the master creates jobs

$$\mathcal{J}_0 = \{(\mathcal{G}_0^w, n_0^*), (\emptyset, \{\bar{T}_i : i \in \mathcal{G}_0^w\})\}_{w=1}^p,$$

   where $\bar{T}_i$ is the average simulation completion time across all replications n_0^* from simulating system i. After calculating statistics $\bar{T}_i$, the simulation output is thrown away, due to potential dependence between the output random variable Y_{ir} and the simulation replication completion time T_{ir}.
2. Initialization jobs: Using information from the optional load-balancing step if available, the master partitions the systems in $\mathcal{S}$ into load-balanced simulation groups $\mathcal{G}_1^1, \ldots, \mathcal{G}_1^p$ for each processor. Then the master creates jobs

$$\mathcal{J}_1 = \{(\mathcal{G}_1^w, n_0), (\emptyset, \{(\bar{Y}_i(n_0), S_i^2(n_0), \mathcal{C}_1) : i \in \mathcal{G}_1^w\})\}_{w=1}^p,$$

   where $\mathcal{C}_1$ is a screening calculation that only reports the surviving systems and their sufficient statistics to the master. The master updates the surviving systems $\mathcal{Q}$.
3. Sequential phase jobs: The master divides the systems into approximately load-balanced screening groups $\mathcal{G}_2^1, \ldots, \mathcal{G}_2^p$ using rule $\mathfrak{R}_1^{\mathrm{GSP}}$, so that each processor always screens the same set of systems. The master also uses rule $\mathfrak{R}_2^{\mathrm{GSP}}$ to determine an appropriate "batch size" b_i of simulation replications to obtain from each system $i \in \mathcal{Q}$ in each simulation job, so that the master is not overwhelmed with communication. The sequential phase ends when a pre-determined maximum number of batches has been simulated, or when $|Q| = 1$.

   a. Simulation jobs: The master creates and maintains an ordered list of batched simulation jobs for each system $i \in \mathcal{Q}$. Whenever a worker becomes idle and the master indicates that some systems in its screening group are not ready for screening, the worker requests the next simulation job in the list, for any system $i \in \mathcal{Q}$. For each system still in contention $i \in \mathcal{Q}$, the νth simulation job is

$$J_{i,\nu} = \{(i, b_i), (\emptyset, \bar{Y}(b_i))\}, \nu = 1, 2, \ldots.$$

   b. Within-group and best-across-processor screening jobs: Whenever a processor becomes idle and the νth simulation batch has completed for all systems

in its screening group, the processor pulls the "screening job" for its group from the master,

$$J_\nu^w = \{\emptyset, (\{J_{i,\nu} : i \in \mathcal{G}_2^w\}, \mathcal{C}_\nu^w)\},$$

where $\mathcal{C}_\nu^w$ is an all pairwise screening job within the group $\mathcal{G}_2^w$, as well as among the best systems who have completed batch ν from the other screening groups. The processor then reports the indices of eliminated systems to the master, who updates the set of systems still in contention, $\mathcal{Q}$. Note that the νth screening must occur before the $(\nu + 1)$th screening, and so on. Per Definition 12.3, each within-group screening that uses the best systems from screening groups on *all* the other processors constitutes the end of a stage.

4. Rinott stage jobs: If $|\mathcal{Q}| > 1$, the master uses a rule $\mathfrak{R}$ to determine the Rinott stage sample sizes $N_{i,4}$ for all remaining systems $i \in \mathcal{Q}$. Let N_i be the total number of simulation replications observed from each system $i \in \mathcal{Q}$ so far before the Rinott stage, and define $N_{i,4}^+ := \max\{0, N_{i,4} - N_i\}$ as the number of additional simulation replications required from system i. The master then arranges the required additional simulation replications for each system into load-balanced "batched" jobs; for ease of exposition, we omit the batching notation in this stage. Then for all $i \in \mathcal{Q}$ such that $N_{i,4}^+ > 0$, the master creates the jobs

$$J_{4,i} = \{(i, N_{i,4}^+), (\emptyset, \bar{Y}_i(N_{i,4}))\}.$$

After all simulation replications terminate, the master updates the sample means with the latest data and returns the estimated best system $\hat{K}$.

12.6.2 Parallel Fixed-Budget Procedures

Luo et al. (2000) is the first reported effort to parallelize an R&S procedure, specifically OCBA in the fixed-budget setting. Their base algorithm resembles Algorithm 3, and they assume a master–worker environment with a small number of workers ($p \leq 3$ in their experiments).

The master creates/executes three types of jobs:

1. Initialization jobs:

$$\left[\{(1, n_0), (\emptyset)\}, \{(2, n_0), (\emptyset)\} \ldots, \{(k, n_0), (\emptyset)\}, \{(\emptyset), (\mathcal{P}_0, \mathcal{C}_0)\}\right]$$

where the set $\mathcal{P}_0$ includes the k preceding simulation jobs, and the calculations $\mathcal{C}_0$ include computing the marginal sample means and variance of the k systems.

2. Simulation jobs: In the ℓth stage, p jobs $\{(i_j, \Delta), (\emptyset)\}$ for $j = 1, 2 \ldots, p$ are created, where $i_j \in \{1, 2, \ldots, k\}$ denote p distinct systems, each allocated the same number of replications, Δ. These jobs are executed by the p workers in parallel.

3. OCBA jobs: $\{(\emptyset), (\mathcal{P}_\ell, \mathcal{C}_\ell)\}$, where $\mathcal{P}_\ell$ contains all of the simulation jobs from the ℓth simulation stage, and $\mathcal{C}_\ell$ performs the OCBA optimization to find the p systems for whom an allocation of Δ additional replications would most rapidly increase an approximate posterior PCS expression. Simulation jobs are then created for these p systems.

By having the OCBA job hold for the return of all of the ongoing simulation jobs, this algorithm enforces the single-processor assumptions behind OCBA at each stage. As noted by the authors, there is a loss of statistical efficiency by simulating the top-p OCBA systems at each stage, rather than simulating the single best then reevaluating, but there is a gain in computational efficiency. The algorithm terminates when a fixed number-of-replications budget is expended. A related paper by Yoo et al. (2009) also applies OCBA in a parallel *search* setting where not all systems are expected to be simulated.

12.6.3 Available Implementations of Parallel R&S Procedures

To the best of our knowledge, only one commercial simulation product, Simio (http://www.simio.com), has implemented R&S procedures that exploit parallel computing. Simio has implemented two fixed-precision procedures: $\mathcal{KN}$ (Kim and Nelson 2001, 2006a) which uses multiple processors on a local PC, and GSP (Ni et al. 2017) which is specifically designed to use high-performance or cloud computing. $\mathcal{KN}$ gains efficiency by obtaining replications in parallel; in every other sense it is the single-processor algorithm and it implements full synchronization at every stage.

There are also public code repositories that contain parallel versions of R&S procedures. In this paragraph, the citations provide links to code repositories that are publicly available at the time of writing. GSP has been implemented in MapReduce (Ni 2015a), MPI (Ni 2015b), and Spark (Ni 2015c). Code for a parallel version of OCBA is available (Li 2017). As a repository for the simulation optimization community, http://www.simopt.org (Henderson 2016), also contains test problems for a variety of problem types, as well as an algorithms library with publicly available code.

12.7 A Future Research Agenda

Effective and efficient parallel R&S procedures of the future seem likely to be obtained by a careful coordination of a number of ideas. Here is a part of the roadmap as we see it.

Assignment of jobs to processors is clearly a type of stochastic parallel-machine scheduling problem as addressed by the operations research literature (see for

instance Pinedo 2015). The objective in such problems is often to minimize makespan, which is analogous to our objective in the fixed-precision formulation, and sequencing constraints are similar to our dependence of certain computations on the completion of particular jobs, $(\mathcal{P}_j, \mathcal{C}_j)$. A key difference is that the jobs that need scheduling in parallel R&S may evolve based on the simulation outputs obtained from earlier jobs, rather than being all available in advance or arriving according to some exogeneous stochastic process. Nevertheless, this is a deep literature whose lessons should not be ignored.

Strategies that avoid full coupling seem critical as the number of all pairwise comparisons grows as $O(k^2)$. Thus, as k increases it becomes computationally prudent to simulate more outputs than strictly needed for, say, correct selection to avoid coupling. This can be done from at least two directions:

1. Distributed screening: Couplings of $k' \ll k$ systems to screen out inferior systems and pass competitors to full couplings, thereby reducing the comparisons to $O((k')^2)$.
2. Distributed killers: Obtaining high-precision estimates of an apparently good solution and distributing it to all or groups of systems to screen out inferior ones; this type of screening is $O(k)$.

The fixed-budget formulation, when expressed as a limit on the number of simulation replications, has always been somewhat artificial. A fixed *monetary* or *time* budget for parallel computation, on the other hand, is both concrete and relevant. To us, the joint choice of a number of processors p and jobs to execute $\mathcal{J}$ to minimize expected loss with respect to a monetary budget looks very challenging indeed. We suspect that a strategy that chooses p based on a priori problem characteristics, and then treats it as fixed when optimizing over $\mathcal{J}$, will be the most productive avenue.

Finally, parallel R&S for very large numbers of systems should cause us to revisit the standard R&S objectives as described in Sects. 12.3.1–12.3.2. For very large k a PGS guarantee seems more relevant and easier to obtain than a PCS guarantee, as it seems likely there are many close competitors. More critically, *any* objective that returns a *single* system $\widehat{K}$ without additional inference about the others seems questionable. Consider an alternative objective:

Suppose, based on previous experience with similar problems, a known standard for "good" performance of $\mu^\star$ can be established. Finding, with high probability, the subset of systems with $\mu_i \geq \mu^\star$ is a fully uncoupled problem that is embarrassingly parallel. A related approach by Singham and Szechtman (2016) defines inclusion of inferior systems in the subset as a "false discovery" and sets as the objective bounding the false discovery *rate*. In terms of both conservatism of the inference and growth of computation, these ideas scale better than the traditional objectives.

12.8 WSC 2017

At the time of writing, we are aware of at least one paper on parallel R&S under review for WSC 2017. Thus parallel R&S continues to be an active research area at WSC.

Acknowledgements Hunter's research was partially supported by the National Science Foundation under Grant Number CMMI-1554144. Nelson's research was partially supported by the National Science Foundation under Grant Number CMMI-1537060 and GOALI co-sponsor SAS Institute.

References

Andradóttir S, Kim SH (2010) Fully sequential procedures for comparing constrained systems via simulation. Nav Res Logist 57(5):403–421. doi:10.1002/nav.20408

Barlas G, Kaufman M (2015) Multicore and GPU programming: an integrated approach. Elsevier

Bechhofer RE (1954) A single-sample multiple decision procedure for ranking means of normal populations with known variances. Ann Math Stat 25(1):16–39

Bechhofer RE, Santner TJ, Goldsman D (1995) Design and analysis of experiments for statistical selection. Screening and multiple comparisons, Wiley, New York

Boesel J, Nelson BL, Kim SH (2003) Using ranking and selection to 'clean up' after simulation optimization. Oper Res 51(5):814–825

Branke J, Chick SE, Schmidt C (2007) Selecting a selection procedure. Manage Sci 53(12):1916–1932

Bubeck S, Cesa-Bianchi N et al (2012) Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and trends$_\wedge$. Mach Learn 5(1):1–122

Chen CH, Lee LH (2010) Stochastic simulation optimization: an optimal computing budget allocation. World Scientific, Hackensack, NJ

Chen CH, Lin J, Yücesan E, Chick SE (2000) Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Discrete Event Dyn Syst 10(3):251–270. doi:10.1023/A:1008349927281

Chen EJ (2005) Using parallel and distributed computing to increase the capability of selection procedures. In: Kuhl ME, Steiger NM, Armstrong FB, Joines JA (eds) Proceedings of the 2005 winter simulation conference. Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 723–731

Chick SE, Branke J, Schmidt C (2010) Sequential sampling to myopically maximize the expected value of information. INFORMS J Comput 22(1):71–80

Dong J, Zhu Y (2016) Three asymptotic regimes for ranking and selection with general sample distributions. In: Roeder TMK, Frazier PI, Szechtman R, Zhou E, Huschka T, Chick SE (eds) Proceedings of the 2016 Winter Simulation Conference. IEEE, Institute of Electrical and Electronics Engineers Inc, Piscataway, NJ, pp 277–288

Dudewicz EJ (1976) Statistics in simulation: how to design for selecting the best alternative. In: Highland HJ, Schriber TJ, Sargent RG (eds) Proceedings of the 1976 bicentennial winter simulation conference, pp 67–71. http://informs-sim.org/wsc76papers/1976_0012.pdf

Dudewicz EJ, Dalal SR (1975) Allocation of observations in ranking and selection with unequal variances. Sankhya Indian J Stat B37:28–78

Fabian V (1964) Note on Anderson's sequential procedures with triangular boundary. Ann Stat 2(1):170–176

Fan W, Hong LJ, Nelson BL (2016) Indifference-zone-free selection of the best. Oper Res 64(6):1499–1514

Feldman G, Hunter SR (2016) SCORE allocations for bi-objective ranking and selection. Optim Online. http://www.optimization-online.org/DB_HTML/2016/06/5469.html

Feldman G, Hunter SR, Pasupathy R (2015) Multi-objective simulation optimization on finite sets: optimal allocation via scalarization. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 3610–3621. doi:10.1109/WSC.2015.7408520

Frazier PI, Powell WB, Dayanik S (2008) A knowledge-gradient policy for sequential information collection. SIAM J Control Optim 47(5):2410–2439

Fu M (ed) (2015) Handbook of simulation optimization. In: International series in operations research & management science, vol 216. Springer, New York. doi:10.1007/978-1-4939-1384-8

Glynn PW, Heidelberger P (1990) Bias properties of budget constrained simulations. Oper Res 38(5):801–814

Glynn PW, Heidelberger P (1991) Analysis of parallel replicated simulations under a completion time constraint. ACM Trans Model Comput Simul 1(1):3–23

Glynn PW, Juneja S (2004) A large deviations perspective on ordinal optimization. In: Ingalls RG, Rossetti MD, Smith JS, Peters BA (eds) Proceedings of the 2004 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 577–585. doi:10.1109/WSC.2004.1371364

Goldsman D (1983) Ranking and selection in simulation. In: Roberts S, Banks J, Schmeiser B (eds) Proceedings of the 1983 winter simulation conference, pp 387–393. http://informs-sim.org/wsc83papers/1983_0017.pdf

Goldsman D, Nelson BL (1998) Statistical screening, selection, and multiple comparison procedures in computer simulation. In: Medieros DJ, Watson EF, Carson JS, Manivannan MS (eds) Proceedings of the 1998 winter simulation conference. Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp 159–166

Gupta, S.S., Panchapakesan, S.: Multiple decision procedures: theory and methodology of selecting and ranking populations. SIAM (2002)

Heidelberger P (1988) Discrete event simulations and parallel processing: statistical properties. Siam J Stat Comput 9(6):1114–1132

Henderson SG, Pasupathy R (2016) Simulation optimization library. http://www.simopt.org

Hong LJ, Nelson BL (2006) Discrete optimization via simulation using COMPASS. Oper Res 54(1):115–129

Hsu JC (1984) Constrained simultaneous confidence intervals for multiple comparisons with the best. Ann Stat 12:1136–1144

Hunter SR, Applegate EA, Arora V, Chong B, Cooper K, Rincón-Guevara O, Vivas-Valencia C (2017) An introduction to multi-objective simulation optimization. Optim Online. http://www.optimization-online.org/DB_HTML/2017/03/5903.html

Hunter SR, McClosky B (2016) Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. IIE Trans 48(6):565–578. doi:10.1080/0740817X.2015.1096430

Jamieson K, Nowak R (2014) Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In: 2014 48th Annual conference on information sciences and systems (CISS). IEEE, pp 1–6

Kim SH, Nelson BL (2001) A fully sequential procedure for indifference-zone selection in simulation. ACM Trans Model Comput Simul 11(3):251–273

Kim S, Nelson BL (2006a) On the asymptotic validity of fully sequential selection procedures for steady-state simulation. Oper Res 54(3):475–488. doi:10.1287/opre.1060.0281

Kim SH, Nelson BL (2006b) Selecting the best system. In: Henderson SG, Nelson BL (eds) Simulation, handbooks in operations research and management science, vol 13. Elsevier, Amsterdam, pp 501–534

Lee LH, Chew EP, Teng S, Goldsman D (2010) Finding the non-dominated Pareto set for multi-objective simulation models. IIE Trans 42:656–674. doi:10.1080/07408171003705367

Li H (2017) Parallel OCBA. https://gitlab.com/o2des_dev/ParallelOCBA

Luo J, Hong LJ (2011) Large-scale ranking and selection using cloud computing. In: Jain S, Creasey RR, Himmelspach J, White KP, Fu M (eds) Proceedings of the 2011 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 4051–4061. doi:10.1109/WSC.2011.6148094

Luo J, Hong LJ, Nelson BL, Wu Y (2015) Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. Oper Res 63(5):1177–1194. doi:10.1287/opre.2015.1413

Luo YC, Chen CHYücesan E, Lee I (2000) Distributed web-based simulation optimization. In: Joines JA, Barton RR, Kang K, Fishwick PA (eds) Proceedings of the 2000 winter simulation conference. Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 1785–1793. doi:10.1109/WSC.2000.899170

Negoescu DM, Frazier PI, Powell WB (2011) The knowledge-gradient algorithm for sequencing experiments in drug discovery. INFORMS J Comput 23(3):346–363

Nelson B (2013) Foundations and methods of stochastic simulation: a first course. Springer (2013)

Nelson BL, Swann J, Goldsman D, Song W (2001) Simple procedures for selecting the best simulated system when the number of alternatives is large. Oper Res 49(6):950–963

Ni EC (2015a) MapRedRnS: parallel ranking and selection using mapreduce. https://bitbucket.org/ericni/mapredrns

Ni EC (2015b) mpiRnS: Parallel ranking and selection using MPI (2015). https://bitbucket.org/ericni/mpirns

Ni EC (2015c) SparkRnS: parallel ranking and selection using spark. https://bitbucket.org/ericni/sparkrns

Ni EC, Ciocan DF, Henderson SG, Hunter SR (2015) Comparing message passing interface and mapreduce for large-scale parallel ranking and selection. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 3858–3867. doi:10.1109/WSC.2015.7408542

Ni EC, Ciocan DF, Henderson SG, Hunter SR (2017) Efficient ranking and selection in parallel computing environments. Oper Res 65(3):821–836. doi:10.1287/opre.2016.1577

Ni EC, Henderson SG, Hunter SR (2014) A comparison of two parallel ranking and selection procedures. In: Tolk A, Diallo SD, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, pp 3761–3772. doi:10.1109/WSC.2014.7020204

Ni EC, Hunter SR, Henderson SG (2013) Ranking and selection in a high performance computing environment. In: Pasupathy R, Kim S, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference. IEEE, Institute of Electrical and Electronics Engineers, Inc, Piscataway, NJ, 833–845. doi:10.1109/WSC.2013.6721475

Pasupathy R, Ghosh S (2013) Simulation optimization: a concise overview and implementation guide. In: Topaloglu H (ed) TutORials in operations research, chap 7. INFORMS, Catonsville, MD, pp 122–150. doi:10.1287/educ.2013.0118

Pasupathy R, Hunter SR, Pujowidianto NA, Lee LH, Chen C (2014) Stochastically constrained ranking and selection via SCORE. ACM Trans Model Comput Simul 25(1):1–26. doi:10.1145/2630066

Paulson E (1964) A sequential procedure for selecting the population with the largest mean from k normal populations. Ann Math Stat 35(1):174–180

Peng Y, Chen C, Fu MC, Hu J (2016) Dynamic sampling allocation and design selection. INFORMS J Comput 28(2):195–208. doi:10.1287/ijoc.2015.0673

Pinedo M (2015) Scheduling. Springer

Rinott Y (1978) On two-stage selection procedures and related probability-inequalities. Commun Stat Theory Methods A7:799–877

Ryzhov IO (2016) On the convergence rates of expected improvement methods. Oper Res 64(6):1515–1528. doi:10.1287/opre.2016.1494

Schmeiser B (1982) Batch size effects in the analysis of simulation output. Oper Res 30(3):556–568

Singham DI, Szechtman R (2016) Multiple comparisons with a standard using false discovery rates. In: Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 501–511

Turnquist MA, Sussman JM (1976) A Bayesian approach to the design of simulation experiments. In: Highland HJ, Schriber TJ, Sargent RG (eds) Proceedings of the 1976 bicentennial winter simulation conference, pp 59–64. http://informs-sim.org/wsc76papers/1976_0011.pdf

Wang H, Pasupathy R, Schmeiser BW (2013) Integer-ordered simulation optimization using R-SPLINE: retrospective search using piecewise-linear interpolation and neighborhood enumeration. ACM Trans Model Comput Simul 23(3): doi:10.1145/2499913.2499916

Xu J, Nelson BL, Hong LJ (2010) Industrial Strength COMPASS: A comprehensive algorithm and software for optimization via simulation. ACM Trans Model Comput Simul 20:1–29

Yoo T, Cho H, Yücesan E (2009) Web services-based parallel replicated discrete event simulation for large-scale simulation optimization. Simulation 85(7):461–475

Chapter 13
A History of Military Computer Simulation

Raymond R. Hill and Andreas Tolk

Abstract The history of military simulation dates back to the earliest ages of conflict. Throughout the history, military personnel has employed cognitive devices to comprehend and plan for conflict. This chapter provides a history of military simulation, starting with a review of the board-based war games, progressing through the computerization of those war games and completing with a discussion of the modern world of military simulation encompassing desktop-based analytical simulations, video games, and distributed training and analytical environments. It provides a background for the contributions of the military track that has been a pillar of the Winter Simulation Conference for many years.

13.1 Introduction

The military is a big consumer of models and simulation. In the United States (US) Department of Defense (DoD) alone, there are over 3300 simulations registered as in use. This number is quite reasonably a lower bound given not all simulations are registered. Arming, training, and preparing the DoD to accomplish its myriad tasks is incredibly complex. The DoD, as with the military organizations of all nations, comprises multiple components, each of which might feature multiple commands with thousands of personnel across hundreds of career fields, operating and supporting thousands of systems to accomplish their broad range of missions. One of those missions, armed conflict, is arguably the most complex of human endeavors. Since the history of conflict between groups, clans, tribes, on up to nations parallels the history of mankind, it is not surprising that the use of simulation as a decision aid to help understand and prepare for conflict has a similarly long history.

R.R. Hill (✉)
Air Force Institute of Technology, Hobson Way, Wright-Patterson AFB, OH 2950, USA
e-mail: rayrhill@gmail.com

A. Tolk
The MITRE Corporation, Hampton, VA, USA
e-mail: atolk@mitre.org

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_13

The immediate reaction to the above statement was likely something along the lines of "that is not possible, computers are too new of a device." This of course is true, but when presenting a history of military computer simulation one must really start with the use of thought-based or group-discussion models by the military, and that type of use does not require the computer, it merely requires the use of experienced, thinking individuals.

Within this chapter, we want to give our own interpretation of an overview of the developments that led to computer simulation first, as many ideas developed in the early phases still dominate our conceptual view of simulation to this day, such as figures with well-defined capabilities following a set of common rules on a common game board. We then look into some of the main developments in a computer simulation that influenced work presented and discussed in the military track before giving a short history of the military track itself.

This chapter has been written from the viewpoint of professionals who organized, conducted, and participated in the military track of the Winter Simulation Conference over several years. We are well aware of many historical overviews of military simulation, many of them written to commemorate anniversaries or special recognitions, such as Bergin (2000), Davis (2010), Thorpe (2010), or Shiflett (2013). Several chapters in textbooks are giving historical perspectives too, such as Little (2006), Banks (2009), and Loper and Turnitsa (2012). These are only examples of additional approaches to capture the history of military simulation, and many additional worthwhile publications are published not covered in this enumeration. All these works, like ours, are necessarily incomplete. We, however, seek to provide a more comprehensive chronology of military simulation and provide an overview of the contributions of the Winter Simulation Conference and its Military Track, to this history.

Our history of military computer simulation starts with a brief look at the beginnings of the war games and war planning efforts that qualify as thought-based simulation modeling and analysis. From that, we will progress into the early use of training devices for simulation and the early use of the computer for the computational implementation of training and war games. At that point, we will then move into the modern era of military simulation; the era explicitly involving the computer and all its inherent power and flexibility.

13.2 War Games Preparing the Way for Simulation

Humans are incredible problem solvers. We consider problems facing us and consider how various courses of action might solve the problem. With experience, we typically get more efficient, and effective, at the problem-solving process. Fortunately, or unfortunately depending upon your point of view, combat experience is not easily gained meaning other mechanisms are required to build up military experience necessary for military success.

This motivation to find another mechanism to hone military skills led to the early development of "war games." As far back as 3000 B.C., the game of Wei Hai employed colored stones to represent opposing forces. Personnel would position and move the stones to represent the deployment and employment of forces. The simulated conflict would be resolved, or "adjudicated" based on expert judgment. The game could be used to practice and hone strategic planning skills or to plan upcoming engagements. The game of chess similarly evolved out of military war games and references are found as early as 500 B.C. in India. The game was used by leaders, and their subordinates, again to hone their strategic planning skills (Smith 2010).

These board games evolved over time, growing in size and complexity, to meet changing needs. The games eventually evolved into table top versions more aligned with group deliberations on plans or actions. Early Vikings and Celtics are credited with explicitly considering various scenarios using these table top games (e.g., imagine a map laid out on a table with various pieces placed to represent forces (Smith 2010)). These games were likely largely adjudicated based on the experience of those arrayed around the playing table. Such an approach would likely have led to decisions biased by the beliefs of those involved, particularly those more senior in the group.

In the mid-1600s, these board games had evolved to include more independent adjudication of game moves, with the rules of these decisions based often on actual combat experience. Kriegsspiel, appearing around 1811, is arguably viewed as the first real war game simulation as dice were employed to introduce randomness into the outcomes of the moves implemented during the game (Loper and Turnitsa 2012). The left side of Fig. 13.1 depicts a modern version of a game while on the right is a drawing of the German staff employing the game in their planning efforts.

Over time, game complexity increased. New Kriegsspiel in 1798 involved 3,600 squares on the playing board and 60 pages of rules governing the conduct of the game. By the early 1800s, the Prussians had replaced the board game with map-like charts, moving pieces, realistic movements, and randomness in the outcomes (via a dice roll). This hard-to-learn game came to be known as Rigid Kriegsspiel and was replaced in the late 1800s by the easier to learn, faster to play, essentially rule

Fig. 13.1 On the left, a depiction of a Kriegsspiel board game. On right, depiction of German staff engaged in Kriegsspiel (PreparedX 2017)

free version which came to be known as Free Kriegsspiel. This Free version stepped back in complexity to rely more on player experience and judgment to adjudicate the moves made by the players. Despite the detail involved and use of actual combat data to define the rules, the Rigid form was simply too complex to learn and too slow to play. The Free form moved quicker and became the favored approach finding success through WWII (Shrader 2006). In America, war games arose in the late 1800s having similar detailed rule structures as found in the European games. While used in military schools, the operational impact of these games was minimal due to the excessive time required to perform the requisite mathematical calculations (Loper and Turnitsa 2012). Furthermore, General W.T. Sherman, as the Commanding General of the US Army, still under the impression of the brutal encounters of the Civil War that required many more human factors than could be captured in wargaming, has been said to discourage the use of this approach by stating: Men are not wooden blocks!

Throughout history, war gaming played a significant role in military training and planning. Many of the successful military campaigns were thoroughly war gamed (Shrader 2006). It is unclear whether the unsuccessful campaigns were not war gamed; losers in military actions rarely get their history told. Improving the accuracy of these war games was possible and was actually achieved, but humans can only calculate so fast. Thus, as noted above with the American experience, the detailed war games were found too cumbersome for regular use. What was needed was a way to free the modeler from the tedious calculations required by the rigid games allowing them to experience the flow of the Free games with the accuracy of the Rigid games.

13.3 Military Computer Simulation

Modern computer simulation ties back to the military needs of WWII. It was the intense calculations associated with military system engineering and analysis that really raised interest in mechanical computing calculators. Areas such as ballistics and crypto-analysis, which had required many hours of manual calculations, could be done in seconds when using the automated device. For instance, calculating a 60 s missile trajectory required 20 h of manual calculations, but just about 30 s on the mechanical calculator (Shrader 2006). The Electronic Numerical Integrator and Calculator (ENIAC) appeared in 1946 and is often viewed as the first modern computer.

As related by Metropolis (1987), the combination of statistical sampling and this new advanced calculating machine led to what we now routinely call Monte Carlo simulation. In those formative years, this computerized statistical sampling method was used to calculate the complex equations arising in the US Nuclear Program. Not surprisingly, Monte Carlo simulation was initially a classified method. Clearly, computer calculations changed the nature of computing and with its engineering and development. Motivated by early success, computing technology continued to meet the increasing demand.

The business world took notice of the computer and its capabilities as well. This naturally prompted more rapid growth in the technology. By the 1950s and 1960s, computer use, albeit of the very large, time-sharing type, were quite commonplace. In the military, optimization and simulation were among the most popular of the analytical uses of the computer (Shrader 2006).

Computing technology enabled the return of the more rigid war games by reducing the time required for the necessary calculations. Detailed war games became feasible (and practical) providing the growing cadre of operational researchers with a means of iteratively hypothesizing, learning, and improving their understanding of military weapon systems and their employment. The US Army experience is a great example, the Army Operations Research Office (ORO) in particular.

13.3.1 The Computer Mainstreams the War Game

In the early 1950s, the Army ORO became an early adopter of the computer-facilitated war game. These computerized games were largely used as a research tool, not really for the development of strategy or doctrine. The various divisions within the ORO focused on issues associated with nuclear attack and response, tactical battle capabilities, intelligence management, logistics planning, and continental air defense (Shrader 2006). One of the premier models, Carmonette, examined company-level ground operations involving infantry, tank operations, and mortar operations. Despite the early limitations of computing systems, Carmonette could represent up to about 200 entities in the simulated battles (Shrader 2006).

Computer support for wargaming initially involved running the calculations required to determine move outcomes based on the input from the players. The games provided insight to develop models to answer questions regarding resources and operations. Some of the early computer games really were intended to provide just enough insight to facilitate defining a more comprehensive wargaming approach. These early computer games were digital representations of the board games; however, it did not take a long time for the modelers to realize they could build more mathematically rigorous representations of the scenario. Soon games focused on the effects of weapons, systems, tactics, and logistics, among other aspects. Thus, the use of the computer quickly evolved from helping the top-level nature of the wargame (infer details from the high-level game response) to an environment in which the model output could provide insight on all levels (Shrader 2006).

A problem with wargames is repeatability and replication. It is hard for human participants to fully repeat their actions, especially when those participants believe another approach may work better. It is also tough to do the same task many times, even if there are allowances for freedom of action. Computers, however, are quite adept at repeatability and replication. The full automation of wargames thus arose as a natural consequence of the needs for military analysis and thereby created the area of analytical (now more often called constructive) simulation that arguably dominates the military use of simulation.

13.3.2 Rise of the Analytical Simulations

As military modeling and simulation matured, coupled with the subsequent advances in computing machinery, military simulation quickly grew associated with quantitative representations of combat in computer programs. The "art" of military modeling, actually military science in general, is the integration of modeling and simulation output with operational context to generate meaningful insights useful for decision making. In the growing technology of computer-based military simulation, this art closely aligned with the continual challenge of trading off simulation detail with the uncertainties associated with those details.

Changes in simulation modeling details, as the simulation models moved further away from just automating wargames, involved adding fidelity in the systems representations, adding greater diversity in the systems simulated within some scenario, and expanding the time frame considered in the simulation scenario. It turns out there is a very close relationship among the number of entities modeled and the corresponding level of detail and time period considered. This correspondence leads to initial characterizations of these emerging simulations as few-on-few, many-on-many and force-on-force (Battilega and Grange 1984).

Carmonette developed in the 1950s focused on small unit ground combat. Factors modeled included weapons aiming, target acquisition, the firing of the weapon and an assessment of target strike and destruction (Battilega and Grange 1984). An example analysis involving Carmonette would be tank duels or interactions between two ground force platoons.

In the mid-1970s, models of air, combat seemed to have really come into use. TAC AVENGER modeled a two aircraft air duel. The aircraft performance was modeled using fairly detailed engineering data. Aircraft engaged each other using a gun and/or missile systems. The corresponding projectiles from each system were modeled at engineering level detail. Aircraft within the simulation chose maneuvers, determined weapons to employ, and the firing actions to take. End-game outcomes assessed the projectile impact and kill probabilities against the specific target.

During the early 1900s, systems of differential equations were developed to describe force-on-force combat, as summarized among others in Taylor (1983). These abstract concepts were well suited to combat simulation use in force-on-force situations since engineering level detailed data was not necessary for their use. By the mid-1970s, force-on-force, or campaign-level models, used these systems of Lanchester equations to model large-scale combat. BALFRAM was an early example of such a model. It integrated land, sea, and air forces to examine aspects of combat analysis such as force planning, systems utilization, and of course operational effectiveness.

These early models collectively helped examine myriad scenarios to help answer a wide variety of questions. Naturally, the nature of any question–answer dialog is the generation of yet more questions. Couple this growth in the number, type, and depth of analysis needed to answer the questions with the rapid growth in computing capabilities, and it is quite logical to find that the military simulation models quickly

expanded in size, scope and complexity. Model expansion involved greater modeling details, larger-sized forces and greater time periods modeled. In addition, the models improved in terms of data required and its fidelity, graphical displays of the combat scenario modeled, and the output data generated and subsequently processed into information used to inform decision making.

Over time, these models came to be classified into four broad categories: engineering models, engagement models, mission models, and campaign models. These categories differentiated models by the level of detail in the modeling of simulation entities, the number and diversity of the forces considered, and the time frame captured in the simulation.

Engineering simulation models featured the greatest level of detail and consider systems over a very short time frame. Examples include a torpedo or missile fly-out. The engineering models will often capture environmental aspects as well as the physics involved in the system operation. These models might even involve human input in configurations known as human-in-the-loop simulation. Typical model outputs are aggregated for use in other models.

Engagement simulation models are concerned with short duration events, such as an air-to-air duel, a tank duel or the firefighter between two ground force squads. These encounters typical last minutes to hours. Engineering detail, often aggregated from engineering model output, are used to accurately capture the physics as well as the operational aspects of the scenario. They are usually applied on the tactical level.

Mission simulation models are concerned with events that last hours and involve potentially many diverse systems. These models have less engineering detail than in the previously described models as they are typically more concerned with the operational aspects of the interactions among these disparate systems (Hill et al. 2001). Exceptions, of course, exist such as the engineering level detail that might be required in a mission model used to assess the impact of radar-based protection systems; such simulations would need to model the radar signal at a fairly high level of fidelity. As a rule, these models cope with the operational level.

Campaign simulation models cover the longest period of time, encompass the largest number of assets, in multiple mediums and multiple service components. A campaign model will usually model Army assets at the Corps level, Naval forces, and full Air Force wings. The focus is on the battles that comprise a war against some adversary and thus the time frame in these simulations will generally be on the order of weeks to months. These models focus on the strategic level, including questions of logistics and long-term sustainability of military operations.

A common model of these military simulation categories arose in the mid-1990s and is often referred to as the DoD Simulation Pyramid or the Hierarchy of Models (Hill et al. 2001). Figure 13.2 is one such instance of the hierarchy. Each of the four levels depicts a category of military simulation (as labeled to the right of the pyramid). As we move up the pyramid, the models are increasingly aggregated in terms of model detail; model fidelity increases as we move down the pyramid. The shape of the pyramid also captures the general use of the simulations within each category. The wider, lower levels cover a much broader range of issues than do the narrow levels, whose models have a more focused purpose. Entries within each level

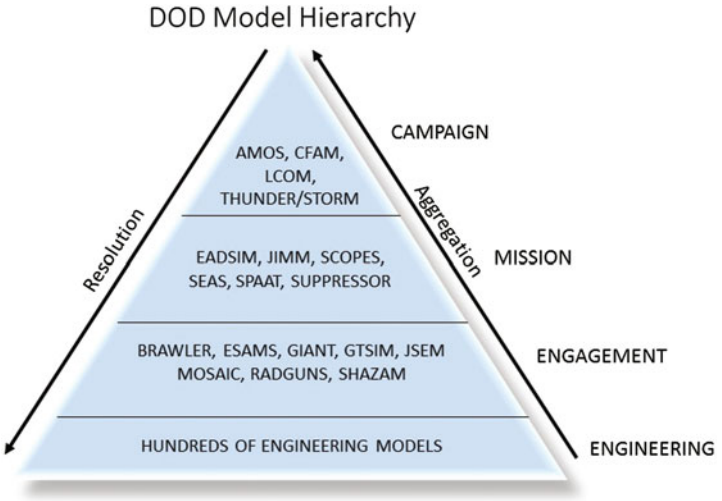


Fig. 13.2 The well-know model pyramid adopted and extensively used within the US DoD simulation world. The pyramid depicts the four general categories of models, how the categories vary in terms of modeling resolution and aggregation, and provides modern examples of models within each category

are representative models associated with that level/category. In general, results from lower levels are aggregated and used as inputs to models on higher levels.

This construct of families of models required human intervention to move data (and scenarios) among the models. A natural challenge to the technologists thus arose, why not let the simulations interconnect, or "talk to each other" to remove the time and labor intensive process of modeling data among the models. The answer to this simple question required complex software and communications engineering but yielded a result that now drives the predominant form of military simulation.

13.3.3 Military Simulation Goes Distributed

While computer simulation for war gaming was often considered a topic for special defense conferences, and many papers dealing with analytical simulation targeted the military operations research community, the rise of distributed simulation was in the focus of the simulation community. Military training was and is a challenging environment that draws lots of research interest and was also the main topic featured in WSC. With the maturing of computer technology, distributed simulation transformed the way the armed forces conducted planning, training, testing, and many of its other functions and tasks. In parallel to the growing importance of computers for command and control tasks, the development of training systems utilized, modernized, and sometimes even revolutionized the use of computer simulation. Thus,

distributed simulation augmented the traditional methods of war gaming, although war gaming still has its place in many modern applications of military problem domains, such as in cybersecurity (Turnitsa 2016).

13.3.3.1 SIMNET and Distributed Interactive Simulation

The modern story of distributed simulation starts in 1983 with the Simulator Networking (SIMNET) program. It was initiated by the Defense Advanced Research Projects Agency (DARPA) as one of the first attempts to exploit the developments in communications technology for simulation. First-hand accounts are given by Miller and Thorpe (1995) and Cosby (1995). The idea was to network a group of simulators together and exchange information that would allow the teams using these simulators to collaborate like they would do in their operational systems. This was surely a DARPA project, as the best the US Air Force could manage at the time were two simulators, and the SIMNET first target was using 20 simulators (Hapgood 1997). SIMNET objectives were to bring armor, mechanized infantry, helicopters, artillery, communications, and logistics components together into a common, situated, virtual battlefield. Simulator crews were supposed to observe each other, communicate via radio channels, and observe each other effects. Per Loper and Turnitsa (2012), SIMNET was based on six design principles:

- Object/Event Architecture—the world is modeled as a collection of objects which interact using events.
- Common Environment—the world shares a common understanding of terrain and other cultural features.
- Autonomous Simulation Nodes—simulations send events to other simulations and receivers determine if that information is relevant.
- Transmission of Ground Truth Information—each simulation is responsible for local perception and modeling the effects of events on its objects.
- Transmission of State Change Information—simulations transmit only changes in the behavior of the object(s) they represent.
- Dead Reckoning Algorithms—simulations extrapolate the current position of moving objects based on its last reported position.

The demonstration used commercially available computer networks to interconnect simulators and represent the virtual world graphically. The companies Delta Graphics, Inc., Perceptronics, Inc., and Bolt, Beranek and Newman (BBN), Inc., were contracted with the development of graphics, network, and vehicle simulators. Three years after the project start, a platoon-level system had been developed, and after 3 more years, approximately 250 simulators were used by the US Army for their team training in Fort Benning, GA, Fort Rucker, AL, Fort Knox, KY, Fort Leavenworth, KS, and in Grafenwöhr, Germany. These technical and application success stories spawned growing interest with industry, and soon after the demonstrations, the IEEE 1278 Standard on Distributed Interactive Simulation (DIS) emerged (Calvin et al. 1993). The idea was to keep the standard easy to understand, easy

to implement, and open for future developments. It was developed over a series of Workshops, mainly organized by the Institute for Simulation and Training (IST) of the University of Central Florida in Orlando. In 1993, the first version of this standard was agreed upon. The participation of the US Army Simulation Training and Instrumentation Command in Orlando, FL, ensured the applicability of these standardized solutions for the military customer. The design principles of DIS remained the same as those of SIMNET. As the focus had been on proving the principle and feasibility in applicable form for SIMNET, with DIS it shifted to the production of good guidelines and the definition and standardization of protocol data units (PDU): a standardized information exchange specification with fully agreed upon syntax and semantics.

DIS rapidly became a worldwide standard that was adapted for countless simulation systems and continues to support military training. Many of the new methods include migration support by providing DIS interfaces or gateways to facilitate the integration of simulation systems supporting the DIS principles. Additional details are provided in U.S. Congress, Office of Technology Assessment (OTA) (1995).

13.3.3.2 The Aggregate Level Simulation Protocol

While DIS focused on the networking of simulators, DARPA also recognized the need to support computer assisted exercises (CAX). As described by Cayirci and Marincic (2009), a CAX is an exercise using computer models designed to place the command and control element of a headquarters in a realistic, stressful combat-like environment to stimulate decision making, command and control staff interaction and coordination. While the focus of simulator training lies on training of individuals and small teams operating the weapon system simulated, the whole headquarter with its different cells and command and control systems builds the training group. Supporting a CAX, therefore, required another approach to interconnect the operational environment used by these Headquarters personnel. Thus, they defined the infrastructure that supports the military user with all information necessary to optimize the decision process with a focus on Command and Control Information Technology.

To support this new application domain, DARPA initiated the Aggregate Level Simulation Protocol (ALSP) extending distributed simulation to the force-level training community. In contrast to the tactical level supported by DIS, different aggregation levels of unit representations are possible, the exercises were distributed over a larger geographic domain potentially worldwide, and causality played a more important role. Because of the evaluation of these new requirements, the ALSP recognized that various time management schemes and more complex simulated object attribute management requirements were needed. The resulting design principles were the following (Weatherly et al. 1991):

- Simulations need to be able to cooperate over a common network to form confederations.
- Simulations must be able to publish objects of common interest and subscribe to objects they are interested in. Objects controlled by other simulation systems are ghosted.
- Within a confederation, temporal causality must be maintained.
- Simulations should be able to join and exit a confederation without major impact on the balance of the other participating simulations.
- The system should be network-based with no central controllers or arbitrators.
- Interactions among participating components do not require knowledge of confederation participants and should support an object-oriented view of interactions.

To implement these principles, ALSP focused on developing a special communication infrastructure, the ALSP Infrastructure Software (AIS), allowing for an extended set of services as well as a new format for the information exchange elements, the Interface Control Document (ICD), provided the capability to communicate the higher variety of information to be exchanged between the headquarters and the supporting combat simulation systems. The AIS comprised two software module categories with different tasks.

- The ALSP Common Module (ACM) was the interface to the simulation systems. It provided the interface to exchange messages in form of human readable text between the ACM and the simulation systems. These messages are defined in the ICD. This provided a high flexibility regarding what type of information could be exchanged, but also required a higher degree of coordination when the ICD content was agreed upon in the preparation phase for an exercise.
- ALSP Broadcast Emulator (ABE) provided the infrastructure services to orchestrate the execution of the distributed exercise. These software modules did not interface with the simulation systems, but they connected the ACM. ABE provided a set of services to support confederation, data, time, and event management.

The resulting ALSP specification was successfully implemented and supported over several years. The Joint Training Confederation (JTC) was the largest application of ALSP. The JTC has been used since 1992 to train military officers all over the world, including the United States, Germany, Korea, and Japan. In 1997, twelve simulation systems from varied armed services participated in a JTC worldwide exercise (Prochnow et al. 1997). Although ALSP was never standardized by an official standardization body, it proved new concepts and their feasibility shaping the current parallel and distributed simulation community. The reason for not going for standardization was the emergence of another new idea: the development of a common simulation interoperability standard that could merge both worlds of DIS and ALSP, the Standard for Modeling and Simulation (M&S) High Level Architecture (HLA).

13.3.3.3 The High-Level Architecture

The successful DIS and ALSP experiments did lead to the realization by the US Congress that distributed simulation technology could yield great benefits to the Department of Defense (DoD). The National Defense Authorization Act of 1991 called for a joint office to *"establish a coordinated DoD-wide approach to simulations and training devices for both acquisition and training..., to establish interoperability standards and protocols, and to develop a long-term plan to guide the development of simulators and training devices."* The Defense Modeling and Simulation Office (DMSO) was soon established to foster joint interoperability and model reuse among the armed service M&S efforts. The development of common simulation interoperability and related standards was a high priority objective. After review of alternative solutions, an Architecture Management Group (AMG) was established in 1995 to develop the High-Level Architecture (HLA), with significant political and financial support. This program was very ambitious. The DoD Joint Requirements Oversight Council (JROC) originally even planned to stop funding for simulation efforts that would not support the new standard after a certain adjustment time, and even exclude non-compliant solutions completely from the use in the DoD. This idea proved to be infeasible over time and was not enforced, but it shows the seriousness of the efforts.

In parallel to these national efforts, DMSO was also active within NATO. DMSO actively pushed for the development of a NATO M&S Master Plan (MSMP) that would ensure international support of the new vision of a common family of standards that would enhance the multi-national training capabilities. The Conference of National Armament Directors (CNAD) chartered a Steering Group on NATO Simulation Policy and Applications in 1996 and developed the MSMP, which was endorsed by the Military Committee and the CNAD, approved by all NATO nations, and issued in December 1998 by the North Atlantic Council. The MSMP also identified HLA as the common standard.

The design principles of HLA are described by Kuhl et al. (1999). They are realized in all implementations.

- Simulation systems communicate via the common Runtime Infrastructure (RTI) via standardized interfaces. Simulation systems are called federates and build together with the RTI the federation.
- The information exchange elements are defined using the standardized Object Model Template (OMT) which defines persistent elements (objects with attributes) and transient elements (interactions with parameters). Ten basic rules define the principles for this information exchange.
- The RTI provides services for the management of the federation, time, and the information exchange between the simulation systems and ensures consistency within the federation:
  - Federation management: Creating, joining, and managing federations, saving and restoring federations, and synchronizing federates.

- Declaration management: Defining publication and subscription of information exchange elements types for the federate.
- Object management: Defining the use of instantiated objects and interactions information exchange element objects for the federate.
- Ownership management: Defining ownership of objects, including how to transfer it.
- Time management: Defining the time paradigms and their synchronization.
- Data distribution management: Defining constraints allowing for the optimization of data traffic between federates.

The development and standardization of HLA happened in various phases. Supported by the US DoD, the first phase resulted in the definition of the HLA 1.3 NG. To foster a high adoption rate, the necessary software packages developed for the prototypical implementation were distributed for free to interested members of the simulation industry.

With the acceptance by NATO, the international community pushed for an international standard, comparable to the IEEE 1278 DIS standard, which at the time was the standard of choice for NATO simulation groups. Because of this international effort, HLA was submitted to IEEE for standardization, and this resulted in the IEEE 1516–2000 HLA standard family. Within the standardization process, the HLA 1.3 NG solutions were generalized and elevated to the current technologies, i.e., the number and possible values of calibration parameters were generally captured as extensible enumerations, and the definition of structures was transformed from Backus-Naur-Form to XML.

The review of the HLA standard family 10 years later resulted in additional adaptations of new technical solutions, such as semantic web technologies and increased modularization of HLA components. The updated IEEE 1516–2010 HLA is the current version.

Beside the technical specifications of the guiding rules, the interface between RTI and federates, and the structure of the information exchange in form of the OMT, the standard family comprised also guidelines for the federation development and execution processes (FEDEP) as well as how to conduct verification and validation for federations.

13.3.3.4 The Test and Training Enabling Architecture and Mixed Approaches

In parallel to the HLA, another effort to provide increased benefits led to the development of the Test and Training Enabling Architecture (TENA). In contrast to the HLA, TENA purposefully focuses on support of test ranges for military applications (Powell and Noseworthy 2012). TENA allows for the definition of an object-oriented Logical Range Object Model that avoids, similar to the standardization of PDUs within DIS, ambiguities. The TENA philosophy is based on the understanding that interoperability requires not only a common architecture, but also the ability to

meaningfully communicate, which requires a common language and a common communication mechanism. Furthermore, a common context in form of a common understanding of the environment, a common understanding of time, and a set of common technical processes is needed. The TENA infrastructure provides integrated solutions, plus it provides places to store reusable components in form of TENA repository. This systemic support for interoperable solutions comes with a price: TENA is difficult to extend beyond the focus of test and training on ranges. While HLA had the objective to be broadly applicable for all simulation paradigms and all application domains, TENA was designed to optimize the support of its application domain.

The recent years of distributed simulation development are characterized by the insight that one common standard that fits all purposes is unlikely to be ever accomplished. This resulted in the increased implemented and utilization of mixed approached that allow for the use of various simulation interoperability standards within the same architecture. The resulting multi-domain architectures connect the various solutions using mainly proxies or gateways. To support these endeavors, the HLA specific FEDEP has been generalized to support such mixed approaches, resulting in the IEEE 1730–2010 Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). DSEEP has been extended to support Multi-Architecture Overlays that support more than one simulation interoperability standards.

Distributed simulation is now accepted as a training support that is expected and taken for granted by members of the armed forces. Pilots train and practice on simulators before they enter the aircraft for many hours. The latest US fighter F-35 does not even have a trainer version with an extra seat for the instructor any more. Instead, the new pilot learns everything in the simulator, including flight formations with others within a distributed training environment. Within NATO, the use of CAX to train and practice international units is standard practice. Bruzzone and Massel (2017) present a view of the military history of distributed simulation from the NATO perspective.

13.4 The Military Track of the Winter Simulation Conference

Over the last few decades, there have been just a few outlets for publicizing military simulation activities and results. Arguably the leading conference on simulation has been the Winter Simulation Conference. Initiated in 1967, the WSC has gained the respect of simulation professionals worldwide, especially with its focus on high quality papers and presentations. There are other quality conferences with a defense focus: Western Decision Sciences Institute, the Annual Conference of the Institute of Industrial and Systems Engineers, the Summer Computer Simulation Conference (SCSC), and the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), recognized as the world's largest modeling, simulation, and training

conference. The symposiums of the Military Operations Research Society (MORS) also contribute to simulation-related research. However, our focus here is on the WSC.

The very first WSC in 1967 featured a military keynote while the general chair was with the Air Force. Various military papers appeared in those early years. An interesting paper on a pure military topic was presented at the WSC75 in the Simulation for Government track on the topic of "A Security Force-Adversary Engagement Simulation" by H.A. Bennett from the Sandia Laboratories (Bennett 1975).

At the WSC77, the session on Military systems was introduced, involving two papers. Alfonso A. Diaz from the US Training and Doctrine Command presented "A Cost and Operational Effectiveness Analysis of the Army Utility Tactical Transport Aircraft System," (Diaz 1977) and the paper on "The Generation and Use of Parameterized Terrain in Land Combat Simulation" was presented by Sam H. Parry from the Naval Postgraduate School (Parry 1977).

Although some papers with military references were presented thereafter, it was not before 1983 that a permanent Military Application session was introduced to provide a permanent home for such application studies. These sessions attracted between two and five papers per year, with peaks in 1988 (12 papers) and 1991 (9 papers). It was not until 1993 that the Military Application topic attracted enough papers that a track with several sessions could be established in the WSC program. This increase was likely connected with the congressional recognition and following support of modeling and simulation, as described in the distributed simulation subsection.

In 1995, the Military Track introduced its own keynote session with a clear focus on defense related issues. There were a significant number of technical HLA-related papers during the years of 1995 and 2000, but they never dominated the track. However, motivated by the new technical perspective, more technical papers were submitted that not only focused on the application domain, but evaluated new technologies or methods regarding their applicability.

Since 1998, the Military Application track was conducted as one of the main tracks in the WSC, with full day sessions for all days of the conference. Since 2007, the numbers started to drop some and sessions were combined together with the Homeland Security track, which was established in 2004. Although many interesting papers continued to be presented, the number of accepted papers dropped to 10 in 2014, so that the program committee decided to merge the Military Application and the Homeland Security tracks into the new track on Military, Homeland Security & Emergency Response, covering all aspects of defense and security application in a common track, again with full day sessions for all days of the conference. Figure 13.3 provides a summary of the activity in the military track over the years. These numbers do not include papers with military application topics that were featured in other tracks.

Since its instantiation, the Military Application track has conducted keynotes, panel discussions, and paper sessions highlighting the requirements, benefits, and technology challenges and solutions. New concepts such as distributed simulation, multi-resolution and multi-domain modeling, high resolution visualization,

Fig. 13.3 A compilation of the papers falling under the Military Track for the Winter Simulation Conference. The number of papers is provided for each bar. Note that for many years, the Military Track has featured a keynote speaker, meaning no or at most one paper is associated with a full session

applicability of web-based applications, cloud-based, and many more were presented and discussed in the international context. Tutorials were also supported.

Early military papers in the WSC were parametric, or deterministic in nature. Early efforts by Bennett (1975) and Link and Shapiro (1979) examined small force engagements based on difference equations or detailed scripts. Mekaru and Barclay (1984) modeled deep space intercepts while Graves and Clark (1983) modeled the budget planning process. However, early languages like Q-GERT, SLAM, and Simscript allowed military modelers to take advantage of computer simulation capabilities. Parry (1978) used Simscript to examine battalion-level engagements, Mortenson (1981) used Q-GERT to model aircraft repair centers (called depots), and Armstrong et al. (1983) used SLAM to study mobility aircraft scheduling. Clark et al. (1984) seem to provide the first WSC military paper using experimental design to examine model results, in their case focused on logistics policy analysis. This work preceded the tutorial nature of the Roberts and Morrissey (1986) work that explained using design of experiments to examine a targeting algorithm.

As simulation capabilities evolved so did the application of these capabilities in papers featured in the military track. By 1987, training simulation discussions really started appearing; Childs and Lubaczewski (1987) discuss such a system for battalion and brigade commanders. By 1988, the military simulation track seemed to take hold with its first truly sizable track, and this growth resulted in a military track with focused sessions by 1992. Those sessions in 1992 interestingly enough were focused on airlift, wargaming, and decision support based on simulation modeling; topics that are still examined today.

Combat support-focused topics began appearing about this same time. Schuppe (1989) examined pilot workload, which continues to be a major research issue. Human factors modeled associated with various aspects of the military mission garnered 1–2 sessions per track in the early 2000s. The influence of DIS and ALSP appeared in the military track by 1994 with a focus paper on the topic in 1999 (Nance 1999). Unmanned vehicle research and agent-based modeling applications are hot topics in 2017; these emerged in the military applications track by the early 2000s. The military track even seems to have the distinct honor of having

a Winter Simulation session in the Winter Simulation Conference (it was a session focused on seasonal issues in military operations).

Over the years, the Military Track has featured numerous important defense leaders as keynote speakers, held five important panel discussions, one involving the senior operational research analyst from each of the services, and provided numerous papers covering a plethora of important topics to military planning and operations. The current Track on Military, Homeland Security & Emergency Response continues to be a pillar of the annual WSC program and a track that influences military decision making with its scientific contributions.

13.5 Examples for Current Challenges

Given the long history of military simulation and the broad range of specific topics that fall into the realm of military simulation, any history of military simulation is necessarily incomplete. In this history, we have tried to touch on many of the influences in a military simulation that have led to current state of military computer simulation. Fortunately for those involved in military simulation, now is an exciting time. There are a wide variety of challenges and opportunities in the military simulation. Our intent here in this last section is to highlight some of these opportunities and challenges.

13.5.1 Live, Virtual and Constructive Simulation

The tremendous advances in distributed simulation have evolved into an environment in which live assets can communicate with training systems involving humans in the loop (virtual assets) and with purely analytical models (constructive models). The resulting LVC environment provides access to the range and number of military systems not available on the physical test ranges. Thus, LVC holds tremendous promise for the system demonstration and training challenges in the future. The LVC also holds promise for the test and evaluation world. In such applications, new systems can be embedded in the complex, interconnected military environment envisioned for the future while communicating and interacting with the systems of today, all within some common environment to ascertain the new system appropriateness in a system-of-systems context. The latter application requires new ways of thinking about the humans in distributed simulations, but does provide an opportunity to fundamentally change the way the military tests and considers new weapon systems and the various support systems (see discussion in Hodson and Hill (2013)). This research must address how to fully integrate games into the LVC. Games and their physical engines can provide valuable functionality to federations, as shown by Valerde and Sun (2017). Games also have been proven useful for rapidly generating sufficiently realistic behavior of entities without requiring a huge amount

of personnel to do so. Games are providing valuable means to support generating realistic visual representation, e.g., of avatars in simulated teleconferences, or realistic video streams of simulated drones that fly through synthetic environments. Finally, the use of augmented reality opens new training possibilities by mixing simulated reality with real environments. Current research focuses more on convergence instead of simple integration.

13.5.2 Web-Based and Cloud-Based Simulation

The military track featured several papers on the efficient use of web-based and cloud-based technology, such as Cayirci (2013). With the progress of general computational methods supporting distributed and high performance computing, the adaption of such technologies in support of military and defense simulation is gaining increased interest. M&S as a service is often seen as the possible technology solution enabling the convergence of solutions, although conceptual challenges remain, as shown in Taylor et al. (2015).

This does not only address the use of such emerging technologies within simulation systems and to support their distribution and execution, but also tools that allow better governance of such complex endeavors: how to manage the development and execution of such simulation systems? How do these new developments affect ideas of common services to be shared with partners, definition of reusable simulation components in repositories, etc.

13.5.3 Computational Social Sciences

The human element has always been a key determinant of military success or failure. Throughout the history of military simulation, we have included human behavior in our analyses. The revolutionary changes brought about by the computer in the 1940s and the 1950s did much to improve our mathematical representation of combat, but our representations of the human in that combat scenario has been slow to catch up. The use of agent-based simulations has brought about tremendous advancements in the social sciences and some of these advances will surely find their way into combat modeling. There has already been steps in this direction particularly with the work in the Naval Postgraduate School SEED Center—promoting simulation experiments and efficient designs—and prior to that, the work under Project Albert. Currently, the use of generative simulation, as envisioned in Epstein (1999) and today increasingly supported by Agent Zero model implementations Epstein (2006). The usefulness for defense related analysis, but also potentially for training and education, is a topic of current research.

13.5.4 Unmanned Assets

Unmanned assets, particularly unmanned aerial vehicle (UAV) assets hold great promise for future combat operations. Unfortunately, it is unclear which parts of that promise are achievable and which parts are hype. The use of advanced simulations can go a long way to discerning the achievable and not so achievable aspects of the autonomy challenge facing the military in the future. One of the pressing issues in this context is the need to command and control mixed units, made up of human soldiers as well as robots, including the necessary human-robot interfaces needed to enable efficient combat operations. These concepts also must be represented in training systems, so that officers learn how to utilize such mixed units efficiently. Several aspects of these challenges are currently evaluated by the Modeling and Simulation for Autonomous Systems (MESAS) workshops organized by the NATO Center of Excellence, see among others Blais (2016).

13.5.5 Other Emergent Challenges

The topics listed in this section are neither complete nor exclusive. Many additional challenges have been and are identified, and their number grows steadily. Among these are the following ones.

- The community is still looking for good solutions for multi-level security protocols that allow data sharing in an exercise with all participants on various levels of trust. This includes multi-domain challenges as well.
- Urban operations in Megacities will become a challenge, as more and more people are moving into big cities, mostly on the coast. While traditional military operations in the age of the Cold War avoided urban areas, in the future the combat in such environments will become a likely option. Burns et al. (2015) address the fragility of the Global Positioning System in such an urban scenario.
- Many scenarios already include massive cyberattack activities, but only as an event, not as a simulated operational activity. Cyber warfare needs to be modeled by itself as well as an integrated activity of combat operations. This includes, but is not limited to, the cyber-attacks against command and control systems. Unfortunately, there is still the question of just how to model cyber.

Many more topics can be added to this list of emergent challenges. Over the years, the military application domain has proven to be one of the most challenging fields to be supported by M&S. The problems are complex and require solutions that draw from research from many related domains in the technical as well as the operational realm. New technologies and methods continuously contribute to innovative solutions that are worth to be presented and discussed with international peers, in particular as the focus of military operations is no longer limited to the traditional combat sphere.

13.6 Concluding Remarks

The focus of this chapter was to provide a framework of main events and topics, not the presentation of main research results. However, many methods and tools resulting from the research such as they are presented and discussed in the military track are presented in more detail in textbooks and research reports. Examples for such textbooks are Bracken et al. (1995), Cayirci and Marincic (2009), Deitz et al. (2009), Washburn and Kress (2009), Strickland (2011), and Tolk (2012). The interested reader is referred to this literature for further studies.

This chapter is titled "A History" for a specific reason; it is by no means neither a complete nor a comprehensive history. It is "A" history because its content and presentation is influenced heavily by the biases and experiences of the authors. It is fully expected that anyone reading the history presented will react with a "Why wasn't N" included, and the reaction is fully justified. For instance we did not cover the tremendous infusion of funding into modeling and simulation through organizations like the Defense Modeling and Simulation Office or the hundreds of millions of dollars spent on the Joint Simulation System (JSIMS) (Bennington 1995) or the Joint Warfare System (JWARS) (Stone and McIntyre 2001). We also chose to not discuss the incredible wide range of engineering models used in weapon systems design, in weapon system survivability studies, or in weapon systems lethality studies, to name just three areas. We also left off important areas like gaming technology to military use or simulation support to large-scale exercises; these could constitute entire theses on their own. Our intent was to layout a cogent flow of how military planning and analysis has evolved into the computer-reliant complex that exists today and entertain a little along the way.

To this end, our chronological coverage is more complete than found in some of the other histories and we provide insight into the evolution of the Winter Simulation Conference, Military Track, and the contributions made in that forum to the field of military simulation.

Acknowledgements This research was supported in part by the Office of the Secretary of Defense, Directorate of Operational Test and Evaluation and the Test Resource Management Center under the Science of Test research program. Disclaimer: The authors affiliation with the Air Force Institute of Technology and The MITRE Corporation are provided for identification purposes only, and are not intended to convey or imply AFITs or MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the authors. The views expressed do not reflect the official policy or position of the United States Air Force, U.S. Department of Defense or the U.S. Government. The publication has been approved for public release and unlimited distribution: case number 17-2080.

References

Armstrong GR, Coleman JW, Hamilton VH, Poe JW (1983) Priority dispatch and aircraft scheduling: A study of strategic airlift scheduling. Proceedings of the 1983 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 579–587

Banks CM (2009) What is modeling and simulation. In: Sokolowski JA, Banks CM (eds) Principles of modeling and simulation. Wiley, Hoboken, NJ, pp 3–24

Battilega JA, Grange JK (1984) The military applications of modeling. Air Force Institute of Technology, Wright-Patterson AFB, OH

Bennett HA (1975) A security force-adversary engagement simulation. Proceedings of the 1975 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 285–291

Bennington RW (1995) Joint simulation system (jsims): An overview. Proceedings of the 1995 national aerospace and electronics conference. Institute of Electrical and Electronics Engineers Inc., Dayton, OH, pp 804–809

Bergin TJ (2000) 50 years of army computing from eniac to msrc. Technical report, DTIC Document

Blais C (2016) Challenges in representing human-robot teams in combat simulations. In: Hodicky J (ed), international workshop on modeling and simulation for autonomous systems. Springer International Publishing, pp 3–16

Bracken J, Kress M, Rosenthal RE (1995) Warfare modeling. Wiley, Incorporated

Bruzzone AG, Massel M (2017) Simulation based military training. In: N this information (ed), guide to simulation-based disciplines—advancing our computational future. Springer International Publishing, pp. 315–361

Burns A, Miller JO, Hill R (2015) Using seas to assess gps constellation resiliency in an urban canyon environment. In: Proceedings of the 2015 winter simulation conference. Piscataway, New Jersey

Calvin J, Dickens A, Gaines B, Metzger P, Miller D, Owen D (1993) The simnet virtual world architecture. Proceedings of the virtual reality annual international symposium. Institute of Electrical and Electronics Engineers Inc., Hoboken, New Jersey, pp 450–455

Cayirci E (2013) A joint trust and risk model for msaas mashups. In: Proceedings of the 2013 winter simulation conference. Piscataway, New Jersey

Cayirci E, Marincic D (2009) Computer assisted exercises and training: a reference guide. Wiley, New York, NY

Childs C, Lubaczewski T (1987) A battalion/brigade training simulation. Proceedings of the 1987 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 876–885

Clark TD, Prueitt GC, Smith RL (1984) A policy analysis for logistical support of multiple U.S. Amry contingency requirements. In: Proceedings of the 1984 winter simulation conference. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, pp 723–726

Cosby L (1995) Simnet: an insiders perspective, ida document d-1661. Institute for Defense Analyses, Defense Technical Information Center, Washington, DC

Davis PK (2010) Paradigm-level issues in m&s: historical lessons and current challenges. In: Interservice/industry training, simulation, and education conference (I/ITSEC) 2010, p Paper No. IF1002

Deitz PH, Reed HL Jr, Klopcic JT, Walbert JN et al (2009) Fundamentals of ground combat system ballistic vulnerability/lethality. The American Institute of Aeronautics and Astronautics, Reston VA

Diaz A (1977) A cost and operational effectiveness analysis of the army utility tactical transport aircraft system. Proceedings of the 1977 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 413–421

Epstein J (1999) Agent-based computational models and generative social science. Complexity 4(5):41–60

Epstein J (2006) Generative social science: studies in agent-based computational modeling. Princeton University Press, Princeton, NJ

Graves S, Clark R (1983) A dynamic long range budget model to assist navy planners. Proceedings of the 1981 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 663–666

Hapgood F (1997) Simnet. Wired https://www.wired.com/1997/04/ff-simnet/. Accessed May 3, 2017

Hill RR, McIntyre GA, Miller JO (2001) Applications of discrete event simulation modeling to military problems. In: Peters BA, Smith JS, Medeiros DJ, Rohrer MW (eds) Proceedings of the 2001 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 780–788

Hodson DD, Hill RR (2013) The art and science of live, virtual, and constructive simulation for test and evaluation. J Def Model Simul 41(1):6–19

Kuhl F, Weatherly R, Daymann J (1999) Creating computer simulation systems: an introduction to the high level architecture. Prentice Hall, Upper Saddle River, NJ

Link BD, Shapiro HD (1979) Tsem, a flexible scenario based small forces model. Proceedings of the 1979 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 331–345

Little D (2006) History and basics of m&s. In: Integration of modeling and simulation. Educational Notes RTO-EN-MSG-043, Neuilly-sur-Seine, France, pp 1.1–1.4

Loper ML, Turnitsa CD (2012) History of combat modeling and distributed simulation. In: Tolk A (ed) Engineering principles of combat modeling and distributed simulation. Wiley, Hoboken, NJ, pp 331–355

Mekaru MM, Barclay RC (1984) Direct ascent, deep space intercept model. Proceedings of the 1984 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 745–751

Metropolis N (1987) The beginning of the monte carlo method. Los Alamos Sci, 125–130

Miller DC, Thorpe JA (1995) Simnet: the advent of simulator networking. Proc IEEE 83(8):1114–1123

Mortenson RE (1981) Maintenance planning and scheduling using network simulations. Proceedings of the 1981 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 333–340

Nance RE (1999) Distributed simulation with federated models: Expectations, realizations and limitations. Proceedings of the 1999 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 1026–1031

Parry SH (1977) The generation and use of parameterized terrain in land combat simulation. Proceedings of the 1977 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 423–431

Parry SH (1978) Star: Simulation of tactical alternative responses. Proceedings of the 1978 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 563–569

Powell ET, Noseworthy JR (2012) The test and training enabling architecture (tena). In: Tolk A (ed), Engineering principles of combat modeling and distributed simulation. Wiley, New York, pp 449–477

PreparedX (2017) PreparedX. http://www.preparedex.com/war-gaming-part-1-the-history/. Accessed March 15, 2017

Prochnow D, Page E, Fischer MC (1997) Management of the joint training confederation family of specifications. In: Proceedings of the simulation interoperability workshop. Society for Computer Simulation, Baltimore, MD

Roberts BH, Morrissey DW (1986) The use of statistical experimental design techniques for the systematic improvement of an automated, heuristic targeting algorithm. Proceedings of the 1986 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 802–807

Schuppe TF (1989) A comparison of serial and parallel processing simulation models of pilot workload. Proceedings of the 1989 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 1003–1009

Shiflett JE (2013) Observations on the development and implementation of distributed simulation. In: Interservice/Industry training, simulation, and education conference (I/ITSEC) 2010. p. Paper No. 1F1301

Shrader CR (2006) History of operations research in the united states army, Volume I: 1942–1962, office of the deputy under secretary of the army for operations research, Washington, DC

Smith R (2010) The long history of gaming in military training. Simul Gaming 41(1):6–19

Stone GF, McIntyre GA (2001) The joint warfare system (jwars): a modeling and analysis tool for the defense department. In: Proceedings of the 2001 winter simulation conference. Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp 691–696

Strickland J (2011) Mathematical modeling of warfare and combat phenomenon. Lulu Enterprises Inc., Raleigh, NC

Taylor JG (1983) Lanchester models of warfare. Operations Research Society of America

Taylor SJ, Khan A, Morse KL, Tolk A, Yilmaz L, Zander J, Mosterman PJ (2015) Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens. Simulation 91(7):648–665

Thorpe JA (2010) Trends in modeling, simulation, & gaming: Personal observations about the past thirty years and speculation about the next ten. In: Interservice/Industry training, simulation, and education conference (I/ITSEC) 2010. p Paper No. IF1001

Tolk A (2012) Engineering principles of combat modeling and distributed simulation. Wiley, Hoboken, NJ

Turnitsa CD (2016) Adjudication in wargaming for discovery. CSIAC J Cyber Secur Inf Syst 4(3):28–35

U.S. Congress, Office of Technology Assessment (OTA) (1995). Distributed Interaction Simulation of Combat, Report OTA-BP-ISS-151, U.S. Government Printing Office, Washington D.C

Valerde O, Sun L (2017) Through the virtual barrier: Virtual prototyping and analysis with model-based systems engineering. In: Proceedings of the spring simulation multi-conference. San Diego, CA

Washburn AR, Kress M (2009). Combat modeling. Springer

Weatherly R, Seidel D, Weissman J (1991) Aggregate level simulation protocol. In: Proceedings of the summer computer simulation conference. Society for Computer Simulation, Baltimore, MD

Chapter 14
Modeling and Analysis of Semiconductor Manufacturing

John W. Fowler and Lars Mönch

Abstract This chapter provides a brief history of semiconductor manufacturing papers in the Winter Simulation Conference (WSC) along with some other key events that led to the development of a community of academicians and semiconductor manufacturing practitioners focused on modeling and analysis of semiconductor manufacturing. Among these influential events were the first Modeling and Analysis of Semiconductor Manufacturing (MASM) conferences. The MASM conference became a conference within the WSC in 2008. We examine this by considering one decade at a time starting with the 1980s.

14.1 Introduction

The demand for and uses of semiconductor devices is constantly expanding. We continue to have new and improved consumer electronics which are powered by semiconductor devices. Increasingly, a larger and larger percentage of the content of transportation and communication systems depends on semiconductor devices. The Internet of Things revolution has been made possible by the vast array of semiconductor devices.

A semiconductor device is a highly miniaturized, integrated electronic circuit consisting of thousands of components. The fabrication of semiconductor devices (or Integrated Circuits—IC's) is surely among the most complex manufacturing processes in existence (cf. Mönch et al. 2013). This complexity is caused by many factors including multiple products, routes with several hundred process steps, batch processing, and a large number of machines. A batch is a set of lots that are processed at the same time on a single machine.

J.W. Fowler (✉)
Arizona State University, Phoenix, USA
e-mail: john.fowler@asu.edu

L. Mönch
University of Hagen, Hagen, Germany

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_14

In the early days of the semiconductor industry, all that was necessary for a semiconductor company to make money was to design a good product. However, over the last three decades, increased competition has required semiconductor companies to also be able to manufacture their products in a more efficient and cost-effective manner.

Semiconductor companies have increasingly turned to data-intensive modeling and analysis tools and techniques because of their potential to significantly improve manufacturing performance, and hence the bottom line. The semiconductor manufacturing modeling and analysis community has been working over the last 30 years to modify general purpose manufacturing modeling tools and techniques to handle the intricacies and complexities of semiconductor manufacturing.

This chapter provides a brief history of semiconductor manufacturing papers in the Winter Simulation Conference (WSC) along with some other key events that led to the development of a community of academicians and semiconductor manufacturing practitioners focused on modeling and analysis of semiconductor manufacturing. Among these influential events were the first Modeling and Analysis of Semiconductor Manufacturing (MASM) conferences. The MASM conference became a conference within the WSC in 2008. We examine this by considering one decade at a time starting with the 1980s.

14.2 The 1980s

By the mid-1980s, the semiconductor industry was highly competitive with numerous players that had to begin to compete on dimensions other than just product design. Cost and ability to deliver product were becoming increasingly important. In response to these pressures, a few companies begin to use discrete event simulation to help manage their wafer fabrication facilities. Dayhoff and Atherton (1984) was one of the earliest papers to discuss the use of discrete event simulation to analyze semiconductor manufacturing operations. A couple of years later, Dayhoff and Atherton (1986) described the use of simulation to estimate the impact of various dispatching schemes on the inventory, cycle time, and throughput of a wafer fab.

In 1987, the SEMATECH consortium was formed as a jointly funded partnership between the Defense Advanced Research Projects Agency (DARPA) and 14 United States (US) semiconductor manufacturing companies in response to a growing share of the semiconductor market being captured by Japanese companies. DARPA wanted to ensure the domestic capability to produce IC's would persist. As part of the startup, SEMATECH established several Centers of Excellence. Two of these included a modest amount of funds to study semiconductor manufacturing operations: one at The University of California Berkeley (UC Berkeley) (led by Roger Glassey and Robert Leachman) and the other at Texas A&M University (led by Don Phillips, Guy Curry, and Bryan Deuermeyer). This funding was recognition by the companies that they needed to improve operations to remain competitive and

sent a signal to academic researchers that obtaining funding to do research in this area was possible.

Apparently, the first WSC semiconductor manufacturing paper was by Peter Waller of Intel (cf. Waller 1986). This paper describes the process of building a discrete event simulation model of a manufacturing facility and gives an example using the process for a wafer fab. In 1988, there were two WSC semiconductor manufacturing papers. Prasad and Rangaswami (1988), two Intel employees, analyze different automated guided vehicle (AGV) systems using simulation. Pitts (1988) describes the use of simulation to improve operations at a research and experimentation wafer fabrication.

The year 1989 saw a doubling of the number of semiconductor manufacturing papers at WSC. The four papers included a 3-paper session entitled "Scheduling for Semiconductor Manufacturing Lines". This session included a paper from UC Berkeley (Glassey and Petrakian 1989), a paper from AT&T Bell Labs (Johri 1989), and a paper from IBM (Miller 1989). In addition, there was a paper (Hood et al. 1989) by an IBM T.J. Watson Research Center researcher (Sarah Hood) and two consultants from Systems Modeling Corporation (Amy Amamoto and Antonie Vandenberge) that described the development of a detailed generic simulation model of a wafer fab. We note that most of the documented efforts in the 1980s were done by the companies rather than by academics.

14.3 The 1990's

There were two papers in the 1990 WSC proceedings. The first, Atherton et al. (1990), discusses the first wafer fabrication facility simulator (ACHILLES) and a cluster tool simulator (THOR). The ACHILLES software contains the logic laid out in a US Patent. The second 1990 paper (Hood and Welch 1990) discusses how to employ experimental design techniques to study wafer fab operations via a simulation model. Baum and O'Donnell (1991) was the only WSC semiconductor paper in 1991. It discussed the modeling of labor and the impact of machine down times in wafer fabrication at National Cash Register NCR.

In 1991, SEMATECH formed an Operational Modeling group that was led by Mohammad Ibrahim, an assignee from Advanced Micro Devices (AMD.) This group played an instrumental role in increasing the use of simulation (and other operations research techniques) in the semiconductor industry. Over the next 5 years, members of the group included Neal Pierce (Harris Semiconductor), Jerry Weckman (former Texas Instruments employee), John Fowler (former AMD intern), John Konopka and Shekar Krishnaswamy (IBM), and Ricki Ingalls (former Compaq employee). Tom Jefferson, who was the 2008 WSC General Chair and a long-time Intel employee, and Scott Mason (Clemson Professor and long-time MASM contributor) were interns with the group. Lee Schruben (UC Berkeley) and Pres White (University of Virginia) each spent a year in the group as a visiting professor. The Operational Modeling group had three main activities: (1) providing

training to member company personnel just starting to employ operations research tools and techniques, (2) developing operations research tools for the companies, and (3) supporting internal SEMATECH projects with modeling.

Among the training activities were a half day short course for managers taught by Averill Law and two separate 2.5-day short courses taught by David Kelton and Lee Schruben for simulation analysts. Tool development included (1) managing a project with Systems Modeling Corporation to develop the Wafer Fabrication Template (one of the first Arena templates); (2) internal development of a supply chain optimization model called the Manufacturing Enterprise Model; and (3) the internal development of the Semiconductor Workbench for Integrated Modeling (SWIM), which was built to manage the flow of data used by individual models and the flow of data between models.

One of the major projects undertaken by the SEMATECH Operational Modeling group was the Measurement and Improvement of Manufacturing Capacity (MIMAC) project. This project was done in conjunction with the Joint European Submicron Silicon Initiative (JESSI) consortium. In the project, a 2-day workshop on the relationship of cycle time and capacity was developed and delivered in London, Munich, Nantes, Boston, Austin, and Phoenix. As a part of the project, the researchers developed a standard format for an operational model of a wafer fab. Professor Robert Leachman (UC Berkeley) was contracted to develop datasets of six wafer fabs that he had visited in the Sloan Study for Competitive Semiconductor Manufacturing that he was co-leading at the time. These data sets were made available to researchers (and software developers) so that ideas they had for improved wafer fab models could be tested with real data. The datasets have been used extensively by the research community ever since. While the datasets are now more than 20 years old and do not fully reflect today's wafer fabs, they are still the best source of data representing wafer fabrication operations. The data sets are now hosted by Professor Lars Mönch at the University of Hagen along with more recently developed datasets of backend operations and supply chain operations. The data sets can downloaded at http://p2schedgen.fernuni-hagen.de/index.php?id=242.

In 1992, Reha Uzsoy, Chung-Yee Lee, and Louis Martin-Vega (cf. Uzsoy et al. 1992) published part 1 of a two-part series of papers that reviewed models developed for production planning and scheduling in the semiconductor industry. This paper raised the awareness of the complexity and challenges faced by semiconductor manufacturers. According to Google Scholar on April 29, 2017 it has been cited more than 650 times and is still frequently cited.

Returning to WSC papers, there were six WSC papers with one from SEMATECH and the rest from semiconductor companies (including one from Japan). There were three, four, and one papers in 1993, 1994, and 1995, respectively. All but two of these were from semiconductor companies; one was from a software vendor (LeBaron and Pool (1994) and one from academia (Schömig and Mittler 1995). Two of the papers dealt with automated material handling (Nadoli and Rangaswami 1993 and Pierce and Stafford 1994) and two with cluster tools (Mauer and Schelasin 1993 and LeBaron and Pool 1994). It should be noted that during this time the Advanced Semiconductor Manufacturing Conference (ASMC) and the

Semicon West conference began to publish papers devoted to semiconductor manufacturing operations.

Another data set that has been used extensively by the research community was introduced by Dr. Karl Kempf and a colleague from Intel Corp (cf. Spier and Kempf 1995). The data set is called the Mini-Fab model and it consists of only six steps and five machines in three workstations. Despite its limited size, it contains most of the key operations elements of a wafer fab: multiple products, batch and serial steps, significant setups, reentrant flow, operators, and downtime. This data set and variants of it have been used to test out numerous planning and scheduling methodologies over the last 20 years.

The 1996 WSC had two firsts regarding semiconductor manufacturing. The first of these is that Dr. Karl Kempf gave the keynote speech that was titled "Simulating Semiconductor Manufacturing Systems: Success, Failures, and Deep Questions". The second is that this was the first WSC with two full sessions devoted to semiconductor manufacturing. Five of the six papers were from industry.

The year 1997 saw the first of three consecutive 3-year research programs launched. The program was co-funded by the National Science Foundation (NSF) and the Semiconductor Research Corporation (SRC) at a level of $1.2 M per year, for 3 years (total of $3.6 M) and was called "Operational Methods in Semiconductor Manufacturing". There were three research teams funded: (1) Cornell University—principal investigator (PI) Lee Schruben; (2) University of Maryland—PI Michael Fu; (3) A collaborative project between Arizona State University (ASU) (PI-John Fowler), Massachusetts Institute of Technology (PI-Stan Gershwin), and the University of Illinois (PI-P.R. Kumar). The topics covered are shown in Fig. 14.1. There were numerous Master's and PhD. Students funded in this program. Most of the students either went to work for a semiconductor company or became a professor and continued to research operations issues in semiconductor manufacturing. The funding of this program led directly and indirectly to more academic research in this area and ultimately to more semiconductor manufacturing papers at WSC.

While there was only a single semiconductor paper at WSC 1997 (cf. Jain and Chan), there was (effectively) a three-session mini-track in 1998 and a panel discussion on "Operational Modeling and Simulation in Semiconductor Manufacturing"

Fig. 14.1 Topics covered by NSF operational methods in Semiconductor Manufacturing program

Decision Area	ASU	CU	MIT	UI	UM
Capacity Expansion & Tool Acquisition	√	√	√		√
Demand Forecasting		√			
Order Release & Scheduling	√		√	√	
Performance Evaluation		√	√	√	√
Tool & Workforce Mgmt.	√	√	√		
Process Issues					√

(cf. Fowler et al. 1998). Four of the nine papers were authored by academic teams and one other was a joint industry-academia paper. In 1999, there was formally a four-session semiconductor manufacturing mini-track with five papers by companies, four from academic institutions, and three joint industry-academia papers.

To wrap up the decade, a precursor to the Modeling and Analysis of Semiconductor Manufacturing (MASM) conference was held as part of the Society of Computer Simulation's Western Multi-Conference. The conference was called the "International Conference on Semiconductor Manufacturing Operational Modeling and Simulation (SMOMS'99)". As implied by the name, the focus was not just on simulation, but all operations research tools and techniques. The chairs of the conference were John Fowler (ASU), Jeffery Cochran (ASU), and Courtland Hilton (Intel). The keynote presentation was by Ditmar Kranzer (Siemens AG—now Infineon Technologies) and the topic was "The Need for Operational Modeling and Simulation in Semiconductor Manufacturing". There were about 100 attendees from around the globe with an approximately equal mix of industry and academic folks. There were 36 papers with the following breakdown of topics:

- Performance Evaluation—11
- Cluster Tools and Equipment Availability—3
- Scheduling—5
- Material Handling—4
- Yield and Process Modeling—3
- Batching and Assembly, Packaging and Test Modeling—2
- Cost Modeling—3
- Capacity Modeling—5.

14.4 The 2000s

The momentum generated at the end of 1990s continued in terms of funded research and academic participation into the 2000s. The first International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM 2000) was held on May 10–12 in Tempe, Arizona. The conference chairs were John Fowler (ASU) Jeffery Cochran (ASU), Steven Brown (Infineon Technologies), and Ching-En Lee (National Chiao Tung University, Taiwan). Professor Robert Leachman (UC Berkeley) was the keynote speaker; however, due to fog in the Bay area, his talk was delivered by John Plummer from National Semiconductor. There were over 200 attendees from around the globe with an approximately equal mix of industry and academic participants. There were 59 papers with the following breakdown of topics, number of papers, and track chairs:

- Equipment Productivity—15—Robert Leachman (UC Berkeley), Tae-Eog Lee (Korea Advanced Institute of Science and Technology (KAIST)), Hans Wimberger (Infineon)

- Operational Modeling and Simulation—23—Javier Bonal (Lucent-Madrid), Devadas Pillai (Intel)
- Statistical Methods—13—Elart von Colani (University of Würzburg), Paul Tobias (SEMATECH), T. Tada (Mitsubishi)
- Supply Chain Management—8—Sanjay Jain (Gintic), Dan Shunk (ASU).

The 2000 WSC had a (then) record number of 15 papers with nine of them by a university team and four industry-academia papers. Note that this was the first WSC that had more academic than industry talks. There was a full session (three papers) on equipment modeling and another one on material handling. This trend continued at WSC 2001 where there were 12 papers with eight from academia and three that were industry-academia papers.

Also in 2001, there was a second SMOMS conference with John Fowler (ASU), Jeffery Cochran (ASU), Javier Bonal (Agere Systems), and Tae-Eog Lee (KAIST) as the conference chairs. There were two plenary presentations: one by Sarah Hood and Stuart Bermon of IBM and one by David Anderson and Walt Trybula of SEMATECH. There were 26 papers with the following breakdown:

- Simulation Methodology—4
- Planning and Scheduling—4
- Scheduling and Dispatching—4
- Performance Evaluation—4
- Supply Chain Management—3
- Equipment Level Modeling—2
- Backend and Electronics Modeling—3

A second major research funding program was started in 2001. The **F**actory **O**perations **R**esearch **Ce**nter (FORCe) was jointly funded by the Semiconductor Research Corporation and International SEMATECH (it had recently become an international consortium with no government funds) at $800 K per year for 3 years. International SEMATECH worked with Brooks Automation to develop a generic 300 mm wafer fab model for use by the research teams. There were five projects funded in this program. Note that two of the projects involved universities outside the US. The subject of each project along with the universities and PI's are given below:

- New approaches to simulation of wafer fabrication—Arizona State University (Jerry Mackulak), UC Berkeley (Lee Schruben)
- Preventive maintenance scheduling in semiconductor manufacturing fabs—University of Maryland (Michael Fu, Steve Marcus), University of Cincinnati (Emmanuel Fernandez)
- Demand data mining and planning in semiconductor manufacturing—National Taiwan University (Argon Chen, Shi-Chung Chang, Andy Guo)

- Scheduling of semiconductor wafer fabrication facilities—Arizona State University (John Fowler), University of Arkansas (Scott Mason), University of Würzburg (Oliver Rose), Technical University of Ilmenau (Lars Mönch), and Fraunhofer Institute for Manufacturing Engineering and Automation (Roland Sturm)
- Demand Forecasting (IBM custom funding)—Cornell University (Robin Roundy)

The goal for each project was to spend the first year of the project understanding the problem and benchmarking current approaches. The second year was devoted to developing algorithms and solutions. Finally, the third-year funding was for implementation and to explore the possibility of commercialization.

The second MASM conference was held on April 10–12, 2002 in Tempe, Arizona. The committee chairs were John Fowler (ASU), Jerry Mackulak (ASU), Alexander Schömig (Infineon Technologies), and Der-Bau Perng (National Chiao Tung University—Taiwan). The keynote talk was given by Devadas Pillai from Intel and it emphasized challenges that lie ahead with the introduction of 300 mm wafers. There were 74 papers with the following breakdown of topics, number of papers, and track chairs:

- Equipment Productivity—10—Tae-Eog Lee (KAIST), Oliver. Rose (University of Würzburg), Jeffery Cochran (ASU)
- Operational Modeling and Simulation—39—Sarah Hood (IBM), Dominique Mercier (ST Microelectronics), Ron Billings (International SEMATECH)
- Statistical Methods—14—Paul Tobias (International SEMATECH), George Runger (ASU)
- Supply Chain Management—11—Karl Kempf (Intel), Argon Chen (National Taiwan University)

The next three WSC's (2002, 2003, 2004) each had 12 papers. There was a mix of academic and industry papers in these years, but there was increased international participation.

In 2004, the National Science Foundation joined the Semiconductor Research Corporation, and International SEMATECH in funding the Factory Operations Research Center II (FORCe II) at a level of $1.2 M per year for 3 years. The projects funded and the PIs are given below:

- Fab-wide Control and Disruption Management in High Volume Semiconductor Manufacturing—S. J. Qin, G. Yu, J. Hasenbein, E. Kutanoglu (University of Texas at Austin)
- Incorporating Nonlinear Phenomena in Semiconductor Supply Chain Planning Models—Reha Uzsoy, Ron Rardin, J. Asmundsson, (Purdue University)
- Hierarchical Modeling of Yield and Defectivity to Improve Factory Operations—Doug Montgomery (ASU), Christina Mastrangelo (University of Washington)

- Demand Planning and Supply Chain Coordination in the Contract Manufacturing Environments—S. David Wu, Rosemary Berger (Lehigh University)
- Multi-product Cycle Time and Throughput Evaluation via Simulation on Demand—Bruce Ankenman, Barry Nelson (Northwestern), John Fowler (ASU), Jerry Mackulak (ASU)

Unfortunately this was the last major funding program for semiconductor manufacturing operations (to date).

The 2005 MASM conference was held in Singapore on October 6–7, 2005. Peter Lendermann (Singapore Institute of Manufacturing Technology) was the General Chair, John Fowler was the Program Chair, and Amit Kumar (Singapore Institute of Manufacturing Technology) was the Proceedings Chair. There were well over 200 attendees from around the globe. KC Ang from Chartered Semiconductor Ltd. gave a keynote address on automation in the 300 mm nanotechnology fab. There were 51 papers with the following breakdown:

- Equipment Productivity—14 papers
- Operational Modeling and Simulation—18 papers
- Statistical Methods—8 papers
- Supply Chain Management—5 papers
- Enabling Computing Technologies—6 papers

A special issue of the *International Journal of Production Research* (Vol. 45, No. 3, 2007) was produced with papers from the conference.

The 2005, 2066, and 2007 WSC's had 9, 13, and 12 papers, respectively. A number of the papers presented were as a result of the FORCe programs. In 2007, there were 23 papers that were part of a MASM track that was held as part of the 2007 IEEE (The Institute of Electrical and Electronics Engineers) International Conference on Automation Science and Engineering (CASE 2007). The conference was held in Phoenix and the overall best paper in CASE 2007 was a MASM paper that dealt with multiple cluster tools (cf. Chan et al. 2007).

After several years of holding separate MASM conferences, in 2008 the MASM conference found its current home as part of the Winter Simulation Conference. MASM is more than just a track within WSC because papers that do not use simulation are allowed. The MASM chairs were John Fowler (ASU), Lars Mönch (University of Hagen), and Chen-Fu Chien (TSMC). There was a panel session with a title "Modeling and Analysis of Semiconductor Manufacturing in a Shrinking World: Challenges and Successes" with panelists Chen-Fu Chien (TSMC), Stéphane Dauzère-Pérès (Ecole des Mines de Saint-Etienne), Hans Ehm (Infineon Technologies AG), John W. Fowler (Arizona State University), Zhibin Jiang (Shanghai Jiao Tong University), Shekar Krishnaswamy (AMD), Lars Mönch (University of Hagen), and Reha Uzsoy (North Carolina State University). This

session lead to a paper in the *European Journal of Industrial Engineering* (cf. Chien et al. 2011). There were a total of 44 papers in 16 sessions. The breakdown is as follows:

- Operational Modeling and Simulation (9 sessions/24 papers)
- Supply Chain Management and Fab Economics (5 sessions/14 papers)
- Enabling Computing Techniques and Statistical Methods (2 sessions/6 papers).

A special issue of the *European Journal of Industrial Engineering* (Vol. 5, No. 3, 2011) was published with papers from the MASM 2008 conference.

Making MASM a conference within WSC was well received by all and it was agreed that it would continue. Scott Mason (Clemson University) was the MASM chair in 2009. Karl Kempf (Intel) gave the keynote. There was a panel session entitled "Are Simulation Standards in Our Future?" with Hans Ehm (Infineon), Leon McGinnis (Georgia Institute of Technology), and Oliver Rose (Dresden University of Technology) as panelists. There were 8 sessions and 18 papers. GlobalFoundries provided funds for a small reception.

14.5 The 2010s and Beyond

As of 2010, the MASM Conference had become a regular conference within the Winter Simulation Conference. Table 14.1 shows the number of sessions and papers for each MASM conference this decade along with the Conference Chair(s) and the keynote speaker or the title of a panel. An informal group of colleagues from Asia, Europe, and the United States decide on who will be the next group of conference chairs. In 2012 and since 2014, there have been three co-chairs with one from each of the three regions: Asia, Europe, North America. Journal papers that were extensions of MASM 2012 and IEEE Automation Science and Engineering (CASE) 2012 papers were published in a special issue of *Computers & Operations Research* (2015, Vol. 53). In 2012, Infineon Technologies hosted a reception for the MASM conference and there has been a reception in all but one WSC conference since then.

Figure 14.2 shows the number of semiconductor manufacturing papers at WSC since 1996. Clearly, there has been a large increase in the number of papers since MASM became part of WSC in 2008. MASM will continue to be a part of WSC for the foreseeable future. Over the decades there has been a slight shift in the topics covered in the WSC and MASM Conferences. There are relatively fewer papers today at the equipment level and relatively more at the supply chain level. The fact that a Dagstuhl Seminar was held on February 7–12, 2016 that was focused on "Modeling and Analysis of Semiconductor Supply Chains" (http://www.dagstuhl.de/16062) is evidence of the increasing emphasis by the MASM community on supply chain issues. Anyone who is interested in getting involved in the MASM community is encouraged to contact either of the authors of this chapter.

Table 14.1 MASM conference information for 2010–2017

Year	Sessions	Papers	MASM Chair(s)	Keynote speaker or panel
2010	5	14	David Jimenez, Wright Williams & Kelly, Inc.	
2011	12	36	Stéphane Dauzère-Pérès (Ecole des Mines de Saint-Etienne) John Fowler (Arizona State University)	Panel: Challenging Issues and Emerging Trends in Modeling and Analysis of Semiconductor Manufacturing
2012	12	33	Argon Chen (National Taiwan University), Andy Ham (GLOBAL FOUNDRIES), Lars Mönch (University of Hagen)	Kurt Gruber (CVP Corporate Supply Chain, Infineon Technologies AG)
2013	9	27	Claude Yugma (Ecole des Mines de Saint-Etienne), Jesus A. Jimenez (Texas State University)	Impacts of Imminent Changes in the Semiconductor Industry, Julian Richards (SEMATECH)
2014	10	29	John Fowler (Arizona State University), Lars Mönch (University of Hagen), James R. Morrison (KAIST)	(Almost) Present at the Creation: 25 Years of Modeling and Simulation in Semiconductor Manufacturing, Reha Uzsoy (North Carolina State University)
2015	9	14	Jesus Jimenez (Texas State University), Gerald Weigert (Technische Universität Dresden), Kan Wu (Nanyang Technological University)	MASM: A Look Back and a Peek Ahead, John Fowler (Arizona State University)
2016	8	22	Stéphane Dauzère-Pérès (Ecole des Mines de Saint-Etienne), Adar Kalir (Intel), Reha Uzsoy (North Carolina State University)	The Engineering of Speed and Delivery, Robert C. Leachman (University of California at Berkeley)
2017	TBD	TBD	John W. Fowler (Arizona State University), Lars Mönch (University of Hagen), Youin Choung (Ajou University)	Achievements and Lessons Learned from a Long-term Academic-Industrial Collaboration, Stéphane Dauzère-Pérès (Ecole des Mines de Saint-Etienne)

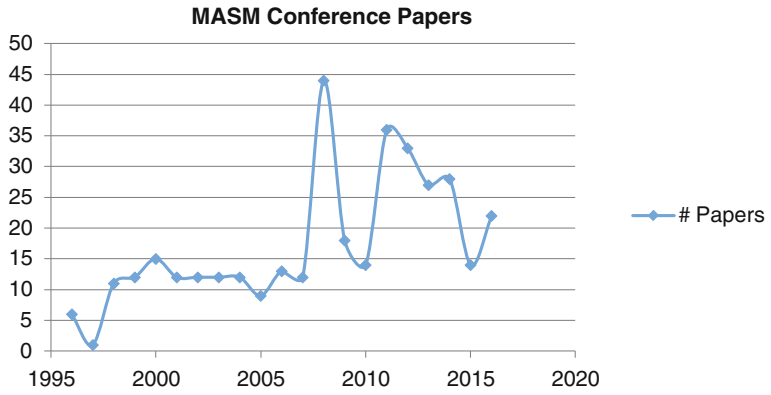


Fig. 14.2 MASM papers at WSC 1996–2016

References

Atherton RW, Atherton LF, Pool MA (1990) Detailed simulation for semiconductor manufacturing. In: Proceedings of the 1990 winter simulation conference, pp 659–663

Baum SS, O'Donnell CM (1991) An approach to modeling labor and machine down time in semiconductor fabrication. In: Proceedings of the 1991 winter simulation conference, pp 448–454

Chan WK, Yi J, Ding S (2007) On the optimality of one-unit cycle scheduling of multi-cluster tools with single-blade robots. In: Proceedings of the 2007 IEEE international conference on automation science and engineering, CASE 2007, pp 392–397

Chien C-F, Dauzère-Pérès S, Ehm H, Fowler JW, Jiang Z, Krishnaswamy S, Lee T-E, Mönch L, Uzsoy R (2011) Modelling and analysis of semiconductor manufacturing in a shrinking world: challenges and successes. Eur J Indus Eng 5(3):254–271

Dayhoff JE, Atherton RW (1984) Simulation of VLSI manufacturing areas. VLSI Design. 85

Dayhoff JE, Atherton RW (1986) Signature analysis: simulation of inventory, cycle time and throughput tradeoffs in wafer fabrication. IEEE Trans Compon Hybrids Manuf Technol 9:518–525

Fowler JW, Fu MC, Schruben LW, Brown S, Chance F, Cunningham S, Hilton C, Janakiram M, Stafford R, Hutchby J (1998) Operational modeling & simulation in semiconductor manufacturing. In: Proceedings of the 1998 winter simulation conference, pp 1035–1040

Glassey CR, Petrakian RG (1989) The use of bottleneck starvation avoidance with queue predictions in shop floor control. In: Proceedings of the 1989 winter simulation conference, pp 908–917

Hood SJ, Amamoto AFB, Vandenberge AT (1989) A modular structure for a highly detailed model of semiconductor manufacturing. In: Proceedings of the 1989 winter simulation conference, pp 811–817

Hood SJ, Welch PD (1990) The application of experimental design to the analysis of semiconductor manufacturing lines. In: Proceedings of the 1990 winter simulation conference, pp 303–309.

Jain S, Chan S (1997) Experiences with backward simulation based approach for lot release planning. In: Proceedings of the 1997 winter simulation conference, pp 773–780

Johri PK (1989) Dispatching in an integrated circuit wafer fabrication line. In: Proceedings of the 1989 winter simulation conference, pp 918–921

LeBaron HT, Pool M (1994) The simulation of cluster tools: a new semiconductor manufacturing technology. In: Proceedings of the 1994 winter simulation conference, pp 907–912
Mauer JL, Schelasin RE (1993) The simulation of integrated tool performance in semiconductor manufacturing. In: Proceedings of the 1993 winter simulation conference, pp 814–818
Miller DJ (1989) Implementing the results of a manufacturing simulation in a semiconductor line. In: Proceedings of the 1989 winter simulation conference, pp 922–929
Mönch L, Fowler JW, Mason SJ (2013) Production planning and control for semiconductor wafer fabrication facilities: modeling, analysis, and systems. Springer
Nadoli G, Rangaswami M (1993) An integrated modeling methodology for material handling systems design. In: Proceedings of the 1993 winter simulation conference, pp 785–789
Pierce NG, Stafford R (1994) Modeling and simulation of material handling for semiconductor wafer fabrication. In: Proceedings of the 1994 winter simulation conference, pp 900–906
Pitts KA (1988) Discrete-event simulation of wafer fabrication facility. In: Proceedings of the 1988 winter simulation conference, pp 712–718
Prasad K, Rangaswami M (1988) Analysis of different AGV control systems in an integrated IC manufacturing facility, using computer simulation. In: Proceedings of the 1988 winter simulation conference, pp 568–574
Schömig AK, Mittler M (1995) Autocorrelation of cycle times in semiconductor manufacturing systems. In: Proceedings of the 1995 winter simulation conference, pp 865–872
Spier J, Kempf K (1995) Simulation of emergent behavior in manufacturing systems. In: Proceedings of the advanced semiconductor manufacturing conference and workshop, pp 90–94
Uzsoy R, Lee CY, Martin-Vega LA (1992) A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning. IIE Trans 24(4):47–60
Waller PM (1986) December. The manufacturing modeling process: from conception to shop floor application. In: Proceedings of the 1986 winter simulation conference, pp 559–561

Chapter 15
Social and Behavioral Simulation

Charles M. Macal and Chaitanya Kaligotla

Abstract Social and behavioral simulation is an emerging field at the intersection of computational social science and simulation modeling and analysis. Modeling how individual and heterogeneous agents "behave" by converting sensory inputs to decisions, emotions, or actions, is the essence of behavioral simulation. The essence of social simulation, analogously, is modeling how these agents "interact" with each other, and behave collectively as a group, as a function of their behavioral imperatives. When techniques from these two fields are combined, social and behavioral simulation connects individual behaviors at the micro-level to system-level behaviors at the macro-level, allowing the study of dynamic social behavior. This kind of modeling can lead to new insights into the causal mechanisms that underlie social systems. Until now, social and behavioral simulation has consisted largely of innovative applications of simulation to illustrate social or group behavior. More recently, the ability of agent-based modeling, system dynamics, network analysis, and associated techniques to study these micro-macro interactions is unparalleled. The field is ripe for methodological and theoretical advancements. This chapter describes the history of social and behavioral simulation from the perspective of the Winter Simulation Conference community, provides a thematic overview of the questions being addressed, and discusses possible future directions.

15.1 Social and Behavioral Simulation: An Introduction

Social and Behavioral simulation sits in the intersection of social sciences and computational sciences, and is focused on the dynamic modeling and simulation of human behavior and social interactions, often in complex systems. Simulation helps us understand and plan for emergent system behavior of self-organizing patterns and order, a distinct feature of complex systems.

C.M. Macal (✉) · C. Kaligotla
Argonne National Laboratory, System Science Centre,
9700 S Cass Ave, Argonne, IL 60439, USA
e-mail: macal@anl.gov

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations,
Methods and Applications, DOI 10.1007/978-3-319-64182-9_15

Social and Behavioral Simulation has been a part of the Winter Simulation Conference from the very beginning. This chapter briefly discusses the history and evolution of the diverse set of work related to Social and Behavioral Simulation (hereafter denoted by *SBS*), from the perspective of the Winter Simulation Conference community. In Sects. 15.1.1 through 15.1.4, we elucidate the motivation and relevance for *SBS* within the broader field of Modeling and Simulation. The origins and evolution of *SBS* techniques are briefly discussed in Sect. 15.2, along with a focus on work presented at the Winter Simulation Conference (*WSC*) over the past decades. In Sect. 15.3, we discuss the research and application frontiers for *SBS* and motivate its continued significance and development within the WSC community and beyond.

15.1.1 Understanding Human and Social Behavior: Empirical, Analytical, Modeling & Simulations

The drive to understand human behavior is fundamental, and almost as old as recorded history. From the earliest underpinnings in mythology and early work in philosophy to modern academic fields wholly dedicated to understanding human behavior, we as a species have always endeavored to better understand ourselves, and society at large. The story of present day *SBS* derives from this need to understand collective human behavior in large groups (hence the word "social" behavior), marked by interactions with one another and with the world around us.

Theories of behavior were first conceived from empirical observations, driven in large part by fields across the social sciences and psychology. In recent decades, however, modeling and simulation of human and social behavior has become recognized as an important area of research and practice in other fields beyond the social sciences. The maturing of fields like behavioral sciences, psychology, economics (especially game theory and utility theory), and social science, along with new methods in the computational sciences and in modeling and simulation, has led to the development of more integrated behavioral models. Early work on modeling behavior is well detailed in Guetzkow et al. (1972). Refer to Epstein and Axtell (1996) and Epstein (2006) for an overview of modeling approaches.

The underlying motivation of better understanding human and social behavior has led to recent work leveraging a mixture of methodologies—empirical, analytical, and modeling and simulation—to yield greater insights. Axtell (2000) also argues how *SBS* techniques leverage other methods to model and understand a broad set of problems. Ultimately, a key in the past and future of *SBS* is the use of various tools, techniques and methods to gain new knowledge on fundamental human and social behavior. Figure 15.1 represents the cycle of knowledge relating to the field of human and social behavior in general. We have roughly followed the path from seeking to understanding human and social behavior from empirical observations to modeling human behavior as mathematical and computational

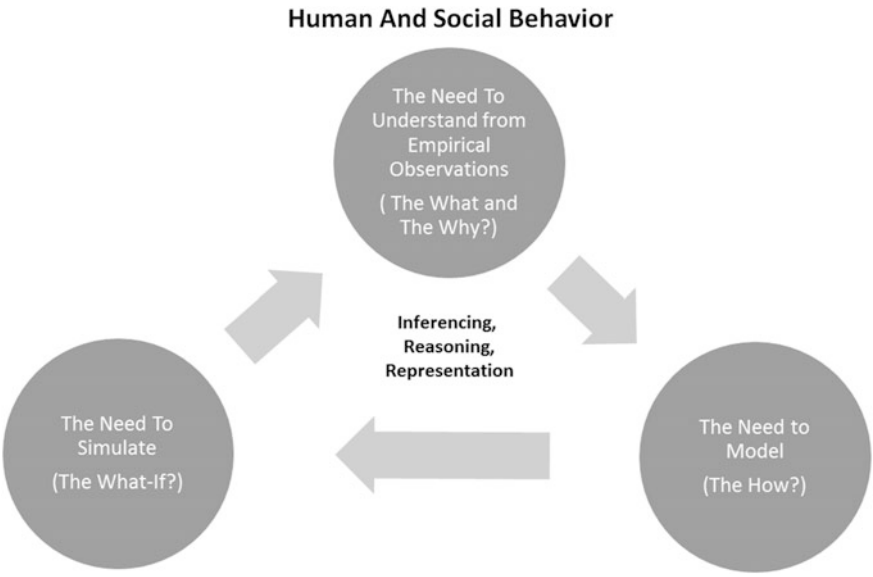


Fig. 15.1 The cycle of knowledge. The need to understand human and social behavior typically begins from empirical observations, leading to the modeling of such behaviors to, representing the processes and predicting outcomes, which then leads to the need to simulate human and social behavior to increase knowledge and understanding, which in turn leads to new questions and observations

representations, to simulating such behaviors and gaining insights leading back to defining new empirical observations that should be made.

15.1.2 Caveats in Modeling Human and Social Behavior

Some models of human behavior in fields like operations research and operations management are characterized by normative, rational, and predictive models of people or agents (Tolk et al. 2015). Other fields like behavioral economics develop descriptive behavioral models characterized by dynamic actors who are boundedly rational (Simon 1982). Each representative model has its strengths and weaknesses. The former model, for instance, is often used in the context of optimization, i.e., what is the best strategy for a rational actor to adopt, while the latter model is typically applied in the context of predicting behavior, i.e., what a non-perfectly-rational actor is expected to actually do.

Modeling methodology has two distinct approaches to behavior and decision making. One is a prescriptive view of behavior, focusing on how decisions ought to be made, which assumes normative views of rational behavior (traditionally considered perfect rationality). Comparatively, a descriptive approach to modeling

behavior focuses on understanding how decisions are actually made, and does not make strong rationality assumptions. For a comprehensive review of social-behavioral models, across analytical, empirical, and descriptive techniques, refer to Castellano et al. (2009), Epstein and Axtell (1996), Axtell (2000) and Epstein (2006).

One reason for the widespread popularity of *SBS* methods is the need for, and the growing ability to describe, truer representations of human behavior in a wide variety of modeling contexts.

15.1.3 SBS: Features and Relevance

Social and Behavioral simulation is used for problems across a range of domains motivated by the need to understand the nexus between human behavior, social interactions, and real-world systems, to manage and predict outcomes of interest. It is prudent to note that the relevance of using modeling and simulation methodologies depends (mostly) on contexts where empirical, experimental or analytical methods alone might not be adequate.

Epstein (2006, 2008) and Axtell (2000), list a number of reasons beyond prediction, for the relevance of *SBS*. Axtell (2000) presents three distinct uses of *SBS* techniques: (a) a novel method or application of traditional Operations Research simulation, (b) mathematically non-tractable or non-solvable models, and (c) classes of problems with clear non-quantitative models.

While *SBS* has consisted largely of innovative applications of modeling and simulation to illustrate social or group behavior, the ability of agent-based, system dynamics, and more recently hybrid modeling (that combine two or more of these techniques in a single model) techniques to study micro-macro interactions is becoming a popular approach to modeling complex systems that contain decision making agents, whether human or artificial.

The representation of how individual and heterogeneous agents "behave," by modeling how sensory inputs are transformed into decisions, emotions, or actions, is the essence of behavioral simulation. The essence of social simulation, analogously, is the modeling of how these agents "interact" with each other collectively, as a function of their behavioral imperatives. *SBS* connects individual behaviors at the micro-level to system-level behaviors at the macro-level, allowing the study of dynamic social behavior, as well as collective behavior, leading to potential new insights into the causal mechanisms that underlie social systems.

Extant literature (Macal 2016; Kunc 2016; Brailsford 2014; Epstein 2006; Martinez-Moyano and Macal 2016; Axtell 2000; Epstein and Axtell 1996) suggests the following fundamental features of *SBS*:

i. Simulating true representative models of human and social behavior, especially heterogeneous agents and interactions.
ii. Ability to describe individual behavior dynamically, in a wide variety of contexts.

iii. Moving beyond normative behavioral decision theory to more descriptive and causative behavioral models.
iv. The recognition of emergent system behavior in social and behavioral systems.

15.1.4 A Note on Emergence

Emergence can be described as a property of a complex system, arising from, but distinct to, the individual characteristics and interactions of entities in the system. This has been widely observed, and some famous examples of such a system property (in context of individual and social behavior) is that of the "flocking boids" model (Reynolds 1987) or the "game of life" model (Gardner 1970).

Conceptually, emergence refers to the seemingly spontaneous appearance of patterns, structure, or more generally, of order, that we observe in complex systems, including social and behavioral simulations. This emergent order is often described as surprising, since it is something that we have not explicitly programmed into our models and often do not expect to see. In the case of an agent-based model, the agents often appear to self-organize in the agent state space into higher-level entities that act in unison. Why should this happen? According to the Second Law of Thermodynamics, entropy (disorder) always increases in a closed system without a supply of energy. Don't we have to pay extra for this organization?

In the Boids model, for example, we might expect that the Boids agents would randomly distribute themselves over the spatial landscape, perhaps moving and spinning, but with little apparent coherence. In fact, that is what the Boids do for many settings of the model parameters. The Boids model has three parameters, one per rule: attraction, repulsion, and alignment. However, for many other combinations of the parameters, what we see are groups of Boids forming and traveling together. This is reminiscent of flocks, or schools, or colonies of animals in the natural world exhibiting similar behaviors and collective patterns. Sometimes, multiple groups form, and the groups interact in interesting ways. The Boids model, as abstract as it is, is a good example of the social behaviors operating in the social-behavioral state space.

What we do not see in the Boids model is the emergence of group behaviors or norms that come out of the group. There is no awareness on the part of the Boids agents that they are organized; the emergent "organization" that we observe has no awareness of itself, and no ability to create new group behaviors, affordances, or constraints for the individuals in the group to adhere. This is reminiscent of an ant colony, a set of mindless individuals acting on their own set of behavioral rules that have been passed down and perfected for survival through an evolutionary process. Although they are comprised of mindless individuals, we can still think of the colony as having a collective behavior; for example, the behavior that causes a

colony of Army ants to uproot itself and forage for miles through a rain forest looking for their next home.

Humans, on the other hand, are capable of self-organization and creating organizations that themselves make new rules of behavior that come back to act as affordances or constraints on the individuals of the group. This process is called downward causation in the literature (Gilbert 2002; Salgado and Gilbert 2013; Sawyer 2000), and it closes the loop between the individuals creating group behavior and that behavior then acting upon the individuals. Currently, this downward causation process cannot be modeled in social and behavioral simulation, but would be a significant advancement in the field if methods could be developed for modeling this kind of process.

For all its strengths, modeling, by itself, is imperfect, as are qualitative and empirical descriptions of behavior. Simulation bridges an important gap—especially in its ability to reproduce emergent behaviors seen in large systems that involve human interaction. This is arguably the most important reason d'être for *SBS*, the relevance of which can be seen in the broad range of applications described in the following section.

15.2 Social and Behavioral Simulation: Methods and Applications

SBS has been a part of the Winter Simulation Conference from the very beginning —WSC archives revealed a session on Simulation of Human Behavior in 1968 and Urban/Social Systems in 1969. Papers included: models of innovation diffusion using stochastic simulation methods (Carroll and Hanneman 1968); management decision games using discrete event simulation (DES) methods (Bubenko 1968); mail service systems for the US Post Office (Tuan and Nee 1969); health and social service programs using DES methods (Bremner and Eicker 1969); and electricity adoption in village communities (Pfaff and Jambotkar 1969).

This section provides a brief history of the methods and applications social and behavioral simulation, especially from the perspective of the WSC, and provides a thematic overview of the main application areas.

15.2.1 Evolution of Methods and Techniques

Goldsman et al. (2010) provide an overview of the history of simulation as a field, including the types of simulation, programming environments, and application domains. Most *SBS* methods have their origins from Cellular Automata (CA) concepts and Discrete Event Simulation (DES), progressing into Stochastic Simulation including Monte Carlo methods, System Dynamic Simulation models and eventually and recently into Agent-Based methods (ABMS) and Hybrid Simulation

methods. We first present a brief history of the classic literature in the field of *SBS* and then describe methodological work at the WSC specifically. In the following subsections, we present our view of *SBS* and its indefinitive history and one interpretation of how it came to be. We recognize that others may interpret the origins and developments of *SBS* differently.

15.2.1.1 The Classics:

The origins of *SBS* began from the groundwork laid by Von Neumann and Morgenstern (1944), followed by concepts of cellular automata in Neumann and Burks (1966). Conway (1970) arguably produced the first ever *SBS* called the "*game of life*", which was very popular (Gardner 1970). Schelling (1971) created a *SBS* similar model to explain segregation as a system property, irrespective of individual intent. Sakoda (1971) is credited with one of the earlier computational *SBS* models of social interactions, showing how continuous social interactions leads to social structure. Another famous *SBS* model showing emergent social structure and behavior was Reynolds (1987). A generative and simple explanation for the emergence of social processes was described in (Axelrod 1984; Epstein and Axtell 1996). Axelrod and Hamilton (1981) in their "*Tit for Tat*" model used repeated prisoner dilemma games to show the dominance of dynamic behavior strategies by agents.

Figure 15.2 shows the timeline of the classical papers in *SBS* as a field, and shows a sampling of papers from the WSC, mapping out some key methodological tutorials and papers.

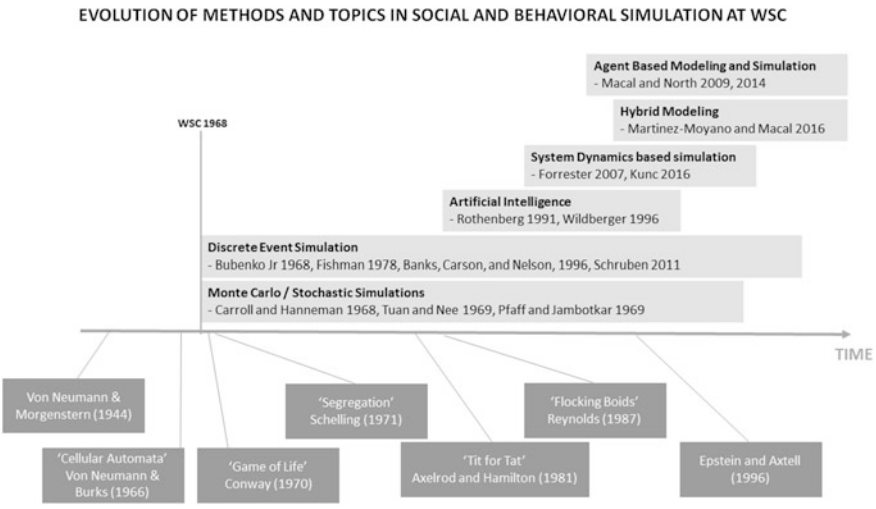


Fig. 15.2 Evolution of major techniques in *SBS*, including a sampling of WSC papers

15.2.1.2 Methods and Techniques at WSC:

Cellular Automata, Stochastic Simulation and AI

Wainwright (1974) is one of the earlier papers we could find in the WSC archives, who talks about Cellular Automata theory from Conway (1970), and describes how complex systems arise from simple rules. Wildberger (1996) provides a technical overview of genetic algorithms and cellular automata for *SBS* and argues for a combination of methods involving discrete event simulations with multiple agents to generate insights on emergent social behavior. To the best of our knowledge, the first tutorial on using Artificial Intelligence methods at the WSC is Rothenberg (1991), who noted that discrete event simulation techniques benefited from, and contributed to earlier work in AI, especially the refinement of the object-oriented paradigm and sensitivity analysis. He predicted that AI and simulation would come together to create value in two distinct ways—modeling motivation, which seeks to model cognitive processes of intelligence, and the engineering motivation, which seeks to solve problems beyond conventional intelligence.

System Dynamics

System Dynamics (SD) pioneered by Forrester (1961), is a longstanding and robust technique for modeling social and behavioral systems. Refer to Forrester (2007) for seminal work on the development of SD, it's methods and applications. At the WSC, Kunc (2016) provides a recent tutorial on SD methods to model behavior, and discusses challenges and best practices. SD methods have proven to be highly flexible, as common differential equation models are easily converted into difference equations, which are representative of SD models. These models have been successfully applied in epidemiology, innovation diffusion, and other application domains (Kunc 2016).

Discrete Event Simulation

DES owes its origins to the field of Operations Research in the 50s, and gained in popularity in the late 60s with the advent of computing hardware and software. Refer to Fishman (1978) and Banks, Carson, and Nelson, 1996 for a comprehensive overview. At the WSC, Bubenko (1968), presented one of the very first *SBS* papers using a DES model to highlight management decision making as a social process. Maria (1997) provided a tutorial on using DES, emphasizing the best practices in modeling, designing and analyzing a simulation to study real-world systems. More recently, Schruben (2011) introduced *Activity Interaction*, a new approach for modeling system dynamics, and *Self-Simulating Systems,* a new method for discrete event/hybrid modeling to generate real-time simulations. This has relevance to *SBS*

in terms of new hybrid techniques to solve some specific problems where the use of DES methods in combination with agent-based methods might be beneficial.

Agent-Based Modeling and Simulations

Agent-based modeling and simulation (denoted by ABMS) combines elements of game theory, Monte Carlo methods, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming. The use of ABMS is becoming increasing popular, due in large part to the flexibility and wide applicability in its use. See Macal (2016) and (Macal 2009) for a comprehensive overview of the literature and methods for modeling social and behavioral systems, and their applications.

At the WSC, an early tutorial was given by Macal and North (2005) describing the theoretical and practical foundations of agent-based methods, while highlighting the relationships between ABMS and other traditional modeling techniques. Macal (2010) lists similarities and in some cases, equivalences between the SD and ABMS approaches in the infectious disease domain. More recently, increasingly comprehensive tutorials followed at the WSC, in Macal and North (2009) and Macal and North (2014).

Hybrid Simulations

Another recent trend at the WSC has been to combine different modeling approaches to capture the complexity of real-world systems. These have been commonly referred to as Hybrid Modeling and Simulation methods. Brailsford et al. (2013) introduce the concepts of the method and provide a narrative description of the process of building a hybrid model. Brailsford (2014) compares simulation methodologies of DES, ABMS, and Hybrid Modeling, and argues that the latter should be a matter or last-resort due to the inherent complications of analysis and interpretations of results arising using hybrid modeling. Nonetheless, the complexity of real-world systems may necessitate using a combination of multiple modeling approaches. At the most recent WSC, Martinez-Moyano and Macal (2016) discuss the broader use, techniques, and applications for Hybrid Simulations.

15.2.2 Major Application Areas Across WSC Papers

The first two years of the WSC, 1968 and 1969, already proved the wide relevance of *SBS*, with applications to modeling innovation diffusion, to management decision making, to healthcare and social services, and other public services. Listing all the application domains of *SBS* is beyond the scope of this chapter, consider for

instance Table 1 from Macal (2016), which lists 18 distinct disciplines using ABMS methods. We could expect the same by extending the exercise to the wider field of *SBS*. Here, we highlight a sampling of major application domains for the *SBS* methods described in the previous section and list examples of some interesting work.

15.2.2.1 Healthcare and Health Services

SBS is useful for modeling human behavior both individually, and collectively, in the context of public and private health systems, which are complex and critically relevant across counties, cities, and countries. Noble et al. (2012), for example, use ABMS as a tool for generating scenarios to inform public policy making. Addressing the challenges represented by UK's aging population, they use *SBS* to model a concept they term "linked lives" which is a form of emergent social behavior, as a means to develop a synthetic population to test hypotheses and alternatives for policy making, by measuring endogenous outcomes of the average cost of state-funded care.

Fakhimi et al. (2014) provide a case study using a novel approach of hybrid ABMS-DES modeling for strategic planning and simulation analytics of the London ambulance service. Modeling the triple objectives of financial constraints, environmental impact, and service quality, they use data from the *SBS* model to analyze different sustainable planning strategies.

Closely related, Anagnostou et al. (2013) study a representative hybrid ABMS-DES model to simulate emergency medical services in London. Seeking to demonstrate the potential of distributed hybrid simulation, they report the relevance and promise of *SBS* as a "what-if" analysis tool. All these papers highlight the relevance, value and use of *SBS*.

15.2.2.2 Disease Modeling

Brailsford et al. (2013) use a behavioral simulation model for a disease called age-related macular degeneration (AMD), a common cause of sight loss in people aged over 65. They use the hybrid model to explore the wider links between the health and social care systems in the UK. Wong, Goldsman, and Tsui (2015) use *SBS* to evaluate the effectiveness of various practical non-medical interventions in containing a potential pandemic outbreak, such as H1N1, in Hong Kong. They describe a detailed community interaction dynamics model, taking into account the system dynamics, the natural history of influenza, and social and behavioral dynamics, to model the impact of a combination of various school closure modes, triggers, types, and durations, on the course of the disease outbreak. Their simulation results suggest that the strategy of closing all types of schools generally outperforms that of closing only a subset, especially if the closure period is substantial.

These papers highlight the value of *SBS* to provide insights and sound metrics used to guide practical decision making at a public policy level.

15.2.2.3 Public Policy

The following two papers describe good examples of the applications of *SBS* methods and techniques in modeling policy issues and the insights and benefits to inform policy.

Padgham et al. (2015) use social simulation for community awareness. Following a Belief Desire Intention (BDI) approach to modeling cognitive agents using ABMS, they report that the use of an interactive simulation was more effective than print media for getting across key messages. Wilcox (2011) uses *SBS* as a method to formulate a theory and empirically test for the same, in the domain of neighborhood crime. He simulates a system of social relationships using ABMS to reflect applicable social theory and reports that the use of *SBS* does increase the precision of theoretical arguments, but cautions about the increased rigor necessary in elaborating model specifics and the data analysis required for validation and verification.

Allen and Davis (2010) propose an agent-based simulation model based on social impact theory, to investigate factors that influence student selection of STEM majors in U.S. colleges. Their computational experiments suggest policy insights and social benefits related to reaching students early, segregating STEM students, and making changes to the job market. The power of using *SBS* in informing policy is shown by the insights from their study, which includes that small environmental change could precipitate a dramatic shift in outcomes.

15.2.2.4 Social Influence, Opinions, Rumors, and Social Networks

A recent application domain for *SBS* is to study human and social behavior online, both individually and for the internet community as a whole. The study of social influence, opinion and rumor dynamics on social networks is in its infancy and has been featured in *SBS* tracks in the last two WSC meetings. A sampling of papers on the topic is discussed next.

Merlone et al. (2015) study social influence online, analyzing minority (group) influence on opinion dynamics. They provide a heterogeneous model in which minority opinion holders strategically choose their social behavior to exert influence. Simulating opinion dynamics, they show how opinion dynamics is affected by the presence of such a strategic minority.

Birdsey et al. (2015) analyze emergent behavior in Twitter and propose a definition of emergent behavior focused on the pervasiveness of a topic within a community. Specifically, they extend an existing stochastic model for user behavior, focusing on advocate–follower relationships and model the behavior of users on Twitter to predict when topics reach a certain stage of evolution.

Kaligotla et al. (2015) develop an ABMS framework to study agent behavior in the spread of competing rumors through measured networked interactions, whereby agents update their beliefs with respect to a rumor, and attempt to influence peers through interactions, uniquely shaping group behavior in the spread of rumors. Kaligotla et al. (2016) continue to develop the model to focus on broadcasting opinions (one-to-many interactions) alongside narrowcasts (one-to-one interactions) in social media conversations, explicitly considering behavioral characteristics of agents and the properties of the underlying network. Their generalized model for the spread of influence on social media discussion sites, and simulation experiments indicates that increased broadcasting (in terms of frequency, depth, and a number of broadcasters) increases homogeneity in an evolving scale-free network.

While we have only provided a sampling of papers from the WSC, an interesting observation to note is the diverse range of applications of *SBS* and the evolution of methods, from the very beginning of the WSC to today. The Winter Simulation Conference has been, and will continue to be, an ideal home for this field of work given the confluence of academic researchers, practitioners, and software vendors, all dedicated to fundamentally understanding and solving real-world problems involving human and social systems through simulation methods.

15.3 Next for Social and Behavioral Simulation: Frontiers and Opportunities

A 2015 WSC panel discussion (Tolk et al. 2015) on the need for a national research agenda for modeling and simulation included a section on *SBS*. Common questions for further generalizations and domain-independent disciplines were on the topics of validation and verification, reusability and composability, credibility and trustworthiness, complexity and uncertainty, and going beyond domain-specificity.

In this section, we focus on *SBS* in particular, highlighting the challenges and frontiers, while motivating its continued significance and development within the WSC community.

15.3.1 Challenges in SBS: Validation and Verification

Bharathy and Silverman (2010) present a well-recognized and experienced argument that validating social systems is not a trivial task. They propose assessing the validity under four broad dimensions of methodological validity—internal validity, external validity and qualitative, causal and narrative validity. Hofmann (2015), however, reviews and reassesses technical and philosophical arguments on the limits of empirical validation with respect to social simulation. He proffers an argument that a single experimental frame cannot guarantee the objective "validity"

of simulation results since the data used for validation, the experimental frame, and the simulation model itself are interdependent. His recommendation is to put the focus on model transparency with respect to assumptions, model, and data, while empirical validation on the macro-level should be the focus when comparisons with similar models are possible.

All of the methods and approaches to the validation of simulation models, in general, apply to social and behavioral systems modeling (Sargent 2013). Social systems modeling, and agent-based modeling in particular, has additional requirements for validation. These include validating the agent behavioral models and validating the emergent phenomenon that arise in the models. Good examples of validated social systems models are needed and lessons learned from the process will help establish a standard practice. For example, Bert et al. (2014) provide lessons from validating a model of the Argentinean agricultural system.

15.3.2 Methodological and Research Frontiers

Schruben (2011) compares two fundamental historical approaches to developing new simulation modeling methodology: academic and commercial, noting that while the twain (academics and software vendors) do not meet, there is a need for academics to work together with practitioners, beyond fundamental academic research, to include actual implementations.

There are several research areas that show great promise for advancing the capabilities of social and behavioral systems modeling to solve real-world problems of great significance. A few of these research areas are described next.

15.3.2.1 Behavioral Modeling

The behavioral modeling challenge for social and behavioral systems modeling is to develop better representations of agent behavior and the methods that populate behavioral models with the requisite data. One of the primary reasons that people are interested in social systems modeling is because they would like to include truer representations of human behavior into their models and understand how the diversity of behaviors plays itself out over the modeled population. Advancements in behavioral economics and behavioral operations management have fueled interest in better models of behavior. Causative agent behavioral models, based on insights from behavioral economics (Kahneman 2011) and cognitive sciences (Sun 2006) that include social and emotional factors could be essential elements of more predictive, and useful, social and behavioral models. For example, Epstein (2014) introduces a new theoretical entity, Agent_Zero, a conceptual software individual endowed with distinct emotional or affective, cognitive or deliberative, and social aspects, grounded in contemporary neuroscience that represents explicit causative factors underlying agent behaviors.

15.3.2.2 Simulation Analytics

The simulation analytics challenge for social and behavioral systems modeling is to develop the methods and tools, such as data analytics and statistical analysis techniques, for extracting meaningful information from simulation results. Social systems modeling's gain in complexity is at the loss of analytical tractability and the ability to derive facts a priori about agent-based models, such as relating micro-level agent behaviors to macro-level system outcomes. Instead, agent-based models must be simulated on a computer. Computational experiments must be cleverly designed in advance to efficiently obtain the data that can be used to understand model behaviors, sensitivities to parameters, and how uncertainties in input data and structural relationships (e.g., agent behaviors) are propagated to model outputs. Data analytics approaches turned to simulation output data such as simulation analytics (Nelson 2016) will be needed. In addition, new statistical methods and existing methods implemented for large-scale application of data analytics are needed to support social and behavioral modeling requirements (ten Broeke et al. 2016; Thiele et al. 2014).

15.3.2.3 Hybrid Modeling

The hybrid modeling challenge is to understand how agent-based modeling can be effectively used with other simulation and modeling techniques operating together in the same "hybrid" model in such a way that each technique addresses the part of the problem that it does best. Agent-based models need not be viewed as displacing other simulation techniques (Siebers et al. 2010). The ABMS/SD and ABM/DES combinations are the agent-related hybrid configurations that have received the most attention (Heath et al. 2011). The hybrid modeling challenge has both logical (how to link models together in a way that makes sense) and mechanistic elements (how to link two existing models together that use disparate modeling tools). The challenge is to adopt these approaches as standard practice throughout the simulation community.

15.3.2.4 Large-Scale Agent-Based Modeling and Simulation

The large-scale agent-based modeling challenge is to efficiently and effectively simulate large-scale agent-based models, consisting of millions of agents, at the city scale, or even billions of agents, at the global scale. The computing challenge is to develop algorithms and software for distributing agent-based models, or their interacting components, on high-performance computing, cloud computing, and other platforms (Collier et al. 2015). Research challenges include how to dynamically balance simulation workloads, interact with running simulations, and efficiently collect model outputs for further analysis. A large-scale social systems modeling challenge is to engineer a process for efficiently developing synthetic

populations of agents, whether agents represent actual people, which comes with the associated data access and privacy issues, or only surrogate agents that correspond to the population, but only in the aggregate, to properly address anonymity concerns.

15.3.3 Conclusion: Looking Forward for SBS at the WSC

Social systems modeling, with the need to include people, their behaviors and actions, as part of the complex systems that we model, has had a role in WSC since its inception. Beginning with the normative, optimizing approaches to modeling behavior, and more recently with the social and behavioral sciences disciplines such as behavioral economics and behavioral operations management that focus on the descriptive, we are gaining many new insights on how people behave in a variety of contexts. While we have only provided a sampling of papers from the WSC, an interesting observation to note is the diverse range of applications of *SBS* and the evolution of methods, from the very beginning of the WSC to today.

This is fertile ground for developing new models and methods that provide new insights into how the complex systems, in which we are all a part of, behave. In fact, social and behavior simulation is getting increasing recent attention in several new domains. One example as in the military domain (Bharathy et al. 2012; Shults et al. 2017) that demonstrates that *SBS* is making a difference in real-world analysis and applications.

By providing a forum among researchers and practitioners for new ideas and approaches founded on rigorous simulation principles, WSC will continue to play an important role in promoting and advancing the field of social and behavioral modeling. The Winter Simulation Conference has been, and will continue to be, an ideal home for this field of work given the confluence of academic researchers, practitioners, and software vendors, all dedicated to fundamentally understanding and solving real-world problems involving human and social systems through simulation methods.

References

Allen TT, Nixon D (2010) A simple agent-based social impact theory model of student STEM selection. In: Proceedings of the winter simulation conference, pp 278–289

Anagnostou A, Athar N, Simon JET (2013) Distributed hybrid agent-based discrete event emergency medical services simulation. In: Simulation conference (WSC), 2013, Winter, IEEE, pp 1625–1636

Axelrod R (1984) The evolution of cooperation

Axelrod R, Hamilton D (1981) The evolution of cooperation. Science 21(1):1390

Axtell Robert (2000) Why agents?: on the varied motivations for agent computing in the social sciences

Banks J, Carson, JS, Nelson BL. n.d. Discrete-event system simulation. 1996. Prentice-Hall
Bert FE, Rovere SL, Macal CM, North MJ, Podestá GP (2014) Lessons from a comprehensive validation of an agent based-model: the experience of the pampas model of argentinean agricultural systems. Ecological Model 273(February):284–298. doi:10.1016/j.ecolmodel.2013.11.024
Bharathy Gnana K, Barry S (2010) Validating agent based social systems models. In: Proceedings of the winter simulation conference, pp 441–453. Winter simulation conference. http://dl.acm.org/citation.cfm?id=2433559
Bharathy GK, Levent Y, Andreas T (2012) Agent directed simulation for combat modeling and distributed simulation. In: Engineering principles of combat modeling and distributed simulation, edited by Andreas Tolk, John Wiley & Sons, Inc. pp 669–713. doi:10.1002/9781118180310.ch27
Birdsey L, Claudia S, Yong MT (2015) Twitter knows: understanding the emergence of topics in social networks. In: Winter simulation conference (WSC). IEEE, pp 4009–4020
Brailsford, SC (2014) Modeling human behavior-an (Id) entity crisis? In: simulation conference (WSC), Winter. IEEE, pp 1539–1548
Brailsford SC, Joe V, Stuart R, Andrew C, Andrew JL (2013) Hybrid simulation for health and social care: the way forward, or more trouble than it's worth? In: simulation conference (WSC), Winter. IEEE, pp 258–269. http://ieeexplore.ieee.org/abstract/document/6721425/
Bremner BA, William FE (1969) HAWSIM simulation of social services. In: Proceedings of the third conference on applications of simulation, pp 396–401
Broeke GT, George van V, Arend L (2016) Which sensitivity analysis method should i use for my agent-based model? J Artif Soc Soc Simul 19(1). doi:10.18564/jasss.2857
Bubenko Jr, JA (1968) Simulating a 'management game' with programmed decisions. In: Proceedings of the second conference on applications of simulations, pp 222–229. Winter Simulation Conference
Carroll TW, Gerhard JH (1968) Two Models of innovation diffusion. In: Proceedings of the second conference on applications of simulations, pp 158–162. Winter Simulation Conference
Castellano C, Fortunato S, Loreto V (2009) Statistical physics of social dynamics. Rev Mod Phys 81(2):591
Collier NT, Ozik J, Macal CM (2015) Large-scale ABM with RepastHPC: a case-study in parallelizing a distributed ABM. In: PADABS 2015 Proceedings. Vienna, Austria
Conway J (1970) The game of life. Scient Am 223(4):4
Epstein JM (2006) Generative social science: studies in agent-based computational modeling. Princeton University Press
Epstein JM (2008) Why model? J Artif Soc Soc Simul 11(4):12
Epstein JM (2014) Agent_Zero: Toward neurocognitive foundations for generative social science. Princeton University Press
Epstein JM, Robert A (1996) Growing artificial societies: social science from the bottom up. Brookings Institution Press
Fakhimi M, Anastasia A, Lampros S, Simon JET (2014) A hybrid agent-based and discrete event simulation approach for sustainable strategic planning and simulation analytics. In: Simulation conference (WSC), 2014 Winter, pp 1573–1584. IEEE
Fishman, GS (1978) Principles of discrete event simulation.[book Review]
Forrester, J (1961) W. Industrial dynamics. Cambridge Mass: MITPress 1 (961):1–464
Forrester JW (2007) System dynamics—a personal view of the first fifty years. Syst Dyn Rev 23 (2–3):345–358
Gardner M (1970) Mathematical games: the fantastic combinations of john conway's new solitaire game 'life'. Sci Am 223(4):120–123
Gilbert N (2002) Varieties of emergence. In: Agent 2002 conference: social agents: ecology, exchange, and evolution, Chicago, 11–12. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.461.8097&rep=rep1&type=pdf
Goldsman D, Nance RE, Wilson JR (2010) A brief history of simulation revisited. In: Proceedings of the winter simulation conference, pp 567–574. Winter Simulation Conference

Guetzkow HS, Philip K, Randall LS (1972) Simulation in social and administrative science
Heath SK, Brailsford SC, Arnold B, Charles MM (2011) Cross-paradigm simulation modeling: challenges and successes. In: Simulation conference (WSC), proceedings of the 2011 Wi*nter*, pp 2783–2797. IEEE. http://ieeexplore.ieee.org/abstract/document/6147983/
Hofmann MA (2015) Limits of empirical validation: a review of arguments with respect to social simulation. In: Proceedings of the 2015 winter simulation conference, pp 4069–4080. IEEE Press. http://dl.acm.org/citation.cfm?id=2889158
Kahneman D (2011) Thinking, fast and slow. Macmillan
Kaligotla C, Yücesan E, Chick SE (2015) An agent based model of spread of competing rumors through online interactions on social media. In: Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 3985–3996
Kaligotla C, Enver Y, Stephen EC (2016) The impact of broadcasting on the spread of opinions in social media conversations. In: Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 3476–3487
Kunc M (2016) System dynamics: a behavioral modeling method. In: proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 53–64
Macal CM (2009) Agent based modeling and artificial life. In: Encyclopedia of complexity and systems science, pp 112–131. Springer
Macal CM (2010) To agent-based simulation from system dynamics. In: proceedings of the winter simulation conference, pp 371–382. Winter Simulation Conference
Macal CM (2016) Everything you need to know about agent-based modelling and simulation. J Simul 10(2): 144–156
Macal CM, Michael JN (2005) Tutorial on agent-based modeling and simulation. In Simulation conference, 2005 Proceedings of the Winter. IEEE, 14 p
Macal CM, Michael JN (2009) Agent-based modeling and simulation. In: winter simulation conference, pp 86–98. Winter simulation conference
Macal CM, Michael JN (2014) Introductory tutorial: agent-based modeling and simulation. In: Proceedings of the 2014 winter simulation conference, pp 6–20. Winter simulation conference
Maria A (1997) Introduction to modeling and simulation. In: Proceedings of the 29th conference on winter simulation. IEEE Computer Society, pp 7–13
Martinez-Moyano IJ, Macal CM (2016) A primer for hybrid modeling and simulation. In: Proceedings of the 2016 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 133–147
Merlone U, Radi, D, Romano A (2015) Minority influence in opinion spreading. In: Proceedings of the Winter Simulation Conference, Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 3997–4008
Nelson BL (2016) Some tactical problems in digital Simulation'for the next 10 years. J Simul 10 (1):2–11
Neumann, J von, Arthur WB (1966) Theory of self-reproducing automata. University of Illinois Press Champaign
Noble J, Eric S, Jakub B, Stuart R, Maria E, Seth B, Athina V, Jane F (2012) Linked Lives: the utility of an agent-based approach to modeling partnership and household formation in the context of social care. In: Proceedings of the Winter Simulation Conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 1046–1057
Padgham L, Dhirendra S, Fabio Z (2015) Social simulation for analysis, interaction, training and community awareness. In Proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 3130–3131
Pfaff M, Jambotkar CG (1969) An adaptive computer model of the adoption of electricity in a village community. In: Proceedings of the third conference on applications of simulation, pp 402–417. Winter Simulation Conference
Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. ACM SIGGRAPH Comput Graph 21(4):25–34

Rothenberg J (1991) Tutorial: artificial intelligence and simulation. In: proceedings of the 23rd conference on winter simulation, pp 218–222. IEEE Computer Society
Sakoda JM (1971) The checkerboard model of social interaction. J Math Sociol 1(1):119–132
Salgado M, Gilbert N (2013) Emergence and communication in computational sociology. J Theor Soc Behav 43(1):87–110
Sargent RG (2013) Verification and validation of simulation models. J Simul 7(1):12–24
Sawyer KR (2000) Simulating emergence and downward causation in small groups. In: Multi-agent-based simulation, pp 49–67. Springer
Schelling TC (1971) Dynamic models of segregation. J Math Sociol 1(2):143–186
Schruben, L (2011) Activity interaction and self-simulating systems. In: proceedings of the winter simulation conference, pp 2900–2908. Winter Simulation Conference
Shults FL, Lane JE, Wildman WJ, Diallo S, Lynch CJ, Gore R (2017) Modelling terror management theory: computer simulations of the impact of mortality salience on religiosity. Religion, Brain & Behavior, 1–24. doi:10.1080/2153599X.2016.1238846
Siebers PO, Macal CM, Garnett J, Buxton D, Pidd M (2010) Discrete-event simulation is dead, long live agent-based simulation! J Simul 4(3):204–210. doi:10.1057/jos.2010.14
Simon HA (1982) Models of bounded rationality: empirically grounded economic reason. vol. 3. MIT press
Sun R (2006) cognition and multi-agent interaction: from cognitive modeling to social simulation. Cambridge University Press
Thiele JC, Kurth W, Grimm V (2014) Facilitating parameter estimation and sensitivity analysis of agent-based models: a cookbook using NetLogo and R. J Artif Soc Soc Simul 17(3):11
Tolk A, Balci O, Combs CD, Fujimoto R, Macal CM, Nelson BL, Zimmerman P (2015) Do we need a national research agenda for modeling and simulation? In: proceedings of the 2015 winter simulation conference. Institute of Electrical and Electronic Engineers Inc, Piscataway, NJ, pp 2571–2585
Tuan PL, David SN (1969) MASS—a mail service simulation. In: proceedings of the third conference on applications of simulation, pp 382–395. Winter Simulation Conference
Von Neumann J, Morgenstern O (1944) Theory of games and economic behavior
Wainwright RT (1974) Life is universal! In Proceedings of the 7th conference on winter simulation-vol 2, pp 449–459. Winter Simulation Conference
Wilcox SP (2011) Agentizing the social science of crime. In: Simulation conference (WSC), proceedings of the 2011 Winter. IEEE, 333–344
Wildberger AM (1996) Introduction & overview of 'artificial Life'—evolving Intelligent agents for modeling & simulation. In: Proceedings of the 28th conference on winter simulation, pp 161–168. IEEE Computer Society
Wong ZS-Y, Goldsman D, Tsui K-L (2015) School closure strategies for the 2009 hong kong H1N1 influenza pandemic. In: Proceedings of the 2015 winter simulation conference. IEEE Press, pp 3961–3972

Chapter 16
Analysis of M&S Literature Published in the Proceeding of the Winter Simulation Conference from 1981 to 2016

Navonil Mustafee and Paul Fishwick

Abstract In the era of big data, data science and data analytics, it is logical to address the impact of the Winter Simulation Conference (WSC) by treating the papers (and metadata associated with publications) as raw data. This data then serves as a treasure trove from which to answer questions and pose new ones about modelling and simulation as a field to the extent that WSC, its associated authors, serves as a proxy for the field. Since the goal was to pick the years 1981 and 2016 as beginning and end values demarcating a year band, there were challenges. One challenge was that one data set ISI/CPCI-S begins in 1989 and so does not offer complete coverage. However, we address this by fusing this data set with Scopus, which begins in 1981. Unless there is a single authoritative source, some type of fusion process is necessary. We gathered the data, validated the data, merged or fused data where necessary and then delineated specific objectives. We then produced the results along with analysis. In any good data analysis, we find curious patterns in the data or other places where we should look further. In summary, formal data analysis gives us a 50,000 foot view of WSC but also points to new questions and pathways for further exploration in Modeling and Simulation (M&S).

16.1 Introduction

The Winter Simulation Conference is a leading international conference, which disseminates peer-review research and recent advances in the field of M&S. The conference is co-sponsored by M&S societies and standards bodies, such as, the

N. Mustafee (✉)
The Business School, University of Exeter, Streatham Court, Rennes Drive, Exeter EX4 4ST, UK
e-mail: n.mustafee@exeter.ac.uk

P. Fishwick
School of Arts, Technology, and Emerging Communication, The University of Texas at Dallas, 800 W. Campbell Road, ATC10, Richardson, TX 75080-3021, USA
e-mail: Paul.Fishwick@utdallas.edu

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9_16

American Statistical Association (ASA), *ACM Special Interest Group on Simulation and Modeling (ACM/SIGSIM)*, *IEEE Systems, Man, and Cybernetics Society (IEEE/SMC)*, *Institute for Operations Research and the Management Sciences—Simulation Society (INFORMS-SIM)*, *Institute of Industrial and Systems Engineers (IISE)*, *National Institute of Standards and Technology (NIST)* and *The Society for Modeling and Simulation International (SCS)*. The conference is governed by the Board of Directors, with the operational matters usually the purview of the annual WSC Committees. The conference is also supported by the WSC Foundation (WSCF), whose core purpose is to develop and manage funds that would ensure continuity of the conference. The quality of the conference has meant that attendance levels are high and there is representation from both academia and industry (including defense). As an example, the last seven editions of the conference (2010–2016) were attended by 600–700 delegates. The 2017th edition marks the remarkable 50th Anniversary of the WSC. To mark the golden jubilee, the authors present a profile of research published during the past 36 years of WSC.

The authors draw inspiration from their previous work, done to mark the 60th anniversary year of the *SCS*, where they prepared a profiling study of literature published in the Society's journal—*Simulation: The Transactions of the SCS* (Mustafee et al. 2012)—and presented a co-citation analysis for the same journal (Mustafee et al. 2014). Similar to their previous work, which they considered to be a *fitting tribute* to those "scientists and engineers, who had actively shaped and influenced the growth and development of SCS and continue to contribute to the theory, methodology, and applications of simulation science" (Yilmaz 2011), the book chapter acknowledges the contributions of the WSC authors and their affiliated institutions in the development of the field of M&S. Our literature analysis is also an art of introspection (Palvia et al. 2017); it facilitates, among others, the numerous simulation societies and sponsors of the conference, the *WSC Board of Directors* and the readers to reflect on the scholarly content that have been published over the years and how this has shaped our discipline. For those new to M&S, our chapter will not only provide an overview of the field but will inform readers of the leading authors and their seminal pieces of work (achieved through citation analysis). Previous profiling studies have mainly been done for journals, for example, Palvia et al. (2007) presented a profiling study of the journal *Information and Management (I&M); the* authors have conducted similar analysis for the *European Journal of Information Systems (EJIS)* (Dwivedi and Kuljis 2008), *Information Systems Frontiers* (Dwivedi et al. 2009), *Journal of the Operational Research Society* (Katsaliaki et al. 2010), and *Simulation: Transactions* (Mustafee et al. 2012). Further, there have been several studies that have compared between journals, for example *Management Information Systems Quarterly (MISQ)* and *I&M* (Claver et al. 2000), *MISQ* and *EJIS* (Mustafee 2011), and *I&M, EJIS* and *MISQ* (Palvia et al. 2017).

Having provided an overview of the conference and the purpose of this study, we now list the objectives which will define the variables for data collection and its subsequent analysis. Our objectives are, (a) to analyze authorship and identify the authors with the most number of publications in the period considered in this study,

(b) to determine the institutions and geographical locations associated with the majority of publications, (c) to identify the most-cited papers through citation analysis, (d) to identify the research areas under which the WSC papers are classified, and (e) to identify the top funding organizations. The findings of the study will thus present a ranking of the most productive authors, institutions, etc.; however, we would like to voice a note of caution to the readers with regard to interpreting this data. Such findings should be regarded as indicative only of WSC activity. This is because our conference-specific profiling exercise does not take into consideration several leading researchers, institutions and research papers because they have not been published in WSC (e.g. researchers may have been published in ACM SIGSIM PADS Conference, Spring Simulation Multi-Conference, etc.) and/or within the timeframe of the analysis.

The remainder of this paper is organized as follows. In the following section, we present the methodology that was employed to conduct the profiling exercise. Here, we describe the abstract and indexing databases that were used, the validation of the data set and the constitution of the final set to inform analysis. In Sect. 16.3 we present the results and discuss the findings. The chapter concludes with Sect. 16.4.

16.2 Methodology

We used the *ISI Web of Knowledge* database and selected the *Conference Proceedings Citation Index for Science (CPCI-S)*. The ISI/CPCI-S provides proceedings coverage for conferences in Science, and together with the equivalent index for Social Science & Humanities (CPCI-SSH), the CPCI provides coverage of over 148,000 conference proceedings from 1990 with more than 400,000 conference proceedings included in the index every year (Clarivate Analytics, n.d). The CPCI index starts from 1990 and therefore the "*1990 Winter Simulation Conference Proceedings*" is the start point of our analysis. The end point of our analysis is the "*2016 Winter Simulation Conference Proceedings*". Thus, we planned for our analysis to include a total of 27 years (1990–2016, both years inclusive).

We used the Basic Search option from the Web of Science™ Core Collection and selected "Publication Name" as the field to be searched. Although, it would have been possible for us to retrieve records through a combination of terms associated with the conference title and wild char characters, we decided to selected publication name from the index. This gave us a total of over 9300 records from 27 conference proceedings. However, the output thus attained showed, in some cases, the lack of structure/indexing, for example the output included 145 records attributed to the "*1989 WSC Proceedings*" (although the ISI/CPCI-S database starts from 1990), some proceedings names/volumes did not include the year and seemed to have been clubbed together (e.g. one publication had a generic name "*Winter Simulation Conference Proceedings"* and showed over 2300 associated records). Through the analysis of the record set (pertaining to publication name), it became obvious that there would be some duplication, however, our previous experience

has shown that ISI seldom includes duplicated records in the main search results. We therefore decided to include all the 27 proceedings with the "OR" operator for the search field—*Publication Name*. We conducted the search on ISI/CPCI-S; this resulted in a total of 6,985 records (as of May 2017). Comparing this with the initial record count of 9300 records, gives us 2315 records that appear to be duplicated.

16.2.1 Validation of the Data Set

We used the Scopus® citation database for the validation of our record set. Scopus, which is owned by Elsevier, is marketed as the largest abstract and citation database of peer-reviewed journals, books and conference proceedings. In terms of conference coverage, it includes close to 8 million conference papers from nearly 100,000 worldwide events, such as, conferences, coverage for society meetings, etc. (Elsevier n.d.). Whereas for ISI/CPCI-S, the dataset starts from 1989, the start date for Scopus archived records for WSC begins from 1981. We, therefore, considered extending our search to include both Scopus and ISI/CPCI-S (ISI Web of Knowledge), and which would give us a period of analysis from 1981 to 2016 (both years inclusive)—a total of 36 years! This would mean combining both the databases and then running our analysis. However, this does have a downside. With more databases being included, there are higher chances of errors and omission. In indexing databases, and because of the scale on which they operate, most of the records are automatically imported and curation of individual records is not often possible by the database provider. A further level of error may be introduced by us (the authors'), because of the task of merging records retrieved in different formats. We, therefore, decided on using data analysis features that are included in both ISI Web of Knowledge and Scopus.

In Table 16.1, we tabulate results retrieved from the ISI/CPCI-S and Scopus. The leftmost column lists the year that the conference was held. This may be different to the year the records were archived in the respective databases, and indeed, we identified a few errors in Scopus where the publication year was interchanged with the year the conference was held. For example, a basic Scopus search will reveal WSC publications for the year 2017; however, these are the *"Proceedings for the 2016 Winter Simulation Conference"*. The second and third column show the number of records retrieved from CPSI-S and its corresponding proportion in regard to total records. The fourth and fifth columns show the same data but for Scopus. As can be seen, the total number of records retrieved using CPSI-S and Scopus is 6985 and 7920 respectively. Scopus includes publications from 1981 to 2016, the only proceedings missing is from 1992. On the other hand, ISI/CPCI-S includes publications from 1989 onwards (and therefore papers from 1981 to 1988 are not available); further, the 1999 and the 2015 WSC proceedings are not included. We have also compared the differences in terms of the number of records reported by the two databases. For most years, this difference is not

Table 16.1 Records retrieved from ISI/CPCI-S web of knowledge and scopus

Publication years	ISI web of science/CPCI-S		Scopus		Difference in #Records (ISI-Scopus)
	Record count	% of total records	Record count	% of total records	
1981	N/A	N/A	94	1.19	(94)
1982	N/A	N/A	80	1.01	(80)
1983	N/A	N/A	100	1.26	(100)
1984	N/A	N/A	107	1.35	(107)
1985	N/A	N/A	97	1.22	(97)
1986	N/A	N/A	130	1.64	(130)
1987	N/A	N/A	136	1.72	(136)
1988	N/A	N/A	131	1.65	(131)
1989	145	2.08	146	1.84	(1)
1990	159	2.28	160	2.02	(1)
1991	162	2.32	162	2.05	0
1992	182	2.61	N/A	N/A	182
1993	206	2.95	206	2.60	0
1994	217	3.11	217	2.74	0
1995	215	3.08	213	2.69	2
1996	213	3.05	214	2.70	(1)
1997	201	2.88	201	2.54	0
1998	237	3.39	237	2.99	0
1999	N/A	N/A	244	3.08	(244)
2000	281	4.02	107	1.35	174
2001	223	3.19	159	2.01	64
2002	280	4.01	261	3.30	19
2003	264	3.78	132	1.67	132
2004	280	4.01	283	3.57	(3)
2005	341	4.88	343	4.33	(2)
2006	320	4.58	298	3.76	22
2007	291	4.17	297	3.75	(6)
2008	395	5.65	367	4.63	28
2009	296	4.24	316	3.99	(20)
2010	310	4.44	314	3.96	(4)
2011	386	5.53	390	4.92	(4)
2012	339	4.85	345	4.36	(6)
2013	347	4.97	348	4.39	(1)
2014	353	5.05	349	4.41	4
2015	N/A	N/A	395	4.99	(395)
2016	342	4.90	341	4.31	1
Total	**6985**	**100**	**7920**	**100**	**(935)**

significant, with the exception of year 2000 (174 additional records in ISI/CPCI-S), 2001 (64), 2002 (19), 2003 (132), 2006 (22) and 2008 (28). Considering all the adjustments detailed above, Scopus has an additional 900 records. However, these should not be mistaken to be unique records, the calculation of which is presented next.

16.2.2 Defining the Data Set for Analysis

Our previous discussion has shown that, in order to provide the maximum coverage for the papers published in the proceedings of the conference, we will need to consider both records retrieved from both ISI/CPCI-S and Scopus. For records that are missing from either of the two databases, we consider the one which has the information (obviously), however for the 25 years (1989–1991; 1993–1998; 2000–2014; 2017) for which records are present in both the databases, we select ISI/CPCI-S over Scopus. Our rationale for this is the significant number of additional records (including a total of 439 additional records in 2000–2003, 2006 and 2008) that are reported by ISI over Scopus (see Table 16.1). Our final dataset therefore includes:

- The timespan of analysis includes **36 years**, starting from **1981 until 2016**
- Years 1981–1988 (total of 8 years): Data set from Scopus
- Years 1989–1998 (10 years): ISI/CPCI-S
- Year 1999 (1 year): from Scopus
- Year 2000–2014 (15 years): ISI/CPCI-S
- Year 2015 (1 year): Scopus
- Year 2016 (1 year): ISI/CPCI-S
- In total we cover a total of **8499** records from both ISI/CPCI-S (26 years) and Scopus (10 years)

Figure 16.1 shows the total number of publications published in the proceedings from 1981 to 2016. The label for the year indicates whether the data was included from Scopus, in which case (S) is appended, or from ISI (only the year is presented). The primary axis on the left denotes the number of publications, with the secondary axis indicating the % of records. There is obviously a correlation between the total number of publications and % publications for each year. The figure illustrates that, the 1982 Proceedings of WSC had the least number of papers (80) which contributed to 0.94% of the total papers included in our analysis. On the other side of the scale, we have the 2008 and 2015 WSC Proceedings, each of which published a total of 395 papers each (4.65%). The average number of articles is therefore 236.

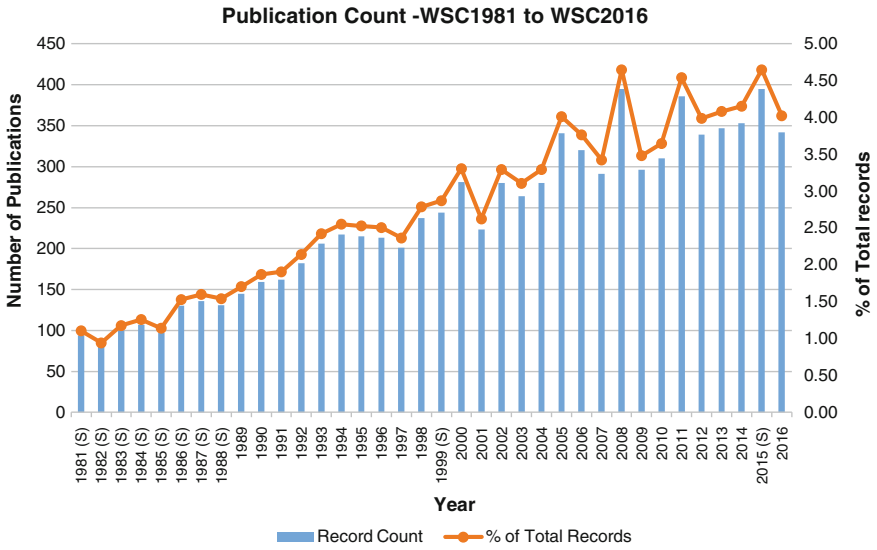


Fig. 16.1 No. of papers published in the proceedings each year and % contribution to total number of WSC articles (using both ISI/CPCI-S and scopus)

The trend in Fig. 16.1 is upward even if non-monotonic. This is good news for the conference as a whole, showing growth in terms of publications. Will this growth move to a steady state and cap at a certain level? Does the community want this? Can we correlate the minima and maxima points to other events or situations —for example, we might pose the dip in 2001 to be due to decreased travel. Expanding on this, WSC2001 was held in Arlington (VA) shortly after the September 2001 attacks. This had a substantial impact on the attendance and finances of the conference. Indeed, the origins of the WSC Foundation can be traced to this event (WSC Foundation 2017). What about the peaks in 2008 (Miami, FL), 2011 (Phoenix, AZ) and 2015 (Huntington Beach, CA)?

16.3 Findings

This section will present findings based on the analysis of, (a) authorship and editorships of WSC proceedings (Sect. 16.3.1); (b) authors' institutional affiliations (Sect. 16.3.2); (c) geographical location of the authors' institutional affiliations (Sect. 16.3.3) (d) WSC papers that have received the highest number of citations (Sect. 16.3.4); (e) analysis based on research areas (Sect. 16.3.5), and, finally, (f) sources of funding (Sect. 16.3.5).

16.3.1 Analysis Based on Authorship and Editors of Proceedings

Table 16.2 lists authors with 35 or more WSC publications within our timeframe of analysis. Authors could be sole authors, first authors or co-authors. We have used both ISI/CSCI-S and Scopus databases using the methodology defined in Sect. 16.2. We present database-specific counts (columns 2 and 3; columns 6 and 7) and the total authorship count (columns 4 and 8).

What are we to do with the author counts? Other than these counts indicating prolific writing within WSC, are there any counts due to co-authorship? If we were to plot the number of publications over time for one author, what would we find? We might also ask ourselves about the most prolific authors. For example, for Nelson and Wilson, how have their research topics (i.e. the topic of the paper) evolved or changed over time or are there one consistent topic for each?

WSC publishes high-quality conference proceedings. This is overseen by the Proceedings Co-editors, the Program Chair(s) and the General Chair. The co-editors ensure that the WSC template is being meticulously followed by the authors. Papers are electronically submitted in both PDF and word versions through the conference paper management system (Linklings). If the paper is accepted, the final version (i.e. the original paper with changes subsequent to peer-review) is uploaded through Linklings. The co-editors then use this source file to check conformity with the published template. If the paper requires minor formatting changes, then they may be actioned directly by the co-editors; however, for papers requiring changes that are non-trivial, the source file may be sent back to the authors with the request to make changes that conform to the conference guidelines. Considering the volume of papers published in WSC, the resultant editing effort and which conforms to the

Table 16.2 List of authors with 35 or more publications in WSC proceedings

Authors	ISI/CSCI-S count	Scopus count	Total	Authors	ISI/CSCI-S count	Scopus count	Total
Nelson, B.L.	72	14	86	Henderson, S.G.	38	4	42
Wilson, J.R.	63	21	84	Nicol, D.M.	35	7	42
Goldsman, D.	55	17	72	Kelton, W.D.	29	12	41
Law, A.M.	47	20	67	Balci, O.	32	8	40
Sargent, R.G.	40	20	60	Nance, R.E.	31	9	40
Rose, O.	53	4	57	Verbraeck, A.	39	0	39
Schriber, T.J.	40	14	54	Paul, R.J.	34	3	37
Glynn, P.W.	36	15	51	Fishwick, P.A.	33	3	36
L'ecuyer, P.	38	10	48	Chen, C.H.	32	4	36
Henriksen, J.O.	32	12	44	Uhrmacher, A.M.	35		35
Schmeiser, B.W.	31	13	44	Alexopoulos, C.	32	3	35
Takakuwa, S.	43		43	Yucesan, E.	30	5	35
Taylor, S.J.E.	42		42	Brunner, D.T.	30	5	35
Fu, M.C.	39	3	42				

Table 16.3 **WSC** proceedings editors and the volume of articles edited (over 275 articles only)

Editors	Edited	Editors	Edited	Editors	Edited
Himmelspach, J.	725	Morrice, D.J.	477	Yilmaz, L. Tolk, A. Ryzhov, I.O. Diallo, S.Y.	353 each
Jain, S.	696	Medeiros, D.J.	460	Steiger, N.M. Kuhl, M.E. Armstrong, F.B.	341 each
Joines, J.A.	622	Manivannan, M. S.	454	Pasupathy, R. Laroque, C.	339 each
Yucesan, E.	590	Goldsman, D.	397	Montoyatorres, J. Johansson, B. Hugan, J.	310 each
Smith, J.S. Peters, B.A.	503 each	Swain, J.J.	395	Fishwick, P.A. Barton, R.R.	281 each
Kang, K.	496	Creasey, R.	386	Snowdon J.L. Rossetti M.D. Ingalls R.G. Chen C.H.	280 each
Charnes, J.M.	493	Nelson, B.L.	363		

process described above, is substantial. This effort is usually shared by three or more WSC proceedings co-editors. WSC acknowledges the contribution of the editors by suggesting a referencing style for papers from WSC conferences, which should include the name of the editors. Table 16.3 presents the list of editors (it is also the convention to include the Program Chair(s) and the General Chair as co-editors) and the number of papers that were edited by the extended team. The number of edited articles could be a guide to ask the editors for their advice on editing process, or good practice in framing an article for WSC—what makes a good WSC article?

16.3.2 Analysis Based on Authors' Institutional Affiliations

Tables 16.4 and 16.5 present analysis based on the geographical location of the authors' institutional affiliations. In Table 16.4, we present the name of the institution, followed by country and the number of records retrieved from ISI/CSCI-S and Scopus respectively, followed by the total count. As can be seen from the table, of the 29 institutes with 50 or more publications, 22 institutions (approx. 75%) are from the US, two each from the UK and Singapore, and one each from The Netherlands, Germany and Canada. This list might aid future students, who are seeking graduate opportunities for M&S. The regional data is also noteworthy—would more non-US papers be present in WSC, if the conference were held more

Table 16.4 List of institutions with 50 or more publications in WSC proceedings

Institution name	Country	ISI/CSCI-S count	Scopus count	Total count
Georgia Institute of Technology	US	172	33	205
Virginia Polytechnic Institute and State University (VIRGINIA TECH)	US	104	26	130
University of Michigan	US	101	25	126
Purdue University	US	85	27	112
Brunel University London	UK	83	9	92
Cornell University	US	72	18	90
Pennsylvania State University	US	72	17	89
University of Central Florida	US	77	8	85
IBM Corp	US	74	11	85
MITRE CORP	US	68	16	84
US Navy	US	83	0	83
Northwestern University	US	70	10	80
Arizona State University	US	65	9	74
Nanyang Technological University	Singapore	61	12	73
National University of Singapore	Singapore	61	9	70
University of Virginia	US	60	10	70
University of Alberta	Canada	61	8	69
University of Illinois	US	62	6	68
Old Dominion University	US	54	11	65
University of Maryland	US	61	3	64
University of Arizona	US	47	16	63
University of Southampton	UK	55	7	62
University of Wisconsin	US	32	26	58
Columbia University	US	51	6	57
Virginia Polytechnic Institute and State University	US	30	26	56
North Carolina State University	US	44	11	55
Texas A&M University	US	32	22	54
Delft University of Technology	Netherlands	51	0	51
Technical University of Dresden	Germany	50	0	50

regularly outside of the US? Students looking to enter industry will use this data to ponder potential places to interview and work. For example, IBM and MITRE are highly active in the technical part of the conference.

Unlike Table 16.4 which presents data from both ISI/CSCI-S and Scopus, in Table 16.5 we use a different variable to categorize institutions in a higher level of granularity—this is referred to as Organizations-Enhanced in ISI/CSCI-S database. This allows us to search for preferred organizations names and/or their name variants (Thomson Reuters 2013). To take an example, University System of

Table 16.5 List of institutions (using ISI/CSCI-S—organizations enhanced) with 70 or more publications in WSC proceedings

Organizations enhanced	Records	% ef 6985	Organizations enhanced	Records	% ef 6985
University System of Georgia	226	3.24	Purdue University	90	1.29
United States Department of Defense	222	3.18	Pennsylvania Commonwealth System of Higher Education (PCSHE)	90	1.29
Georgia Institute of Technology	176	2.52	North Carolina State University	87	1.25
State University System of Florida	158	2.26	Naval Postgraduate School	85	1.22
International Business Machines (IBM)	130	1.86	Brunel University	83	1.19
Virginia Polytechnic Institute State University	105	1.50	University of Central Florida	77	1.10
United States Navy	105	1.50	Penn State University	75	1.07
University of Michigan	104	1.49	Cornell University	72	1.03
University of North Carolina	100	1.43	Northwestern University	70	1.00
Department of Energy	90	1.29			

Georgia (USG) includes research universities such as Augusta University, Georgia Institute of Technology, Georgia State University, University of Georgia and several other State Universities, Comprehensive Universities and State Colleges (USG 2017). It should therefore come as no surprise that the number of USG records identified in Table 16.5 is higher than records identified for Georgia Institute of Technology, since the latter is a sub-set of the former.

16.3.3 Analysis Based on Authors' Geographic Location

Table 16.6 creates geographically related questions similar to Table 16.4. The countries in the lead are the USA, Germany and the UK. Why? The venue for the conference might be one clue, but are there other reasons? What strategies can the WSC Board apply to diversify these numbers? Perhaps we need better social networking for M&S students and researchers?

Table 16.6 Geographical location of authors' affiliation (data from both ISI/CSCI and Scopus; total number of records retrieved which has the county field is 7499)

Countries	Records	% of 7499	Countries	Records	% of 7499
USA	4420	58.94	Sweden	101	1.35
Germany	487	6.49	South Korea	89	1.19
United Kingdom	325	4.33	Spain	74	0.99
Canada	322	4.29	Italy	71	0.95
Singapore	181	2.41	Taiwan	60	0.80
Netherlands	178	2.37	Austria	59	0.79
China	160	2.13	Australia	58	0.77
France	138	1.84	Ireland	50	0.67
Brazil	115	1.53	India	42	0.56
Japan	112	1.49	Turkey	38	0.51

16.3.4 Citation Analysis

Citation analysis is presented by merging data retrieved from ISI/CSCI-S and Scopus (refer to the section on methodology). These records pertain to the total number of WSC papers published/per year and the total citations received/year. It is to be noted that the number of citations will vary from one database to the other. For example, recent studies have compared databases to illustrate that indexing databases possess some shortcomings which may affect the quality and the precision of citation data (Clarke 2008a, b; Jacso 2005). For example, Jacso (2005) found that *Google Scholar* records citations from all sources including conferences, book chapters, working papers and other non-traditional sources, which may affect the quality of citation data. In the case of both ISI/CSCI-S and Scopus, citations are recorded only if the citing articles have also been indexed by the database. What does this mean? A database which is relatively selective in the inclusion of scholarly artefacts for indexing (for example, *ISI Web of Knowledge* only indexes journals with an associated impact factor) will typically report lesser number of citations, when compared to databases with a more comprehensive listing, for example, Google Scholar and Scopus.

As shown in Fig. 16.2, generally, the citations are increasing in number over time, suggesting several things: Are WSC authors increasing the sizes of their reference sections? Are the WSC proceedings being cited from outside of WSC papers? This result is highly favourable for WSC in terms of impact over time. More people are reading WSC papers. Hopefully, the answer is not that we are just citing more of ourselves although some of that is beneficial to demonstrate our research being built upon our peers.

Table 16.7 presents a list of articles with over 50 citations. A study of these papers may enlighten us as to what areas or topics are of most interest to others. From some of the papers, we may logically conclude that papers of a tutorial,

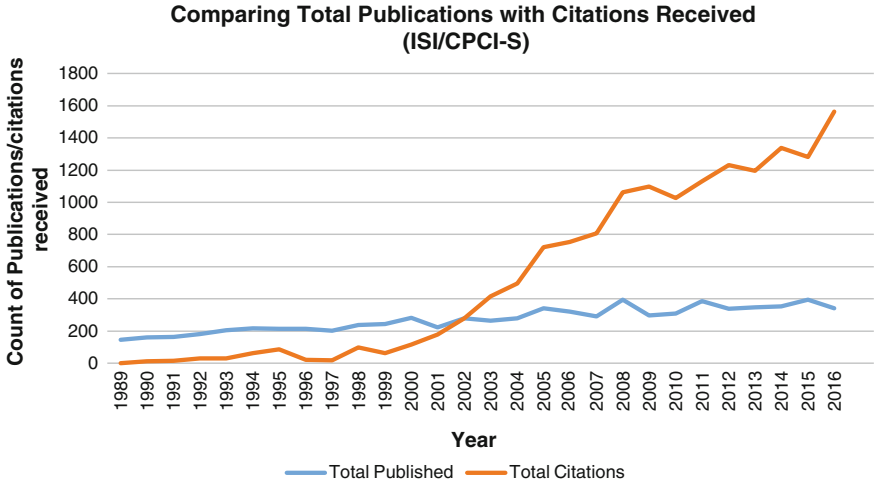


Fig. 16.2 The Total Number of Articles Published from 1989 = 2016 (except for 1999 and 2015) and the total number of citations received from these articles (Using data from ISI/CPCI-S)

survey or introductory nature are more generally read and so cited. Consider the two high-ranking papers by papers by *Sargent* and *Macal* as random examples. It is interesting to note that 9 of the 25 papers listed in Table 16.7 were published in the proceedings of the 1999 Winter Simulation Conference; this includes four of the top seven highly cited papers. What could be the reason for this? One possible explanation is that the records for WSC 1999 were extracted from Scopus—see Fig. 16.1; note the label *1999(S)* in the x-axis. As mentioned earlier, Scopus provides a more comprehensive database, and it is arguable that it indexes relatively higher number of articles (when compared to ISI/CSCI-S), which have cited papers from the 1999 WSC proceedings.

Consider the *average number of citations per year* (column 3; Table 16.7)—a value obtained by dividing the total number of citations by the total number of years subsequent to its publication. This metric normalizes the relative advantage that is intrinsic with papers published in an earlier proceedings when compared with those published in subsequently editions (citations increase over time). The 2005 paper by *Sargent* on the verification and validation of simulation models has the highest number of average citations (approx. 10 citations per year), this is followed by the 2005 paper on simulation optimization by *Fu* et al. (9 citations/year), and the 2009 paper by *Macal* and *North* on agent-based M&S (8/year).

Table 16.7 Articles with more than 50 citations (as of May 2017)

Article	Total citations	Avg citations/Year
Hajjar, D., and Abourizk, S. (1999). Simphony: An Environment For Building Special Purpose Construction Simulation Tools. *1999 WSC*, pp. 998–1006.	127	6.68
Chang, X. (1999). Network Simulations With OPNET. *1999 WSC*, pp. 307–316.	124	6.53
Sargent, RG. (2005). Verification And Validation Of Simulation Models. *2005 WSC*, pp. 130–143.	123	9.46
Azadivar, F. (1999). Simulation Optimization Methodologies. *1999 WSC*, pp. 93–100.	119	6.26
Meketon, MS., and Schmeiser, B. (1984). Overlapping Batch Means: Something For Nothing? *1984 WSC*, pp. 227–230.	112	3.29
Fu, MC., Glover, FW., and April, J. (2005). Simulation Optimization: A Review, New Developments, And Applications. *2005 WSC*, pp. 83–95.	109	8.38
Sargent, RG. (1999). Validation And Verification Of Simulation Models. *1999 WSC*, pp. 39–48.	94	4.95
Glynn, PW. (1987). Likelihood Ratio Gradient Estimation: An Overview. *1987 WSC*, pp. 366–375.	89	2.87
Macal, CM, and North, MJ. (2005). Tutorial On Agent-Based Modeling And Simulation. *2005 WSC*, pp. 2–15.	85	6.54
Carson, Y., and Maria, A. (1997). Simulation Optimization: Methods And Applications. *1997 WSC*, pp. 118–126.	85	4.05
Rossetti, MD., Trzcinski, GF., And Syverud, SA. (1999). Emergency Department Simulation And Determination Of Optimal Attending Physician Staffing Schedules. *1999 WSC*, pp. 1532–1540.	79	4.16
Schwetman, H. (1986). CSIM: A C-Based, Process-Oriented Simulation Language. *1986 WSC*, pp. 387–396.	77	2.41
Kleijnen, JPC. (1999). Validation Of Models: Statistical Techniques And Data Availability. *1999 WSC*, pp. 647–654.	75	3.95
Swisher, JR., Hyden, PD., Jacobson, SH., and Schruben, LW. (2000). A Survey Of Simulation Optimization Techniques And Procedures. *2000 WSC*, pp. 119–128.	72	4
Macal, CM., and North, MJ. (2009). Agent-Based Modelling And Simulation. *2009 WSC*, pp. 86–98.	70	7.78
Macal, CM., and North, MJ.(2006). Tutorial On Agent-Based Modeling And Simulation Part 2: How To Model With Agents. *2006 WSC*, pp. 73–83.	70	5.83
Glover, F., Kelly, JP., and Laguna, M. (1999). New Advances For Wedding Optimization And Simulation. *1999 WSC*, pp. 255–260.	66	3.47
April, J., Glover, F., Kelly, JP., and Laguna, M. (2003). Practical Introduction To Simulation Optimization. *2003 WSC*, pp. 71–78.	65	4.33

(continued)

Table 16.7 (continued)

Article	Total citations	Avg citations/Year
Fujimoto, RM. (1999). Parallel And Distributed Simulation. *1999 WSC*, pp. 122–131.	65	3.42
Dahmann, JS., Fujimoto, RM., and Weatherly, RM. (1997). The Department Of Defense High Level Architecture. *1997 WSC*, pp. 142–149.	62	2.95
Angerhofer, BJ., and Angelides, MC. (2000). System Dynamics Modelling In Supply Chain Management: Research Review. *2000 WSC*, pp. 342–351.	61	3.39
Page, EH., and Opper, JM. (1999). Observations On The Complexity Of Composable Simulation. *1999 WSC*, pp. 553–560.	53	2.79
Olafsson, S, and Kim, J. (2002). Simulation Optimization. *2002 WSC*, pp. 79–84.	52	3.25
Balci, O. (1997). Verification, Validation And Accreditation Of Simulation Models. *1997 WSC*, pp. 135–141.	52	2.48
Chen, HC., Chen, CH., Dai, LY., and Yucesan, E. (1997). New Development Of Optimal Computing Budget Allocation For Discrete Event Simulation. *1997 WSC*, pp. 334–341.	52	2.48

16.3.5 Analysis Based on Research Area

The research areas pertaining to the publications is extracted from both ISI/CPCI-S and Scopus, as per the methodology outlined in Sect. 16.2. The research area is assigned to source publication; a publication may have more than one research area. As can be seen in Table 16.8, both ISI/CPCI-S and Scopus identify the top three research areas associated with the WSC publication, namely, *Computer Science*, *Engineering*, and *Operations Research and Management Science* (note: it is arguable that *ORMS* is identified as *Mathematics* in Scopus). Indeed, these have been the core parts of WSC since its inception as a conference. Records from Scopus show over 1100 records classified under both *Engineering* and *Chemical Engineering*. Since WSC is primarily a conference on discrete-event simulation, and simulation pertaining to chemical reactions and diffusion is perhaps more suited to the continuous paradigm of modelling, we conclude that there may be misclassification in tagging approx. 74% of records as *Chemical Engineering* under Scopus.

In Table 16.8, other research areas stand out: *Transportation* and *Computational Biology* are identified with some prominence in the ISI/CPCI-S database. However, it has to be ascertained whether these publications are specific to a particular year, or indeed, is this further evidence of interdisciplinary M&S? The analysis presented in Table 16.9 will try to answer this question. The analysis presented is specific to

Table 16.8 Classification of articles according to research areas (note: one article can be classified under multiple areas)

Research areas	ISI/CPCI-S		Scopus	
	Records classified under research area	% of total records retrieved from CSCI-S (6985)	Records classified under research area	% of Total records retrieved from scopus (1514)
Computer science	6337	90.72	1514	100
Engineering	3797	54.36	1119	73.91
Operations research and management science	2250	32.21	0	0
Mathematics	1477	21.15	1514	100
Mathematical computational biology	310	4.44	0	0
Transportation	237	3.39	0	0
Telecommunication	162	2.32	0	0
Chemical engineering	0	0	1119	73.91

ISI Web of Science. WOS Categories are the subject categories of the source publication. Since a source publication can be assigned to multiple subject categories, articles from those publications could therefore contain multiple subject categories. Comparing the WOS Categories with the total number of WSC publications/year, we make the following observations:

- *Ergonomics* was defined as subject category for the WSC 1989 conference proceedings
- *Telecommunications*—WSC 1991
- *Computer Science Cybernetics* and *Statistics Probability*—WSC 1996
- *Transportation*—WSC 1998
- Computer Science Artificial Intelligence—WSC 2006
- Engineering Multidisciplinary—WSC 2008
- Mathematical Computational Biology—WSC 2010
- The remaining ISI WOS categories have been attributed to a minimum of two WSC conference proceedings

Does the theme of the conference dictate further subject-wise categorisation of the WSC proceedings (as listed above), and which extends the traditional classification of WSC as a multi-disciplinary conference with primary focus on Computer Science, Engineering and ORMS/Mathematics?

Table 16.9 Classification of articles as per ISI web of science categories (*note* one article can be classified under multiple ISI categories)

ISI web of science categories	Records	% of total records retrieved from ISI/CSCI-S (6985)
Computer science interdisciplinary applications	3802	54.431
Engineering electrical electronic	2849	40.787
Computer science software engineering	2558	36.621
Operations research management science	2250	32.212
Computer science theory methods	1977	28.304
Computer science information systems	1355	19.399
Mathematics applied	1264	18.096
Engineering manufacturing	1239	17.738
Engineering industrial	708	10.136
Engineering multidisciplinary	396	5.669
Computer science artificial intelligence	320	4.581
Mathematical computational biology	310	4.438
Transportation	237	3.393
Statistics probability	213	3.049
Computer science cybernetics	213	3.049
Telecommunications	162	2.319
Ergonomics	145	2.076

16.3.6 Analysis Based on Funding Body

Our final analysis is specific to funding bodies that have been acknowledged by the WSC authors. The data for this is retrieved using the ISI/CPCI-S database. This means that only 26 years of data is being considered here (see Sect. 16.2). Funding agencies are essential for promoting academic, and non-profit, use of M&S. By knowing these agencies in different countries, it may be possible for researchers from different countries to work together to seek new hybrid funding and new collaboration. As can be seen from Table 16.10, almost half of the 275-odd studies listed below have received funding from the NSF. The predominance of US continues when we consider funding sources from US military and government departments—a total of 47 studies have been funded through channels, such as, DoE, AFOSR and NIH. Canada is a distant second with a total of 18 studies funded studies, followed by Spain with 16 studies funded through channels like various Spanish ministries.

Table 16.10 Funding bodies acknowledged in WSC publications (ISI/CPCI-S)

Funding body	Country	Studies funded
National Science Foundation (NSF), *including, NSF Grant Opportunities for Academic Liaison with Industry (GOALI)*	US	133
Department of energy	US	15
Air Force Office of Scientific Research (AFOSR)	US	12
Hong Kong Research Grants Council (RGC)	Hong Kong	12
Government of Canada Research Chairs	Canada	10
Office of Naval Research	US	9
Spanish Ministry of Economy and Competitiveness	Spain	8
Natural Sciences and Engineering Research Council of Canada (NSERC)	Canada	8
National Natural Science Foundation of China (NSFC)	China	8
United States Government	US	7
Fonds Européen de Développement Économique et Régional (FEDER)	EU	7
Engineering and Physical Sciences Research Council (EPSRC)	UK	7
Japan Society for the Promotion of Science (JSPS)	Japan	5
Spanish Ministry of Science and Innovation	Spain	4
National Institute of Health (NIH)	US	4
Intel Research Council	US	4
German Research Foundation (DFG)	Germany	4
Finnish Funding Agency for Technology and Innovation (TEKES)	Finland	4
Department of Universities, Research & Information Society of the Catalan Government	Spain	4
Coordination for the Improvement of Higher Education Personnel (CAPES)	Brazil	4
Australian Research Council (ARC)	Australia	4

16.4 Conclusion

We have presented a bottom-up, data-driven analysis of WSC papers from 1981 to 2016; a total of 8499 records were covered during this 36-year period. The goal in this analysis is to first obtain the data and then to "read the data" through the subjective process of asking questions. Creating data for its own sake is illogical, but starting with data and a set of objectives is the first step towards mining gems in the data set. Classic approaches to analyzing data sets involve looking for maxima, minima, points of inflection (in curves) and noting outliers. Each finding in this manuscript should be considered with a "query lens". What new questions do the data suggest? Generally, the analysis of data generates insight, but also a wealth of new questions we would not have known to ask ahead of time. We hope that while

this data analysis is preliminary, that it will be a starting point for new questions and also new data-driven analyses. All of the data indicates that WSC is a growing and vibrant conference, with the number of submissions and citations on the increase.

References

Clarivate Analytics (n.d.) Conference proceedings citation index. http://wokinfo.com/products_tools/multidisciplinary/webofscience/cpci/. Accessed date May 2017

Clarke R (2008a) An exploratory study of information systems researcher impact. Commun Assoc Inf Syst 22(1):1–32

Clarke R (2008b) A citation analysis of Australian information systems researchers: towards a new era. Commun Assoc Inf Syst 15(2):35–56

Claver E, Gonzalez R, Llopis J (2000) An analysis of research in information systems (1981–1997). Inf Manage 37(4):181–195

Dwivedi YK, Lal B, Mustafee N, Williams MD (2009) Profiling a decade of Information systems Frontiers' research. Inf Syst Front 11(1):87–102

Dwivedi YK, Kuljis J (2008) Profile of IS research published in the European journal of information systems. Eur J Inf Syst 17(6):678–693

Elsevier (n.d.) Who uses Scopus™. https://www.elsevier.com/solutions/scopus/content. Accessed date May 2017

Jacso P (2005) As we may search—comparison of major features of the Web of Science, Scopus, and Google Scholar citation-based and citation-enhanced databases. Curr Sci 89(9):1537–1547

Katsaliaki K, Mustafee N, Dwivedi YK, Williams T, Wilson JM (2010) A profile of OR research and practice published in the journal of the operational research society. J Oper Res Soc 61 (1):82–94

Mustafee N (2011) Evolution of IS research based on literature published in two leading IS journals—EJIS and MISQ. In: Proceedings of the 19th European conference on information systems (ECIS 2011), Helsinki, Finland, June 9–11, 2011. Paper 228

Mustafee N, Katsaliaki K, Fishwick P, Williams MD (2012) SCS—60 years and counting! A time to reflect on the Society's scholarly contribution to M&S from the turn of the Millennium. Simul Trans Soc Model Simul Int 88(9):1047–1071

Mustafee N, Katsaliaki K, Fishwick P (2014) Exploring the modelling & simulation knowledge base through journal co-citation analysis. Scientometrics 98(3):2145–2159

Palvia P, Pinjani P, Sibley EH (2007) A profile of information systems research published in information and management. Inf Manag 44(1):1–11

Palvia P, YK CP, Kakhki MD, Ghoshal T, Uppala V, Wang W (2017) A decade plus long introspection of research published in information & management. Inf Manage 54(2):218–227

Reuters T (2013) Web of Science® Help: Searching the Organizations—Enhanced List. http://images.webofknowledge.com/WOKRS59B4/help/WOS/hp_organizations_enhanced_index.html. Accessed date May 2017

USG (2017) University System of Georgia—Institutions. http://www.usg.edu/institutions/. Accessed date May 2017

WSC Foundation (2017) History of the Winter Simulation Conference (WSC) foundation. http://connect.informs.org/wscfoundation/history. Accessed date May 2017

Yilmaz L (2011) Reflections on the 60th year anniversary of SCS. Simul Trans Soc SCS 87(1–2): 3–4

Index

A. Tolk et al. (eds.), *Advances in Modeling and Simulation*, Simulation Foundations, Methods and Applications, DOI 10.1007/978-3-319-64182-9