The UMAP Journal

Vol. 29, No. 3

Subscription Rates for 2008 Calendar Year: Volume 29

Institutional Web Membership (Web Only)

Institutional Web Memberships does not provide print materials. Web memberships allow members to search our online catalog, download COMAP print materials, and reproduce for classroom use.

(Domestic) #2830 $449 **(Outside U.S.) #2830 $449**

Institutional Membership (Print Only)

Institutional Memberships receive print copies of The UMAP Journal quarterly, our annual CD collection UMAP Modules, Tools for Teaching, and our organizational newsletter Consortium.

(Domestic) #2840 $289 **(Outside U.S.) #2841 $319**

Institutional Plus Membership (Print Plus Web)

Institutional Plus Memberships receive print copies of the quarterly issues of The UMAP Journal, our annual collection UMAP Modules, Tools for Teaching, our organizational newsletter Consortium, and on-line membership that allows members to search our online catalog, download COMAP print materials, and reproduce for classroom use.

(Domestic) #2870 $569 **(Outside U.S.) #2871 $599**

For individual membership options visit www.comap.com for more information

To order, send a check or money order to COMAP, or call toll-free 1-800-77-COMAP (1-800-772-6627).

The UMAP Journal is published quarterly by the Consortium for Mathematics and Its Applications (COMAP), Inc., Suite 3B, 175 Middlesex Tpke., Bedford, MA, 01730, in cooperation with the American Mathematical Association of Two-Year Colleges (AMATYC), the Mathematical Association of America (MAA), the National Council of Teachers of Mathematics (NCTM), the American Statistical Association (ASA), the Society for Industrial and Applied Mathematics (SIAM), and The Institute for Operations Research and the Management Sciences (INFORMS). The Journal acquaints readers with a wide variety of professional applications of the mathematical sciences and provides a forum for the discussion of new directions in mathematical education (ISSN 0197-3622).

Periodical rate postage paid at Boston, MA and at additional mailing offices.

Send address changes to: info@comap.com

COMAP, Inc., Suite 3B, 175 Middlesex Tpke., Bedford, MA, 01730

Vol. 29, No. 3 2008

Table of Contents

Publisher's Editorial

Change

Solomon A. Garfunkel
Executive Director
COMAP, Inc.
175 Middlesex Turnpike, Suite 3B
Bedford , MA 01730–1459
s.garfunkel@mail.comap.com

This is the season of change—for good and bad. As I write this editorial, the election is a little less than a month away. The financial markets are imploding and the country appears more than ready to head in a new direction, even if it is unsure where that direction will take us. By the time you read this, many things will be clear. We will have a new administration, perhaps a very new administration. And we will likely be living individually and collectively on less—perhaps a lot less.

No matter. Some things still need to be done. I won't speak in this forum about health care or infrastructure or other changes in foreign and domestic policy—but I will speak of mathematics education. As small as our issues may seem at times of national and international stress, education—especially technical education—can always provide a way out and up. We cry out for mathematical and quantitative literacy, not because we are lobbyists or a special interest group trying to raise teacher salaries. We cry out for literacy because knowledge is the only way to prevent the abuses whose consequences we now endure.

How many times in these last few months have we heard about people who didn't understand the terms of their mortgages; of managers and bankers who didn't understand their degree of risk; of policy makers who didn't understand how the dominos could fall? Yes, derivatives are confusing. And yes, derivatives of derivatives are more confusing. But isn't this just a perfect example of why we talk about teaching mathematical modeling as a life skill? Mathematics education is not a zero-sum game. We don't want our students to learn more mathematics than other countries' students. That is just a foolish argument used to raise money, that is, the fear that another country will out perform us or another state will take our

The UMAP Journal 29 (3) (2008) 185–186.

high tech jobs.

The problem is much, much bigger. There simply are not enough mathematically-trained people in the world to run the world. The proof of that statement is all around us. And it is as much in our interest that the world's people become more quantitatively literate as it is that the citizens of our city, our state, and our country do. In theory, now there is less money to fund changes in mathematics education. But we must. We must see the issues and problems, as global issues and problems and work together to solve them.

The good news is that the energy and commitment to do the job are here. At the recent conference on the Future of Mathematics Education, co-sponsored by Math is More, I met with mathematics and mathematics education researchers, with college and high school faculty, with state and local administrators, with policy-makers, and with employers. We no longer talked about why; we talked about how. The need and desire for real change was palpable. And the energy was both exciting and challenging. People kept asking, "What can I do?"—as a classroom teacher, as a supervisor of mathematics, as a staff developer, as a curriculum developer, as a policy maker.

So while the times and problems are difficult, the will for positive change is here. Now is the time for all of us to gather together to make that change a reality.

About This Issue

Paul J. Campbell
Editor

This issue runs longer than a regular 92-page issue, to more than 200 pages. However, not all of the articles appear in the paper version. Some appear only on the *Tools for Teaching* 2008 CD-ROM (and at `http://www.comap.com` for COMAP members), which will reach members and subscribers later and will also contain the entire 2008 year of *Journal* issues.

All articles listed in the table of contents are regarded as published in the *Journal*. The abstract of each appears in the paper version. Pagination of the issue runs continuously, including in sequence articles that do not appear in the paper version. *So if, say, p. 250 in the paper version is followed by p. 303, your copy is not necessarily defective!* The articles on the intervening pages are on the CD-ROM.

We hope that you find this arrangement agreeable. It means that we do not have to procrusteanize the content to fit a fixed number of paper pages. We might otherwise be forced to select only two or three Outstanding MCM papers to publish. Instead, we continue to bring you the full content.

Modeling Forum

Results of the 2008 Mathematical Contest in Modeling

Frank Giordano, MCM Director
Naval Postgraduate School
1 University Circle
Monterey, CA 93943-5000
frgiorda@nps.navy.mil

Introduction

A total of 1,159 teams of undergraduates, from 338 institutions and 566 departments in 14 countries, spent the first weekend in February working on applied mathematics problems in the 24th Mathematical Contest in Modeling.

The 2008 Mathematical Contest in Modeling (MCM) began at 8:00 P.M. EST on Thursday, February 14 and ended at 8:00 P.M. EST on Monday, February 18. During that time, teams of up to three undergraduates were to research and submit an optimal solution for one of two open-ended modeling problems. Students registered, obtained contest materials, downloaded the problems at the appropriate time, and entered completion data through COMAP's MCM Website. After a weekend of hard work, solution papers were sent to COMAP on Monday. The top papers appear in this issue of *The UMAP Journal*.

Results and winning papers from the first 23 contests were published in special issues of *Mathematical Modeling* (1985–1987) and *The UMAP Journal* (1985–2007). The 1994 volume of *Tools for Teaching*, commemorating the tenth anniversary of the contest, contains the 20 problems used in the first 10 years of the contest and a winning paper for each year. That volume and the special MCM issues of the *Journal* for the last few years are available from COMAP. The 1994 volume is also available on COMAP's special *Modeling Resource* CD-ROM. Also available is *The MCM at 21* CD-ROM, which contains the 20 problems from the second 10 years of the contest, a winning paper from each year, and advice from advisors of Outstanding teams. These CD-ROMs can be ordered from COMAP at http://www.comap.com/product/cdrom/index.html.

The UMAP Journal 29 (3) (2008) 187–222.

This year's Problem A asked teams to consider the effects on land from the melting of the North Polar ice cap due to the predicted increase in global temperatures. Specifically, teams were asked to model the effects on the coast of Florida due to the melting every 10 years for the next 50 years, with particular attention to large metropolitan areas. Additionally, they were asked to propose appropriate responses to deal with the melting.

Problem B asked teams to develop an algorithm to construct Sudoku puzzles of varying difficulty. The problem required teams to develop metrics to define a difficulty level. Further, the team's algorithm and metrics were to be extensible to a varying number of difficulty levels, and they should illustrate their algorithm with at least four difficulty levels. The team's solution had to analyze the complexity of their algorithm.

The 9 Outstanding solution papers are published in this issue of *The UMAP Journal*, along with relevant commentaries.

In addition to the MCM, COMAP also sponsors the Interdisciplinary Contest in Modeling (ICM) and the High School Mathematical Contest in Modeling (HiMCM). The ICM runs concurrently with MCM and offers a modeling problem involving concepts in operations research, information science, and interdisciplinary issues in security and safety. The 2009 problem will have an environmental science theme. Results of this year's ICM are on the COMAP Website at `http://www.comap.com/undergraduate/contests`; results and Outstanding papers appeared in Vol. 29 (2008), No. 2. The HiMCM offers high school students a modeling opportunity similar to the MCM. Further details about the HiMCM are at `http://www.comap.com/highschool/contests`.

Problem A: Take a Bath

Consider the effects on land from the melting of the North Polar ice cap due to the predicted increase in global temperatures. Specifically, model the effects on the coast of Florida every 10 years for the next 50 years due to the melting, with particular attention given to large metropolitan areas. Propose appropriate responses to deal with this. A careful discussion of the data used is an important part of the answer.

Problem B: Creating Sudoku Puzzles

Develop an algorithm to construct Sudoku puzzles of varying difficulty. Develop metrics to define a difficulty level. The algorithm and metrics should be extensible to a varying number of difficulty levels. You should illustrate the algorithm with at least 4 difficulty levels. Your algorithm should guarantee a unique solution. Analyze the complexity of your algorithm. Your objective should be to minimize the complexity of the algorithm and meet the above requirements.

The Results

The solution papers were coded at COMAP headquarters so that names and affiliations of the authors would be unknown to the judges. Each paper was then read preliminarily by two "triage" judges at either Appalachian State University (Polar Melt Problem) or at the National Security Agency (Sudoku Problem). At the triage stage, the summary and overall organization are the basis for judging a paper. If the judges' scores diverged for a paper, the judges conferred; if they still did not agree, a third judge evaluated the paper.

Additional Regional Judging sites were created at the U.S. Military Academy and at the Naval Postgraduate School to support the growing number of contest submissions.

Final judging took place at the Naval Postgraduate School, Monterey, CA. The judges classified the papers as follows:

	Outstanding	Meritorious	Honorable Mention	Successful Participation	Total
Polar Melt Problem	4	64	182	315	565
Sudoku Problem	5	95	296	198	594
	9	159	378	513	1159

The 9 papers that the judges designated as Outstanding appear in this special issue of *The UMAP Journal*, together with commentaries. We list those teams and the Meritorious teams (and advisors) below; the list of all participating schools, advisors, and results is in the **Appendix**.

Outstanding Teams

Institution and Advisor	**Team Members**

Polar Melt Papers

"The Impending Effects of North Polar Ice Cap Melt"

College of Idaho	Benjamin Coate
Caldwell, ID	Nelson Gross
Michael P. Hitchman	Megan Longo

"A Convenient Truth: Forecasting Sea Level Rise"

Duke University	Jason Chen
Durham, NC	Brian Choi
Scott McKinley	Joonhahn Cho

"Fighting the Waves: The Effect of North Polar Ice Cap Melt on Florida"
University at Buffalo
Buffalo, NY
John Ringland

Amy M. Evans
Tracy L. Stepien

"Erosion in Florida: A Shore Thing"
University of Delaware
Newark, DE
Louis Frank Rossi

Matt Thies
Bob Liu
Zachary W. Ulissi

Sudoku Papers

"A Difficulty Metric and Puzzle Generator for Sudoku"
Harvard University
Cambridge, MA
Clifford H. Taubes

Christopher Chang
Zhou Fan
Yi Sun

"Taking the Mystery out of Sudoku Difficulty: An Oracular Model"
Harvey Mudd College
Claremont, CA
Jon Jacobsen

Sarah Fletcher
Frederick Johnson
David R. Morrison

"Difficulty-Driven Sudoku Puzzle Generation"
Harvey Mudd College
Claremont, CA
Zach Dodds

Martin Hunt
Christopher Pong
George Tucker

"Ease and Toil: Analyzing Sudoku"
University of Alaska Fairbanks
Fairbanks, AK
Orion S. Lawlor

Seth B. Chadwick
Rachel M. Krieg
Christopher E. Granade

"Cracking the Sudoku: A Deterministic Approach"
Youngstown State University
Youngstown, OH
George T. Yates

David Martin
Erica Cross
Matt Alexander

Meritorious Teams

Polar Melt Problem (65 teams)
Ann Arbor Huron High School, Mathematics, Ann Arbor, MI (Peter A. Collins)
Beihang University, Beijing, China (HongYing Liu)
Beijing Normal University, Beijing, Beijing, China (Li Cui)
Beijing Normal University, Beijing, (Laifu Liu)
Beijing University of Posts & Telecommunications, Electronic Engineering, Beijing, China (Zuguo He)
Beijing University of Posts & Telecommunications, Applied Mathematics, Beijing, China (Hongxiang Sun)
Central South University, Mechanical Design and Manufacturing Automation, Changsha, Hunan, China (Xinge Liu)
Central University of Finance and Economics, Applied Mathematics, Beijing, China (Donghong Li)
China University of Mining and Technology, Beijing, China (Lei Zhang) (two teams)
China University of Petroleum (Beijing), Beijing, China (Ling Zhao)
China University of Petroleum (East China), Qingdao, Shandong, China (Ziting Wang)
Chongqing University, Applied Chemistry, Chongqing, China (Zhiliang Li)
College of Charleston, Charleston, SC (Amy Langville)
Concordia College–New York, Bronxville, NY (Karen Bucher)
Dalian University of Technology, Software, Dalian, Liaoning, China (Zhe Li)
Donghua University, Shanghai China (Liangjian Hu)
Duke University, Durham, NC (Mark Huber)
East China University of Science and Technology, Physics, Shanghai, China (Lu ,hong)
Gannon University, Mathematics, Erie, PA (Jennifer A. Gorman)
Hangzhou Dianzi Unniversity, Information and Mathematics Science, Hangzhou, Zhejiang, China (Wei Li)
Harbin Institute of Technology Shiyan School, Mathematics, Harbin, Heilongjiang, China (Yunfei Zhang)
Hiram College, Hiram, OH (Brad S. Gubser)
McGill University, Mathematics and Statistics, Montreal, Quebec, Canada (Nilima Nigam)
Nankai University, Management Science and Engineering, Tianjin, Tianjin, China (Wenhua Hou)
National University of Defense Technology, Mathematics and Systems Science, Changsha, Hunan, China (Xiaojun Duan)
National University of Defense Technology, Mathematics and Systems Science, Changsha, Hunan, China (Yi Wu)
National University of Ireland, Galway, Galway, Ireland (Niall Madden)
National University of Ireland, Galway, Mathematical Physics, Galway, Ireland (Petri T. Piiroinen)
Ningbo Institute of Technology of Zhejiang University, Ningbo, China (Lihui Tu)
Northwestern Polytechnical University, Applied Physics, Xián, Shaanxi, China (Lei Youming)
Northwestern Polytechnical University, Applied Chemistry, Xián, Shaanxi, China (Sun Zhongkui)

Northwestern Polytechnical University, Natural and Applied Science, Xián, Shaanxi, China (Zhao Junfeng)
Oregon State University, Corvallis, OR (Nathan L. Gibson)
Pacific University, Physics, Forest Grove, OR (Juliet Brosing)
Peking University, Beijing, China (Sharon Lynne Murrel)
Providence College, Providence, RI, (Jeffrey T. Hoag)
Rensselaer Polytechnic Institute, Troy, NY (Peter R. Kramer)
University of Electronic Science and Technology of China, Applied Mathematics, Chengdu, Sichuan, China (Li Mingqi)
Shanghai Foreign Language School, Computer Science, Shanghai, China (Yue Sun)
Shanghai University of Finance & Economics, Applied Mathematics, Shanghai, China (Zhenyu Zhang)
Sichuan University, Electrical Engineering and Information, Chengdu, Sichuan, China (Yingyi Tan)
Slippery Rock University, Slippery Rock, PA (Richard J. Marchand)
South China Agricultural University, GuangZhou, Guangdong (ShaoMei Fang)
South China University of Technology, Guangzhou, Guangdong, China (Qin YongAn)
Sun Yat-Sen (Zhongshan) Univerisity, Guangzhou, Guangdong, China (GuoCan Feng)
Tsinghua University, Beijing, China (Jun Ye)
Tsinghua University, Beijing, China (Zhiming Hu)
Union College, Schenectady, NY (Jue Wang)
U.S. Military Academy, West Point, NY (Edward Swim)
University College Cork, Cork, Ireland (Benjamin W. McKay)
University College Cork, Cork, Ireland (Liya A. Zhornitskaya)
University of Guangxi, Mathematics & Information Science, Nanning, Guangxi, China (Ruxue Wu)
University of Guangxi, Mathematics & Information Science, Nanning, Guangxi, China (Zhongxing Wang)
University of Science and Technology Beijing, Beijing, China (Hu Zhixing)
University of Technology Jamaica, Chemical Engineering, Kingston, Jamaica, West Indies (Nilza G. Justiz-Smith)
Worcester Polytechnic Institute, Worcester, MA (Suzanne L. Weekes)
Wuhan University, Wuhan, Hubei, China (Yuanming Hu)
Xi'an Jiaotong University, Xian, Shaanxi, China (Jing Gao)
Xi'an Jiaotong University, Center for Mathematics Teaching and Experiment, Xian, Shaanxi, China (Xiaoe Ruan)
Xuzhou Institute of Technology, Xuzhou, Jiangsu, (Li Subei)
York University, Mathematics and Statistics, Toronto, ON, Canada, (Hongmei Zhu)
Yunnan University, Computer Science, Kunming, China (Shunfang Wang)
Zhejiang University, Hangzhou, Zhejiang, China (Zhiyi Tan)
Zhuhai College of Jinan University, Computer Science, Zhuhai, Guangdong, China (Zhang YunBiu)

Sudoku Problem (96 teams)
Beihang University, Beijing, China (Sun Hai Yan)
Beijing Institute of Technology, Beijing, China (Guifeng Yan)
Beijing Institute of Technology, Beijing, China (Houbao Xu)

Beijing Normal University, Beijing, China (Laifu Liu)
Beijing University of Posts & Telecommunications, Electronics Infomation Engineering, Beijing, China (Jianhua Yuan)
Bethel University, Arden Hills, MN (Nathan M. Gossett)
Cal Poly San Luis Obispo, San Luis Obispo, CA (Lawrence Sze)
Carroll College, Chemistry, Helena, MT (John C. Salzsieder)
Cheshire Academy, Cheshire, CT (Susan M Eident)
Clarkson University, Computer Science, Potsdam, NY (Katie Fowler)
College of Wooster, Wooster, OH (John R. Ramsay)
Dalian Maritime University, Dalian, Liaoning, China (Naxin Chen)
Dalian University of Technology, Software School, Dalian, Liaoning, China (Zhe Li) (two teams)
Daqing Petroleum Institute, Daqing, Heilongjiang, China (Kong Lingbin)
Daqing Petroleum Institute, Daqing, Heilongjiang, China (Yang Yunfeng)
Davidson College, Davidson NC (Richard D. Neidinger) (two teams)
East China Normal University, Shanghai, China (Yongming Liu)
East China University of Science and Technology, Shanghai, China (Su Chunjie)
Hangzhou Dianzi University, Information and Mathematics Science, Hangzhou, Zhejiang, China (Zheyong Qiu)
Harbin Institute of Technology, School of Astronautics, Management Science, Harbin, Heilongjiang, China (Bing Wen)
Harbin Institute of Technology, School of Science, Mathematics, Harbin, Heilongjiang, China (Yong Wang)
Harvey Mudd College, Computer Science, Claremont, CA (Zach Dodds)
Humboldt State University, Environmental Resources Engineering, Arcata, CA (Brad Finney)
James Madison University, Harrisonburg, VA (David B. Walton)
Jilin University, Changchun, Jilin, China (Huang Qingdao)
Jilin Universit, Changchun, Jilin, China (Xianrui Lu)
Korea Advanced Institute of Science & Technology, Daejeon, Korea (Yong-Jung Kim)
Luther College, Computer Science, Decorah, IA (Steven A. Hubbard)
Nanjing Normal University, Computer Science, Nanjing, Jiangsu, China (Wang Qiong)
Nanjing University, Nanjing, Jiangsu, China (Ze-Chun Hu)
Nanjing University of Posts & Telecommunications, Nanjing, Jiangsu, China (Jin Xu)
Nanjing University of Posts & Telecommunications, Nanjing, Jiangsu, China (Jun Ye)
National University of Defense Technology, Mathematics and Systems Science, Changsha, Hunan, China (Dan Wang)
National University of Defense Technology Mathematics and Systems Science, Changsha, Hunan, China (Meihua Xie)
National University of Defense Technology, Mathematics and Systems Science, Changsha, Hunan, China (Yong Luo)
Naval Aeronautical Engineering Academy (Qingdao), Machinery, Qingdao, Shandong, China (Cao Hua Lin)
North Carolina School of Science and Mathematics, Durham, NC (Daniel J. Teague)
Northwestern Polytechnical University, Xi'an, Shaanxi, China (Xiao Huayong)

Northwestern Polytechnical University, Xi'an, Shaanxi, China (Yong Xu)
Northwestern Polytechnical University, Xi'an, Shaanxi, China (Zhou Min)
Oxford University, Oxford, United Kingdom (Jeffrey H. Giansiracusa) (two teams)
Päivölä College of Mathematics, Tarttila, Finland (Janne Puustelli)
Peking University, Beijing, China (Xin Yi)
Peking University, Beijing, China (Xufeng Liu)
Peking University, Beijing, China (Yulong Liu)
Peking University, Financial Mathematics, Beijing, China (Shanjun Lin)
PLA University of Science and Technology, Meteorology, Nanjing, Jiangsu, China (Shen Jinren)
Princeton University, Operations Research and Financial Engineering, Princeton, NJ (Warren B. Powell)
Princeton University, Princeton, NJ (Robert Calderbank)
Renmin University of China, Finance, Beijing, China (Gao Jinwu)
Rensselaer Polytechnic Institute, Troy, NY (Donald Drew)
Shandong University, Software, Jinan, Shandong, China (Xiangxu Meng)
Shandong University, Mathematics & System Sciences, Jinan, Shandong, China (Bao Dong Liu)
Shandong University, Mathematics & System Sciences, Jinan, Shandong, China (Xiao Xia Rong)
Shandong University at Weihai, Weihai, Shandong, China (Yang Bing and Song Hui Min)
Shandong University at Weihai, Weihai, Shandong, China (Cao Zhulou and Xiao Hua)
Shanghai Foreign Language School, Shanghai, China (Liang Tao)
Shanghai Foreign Language School, Shanghai, China (Feng Xu)
Shanghai Sino European School of Technology, Shanghai, China (Wei Huang)
Shanghai University of Finance and Economics, Shanghai, China (Wenqiang Hao)
Shijiazhuang Railway Institute, Engineering Mechanics, Shijiazhuang, Hebei, China (Baocai Zhang)
Sichuan University, Chengdu, China (Qiong Chen)
Slippery Rock University, Physics, Slippery Rock, PA (Athula R Herat)
South China Normal University, Science of Information and Computation, Guangzhou, Guangdong, China (Tan Yang)
noindent South China University of Technology, Guangzhou, Guangdong, China (Liang ManFa)
South China University of Technology, Guangzhou, Guangdong, China (Liang ManFa)
South China University of Technology, Guangzhou, Guangdong, China (Qin YongAn)
Southwest University, Chongqing, China (Lei Deng)
Southwest University, Chongqing, China (Xianning Liu)
Southwestern University of Finance and Economics, Economics and Mathematics, Chengdu, Sichuan, China (Dai Dai)
Sun Yat-Sen (Zhongshan) University, Guangzhou, Guangdong, China (XiaoLong Jiang)
Tsinghua University, Beijing, China (Jun Ye)
University of Califonia–Davis, Davis, CA (Eva M. Strawbridge)
University of Colorado–Boulder, Boulder, CO (Anne M. Dougherty)

University of Colorado–Boulder, Boulder, CO (Luis Melara)
University of Delaware, Newark, DE (Louis Frank Rossi)
University of Iowa, Iowa City, IA (Ian Besse)
University of New South Wales, Sydney, NSW, Australia (James W. Franklin)
University of Puget Sound, Tacoma, WA (Michael Z. Spivey)
University of Science and Technology Beijing, Computer Science and Technology, Beijing, China (Zhaoshun Wang)
University of Washington, Applied and Computational Mathematical Sciences, Seattle, WA (Anne Greenbaum)
University of Western Ontario, London, ON, Canada (Allan B. MacIsaac)
University of Wisconsin–La Crosse, La Crosse, WI (Barbara Bennie)
University of Wisconsin–River Falls, River Falls, WI (Kathy A. Tomlinson)
Wuhan University, Wuhan, Hubei, China (Liuyi Zhong)
Wuhan University, Wuhan, Hubei, China (Yuanming Hu)
Xi'an Communication Institute, Xi'an, Shaanxi, China (Xinshe Qi)
Xidian University, Xi'an, Shaanxi, China (Guoping Yang)
Xidian University, Xi'an, Shaanxi, China (Jimin Ye)
Xidian University, Industrial and Applied Mathematics, Xi'an, Shaanxi, China (Qiang Zhu)
Zhejiang University, Hangzhou, Zhejiang, China (Yong Wu)
Zhejiang University City College, Information and Computing Science, Hangzhou, Zhejiang, China (Gui Wang)
Zhejiang University of Finance and Economics, Hangzhou, Zhejiang, China (Ji Luo)

Awards and Contributions

Each participating MCM advisor and team member received a certificate signed by the Contest Director and the appropriate Head Judge.

INFORMS, the Institute for Operations Research and the Management Sciences, recognized the teams from the College of Idaho (Polar Melt Problem) and University of Alaska Fairbanks (Sudoku Problem) as INFORMS Outstanding teams and provided the following recognition:

- a letter of congratulations from the current president of INFORMS to each team member and to the faculty advisor;
- a check in the amount of $300 to each team member;
- a bronze plaque for display at the team's institution, commemorating their achievement;
- individual certificates for team members and faculty advisor as a personal commemoration of this achievement;
- a one-year student membership in INFORMS for each team member, which includes their choice of a professional journal plus the *OR/MS Today* periodical and the INFORMS society newsletter.

The Society for Industrial and Applied Mathematics (SIAM) designated one Outstanding team from each problem as a SIAM Winner. The teams were from the University at Buffalo (Polar Melt Problem) and Harvard University (Sudoku Problem). Each of the team members was awarded a $300 cash prize and the teams received partial expenses to present their results in a special Minisymposium at the SIAM Annual Meeting in San Diego, CA in July. Their schools were given a framed hand-lettered certificate in gold leaf.

The Mathematical Association of America (MAA) designated one Outstanding North American team from each problem as an MAA Winner. The teams were from Duke University (Polar Melt Problem) and Harvey Mudd College Team (Hunt, Pong, and Tucker; advisor Dodds) (Sudoku Problem). With partial travel support from the MAA, the Duke University team presented their solution at a special session of the MAA Mathfest in Madison, WI in August. Each team member was presented a certificate by Richard S. Neal of the MAA Committee on Undergraduate Student Activities and Chapters.

Ben Fusaro Award

One Meritorious or Outstanding paper was selected for each problem for the Ben Fusaro Award, named for the Founding Director of the MCM and awarded for the fifth time this year. It recognizes an especially creative approach; details concerning the award, its judging, and Ben Fusaro are in Vol. 25 (3) (2004): 195–196. The Ben Fusaro Award winners were the University of Buffalo (Polar Melt Problem) and the University of Puget Sound (Sudoku Problem).

Judging

Director
Frank R. Giordano, Naval Postgraduate School, Monterey, CA

Associate Director
William P. Fox, Dept. of Defense Analysis, Naval Postgraduate School, Monterey, CA

Polar Melt Problem

Head Judge
Marvin S. Keener, Executive Vice-President, Oklahoma State University, Stillwater, OK

Associate Judges

William C. Bauldry, Chair, Dept. of Mathematical Sciences, Appalachian State University, Boone, NC (Head Triage Judge)

Patrick J. Driscoll, Dept. of Systems Engineering, U.S. Military Academy, West Point, NY

Ben Fusaro, Dept. of Mathematics, Florida State University, Tallahassee, FL (SIAM Judge)

Jerry Griggs, Mathematics Dept., University of South Carolina, Columbia, SC (Problem Author)

Mario Juncosa, RAND Corporation, Santa Monica, CA (retired)

Michael Moody, Olin College of Engineering, Needham, MA (MAA Judge)

David H. Olwell, Naval Postgraduate School, Monterey, CA (INFORMS Judge)

John L. Scharf, Mathematics Dept., Carroll College, Helena, MT (Ben Fusaro Award Judge)

Sudoku Problem

Head Judge

Maynard Thompson, Mathematics Dept., University of Indiana, Bloomington, IN

Associate Judges

Peter Anspach, National Security Agency, Ft. Meade, MD (Head Triage Judge)

Kelly Black, Mathematics Dept., Union College, Schenectady, NY

Karen D. Bolinger, Mathematics Dept., Clarion University of Pennsylvania, Clarion, PA

Jim Case (SIAM Judge)

Veena Mendiratta, Lucent Technologies, Naperville, IL (Problem Author)

Peter Olsen, Johns Hopkins Applied Physics Laboratory, Baltimore, MD

Kathleen M. Shannon, Dept. of Mathematics and Computer Science, Salisbury University, Salisbury, MD (MAA Judge)

Dan Solow, Mathematics Dept., Case Western Reserve University, Cleveland, OH (INFORMS Judge)

Michael Tortorella, Dept. of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ

Marie Vanisko, Dept. of Mathematics, Carroll College, Helena MT (Ben Fusaro Award Judge)

Richard Douglas West, Francis Marion University, Florence, SC

Dan Zwillinger, Raytheon Company, Sudbury, MA

Regional Judging Session at U.S. Military Academy

Head Judge
Patrick J. Driscoll, Dept. of Systems Engineering, United States Military Academy (USMA), West Point, NY

Associate Judges
Tim Elkins, Dept. of Systems Engineering, USMA
Michael Jaye, Dept. of Mathematical Sciences, USMA
Tom Meyer, Dept. of Mathematical Sciences, USMA
Steve Henderson, Dept. of Systems Engineering, USMA

Regional Judging Session at Naval Postgraduate School

Head Judge
William P. Fox, Dept. of Defense Analysis, Naval Postgraduate School (NPS), Monterey, CA

Associate Judges
William Fox, NPS
Frank Giordano, NPS

Triage Session for Polar Melt Problem

Head Triage Judge
William C. Bauldry, Chair, Dept. of Mathematical Sciences, Appalachian State University, Boone, NC

Associate Judges
Jeff Hirst, Rick Klima, and and René Salinas
—all from Dept. of Mathematical Sciences, Appalachian State University, Boone, NC

Triage Session for Sudoku Problem

Head Triage Judge
Peter Anspach, National Security Agency (NSA), Ft. Meade, MD

Associate Judges
Other judges from inside and outside NSA, who wish not to be named.

Sources of the Problems

The Polar Melt Problem was contributed by Jerry Griggs (Mathematics Dept., University of South Carolina, Columbia, SC), and the Sudoku Problem by Veena Mendiratta (Lucent Technologies, Naperville, IL).

Acknowledgments

Major funding for the MCM is provided by the National Security Agency (NSA) and by COMAP. We thank Dr. Gene Berg of NSA for his coordinating efforts. Additional support is provided by the Institute for Operations Research and the Management Sciences (INFORMS), the Society for Industrial and Applied Mathematics (SIAM), and the Mathematical Association of America (MAA). We are indebted to these organizations for providing judges and prizes.

We also thank for their involvement and support the MCM judges and MCM Board members for their valuable and unflagging efforts, as well as

- Two Sigma Investments. (This group of experienced, analytical, and technical financial professionals based in New York builds and operates sophisticated quantitative trading strategies for domestic and international markets. The firm is successfully managing several billion dollars using highly automated trading technologies. For more information about Two Sigma, please visit `http://www.twosigma.com`.)

Cautions

To the reader of research journals:

Usually a published paper has been presented to an audience, shown to colleagues, rewritten, checked by referees, revised, and edited by a journal editor. Each paper here is the result of undergraduates working on a problem over a weekend. Editing (and usually substantial cutting) has taken place; minor errors have been corrected, wording altered for clarity or economy, and style adjusted to that of *The UMAP Journal*. The student authors have proofed the results. Please peruse their efforts in that context.

To the potential MCM Advisor:

It might be overpowering to encounter such output from a weekend of work by a small team of undergraduates, but these solution papers are highly atypical. A team that prepares and participates will have an enriching learning experience, independent of what any other team does.

COMAP's Mathematical Contest in Modeling and Interdisciplinary Contest in Modeling are the only international modeling contests in which students work in teams. Centering its educational philosophy on mathematical modeling, COMAP uses mathematical tools to explore real-world problems. It serves the educational community as well as the world of work by preparing students to become better-informed and better-prepared citizens.

Appendix: Successful Participants

KEY:
P = Successful Participation
H = Honorable Mention
M = Meritorious
O = Outstanding (published in this special issue)

INSTITUTION	DEPT.	CITY	ADVISOR		
ALASKA					
U. Alaska Fairbanks	CS	Fairbanks	Orion S. Lawlor	B	H
U. Alaska Fairbanks	CS	Fairbanks	Orion S. Lawlor	B	O
ARIZONA					
Northern Arizona U.	Math & Stats	Flagstaff	Terence R. Blows	A	H
CALIFORNIA					
Cal Poly San Luis Obispo	Math	San Luis Obispo	Lawrence Sze	B	M
Cal Poly San Luis Obispo	Math	San Luis Obispo	Lawrence Sze	B	H
California State Poly. U.	Physics	Pomona	Kurt Vandervoort	B	P
California State Poly. U.	Math & Stats	Pomona	Joe Latulippe	B	P
Calif. State U. at Monterey Bay	Math & Stats	Seaside	Hongde Hu	A	H
Calif. State U. at Monterey Bay	Math & Stats	Seaside	Hongde Hu	A	H
Calif. State U. Northridge	Math	Northridge	Gholam-Ali Zakeri	B	P
Cal-Poly Pomona	Math	Pomona	Hubertus F. von Bremen	A	H
Cal-Poly Pomona	Physics	Pomona	Nina Abramzon	B	P
Harvey Mudd C.	Math	Claremont	Jon Jacobsen	A	H
Harvey Mudd C.	Math	Claremont	Jon Jacobsen	B	O
Harvey Mudd C.	CS	Claremont	Zach Dodds	B	M
Harvey Mudd C.	CS	Claremont	Zach Dodds	B	O
Humboldt State U.	Env'l Res. Eng.	Arcata	Brad Finney	A	H
Humboldt State U.	Env'l Res. Eng.	Arcata	Brad Finney	B	M
Irvine Valley C.	Math	Irvine	Jack Appleman	A	P
Pomona C.	Math	Claremont	Ami E. Radunskaya	A	H
Saddleback C.	Math	Mission Viejo	Karla Westphal	A	P
U. of Califonia Davis	Math	Davis	Eva M. Strawbridge	A	P
U. of Califonia Davis	Math	Davis	Eva M. Strawbridge	B	M
U. of California Merced	Natural Sci.	Merced	Arnold D. Kim	B	H
U. of San Diego	Math	San Diego	Cameron C. Parker	A	P
U. of San Diego	Math	San Diego	Cameron C. Parker	B	H
COLORADO					
U. of Colorado - Boulder	Appl. Math.	Boulder	Anne M. Dougherty	A	H
U. of Colorado - Boulder	Appl. Math.	Boulder	Bengt Fornberg	A	H
U. of Colorado - Boulder	Appl. Math.	Boulder	Anne Dougherty	B	M
U. of Colorado - Boulder	Appl. Math.	Boulder	Bengt Fornberg	B	H
U. of Colorado - Boulder	Appl. Math.	Boulder	Luis Melara	B	M
U. of Colorado Denver	Math	Denver	Gary A. Olson	A	P
CONNECTICUT					
Cheshire Acad.	Math	Cheshire	Susan M. Eident	B	M
Connecticut C.	Math	New London	Sanjeeva Balasuriya	A	P
Sacred Heart U.	Math	Fairfield	Peter Loth	B	P
Southern Connecticut State U.	Math	New Haven	Ross B. Gingrich	A	H
Southern Connecticut State U.	Math	New Haven	Ross B. Gingrich	B	H
DELAWARE					
U. of Delaware	Math Sci.	Newark	Louis Frank Rossi	A	O
U. of Delaware	Math Sci.	Newark	John A. Pelesko	B	P
U. of Delaware	Math Sci,	Newark	Louis Rossi	B	M
FLORIDA					
Bethune-Cookman U.	Math	Daytona Beach	Deborah Jones	A	P
Jacksonville U.	Math	Jacksonville	Robert A. Hollister	A	H

INSTITUTION	DEPT.	CITY	ADVISOR		
GEORGIA					
Georgia Southern U.	Math Sci.	Statesboro	Goran Lesaja	A	P
Georgia Southern U.	Math Sci.	Statesboro	Goran Lesaja	B	H
U. of West Georgia	Math	Carrollton	Scott Gordon	A	H
IDAHO					
C. of Idaho	Math/Phys. Sci.	Caldwell	Michael P. Hitchman	A	O
ILLINOIS					
Greenville C.	Math	Greenville	George R. Peters	A	P
INDIANA					
Goshen C.	Math	Goshen	Patricia A. Oakley	B	H
Rose-Hulmann Inst. of Tech.	Chemistry	Terre Haute	Michael Mueller	B	H
Rose-Hulman Inst. of Tech.	Chemistry	Terre Haute	Michael Mueller	B	H
Rose-Hulman Inst. of Tech.	Math	Terre Haute	William S. Galinaitis	A	H
Rose-Hulman Inst. of Tech.	Math	Terre Haute	William S. Galinaitis	B	P
Saint Mary's C.	Math	Notre Dame	Natalie K. Domelle	A	H
Saint Mary's C.	Math	Notre Dame	Natalie K. Domelle	B	H
IOWA					
Coe C.	Math Sci.	Cedar Rapids	Calvin R. Van Niewaal	B	H
Grand View C.	Math & CS	Des Moines	Sergio Loch	A	H
Grand View C.	Math & CS	Des Moines	Sergio Loch	A	H
Grinnell C.	Math & Stats	Grinnell	Karen L. Shuman	A	H
Luther C.	CS	Decorah	Steven A. Hubbard	B	H
Luther C.	CS	Decorah	Steven A. Hubbard	B	M
Luther C.	Math	Decorah	Reginald D. Laursen	B	H
Luther C.	Math	Decorah	Reginald D. Laursen	B	H
Simpson C.	Comp. Sci.	Indianola	Paul Craven	A	H
Simpson C.	Comp. Sci.	Indianola	Paul Craven	A	P
Simpson C.	Math	Indianola	William Schellhorn	A	H
Simpson C.	Math	Indianola	Debra Czarneski	A	P
Simpson C.	Math	Indianola	Rick Spellerberg	A	P
Simpson C.	Physics	Indianola	David Olsgaard	A	P
Simpson C.	Math	Indianola	Murphy Waggoner	B	P
Simpson C.	Math	Indianola	Murphy Waggoner	B	H
U. of Iowa	Math	Iowa City	Benjamin J. Galluzzo	A	H
U. of Iowa	Math	Iowa City	Kevin Murphy	A	H
U. of Iowa	Math	Iowa City	Ian Besse	B	M
U. of Iowa	Math	Iowa City	Scott Small	B	H
U. of Iowa	Math	Iowa City	Benjamin Galluzzo	B	H
KANSAS					
Kansas State U.	Math	Manhattan	David R. Auckly	B	H
Kansas State U.	Math	Manhattan	David R. Auckly	B	H
KENTUCKY					
Asbury C.	Math & CS	Wilmore	David L. Coulliette	A	H
Asbury C.	Math & CS	Wilmore	David L. Coulliette	B	H
Morehead State U.	Math & CS	Morehead	Michael Dobranski	B	P
Northern Kentucky U.	Math	Highl& Heights	Lisa Joan Holden	A	H
Northern Kentucky U.	Math	Highl& Heights	Lisa Holden	B	P
Northern Kentucky U.	Phys. & Geo.	Highl& Heights	Sharmanthie Fernando	A	P
LOUISIANA					
Centenary C.	Math & CS	Shreveport	Mark H. Goadrich	B	P
Centenary C.	Math & CS	Shreveport	Mark H. Goadrich	B	H
MAINE					
Colby C.	Math	Waterville	Jan Holly	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
MARYLAND					
Hood C.	Math	Frederick	Betty Mayfield	A	P
Loyola C.	Math Sci.	Baltimore	Jiyuan Tao	A	H
Loyola C.	Math Sci.	Baltimore	Jiyuan Tao	B	H
Mount St. Mary's U.	Math	Emmitsburg	Fred Portier	B	P
Salisbury U.	Math & CS	Salisbury	Troy V. Banks	B	P
Villa Julie C.	Math	Stevenson	Eileen C. McGraw	A	H
Washington C.	Math & CS	Chestertown	Eugene P. Hamilton	A	P
MASSACHUSETTS					
Bard C./Simon's Rock	Math	Great Barrington	Allen B. Altman	A	P
Bard C./Simon's Rock	Math	Great Barrington	Allen Altman	B	P
Bard C./Simon's Rock	Physics	Great Barrington	Michael Bergman	A	P
Harvard U.	Math	Cambridge	Clifford H. Taubes	B	O
Harvard U.	Math	Cambridge	Clifford H. Taubes	B	H
U. of Mass. Lowell	Math Sci.	Lowell	James Graham-Eagle	B	P
Worcester Poly. Inst.	Math Sci.	Worcester	Suzanne L. Weekes	A	M
Worcester Poly. Inst.	Math Sci.	Worcester	Suzanne L. Weekes	A	H
MICHIGAN					
Ann Arbor Huron HS	Math	Ann Arbor	Peter A. Collins	A	M
Lawrence Tech. U.	Math & CS	Southfield	Ruth G. Favro	A	H
Lawrence Tech. U.	Math & CS	Southfield	Guang-Chong Zhu	A	P
Lawrence Tech. U.	Math & CS	Southfield	Guang-Chong Zhu	A	P
Lawrence Tech. U.	Math & CS	Southfield	Ruth Favro	B	H
Siena Heights U.	Math	Adrian	Jeff C. Kallenbach	A	P
Siena Heights U.	Math	Adrian	Tim H. Husband	A	P
Siena Heights U.	Math	Adrian	Tim H. Husband	B	P
MINNESOTA					
Bethel U.	Math & CS	Arden Hills	Nathan M. Gossett	B	M
Carleton C.	Math	Northfield	Laura M. Chihara	A	H
Northwestern C.	Sci. & Math.	St. Paul	Jonathan A. Zderad	A	P
MISSOURI					
Drury U.	Math & CS	Springfield	Keith James Coates	A	H
Drury U.	Math & CS	Springfield	Keith James Coates	A	P
Drury U.	Physics	Springfield	Bruce W. Callen	A	P
Drury U.	Physics	Springfield	Bruce W. Callen	A	H
Saint Louis U.	Math & CS	St. Louis	David A. Jackson	B	H
Saint Louis U.	Eng., Aviation & Tech.	St. Louis	Manoj S. Patankar	A	H
Truman State U.	Math & CS	Kirksville	Steve Jay Smith	B	H
U. of Central Missouri	Math & CS	Warrensburg	Nicholas R. Baeth	A	P
U. of Central Missouri	Math & CS	Warrensburg	Nicholas R. Baeth	B	P
MONTANA					
Carroll C.	Chemistry	Helena	John C. Salzsieder	B	M
Carroll C.	Chemistry	Helena	John C. Salzsieder	A	P
Carroll C.	Math., Eng. , & CS	Helena	Holly S. Zullo	B	H
Carroll C.	Math., Eng., & CS	Helena	Mark Parker	A	H
NEBRASKA					
Nebraska Wesleyan U.	Math & CS	Lincoln	Melissa Claire Erdmann	A	P
Wayne State C.	Math	Wayne	Tim Hardy	A	P
NEW JERSEY					
Princeton U.	Math	Princeton	Robert Calderbank	B	M
Princeton U.	OR & Fin. Eng.	Princeton	Robert J. Vanderbei	B	H
Princeton U.	OR & Fin. Eng.	Princeton	Robert J. Vanderbei	B	H
Princeton U.	OR & Fin. Eng.	Princeton	Warren B. Powell	B	P
Princeton U.	OR & Fin. Eng.	Princeton	Warren B. Powell	B	M

INSTITUTION	DEPT.	CITY	ADVISOR		
Richard Stockton C.	Math	Pomona	Brandy L. Rapatski	A	H
Rowan U.	Math	Glassboro	Paul J. Laumakis	B	P
Rowan U.	Math	Glassboro	Christopher Jay Lacke	B	H
NEW MEXICO					
NM Inst. Mining & Tech.	Math	Socorro	John D. Starrett	B	P
New Mexico State U.	Math Sci.	Las Cruces	Caroline P. Sweezy	A	P
NEW YORK					
Clarkson U.	Comp. Sci.	Potsdam	Katie Fowler	B	H
Clarkson U.	Comp. Sci.	Potsdam	Katie Fowler	B	M
Clarkson U.	Math	Potsdam	Joseph D. Skufca	A	H
Clarkson U.	Math	Potsdam	Joseph D. Skufca	B	P
Colgate U.	Math	Hamilton	Dan Schult	B	H
Concordia C.	Bio. Chem. Math.	Bronxville	Karen Bucher	A	M
Concordia C.	Math	Bronxville	John F. Loase	A	H
Concordia C.	Math	Bronxville	John F. Loase	B	H
Cornell U.	Math	Ithaca	Alexander Vladimirsky	B	H
Cornell U.	OR & Ind'l Eng.	Ithaca	Eric Friedman	B	H
Ithaca C.	Math	Ithaca	John C. Maceli	B	H
Ithaca C.	Physics	Ithaca	Bruce G. Thompson	B	H
Nazareth C.	Math	Rochester	Daniel Birmajer	A	P
Rensselaer Poly. Inst.	Math Sci.	Troy	Peter R. Kramer	A	M
Rensselaer Poly. Inst.	Math Sci.	Troy	Peter R. Kramer	B	H
Rensselaer Poly. Inst.	Math Sci.	Troy	Donald Drew	B	M
Rensselaer Poly.Inst.	Math Sci.	Troy	Donald Drew	B	H
Union C.	Math	Schenectady	Jue Wang	A	M
U.S. Military Acad.	Math Sci.	West Point	Edward Swim	A	M
U.S. Military Acad.	Math Sci.	West Point	Robert Burks	B	H
U. at Buffalo	Math	Buffalo	John Ringland	A	O
U. at Buffalo	Math	Buffalo	John Ringland	B	H
Westchester Comm. Coll.	Math	Valhalla	Marvin Littman	B	P
NORTH CAROLINA					
Davidson C.	Math	Davidson	Donna K. Molinek	B	H
Davidson C.	Math	Davidson	Donna K. Molinek	B	H
Davidson C.	Math	Davidson	Richard D. Neidinger	B	M
Davidson C.	Math	Davidson	Richard D. Neidinger	B	M
Duke U.	Math	Durham	Scott McKinley	A	O
Duke U.	Math	Durham	Mark Huber	A	M
Duke U.	Math	Durham	David Kraines	B	H
Duke U.	Math	Durham	Dan Lee	B	P
Duke U.	Math	Durham	Lenny Ng	B	H
Duke U.	Math	Durham	Bill Pardon	B	H
Meredith C.	Math & CS	Raleigh	Cammey Cole Manning	A	H
NC Schl of Sci. & Math.	Math	Durham	Daniel J. Teague	B	M
NC Schl of Sci. & Math.	Math	Durham	Daniel J. Teague	B	H
U. of North Carolina	Math	Chapel Hill	Sarah A. Williams	A	H
U. of North Carolina	Math	Chapel Hill	Brian Pike	A	H
Wake Forest U.	Math	Winston Salem	Miaohua Jiang	A	H
Western Carolina U.	Math & CS	Cullowhee	Jeff Lawson	A	H
Western Carolina U.	Math & CS	Cullowhee	Erin K. McNelis	B	H
OHIO					
C. of Wooster	Math & CS	Wooster	John R. Ramsay	B	M
Hiram C.	Math	Hiram	Brad S. Gubser	A	M
Kenyon C.	Math	Gambier	Dana C. Paquin	A	H
Malone C.	Math & CS	Canton	David W. Hahn	A	H
Malone C.	Math & CS	Canton	David W. Hahn	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Miami U.	Math & Stats	Oxford	Doug E. Ward	A	P
Miami U.	Math & Stats	Oxford	Doug E. Ward	B	H
U. of Dayton	Math	Dayton	Youssef N. Raffoul	B	H
Xavier U.	Math & CS	Cincinnati	Bernd E. Rossa	A	H
Xavier U.	Math & CS	Cincinnati	Bernd E. Rossa	B	H
Youngstown State U.	Math & Stats	Youngstown	George T. Yates	A	H
Youngstown State U.	Math & Stats	Youngstown	Angela Spalsbury	A	H
Youngstown State U.	Math & Stats	Youngstown	Angela Spalsbury	A	H
Youngstown State U.	Math & Stats	Youngstown	Gary J. Kerns	A	H
Youngstown State U.	Math & Stats	Youngstown	Paddy W. Taylor	A	P
Youngstown State U.	Math & Stats	Youngstown	Paddy W. Taylor	A	H
Youngstown State U.	Math & Stats	Youngstown	George Yates	B	O
Youngstown State U.	Math & Stats	Youngstown	Gary Kerns	B	H
OKLAHOMA					
Oklahoma State U.	Math	Stillwater	Lisa A. Mantini	B	H
SE Okla. State U.	Math	Durant	Karl H. Frinkle	A	P
OREGON					
Lewis & Clark Coll.	Math Sci.	Portland	Liz Stanhope	A	P
Linfield C.	Comp. Sci.	McMinnville	Daniel K. Ford	B	H
Linfield C.	Math	McMinnville	Jennifer Nordstrom	A	H
Linfield C.	Math	McMinnville	Jennifer Nordstrom	B	H
Oregon State U.	Math	Corvallis	Nathan L. Gibson	A	M
Oregon State U.	Math	Corvallis	Nathan L. Gibson	A	P
Oregon State U.	Math	Corvallis	Vrushali A. Bokil	B	H
Pacific U.	Math	Forest Grove	Michael Boardman	B	H
Pacific U.	Math	Forest Grove	John August	A	H
Pacific U.	Physics	Forest Grove	Juliet Brosing	A	M
Pacific U.	Physics	Forest Grove	Steve Hall	A	P
PENNSYLVANIA					
Bloomsburg U.	Math, CS, & Stats	Bloomsburg	Kevin Ferland	A	H
Bucknell U.	Math	Lewisburg	Peter McNamara	B	H
Gannon U.	Math	Erie	Jennifer A. Gorman	A	M
Gettysburg C.	Math	Gettysburg	Benjamin B. Kennedy	B	H
Gettysburg C.	Math	Gettysburg	Benjamin B. Kennedy	B	P
Juniata C.	Math	Huntingdon	John F. Bukowski	A	H
Shippensburg U.	Math	Shippensburg	Paul T. Taylor	A	H
Slippery Rock U.	Math	Slippery Rock	Richard J. Marchand	A	M
Slippery Rock U.	Math	Slippery Rock	Richard J. Marchand	B	H
Slippery Rock U.	Physics	Slippery Rock	Athula R. Herat	B	M
U. of Pittsburgh	Math	Pittsburgh	Jonathan Rubin	B	H
Westminster C.	Math & CS	New Wilmington	Barbara T. Faires	A	H
Westminster C.	Math & CS	New Wilmington	Barbara T. Faires	A	H
Westminster C.	Math & CS	New Wilmington	Warren D. Hickman	B	H
Westminster C.	Math & CS	New Wilmington	Carolyn K. Cuff	B	P
RHODE ISLAND					
Providence C.	Math	Providence	Jeffrey T. Hoag	A	M
SOUTH CAROLINA					
C. of Charleston	Math	Charleston	Amy Langville	A	M
C. of Charleston	Math	Charleston	Amy Langville	B	H
Columbia C.	Math & Comp.	Columbia	Nieves A. McNulty	B	H
Francis Marion U.	Math	Florence	David W. Szurley	B	H
Midlands Technical Coll.	Math	Columbia	John R. Long	A	H
Midlands Technical Coll.	Math	Columbia	John R. Long	B	P
Wofford C.	Comp. Sci.	Spartanburg	Angela B. Shiflet	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
SOUTH DAKOTA					
SD Schl of Mines & Tech.	Math & CS	Rapid City	Kyle Riley	B	P
TENNESSEE					
Belmont U.	Math & CS	Nashville	Andrew J. Miller	A	H
Tennessee Tech U.	Math	Cookeville	Andrew J. Hetzel	B	H
U. of Tennessee	MAth	Knoxville	Suzanne Lenhart	A	P
TEXAS					
Angelo State U.	Math	San Angelo	Karl J. Havlak	B	H
Angelo State U.	Math	San Angelo	Karl J. Havlak	B	P
Texas A& M–Commerce	Math	Commerce	Laurene V. Fausett	A	H
Trinity U.	Math	San Antonio	Peter Olofsson	B	P
Trinity U.	Math	San Antonio	Diane Saphire	B	H
VIRGINIA					
James Madison U.	Math & Stats	Harrisonburg	Ling Xu	A	P
James Madison U.	Math & Stats	Harrisonburg	David B. Walton	B	M
Longwood U.	Math & CS	Farmville	M. Leigh Lunsford	A	P
Longwood U.	Math & CS	Farmville	M. Leigh Lunsford	B	H
Maggie Walker Gov. Schl	Math	Richmond	John Barnes	B	H
Mills E. Godwin HS	Sci. Math Tech.	Richmond	Ann W. Sebrell	B	H
Mills E. Godwin HS	Sci. Math Tech.	Richmond	Ann W. Sebrell	B	P
Roanoke C.	Math CS Phys.	Salem	David G. Taylor	A	P
U. of Richmond	Math & CS	Richmond	Kathy W. Hoke	B	H
U. of Virginia	Math	Charlottesville	Irina Mitrea	B	H
U. of Virginia	Math	Charlottesville	Tai Melcher	B	H
Virginia Tech	Math	Blacksburg	Henning S. Mortveit	B	H
Virginia Western	Math	Roanoke	Steve Hammer	A	P
WASHINGTON					
Central Washington U.	Math	Ellensburg	James Bisgard	A	P
Heritage U.	Math	Toppenish	Richard W. Swearingen	B	H
Pacific Lutheran U.	Math	Tacoma	Rachid Benkhalti	A	H
Pacific Lutheran U.	Math	Tacoma	Rachid Benkhalti	B	H
Seattle Pacific U.	Electr. Eng.	Seattle	Melani Plett	B	H
Seattle Pacific U.	Math	Seattle	Wai Lau	B	H
Seattle Pacific U.	Math	Seattle	Wai Lau	B	H
U. of Puget Sound	Math	Tacoma	Michael Z. Spivey	A	H
U. of Puget Sound	Math	Tacoma	Michael Z. Spivey	B	M
U. of Washington	Appl./Comp'l Math.	Seattle	Anne Greenbaum	B	M
U. of Washington	Appl./Comp'l Math.	Seattle	Anne Greenbaum	A	H
U. of Washington	Math	Seattle	James Morrow	B	H
U. of Washington	Math	Seattle	James Allen Morrow	A	H
Washington State U.	Math	Pullman	Mark F. Schumaker	B	H
Western Washington U.	Math	Bellingham	Tjalling Ypma	A	H
Western Washington U.	Math	Bellingham	Tjalling Ypma	A	H
WISCONSIN					
Beloit C.	Math & CS	Beloit	Paul J. Campbell	B	H
U. of Wisc.–La Crosse	Math	La Crosse	Barbara Bennie	B	M
U. of Wisc.–Eau Claire	Math	Eau Claire	Simei Tong	B	H
U. of Wisc.–River Falls	Math	River Falls	Kathy A. Tomlinson	B	M

INSTITUTION	DEPT.	CITY	ADVISOR		
AUSTRALIA					
U. of New South Wales	Math & Stats	Sydney	James W. Franklin	A	H
U. of New South Wales	Math & Stats	Sydney	James W. Franklin	B	M
U. of S. Queensland	Math & Comp.	Toowoomba	Sergey A. Suslov	B	H
CANADA					
McGill U.	Math & Stats	Montreal	Nilima Nigam	A	M
McGill U.	Math & Stats	Montreal	Nilima Nigam	A	P
U. Toronto at Scarborough	CS & Math.	Toronto	Paul S. Selick	A	H
U. of Western Ontario	Appl. Math.	London	Allan B. MacIsaac	B	M
York U.	Math & Stats	Toronto	Hongmei Zhiu	A	M
CHINA					
Anhui					
Anhui U.	Appl. Math	Hefei	Ranchao Wu	B	H
Anhui U.	Appl. Math	Hefei	Quanbing Zhang	B	H
Anhui U.	Electr. Eng.	Hefei	Quancal Gan	B	H
Anhui U.	Electr. Eng.	Hefei	Quancal Gan	B	H
Hefei U. of Tech.	Appl. Math	Hefei	Yongwu Zhou	B	P
Hefei U. of Tech.	Comp'l Math	Hefei	Youdu Huang	A	H
Hefei U. of Tech.	Math	Hefei	Xueqiao Du	A	H
Hefei U. of Tech.	Math	Hefei	Huaming Su	B	P
Hefei U. of Tech.	Math	Hefei	Huaming Su	A	P
Hefei U. of Tech.	Math	Hefei	Xueqiao Du	B	P
U. of Sci. & Tech. of China	CS	Hefei	Lixin Duan	A	H
U. of Sci. & Tech. of China	Electr. Eng./InfoSci.	Hefei	Xing Gong	B	P
U. of Sci. & Tech. of China	Gifted Young	Hefei	Weining Shen	A	P
U. of Sci. & Tech. of China	Modern Physics	Hefei	Kai Pan	A	P
U. of Sci. & Tech. of China	InfoSci. & Tech.	HeFei	Dong Li	B	P
U. of Sci. & Tech. of China	Physics	Hefei	Zhongmu Deng	A	P
Beijing					
Acad. of Armored Force Eng.	Funda. Courses	Beijing	Chen Jianhua	B	P
Acad. of Armored Force Eng.	Funda. Courses	Beijing	Chen Jianhua	B	H
Acad. of Armored Force Eng.	Mech. Eng.	Beijing	Han De	A	P
Beihang U.	Advanced Eng.	Beijing	Wu San Xing	B	H
Beihang U.	Astronautics	Beijing	Sanxing Wu	B	P
Beihang U.	Astronautics	Beijing	Jian Ma	B	P
Beihang U.	Sci.	Beijing	Linping Peng	A	P
Beihang U.	Sci.	Beijing	Sun Hai Yan	B	M
Beihang U.	Sci.	Beijing	Sun Hai Yan	B	H
Beihang U.	Sci.	Beijing	HongYing Liu	A	M
Beijing Electr. Sci. & Tech. Inst.	Basic Education	Beijing	Cui Meng	A	P
Beijing Electr. Sci. & Tech. Inst.	Basic Education	Beijing	Cui Meng	A	P
Beijing Forestry U.	Info	Beijing	Jie Ma	B	H
Beijing Forestry U.	Info	Beijing	Xiaochun Wang	A	P
Beijing Forestry U.	Math	Beijing	Mengning Gao	B	P
Beijing Forestry U.	Math	Beijing	Li Hongjun	A	P
Beijing Forestry U.	Math	Beijing	Xiaochun Wang	B	H
Beijing Forestry U.	Mech. Eng.	Beijing	Zhao Dong	B	H
Beijing Forestry U.	Sci.	Beijing	Xiaochun Wang	A	P
Beijing Forestry U.	Sci.	Beijing	Mengning Gao	A	P
Beijing High Schl Four	Math	Beijing	Jinli Miao	A	P
Beijing High Schl Four	Math	Beijing	Jinli Miao	B	H
Beijing Inst. of Tech.	InfoTech.	Beijing	Hongzhou Wang	A	P
Beijing Inst. of Tech.	Math	Beijing	Houbao XU	B	P
Beijing Inst. of Tech.	Math	Beijing	Hua-Fei Sun	A	H
Beijing Inst. of Tech.	Math	Beijing	Bing-Zhao Li	A	P
Beijing Inst. of Tech.	Math	Beijing	Hongzhou Wang	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Beijing Inst. of Tech.	Math	Beijing	Chunguang Xiong	A	P
Beijing Inst. of Tech.	Math	Beijing	Xuewen Li	A	H
Beijing Inst. of Tech.	Math	Beijing	Xiuling Ma	A	P
Beijing Inst. of Tech.	Math	Beijing	Xiaoxia Yan	B	H
Beijing Inst. of Tech.	Math	Beijing	Guifeng Yan	B	M
Beijing Inst. of Tech.	Math	Beijing	Hongzhou Wang	B	H
Beijing Inst. of Tech.	Math	Beijing	Houbao XU	B	M
Beijing Jiaotong U.	Appl. Math	Beijing	Jing Zhang	A	P
Beijing Jiaotong U.	Math	Beijing	Weijia Wang	A	P
Beijing Jiaotong U.	Math	Beijing	Hong Zhang	A	P
Beijing Jiaotong U.	Math	Beijing	Faen Wu	A	P
Beijing Jiaotong U.	Math	Beijing	Pengjian Shang	A	P
Beijing Jiaotong U.	Math	Beijing	Xiaoxia Wang	A	P
Beijing Jiaotong U.	Math	Beijing	Zhonghao Jiang	A	P
Beijing Jiaotong U.	Math	Beijing	Bingli Fan	A	P
Beijing Jiaotong U.	Math	Beijing	Bingtuan Wang	B	P
Beijing Jiaotong U.	Math	Beijing	Weijia Wang	B	P
Beijing Jiaotong U.	Math	Beijing	Keqian Dong	B	P
Beijing Jiaotong U.	Math	Beijing	Bingli Fan	B	P
Beijing Jiaotong U.	Math	Beijing	Shangli Zhang	A	P
Beijing Jiaotong U.	Math	Beijing	Jun Wang	B	H
Beijing Jiaotong U.	Math	Beijing	Minghui Liu	A	P
Beijing Jiaotong U.	Math	Beijing	Xiaoming Huang	A	H
Beijing Jiaotong U.	Math	Beijing	Minghui Liu	B	H
Beijing Jiaotong U.	Statistics	Beijing	Weidong Li	B	H
Beijing Lang. & Culture U.	CS	Beijing	Guilong Liu	B	H
Beijing Normal U.	Geography	Beijing	Yongjiu Dai	A	P
Beijing Normal U.	Math	Beijing	Yingzhe Wang	A	H
Beijing Normal U.	Math	Beijing	He Qing	A	H
Beijing Normal U.	Math	Beijing	Li Cui	A	M
Beijing Normal U.	Math	Beijing	Laifu Liu	B	M
Beijing Normal U.	Math	Beijing	Liu Yuming	A	P
Beijing Normal U.	Math	Beijing	Laifu Liu	A	M
Beijing Normal U.	Math	Beijing	Zhengru Zhang	B	H
Beijing Normal U.	Math	Beijing	Haiyang Huang	A	P
Beijing Normal U.	Math	Beijing	Haiyang Huang	A	P
Beijing Normal U.	Resources	Beijing	Jianjun Wu	A	P
Beijing Normal U.	Stats	Beijing	Chun Yang	A	P
Beijing Normal U.	Stats & Financial Math.	Beijing	Xingwei Tong	A	P
Beijing Normal U.	Stats & Financial Math.	Beijing	Cui Hengjian	A	P
Beijing Normal U.	Stats & Financial Math.	Beijing	Jacob King	B	H
Beijing Normal U.	Stats & Financial Math.	Beijing	Shumei Zhang	B	H
Beijing Normal U.	Sys. Sci.	Beijing	Zengru Di	A	P
Beijing U. of Aero. & Astro.	Aero. Sci. & Eng.	Beijing	Linping Peng	A	H
Beijing U. of Aero. & Astro.	Instr. Sci. & Opto-electr. Eng.	Beijing	Linping Peng	A	H
Beijing U. of Chem. Tech.	Electr. Sci.	Beijing	Xiaoding Shi	B	P
Beijing U. of Chem. Tech.	Electr. Sci.	Beijing	Guangfeng Jiang	A	P
Beijing U. of Chem. Tech.	Math	Beijing	Jinyang Huang	A	P
Beijing U. of Chem. Tech.	Math	Beijing	Xinhua Jiang	A	H
Beijing U. of Chem. Tech.	Math & InfoSci.	Beijing	Hui Liu	B	P
Beijing U. of Posts & Tele.	Appl. Math	Beijing	Zuguo He	A	H
Beijing U. of Posts & Tele.	Appl. Math	Beijing	Hongxiang Sun	A	M
Beijing U. of Posts & Tele.	Appl. Math	Beijing	Hongxiang Sun	A	P
Beijing U. of Posts & Tele.	Automation	Beijing	Jianhua Yuan	B	H
Beijing U. of Posts & Tele.	Automation	Beijing	Jianhua Yuan	B	P
Beijing U. of Posts & Tele.	Comm. Eng.	Beijing	Xiaoxia Wang	A	P
Beijing U. of Posts & Tele.	Comm. Eng.	Beijing	Xiaoxia Wang	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Beijing U. of Posts & Tele.	Comm. Eng.	Beijing	Zuguo He	B	H
Beijing U. of Posts & Tele.	CS & Tech.	Beijing	Hongxiang Sun	B	H
Beijing U. of Posts & Tele.	Econ. & Mgmnt	Beijing	Tianping Shuai	A	P
Beijing U. of Posts & Tele.	Electr. Eng.	Beijing	Qing Zhou	A	P
Beijing U. of Posts & Tele.	Electr. Eng.	Beijing	Zuguo He	A	M
Beijing U. of Posts & Tele.	Electr. Eng.	Beijing	Zuguo He	B	P
Beijing U. of Posts & Tele.	Electr. & Information Eng.	Beijing	Xinchao Zhao	B	H
Beijing U. of Posts & Tele.	Electr. & Information Eng.	Beijing	Jianhua Yuan	B	M
Beijing U. of Tech.	Appl. Sci.	Beijing	Xue Yi	A	P
Beijing Wuzi U.	Info	Beijing	Advisor Group	A	P
Beijing Wuzi U.	Info	Beijing	Advisor Group	A	P
Beijing Wuzi U.	Math	Beijing	Advisor Group	B	P
Beijing Wuzi U.	Math	Beijing	Advisor Group	B	H
Central U. of Finance & Econ.	Appl. Math	Beijing	Xiuguo Wang	A	P
Central U. of Finance & Econ.	Appl. Math	Beijing	Xiuguo Wang	A	H
Central U. of Finance & Econ.	Appl. Math	Beijing	Zhaoxu Sun	A	P
Central U. of Finance & Econ.	Appl. Math	Beijing	Donghong Li	A	M
Central U. of Finance & Econ.	Appl. Math	Beijing	Xiaoming Fan	B	H
China Agricultural U.	Sci.	Beijing	Zou Hui	A	P
China Agricultural U.	Sci.	Beijing	Li GuoHui	B	P
China Agricultural U.	Sci.	Beijing	Shi YuanChang	B	P
China Agricultural U.	Sci.	Beijing	Yang JianPing	B	H
China U. of GeoSci.	InfoTech.	Beijing	Cuixiang Wang	A	P
China U. of GeoSci.	InfoTech.	Beijing	Shuai Zhang	A	P
China U. of GeoSci.	InfoTech.	Beijing	Cuixiang Wang	B	H
China U. of GeoSci.	InfoTech.	Beijing	Shuai Zhang	B	P
China U. of GeoSci.	Math	Beijing	Linlin Zhao	A	P
China U. of GeoSci.	Math	Beijing	Huang	B	H
China U. of Mining & Tech.	Math Sci.	Beijing	Lei Zhang	A	M
China U. of Mining & Tech.	Math Sci.	Beijing	Lei Zhang	A	M
China U. of Mining & Tech.	Sci.	Beijing	Ping Jing	A	P
China U. of Mining & Tech.	Sci.	Beijing	Ping Jing	A	P
China U. of Petroleum	Math & Physics	Beijing	Ling Zhao	A	M
China U. of Petroleum	Math & Physics	Beijing	Xiaoguang Lu	A	H
China U. of Petroleum	Math & Physics	Beijing	Pei Wang	B	H
China Youth U. for Polit. Sci.	Econ.	Beijing	Yanxia Zheng	B	P
North China Electr. Power U.	Automation	Beijing	Xiangjie Liu	A	H
North China Electr. Power U.	Automation	Beijing	Guotian Yang	B	H
North China Electr. Power U.	Electr. Eng.	Beijing	Yini Xie	B	P
North China Electr. Power U.	Electr. Eng.	Beijing	Yongqiang Zhu	B	H
North China Electr. Power U.	Math & Physics	beijing	Qirong Qiu	A	P
North China Electr. Power U.	Math & Physics	Beijing	Qirong Qiu	B	H
North China U. of Tech.	Math & InfoSci.	Beijing	Quan Zheng	B	H
Peking U.	Ctr for Econ. Res.	Beijing	Qiang Gong	B	H
Peking U.	CS	BeiJing	Lida Zhu	B	P
Peking U.	Econ.	Beijing	Dong Zhiyong	B	H
Peking U.	Financial Math	Beijing	Shanjun Lin	A	H
Peking U.	Financial Math	Beijing	Shanjun Lin	B	M
Peking U.	Journalism & Comm.	Beijing	Hua Sun	B	H
Peking U.	Life Sci.	Beijing	Chengcai An	A	P
Peking U.	Machine Intelligence	Beijing	Juan Huang	B	P
Peking U.	Machine Intelligence	Beijing	Juan Huang	B	P
Peking U.	Math Sci.	Beijing	Xufeng Liu	A	P
Peking U.	Math Sci.	Beijing	Yulong Liu	B	H
Peking U.	Math Sci.	Beijing	Yulong Liu	B	M
Peking U.	Math Sci.	Beijing	Minghua Deng	B	P
Peking U.	Math Sci.	Beijing	Sharon Lynne Murrel	A	M

INSTITUTION	DEPT.	CITY	ADVISOR		
Peking U.	Math Sci.	Beijing	Xin Yi	B	M
Peking U.	Math Sci.	Beijing	Xin Yi	B	P
Peking U.	Math Sci.	Beijing	Xufeng Liu	B	M
Peking U.	Math Sci.	Beijing	Minghua Deng	A	P
Peking U.	Mechanics	Beijing	Zhi Li	A	P
Peking U.	Physics	Beijing	Xiaodong Hu	A	P
Peking U.	Physics	Beijing	Xiaodong Hu	B	H
Peking U.	Physics	Beijing	Yongqiang Sun	B	P
Peking U.	Quantum Electronics	Beijing	Zhigang Zhang	B	H
Peking U.	Sci. & Eng. Comp.	Beijing	Peng He	B	H
Renmin U. of China	Finance	Beijing	Gao Jinwu	B	M
Renmin U. of China	Info	Beijing	Yonghong Long	A	P
Renmin U. of China	Info	Beijing	Yong Lin	B	H
Renmin U. of China	Info	Beijing	Yong Lin	B	H
Renmin U. of China	Info	Beijing	Litao Han	B	P
Renmin U. of China	Math	Beijing	Jinwu Gao	A	H
Tsinghua U.	Math	Beijing	Jun Ye	A	M
Tsinghua U.	Math	Beijing	Zhiming Hu	A	M
Tsinghua U.	Math	Beijing	Zhiming Hu	A	P
Tsinghua U.	Math	Beijing	Jun Ye	B	M
U. of Sci. & Tech. Beijing	Appl. Math	Beijing	Wang Hui	A	P
U. of Sci. & Tech. Beijing	Appl. Math	Beijing	Hu Zhixing	A	M
U. of Sci. & Tech. Beijing	Appl. Math	Beijing	Zhu Jing	A	P
U. of Sci. & Tech. Beijing	Appl. Math	Beijing	Hu Zhixing	B	H
U. of Sci. & Tech. Beijing	CS & Tech.	Beijing	Zhaoshun Wang	B	M
U. of Sci. & Tech. Beijing	Math	Beijing	Zhu Jing	B	H
U. of Sci. & Tech. Beijing	Math	Beijing	Wang Hui	A	H
Chongqin					
Chongqing Normal U.	Math & CS	Chongqing	Xuewen Liu	B	P
Chongqing Normal U.	Math & CS	Chongqing	Yan Wei	B	P
Chongqing U.	Appl. Chemistry	Chongqing	Zhiliang Li	A	M
Chongqing U.	Math & Phys., Info. & CS	Chongqing	Li Fu	A	H
Chongqing U.	Software Eng.	Chongqing	Xiaohong Zhang	A	P
Chongqing U.	Stats & Act'l Sci.	Chongqing	Tengzhong Rong	A	P
Chongqing U.	Stats & Act'l Sci.	Chongqing	Zhengmin Duan	A	H
Chongqing U.	Stats & Act'l Sci.	Chongqing	Zhengmin Duan	B	H
Southwest U.	Appl. Math	Chongqing	Yangrong Li	B	H
Southwest U.	Appl. Math	Chongqing	Xianning Liu	B	M
Southwest U.	Math	Chongqing	Lei Deng	B	M
Southwest U.	Math	Chongqing	Lei Deng	B	H
Fujian					
Fujian Agri. & Forestry U.	Comp. & InfoTech.	Fuzhou	Lurong Wu	A	H
Fujian Agri. & Forestry U.	Comp. & InfoTech.	Fuzhou	Lurong Wu	B	H
Fujian Normal U.	CS	Fuzhou	Chen Qinghua	B	H
Fujian Normal U.	Education Tech.	Fuzhou	Lin Muhui	B	P
Fujian Normal U.	Math	Fuzhou	Zhiqiang Yuan	A	P
Fujian Normal U.	Math	Fuzhou	Zhiqiang Yuan	B	H
Quanzhou Normal U.	Math	Quanzhou	Xiyang Yang	A	H
Guangdong					
Jinan U.	Electr.	Guangzhou	Shiqi Ye	B	H
Jinan U.	Math	Guangzhou	Shizhuang Luo	A	P
Jinan U.	Math	Guangzhou	Daiqiang Hu	B	H
Shenzhen Poly.	Electr. & InfoEng.	Shenzhen	JianLong Zhong	B	H
Shenzhen Poly.	Ind'l Training Ctr	Shenzhen	Dong Ping Wei	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Shenzhen Poly.	Ind'l Training Ctr	Shenzhen	Hong Mei Tian	A	H
Shenzhen Poly.	Ind'l Training Ctr	Shenzhen	ZhiYong Liu	A	P
Shenzhen Poly.	Ind'l Training Ctr	Shenzhen	Jue Wang	B	P
South China Agricultural U.	Math	Guangzhou	ShaoMei Fang	A	M
South China Agricultural U.	Math	Guangzhou	ShaoMei Fang	B	H
South China Agricultural U.	Math	Guangzhou	ShengXiang Zhang	B	P
South China Agricultural U.	Math	Guangzhou	ShengXiang Zhang	B	P
South China Normal U.	Info & Computation	Guangzhou	Tan Yang	B	M
South China Normal U.	Math	Guangzhou	Henggeng Wang	A	H
South China Normal U.	Math	Guangzhou	Shaohui Zhang	B	P
South China Normal U.	Math	Guangzhou	Hunan Li	B	H
South China U. of Tech.	Appl. Math	Guangzhou	Qin YongAn	A	M
South China U. of Tech.	Appl. Math	Guangzhou	Huang Ping	A	P
South China U. of Tech.	Appl. Math	Guangzhou	Qin YongAn	B	M
South China U. of Tech.	Appl. Math	Guangzhou	Liang ManFa	B	M
South China U. of Tech.	Appl. Math	Guangzhou	Liang ManFa	B	M
Sun Yat-Sen (Zhongshan) U.	Comp. Sci.	Guangzhou	ZePeng Chen	B	H
Sun Yat-Sen (Zhongshan) U.	Math	Guangzhou	GuoCan Feng	A	M
Sun Yat-Sen (Zhongshan) U.	Math	Guangzhou	GuoCan Feng	B	H
Sun Yat-Sen (Zhongshan) U.	Math	Guangzhou	ZhengLu Jiang	B	H
Sun Yat-Sen (Zhongshan) U.	Math	Guangzhou	XiaoLong Jiang	B	M
Zhuhai C. of Jinan U.	CS	Zhuhai	Zhang YunBiu	A	M
Zhuhai C. of Jinan U.	CS	Zhuhai	Zhang YunBiu	A	P
Zhuhai C. of Jinan U.	Packaging Eng.	Zhuhai	Zhiwei Wang	A	P
Guangxi					
GuangXi Teachers Educ. U.	Math & CS	Nanning	Mai Xiongfa	A	P
GuangXi Teachers Educ. U.	Math & CS	Nanning	Wei Chengdong	A	P
GuangXi Teachers Educ. U.	Math & CS	Nanning	Su Huadong	B	P
GuangXi Teachers Educ. U.	Math & CS	Nanning	Chen Jianwei	B	P
Guilin U. of Electr. Tech.	Math & Comp'l Sci.	Guilin	Yongxiang Mo	A	P
Guilin U. of Electr. Tech.	Math & Comp'l Sci.	Guilin	Ning Zhu	A	P
Guilin U. of Electr. Tech.	Math & Comp'l Sci.	Guilin	Ning Zhu	B	P
U. of Guangxi	Math & InfoSci.	Nanning	Ruxue Wu	A	M
U. of Guangxi	Math & InfoSci.	Nanning	Ruxue Wu	A	P
U. of Guangxi	Math & InfoSci.	Nanning	Zhongxing Wang	A	M
U. of Guangxi	Math & InfoSci.	Nanning	Chunhong Li	A	P
U. of Guangxi	Math & InfoSci.	Nanning	Yuejin Lv	B	P
Hebei					
Hebei Poly. U.	Light Industry	Tangshan	Lihui Zhou	A	P
Hebei Poly. U.	Light Industry	Tangshan	Yan Yan	B	H
Hebei Poly. U.	Light Industry	Tangshan	Shaohong YAN	B	P
Hebei Poly. U.	Sci.	Tangshan	Yamian Peng	B	H
Hebei Poly. U.	Sci.	Tangshan	Lihong LI	B	P
Hebei U.	Math & CS	Baoding	Qiang Hua	A	P
Hebei U.	Math & CS	Baoding	Qiang Hua	B	H
North China Electr. Power U.	Funda. Courses	Baoding	JinWei Shi	A	P
North China Electr. Power U.	Funda. Courses	Baoding	Gendai Gu	A	P
North China Electr. Power U.	Math & Physics	Baoding	Po Zhang	B	H
North China Electr. Power U.	Math & Physics	Baoding	Jinggang Liu	A	P
North China Electr. Power U.	Math & Physics	Baoding	Huifeng Shi	A	P
North China Electr. Power U.	Math & Physics	Baoding	Yagang Zhang	A	P
North China Electr. Power U.	Math & Physics	Baoding	Jinggang Liu	B	H
Shijiazhuang Railway Inst.	Eng. Mechanics	Shijiazhuang	Baocai Zhang	B	M
Shijiazhuang Railway Inst.	Eng. Mechanics	Shijiazhuang	Baocai Zhang	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Helongjiang					
Daqing Petroleum Inst.	Math	Daqing	Yang Yunfeng	A	P
Daqing Petroleum Inst.	Math	Daqing	Yang Yunfeng	B	M
Daqing Petroleum Inst.	Math	Daqing	Kong Lingbin	B	M
Harbin Eng. U.	Appl. Math	Harbin	Gao Zhenbin	A	P
Harbin Eng. U.	Appl. Math	Harbin	Gao Zhenbin	A	P
Harbin Eng. U.	Info & CS	Harbin	Zhang Xiaowei	A	H
Harbin Eng. U.	Info & CS	Harbin	Zhang Xiaowei	A	P
Harbin Eng. U.	Math	Harbin	Luo Yuesheng	B	H
Harbin Inst. of Tech.	Astro: Mgmt Sci.	Harbin	Bing Wen	B	M
Harbin Inst. of Tech.	Astro: Mgmt Sci.	Harbin	Bing Wen	A	H
Harbin Inst. of Tech.	Astro: Math	Harbin	Dongmei Zhang	B	H
Harbin Inst. of Tech.	Astro: Math	Harbin	Jiqyun Shao	A	P
Harbin Inst. of Tech.	Astro: Math	Harbin	Jiqyun Shao	B	H
Harbin Inst. of Tech.	CS	Harbin	Zheng Kuang	A	P
Harbin Inst. of Tech.	CS & Tech.	Harbin	Lili Zhang	A	H
Harbin Inst. of Tech.	EE & Aut.: Math.	Harbin	Guanghong Jiao	A	P
Harbin Inst. of Tech.	Mgmnt Sci.	Harbin	Jianguo Bao	A	P
Harbin Inst. of Tech.	Mgmnt Sci.	Harbin	Jianguo Bao	B	H
Harbin Inst. of Tech.	Mgmnt: Math	Harbin	Boping Tian	B	H
Harbin Inst. of Tech.	Mgmnt Sci. & Eng.	Harbin	Hong Ge	A	P
Harbin Inst. of Tech.	Mgmnt Sci. & Eng.	Harbin	Wei Shang	A	P
Harbin Inst. of Tech.	Mgmnt Sci. & Eng.	Harbin	Wei Shang	B	H
Harbin Inst. of Tech.	Math	Harbin	Xianyu Meng	B	P
Harbin Inst. of Tech.	Math	Harbin	Xianyu Meng	B	P
Harbin Inst. of Tech.	Math	Harbin	Yong Wang	B	H
Harbin Inst. of Tech.	Math	Harbin	Yong Wang	B	M
Harbin Inst. of Tech.	Math	Harbin	Chiping Zhang	B	P
Harbin Inst. of Tech.	Math	Harbin	Guofeng Fan	A	P
Harbin Inst. of Tech.	Math	Harbin	Shouting Shang	B	H
Harbin Inst. of Tech.	Math	Harbin	Guofeng Fan	B	P
Harbin Inst. of Tech.	Math	Harbin	Daohua Li	A	H
Harbin Inst. of Tech.	Math	Harbin	Daohua Li	B	P
Harbin Inst. of Tech.	Math	Harbin	Baodong Zheng	A	P
Harbin Inst. of Tech.	Math	Harbin	Boying Wu	A	P
Harbin Inst. of Tech.	Math	Harbin	Bo Han	B	P
Harbin Inst. of Tech.	Math	Harbin	Bo Han	B	P
Harbin Inst. of Tech.	Network Project	Harbin	Xiaoping Ji	A	P
Harbin Inst. of Tech.	Sci.	Harbin	Boying Wu	B	H
Harbin Inst. of Tech.	Software Eng.	Harbin	Yan Liu	A	P
Harbin Inst. of Tech.	Software Eng.	Harbin	Yan Liu	A	P
Harbin Inst. of Tech., Shiyan School	Math	Harbin	Xiaofeng Shi	B	H
Harbin Inst. of Tech., Shiyan School	Math	Harbin	Kean Liu	A	H
Harbin Inst. of Tech., Shiyan School	Math	Harbin	Kean Liu	B	P
Harbin Inst. of Tech., Shiyan School	Math	Harbin	Yunfei Zhang	A	H
Harbin Inst. of Tech., Shiyan School	Math	Harbin	Yunfei Zhang	A	M
Harbin U. of Sci. & Tech.	Math	Harbin	Dongmei Li	A	P
Harbin U. of Sci. & Tech.	Math	Harbin	Fengqiu Liu	A	P
Harbin U. of Sci. & Tech.	Math	Harbin	Dongyan Chen	B	H
Harbin U. of Sci. & Tech.	Math	Harbin	Shuzhong Wang	B	H
Harbin U. of Sci. & Tech.	Math	Harbin	Guangyue Tian	B	H
Heilongjiang Inst. of Sci. & Tech.	Math & Mech.	Harbin	Hongyan Zhang	A	H
Heilongjiang Inst. of Sci. & Tech.	Math & Mech.	Harbin	Hui Chen	A	P
Heilongjiang Inst. of Sci. & Tech.	Math & Mech.	Harbin	Yanhua Yuan	B	P
Heilongjiang Inst. of Tech.	Math	Harbin	Dalu Nie	B	P
Heilongjiang Inst. of Tech.	Math	Harbin	Dalu Nie	B	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Henan					
Zhengzhou Inst. of Electr. Tech.	The Second	Zhengzhou	Xiaoyong Zhang	A	P
Zhengzhou Inst. of Electr. Tech.	The Second	Zhengzhou	Xiaoyong Zhang	B	H
Zhengzhou Inst. of Electr. Tech.	The Third	Zhengzhou	Lixin Jia	B	H
Zhengzhou Inst. of Electr. Tech.	The Third	Zhengzhou	Lixin Jia	B	H
Zhengzhou Inst. of Survey/Map.	Cart./Geo. InfoEng.	Zhengzhou	Shi Bin	A	P
Zhengzhou Inst. of Survey/Map.	Cart./Geo. InfoEng.	Zhengzhou	Shi Bin	B	P
Zhengzhou Inst. of Survey/Map.	Geod./Navig. Eng.	Zhengzhou	Li Guohui	A	P
Zhengzhou Inst. of Survey/Map.	Geod./Navig. Eng.	Zhengzhou	Li Guohui	B	P
Zhengzhou Inst. of Sci.	Appl. Math	Zhengzhou	Jianfeng Guo	B	P
Zhengzhou Inst. of Sci.	Appl. Math	Zhengzhou	Zhibo Lu	B	H
Zhengzhou Inst. of Sci.	Appl. Physics	Zhengzhou	Yuan Tian	A	H
Hubei					
Huazhong Normal U.	Math & Stats	Wuhan	Bo Li	A	P
Huazhong U. of Sci. & Tech.	Electr. & InfoEng.	Wuhan	Yan Dong	B	P
Huazhong U. of Sci. & Tech.	Ind'l & Mfg SysEng.	Wuhan	Haobo Qiu	A	P
Huazhong U. of Sci. & Tech.	Ind'l & Mfg SysEng.	Wuhan	Liang Gao	B	P
Huazhong U. of Sci. & Tech.	Math	Wuhan	Zhengyang Mei	B	P
Huazhong U. of Sci. & Tech.	Math	Wuhan	Zhengyang Mei	B	H
Three Gorges U.	Math	Yichang City	Qin Chen	A	P
Wuhan U.	Appl. Math	Wuhan	Yuanming Hu	B	P
Wuhan U.	Appl. Math	Wuhan	Yuanming Hu	B	M
Wuhan U.	Civil Eng.: Math	Wuhan	Yuanming Hu	A	H
Wuhan U.	CS	Wuhan	Hu Yuanming	B	P
Wuhan U.	Electr. Info	Wuhan	Yuanming Hu	B	P
Wuhan U.	InfoSecurity	Wuhan	Xinqi Hu	A	H
Wuhan U.	Math	Wuhan	Yuanming Hu	A	M
Wuhan U.	Math & Appl. Math	Wuhan	Chengxiu Gao	B	P
Wuhan U.	Math & Stats	Wuhan	Hu Yuanming	A	H
Wuhan U.	Math & Stats	Wuhan	Shihua Chen	A	H
Wuhan U.	Math & Stats	Wuhan	Liuyi Zhong	B	M
Wuhan U.	Math & Stats	Wuhan	Xinqi Hu	B	P
Wuhan U.	Math & Stats	Wuhan	Gao Chengxiu	B	P
Wuhan U.	Math & Stats	Wuhan	Yuanming Hu	B	P
Wuhan U.	Software Eng.	Wuhan	Liuyi Zhong	A	P
Wuhan U. of Sci. & Tech.	Sci.	Wuhan	Advisor Team	B	P
Wuhan U. of Sci. & Tech.	Sci.	Wuhan	Advisor Team	B	P
Wuhan U. of Tech.	Math	Wuhan	Huang Xiaowei	A	P
Wuhan U. of Tech.	Math	Wuhan	Zhu Huiying	A	P
Wuhan U. of Tech.	Math	Wuhan	Chu Yangjie	A	P
Wuhan U. of Tech.	Math	Wuhan	Liu Yang	B	H
Wuhan U. of Tech.	Physics	Wuhan	He Lang	B	P
Wuhan U. of Tech.	Physics	Wuhan	Chen Jianye	A	P
Wuhan U. of Tech.	Physics	Wuhan	Chen Jianye	B	P
Wuhan U. of Tech.	Physics	Wuhan	He Lang	B	P
Wuhan U. of Tech.	Stats	Wuhan	Mao Shuhua	B	P
Wuhan U. of Tech.	Stats	Wuhan	Chen Jiaqing	B	P
Wuhan U. of Tech.	Stats	Wuhan	Li Yuguang	A	P
Hunan					
Central South U.	Automation	Changsha	He Wei	A	P
Central South U.	Biomedicial Eng.	Changsha	Hou Muzhou	A	H
Central South U.	Civil Eng.	Changsha	Shihua Zhu	A	P
Central South U.	CS	Changsha	Xuanyun Qin	B	H
Central South U.	CS	Changsha	Xuanyun Qin	B	H
Central South U.	Eng. Mgmnt	Changsha	Zhoushun Zheng	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Central South U.	Infomation & CS	Changsha	Zhoushun Zheng	B	H
Central South U.	Info-phys. & Geo. Eng.	Changsha	Zhang Yan	A	P
Central South U.	Info-phys. & Geo. Eng.	Changsha	Zhang Yan	A	P
Central South U.	Material Physics	Changsha	Xuanyun Qin	B	H
Central South U.	Math & Appl. Math.	Chang sha	Cheng Liu	A	H
Central South U.	Math	Changsha	He Wei	A	P
Central South U.	Mech. Des. & Mfg Aut.	Changsha	Xin Ge Liu	A	M
Central South U.	Mech. Des. & Mfg Aut.	Changsha	Xin Ge Liu	A	P
Central South U.	Traffic & Info. Eng.	Changsha	Cheng Liu	A	P
Changsha U. of Sci. & Tech.	Math & CS	Changsha	Zhang Tong	A	P
Changsha U. of Sci. & Tech.	Math & CS	Changsha	Liang Dai	B	P
Changsha U. of Sci. & Tech.	Math & CS	Changsha	Quan Xie	B	P
Changsha U. of Sci. & Tech.	Math & CS	Changsha	Liu Tan	B	P
Hunan Inst. of Humanities	Sci. & Tech. Math	Loudi	DiChen Yang	A	H
Hunan U.	Math & Econometrics	Changsha	Huahui Yan	A	P
Hunan U.	Math & Econometrics	Changsha	Yizhao Chen	B	H
Hunan U.	Math & Econometrics	Changsha	Yuanbei Deng	B	H
Hunan U.	Math & Econometrics	Changsha	Changrong Liu	B	P
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Mengda Wu	A	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Lizhi Cheng	A	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Meihua Xie	A	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Yi Wu	A	M
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Xiaojun Duan	A	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Yong Luo	A	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Xiaojun Duan	A	M
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Lizhi Cheng	B	P
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Meihua Xie	B	M
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Dan Wang	B	H
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Dan Wang	B	M
National U. of Defense Tech.	Math & Sys. Sci.	Changsha	Yong Luo	B	M
Inner Mongolia					
Inner Mongolia U.	Math	Huhhot	Zhuang Ma	A	P
Inner Mongolia U.	Math	Huhhot	Mei Wang	A	P
Inner Mongolia U.	Math	Huhhot	Zhuang Ma	B	P
Jiangsu					
Huaiyin Inst. of Tech.	Comp. Sci.	Huaian	Zhuang Yuming	A	P
Huaiyin Inst. of Tech.	Comp. Sci.	Huaian	Zhuang Yuming	B	P
Nanjing Normal U.	CS	Nanjing	Wang Qiong	B	M
Nanjing Normal U.	CS	Nanjing	Wang Qiong	B	H
Nanjing Normal U.	Financial Math	Nanjing	Wang Xiao Qian	A	H
Nanjing Normal U.	Financial Math	Nanjing	Wang Xiao Qian	B	H
Nanjing Normal U.	Math	Nanjing	Zhu Qun Sheng	A	P
Nanjing Normal U.	Math	Nanjing	Zhu Qun Sheng	B	P
Nanjing U.	Chem. & Chem. Eng.	Nanjing	Xujie Shen	B	P
Nanjing U.	Electr. Sci. & Eng.	Nanjing	Haodong Wu	A	H
Nanjing U.	Electr. Eng.	Nanjing	Jianchun Cheng	B	H
Nanjing U.	Env't	Nanjing	Xin Qian	A	H
Nanjing U.	Intensive Instruction	Nanjing	Weihua Huang	B	H
Nanjing U.	Intensive Instruction	Nanjing	Weiyi Su	A	P
Nanjing U.	Math	Nanjing	Ze-Chun Hu	B	M
Nanjing U.	Math	Nanjing	Guo Fei Zhou	A	P
Nanjing U.	Math	Nanjing	Weihua Huang	B	H
Nanjing U.	Math	Nanjing	Ming Kong	B	P
Nanjing U.	Math	Nanjing	Ming Kong	B	H
Nanjing U.	Math	Nanjing	Zechun Hu	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Nanjing U. of Finance & Econ.	Finance Economy	Nanjing	Chen Meixia	B	P
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Aiju Shi	B	P
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Kong Gaohua	B	H
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Kong Gaohua	B	H
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	LiWei Xu	B	H
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Qiu Zhonghua	A	P
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Jun Ye	B	M
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Zhong Hua Qiu	B	P
Nanjing U. of Posts & Tele.	Math & Physics	Nanjing	Jin Xu	B	M
Nanjing U. of Sci. & Tech.	Appl. Math	Nanjing	Peibiao Zhao	B	P
Nanjing U. of Sci. & Tech.	Appl. Math	Nanjing	Chungen Xu	B	H
Nanjing U. of Sci. & Tech.	Math	Nanjing	Zhipeng Qiu	B	H
Nanjing U. of Sci. & Tech.	Stats	Nanjing	Liwei Liu	A	P
Nantong U.	Arch. & Civil Eng.	Nantong	Hongmei Liu	A	P
Nantong U.	Electr. Eng.	Nantong	Guoping Lu	A	P
Nantong U.	Sci.	NanTong	Xiaojian Zhou	B	P
PLA U. of Sci. & Tech.	Comm. Eng.	Nanjing	Yao Kui	A	P
PLA U. of Sci. & Tech.	Meteor.: Appl. Math & Phys.	Nanjing	Shen Jinren	B	M
PLA U. of Sci. & Tech.	Sci.: Appl. Math & Phys.	Nanjing	Liu Shousheng	B	H
PLA U. of Sci. & Tech.	Sci,: Appl. Math & Physics	Nanjing	Teng Jiajun	A	H
Southeast U. at Jiulonghu	Math	Nanjing	Jun Huang	A	H
Southeast U. at Jiulonghu	Math	Nanjing	Enshui Chen	A	P
Southeast U. at Jiulonghu	Math	Nanjing	Jianhua Zhou	B	P
Southeast U. at Jiulonghu	Math	Nanjing	Xiang Yin	B	H
Southeast U.	Math	Nanjing	Feng Wang	A	P
Southeast U.	Math	Nanjing	Xingang Jia	B	H
Southeast U.	Math	Nanjing	Xingang Jia	B	H
Southeast U.	Math	Nanjing	Dan He	A	P
Southeast U.	Math	Nanjing	Liyan Wang	A	P
Southeast U.	Math	Nanjing	Dan He	A	H
Southeast U.	Math	Nanjing	Liyan Wang	B	H
Xi'an Jiaotong–Liverpool U.	E-Finance	Suzhou	Annie Zhu	A	P
Xi'an Jiaotong–Liverpool U.	Financial Math	Suzhou	Ming Ying	A	P
Xi'an Jiaotong–Liverpool U.	Info & Comp.	Suzhou	Liying Liu	A	H
Xi'an Jiaotong–Liverpool U.	Telecomm.	Suzhou	Jingming Guo	A	P
Xuhai C./China U. Mining & Tech.	Math	Xuzhou	Peng Hongjun	A	P
Xuhai C./China U. Mining & Tech.	Math	Xuzhou	Peng Hongjun	A	H
Xuhai C./China U. Mining & Tech.	Physics	Xuzhou	Zhang Wei	A	P
Xuzhou Inst. of Tech.	Math	Xuzhou	Li Subei	A	M
Yangzhou U.	Guangling C.	Yangzhou	Tao Cheng	B	P
Yangzhou U.	InfoEng.	Yangzhou	Weijun Lin	A	P
Yangzhou U.	Math	Yangzhou	Fan Cai	B	H
Jianxi					
Gannan Normal U.	Comp.	GanZhou	Yan Shen Hai	A	H
Gannan Normal U.	Comp.	Ganzhou	Zhan JI Zhou	A	H
Gannan Normal U.	Math	Ganzhou	Xie Xian Hua	B	P
Gannan Normal U.	Math	Ganzhou	Xu Jing Fei	B	P
Jiangxi U. of Finance & Econ.	InfoTech.	Nanchang	Changsheng Hua	B	P
Nanchang Hangkong U.	Appl. Math	Nanchang	Gensheng Qiu	A	H
Nanchang U.	Math	Nanchang	Qingyu Luo	A	P
Nanchang U.	Math	Nanchang	Tao Chen	A	H
Nanchang U.	Math	Nanchang	Liao Chuanrong	A	P
Nanchang U.	Math	Nanchang	Yang Zhao	A	P
Nanchang U.	Math	Nanchang	Chen Yuju	A	H
Nanchang U.	Math	Nanchang	Xianjiu Huang	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Jilin					
Beihua U.	Math	Jilin	Li Tingbin	A	H
Beihua U.	Math	Jilin City	Wei Yuncai	A	P
Beihua U.	Math	Jilin City	Wei Yuncai	A	P
Beihua U.	Math	Jilin City	Chen Zhaojun	A	P
Beihua U.	Math	Jilin City	Zhao Hongwei	A	H
Beihua U.	Math	Jilin City	Yang Yueting	A	P
Beihua U.	Math	Jilin City	Zhang Wei	A	P
Jilin Arch. & Civil Eng. Inst.	Basic Sci.	Changchun	JinLin & Lin Ding	A	P
Jilin U.	Math	Changchun	Huang Qingdao	B	M
Jilin U.	Math	Changchun	Cao Yang	B	H
Jilin U.	Math	Changchun	Yao Xiuling	B	H
Jilin U.	Math	Changchun	Liu Mingji	A	P
Jilin U.	Math	Changchun	Xianrui Lv	B	M
Jilin U.	Math	Changchun	Xianrui Lv	B	P
Jilin U.	Math	Changchun	Shishun Zhao	B	H
Liaoning					
Anshan Normal U.	Math	Anshan	Li Pi Yu	A	P
Anshan Normal U.	Math	Anshan	Zhang Chun	B	P
Anshan Normal U.	Math	Anshan	Liu Hui Min	A	P
Anshan Normal U.	Math	Anshan	Liu Hui Min	B	P
Dalian Fisheries U.	Sci.	Dalian	Zhang Lifeng	A	P
Dalian Jiaotong U.	Sci.	Dalian	Guocan Wang	A	P
Dalian Jiaotong U.	Sci.	Dalian	Guocan Wang	B	P
Dalian Jiaotong U.	Sci.	Dalian	Da-yong Zhou	A	P
Dalian Jiaotong U.	Sci.	Dalian	Da-yong Zhou	B	P
Dalian Maritime U.	Appl. Math	Dalian	Y. Zhang	B	P
Dalian Maritime U.	Appl. Math	Dalian	Y. Zhang	B	H
Dalian Maritime U.	Appl. Math	Dalian	Xinnian Wang	B	H
Dalian Maritime U.	Appl. Math	Dalian	Xinnian Wang	A	P
Dalian Maritime U.	Appl. Math	Dalian	Dong Yu	A	P
Dalian Maritime U.	Math	Dalian	Shuqin Yang	B	P
Dalian Maritime U.	Math	Dalian	Guoyan Chen	B	H
Dalian Maritime U.	Math	Dalian	Naxin Chen	B	M
Dalian Maritime U.	Math	Dalian	Sheng Bi	B	H
Dalian Maritime U.	Math	Dalian	Yun Jie Zhang	A	H
Dalian Nationalities U.	CS	Dalian	Xiaoniu Li	B	H
Dalian Nationalities U.	CS	Dalian	Xiaoniu Li	B	P
Dalian Nationalities U.	CS & Eng.	Dalian	Xiangdong Liu	A	P
Dalian Nationalities U.	CS & Eng.	Dalian	Liming Wang	A	P
Dalian Nationalities U.	CS & Eng.	Dalian	Dejun Yan	A	H
Dalian Nationalities U.	CS & Eng.	Dalian	Liming Wang	B	H
Dalian Nationalities U.	Dean's Office	Dalian	Hengbo Zhang	B	P
Dalian Nationalities U.	Dean's Office	Dalian	Fu Jie	B	H
Dalian Nationalities U.	Dean's Office	Dalian	Rendong Ge	B	P
Dalian Nationalities U.	Dean's Office	Dalian	Rendong Ge	B	P
Dalian Nationalities U.	Dean's Office	Dalian	Yumei Ma	B	H
Dalian Nationalities U.	Dean's Office	Dalian	Yumei Ma	B	P
Dalian Nationalities U.	Innovation Ed.	Dalian	Rixia Bai	A	P
Dalian Nationalities U.	Innovation Ed.	Dalian	Xinwen Chen	B	H
Dalian Nationalities U.	Innovation Ed.	Dalian	Tian Yun	B	H
Dalian Nationalities U.	Innovation Ed.	Dalian	Tian Yun	B	P
Dalian Naval Acad.	Math	Dalian	Feng Jie	A	H
Dalian Naval Acad.	Math	Dalian	Feng Jie	B	H
Dalian Neusoft Inst. of Info	InfoTech & Business Mgmt	Dalian	Sheng Guan	B	H
Dalian Neusoft Inst. of Info	InfoTech & Business Mgmt	Dalian	Qian Wang	B	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Dalian U.	Info. & Eng.	Dalian	He Sun	A	P
Dalian U.	Math	Dalian	Tan Xinxin	A	P
Dalian U.	Math	Dalian	Gang Jiatai	A	P
Dalian U.	Math	Dalian	Gang Jiatai	A	P
Dalian U.	Math	Dalian	Liu Guangzhi	A	P
Dalian U.	Math	Dalian	Zhang Cheng	A	H
Dalian U. of Tech.	Appl. Math	Dalian	Qiuhui Pan	A	P
Dalian U. of Tech.	Appl. Math	Dalian	Liang Zhang	A	P
Dalian U. of Tech.	Appl. Math	Dalian	Liang Zhang	A	P
Dalian U. of Tech.	Appl. Math	Dalian	Qiuhui Pan	B	P
Dalian U. of Tech.	Appl. Math	Dalian	Mingfeng He	B	H
Dalian U. of Tech.	Appl. Math	Dalian	Zhenyu Wu	B	H
Dalian U. of Tech.	Appl. Math	Dalian	Mingfeng He	B	P
Dalian U. of Tech.	Appl. Math	Dalian	Liang Zhang	B	H
Dalian U. of Tech.	Appl. Math	Dalian	Mingfeng He	B	P
Dalian U. of Tech.	City Inst.	Dalian	Xubin Gao	A	H
Dalian U. of Tech.	City Inst.	Dalian	Xubin Gao	A	H
Dalian U. of Tech.	City Inst.	Dalian	Hongzeng Wang	A	P
Dalian U. of Tech.	City Inst.	Dalian	Lina Wan	B	H
Dalian U. of Tech.	Innovation Experiment	Dalian	Wanwu Xi	B	H
Dalian U. of Tech.	Innovation Experiment	Dalian	Lin Feng	B	P
Dalian U. of Tech.	Innovation Experiment	Dalian	Qiuhui Pan	A	P
Dalian U. of Tech.	Software Schl	Dalian	Zhe Li	A	H
Dalian U. of Tech.	Software Schl	Dalian	Zhe Li	A	M
Dalian U. of Tech.	Software Schl	Dalian	Zhe Li	B	M
Dalian U. of Tech.	Software Schl	Dalian	Zhe Li	B	M
Dalian U. of Tech.	Software Schl	Dalian	Zhe Li	B	H
Northeastern U.	Autocontrol	Shenyang	Yunzhou Zhang	B	H
Northeastern U.	Autocontrol	Shenyang	Feng Pan	B	H
Northeastern U.	Comp.	Shenyang	Huilin Liu	B	H
Northeastern U.	InfoSci. & Eng.	Shenyang	Chengdong Wu	A	H
Northeastern U.	InfoSci. & Eng.	Shenyang	Shuying Zhao	A	H
Northeastern U.	Modern Design & Analysis	Shenyang	Xuehong He	A	H
Northeastern U.	Sci.	Shenyang	Ping Sun	A	P
Northeastern U.	Sys. Simulation	Shenyang	JianJiang Cui	A	P
Shenyang Inst. of Aero. Eng.	Electr.	Shenyang	Weifang Liu	A	P
Shenyang Inst. of Aero. Eng.	Electr.	Shenyang	Na Yin	A	P
Shenyang Inst. of Aero. Eng.	Electr.	Shenyang	Lin Li	B	H
Shenyang Inst. of Aero. Eng.	Info & CS	Shenyang	Shiyun Wang	A	H
Shenyang Inst. of Aero. Eng.	Info & CS	Shenyang	Li Wang	A	P
Shenyang Inst. of Aero. Eng.	Info & CS	Shenyang	Yong Jiang	A	P
Shenyang Inst. of Aero. Eng.	North Schl of Sci. & Tech.	Shenyang	Li Lin	A	P
Shenyang Inst. of Aero. Eng.	North Schl of Sci. & Tech.	Shenyang	Wang Xiaoyuan	A	P
Shenyang Inst. of Aero. Eng.	North Schl of Sci. & Tech.	Shenyang	Wang Xiaoyuan	A	P
Shenyang Inst. of Aero. Eng.	North Schl of Sci. & Tech.	Shenyang	Liu Weifang	B	P
Shenyang Inst. of Aero. Eng.	Sci.	Shenyang	Feng Shan	A	P
Shenyang Inst. of Aero. Eng.	Sci.	Shenyang	Limei Zhu	A	H
Shenyang Normal U.	Math & Sys. Sci.	Shenyang	Xiaoyi Li	A	P
Shenyang Normal U.	Math & Sys. Sci.	Shenyang	Yuzhong Liu	A	P
Shenyang Normal U.	Math & Sys. Sci.	Shenyang	Xianji Meng	B	P
Shenyang Pharmaceutical U.	Basic Courses	Shenyang	Rongwu Xiang	A	P
Shenyang Pharmaceutical U.	Basic Courses	Shenyang	Rongwu Xiang	B	P
Shenyang U. of Tech.	Basic Sci.	Shenyang	Chen Yan	A	P
Shenyang U. of Tech.	Basic Sci.	Shenyang	Chen Yan	A	P
Shenyang U. of Tech.	Math	Shenyang	Yan Chen	A	H
Shenyang U. of Tech.	Math	Shenyang	Wang Bo	A	P
Shenyang U. of Tech.	Math	Shenyang	Yan Chen	B	P
Shenyang U. of Tech.	Math	Shenyang	Du Hong Bo	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Shaanxi					
Inst. of Modern Physics	Computational Physics	Xi'an	Jihong Dou	A	H
Inst. of Modern Physics	Computational Physics	Xi'an	Jihong Dou	B	H
Inst. of Visualization	InfoTech.	Xi'an	Liantang Wang	B	H
North U. of China	Math	Taiyuan	Lei YingJie	B	H
North U. of China	Math	TaiYuan	Yang Ming	B	H
North U. of China	Math	Taiyuan	Bi Yong	A	P
North U. of China	Sci.	Taiyuan	Xue Yakui	A	P
Northwest A& F U.	Sci.	Xi'an	Zheng Zheng Ren	A	P
Northwest A& F U.	Sci.	Yangling	Wang Jingmin	A	H
Northwest U.	Ctr Nonlin. Studies	Xi'an	Bo Zhang	B	H
Northwest U.	Ctr Nonlin. Studies	Xi'an	Ming Gou	A	P
Northwest U.	Math	Xi'an	Ruichan He	A	P
Northwest U.	Math	Xi'an	Bo Zhang	B	H
Northwest U.	Physics	Xi'an	YongFeng Xu	B	P
Northwestern Poly. U.	Appl. Chemistry	Xi'an	Sun Zhongkui	A	M
Northwestern Poly. U.	Appl. Chemistry	Xi'an	Tang Yaning	B	H
Northwestern Poly. U.	Appl. Math	Xi'an	Yufeng Nie	B	H
Northwestern Poly. U.	Appl. Math	Xi'an	Zheng Hongchan	B	P
Northwestern Poly. U.	Appl. Physics	Xi'an	Lu Quanyi	B	H
Northwestern Poly. U.	Appl. Physics	Xi'an	Lei Youming	A	M
Northwestern Poly. U.	Natural & Appl. Sci.	Xi'an	Xiao Huayong	B	M
Northwestern Poly. U.	Natural & Appl. Sci.	Xi'an	Zhou Min	B	M
Northwestern Poly. U.	Natural & Appl. Sci.	Xi'an	Yong Xu	B	M
Northwestern Poly. U.	Natural & Appl. Sci.	Xi'an	Zhao Junfeng	A	M
Xi'an Jiaotong U.	Math Teaching & Exp't	Xian	Xiaoe Ruan	A	M
Xi'an Jiaotong U.	Sci. Comp. & Appl. Sftwr	Xi'an	Jian Su	A	H
Xi'an Comm. Inst.	CS	Xi'an	Hong Wang	B	H
Xi'an Comm. Inst.	Electr. Eng.	Xi'an	Jianhang Zhang	B	P
Xi'an Comm. Inst.	Math	Xi'an	Xinshe Qi	B	M
Xi'an Comm. Inst.	Physics	Xi'an	Li Hao	B	H
Xi'an Comm. Inst.	Physics	Xi'an	Dongsheng Yang	A	H
Xi'an Jiaotong U.	Appl. Math	Xi'an	Jing Gao	A	M
Xi'an Jiaotong U.	Sci. Comp. & Appl. Sftwr	Xi'an	Wei Wang	B	H
Xidian U.	Appl. Math	Xi'an	Hailin Feng	A	H
Xidian U.	Comp'l Math	Xi'an	Houjian Tang	B	H
Xidian U.	Ind'l & Appl. Math.	Xi'an	Qiang ZHU	B	M
Xidian U.	Sci.	Xi'an	Guoping Yang	B	M
Xidian U.	Sci.	Xi'an	Jimin Ye	B	M
Taiyuan Inst. of Tech.	Electr. Ass'n Sci. & Tech.	Taiyuan	Fan Xiaoren	B	P
Taiyuan Inst. of Tech.	Electr. Eng.	Taiyuan	Xiao Ren Fan	B	P
Taiyuan U. of Tech.	Math	Taiyuan	Yi-Qiang Wei	B	P
Shandong					
China U. of Petroleum	Math & Comp'l Sci.	Qingdao	Ziting Wang	A	P
China U. of Petroleum	Math & Comp'l Sci.	Qingdao	Ziting Wang	A	M
Liaocheng U.	Math Sci.	Liaocheng	Xianyang Zeng	A	P
Linyi Normal U.	Math	Linyi	Zhaozhong Zhang	A	P
Linyi Normal U.	Math	Linyi	Zhaozhong Zhang	A	P
Linyi Normal U.	Stats	Linyi	Lifeng Gao	B	H
Naval Aero. Eng. Acad.	Machinery	Qingdao	Cao Hua Lin	B	M
QiLu Software C. (SDU)	CS & Tech.	Jinan	Jun Feng Luan	A	P
Qufu Normal U.	Math Sci.	Qufu	Yuzhen Bai	B	P
Shandong U.	CS & Tech.	Jinan	Heji Zhao	A	P
Shandong U.	Econ.	Jinan	Wei Chen	B	H
Shandong U.	Math Finance	Jinan	Yufeng Shi	A	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Shandong U.	Math & Sys. Sci.	Jinan	Bao Dong Liu	B	H
Shandong U.	Math & Sys. Sci.	Jinan	Bao Dong Liu	B	M
Shandong U.	Math & Sys. Sci.	Jinan	Shu Xiang Huang	B	P
Shandong U.	Math & Sys. Sci.	Jinan	Shu Xiang Huang	B	H
Shandong U.	Math & Sys. Sci.	Jinan	Xiao Xia Rong	B	M
Shandong U.	Math & Sys. Sci.	Jinan	Huang Shu Xiang	B	H
Shandong U.	Physics	Jinan	Xiucai Zheng	B	P
Shandong U.	Software	Jinan	Zhang SiHua	B	P
Shandong U.	Software	Jinan	Xiangxu Meng	B	M
Shandong U. at Weihai	Appl. Math	Weihai	YangBing & SongHuiMin	B	M
Shandong U. at Weihai	Appl. Math	Weihai	Cao Zhulou & Xiao Hua	B	M
Shandong U. at Weihai	Appl. Math	Weihai	Zhulou Cao	A	P
Shandong U. at Weihai	InfoSci. & Eng.	Weihai	Huaxiang Zhao	B	P
Shandong U. at Weihai	InfoSci. & Eng.	Weihai	Hua Xiao	A	H
Shandong U. at Weihai	InfoSci. & Eng.	Weihai	Zengchao Mu	B	P
Shandong U. of Sci. & Tech.	Fund. Courses	Qingdao	Fangfang Ma	A	H
Shandong U. of Sci. & Tech.	InfoSci. & Eng.	Qingdao	Xinzeng Wang	B	H
Shandong U. of Sci. & Tech.	InfoSci. & Eng.	Qingdao	Pang Shan Chen	B	H
U. of Jinan	Math	Jinan	Zhenyu Xu	A	P
U. of Jinan	Math	Jinan	Baojian Qiu	A	P
U. of Jinan	Math	Jinan	Honghua Wu	B	H
Shanghai					
Donghua U.	Appl. Math	Shanghai	Yunsheng Lu	A	P
Donghua U.	Info.	Shanghai	Xie Shijie	A	P
Donghua U.	InfoSci. & Tech.	Shanghai	Xianhui Zeng	A	P
Donghua U.	InfoSci. & Tech.	Shanghai	Hongrui Shi	A	P
Donghua U.	Sci.	Shanghai	Liangjian Hu	A	M
East China Normal U.	Finance & Stats	Shanghai	Linyi Qian	A	H
East China Normal U.	Finance & Stats	Shanghai	Yiming Cheng	A	H
East China Normal U.	InfoSci. & Tech.	Shanghai	Ming Li	A	P
East China Normal U.	Math	Shanghai	Yongming Liu	B	M
East China Normal U.	Math	Shanghai	Changhong Lu	B	P
East China U. of Sci. & Tech.	Math	Shanghai	Liu Zhaohui	B	H
East China U. of Sci. & Tech.	Math	Shanghai	Su Chunjie	B	M
East China U. of Sci. & Tech.	Physics	Shanghai	Qin Yan	A	H
East China U. of Sci. & Tech.	Physics	Shanghai	Lu Yuanhong	A	M
Fudan U.	Econ.	Shanghai	Yan Zhang	B	H
Fudan U.	Int'l Finance	Shanghai	Pan Deng	B	H
Fudan U.	Math Sci.	Shanghai	Yuan Cao	B	H
Fudan U.	Math Sci.	Shanghai	Zhijie Cai	B	P
Fudan U.	Physics	Shanghai	Jiping Huang	A	P
Nanyang Model HS	Math	Shanghai	Tuqing Cao	A	P
Nanyang Model HS	Math	Shanghai	Tuqing Cao	B	P
Shanghai Finance U.	Math	Shanghai	Yumei Liang	A	P
Shanghai Finance U.	Math	Shanghai	Rongqiang Che	A	P
Shanghai Finance U.	Math	Shanghai	Xiaobin Li	B	P
Shanghai Finance U.	Math	Shanghai	Keyan Wang	A	P
Shanghai Foreign Lang. Schl	CS	Shanghai	Yue Sun	A	P
Shanghai Foreign Lang. Schl	CS	Shanghai	Yue Sun	A	M
Shanghai Foreign Lang. Schl	Math	Shanghai	Liang Tao	A	P
Shanghai Foreign Lang. Schl	Math	Shanghai	Gan Chen	A	P
Shanghai Foreign Lang. Schl	Math	Shanghai	Gan Chen	A	H
Shanghai Foreign Lang. Schl	Math	Shanghai	Yu Sun	A	H
Shanghai Foreign Lang. Schl	Math	Shanghai	Liang Tao	B	M
Shanghai Foreign Lang. Schl	Math	Shanghai	Yu Sun	B	H
Shanghai Foreign Lang. Schl	Math	Shanghai	Jian Tian	B	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Shanghai Foreign Lang. Schl	Math	Shanghai	Liqun Pan	B	H
Shanghai Foreign Lang. Schl	Math	Shanghai	Liqun Pan	B	H
Shanghai Foreign Lang. Schl	Math	Shanghai	Feng Xu	B	P
Shanghai Foreign Lang. Schl	Math	Shanghai	Feng Xu	B	M
Shanghai Foreign Lang. Schl	Math	Shanghai	Weiping Wang	B	P
Shanghai Foreign Lang. Schl	Math	Shanghai	Weiping Wang	B	H
Shanghai High Schl	Math	Shanghai	Xinyi Yang	A	H
Shanghai Jiading No. 1 Sr. HS	Math	Shanghai	Xilin Xie Yunping Fang	B	H
Shanghai Jiading No. 1 Sr. HS	Math	Shanghai	Xilin Xie Yunping Fang	B	H
Shanghai Jiaotong U.	Math	Shanghai	Baorui Song	A	H
Shanghai Jiaotong U.	Math	Shanghai	Jianguo Huang	A	P
Shanghai Jiaotong U.	Math	Shanghai	Baorui Song	B	P
Shanghai Jiaotong U.	Math	Shanghai	Jianguo Huang	B	P
Shanghai Normal U.	Math & Sci. C.	Shanghai	Jizhou Zhang	A	P
Shanghai Normal U.	Math & Sci. C.	Shanghai	Rongguan Liu	A	P
Shanghai Normal U.	Math & Sci. C.	Shanghai	Xiaobo Zhang	B	H
Shanghai Sino Euro Schl of Tech.	Math	Shanghai	Wei Huang	A	P
Shanghai Sino Euro Schl of Tech.	Math	Shanghai	Wei Huang	B	M
Shanghai Sino Euro Schl of Tech.	Math	Shanghai	Bingwu He	B	P
Shanghai Sino Euro Schl of Tech.	Math	Shanghai	Fuping Tan	B	P
Shanghai U. of Finance & Econ.	Appl. Math	Shanghai	Wenqiang Hao	B	M
Shanghai U. of Finance & Econ.	Appl. Math	Shanghai	Ling Qiu	A	H
Shanghai U. of Finance & Econ.	Appl. Math	Shanghai	Xing Zhang	B	H
Shanghai U. of Finance & Econ.	Appl. Math	Shanghai	Zhenyu Zhang	A	M
Shanghai U. of Finance & Econ.	Appl. Math	Shanghai	Zhenyu Zhang	A	P
Shanghai U. of Finance & Econ.	Econ.	Shanghai	Siheng Cao	A	P
Shanghai U. of Finance & Econ.	Econ.	Shanghai	Yan Sun	A	H
Shanghai U. of Finance & Econ.	Finance	Shanghai	Hao Cha	B	P
Shanghai U. of Finance & Econ.	Stats	Shanghai	Chunjie Wu	A	P
Shanghai U. of Finance & Econ.	Stats	Shanghai	Jialun Du	B	P
Shanghai U.	Math	Shanghai	Yongjian Yang	B	P
Shanghai U.	Math	Shanghai	Yongjian Yang	B	P
Shanghai U.	Math	Shanghai	Donghua Wu	A	P
Shanghai U.	Math	Shanghai	Donghua Wu	B	H
Shanghai U.	Math	Shanghai	Yuandi Wang	A	P
Shanghai U.	Math	Shanghai	Yuandi Wang	A	P
Shanghai Youth Ctr Sci. & Tech. Ed.	Appl. Math	Shanghai	Gan Chen	A	H
Shanghai Youth Ctr Sci. & Tech. Ed.	Sci. Training	Shanghai	Gan Chen	A	P
Shanghai Youth Ctr Sci. & Tech. Ed.	Sci. Training	Shanghai	Gan Chen	B	P
Sydney Inst. of Lang. & Commerce	Math	Shanghai	Youhua He	A	P
Sydney Inst. of Lang. & Commerce	Math	Shanghai	Youhua He	A	P
Tongji U.	Civil Eng.	Shanghai	Jialiang Xiang	A	P
Tongji U.	Math	Shanghai	Jin Liang	A	H
Tongji U.	Math	Shanghai	Hualong Zhang	A	P
Tongji U.	Software	Shanghai	Changshui Huang	B	P
Xuhui Branch/Shanghai Jiaotong U.	Math	Shanghai	Liuqing Xiao	A	H
Xuhui Branch/Shanghai Jiaotong U.	Math	Shanghai	Xiaojun Liu	B	P
Yucai Senior High Schl	Math	Shanghai	Zhenwei Yang	A	H
Yucai Senior High Schl	Math	Shanghai	Xiaodong Zhou	A	P
Yucai Senior High Schl	Math	Shanghai	Zhenwei Yang	B	H
Yucai Senior High Schl	Math	Shanghai	Xiaodong Zhou	B	H
Sichuan					
Chengdu U. of Tech.	InfoMgmnt	Chengdu	Huang Guang Xin	A	H
Chengdu U. of Tech.	InfoMgmnt	Chengdu	Yuan Yong	B	P

INSTITUTION	DEPT.	CITY	ADVISOR		
Sichuan U.	Electr. Eng. & Info.	Chengdu	Yingyi Tan	A	M
Sichuan U.	Electr. Eng. & Info.	Chengdu	Yingyi Tan	B	P
Sichuan U.	Math	Chengdu	unknown	A	P
Sichuan U.	Math	Chengdu	Yonghong Zhao	A	P
Schuan U.	Math	Chengdu	Huilei Han	B	H
Sichuan U.	Math	Chengdu	Hai Niu	B	H
Sichuan U.	Math	Chengdu	Qiong Chen	B	M
Sichuan U.	Math	Chengdu	HuiLei Han	A	P
Southwest Jiaotong U.	Math	Chengdu	Wang Lu	A	P
Southwest Jiaotong U.	Math	Chengdu	Wang Lu	B	H
Southwest Jiaotong U.	Math	Chengdu	Yueliang Xu	B	P
Southwest Jiaotong U.	Math	Chengdu	Yueliang Xu	B	H
Southwest U. of Sci. & Tech.	Sci.	Mianyang	Ke Long Zheng	A	P
Southwestern U. of Finance & Econ.	Econ. Math	Chengdu	Dai Dai	B	M
Southwestern U. of Finance & Econ.	Econ. Math	Chengdu	Dai Dai	B	H
Southwestern U. of Finance & Econ.	Econ. Math	Chengdu	Sun Yunlong	B	H
Southwestern U. of Finance & Econ.	Econ. Math	Chengdu	Chuan Ding	B	H
U. of Elec. Sci. & Tech. of China	Appl. Math	Chengdu	Li Mingqi	A	M
U. of Elec. Sci. & Tech. of China	Appl. Math	Chengdu	He GuoLiang	A	P
U. of Elec. Sci. & Tech. of China	Appl. Math	Chengdu	Li Mingqi	A	P
U. of Electr. Sci. & Tech. of China	Chengdu C.: CS	Chengdu	Qiu Wei	B	P
U. of Elec. Sci. & Tech. of China	Info & CS	Chengdu	Qin Siyi	A	P
Tianjin					
Civil Aviation U. of China	Air Traffic Mgmnt	Tianjin	Zhaoning Zhang	B	H
Civil Aviation U. of China	CS	Tianjin	Xia Feng	B	H
Civil Aviation U. of China	CS & Tech.	Tianjin	Yuxiang Zhang	A	H
Civil Aviation U. of China	CS & Tech.	Tianjin	Chunli Li	B	P
Civil Aviation U. of China	Sci. C.	Tianjin	Zhang Chunxiao	B	H
Civil Aviation U. of China	Sci. C.	Tianjin	Di Shang Chen	B	P
Nankai U.	Automation	Tianjin	Chen Wanyi	B	H
Nankai U.	Econ.	Tianjin	Qi Bin	A	H
Nankai U.	Finance	Tianjin	Fang Wang	B	P
Nankai U.	Informatics & Prob.	Tianjin	Jishou Ruan	A	P
Nankai U.	Informatics & Prob.	Tianjin	Jishou Ruan	A	P
Nankai U.	Info & CS & Tech.	Tianjin	Zhonghua Wu	A	H
Nankai U.	Insurance	Tianjin	Bin Qi	A	P
Nankai U.	Mgmnt Sci. & Eng.	Tianjin	Wenhua Hou	A	M
Nankai U.	Physics	Tianjin	LiYing Zhang	A	H
Nankai U.	Physics	Tianjin	Liying Zhang	B	H
Nankai U.	Software	Tianjin	Wei Zhang	B	P
Nankai U.	Stats	Tianjin	Min-qian Liu	A	H
Tianjin Poly. U.	Sci.	Tianjin	unknown	A	P
Tianjin Poly. U.	Sci.	Tianjin	unknown	B	H
Yunnan					
Chuxiong Normal U.	Math	Chuxiong	Jiade Tang	B	H
Yunnan U.	Comm. Eng'ing	Kunming	Haiyan LI	B	P
Yunnan U.	CS	Kunming	Shunfang Wang	A	P
Yunnan U.	CS	Kunming	Shunfang Wang	A	M
Yunnan U.	Electr. Eng.	Kunming	Haiyan Li	A	P
Yunnan U.	Electr. Eng.	Kunming	Haiyan Li	B	P
Yunnan U.	InfoSci. & Tech.	Kunming	Hong Wei	B	H
Yunnan U.	Stats	Kunming	Bo Zhang	B	P
Yunnan U.	Stats	Kunming	Jie Meng	A	H
Yunnan U.	Stats	Kunming	Jie Meng	A	H

INSTITUTION	DEPT.	CITY	ADVISOR		
Zhejiang					
Hangzhou Dianzi U.	Appl. Physics	Hangzhou	Jianlan Chen	A	H
Hangzhou Dianzi U.	Appl. Physics	Hangzhou	Zhifeng Zhang	B	H
Hangzhou Dianzi U.	Info & Math Sci.	Hangzhou	Wei Li	A	M
Hangzhou Dianzi U.	Info & Math Sci.	Hangzhou	Zheyong Qiu	B	M
Ningbo Inst. of Zhejiang U.	Fund. Courses	Ningbo	Jufeng Wang	A	H
Ningbo Inst. of Zhejiang U.	Fund. Courses	Ningbo	Zhening Li	A	P
Ningbo Inst. of Zhejiang U.	Fund. Courses	Ningbo	Lihui Tu	A	H
Ningbo Inst. of Zhejiang U.	Fund. Courses	Ningbo	Lihui Tu	A	M
Ningbo Inst. of Zhejiang U.	Fund. Courses	Ningbo	Jufeng Wang	B	P
Shaoxing U.	Math	Shaoxing	he jinghui	A	P
Shaoxing U.	Math	Shaoxing	lu jue	A	P
Zhejiang Gongshang U.	Appl. Math	Hangzhou	Zhao Heng	A	P
Zhejiang Gongshang U.	Appl. Math	Hangzhou	Zhao Heng	B	H
Zhejiang Gongshang U.	Info & Comp. Sci.	Hangzhou	Hua Jiukun	A	P
Zhejiang Gongshang U.	Info & Comp. Sci.	Hangzhou	Hua Jiukun	B	H
Zhejiang Gongshang U.	Math	Hangzhou	Ding Zhengzhong	A	P
Zhejiang Gongshang U.	Math	Hangzhou	Ding Zhengzhong	B	H
Zhejiang Normal U.	CS	Jinhua	Qiusheng Qiu	A	P
Zhejiang Normal U.	CS	Jinhua	Zuxiang Sheng	B	P
Zhejiang Normal U.	CS & Tech.	Jinhua	Ying Zhang	A	H
Zhejiang Normal U.	Math	Jinhua	Guolong He	B	P
Zhejiang Normal U.	Math	Jinhua	Yuehua Bu	A	P
Zhejiang Normal U.	Math	Jinhua	Wenqing Bao	B	H
Zhejiang Normal U.	Math	Jinhua	Yuanheng Wang	B	H
Zhejiang Normal U.	Math	Jinhua	Dong Chen	B	H
Zhejiang Sci-Tech U.	Math	Hangzhou	Shi Guosheng	A	P
Zhejiang Sci-Tech U.	Math	Hangzhou	Jiang Yiwei	B	P
Zhejiang Sci-Tech U.	Math	Hangzhou	Luo Hua	B	P
Zhejiang Sci-Tech U.	Psychology	Hangzhou	Hu Jueliang	B	H
Zhejiang U.	Math	Hangzhou	Zhiyi Tan	A	M
Zhejiang U.	Math	Hangzhou	Qifan Yang	A	H
Zhejiang U.	Sci.	Hangzhou	Shengyi Cai	A	P
Zhejiang U.	Sci.	Hangzhou	Shengyi Cai	B	P
Zhejiang U.	Sci.	Hangzhou	Yong Wu	B	M
Zhejiang U. City C.	Info & CS	Hangzhou	Xusheng Kang	A	H
Zhejiang U. City C.	Info & CS	Hangzhou	Gui Wang	A	P
Zhejiang U. City C.	Info & CS	Hangzhou	Xusheng Kang	B	P
Zhejiang U. City C.	Info & CS	Hangzhou	Gui Wang	B	M
Zhejiang U. of Finance & Econ.	Math & Stats	Hangzhou	Ji Luo	B	M
Zhejiang U. of Finance & Econ.	Math & Stats	Hangzhou	Ji Luo	B	H
Zhejiang U. of Finance & Econ.	Math & Stats	Hangzhou	Fulai Wang	B	H
Zhejiang U. of Sci. & Tech.	Math	Hangzhou	Yongzhen Zhu	B	H
Zhejiang U. of Tech.	Jianxing C.	Hangzhou	Shiming Wang	A	P
Zhejiang U. of Tech.	Jianxing C.	Hangzhou	Shiming Wang	A	P
Zhejiang U. of Tech.	Jianxing C.	Hangzhou	Wenxin Zhuo	A	P
Zhejiang U. of Tech.	Jianxing C.	Hangzhou	Wenxin Zhuo	B	H

INSTITUTION	DEPT.	CITY	ADVISOR		
FINLAND					
Helsingin matematiikkalukio	Math	Helsinki	Esa I. Lappi	B	H
Helsinki U. of Tech.	Math/Sys. Anal.	Helsinki	Kenrick Bingham	A	H
Helsinki Upper Sec. Schl	Math	Helsinki	Ville Tilvis	A	H
Helsinki Upper Sec. Schl	Math	Helsinki	Ville Tilvis	A	H
Päivölä C. of Math	Math	Tarttila	Janne Puustelli	B	M
Päivölä C. of Math	Math	Tarttila	Janne Puustelli	B	H
Päivölä C. of Math	Math	Tarttila	Merikki Lappi	B	P
U. of Helsinki	Math & Stats	Helsinki	Petri Ola	A	P
GERMANY					
Jacobs U.	Eng. & Sci.	Bremen	Marcel Oliver	B	H
HONG KONG					
Chinese U. of Hong Kong	CS & Eng.	Shatin	Fung Yu Young	B	P
Chinese U. of Hong Kong	Sys. Eng.	Shatin	Nan Chen	A	H
Hong Kong Baptist U.	Math	Kowloon	Wai Chee Shiu	A	P
Hong Kong Baptist U.	Math	Kowloon	Chong Sze Tong	B	P
Hong Kong U. of Sci. & Tech.	Math	Hong Kong	Min Yan	B	H
INDONESIA					
Institut Teknologi Bandung	Math	Bandung	Agus Yodl Gunawan	A	H
Institut Teknologi Bandung	Math	Bandung	Rieske Hadianti	A	H
IRELAND					
U. C. Cork	Appl. Math	Cork	Liya A. Zhornitskaya	A	M
U. C. Cork	Math	Cork	Benjamin W. McKay	A	M
U. C. Cork	Stats	Cork	Supratik Roy	A	H
National U. of Ireland	Math	Galway	Niall Madden	A	M
National U. of Ireland	Math	Galway	Niall Madden	A	H
National U. of Ireland	Math'l Physics	Galway	Petri T. Piiroinen	A	M
National U. of Ireland	Math'l Physics	Galway	Petri T. Piiroinen	B	H
JAMAICA					
U. of Tech.	Chem. Eng.	Kingston	Nilza G. Justiz-Smith	A	M
U. of Tech.	Chem. Eng.	Kingston	Nilza G. Justiz-Smith	B	H
KOREA					
Korea Adv. Inst. of Sci. & Tech.	Math Sci.	Daejeon	Yong-Jung Kim	B	M
Korea Adv. Inst. of Sci. & Tech.	Math Sci.	Daejeon	Yong-Jung Kim	B	H
MEXICO					
U. Autónoma de Yucatán	Math	Mérida	Eric J. Avila-Vales	B	H
SINGAPORE					
National U. of Singapore	Math	Singapore	Gongyun Zhao	A	H
National U. of Singapore	Math	Singapore	Karthik Natarajan	B	H
SOUTH AFRICA					
Stellenbosch U.	Math Sci.	Stellenbosch	Jacob A.C. Weideman	A	H
Stellenbosch U.	Math Sci.	Stellenbosch	Jacob A.C. Weideman	A	P
UNITED KINGDOM					
Oxford U.	Math	Oxford	Jeffrey H. Giansiracusa	B	M
Oxford U.	Math	Oxford	Jeffrey H. Giansiracusa	B	M

The Impending Effects of North Polar Ice Cap Melt

Benjamin Coate
Nelson Gross
Megan Longo

College of Idaho
Caldwell, ID

Advisor: Michael P. Hitchman

Abstract

Because of rising global temperatures, the study of North Polar ice melt has become increasingly important.

- How will the rise in global temperatures affect the melting polar ice caps and the level of the world's oceans?
- Given the resulting increase in sea level, what problems should metropolitan areas in a region such as Florida expect in the next 50 years?

We develop a model to answer these questions.

Sea level will not be affected by melting of the floating sea ice that makes up most of the North Polar ice cap, but it will be significantly affected by the melting of freshwater land ice found primarily on Greenland, Canada, and Alaska. Our model begins with the current depletion rate of this freshwater land ice and takes into account

- the exponential increase in melting rate due to rising global temperatures,
- the relative land / ocean ratios of the Northern and Southern Hemispheres,
- the percentage of freshwater land ice melt that stays in the Northern Hemisphere due to ocean currents, and
- thermal expansions of the ocean due to increased temperatures on the top layer.

The UMAP Journal 29 (3) (2008) 223–224.

We construct best- and worst-case scenarios. We find that in the next 50 years, the relative sea level will rise 12 cm to 36 cm.

To illustrate the consequences of such a rise, we consider four Florida coastal cities: Key West, Miami, Daytona Beach, and Tampa. The problems that will arise in many areas are

- the loss of shoreline property,
- a rise of the water table,
- instability of structures,
- overflowing sewers,
- increased flooding in times of tropical storms, and
- drainage problems.

Key West and Miami are the most susceptible to all of these effects. While Daytona Beach and Tampa are relatively safe from catastrophic events, they will still experience several of these problems to a lesser degree.

The effects of the impending rise in sea level are potentially devastating; however, there are steps and precautions to take to prevent and minimize destruction. We suggest several ways for Florida to combat the effects of rising sea levels: public awareness, new construction codes, and preparedness for natural disasters.

The text of this paper appears on pp. 237–247.

A Convenient Truth: Forecasting Sea Level Rise

Jason Chen
Brian Choi
Joonhahn Cho

Duke University
Durham, NC

Advisor: Scott McKinley

Abstract

Greenhouse-gas emissions have produced global warming, including melting in the Greenland Ice Sheet (GIS), resulting in sea-level rise, a trend that could devastate coastal regions. A model is needed to quantify effects for policy assessments.

We present a model that predicts sea-level trends over a 50-year period, based on mass balance and thermal expansion acting on a simplified ice-sheet geometry. Mass balance is represented using the heat equation with Neumann conditions and sublimation rate equations. Thermal expansion is estimated by an empirically-derived equation relating volume expansion to temperature increase. Thus, the only exogenous variables are time and temperature.

We apply the model to varying scenarios of greenhouse-gas-concentration forcings. We solve the equations numerically to yield sea-level increase projections. We then project the effects on Florida, as modeled from USGS geospatial elevation data and metropolitan population data.

The results of our model agree well with past measurements, strongly supporting its validity. The strong linear trend shown by our scenarios indicates both insensitivity to errors in inputs and robustness with respect to the temperature function.

Based on our model, we provide a cost-benefit analysis showing that small investments in protective technology could spare coastal regions from flooding. Finally, the predictions indicate that reductions in greenhouse-gas emissions are necessary to prevent long-term sea-level-rise disasters.

The text of this paper appears on pp. 249–265.

The UMAP Journal 29 (3) (2008) 225.

Fighting the Waves: The Effect of North Polar Ice Cap Melt on Florida

Amy M. Evans
Tracy L. Stepien

University at Buffalo, The State University of New York
Buffalo, NY

Advisor: John Ringland

Abstract

A consequence of global warming that directly impacts U.S. citizens is the threat of rising sea levels due to melting of the North Polar ice cap. One of the many states in danger of losing coastal land is Florida. Its low elevations and numerous sandy beaches will lead to higher erosion rates as sea levels increase. The direct effect on sea level of only the North Polar ice cap melting would be minimal, yet the indirect effects of causing other bodies of ice to melt would be crucial. We model individually the contributions of various ice masses to rises in sea level, using ordinary differential equations to predict the rate at which changes would occur.

For small ice caps and glaciers, we propose a model based on global mean temperature. Relaxation time and melt sensitivity to temperature change are included in the model. Our model of the Greenland and Antarctica ice sheets incorporates ice mass area, volume, accumulation, and loss rates. Thermal expansion of water also influences sea level, so we include this too. Summing all the contributions, sea levels could rise 11–27 cm in the next half-century.

A rise in sea level of one unit is equivalent to a horizontal loss of coastline of 100 units. We investigate how much coastal land would be lost, by analyzing relief and topographic maps. By 2058, in the worst-case scenario, there is the potential to lose almost 27 m of land. Florida would lose most of its smaller islands and sandy beaches. Moreover, the ports of most major cities, with the exception of Miami, would sustain some damage.

Predictions from the Intergovernmental Panel on Climate Change (IPCC) and from the U.S. Environmental Protection Agency (EPA) and simulations

---

The UMAP Journal 29 (3) (2008) 226–227.

from the Global Land One-km Base Elevation (GLOBE) digital elevation model (DEM) match our results and validate our models.

While the EPA and the Florida state government have begun to implement plans of action, further measures need to be put into place, because there will be a visible sea-level rise of 3–13 cm in only 10 years (2018).

The text of this paper appears on pp. 267–284.

Erosion in Florida: A Shore Thing

Matt Thies
Bob Liu
Zachary W. Ulissi

University of Delaware
Newark, DE

Advisor: Louis Frank Rossi

Abstract

Rising sea levels and beach erosion are an increasingly important problems for coastal Florida. We model this dynamic behavior in four discrete stages: global temperature, global sea level, equilibrium beach profiles, and applications to Miami and Daytona Beach. We use the Intergovernmental Panel on Climate Change (IPCC) temperature models to establish predictions through 2050. We then adapt models of Arctic melting to identify a model for global sea level. This model predicts a likely increase of 15 cm within 50 years.

We then model the erosion of the Daytona and Miami beaches to identify beach recession over the next 50 years. The model predicts likely recessions of 66 m in Daytona and 72 m in Miami by 2050, roughly equal to a full city block in both cases. Regions of Miami are also deemed to be susceptible to flooding from these changes. Without significant attention to future solutions as outlined, large-scale erosion will occur. These results are strongly dependent on the behavior of the climate over this time period, as we verify by testing several models.

The text of this paper appears on pp. 285–300.

A Difficulty Metric and Puzzle Generator for Sudoku

Christopher Chang
Zhou Fan
Yi Sun

Harvard University
Cambridge, MA

Advisor: Clifford H. Taubes

Abstract

We present here a novel solution to creating and rating the difficulty of Sudoku puzzles. We frame Sudoku as a search problem and use the expected search time to determine the difficulty of various strategies. Our method is relatively independent from external views on the relative difficulties of strategies.

Validating our metric with a sample of 800 puzzles rated externally into eight gradations of difficulty, we found a Goodman-Kruskal γ coefficient of 0.82, indicating significant correlation [Goodman and Kruskal 1954]. An independent evaluation of 1,000 typical puzzles produced a difficulty distribution similar to the distribution of solve times empirically created by millions of users at `http://www.websudoku.com`.

Based upon this difficulty metric, we created two separate puzzle generators. One generates mostly easy to medium puzzles; when run with four difficulty levels, it creates puzzles (or *boards*) of those levels in 0.25, 3.1, 4.7, and 30 min. The other puzzle generator modifies difficult boards to create boards of similar difficulty; when tested on a board of difficulty 8,122, it created 20 boards with average difficulty 7,111 in 3 min.

The text of this paper appears on pp. 305–326.

The UMAP Journal 29 (3) (2008) 229.

Taking the Mystery Out of Sudoku Difficulty: An Oracular Model

Sarah Fletcher
Frederick Johnson
David R. Morrison

Harvey Mudd College
Claremont, CA

Advisor: Jon Jacobsen

Abstract

In the last few years, the 9-by-9 puzzle grid known as Sudoku has gone from being a popular Japanese puzzle to a global craze. As its popularity has grown, so has the demand for harder puzzles whose difficulty level has been rated accurately.

We devise a new metric for gauging the difficulty of a Sudoku puzzle. We use an oracle to model the growing variety of techniques prevalent in the Sudoku community. This approach allows our metric to reflect the difficulty of the puzzle itself rather than the difficulty with respect to some particular set of techniques or some perception of the hierarchy of the techniques. Our metric assigns a value in the range $[0, 1]$ to a puzzle.

We also develop an algorithm that generates puzzles with unique solutions across the full range of difficulty. While it does not produce puzzles of a specified difficulty on demand, it produces the various difficulty levels frequently enough that, as long as the desired score range is not too narrow, it is reasonable simply to generate puzzles until one of the desired difficulty is obtained. Our algorithm has exponential running time, necessitated by the fact that it solves the puzzle it is generating to check for uniqueness. However, we apply an algorithm known as Dancing Links to produce a reasonable runtime in all practical cases.

The text of this paper appears on pp. 327–341.

Difficulty-Driven Sudoku Puzzle Generation

Martin Hunt
Christopher Pong
George Tucker

Harvey Mudd College
Claremont, CA

Advisor: Zach Dodds

Abstract

Many existing Sudoku puzzle generators create puzzles randomly by starting with either a blank grid or a filled-in grid. To generate a puzzle of a desired difficulty level, puzzles are made, graded, and discarded until one meets the required difficulty level, as evaluated by a predetermined difficulty metric. The efficiency of this process relies on randomness to span all difficulty levels.

We describe generation and evaluation methods that accurately model human Sudoku-playing. Instead of a completely random puzzle generator, we propose a new algorithm, *Difficulty-Driven Generation*, that guides the generation process by adding cells to an empty grid that maintain the desired difficulty.

We encapsulate the most difficult technique required to solve the puzzle and number of available moves at any given time into a *rounds* metric. A round is a single stage in the puzzle-solving process, consisting of a single high-level move or a maximal series of low-level moves. Our metric counts the numbers of each type of rounds.

Implementing our generator algorithm requires using an existing metric, which assigns a puzzle a difficulty corresponding to the most difficult technique required to solve it. We propose using our rounds metric as a method to further simplify our generator.

The text of this paper appears on pp. 343–362.

The UMAP Journal 29 (3) (2008) 231.

Ease and Toil: Analyzing Sudoku

Seth B. Chadwick
Rachel M. Krieg
Christopher E. Granade

University of Alaska Fairbanks
Fairbanks, AK

Advisor: Orion S. Lawlor

Abstract

Sudoku is a logic puzzle in which the numbers 1 through 9 are arranged in a 9×9 matrix, subject to the constraint that there are no repeated numbers in any row, column, or designated 3×3 square.

In addition to being entertaining, Sudoku promises insight into computer science and mathematical modeling. Since Sudoku-solving is an NP-complete problem, algorithms to generate and solve puzzles may offer new approaches to a whole class of computational problems. Moreover, Sudoku construction is essentially an optimization problem.

We propose an algorithm to construct unique Sudoku puzzles with four levels of difficulty. We attempt to minimize the complexity of the algorithm while still maintaining separate difficulty levels and guaranteeing unique solutions.

To accomplish our objectives, we develop metrics to analyze the difficulty of a puzzle. By applying our metrics to published control puzzles with specified difficulty levels, we develop classification functions. We use the functions to ensure that our algorithm generates puzzles with difficulty levels analogous to those published. We also seek to measure and reduce the computational complexity of the generation and metric measurement algorithms.

Finally, we analyze and reduce the complexity involved in generating puzzles while maintaining the ability to choose the difficulty level of the puzzles generated. To do so, we implement a profiler and perform statistical hypothesis-testing to streamline the algorithm.

The text of this paper appears on pp. 363–379.

A Crisis to Rival Global Warming: Sudoku Puzzle Generation

Jake Linenthal
Alex Twist
Andy Zimmer

University of Puget Sound
Tacoma, WA

Advisor: Michael Z. Spivey

Abstract

We model solution techniques and their application by an average Sudoku player. A simulation based on our model determines a likely solution path for the player. We define a metric that is linear in the length of this path and proportional to a measure of average difficulty of the techniques used. We use this metric to define seven difficulty levels for Sudoku puzzles.

We confirm the accuracy and consistency of our metric by considering rated puzzles from *USA Today* and `Sudoku.org.uk`. Our metric is superior to a metric defined by the count of initial hints, as well to a metric that measures the constraints placed on the puzzle by the initial hints.

We develop an algorithm that produces puzzles with unique solutions with varying numbers of initial hints. Our puzzle generator starts with a random solved Sudoku board, removes a number of hints, and employs a fast solver to ensure a unique solution. We improve the efficiency of puzzle generation by reducing the expected number of calls to the solver. On average, our generation algorithm performs more than twice as fast as the baseline generation algorithm.

We apply our metric to generated puzzles until one matches the desired difficulty level. Since certain initial board configurations result in puzzles that are more difficult on average than a random configuration, we modify our generation algorithm to restrict the initial configuration of the board, thereby reducing the amount of time required to generate a puzzle of a certain difficulty.

The UMAP Journal 29 (3) (2008) 233–234.

[EDITOR'S NOTE: This Meritorious paper won the Ben Fusaro Award for the Sudoku Problem. The full text of the paper does not appear in this issue of the *Journal*.]

Cracking the Sudoku: A Deterministic Approach

David Martin
Erica Cross
Matt Alexander

Youngstown State University
Youngstown, OH

Advisor: George T. Yates

Summary

We formulate a Sudoku-puzzle-solving algorithm that implements a hierarchy of four simple logical rules commonly used by humans. The difficulty of a puzzle is determined by recording the sophistication and relative frequency of the methods required to solve it. Four difficulty levels are established for a puzzle, each pertaining to a range of numerical values returned by the solving function.

Like humans, the program begins solving each puzzle with the lowest level of logic necessary. When all lower methods have been exhausted, the next echelon of logic is implemented. After each step, the program returns to the lowest level of logic. The procedure loops until either the puzzle is completely solved or the techniques of the program are insufficient to make further progress.

The construction of a Sudoku puzzle begins with the generation of a solution by means of a random-number-based function. Working backwards from the solution, numbers are removed one by one, at random, until one of several conditions, such as a minimum difficulty rating and a minimum number of empty squares, has been met. Following each change in the grid, the difficulty is evaluated. If the program cannot solve the current puzzle, then either there is not a unique solution, or the solution is beyond the grasp of the methods of the solver. In either case, the last solvable puzzle is restored and the process continues.

Uniqueness is guaranteed because the algorithm never guesses. If there

The UMAP Journal 29 (3) (2008) 235–236.

is not sufficient information to draw further conclusions—for example, an arbitrary choice must be made (which must invariably occur for a puzzle with multiple solutions)—the solver simply stops. For obvious reasons, puzzles lacking a unique solution are undesirable. Since the logical techniques of the program enable it to solve most commercial puzzles (for example, most "evil" puzzles from Greenspan and Lee [2008]), we assume that demand for puzzles requiring logic beyond the current grasp of the solver is low. Therefore, there is no need to distinguish between puzzles requiring very advanced logic and those lacking unique solutions.

The text of this paper appears on pp. 381–394.

Pp. 237–248 can be found on the *Tools for Teaching 2008* CD-ROM.

The Impending Effects of North Polar Ice Cap Melt

Benjamin Coate
Nelson Gross
Megan Longo

College of Idaho
Caldwell, ID

Advisor: Michael P. Hitchman

Abstract

Because of rising global temperatures, the study of North Polar ice melt has become increasingly important.

- How will the rise in global temperatures affect the melting polar ice caps and the level of the world's oceans?
- Given the resulting increase in sea level, what problems should metropolitan areas in a region such as Florida expect in the next 50 years?

We develop a model to answer these questions.

Sea level will not be affected by melting of the floating sea ice that makes up most of the North Polar ice cap, but it will be significantly affected by the melting of freshwater land ice found primarily on Greenland, Canada, and Alaska. Our model begins with the current depletion rate of this freshwater land ice and takes into account

- the exponential increase in melting rate due to rising global temperatures,
- the relative land / ocean ratios of the Northern and Southern Hemispheres,
- the percentage of freshwater land ice melt that stays in the Northern Hemisphere due to ocean currents, and
- thermal expansions of the ocean due to increased temperatures on the top layer.

The UMAP Journal 29 (3) (2008) 237–247.

We construct best- and worst-case scenarios. We find that in the next 50 years, the relative sea level will rise 12 cm to 36 cm.

To illustrate the consequences of such a rise, we consider four Florida coastal cities: Key West, Miami, Daytona Beach, and Tampa. The problems that will arise in many areas are

- the loss of shoreline property,
- a rise of the water table,
- instability of structures,
- overflowing sewers,
- increased flooding in times of tropical storms, and
- drainage problems.

Key West and Miami are the most susceptible to all of these effects. While Daytona Beach and Tampa are relatively safe from catastrophic events, they will still experience several of these problems to a lesser degree.

The effects of the impending rise in sea level are potentially devastating; however, there are steps and precautions to take to prevent and minimize destruction. We suggest several ways for Florida to combat the effects of rising sea levels: public awareness, new construction codes, and preparedness for natural disasters.

Introduction

We consider for the next 50 years the effects on the Florida coast of melting of the North Polar ice cap, with particular attention to the cities noted. This question can be broken down into two more-detailed questions:

- What is the melting rate, and its effects on sea level?
- How will the rising water affect the Florida cities, and what can they do to counteract and prepare?

Our models use the geophysical data in **Table 1** and the elevations of cities in **Table 2**.

Table 1.
Geophysical data.

Entity	Value	Unit
Total volume of ice caps	2.422×10^7	km^3
Surface area of world's oceans	3.611×10^8	km^2
Surface area of ice on Greenland	1.756×10^6	km^2
Volume of ice on Greenland	2.624×10^6	km^3

Table 2.
Elevations of Florida cities.

City	Average elevation (m)	Maximum elevation (m)
Key West	2.44	5.49
Miami	2.13	12.19
Daytona Beach	2.74	10.36

Preliminary Discussion of Polar Ice

There are two types of polar ice:

- frozen sea ice, as in the North Polar ice cap; and
- freshwater land ice, primarily in Greenland, Canada, and Alaska.

Frozen Seawater

Melting of frozen seawater has little effect because it is already floating. According to the Archimedean principle of buoyancy, an object immersed in a fluid is buoyed up by a force equal to the weight of the fluid that is displaced by the object. About 10% of sea ice is above water, since the densities of seawater and solid ice are 1026 kg/m^3 and 919 kg/m^3. So, if this ice were to melt, 10% of the original volume would be added as water to the ocean. There would be little effect on relative sea level if the entire North Polar ice cap were to melt.

The Ice Caps

Although the melting of the ice caps will not cause a significant rise in the sea level, several problems will indeed arise if they disappear.

- Initially there will be a small decrease in the average temperature of the oceans in the Northern Hemisphere.
- The ice caps reflect a great deal of sunlight, which in turn helps to reduce temperature in that region. When that ice is gone, additional energy will be absorbed and over time we will see a significant increase in global temperatures, both in the oceans and the air.

Freshwater Ice on Land

When freshwater ice on land melts and runs into the ocean, that water is added permanently to the ocean. The total volume of the ice on Greenland alone is 2.624×10^6 km^3. If all of this ice were to melt and add to the ocean (not taking into account possible shifting/depressing of the ocean floor or added surface area of the ocean), the average global sea level would rise 6.7 m—just from the ice on Greenland.

Our question now becomes:

How will the melting of freshwater land ice affect the relative level of the world's oceans over the next 50 years?

Model 1: Constant Temperature

Predicted Increase in Sea Level

To model the effects of ice-cap melt on Florida, we develop a model that provides a quick estimate of expected flooding. We assume:

- No increase in the rate of current ice-melt.
- Uniform distribution of the water from the ice melt throughout the world's oceans.
- No significant change in global temperatures and weather conditions.

We use the notation:

$\%$ Melt = percentage of land ice melting per decade

V_I = current volume of land ice in Northern Hemisphere

$C_{\mathrm{I}\to\mathrm{W}}$ = conversion factor volume of ice to volume of water = 0.919

$\mathrm{SA}_{\mathrm{WO}}$ = surface area of the world's oceans = $3.611 \times 10^8 \mathrm{km}^2$

For a given decade, our equation becomes

$$\text{Increase in ocean sea level} = \frac{\%\ \text{Melt} \times V_I \times C_{\mathrm{I}\to\mathrm{W}}}{\mathrm{SA}_{\mathrm{WO}}}.$$

Data from satellite images show a decrease in the Greenland ice sheet of 239 km^3 per year [Cockrell School of Engineering 2006]. Extrapolating linearly, after 50 years we get an increase in sea level of 3.3 cm.

We must also take into account the contributions of smaller land ice masses in Alaska and Canada, whose melting is contributing to the ocean sea level rises of 0.025 cm and 0.007 cm per year [Abdalati 2005]. Extrapolating linearly over 50 years, the total from the two is 1.6 cm, giving a total increase in sea level of 4.9 cm $\approx$ 5 cm $\approx$ 2 in. by 2058.

Effects on Major Metropolitan Areas of Florida

Even after 50 years there will not be any significant effect on the coastal regions of Florida, since all of these coastal cites are at least 2 m above sea level on average. There will, however, be correspondingly higher flooding during storms and hurricanes.

Unfortunately, these results are based on simple assumptions that do not account for several factors that play a role in the rising sea level. We move on to a second model, which gets us closer to a realistic value.

Model 2: Variable-Temperature Model

Our next model takes into account the effect of a variable temperature on the melting of the polar ice caps. Our basic model assumes constant overall temperature in the polar regions, which will not be the case.

Predicted Increase in Temperature

The average global temperature rose about 1° C in the 20th century, but over the last 25 years the rate has inreased to approximately °C per century [National Oceanic and Atmospheric Administration (NOAA) 2008]. In addition, much of the added heat and carbon dioxide gas will be absorbed by the ocean, which will increase its temperature.

Consequently, scientists project an increase in the world's temperature by 0.7 to 2.9°C over the next 50 years [Ekwurzel 2007]. An increase in overall temperature will cause freshwater land ice to melt faster, which in turn will cause the ocean to rise higher than predicted by the basic model.

We examine how an increase of 0.7 to 2.9°C over the next 50 years will affect sea level.

Model Results

We consider best- and worst-case scenarios. Again, we linearize; for example, for the best-case scenario of 0.7°C over 50 years, we assume an increase of 0.14°C per decade.

Best-Case Scenario: Increase of 0.7°C Over 50 Years

The ice caps will absorb more heat and melt more rapidly. We calculate sea-level rise at 10-year intervals.

The extra heat Q_x absorbed can be quantified as

$$Q_x = msT,$$

where

x is the duration (yrs),

m is mass of the ice cap (g),

s is the specific heat of ice (2.092 J/g-°C), and

T is the change in overall global temperature (°C).

We find

$$Q_{50} = 4.85 \times 10^{18} \text{ kJ}.$$

To determine how much extra ice will melt in the freshwater land-ice regions due to an overall increase in 0.7°C, we divide the amount of heat absorbed by the ice by the specific latent heat of fusion for water, 334 kJ/kg at 0°C, getting a mass of ice melted of 1.45×10^{16} kg.

Since water has a mass of 1,000 kg per cubic meter, the total volume of water added to the ocean is 1.45×10^{13} m^3. Dividing by the surface area of the ocean gives a corresponding sea-level rise of 4.0 cm.

This volume is in addition to the height of 4.9 cm calculated in the steady-temperature Model 1. Thus, in our best-case scenario, in 50 years the ocean will rise about 9 cm.

Worst-Case Scenario: Increase of 2.9°C Over 50 Years

Using the same equations, we find in our worst-case scenario that in 50 years the ocean will rise about 21 cm.

Model 3: Ocean Volume under Warming

The previous two models determined the total volume of water to be added to the world's oceans as a result of the melting of freshwater land ice. However, they do not take into account the relative surface areas of the oceans of the Northern Hemisphere and the Southern Hemisphere. The difference in the ratios of land area to ocean area in the two hemispheres is quite striking and gives a way of improving our model of water distribution.

Northern Hemisphere Ocean Surface Area

Approximately 44% of the world's ocean surface area is located in the Northern Hemisphere and 56% in the Southern Hemisphere [Pidwirny 2008]. The surface area of the ocean in the Northern Hemisphere is 1.58×10^8 km^2.

Percentage of Ice Melt Staying in the Northern Hemisphere

Similar melting freshwater land-ice is occurring in southern regions. So, we have water pouring down from the North Pole and water rushing up from the South Pole. There is very little information regarding flow rates and distributions of water throughout the world's oceans. Since most of the ice melt is added to the top layer of the ocean, that water will be subject to the major ocean currents, under which water in the Northern Hemisphere mainly stays in the north. For the sake of argument, we assume conservatively that just half of the melted freshwater land ice from the north stays in the Northern Hemisphere.

Expanding Volume Due to Increasing Ocean Temperatures

Several factors contribute to warming the ocean:

- The rising air temperature too will warm the ocean.
- As the polar ice caps melt, they will reflect less and less sunlight, meaning that the ocean will absorb a great deal of that heat.
- Progressively higher levels of carbon dioxide will be forced into the ocean.

In the ocean below 215 m, the pressure and lack of sunlight counteract increases in temperature. The water in the top 215 m of the ocean, however, will warm and expand in volume. Water at that temperature (15°C) has a coefficient of thermal expansion of 2.00×10^{-4} K^{-1}. We estimate the water level rise for the best and worst-case scenarios via:

$$V_{\text{change}} = V_{\text{start}} B T_{\text{change}},$$

where

- V_{start} = initial volume,
- V_{change} = change in volume,
- B = the thermal expansion factor (2.00×10^{-4} K^{-1}), and
- T_{change} = the change in temperature.

By dividing out the surface area of both volumes (roughly equal), we find a change in depth: 2 cm in the best-case scenario, and 12.5 cm in the worst case, after 50 years.

Putting It All Together

Figure 1 shows the results from Model 3. After 50 years, the sea level surrounding Florida will rise between 12 and 36 cm.

Effect on Florida

While the ocean-level rise surrounding each of the four cities will be comparable, there will be differential effects due to topography.

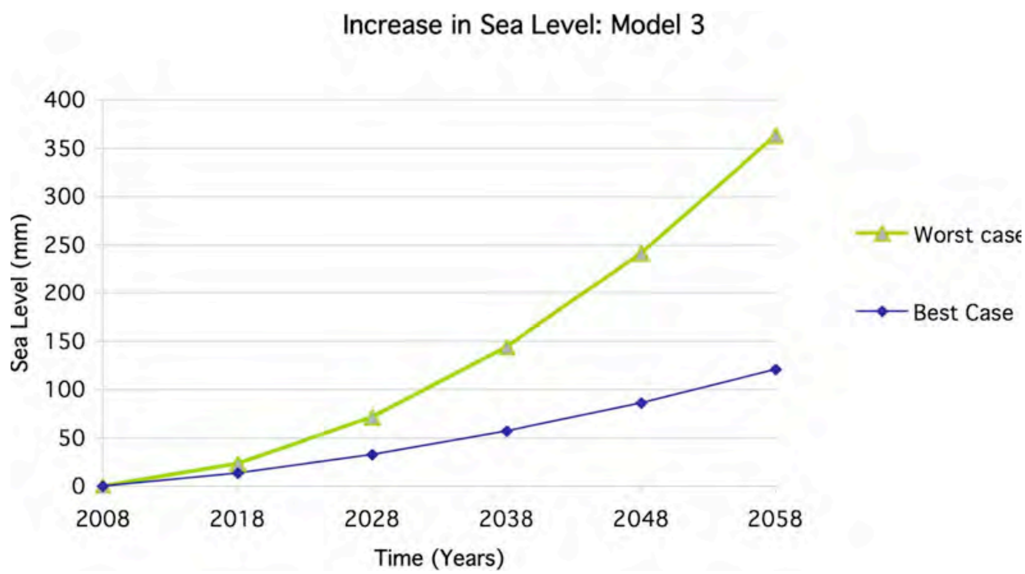


Figure 1. Results from Model 3.

Key West

Key West is the lowest in elevation of our four chosen coastal cities, with an average elevation of 2.44 m. After 50 years, the sea level will rise between 12 cm (4.7 in.) and 36 cm (14.3 in.).

This city is by far the most susceptible to flooding. When the sea level rises, there will be a proportional rise in the water table of the city. So, not only will the city begin to flood at higher elevations than it does currently, but it will also be harder to drain water after storms. In addition, there will be problems with overflowing sewers.

Based on our projections in Model 3, 75% of Key West will be at serious risk for flooding in about 50 years, including the airport. Key West needs to consider how to prevent water from entering the airport area or even start thinking about building a new airport at a higher elevation. [This is of particular importance considering the flooding of Key West in the summer of 2008.]

Miami

Miami will experience problems similar to those of Key West. Under the range of the scenarios, there will be a small loss of beachfront land and some minor flooding along the Miami River. Again, there will be possible problems with overflowing sewers and drainage due to the raised water table. However, one of the biggest problems might arise during a significant storm such as a hurricane. With the added height of the ocean and the low elevation of the Miami downtown area, the city could experience long-lasting floods of up to 36 cm where flooding is now currently minimal.

In 50 years, many buildings could be far too close to the ocean for comfort, and their structural integrity might be compromised.

Daytona Beach

Daytona Beach will experience some loss of shoreline property and be slightly more susceptible to flooding in low-lying areas. In addition, flood risks will be more severe in times of tropical storms and hurricanes. However, since there is a sharp increase in the elevation as one goes inland, flooding will be minimal and city drainage will remain relatively normal.

Tampa

Tampa will experience very little change from its current situation, since its lowest-lying regions are above 8 m. However, Tampa needs to be prepared for additional flooding and possible drainage problems.

General Recommendations for Coastal Florida

- **Limit coastal erosion.** The more erosion, the more beachfront property will be lost.
- **Monitor the water table.** As the sea level rises, so will the water table, which affects foundations of buildings and sewers. It would be advisable to restrict building construction within a set distance of the coast.
- **Prepare for flooding.** Higher sea level will produce greater flooding in storms. Cities should prepare evacuation and emergency plans.
- **Use government information resources.** When it comes to predicting whether or not one's particular town is in danger, there is an excellent online source for viewing potential flood levels. We highly recommend use of such resources of the Federal Emergency Management Agency at `www.fema.gov`.
- **Inform the public now.** Information is the key to preparation, and preparation in turn is the best way to combat the effects of the rising sea level over the years to come.

References

Abdalati, Waleed. 2005. Canada's shrinking ice caps. Arctic Science Journeys radio script. Sea Grant Alaska. `http://seagrant.uaf.edu/news/05ASJ/03.25.05canada-ice.html`. Accessed 17 February 2008.

Chang, Raymond. 2003. *General Chemistry: The Essential Concepts*. 3rd ed. New York: McGraw-Hill.

Chen, J.L., C.R. Wilson, and B.D. Tapley. 2006. Satellite gravity measurements confirm accelerated melting of Greenland Ice Sheet. 7 July 2006. http://www.sciencemag.org/cgi/content/abstract/1129007v1.

Cockrell School of Engineering, University of Texas at Austin. 2006. Greenland's ice loss accelerating rapidly, gravity-measuring satellites reveal. 10 August 2006. http://www.engr.utexas.edu/news/articles/200608101082/index.cfm. Accessed 16 February 2008.

Ekwurzel, B. 2007. Findings of the IPCC Fourth Assessment Report: Climate Change Science. http://www.ucsusa.org/global_warming/science_and_impacts/science/findings-of-the-ipcc-fourth-2.html. Accessed 16 February 2008.

Florida Department of Environmental Protection. n.d. Erosion control line. http://data.labins.org/2003/SurveyData/WaterBoundary/ecl/ecl_search.cfm. Accessed 16 February 2008.

Galapagos ocean currents. 2007. GalapagosOnline. http://www.galapagosonline.com/Galapagos_Natural_History/Oceanography/Currents.html. Accessed 17 February 2008.

Haxby, William. 2000. Water world. http://www.pbs.org/wgbh/nova/warnings/waterworld/. Accessed 16 February 2008.

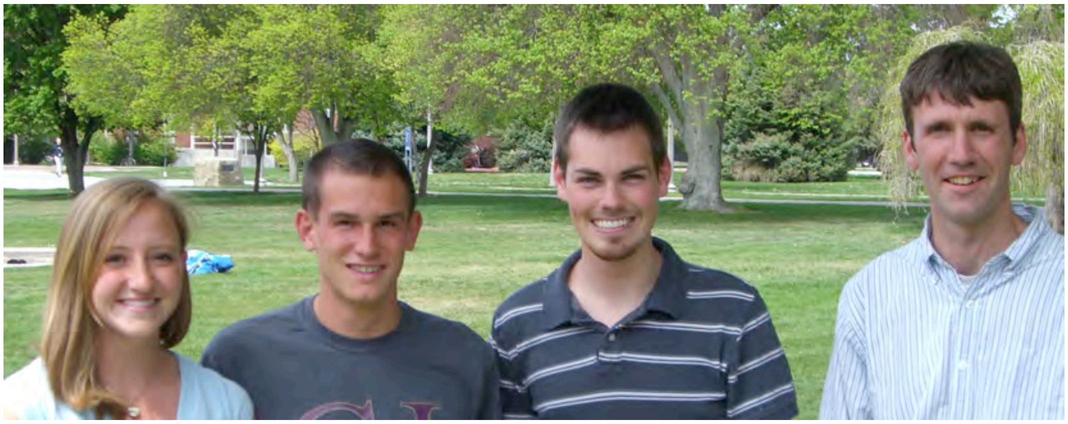
Morano, Marc. 2007. Latest scientific studies refute fears of Greenland melt. U.S. Senate Committee on Environment and Public Works. http://epw.senate.gov/public/index.cfm?FuseAction=Minority.Blogs&ContentRecord_id=175b568a-802a-23ad-4c69-9bdd978fb3cd. Accessed 17 February 2008.

National Oceanic and Atmospheric Administration (NOAA). 2008. NOAA helps prepare East Coast communities for tsunami, storm-driven flood threats. http://www.noaanews.noaa.gov/stories2007/20071203_eastcoasttsunami.html. Accessed 16 February 2008.

National Snow and Ice Data Center. 2008. All about sea ice. http://nsidc.org/seaice/intro.html. Accessed 16 February 2008.

NSTATE, LLC. 2007. The geography of Florida. http://www.netstate.com/states/geography/fl_geography.htm. Accessed 16 February 2008.

Pidwirny, Michael. 2008. Ocean. In The Encyclopedia of Earth, edited by Cutler J. Cleveland. Washington, DC: Environmental Information Coalition, National Council for Science and the Environment. http://www.eoearth.org/article/Ocean.



Team members Megan Longo, Nelson T. Gross, Benjamin Coate, and advisor Dr. Mike Hitchman.

A Convenient Truth: Forecasting Sea Level Rise

Jason Chen
Brian Choi
Joonhahn Cho

Duke University
Durham, NC

Advisor: Scott McKinley

Abstract

Greenhouse-gas emissions have produced global warming, including melting in the Greenland Ice Sheet (GIS), resulting in sea-level rise, a trend that could devastate coastal regions. A model is needed to quantify effects for policy assessments.

We present a model that predicts sea-level trends over a 50-year period, based on mass balance and thermal expansion acting on a simplified ice-sheet geometry. Mass balance is represented using the heat equation with Neumann conditions and sublimation rate equations. Thermal expansion is estimated by an empirically-derived equation relating volume expansion to temperature increase. Thus, the only exogenous variables are time and temperature.

We apply the model to varying scenarios of greenhouse-gas-concentration forcings. We solve the equations numerically to yield sea-level increase projections. We then project the effects on Florida, as modeled from USGS geospatial elevation data and metropolitan population data.

The results of our model agree well with past measurements, strongly supporting its validity. The strong linear trend shown by our scenarios indicates both insensitivity to errors in inputs and robustness with respect to the temperature function.

Based on our model, we provide a cost-benefit analysis showing that small investments in protective technology could spare coastal regions from flooding. Finally, the predictions indicate that reductions in greenhouse-gas emissions are necessary to prevent long-term sea-level-rise disasters.

The UMAP Journal 29 (3) (2008) 249–265.

Introduction

There is strong evidence of global warming; temperatures have increased by about 0.5°C over the last 15 years, and global temperature is at its highest level in the past millennium [Hansen et al. 2000]. One of the feared consequences of global warming is sea-level rise. Satellite observations indicate that a rise of 0.32 ± 0.02 cm annually 1993–1998 [Cabanaes et al. 2001]. Titus et al. [1991] estimate that a 1-meter rise in sea levels could cause $270–475 billion in damages in the U.S. alone.

Complex factors underlie sea-level rise. Thermal expansion of water due to temperature changes was long implicated as the major component, but it alone cannot account for observed increases [Wigley and Raper 1987]. Mass balance of large ice sheets, in particular the Greenland Ice Sheet, is now believed to play a major role. The mass balance is controlled by accumulation (influx of ice to the sheet, primarily from snowfall) and ablation (loss of ice from the sheet, a result of sublimation and melting) [Huybrechts 1999].

Contrary to popular belief, floating ice does not play a significant role. By Archimedes' Principle, the volume increase ΔV of a body of water with density ρ_{ocean} due to melting of floating ice of weight W (assumed to be freshwater, with liquid density ρ_{water}) is

$$\Delta V = W \left(\frac{1}{\rho_{\text{water}}} - \frac{1}{\rho_{\text{ocean}}} \right).$$

The density of seawater is approximately $\rho_{\text{ocean}} = 1024.8 \text{ kg/m}^3$ [Fofonoff and Millard 1983]; the mass of the Arctic sea ice is 2×10^{13} kg [Rothrock and Jang 2005]. Thus, the volume change if all Arctic sea ice melted would be

$$\Delta V = 2 \times 10^{13} \text{ kg} \left(\frac{1}{1000 \text{ kg/m}^3} - \frac{1}{1024.8 \text{ kg/m}^3} \right).$$

Approximating that 360 Gt of water causes a rise of 0.1 cm in sea level [Warrick et al. 1996], we find that volume change accounts for a rise of

$$4.84 \times 10^8 \text{ m}^3 \times \frac{1000 \text{ kg}}{\text{m}^3} \times \frac{1 \text{ Gt}}{9.072 \times 10^{11} \text{ kg}} \times \frac{0.1 \text{ cm}}{360 \text{ Gt}} \approx 0.00015 \text{ cm}.$$

This small change is inconsequential.

We also neglect the contribution of the Antarctic Ice Sheet because its overall effect is minimal and difficult to quantify. Between 1978 and 1987, Arctic ice decreased by 3.5% but Antarctic ice showed no statistically significant changes [Gloersen and Campbell 1991]. Cavalieri et al. projected minimal melting in the Antarctic over the next 50 years [1997]. Hence, our model considers only the Greenland Ice Sheet.

Models for mass balance and for thermal expansion are complex and often disagree (see, for example, Wigley and Raper [1987] and Church et al. [1990]). We develop a model for sea-level rise as a function solely of temperature and time. The model can be extended to several different temperature forcings, allowing us to assess the effect of carbon emissions on sea-level rise.

Model Overview

We create a framework that incorporates the contributions of ice-sheet melting and thermal expansion. The model:

- accurately fits past sea-level-rise data,
- provides enough generality to predict sea-level rise over a 50-year span,
- computes sea-level increases for Florida as a function of only global temperature and time.

Ultimately, the model predicts consequences to human populations. In particular, we analyze the impact in Florida, with its generally low elevation and proximity to the Atlantic Ocean. We also assess possible strategies to minimize damage.

Assumptions

- Sea-level rise is primarily due to the balance of accumulation/ablation of the Greenland Ice Sheet and to thermal expansion of the ocean. We ignore the contribution of calving and direct human intervention, which are difficult to model accurately and have minimal effect [Warrick et al. 1996].
- The air is the only heat source for melting the ice. Greenland's land is permafrost, and because of large amounts of ice on its surface, we assume that it is at a constant temperature. This allows us to use conduction as the mode of heat transfer, due to the presence of a key boundary condition.
- The temperature within the ice changes linearly at the steady state. This assumption allows us to solve the heat equation for Neumann conditions. By subtracting the steady-state term from the heat equation, we can solve for the homogeneous boundary conditions.
- Sublimation and melting processes do not interfere with each other. Sublimation primarily occurs at below-freezing temperatures, a condition during which melting does not normally occur. Thus, the two processes are temporally isolated. This assumption drastically simplifies computation, since we can consider sublimation and melting separately.

- The surface of the ice sheet is homogeneous with regard to temperature, pressure, and chemical composition. This assumption is necessary because there are no high-resolution spatial temperature data for Greenland. Additionally, simulating such variation would require finite-element methods and mesh generation for a complex topology.

Defining the Problem

Let M denote the mass balance of the Greenland Ice Sheet. Given a temperature-forcing function, we estimate the sea-level increases (SLR) that result. These increases are a sum of M and thermal expansion effects, corrected for local trends.

Methods

Mathematically Modeling Sea-Level Rise

Sea-level rise results mostly from mass balance of the Greenland Ice Sheet and thermal expansion due to warming. The logic of the simulation process is detailed in **Figure 1**.

Temperature Data

We create our own temperature data, using input forcings that we can control. We use the EdGCM global climate model (GCM) [Shopsin et al. 2007], based on the NASA GISS model for climate change. Its rapid simulation (10 h for a 50-year simulation) allows us to analyze several scenarios.

Three surface air temperature scenarios incorporate the the low, medium, and high projections of carbon emissions in the IS92 series resulting from the IPCC Third Assessment Report (TAR) [Edmonds et al. 2000]. The carbon forcings are shown in **Figure 2**. All other forcings are kept at default according to the NASA GISS model.

One downside to the EdGCM is that it can output only *global* temperature changes; regional changes are calculated but are difficult to access and have low spatial accuracy. However, according to Chylek and Lohmann [2005], the relationship between Greenland temperatures and global temperatures is well approximated by

$$\Delta T_{\text{Greenland}} = 2.2\Delta T_{\text{global}}.$$

The Ice Sheet

We model the ice sheet as a rectangular box. We assume that each point on the upper surface is at constant temperature T_a, because our climate

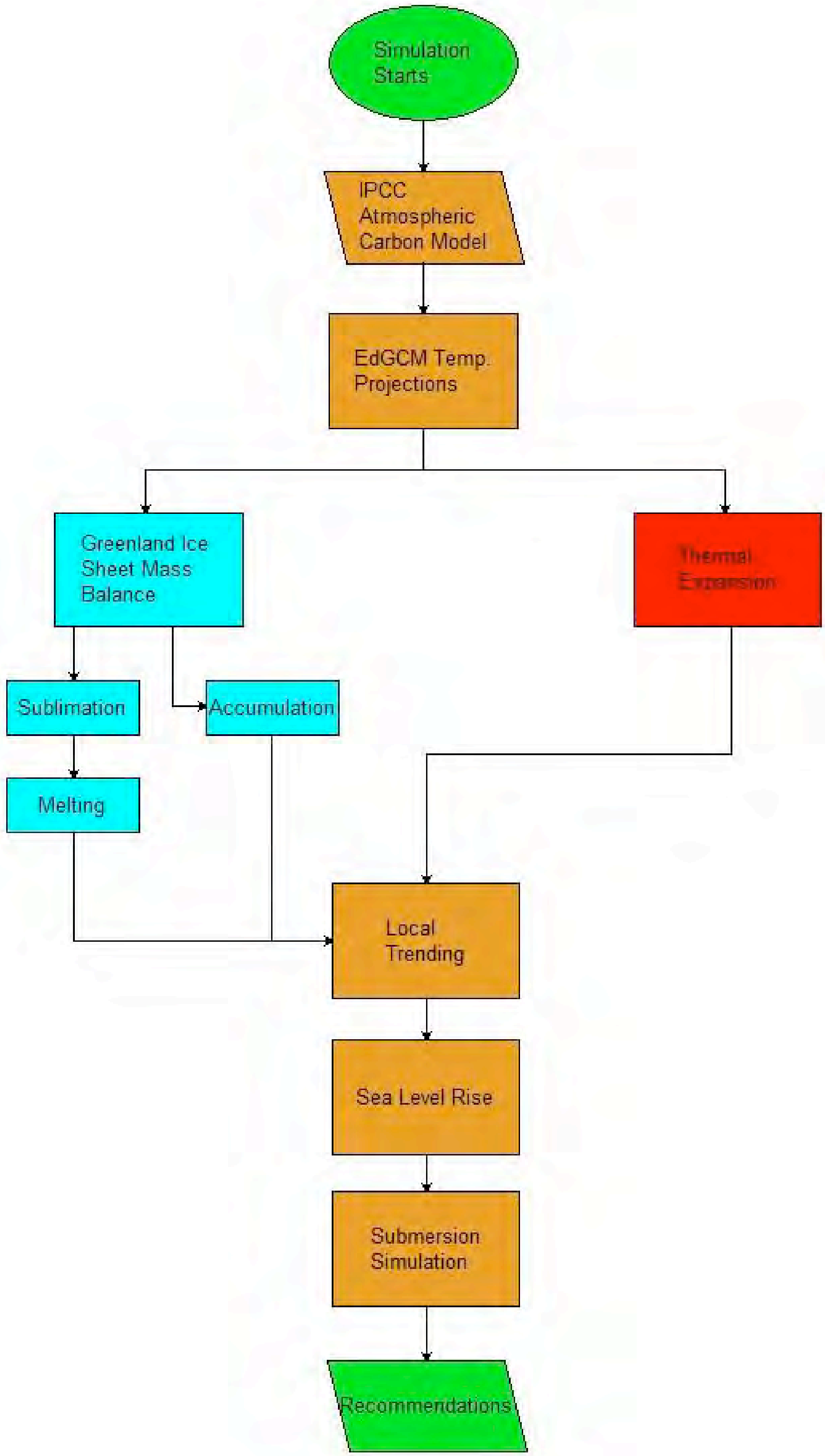


Figure 1. Simulation flow diagram.

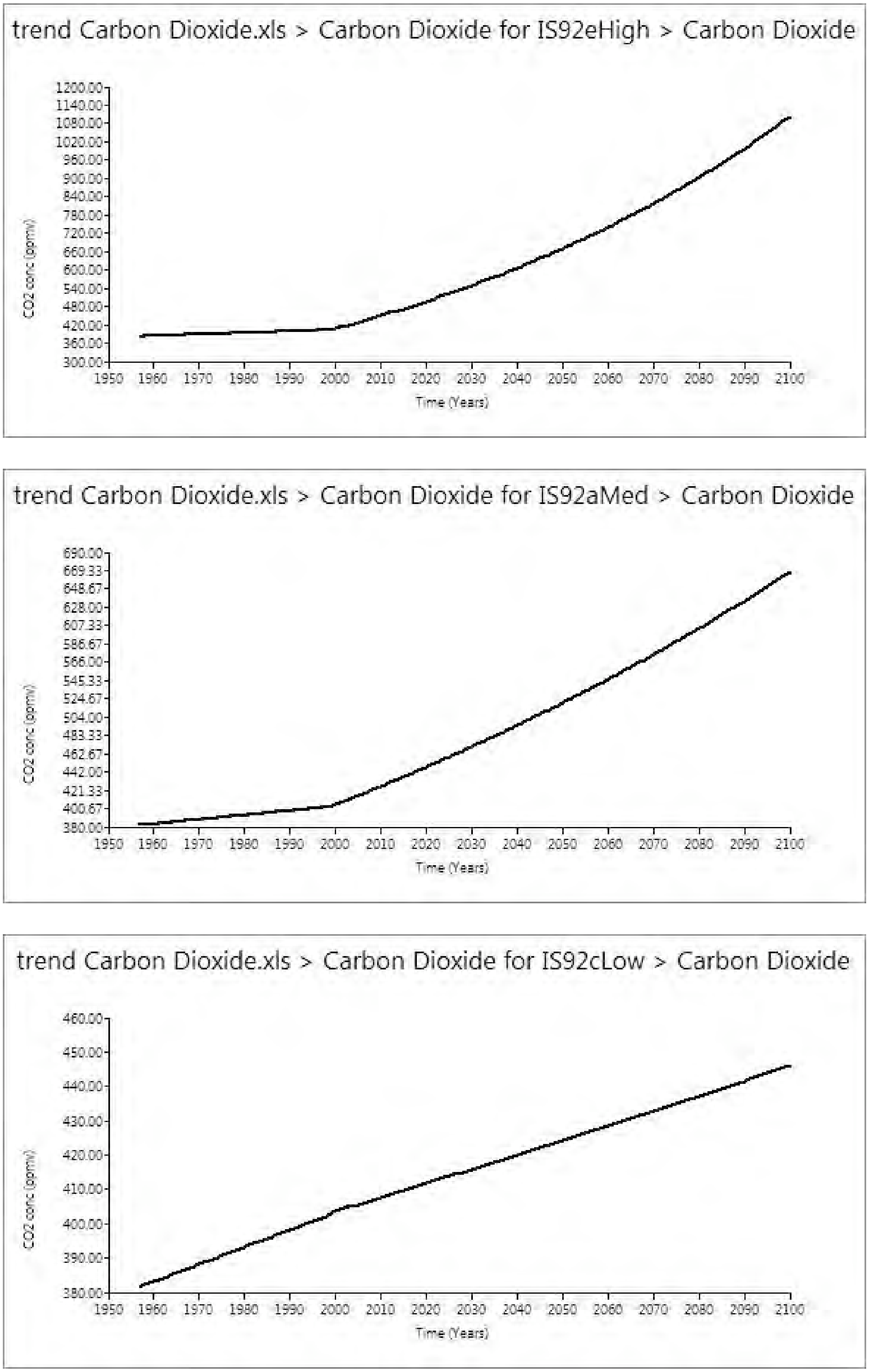


Figure 2. Carbon dioxide forcings for the EdGCM models.

model does not have accurate spatial resolution for Greenland. The lower surface, the permafrost layer, has constant temperature T_l.

To compute heat flux, and thus melting and sublimation through the ice sheet, we model it as an infinite number of differential volumes (**Figure 3**).

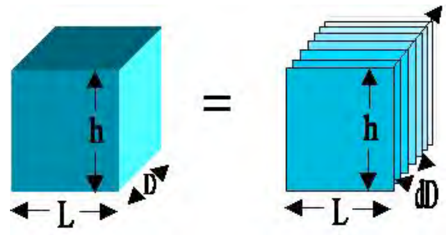


Figure 3. Differential volumes of the ice sheet.

The height h of the box is calculated using data provided by Williams and Ferrigno [1999]:

$$h = \frac{\text{Volume}_{\text{ice}}}{\text{Surface}_{\text{ice}}} = \frac{2.6 \times 10^6 \text{ km}^3}{1.736 \times 10^6 \text{ km}^2} = 1.5 \text{ km}.$$

The primary mode of sea-level rise in our model is through mass balance: accumulation minus ablation.

Mass Balance: Accumulation

Huybrechts et al. [1991] show that the temperature of Greenland is not high enough to melt significant amounts of snow. Furthermore, Knight [2006] shows that the rate of accumulation of ice is well-approximated by a linear relationship of 0.025 m/month of ice. In terms of mass balance, we have

$$M_{\text{ac}} = 0.025LD,$$

where L and D are the length and width of the rectangular ice sheet.

Mass Balance: Ablation

We model the two parts of ablation, sublimation and melting.

Sublimation

The sublimation rate (mass flux) is given by:

$$S_0 = e_{\text{sat}}(T) \left(\frac{M_w}{2\pi RT} \right)^{1/2},$$

where M_w is the molecular weight of water and T is the temperature in kelvins. This expression can be derived from the ideal gas law and the Maxwell-Boltzmann distribution [Andreas 2007]. Substituting Buck's [1981] expression for e_{sat}, we obtain:

$$S_0 = 6.1121 \exp\left[\frac{\left(18.678 - \frac{T}{234.5}\right)T}{257.14+T}\right]\left(\frac{M_w}{2\pi R(T+273.15)}\right)^{1/2},$$

where we now scale T in °C. Buck's equation is applicable over a large range of temperatures and pressures, including the environment of Greenland. To convert mass flux into rate of change of thickness the ice, we divide the mass flux expression by the density of ice, getting the rate of height change as

$$S_h = \frac{6.1121d}{\rho_{\text{ice}}} \exp\left[\frac{\left(18.678 - \frac{T}{234.5}\right)T}{257.14+T}\right]\left(\frac{M_w}{2\pi R(T+273.15)}\right)^{1/2},$$

where d is the deposition factor, given by $d = (1 - \text{deposition rate}) = 0.01$ [Buck 1981].

The thickness of the ice sheet after one timestep (= one month) of the computational model is

$$S(t) = h - S_h t,$$

where h is the current thickness of the ice sheet and t is one timestep. Substituting for S_h the expression above and the molecular weight of water yields

$$S(t) = h - \frac{6.1121 \times 10^{-2} t}{\rho_{\text{ice}}} \exp\left[\frac{(18.678 - \frac{T}{234.5})T}{257.14+T}\right]\left(\frac{M_w}{2\pi R(T+273.15)}\right)^{1/2}.$$

Melting

To model melting, we apply the heat equation

$$U_t(x,t) = kU_{xx}(x,t),$$

using $k = 0.0104$ as the thermal diffusivity of the ice [Polking et al. 2006]. For the Neumann conditions, we assume a steady-state U_s with the same boundary conditions as U and that is independent of time. The residual temperature V has homogeneous boundary conditions and initial conditions found from $U - U_s$. Thus, we can rewrite the heat equation as

$$U(x,t) = V(x,t) + U_s(x,t).$$

The steady-state solution is

$$U_s = T_l + \frac{T_a - T_l}{S(t)}\,x,$$

subject to the constraints $0 < x < S(t)$ and $0 < t < 1$ month. Directly from the heat equation we also have

$$V_t(x,t) = kV_{xx}(x,t) + f, \quad \text{where } f \text{ is a forcing term; and}$$
$$V(0,t) = V\big(S(t),t\big) = 0, \quad \text{for the homogeneous boundary equations.}$$

Since no external heat source is present and temperature distribution depends only on heat conduction, we take as the forcing term $f = 0$. To calculate change in mass balance on a monthly basis, we solve analytically using separation of variables:

$$V(x,t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \exp\left[\frac{-n^2\pi^2 t}{s^2}\right] \cos\left(\frac{n\pi x}{s}\right),$$

where

$$a_0 = \frac{2}{s}\int_0^s \left(T_l + \frac{T_a - T_l}{s}\, x\right) dx = 2T_1 + T_a - T_l = T_l + T_a$$

and

$$\begin{aligned} a_0 &= \frac{2}{s}\int_0^s \left(T_l + \frac{T_a - T_l}{s}\, x\right) \cos\left(\frac{n\pi x}{s}\right) dx \\ &= \left(\frac{s}{n\pi}\right)^2 \big(\cos(n\pi) - 1\big) \\ &= \left(\frac{s}{n\pi}\right)^2 \big((-1)^n - 1\big). \end{aligned}$$

Therefore,

$$V(x,t) = \frac{T_l + T_a}{2} + \sum_{n=1}^{\infty} \frac{2(T_a - T_l)}{(n\pi)^2}\big((-1)^n - 1\big) \exp\left[\frac{-n^2\pi^2 t}{s^2}\right] \cos\left(\frac{n\pi x}{s}\right).$$

Having found $V(x,t)$ and $U_s(x,t)$, we obtain an expression for $U(x,t)$ from

$$U(x,t) = V(x,t) + U_s(x,t).$$

Since U is an increasing function of x, and for $x > k$, we have $U(x,t) > 0$ for fixed t; the ice will melt for $k < x < h$. To determine ablation, we solve $U(k,t) = 0$ for k using the first 100 terms of the Fourier series expansion and the Matlab function `fzero`. We use the new value of k to renew h as the new thickness of the ice sheet for the next timestep.

With these two components, we can finalize an expression for ablation and apply it to a computational model. The sum of the infinitesimal changes in ice sheet thickness for each differential volume gives the total change in thickness. To find these changes, we first note that

$$\begin{aligned}\text{Mass balance loss due to sublimation} &= (h-S)LD,\\ \text{Mass balance loss due to melting} &= (S-k)LD,\end{aligned}$$

where the product LD is the surface area of the ice sheet. In these equations, the "mass balance" refers to net volume change. Thus, ablation is given by

$$M_{\text{ab}} = (h-S)LD + (S-k)LD = (h-k)LD.$$

Mass Balance and Sea-Level Rise

Combining accumulation and ablation into an expression for mass balance, we have

$$M = M_{\text{ac}} - M_{\text{ab}} = 0.025LD - (h-k)LD.$$

Relating this to sea-level rise, we use the approximation 360 Gt water = 0.1 cm sea-level rise. Thus,

$$\text{SLR}_{\text{mb}} = M\rho_{\text{ice}}\,\frac{0.1\text{ cm}}{360\text{ Gt}},$$

which quantifies the sea-level rise due to mass balance.

Thermal Expansion

According to Wigley and Raper [1987], for the current century thermal expansion of the oceans due to increase in global temperature will contribute at least as much to rise in sea level as melting of polar ice [Huybrechts et al. 1991; Titus and Narayanan 1995]. So we incorporate thermal expansion into our model.

Temperature plays the primary role in thermal expansion, but the diffusion of radiated heat, mixing of the ocean, and various other complexities of ocean dynamics must be accounted for in a fully accurate description. We adapt the model of Wigley and Raper [1987]. Based on standard greenhouse-gas emission projections and a simple upwelling-diffusion model, the dependency of the model can be narrowed to a single variable, temperature, using an empirical estimation:

$$\Delta z = 6.89\Delta T k^{0.221},$$

where

Δz is the change in sea level due to thermal expansion (cm),

ΔT is the change in global temperature (°C), and

k is the diffusivity.

Localization

A final correction must be added to the simulation. The rise in sea level will vary regionally rather significantly. The local factors often cited include land subsidence, compaction, and delayed response to warming [Titus and Narayanan 1995]. We thus assume that previous patterns of local sea-level variation will continue, yielding the relationship

$$\text{local}(t) = \text{normalized}(t) + \text{trend}(t - 2008),$$

where

- $\text{local}(t)$ is the expected sea level rise at year t (cm),
- $\text{normalized}(t)$ is the estimate of expected rise in global sea level change relative to the historical rate at year t, and
- trend is the current rate of sea-level change at the locale of interest.

The normalization prevents double-counting the contribution from global warming.

In our model, the rates of sea-level change are averaged over data for Florida from Titus and Narayanan [1995] to give the trend. This is reasonable because the differences between the rates in Florida are fairly small. The normalized (t) at each year is obtained from

$$\text{global}(t) - \text{historical rate}(t - 2008),$$

where $\text{global}(t)$ is the expected sea-level rise at year t from our model and historical rate is chosen uniformly over the range taken from Titus and Narayanan [1995].

Simulating Costs of Sea-Level Rise to Florida

To model submersion of regions of Florida due to sea-level rise, we created a raster matrix of elevations for various locations, using USGS data (GTOPO30) [1996]. The 30-arc-second resolution corresponds to about 1 km; however, to yield a more practical matrix, we lowered the resolution to 1 minute of arc (approx. 2 km).

The vertical resolution of the data is much greater than 1 m. To model low coastal regions, the matrix generation code identified potential sensitive areas and submitted these to the National Elevation Dataset (NED) [Seitz 2007] for refinement. (NED's large size and download restrictions restrict its use to sensitive areas.) The vertical resolution of NED is very high [USGS 2006]. We use these adjustments to finalize the data.

We measure the effect of sea-level rise on populations by incorporating city geospatial coordinates and population into the simulation. We obtained geospatial coordinates from the GEOnames Query Database [National Geospatial Intelligence Agency 2008] and population data from the U.S. Census Bureau [2000].

We used the sea-level rise calculated from our model as input for the submersion simulation, which subtracts the sea-level increase from the elevation. If rising sea level submerges pixels in a metropolitan area, the population is considered "displaced."

A key limitation of the model is that the population is considered to be concentrated in the principal cities of the metropolitan areas, so a highly accurate population count cannot be assessed. This simplification allows quick display of which cities are threatened without the complexity of hard-to-find high-resolution population distribution data.

We checked the model for realism at several different scenarios. As shown in **Figure 4**, our expectations are confirmed:

- 0 m: No cities are submerged and no populations or land areas are affected.
- 10 m: This is slightly higher than if all of the Greenland Ice Sheet melted (approx. 7 m). Many cities are submerged, especially in southern Florida.
- 100 m: Most of Florida is submerged.

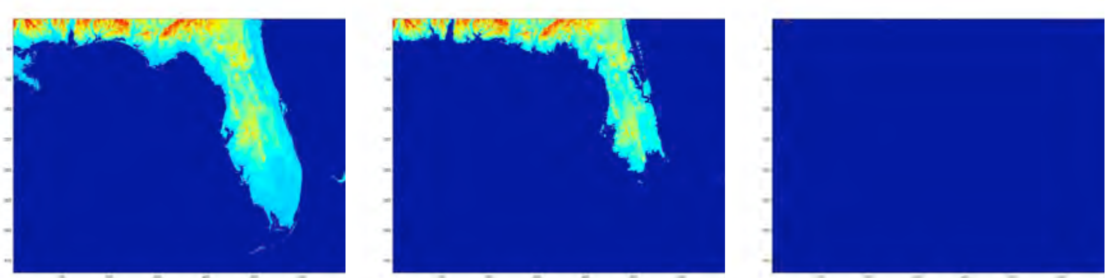


Figure 4. Effects of 0, 10, and 100-meter sea-level rise.

Results

Output Sea-Level-Rise Data

We ran the program with a Matlab script for the IS92e (high), IS92a (intermediate), and IS92c (low) carbon-emissions models. The program produces a smooth trend in sea-level increase for each of the three forcings, as shown in **Figure 5**: Higher temperature corresponds to higher sea-level rise, as expected. The sea-level output data are then used to calculate submersion consequences.

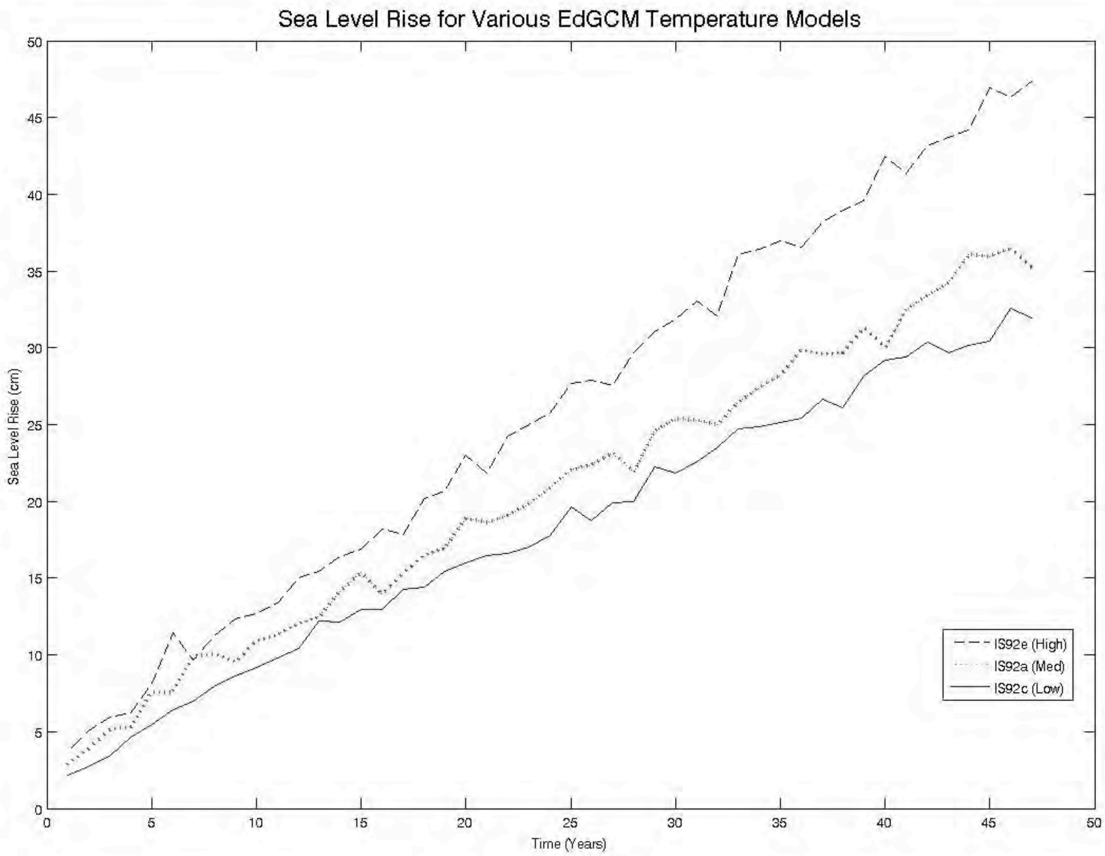


Figure 5. Sea level rise as a function of time for the three temperature models.

Submersion Simulation Results

Output consists of the submerged land area and displaced population statistics. The program quantified the effects noted in **Table 1**. For the low and medium scenarios, no metropolitan areas are submerged until after 30 years. In all scenarios, Miami Beach and Key Largo are submerged after 40 years.

Discussion and Conclusion

The estimated sea-level rises (**Figure 5**) for the three scenarios seem reasonable. The 50-year projection is in general agreement with models proposed by the IPCC , NRC, and EPA (less than 10 cm different from each) [Titus et al. 1991]. Additionally, the somewhat-periodic, somewhat-linear trend is similar to past data of mean sea-level rise collected in various locations [Titus et al. 1991]. Thus, the projections of our model are reasonable.

The high-emission scenario results in a 40–50 cm rise in sea level by 2058, with results from the intermediate scenario 6–10 cm lower and the

Table 1.
Effects under different scenarios (using current population values).

Time	High		Medium		Low	
(yrs)	Displaced ($\times 10^3$)	Submerged (km^2 $\times 10^3$)	Displaced ($\times 10^3$)	Submerged (km^2 $\times 10^3$)	Displaced ($\times 10^3$)	Submerged (km^2 $\times 10^3$)
10	0	6.5	0	6.4	0	6.2
20	12	7.5	0	6.9	0	6.8
30	100	9.2	12	7.7	0	7.1
40	100	9.7	100	9.0	100	8.0
100	135	10.0	100	9.5	100	9.2

low-emission scenario trailing intermediate by 5–8 cm. The model thus works as expected for a wide range of input data: Higher temperatures lead to increased sea level rise.

Overall, the damage due to sea-level change seems unremarkable. Even in the worst-case scenario, in 50 years only 135,000 people are displaced and 10,000 square kilometers are submerged, mostly in South Florida.

However, these projections are only the beginning of what could be a long-term trend. As shown by the control results, a sea-level increase of 10 m would be devastating. Further, not all possible damages are assessed in our simulation. For example, sea-level increases have been directly implicated also in shoreline retreat, erosion, and saltwater intrusion. Economic damages are not assessed. Bulkheads, levees, seawalls, and other structures are often built to counteract the effect of rising sea levels, but their economic impacts are outside the scope of the model.

Our model has several key limitations. The core assumption of the model is the simplification of physical features and dynamics in Greenland. The model assumes an environment where thickness, temperature, and other physical properties are averaged out and evenly distributed. The "sublimate, melt, and snow" dynamics are simulated with a monthly timestep. Such assumptions are too simplistic to capture fully the ongoing dynamics in the ice sheets. But we do not have the data and computing power to perform a full-scale 3-D grid-based simulation using energy-mass balance models, as in Huybrechts [1999].

With regard to minor details of the model, the assumed properties regarding the thermal expansion, localization, and accumulation also take an averaging approach. We make an empirical estimate adapted from Wigley and Raper [1987]. Consequently, our model may not hold over a long period of time, when its submodels for accumulation, thermal expansion, and localization might break down.

The assumptions of the EdGCM model are fairly minimal, and the projected temperature time series for each scenario are consistent with typical carbon projections [Edmonds et al. 2000]. Although the IS92 emissions scenarios are very rigorous, they are the main weakness of the model. Because all of the other parameters depend on the temperature model, our results are particularly sensitive to factors that directly affect the EdGCM output.

Despite these deficiencies, our model is a powerful tool for climate modeling. Its relative simplicity—while it can be viewed as a weakness—is actually a key strength of the model. The model boasts rapid runtime, due to its simplifications. Furthermore, the model is a function of time and temperature only; the fundamentals of our model imply that all sea-level increase is due to temperature change. But even with less complexity, our model is comprehensive and accurate enough to provide accurate predictions.

Recommendations

In the short term, preventive action could spare many of the model's predictions from becoming reality. Key Largo and Miami Beach, which act as a buffer zone preventing salinization of interior land and freshwater, are particularly vulnerable. If these regions flood, seawater intrusion may occur, resulting in widespread ecological, agricultural, and ultimately economical damage. Titus and Narayanan [1995] recommend building sand walls.

In the long term, carbon emissions must be reduced to prevent disasters associated with sea-level rise.

References

Andreas, E.L. 2007. New estimates for the sublimation rate of ice on the Moon. *Icarus* 186, 24–30.

Buck, A.L. 1981. New equations for computing vapor pressure and enhancement factor. *Journal of Applied Meteorology* 20: 1527–1532.

Cabanes, C., A. Cazenave, and C.L. Provost. 2001. Sea level rise during past 40 years determined from satellite and in situ observations. *Science* 294: 840–842.

Cavalieri, D.J., P. Gloersen, C.L. Parkinson, J.C. Comiso, and H.J. Zwally. 1997. Observed hemispheric asymmetry in global sea ice changes. *Science* 278: 1104–1106.

Church, J.A., J.S. Godfrey, D.R. Jackett, and T.T. McDougall. 1990. Sea level rise during past 40 years determined from satellite and in situ observations. *Journal of Climate* 4: 438–456.

Chylek, P., and U. Lohmann. 2005. Ratio of the Greenland to global temperature change: Comparison of observations and climate modeling results. *Geophysical Research Letters* 32: 1–4.

Edmonds, J., R. Richels, and M. Wise. 2000. Ratio of the Greenland to global temperature change: Comparison of observations and climate modeling results: A review. In *The Carbon Cycle*, edited by T.M.L. Wigley and D.S. Schimel, 171–189. Cambridge, UK: Cambridge University Press.

Fofonoff, P., and R.C. Millard, Jr. 1983. Algorithms for computation of fundamental properties of seawater. *UNESCO Technical Papers in Marine Sciences* 44: 1–53.

Gloersen, P., and W.J. Campbell. 1991. Recent variations in arctic and antarctic sea-ice covers. *Nature* 352: 33–36.

Hansen, J., M. Sato, R. Ruedy, A. Lacis, and V. Oinas. 2000. Global warming in the twenty-first century: An alternative scenario. *Proceedings of the National Academy of Science* 97: 9875–9880.

Huybrechts, P. 1999. The dynamic response of the Greenland and Antarctic ice sheets to multiple-century climatic warming. *Journal of Climate* 12: 2169–2188.

________, A. Letreguilly, and N. Reeh. 1991. The Greenland ice sheet and greenhouse warming. *Palaeography, Palaeoclimatology, Palaeoecology* 89: 399–412.

Knight, P. 2006. *Glacier Science and Environmental Change*. Oxford, UK: Wiley-Blackwell.

National Geospatial-Intelligence Agency. 2008. GEOnames Query Database. `http://earth-info.nga.mil/gns/html/index.html` .

Polking, J., A. Boggess, and D. Arnold. 2006. *Differential Equations with Boundary Value Problems*. 2nd ed. Upper Saddle River, NJ: Pearson Education.

Rothrock, D.A., and J. Zhang. 2005. Arctic Ocean sea ice volume: What explains its recent depletion? *Journal of Geophysical Research* 110: 1–10; `http://psc.apl.washington.edu/pscweb2002/pubs/rothrockJRG05.pdf` .

Seitz, M. 2007. Lat/lon to elevation. `http://www.latlontoelevation.com/` .

Shopsin, M., M. Shopsin, and K. Mankoff. 2007. *Education Global Climate Modeling*. New York: Columbia University Press.

Titus J.G., and V.K. Narayanan. 1995. *The Probability of Sea Level Rise*. Washington, DC: U.S. Environmental Protection Agency.

Titus, J.G., R.A. Park, S.P. Leatherman, J.R. Weggel, M.S. Greene, P.W. Mausel, S. Brown, G. Gaunt, M. Trehan, and G. Yohe. 1991. Greenhouse effect and sea level rise: The cost of holding back the sea. *Coastal Management* 19: 171–204; `http://www.owr.ehnr.state.nc.us/ref/16/15086.pdf` .

U.S. Census Bureau. 2000. Census 2000 datasets. `http://www2.census.gov/census_2000/datasets/`.

U.S. Geological Survey (USGS). 1996. GTOPO30 Tile W100N40. Earth Resources Observation and Sciences. `http://edc.usgs.gov/products/elevation/gtopo30/w100n40.html`.

________. 2006. Accuracy of the National Elevation Dataset. `http://ned.usgs.gov/Ned/accuracy.asp`.

Warrick, R., C.L. Provost, M. Meier, J. Oerlemans, and P. Woodworth 1996. *Climate Change 1995: The Science of Climate Change*. Cambridge, UK: Cambridge University Press.

Wigley, T.M.L., and S.C.B. Raper. 1987. Thermal expansion of sea water associated with global warming. *Nature* 330: 127–131.

Williams, Richard S., Jr., and Jane G. Ferrigno. 1999. Estimated present-day area and volume of glaciers and maximum sea level rise potential. U.S. Geological Survey Professional Paper 1386-A. In *Satellite Image Atlas of Glaciers of the World*, Chapter A, edited by Richard S. Williams, Jr., and Jane G. Ferrigno. Washington, DC: U.S. Government Printing Office. `http://www.smith.edu/libraries/research/class/idp108USGS_99.pdf`. An updated version was in press in 2007 and is to be available from `http://pubs.usgs.gov/fs/2005/3056/fs2005-3056.pdf`.

Brian Choi, Joonhahn Cho, and Jason Chen.

Pp. 267–300 can be found on the *Tools for Teaching 2008* CD-ROM.

Fighting the Waves: The Effect of North Polar Ice Cap Melt on Florida

Amy M. Evans
Tracy L. Stepien

University at Buffalo, The State University of New York
Buffalo, NY

Advisor: John Ringland

Abstract

A consequence of global warming that directly impacts U.S. citizens is the threat of rising sea levels due to melting of the North Polar ice cap. One of the many states in danger of losing coastal land is Florida. Its low elevations and numerous sandy beaches will lead to higher erosion rates as sea levels increase. The direct effect on sea level of only the North Polar ice cap melting would be minimal, yet the indirect effects of causing other bodies of ice to melt would be crucial. We model individually the contributions of various ice masses to rises in sea level, using ordinary differential equations to predict the rate at which changes would occur.

For small ice caps and glaciers, we propose a model based on global mean temperature. Relaxation time and melt sensitivity to temperature change are included in the model. Our model of the Greenland and Antarctica ice sheets incorporates ice mass area, volume, accumulation, and loss rates. Thermal expansion of water also influences sea level, so we include this too. Summing all the contributions, sea levels could rise 11–27 cm in the next half-century.

A rise in sea level of one unit is equivalent to a horizontal loss of coastline of 100 units. We investigate how much coastal land would be lost, by analyzing relief and topographic maps. By 2058, in the worst-case scenario, there is the potential to lose almost 27 m of land. Florida would lose most of its smaller islands and sandy beaches. Moreover, the ports of most major cities, with the exception of Miami, would sustain some damage.

Predictions from the Intergovernmental Panel on Climate Change (IPCC) and from the U.S. Environmental Protection Agency (EPA) and simulations

The UMAP Journal 29 (3) (2008) 267–284.

from the Global Land One-km Base Elevation (GLOBE) digital elevation model (DEM) match our results and validate our models.

While the EPA and the Florida state government have begun to implement plans of action, further measures need to be put into place, because there will be a visible sea-level rise of 3–13 cm in only 10 years (2018).

Introduction

Measurements and observations of Earth's ice features (e.g., glaciers, ice sheets, and ice packs) indicate changes in the climate [Kluger 2006; NASA Goddard Institute for Space Studies 2003; Natural Resources Defense Council 2005] and consequent raised ocean levels resulting from their melting.

Over the past 30 years, the amount of ice covering the North Pole has been reduced by 15%–20%. Additionally, the snow season in which ice is restored to the pack has grown shorter. By 2080, it is expected that there will be no sea ice during the summer [Dow and Downing 2007].

Besides the Arctic ice pack, glaciers of around the world are also shrinking. Warmer air and ocean waters cause the melting, and most glaciers have retreated at unparalleled rates over the past 60 years [Dow and Downing 2007].

Two other signs of a changing climate have direct impacts on people: increased weather-related disasters and a rising sea level. In 2005, the United States experienced 170 floods and 122 windstorms, compared with 8 floods and 20 windstorms in 1960. The statistics are similar for other countries, with 110,000 deaths due to weather-related catastrophes worldwide in 2005.

Sea-level rise results are visible. Small, low-lying islands in the Southern Pacific Ocean have either disappeared (e.g., two of the Kiribati islands in 1999) or had to be abandoned by residents (e.g., Carteret Islands in Papua New Guinea). Over the 20th century, the average sea-level rise was roughly 15 cm. If this trend continues, many more islands as well as the coastline of some countries would be lost [Dow and Downing 2007].

Assumptions

All the documentation that we encountered stated the same basic claim: *The North Polar ice cap melting will on its own affect the global ocean level by only a negligible amount.* This claim is simply a matter of the Archimedes Principle: The volume of water that would be introduced to the world's oceans and seas is already displaced by the North Polar ice pack, since it is comprised of frozen sea water floating in the Arctic Ocean.

However, the disappearing Arctic ice pack will speed up global warming, which encourages the melting of other land ice masses on Earth (e.g. Greenland, Antarctica, etc.). Thus, ocean levels *will* rise more as the North

Polar ice cap shrinks, due to indirect effects. In fact, North Polar ice cap melt is used as an "early warning system" for climate change around the world [Arctic Climate Impact Assessment 2004].

Worldwide Consequences of the Warming Arctic

As greenhouse gases increase in the atmosphere, snow and ice in the Arctic form later in the fall and melt earlier in the spring. As a result, there is less snow to reflect the sun's rays and more dark space (land and water) to absorb energy from the sun. The result, then, is the snow and ice forming later and melting earlier: A cycle emerges. Along the same lines, as snow and ice recede on the Arctic tundra, vegetation will grow on the exposed land, which will further increase energy absorption. Even though new trees would take in some of the CO_2 in the atmosphere, it would not be enough to compensate for the human-produced CO_2 causing the warming. Also, humans produce soot that is deposited in the Arctic by wind currents; the soot darkens the snow and further adds to soaking up energy from the sun. All of these changes will vary the world climate and lead to an increased global temperature [Arctic Climate Impact Assessment 2004].

When ice forms on the Arctic ice pack, most of the salt is pushed to the water directly below the mass. Therefore, the salinity of the water where sea ice is being formed and melted increases, which is an important step in *thermohaline circulation*—a system driven by the differences in heat and salt concentration that is related to ocean currents and the jetstream. Heating and melting in the Arctic will greatly affect the ocean currents of the world by slowing thermohaline circulation: The rate of deep water formation will decrease and lead to less warm water being brought north to be cooled. As a result, there will be regional cooling in the northern seas and oceans and an overall thermal expansion in the rest of the world, leading to a rise in sea level [Arctic Climate Impact Assessment 2004; Bentley et al. 2007].

Another direct impact of warming in the Arctic is the melting of permafrost, permanently frozen soil in the polar region. The melting of permafrost could lead to the release of large amounts of carbon dioxide, methane, and other greenhouse gases into the atmosphere [NASA Goddard Institute for Space Studies 2003]. Although warming would have to be fairly significant for this to occur, the consequences could be great, since another cycle of warming will take hold [Arctic Climate Impact Assessment 2004].

As the Arctic warms and global temperatures continue to rise, land ice will melt at an increasing rate. The associated sea-level rise will cause major loss of coastal land around the globe [Dow and Downing 2007].

The Arctic ecosystem itself will be completely disrupted by the warming environment. Food and habitat destruction will have a devastating effect on the mammals, fish, and birds that thrive in this cold environment. What's more, ecosystems farther south will be impacted, because a large number of Arctic animals move there during the summer months in search of food and for breeding purposes [Arctic Climate Impact Assessment 2004; Bentley et al. 2007].

Finally, the warming Arctic will change the lives of humans around the globe. The most directly impacted will be the native peoples living in the North Polar region who depend on the ice pack and northern glaciers as a home and hunting ground. These people will be forced to move farther south and find new means of survival. Many fishing industries that depend on the Arctic as a source of income will see a reduction in catches. There will also be easier access to oil and minerals that lie under the ocean floor—a happy thought for some and horrific for others [Arctic Climate Impact Assessment 2004; Bentley et al. 2007].

We focus on the effect of small ice caps, glaciers, and the Greenland and Antarctica ice sheets melting over the next 50 years, since these will have a direct effect on sea level. Furthermore, we predict the effects of a sea-level rise on Florida. Finally, we propose a response plan.

Modeling Small Ice Caps and Glaciers

Though ice caps and glaciers are small compared to the Greenland and Antarctica ice sheets, they are located in warmer climates and tend to have a quicker reaction rate in response to climate change, and they will cause more-immediate changes in sea level [Oerlemans 1989].

Global Mean Temperature

Global mean temperature is a measure of world-wide temperature change and is based on various sets of data. Overall trends in the temperature change can be detected, and periods of global warming and global cooling can be inferred.

Trends can clearly be seen in annual temperature anomaly data (relative changes in temperature) [NASA Goddard Institute for Space Studies 2008]. **Figure 1** shows global annual mean temperature for January through December of each year.

In the late 1800s, the temperature anomalies were negative yet increasing. For 1930–1970, the temperature anomalies hovered around 0; but then by the 1980s they remained positive and have been increasing.

Assumptions and Formation

We model the contribution of small ice caps and glaciers to sea-level rise by a model that uses a relation between the global mean temperature and the mass change of the small ice caps and glaciers [Oerlemans 1989; Wigley and Raper 1993]. We begin with the equation

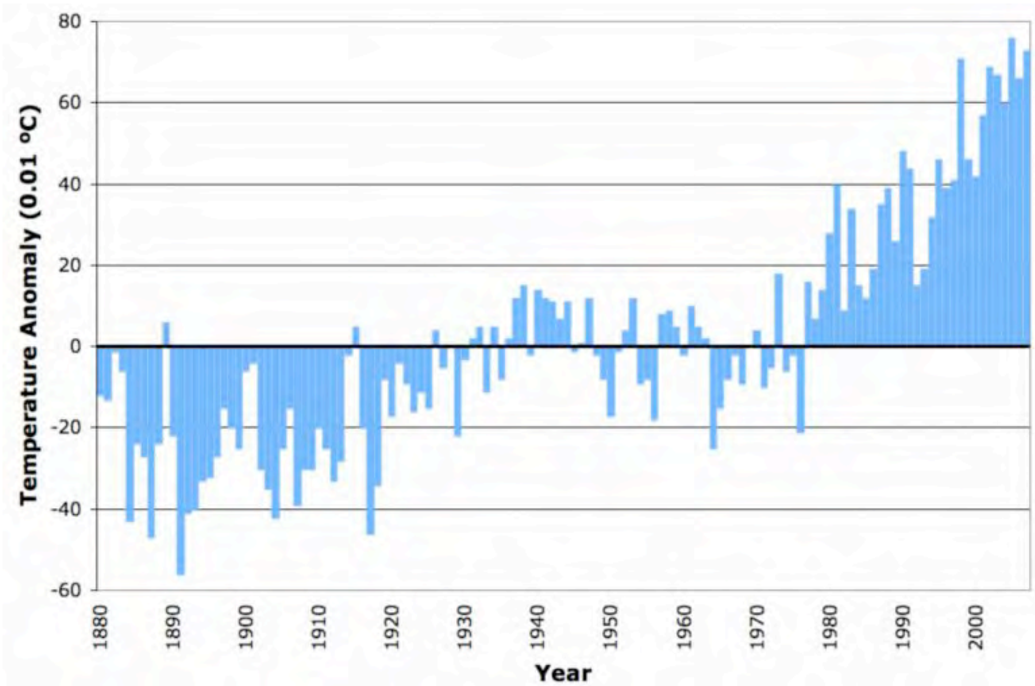


Figure 1. Anomalies in global mean temperature, 1880–2007. The temperature is scaled in 0.01°C. Data from NASA Goddard Institute for Space Studies [2008].

$$\frac{dz}{dt} = \frac{-z + (Z_0 - z)\beta\overline{\Delta T}}{\tau}, \tag{1}$$

where

z is the sea-level change (initially zero) (m),

τ is the relaxation time (years),

β is a constant representing glacier melt sensitivity to temperature change ($^\circ\mathrm{C}^{-1}$),

Z_0 is the initial ice mass in sea-level equivalent, and

$\overline{\Delta T}$ is the global mean temperature change (°C).

We set $Z_0 = 0.45$ m, based on data from Oerlemans [1989]. We use data from Wigley and Raper [1993] for the values of τ and β, and we set these parameters for various estimates of sea-level rise as follows:

Low: $(\tau, \beta) = (30, 0.10)$
Medium: $(\tau, \beta) = (20, 0.25)$
High: $(\tau, \beta) = (10, 0.45)$.

The last parameter to estimate is $\overline{\Delta T}$. This could be done by finding a best-fit curve to the temperature anomaly data of **Figure 1**. For the years after 1980, linear and logarithmic curves appear to fit the data. However, we use a temperature perturbation as an estimate for the change in annual global mean temperature, since this was implemented into models used in Oerlemans [1989]. The equation for the temperature perturbation is

$$T' = \eta(t - 1850)^3 - 0.30, \tag{2}$$

where

η is the constant 27×10^{-8} °K·yr^{-3},

t is the year,

1850 is used as a reference year in which the Earth was in a state unperturbed by global warming, and

0.30 is a vertical shift (°C).

The comparison of **(2)** to the data in **Figure 1** is given in **Figure 2**.

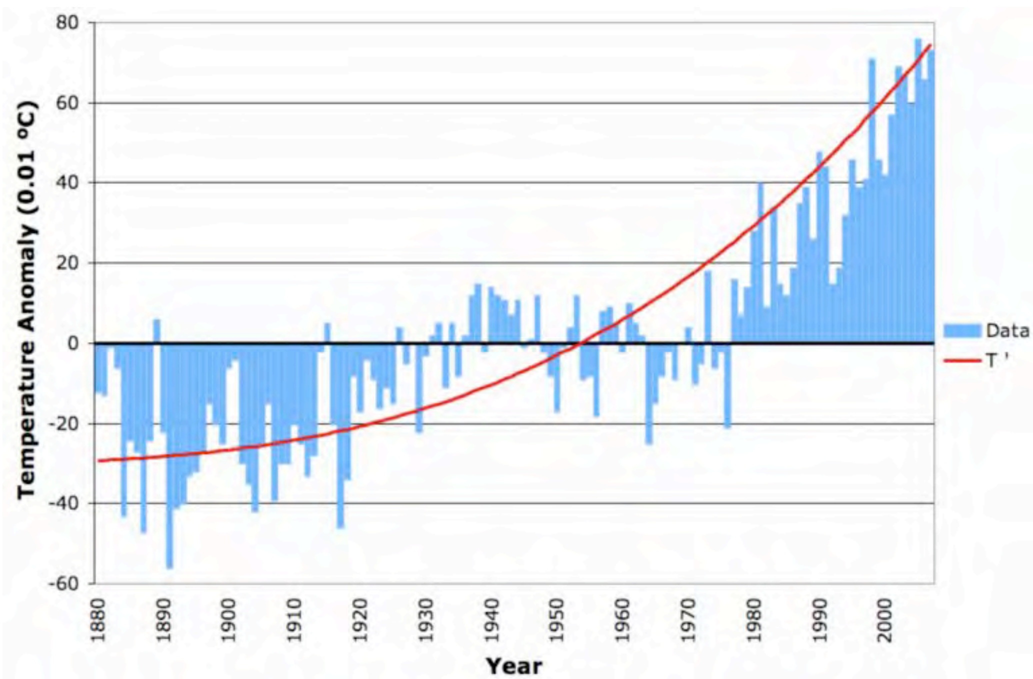


Figure 2. Comparison of T' and actual data.

While the curve is not extremely accurate to each data point, the broad shape of the trend reflects the actual change in the global mean temperature. Fitting a polynomial of high degree could match the data better, but extrapolation past 2007 could be highly inaccurate. The moderate increase in global mean temperature represented by T' is realistic for our purposes of keeping the model simple.

Now we set $\overline{\Delta T}$ equal to T' by plugging **(2)** into **(1)**, giving

$$\frac{dz}{dt} = \frac{-z + (Z_0 - z)\beta(\eta(t-1850)^3 - 0.30)}{\tau}.$$

Results of the Model

Low Sea-Level Rise

With $\tau = 30$ (the relaxation time in years) and $\beta = 0.10$ (the glacier melt sensitivity to temperature change in $°\mathrm{C}^{-1}$), a low sea-level rise is estimated. Using these parameters, there is a decrease in sea level between 1850 and 1910, then a steady increase to a change of about 0.10 m by 2100 (**Figure 3a**). This curve is concave up. Focusing on the years 2008–2058, the change in sea level ranges between 0.015 m and 0.055 m (**Figure 3b**). This curve is also concave up.

Medium Sea-Level Rise

With $\tau = 20$ and $\beta = 0.25$, a medium sea-level rise is estimated. There is a decrease in sea level between 1850 and about 1900, and then a steady increase to a change of about 0.20 m by 2100 (**Figure 3c**). The curve is concave up, with a slight possible change to concave down around 2075. For 2008–2058, the change in sea level ranges between 0.045 m and 0.13 m (**Figure 3d**). This curve is almost linear.

High Sea-Level Rise

With $\tau = 10$ and $\beta = 0.45$, a high sea-level rise is estimated. There is a decrease in sea level between 1850 and 1890, and then a steady increase to a change of about 0.275 m by 2100 (**Figure 3e**). This curve is concave up with a shift to concave down around 2025. Focusing on the years 2008–2058, the change in sea level ranges between 0.10 m and 0.21 m (**Figure 3f**). This curve is concave down.

Modeling Ice Sheets

We focus on modeling the contribution of the ice sheets in Greenland and Antarctica. There are only simple models to simulate changes in volume over time, since "existing ice-sheet models cannot simulate the widespread rapid glacier thinning that is occurring, and ocean models cannot simulate the changes in the ocean that are probably causing some of the dynamic ice thinning" [Bentley et al. 2007].

1880–2100 2008–2058

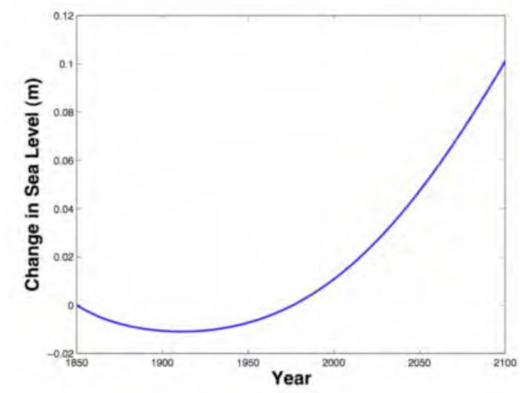


Figure 3a. $\tau = 30$, $\beta = 0.10$.

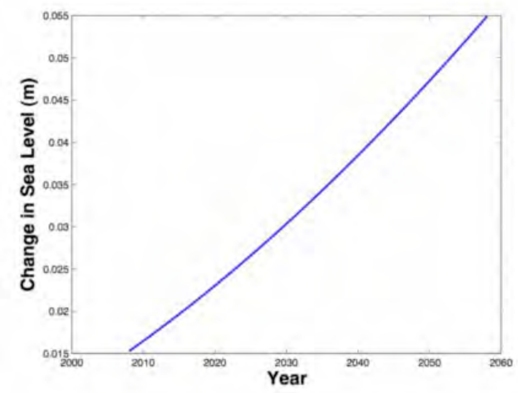


Figure 3b. $\tau = 30$, $\beta = 0.10$.

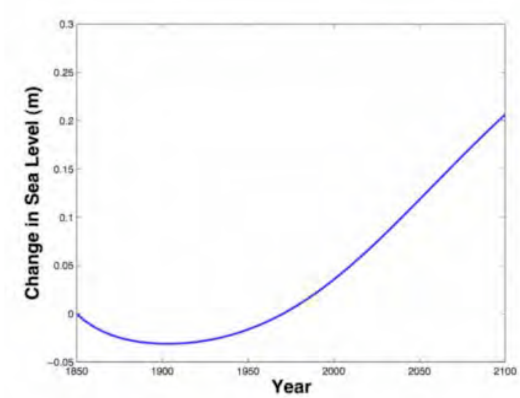


Figure 3c. $\tau = 20$, $\beta = 0.25$.

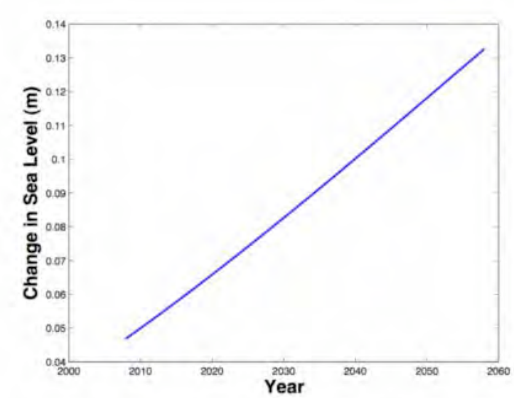


Figure 3d. $\tau = 20$, $\beta = 0.25$.

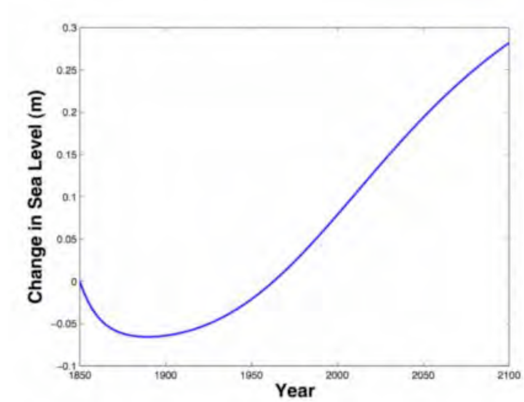


Figure 3e. $\tau = 10$, $\beta = 0.45$.

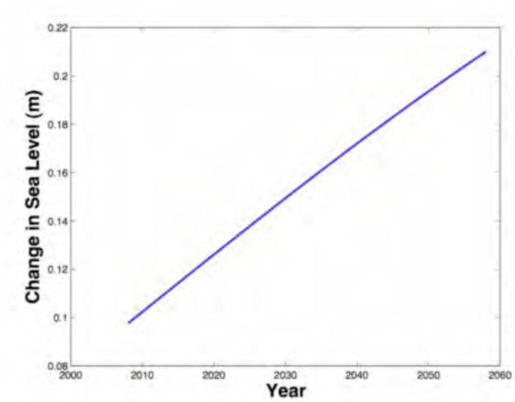


Figure 3f. $\tau = 10$, $\beta = 0.45$.

Figure 3. Change in sea level for small ice caps and glaciers, in m/yr.

Assumptions and Formation

To create a simple model of sea-level rise, we make assumptions about average volumes and ice-loss rates. These averages were taken from a number of sources that used laser measurements as well as past trends to make conclusions as accurately as possible [NASA 2008; Steffen 2008; Thomas et al. 2006]. **Table 1** lists the parameters and their values for our equation to compute the contribution to sea-level rise.

Table 1.
Parameters and their values.

Symbol	Meaning	Value	Units
A_o	Total water area of the Earth	361,132,000	km^2
A_g	Area of Greenland	1,736,095	km^2
A_a	Area of Antarctica	11,965,700	km^2
V_g	Greenland ice sheet volume	2,343,728	km^3
V_a	Antarctica ice sheet volume	26,384,368	km^3
δ_g	Greenland accumulation	26	cm/yr
δ_a	Antarctica Accumulation	16	cm/yr
λ_g	Greenland loss rate (absolute value)	238	km^3/yr
λ_a	Antarctica loss rate (absolute value)	149	km^3/yr
ρ	Fresh water density	1000	kg/m^3
μ	Glacier ice density	900	kg/m^3

The equations for volume changes and corresponding sea-level rise are based on a simple model [Parkinson 1997]. We make a few modifications to this model to show a gradual change over a time period of 50 years (starting in 2008). The basic principle is to convert the loss rates of Greenland (λ_g) and Antarctica (λ_a) into water volumes using:

$$\frac{dV_g}{dt} = \frac{V_g\mu}{\rho}\,\lambda_g, \qquad \frac{dV_a}{dt} = \frac{V_a\mu}{\rho}\,\lambda_a.$$

The total volume change from the contributions of Greenland and Antarctica is a simple matter of addition:

$$\frac{dV}{dt} = \frac{dV_g}{dt} + \frac{dV_a}{dt}.$$

To calculate the total rise in sea level, there is one more aspect to consider—thermal expansion. As water warms, it expands in volume. We calculate the total sea-level rise by adding to the rise γ due to thermal expansion (γ) (approximately 1.775 mm per year [Panel on Policy Implications of Greenhouse Warming 1992]) the rise due to losses in Greenland and Antarctica:

$$\delta = \gamma + \frac{\frac{dV}{dt}}{A_o} \times 1000.$$

Sea-level rises produced by complete melting would be 7 m (Greenland) [Arctic Climate Impact Assessment 2004] and more than 70 m (Antarctica) [Kluger 2006].

Results of the Model

The contributions of the largest ice sheets plus thermal expansion do not raise the sea level as much as might be thought: 5.7 cm after 50 years.

Limitations of the Models

We chose efficiency and simplicity over complex models that apply only to small sections of the world, since they rely heavily on factors of specific ocean temperature, salinity and depth.

Model for Small Ice Caps and Glaciers

Parameter values have uncertainty, because it is difficult to measure the exact area, volume, and sea level equivalent of the small ice caps and glaciers. The same relaxation time and sensitivity values are used for all glaciers; incorporating many individual values would be difficult, because there is no specific information regarding how response time is related to ice volume [Oerlemans 1989].

We set the only cause of the melting of small ice caps and glaciers to be changes in global mean temperature. However, the causes of previous melting have not yet been specifically determined [Oerlemans 1989], so predicting the causes of future melting is limited in scope. Many other factors, such as accumulation and ablation rates, could play a role.

Model for Ice Sheets

The most prevalent uncertainties for the ice sheets are in the loss rates, plus thermal expansion of water. Loss rates were calculated by averaging over a number of decades.

Liquid densities depend on temperature, which does not factor into this model. The density of fresh water is approximately 1,000 kg/m^3 at 4°C [SiMetric 2008], which is the value we use, since the water generated by ice sheets will be near freezing. Similarly, glacier ice density is generally between 830 kg/m^3 and 917 kg/m^3 Parkinson [1997], with an average of about 900 kg/m^3 [Menzies 2002]—the value we use.

The thermal expansion factor contributes the greatest amount of uncertainty to this particular model.

Validation of the Models

Adding the total sea-level rise for the small ice caps, glaciers, and ice sheets results in the overall total rise by 2058 of between 11 cm and 27 cm. Using 2008 as the reference year, beginning in 2018 there is a linear relationship between time and total sea-level rise, as shown in **Figure 4**.

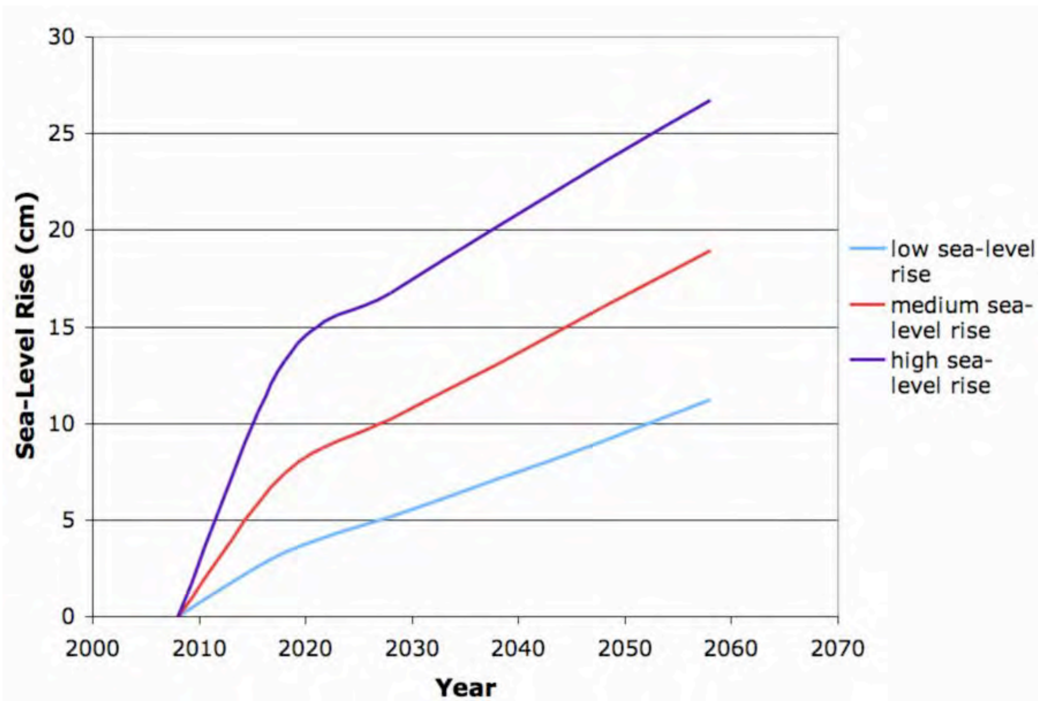


Figure 4. Change in sea level, 2008–2058, for low, medium, and high scenarios.

These results are in the range of sea-level-rise predictions from many sources:

- 50 cm in the next century [Arctic Climate Impact Assessment 2004].
- 10–30 cm by 2050 [Dow and Downing 2007].
- 1 m by 2100 [Natural Resources Defense Council 2005].
- Probabilities of increases by 2050 relative to 1985 are: 10 cm, 83%; 20 cm, 70%; and 30 cm, 55% [Oerlemans 1989]. (Note: The change from 1992 and 2007 was approximately 0.50 cm [Nerem et al. 2008].)
- 8–29 cm by 2030 and 21–71 cm by 2070 [Panel on Policy Implications of Greenhouse Warming 1992].
- 18–59 cm in the next century [U.S. Environmental Protection Agency 2008].
- 20–30 cm increase by 2050 [Wigley and Raper 1993].

Modeling the Coast of Florida

There is a direct relationship between vertical rise in the ocean and a horizontal loss of coastline. Specifically, one unit rise in the sea level corresponds to a horizontal loss of 100 units of land [Panel on Policy Implications of Greenhouse Warming 1992].

Hence, for a worst-case rise of 27 cm by 2058, we estimate the loss of 27 m of coastline. This does not appear to be as disastrous as one might think. We examine the extent of flooding.

Effects on Florida

In 2000 Florida had approximately 16 million people [Office of Economic and Demographic Research 2008]. Maps of population density, geographic relief, and topography show that about 30% of the counties in Florida are at a high risk of losing coastline; these counties also have large populations.

Many of Florida's major cities are located in these counties. We examine how much damage a retreat of 27 m of coastline would affect the cities of Cape Coral, Jacksonville, Pensacola, Miami, St. Petersburg, and Tampa.

Effects on Major Cities

In the next 50 years, most of the major cities are safe from destruction, but the outlying islands and outskirts of the cities are in danger. We measured the distance from the coastline near major cities inland to predict the extent of land that would be covered by water [Google Maps 2008].

- **Cape Coral:** Sanibel and Pine Islands would be mostly flooded, though the city center of Cape Coral would be spared.
- **Jacksonville:** Jacksonville would lose all of its coastal beaches. The city would also be in danger, depending on how much the St. Johns River rises as well. The outskirts of the city will be affected by flooding from the river.
- **Pensacola:** The harbor and the edges of the city would be covered by water, and a large portion of Gulf Breeze, Santa Rosa Island, and the Pensacola Naval Air Station would be submerged.
- **Miami:** Miami would be spared, at least for the next 50 years. Key Biscayne and Miami Beach would not be as lucky, though, and most of the Florida Keys would disappear under the ocean. However, predictions further into the future indicate that Miami will most likely be the first major city of Florida to become completely submerged.
- **St. Petersburg and Tampa:** Edges of St. Petersburg would be under water. The boundaries of Tampa would also be lost due to the surrounding

Old Tampa Bay and Hillsborough Bay. All coastal beaches, such as Treasure Island, would be mostly submerged. This area will have the largest displacement of urban population by the year 2058.

Validation of Loss of Coastal Land

The prediction of Florida coastline loss is validated by simulations from the Global Land One-km Base Elevation (GLOBE) digital elevation model (DEM) and is illustrated in **Figure 5**. The majority of Florida's population would be safe for the next 50 years, but mere loss of land is only one of the problems that would occur due to global warming.

Figure 5. The effect of 27-cm sea-level rise in Florida: The coastline of Florida that would be covered with water shown in red [Center for Remote Sensing of Ice Sheets 2008].

Other Impacts on Florida

The impacts of global warming (enhanced by the melting of the polar ice cap) on Florida could be tremendous [Natural Resources Defense Council 2001]. They include:

- overall changing climate,
- "dying coral reefs,"
- "saltwater intrusion into inland freshwater aquifers" (thus impacting groundwater),
- "an upswing in forest fires,"
- "warmer air and sea-surface temperatures,"
- "retreating and eroding shorelines,"
- health threats,
- and increased hurricane intensity.

A few of the effects are described in further detail below.

Endangered Species and Biodiversity

The World Wildlife Fund has identified the Florida Keys and Everglades, located in southern Florida where there is the greatest risk of lost coastline, as one of the Earth's "200 Most Valuable Ecoregions" [World Wildlife Fund 2008]. Wildlife, including the Florida panther, roseate spoonbill, and green sea turtle, is greatly threatened by habitat loss. Plants and animals will most likely have a difficult time adapting to new climatic conditions and stresses, and the change in biodiversity in Florida will ultimately result in problems for biodiversity in surrounding areas [Dow and Downing 2007].

Tourism

Tourism, one of Florida's biggest industries, is in extreme danger if Florida loses most of its coastline.

Health threats

As of 2000, the annual number of Disability Adjusted Life Years per million people from malnutrition, diarrhea, flooding, and malaria caused by climate-related conditions was under 10 in the United States [Dow and Downing 2007]. However, with higher global temperatures, Florida is at risk for various diseases and pests. Lyme disease is spreading in the United States and flooding of the Florida coastlines could increase the risk of cholera, typhoid, dysentery, malaria, and yellow fever [Dow and Downing 2007].

Food Production

Most of the orange and grapefruit production occurs in southern Florida, and many orchards are located along the coast, so orchards will slowly lose land. Increased salt concentration in groundwater will also threaten citrus crops [Natural Resources Defense Council 2001].

Possible Responses

Responses have been prepared to the various threats that global warming poses to the state of Florida [Florida Environment 2000; Natural Resources Defense Council 2001; U.S. Environmental Protection Agency 2008]. The U.S. Environmental Protection Agency (EPA) and the Florida state government have begun implementing some of these suggestions (marked with an asterisk in the lists below) [U.S. Environmental Protection Agency 2002].

Responses to Changing Landscape

- Limit or stop land development along coastlines.
- * Work to protect coastlines and sand dunes that could weaken and erode.
- Enact a program to prevent people living on the coast from removing vegetation and trees as well as to encourage their planting.
- Set up a fund (either state or national) to aid people in the case of an emergency evacuation due to land loss and to aid those whose businesses will be obliterated.

Responses to Changing Climate

- Improve drainage systems, to decrease flooding and to avert stagnant water (a breeding ground for mosquitoes).
- Make flotation devices a mandatory feature of all homes and businesses in flooding areas.
- Encourage the public to keep emergency preparation kits and provide suggestion lists in supply stores.*
- Build more hurricane shelters and increase standards for new buildings to withstand hurricanes.
- Put permanent fire breaks around large areas at risk of burning.

Responses to Health Threats

- Store malaria pills in preparation to combat an increased mosquito population.
- Make emergency management drills a monthly or bi-monthly event, rotating among major cities.
- Improve interoperability between fire, EMS, and police services.

Responses to Global Warming

- Provide incentives for people to lead a "green" lifestyle, e.g., free street/garage parking for hybrids, tax cuts for purchasing Energy Star products, etc.
- Use heat-reflective paint on the tops of buildings to reduce air conditioning use.
- Encourage renewable energy sources.
- * Work with corporations and companies to reduce their output of greenhouse gases.

- * Work to protect indigenous wildlife and plants as well as the unique landscape, such as the Everglades.

Action should be taken now with the worst-case scenario in mind. The most important aspect, however, is to *keep people informed*; make it clear that *land will be lost* no matter what, but people can slow down the process by becoming part of the solution.

Conclusion

Over the next 50 years, Florida will experience changes to its geography. Melting of ice sheets and glaciers and thermal expansion in the oceans will lead to a gradual rise in sea level.

The loss of land over time illustrates the seriousness of the problems of global warming. Living generations may be faced with the consequences of lost coastal land. If steps are not taken to reduce the increase in sea level, southern Florida will slowly disappear.

References

Arctic Climate Impact Assessment. 2004. *Impacts of a Warming Arctic: Arctic Climate Impact Assessment*. New York: Cambridge University Press. http://amap.no/acia/.

Bentley, Charles R., Robert H. Thomas, and Isabella Velicogna. 2007. Global outlook for ice and snow: Ice sheets. *United Nations Environment Programme* (2007): 99–114.

Center for Remote Sensing of Ice Sheets. 2008. Sea level rise maps and GIS data. https://www.cresis.ku.edu/research/data/sea_level_rise/.

Dow, Kirstin, and Thomas E. Downing, 2007. *The Atlas of Climate Change: Mapping the World's Greatest Challenge*. London: Earthscan.

Florida Environment. 2000. Public plans for sea level rise. http://www.floridaenvironment.com/programs/fe00501.htm.

Google Maps. 2008. http://maps.google.com/.

Kluger, Jeffrey. 2006. Global warming heats up. http://www.time.com/time/magazine/article/0,9171,1176980-1,00.html.

Menzies, John. 2002. *Modern and Past Glacial Environments*. Boston, MA: Butterworth-Heinemann.

National Aeronatics and Space Administration (NASA). 2008. Educational brief: Ice sheets. `http://edmall.gsfc.nasa.gov/99invest.Site/science-briefs/ice/ed-ice.html`.

NASA Goddard Institute for Space Studies. 2003. Recent warming of Arctic may affect worldwide climate. `http://www.nasa.gov/centers/goddard/news/topstory/2003/1023esuice.html`.

________. 2008. GISS surface temperature analysis. `http://data.giss.nasa.gov/gistemp/`.

Natural Resources Defense Council 2001. Global warming threatens Florida. `http://www.nrdc.org/globalwarming/nflorida.asp`.

________. 2005. Global warming puts the Arctic on thin ice. `http://www.nrdc.org/globalWarming/qthinice.asp`.

Nerem, R. Steven, Gary T. Mitchum, and Don P. Chambers. 2008. Sea level change. `http://sealevel.colorado.edu/`.

Oerlemans, Johannes. 1989. A projection of future sea level. *Climatic Change* 15: 151–174.

Office of Economic and Demographic Research: The Florida Legislature. 2008. Florida population. `http://edr.state.fl.us/population.htm`.

Panel on Policy Implications of Greenhouse Warming. 1992. *Policy Implications of Greenhouse Warming: Mitigation, Adaptation, and the Science Base*. Washington, DC: National Academy Press.

Parkinson, Claire L. 1997. Ice sheets and sea level rise. The PUMAS Collection `http://pumas.jpl.nasa.gov/`.

SiMetric. 2008. Density of water (g/cm^3) at temperatures from 0 °C (liquid state) to 30.9 °C by 0.1 °C inc. `http://www.simetric.co.uk/si_water.htm`.

Steffen, Konrad. 2008. Cyrospheric contributions to sea-level rise and variability. `http://globalwarming.house.gov/tools/assets/files/0069.pdf`.

Thomas, R., E. Frederick, W. Krabill, S. Manizade, and C. Martin 2006. Progressive increase in ice loss from Greenland. *Geophysical Research Letters* 33: L1053.

U.S. Environmental Protection Agency. 2002. Saving Florida's vanishing shores. `http://www.epa.gov/climatechange/effects/coastal/saving_FL.pdf`.

________. 2008. Coastal zones and sea level rise. `http://www.epa.gov/climatechange/effects/coastal/`.

Wigley, T.M.L., and S.C.B. Raper. 1993. Future changes in global mean temperature and sea level. In *Climate and Sea Level Change: Observations, Projections and Implications*, edited by R.A. Warrick, E.M. Barrow, and T.M.L. Wigley, 111–133. New York: Cambridge University Press.

World Wildlife Fund. 2008. South Florida. `http://www.worldwildlife.org/wildplaces/sfla/` .

Amy Evans, John Ringland (advisor), and Tracy Stepien.

Erosion in Florida: A Shore Thing

Matt Thies
Bob Liu
Zachary W. Ulissi

University of Delaware
Newark, DE

Advisor: Louis Frank Rossi

Abstract

Rising sea levels and beach erosion are an increasingly important problems for coastal Florida. We model this dynamic behavior in four discrete stages: global temperature, global sea level, equilibrium beach profiles, and applications to Miami and Daytona Beach. We use the Intergovernmental Panel on Climate Change (IPCC) temperature models to establish predictions through 2050. We then adapt models of Arctic melting to identify a model for global sea level. This model predicts a likely increase of 15 cm within 50 years.

We then model the erosion of the Daytona and Miami beaches to identify beach recession over the next 50 years. The model predicts likely recessions of 66 m in Daytona and 72 m in Miami by 2050, roughly equal to a full city block in both cases. Regions of Miami are also deemed to be susceptible to flooding from these changes. Without significant attention to future solutions as outlined, large-scale erosion will occur. These results are strongly dependent on the behavior of the climate over this time period, as we verify by testing several models.

Introduction

The northern ice cap plays an important role in global climate and oceanic conditions, including interactions with the global oceanic currents, regulation of the atmospheric temperature, and protection from solar radiation [Working Group II 2007]. There are significant recent trends in polar melting, global temperature, and global sea level.

The UMAP Journal 29 (3) (2008) 285–300.

By correlating the effects of an increasing sea level on beach erosion, we can strategically develop our coast for the future so that homes and businesses can remain untouched by disaster.

Approach

- Analyze existing arctic and climate models to determine the most reasonable predictions for future changes.
- Identify the best available models for global change.
- Relate the future trends and physical melting processes to time and predicted temperatures.
- Examine and apply the Bruun model for beach erosion.
- Establish realistic physical models and parameters of Daytona Beach and Miami.
- Model the long-term erosion of the beach shores in those beaches.
- Propose cost-effective solutions to minimize the impact of erosion.

Arctic Melting

Justified Assumptions

- The northern ice cap includes the North Polar ice cap (over seawater) and the Greenland ice sheet (over land).
- The IPCC temperature models are accurate and stable within the time period of interest.
- The melting of the North Polar ice cap does not contribute directly to global water levels.
- Tectonic considerations within the IPCC model are relevant to the coast of Florida.
- Changes in oceanic salinity cause negligible changes in sea levels.
- Changes in ocean temperature will lead directly to increases in sea level within the time period of interest.

Polar Ice Cap

The North Polar ice cap is essentially a source of fresh water. Because of its composition and unsupported status, 90% [Stendel et al. 2007] of it is largely suspended beneath the surface of the Arctic Ocean. Since the density of ice is only 10% lower than that of water (0.92 g/cm^3 vs 1.0 g/cm^3), *any melting of the North Polar ice cap contributes negligibly to global water levels*.

The primary effect of the North Polar ice cap is to regulate global and oceanic temperatures, through solar deflection and melting. As the ice cap melts further, this capability is diminished, and temperatures change. Current models for the ice cap, atmosphere, and global temperatures are complex; we capture the time-dependent effects through existing temperature predictions.

Greenland Ice Sheet

Since the Greenland ice sheet is supported on a land mass, its contribution to global climate and sea level is considerably different from the polar ice caps (which are floating ice). Melting ice from the Greenland ice sheet contributes directly to the total volume of water in the oceans. This contribution to global sea levels is not captured directly by existing temperature models and hence must be related back to historic data.

Temperature Effects

The density of water is temperature dependent. As the temperature of the oceans increase, the corresponding decrease in water density will lead to an overall increase in volume.

Salinity Changes

Since both the Greenland ice sheet and the North Polar ice cap are pure freshwater sources, any melting will result in slight reductions in the salinity of the global oceans. The two effects of this interaction are a slight change in density due to the reduced salt content and a possible decrease in the rate at which the North Polar ice cap melts (due to osmotic forces based on the salt concentrations, an effect commonly observed in chemistry).

However, according to the IPCC [Working Group II 2007], these changes are relatively small compared to the thermal effects of the warming process. Thus, these effects are included in our model through the sea level predictions of the IPCC and only applied as a direct relationship to global temperatures.

Tectonic Effects

In addition to global trends from the rising sea level, shifts within the tectonic plates of the Earth have been argued to cause an upward movement of some of the ocean bottoms, and thus contribute to local deviations in the sea level change [Nerem et al. 2006]. Such effects are outside our scope here.

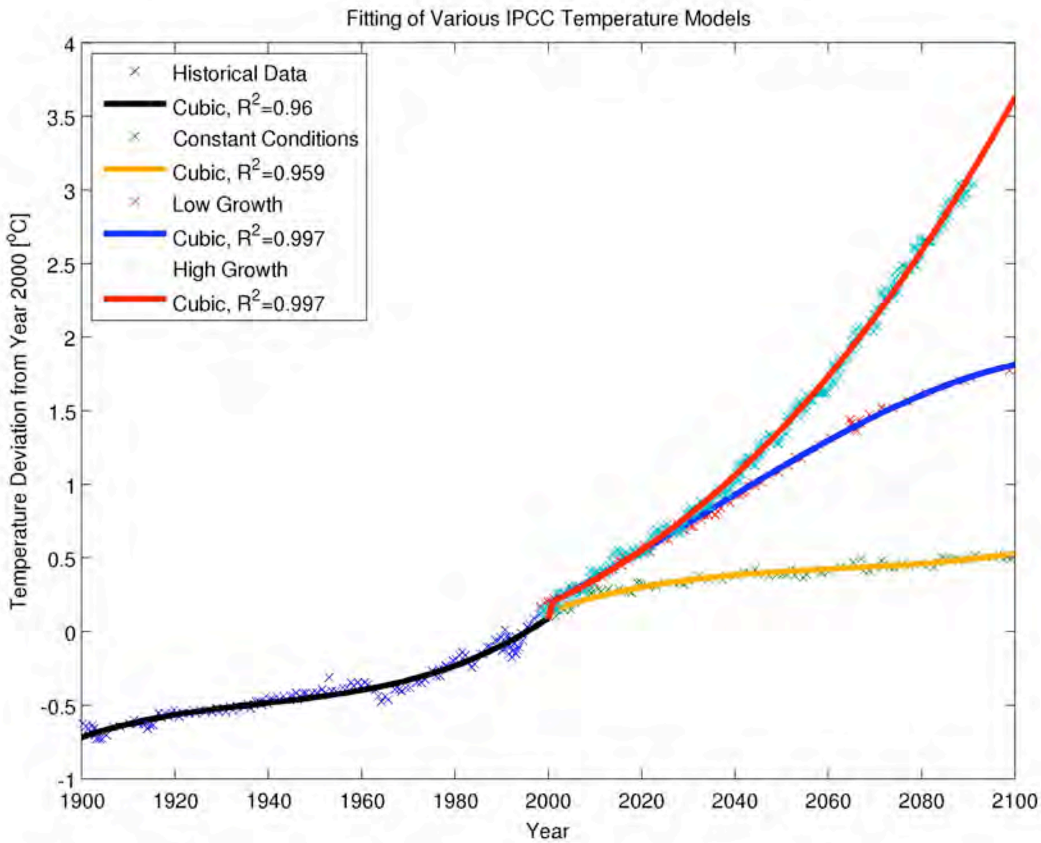


Figure 1. A global temperature model endorsed by the IPCC [Working Group II 2007]

Global Temperature Model

Many large-scale computer simulations and models have been proposed to predict the effects of arctic melting. These results have been compiled and studied by the IPCC fourth assessment report [Working Group II 2007], and its predictions for global temperature are used within this report. Criticism of IPCC modeling is common due to its simplified assumptions, but however we have not seen a better alternative.

We use the temperature models shown in **Figure 1**, which shows historical data and several scenarios for the future. We make graphical fits and show corresponding information. We conclude that simulations that use constant conditions prevailing in 2000 are unrealistic. Therefore, we consider only the low-growth and high-growth model cases. We assume a cubic growth model for temperature change:

$$\Delta T(t) = at^3 + bt^2 + ct + d.$$

Modeling Sea-Level Changes

Justified Assumptions

- The IPCC temperature and sea-level estimates are accurate.
- Sea-level change is global and equal everywhere.
- Sea-level changes can be broken into factors directly related to temperature, and factors whose rate is dependent on temperature.

Sea Level Model

While the IPCC [Working Group II 2007] predicts temperature changes for the next century, the only predictions for sea level changes are possible ranges at the end of the century. To develop time-dependent models for the sea level rise, we correlate these changes to the temperature model.

The IPCC simulations include ranges for the effects of various parameters on the global sea level change [Working Group II 2007]. These effects can be broken roughly into 55% indirect effects leading to temperature change (and thermal expansion), and 45% other volume effects, such as the melting of the Greenland ice sheet (see **Table 1**).

Table 1.
Results from the third IPCC report for 2100 [Working Group II].

Source	Sea Rise (mm)	Mean rise (mm)
Thermal expansion	110–430	270
Glaciers	10–230	130
Greenland ice	20–90	35
Antarctic ice	170–20	−95
Terrestrial storage	83–30	−26.5
Ongoing contributions from ice sheets	0–55	27.5
Thawing of permafrost	0–5	2.5
Total global-average sea level rise	110–770	440

For the 55% of changes related directly to temperature, we consider the corresponding sea level to be proportional to temperature:

$$S_1 = \gamma \Delta T(t),$$
$$\gamma = \frac{\Delta S(2100)}{\Delta T(2100)}.$$

Since the Greenland ice sheet is noticeably devoid of water (whatever melts, runs off the ice sheet), the primary limitation on ice melting is assumed to be limitations of heat transfer from the air above the ice shelf.

To model this, we use a generic heat exchanger rate equation, with an arbitrary thermal coefficient U_a. To determine the rate, we use the average summer temperature of Greenland, 6°C [Vinther et al. 2006]. We integrate the resulting equation and obtain scaling coefficients:

$$\begin{aligned}
\frac{dS_2}{dt} &\propto q = U_a(T_1 - T_2), \\
\Delta S_2 &= \alpha \int_{2000}^{t_f} U_a\big(T_1(t) - T_2\big)\,dt \\
&= \alpha \int_{2000}^{t_f} U_a\big(T + (ax^3 + bx^2 + cx + d) - 0\big)\,dt \\
\beta = \alpha U_a &= \frac{\Delta S(2100)}{\int_{2000}^{t_f}\big(T + (ax^3 + bx^2 + cx + d)\big)\,dt}.
\end{aligned}$$

We determine the scaling coefficient β for each simulation, and calculate the overall sea-level rise as follows:

$$\Delta S(t) = 0.55 S_1(t) + 0.45 S_2(t).$$

The resulting predicted sea-level rises are shown in **Figure 2**. The lower and upper bounds on the predictions are shown by calculating the rises for the lower range of the low-growth model and the upper range of the high-growth model. The predicted sea-level rises for the mean rises of both scenarios through 2050 are quite similar, and using either is sufficient. However, such engineering modeling questions often need to err on the side of caution, so we consider the upper extreme in later models. Historical data are included for comparison and agree reasonably with the predicted trends.

The predicted sea-level increases are shown in **Table 2**.

Table 2.
Model predictions for future sea level rises.

Year	Sea Level Increase (cm)		
2010	4.1	4.4	2.6
2020	6.8	7.7	4.4
2030	9.6	11.5	6.2
2040	12.5	15.6	8.0
2050	15.3	20.2	9.9

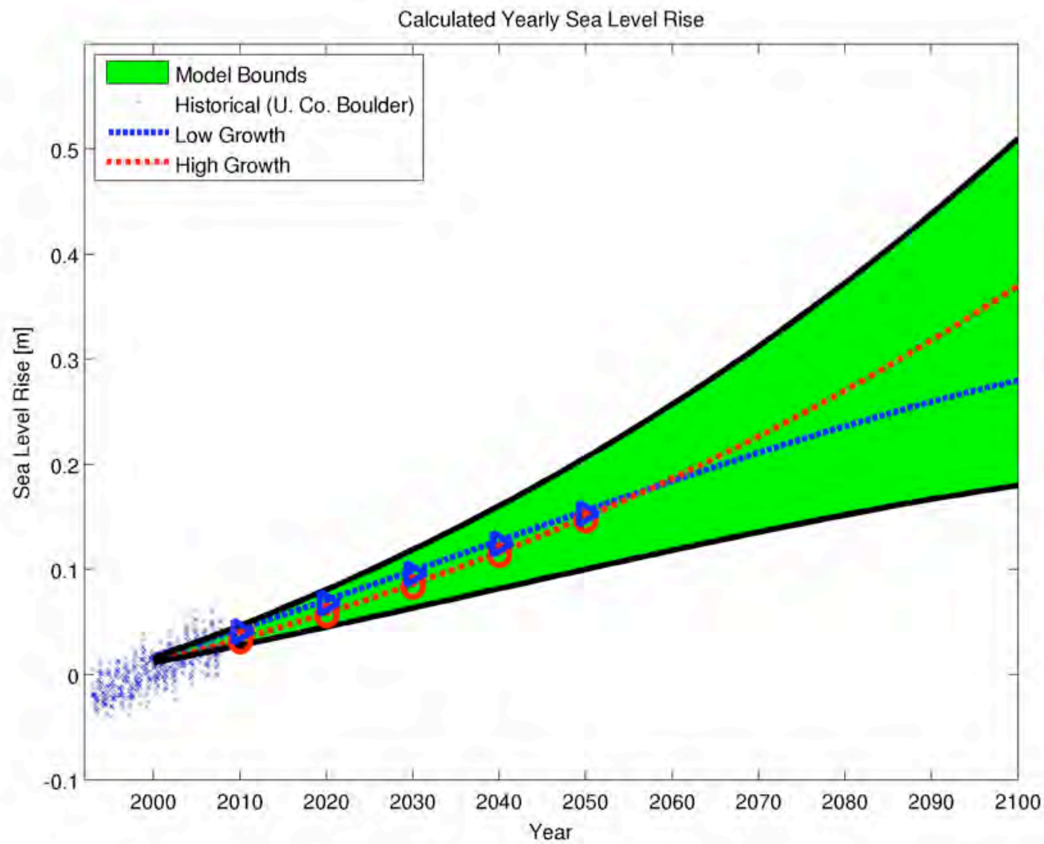


Figure 2. The model for global sea-level changes through 2100.

Beach Erosion Models

Justified Assumptions

- Beach erosion is continuous when observed over long time periods.
- Beach profiles do not change.
- Only direct cause of erosion is sea-level change.

Overview

Beach erosion is complex, since the behavior of the beach depends on a huge number of local beach and weather parameters, as well as being linked to the physical bathymetry of the surrounding sea bed.

Seasonal and Weather Effects

Seasonal temperature changes can cause differing rates of erosion, and winter weather has been observed to cause formation of offshore bars, af-

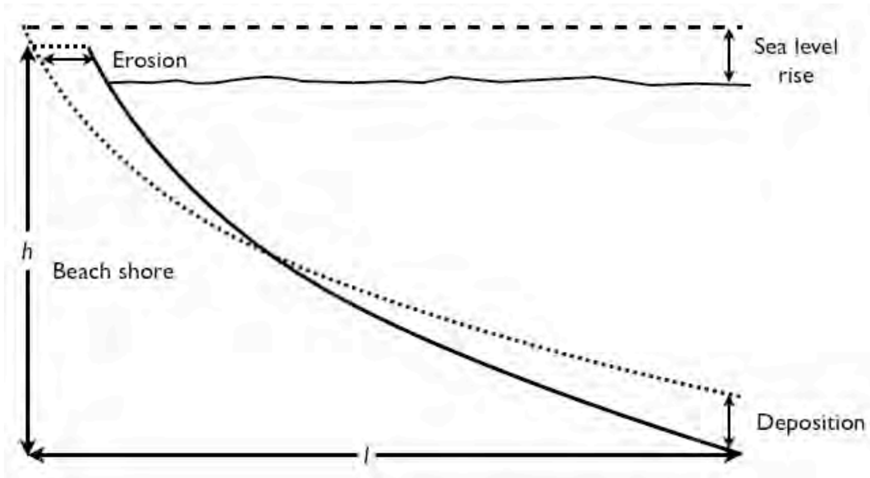


Figure 3. The Bruun model for equilibrium beach profiles [Bruun 1983].

fecting the relative rates of erosion. Storms and hurricanes generally show no lasting long-term effect on the state of a beach [Walton, Todd L. 2007].

Thus, for the purposes of this model, these effects are unimportant. Predicting weather activity is impossible on a short time scale, and attempting to simulate any sort of effects over a long (50-year) period would be unreasonable.

Bruun Model

Instead of modeling transient effects on beach erosion, we use the well-known Bruun model of beach profiles [Herbich and Bretschneider 1992]. At the core of the model is the observation that many beaches fit the general profile:

$$h(x) = Ax^{2/3},$$

where h is the depth of the water, x is the distance from the shoreline, and A is a static parameter related to the average particle size of the beach material. We illustrate the model in **Figure 3**.

Using this model, Bruun found that the ratio between the rise R in sea level and the recession ΔS of a beach front are linearly related through a constant K,

$$R = K\Delta S. \tag{1}$$

The constant K can be calculated using the long-range profile of the coast [Herbich and Bretschneider 1992] via

$$K = \frac{l}{h},$$

where l is the distance from the shoreline and h is the depth at l. We fit the parameter K and use this linear relation to predict future erosion.

Justification of Erosion Model Choice

There has been widespread criticism of the assumptions made by Bruun in his constant-profile model. However, it is the only beach erosion model to have received significant experimental testing. A thorough review of the current state of the Bruun model and additions was performed in Slott [2003], with modifications proposed by Dean in Miller and Dean [2004].

Effects on Florida

Justified Assumptions

- Beach profiles are consistent for all locations on a given beach (city location).
- The profile parameters are time-independent.

Geographical Overview

Florida sits on a shelf projected between the Atlantic Ocean and the Gulf of Mexico. The topography is characterized by extremely low elevation. There are significant urban areas situated along most of the coastline, with significant centers at Tampa Bay on the west coast and at Miami and Daytona on the East coast. In addition, barrier islands are present on much of Florida's east coast, with large implications for modeling.

Primary Effects

We consider two primary effects within our model and examine the flooding implications of a rise in sea level.

We conclude that beach erosion is be the primary effect of a rising sea level. We present these results for several scenarios.

Daytona Beach

Physical Profile

We show a topographical and bathymetric map in **Figure 4** [NOAA 2007]. The elevation is at least several meters for all major inhabited areas, so we neglect the likelihood of direct flooding from the predicted rise.

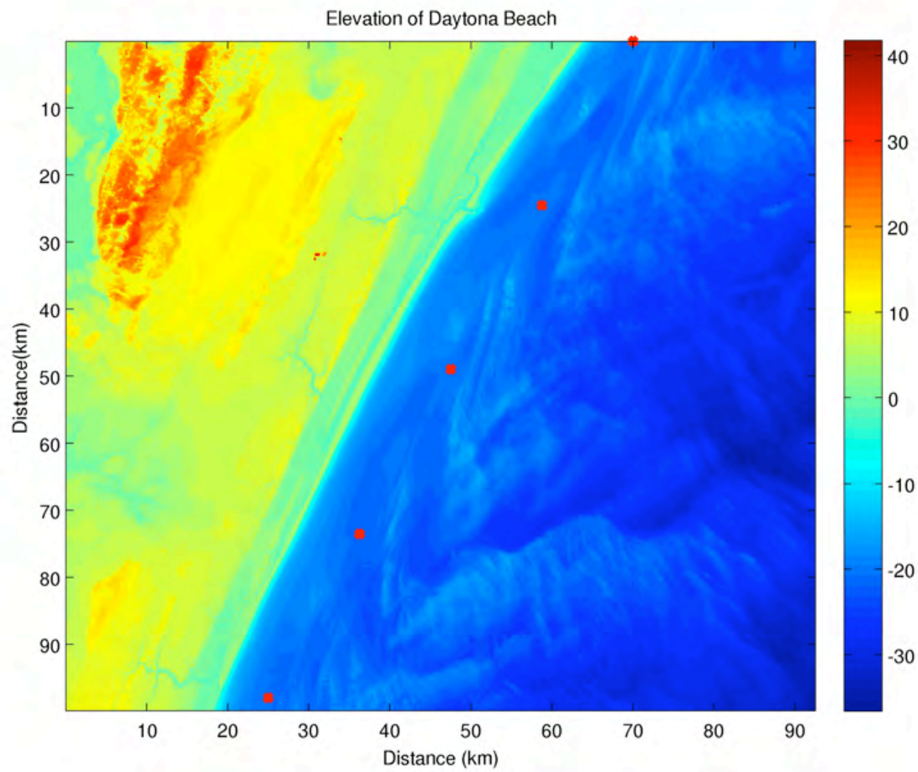


Figure 4. Topography and bathymetry of Daytona Beach, with five sampled points (in red) lying along a line from $(25\ \ 98)$ to $(70\ \ 0)$.

Beach Profile

To determine the constant K in **(1)** for Daytona Beach, we collect sample points (shown in red in **Figure 4**. We use these results with the corresponding elevation and position of the shoreline to determine the ratio as follows:

$$K_i = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{\Delta h}.$$

We show the results of this calculation for all five points in **Table 3** and arrive at a mean value $K = 452$.

We observe the effectiveness of the Bruun approximation when we fit an averaged profile for Daytona Beach (**Figure 5**).

Future Erosion of Daytona Beach

We use the sea levels in **Table 2** to calculate values for the beach recession at the necessary intervals. Daytona Beach contains a series of barrier islands, and we assume that the small separation between them and the mainland will prevent any significant erosion on the Daytona mainland.

Table 3.
Determination of the scaling coefficient K for Daytona Beach.

Point	Distance (km)	Elevation difference (m)	K
1	9.65	20.29	475.7
2	9.39	20.51	457.8
3	9.66	21.15	456.6
4	9.64	22.18	434.44
5	9.22	21.13	436.31
Mean			452 ± 17

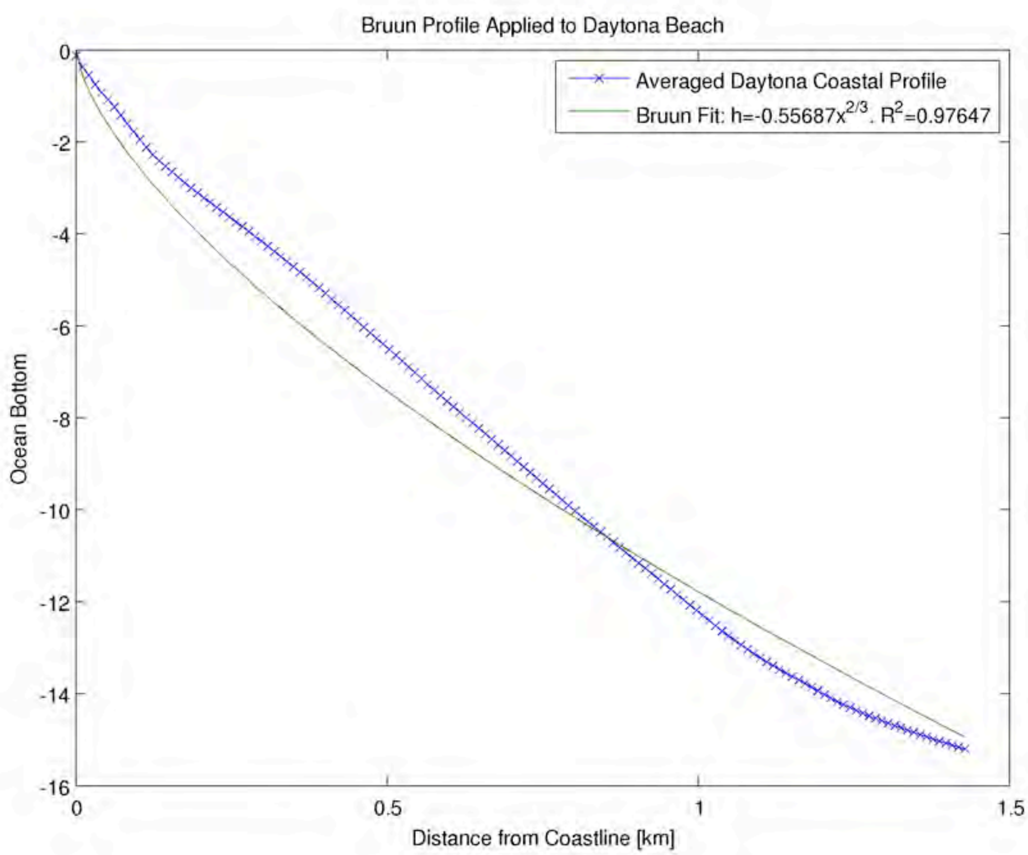


Figure 5. Appropriateness of the Bruun model.

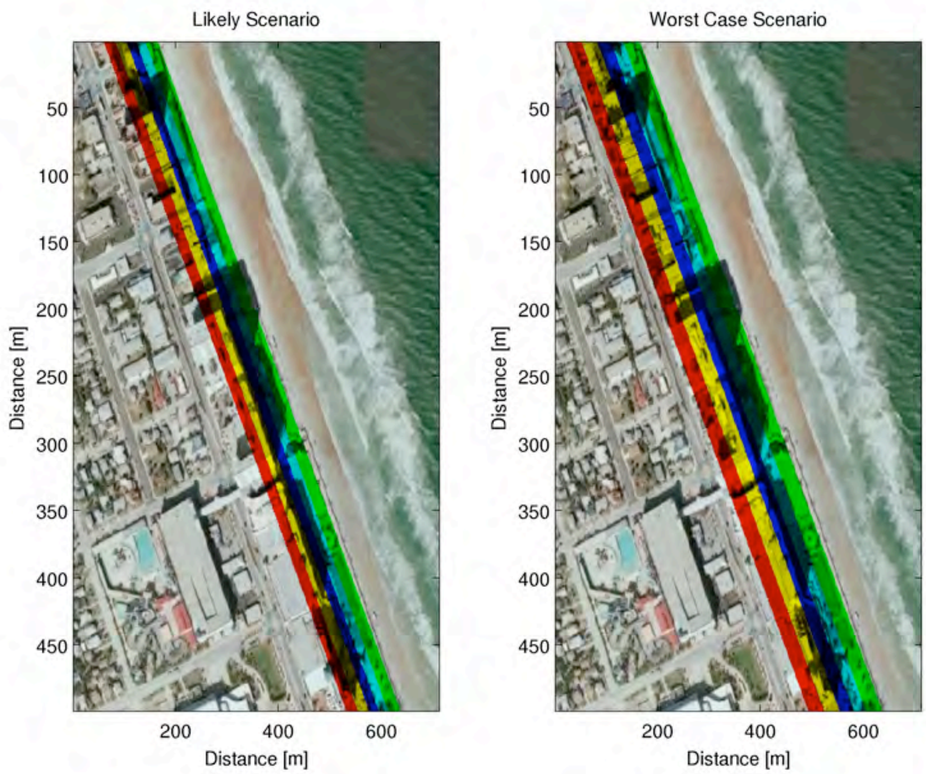


Figure 6. Effect of two climate scenarios on the erosion of Daytona Beach (Overlay: Google Earth [2008]). Shaded regions indicate increments of 10 years from 2000.

To gauge the impact of this erosion, we overlay the results for the likely- and worst-case scenarios for each decade onto a Google Earth [2008] map of Daytona Beach (**Figure 6**). *Nearly a full block width of the city will be destroyed by 2050 if no precautions are put into place.*

Miami Beach

Physical Profile

Again we work with topographical and bathymetric representations [NOAA 2008]. The low elevation of the boundaries of Miami yield problems for the city with the rise in sea level. The effects of the likely 17 cm rise in sea level are visualized in **Figure 7**.

The regions of concern are already surrounded by high walls. They should be reinforced.

Beach Profile

We determine the constant K for Miami in a similar manner to that for Daytona Beach; but rather than using multiple samples, we obtain an aver-

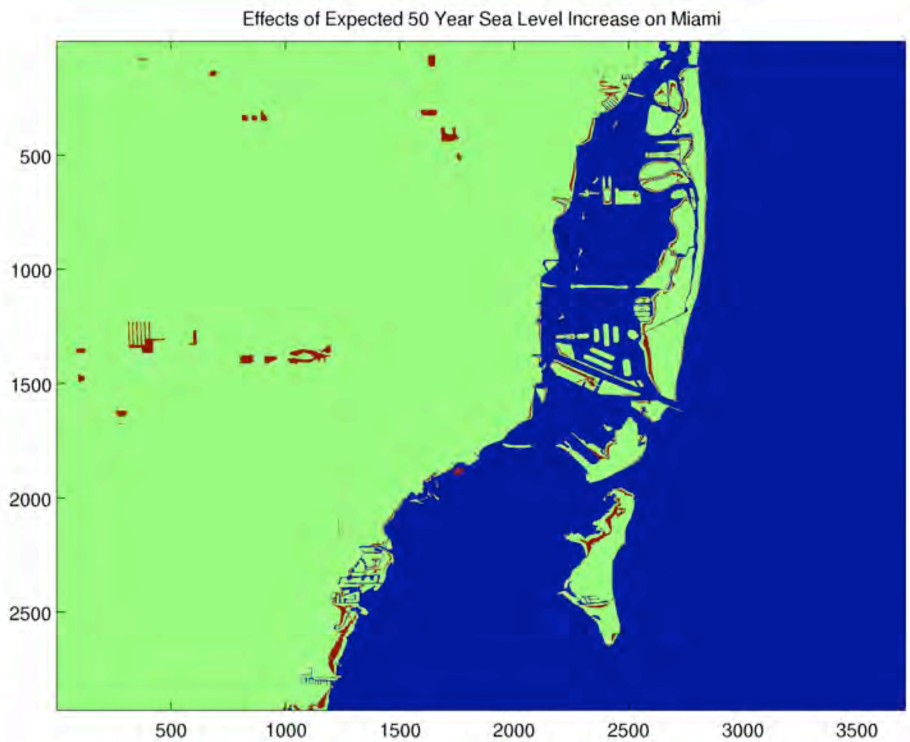


Figure 7. Regions of Miami susceptible to a 17 cm rise in sea level. Dark (blue) is existing water, light (green) is safe land, and dark (red) regions inside the light is susceptible land.

age beach profile through averaging. This results in $K = 520.83$, a higher value than for Daytona Beach, due to the significantly greater gradual slope in the coastal area just off the shore of Miami.

Future Erosion of Miami Beach

We show the results in **Figure 8**. *As with Daytona Beach, without intervention a width of nearly a city block width will be lost to the ocean.*

Common Solution for Daytona and Miami

Our beach erosion model is grounded in the observation that most beaches return to an equilibrium profile based on the average particle sizes reflected in the coefficient A. To take advantage best of the predictions of our model, we propose a solution for Miami and Daytona, based on raising the average height of the curve at the bottom of the slope to allow for a more stable beach front. This is visualized in **Figure 9**.

There are several key benefits to this design. The use of a retainer along the bottom allows the natural tendency of the waves to carry sand and sedimentation to fill in the beach naturally, without the need for costly and

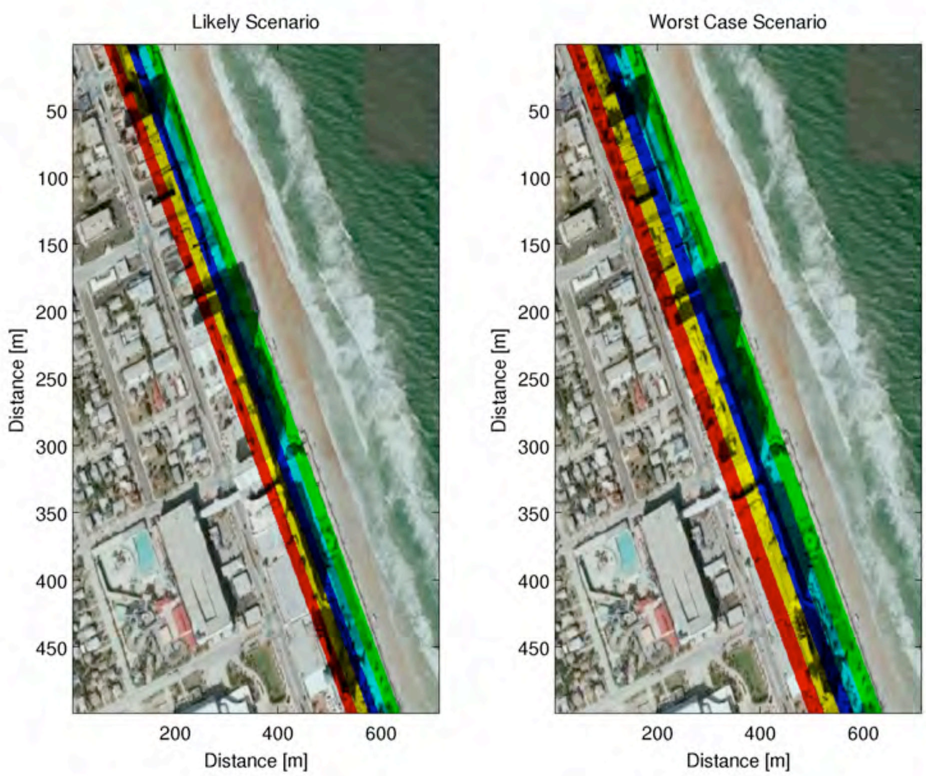


Figure 8. Effect of two climate scenarios on the erosion of Miami Beach. Shaded regions indicate increments of 10 years from 2000.

continuous additions of sand and filler. The ideal design for these retainers would be anchored concrete shapes, built to withstand the continuous force of the waves over long periods.

Conclusion

Several important conclusions can be made about future problems for the coastal cities of Florida. The sea level is definitely rising, and our model linking this activity to changes in the northern ice caps suggest an acceleration of this trend. Our model predicts a likely beach recession of 60 m by 2050, with up to 90 m possible. This recession would severely damage the first block nearest the ocean in each city unless there is intervention. Due to its lower elevation, Miami is significantly more at risk than more northern cities like Daytona, so it should be more concerned.

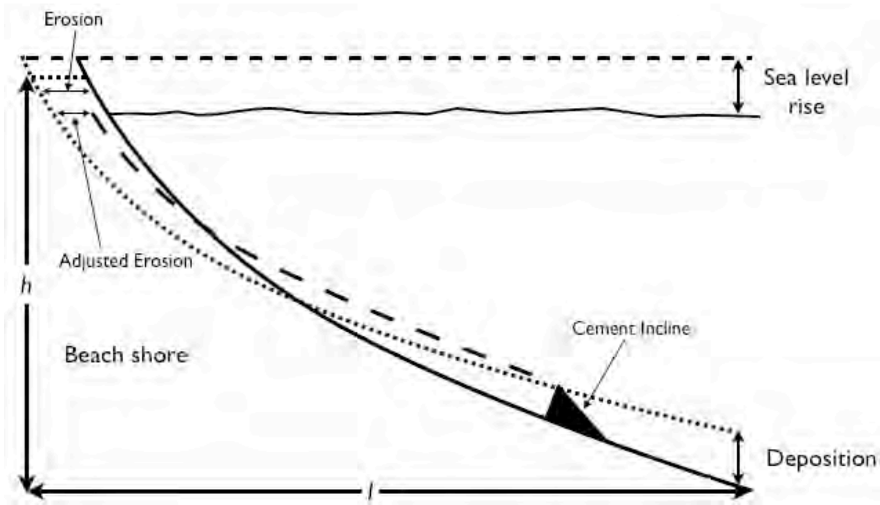


Figure 9. Proposed solution for Daytona and Miami.

References

Bruun, Per. 1983. Review of conditions for uses of the Bruun rule of erosion. *Coastal Engineering* 7 (1) (February 1983): 77–89.

Google Earth. 2008. `http://earth.google.com/` .

Herbich, John B, and Charles L. Bretschneider. 1992. *Handbook of Coastal and Ocean Engineering*. Houston: Gulf Publishing Co.

Intergovernmental Panel on Climate Change, Working Group II. 2007. *Climate Change 2001: Impacts, Adaptation and Vulnerability. Working Group II Contribution to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change.* `http://www.gtp89.dial.pipex.com/chpt.htm` .

Mccarthy, JamesJ., Osvaldo F. Canziani, Neil A. Leary, David J. Dokken, and Kasey S. White (eds.). 2001. *Climate Change 2001: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Third Assessment Report of the Intergovernmental Panel on Climate Change.* New York: Cambridge University Press. `http://www.citeulike.org/user/slow-fi/article/297638` .

Miller, J.K., and R.G. Dean. 2004. A simple new shoreline change model. *Coastal Engineering* 51: 531–556.

Nerem, Robert Steven, Eric Leuliette, and Anny Cazenave. 2006. Present-day sea-level change: A review. *Comptes rendus Géoscience* 338: 1077–1083.

National Oceanic and Atmospheric Administration (NOAA) Satellite and Information Service, National Geophysical Data Center. 2007. Daytona

Coach Lou Rossi (seated) with Mathematical Modeling teams' members (from left) senior Matthew Thies, senior Zachary Ulissi, junior Bob Liu, and freshman Kyle Thomas (kneeling).

Beach, FL 1/3 arc-second tsunami inundation DEM.
`http://www.ngdc.noaa.gov/dem/showdem.jsp?dem=Daytona%20Beach&state=FL&cell=1/3%20arc-second`.

________. 2008. Topographical maps of Florida.

Slott, Jordan. 2003. Shoreline response to sea-level rise: Examining the Bruun rule. Technical report. Nicholas School of the Environment and Earth Sciences, Department of Earth and Ocean Sciences, Duke University, Durham, NC.

Stendel, Martin, Vladimir E. Romanovsky, Jens H. Christensen, and Tatiana Sazonova. 2007. Using dynamical downscaling to close the gap between global change scenarios and local permafrost dynamics. *Global and Planetary Change* 56: 203–214.

Vinther, B.M., K.K. Andersen, P.D. Jones, K.R. Briffa, and J. Cappelen. 2006. Extending Greenland temperature records into the late eighteenth century. *Journal of Geophysical Research* 111, D11105, doi:10.1029/2005JD006810.
`http://www.agu.org/pubs/crossref/2006/2005JD006810.shtml`.

Walton, Todd L., Jr.. 2007. Projected sea level rise in Florida. *Ocean Engineering* 34: 1832–1840.

Judge's Commentary: The Polar Melt Problem Papers

John L. Scharf
Dept. of Mathematics, Engineering, and Computer Science
Carroll College
Helena, MT 59625
jscharf@carroll.edu

Introduction

The 2008 Polar Melt Problem presented teams with the challenge to model the effects over the next 50 years on the coast of Florida from melting of the North Polar ice cap due to the predicted increases in global temperatures. Teams were to pay particular attention to large metropolitan areas and propose appropriate responses to the effects predicted by their models. Teams were also encouraged to present a careful discussion of the data used.

From the judges' perspectives, this problem was especially interesting but at the same time somewhat challenging to judge, because of the wide variety in points of focus that the teams could choose to take: the physics of the model and the physical impacts of rising sea levels on coastal areas; indirect effects such as increases in the frequency and severity of hurricanes; and environmental, societal, and/or economic impacts. Regardless of the choice of focus selected by a team, in the final analysis it was good modeling that allowed the judges to discern the outstanding papers.

Judging

Judging of the entries occurs in three stages. The first stage is Triage, where a judge spends approximately 10 min on each paper. In Triage, a complete and concise Executive Summary is critically important because this is what the triage judges primarily use to pass first judgment on an entry. In reviewing the Executive Summary, judges look to see indications that the paper directly responds to the problem statement, that it uses good modeling practice, and

The UMAP Journal 29 (3) (2008) 301–304.

that the mathematics is sound. Because of the limited time that the triage judges spend on each paper, it is very likely that some potentially good papers get cut from advancing in the competition because of poor Executive Summaries. The importance of a good Executive Summary cannot be overstated.

For those papers that make it past triage, the remaining two stages of judging are the Preliminary Rounds and the Final Judging. In the Preliminary Rounds, the judges read the body of the paper more carefully. The overriding question on the mind of most judges is whether or not the paper addresses the problem and whether it answers all of the specific questions. Papers that rate highly are those that directly respond to the problem statement and specific questions, clearly and concisely show the modeling process, and give concrete results with some analysis of their validity and reliability.

In the Final Judging, the judges give very careful consideration of the methods and results presented. The features that judges look for in an Outstanding Paper are:

- a summary of results and their ramifications;
- a complete and comprehensive description of the model, including assumptions and the refinements that were made during development;
- a mathematical critique of the model, including sensitivity analysis and a description of its strengths and weaknesses; and
- recommendations for possible further work to improve the model.

The judges select as Outstanding the papers best in including and presenting each of these features.

The Papers: The Good

Specifically for the "Take a Bath" problem, the judges identified a number of positive characteristics in the submitted papers. While many teams used regressions on historical sea-level data to predict future sea levels, the papers that were viewed more favorably were those that modeled the melting of the ice and its effects. Some even included thermal expansion of the water due to rising temperatures, and many recognized that melting of the floating portions of the North Polar ice cap would have much less impact than the melting of the ice supported by land in Greenland. While there was a wide range in the sea-level increases predicted by the models, many teams bounded their results using estimates of the total rise in sea levels worldwide if all the ice on Greenland were to melt. This estimate is widely available in the literature, and it enabled many of the teams to make judgments about what increases in sea levels might be reasonable (or unreasonable) to expect over the next 50 years.

The judges also favored papers that adequately addressed the impacts on Florida, especially in the metropolitan areas. Some of these papers predicted large increases in sea levels and showed how the major cities would be impacted, whether it was only on structures near the coasts or in widespread flooding of the urban area. Others predicted small increases in sea levels, in which case the impacts were often limited to increased beach erosion and/or salt water intrusion into fresh water in the ground and on the surface. Good papers also proposed appropriate responses to the effects, whether they were great or small. Other important considerations that some teams investigated were the potential impact of larger and more frequent hurricanes and the impact of rising sea levels on the natural environment in Florida, particularly on the Everglades.

The Papers: The Bad

In some of the submitted papers, the judges also identified negative characteristics that should generally be avoided in good mathematical modeling and reporting. These items can detract from a paper that might otherwise be a good paper, and they may even result in removal of a potentially good paper from further contention:

- Some teams used regression and curve-fitting to develop a model from existing data, and then used the model to extrapolate over the next 50 years. The functions chosen for regression often had no rational basis for fitting the data. As one judge pointed out, "sixth degree polynomials rarely occur in nature." Extrapolation beyond the domain of the regression data must always be used with extreme caution, especially when there is no physical or other rational justification for the regression function in the context of the problem.
- While many of the teams did a good literature search to support their work, others used sources that were questionable. Before they are considered for use in a project, sources of information and data should always be critically judged as to their veracity, validity, and reliability.
- Some teams presented results to a degree of precision that is not appropriate. For example, one paper reported the predicted rise in sea level to a precision of eight significant digits. Modelers must always be cognizant of what degree of precision is appropriate for a given situation.
- Finally, some teams were not careful with units. Units should always be included and should be checked for correctness.

How a team addresses details like those listed here can make a big difference in how a judge rates a paper. Paying proper attention to such details in a team's report can help ensure that an otherwise worthy paper advances in the competition.

Conclusion

By and large, the judges were pleased with the overall quality of the papers submitted for the Polar Melt Problem in the 2008 MCM. Selecting the final Outstanding papers was especially difficult this year because so many of the papers were of high quality and they were competitive. As always, the judges are excited when they see papers that bring new ideas to a problem and go beyond looking up and applying models that are available in the literature. This year the judges had much to be excited about.

About the Author

John L. Scharf is the Robert-Nix Professor of Engineering and Mathematics at Carroll College in Helena, MT. He earned a Ph.D. in structural engineering from the University of Notre Dame, an M.S. degree in structural engineering from Columbia University, and a B.A. in mathematics from Carroll College. He has been on the Carroll College faculty since 1976 and served as Chair of the Department of Mathematics, Engineering, and Computer Science from 1999 to 2005. He also served as Interim Vice President for Academic Affairs during the 2005–06 academic year. He has served as an MCM judge in every year but one since 1996.

Pp. 305–362 can be found on the *Tools for Teaching 2008* CD-ROM.

A Difficulty Metric and Puzzle Generator for Sudoku

Christopher Chang
Zhou Fan
Yi Sun

Harvard University
Cambridge, MA

Advisor: Clifford H. Taubes

Abstract

We present here a novel solution to creating and rating the difficulty of Sudoku puzzles. We frame Sudoku as a search problem and use the expected search time to determine the difficulty of various strategies. Our method is relatively independent from external views on the relative difficulties of strategies.

Validating our metric with a sample of 800 puzzles rated externally into eight gradations of difficulty, we found a Goodman-Kruskal γ coefficient of 0.82, indicating significant correlation [Goodman and Kruskal 1954]. An independent evaluation of 1,000 typical puzzles produced a difficulty distribution similar to the distribution of solve times empirically created by millions of users at `http://www.websudoku.com`.

Based upon this difficulty metric, we created two separate puzzle generators. One generates mostly easy to medium puzzles; when run with four difficulty levels, it creates puzzles (or *boards*) of those levels in 0.25, 3.1, 4.7, and 30 min. The other puzzle generator modifies difficult boards to create boards of similar difficulty; when tested on a board of difficulty 8,122, it created 20 boards with average difficulty 7,111 in 3 min.

Introduction

In Sudoku, a player is presented with a 9×9 grid divided into nine 3×3 regions. Some of the 81 cells of the grid are initially filled with digits

The UMAP Journal 29 (3) (2008) 305–326.

between 1 and 9 such that there is a unique way to complete the rest of the grid while satisfying the following rules:

1. Each cell contains a digit between 1 and 9.
2. Each row, column, and 3×3 region contains exactly one copy of the digits $\{1, 2, \ldots, 9\}$.

A *Sudoku puzzle* consists of such a grid together with an initial collection of digits that guarantees a unique final configuration. Call this final configuration a *solution* to the puzzle. The goal of Sudoku is to find this unique solution from the initial board.

Figure 1 shows a Sudoku puzzle and its solution.

			7	9			5	
3	5	2			8		4	
							8	
	1			7				4
6			3		1			8
9				8			1	
	2							
	4		5			8	9	1
	8			3	7			

8	6	1	7	9	4	3	5	2
3	5	2	1	6	8	7	4	9
4	9	7	2	5	3	1	8	6
2	1	8	9	7	5	6	3	4
6	7	5	3	4	1	9	2	8
9	3	4	6	8	2	5	1	7
5	2	6	8	1	9	4	7	3
7	4	3	5	2	6	8	9	1
1	8	9	4	3	7	2	6	5

Figure 1. Sudoku puzzle and solution from the London *Times* (16 February 2008) [Sudoku n.d.].

We cannot have 8, 3, or 7 appear anywhere else on the bottom row, since each number can show up in the bottommost row only once. Similarly, 8 cannot appear in any of the empty squares in the lower left-hand region.

Notation

We first introduce some notation. Number the rows and columns from 1 to 9, beginning at the top and left, respectively, and number each 3×3 region of the board as in **Figure 2**.

We refer to a cell by an ordered pair (i, j), where i is its row and j its column, and *group* will collectively denote a row, column, or region.

Given a Sudoku board B, define the *Sudoku Solution Graph* (SSG) $S(B)$ to be the structure that associates to each cell in B the set of digits currently thought to be candidates for the cell. For example, in **Figure 1**, cell $(9, 9)$ cannot take the values $\{1, 3, 4, 7, 8, 9\}$ because it shares a group with cells with these values. Therefore, this cell has values $\{2, 5, 6\}$ in the corresponding SSG.

1	2	3
4	5	6
7	8	9

Figure 2. Numbering of 3×3 regions of a Sudoku board.

To solve a Sudoku board, a player applies strategies, patterns of logical deduction (see the **Appendix**). We assume the SSG has been evaluated for every cell on the board before any strategies are applied.

Problem Background

Most efforts on Sudoku have been directed at solving puzzles or analyzing the computational complexity of solving Sudoku [Lewis 2007, Eppstein 2005, and Lynce and Ouaknine 2006]. Sudoku can be solved extremely quickly via reduction to an exact cover problem and an application of Knuth's Algorithm X [2000]. However, solving the $n^2 \times n^2$ generalization of Sudoku is known to be NP-complete [Yato 2003].

We investigate:

1. Given a puzzle, how does one define and determine its difficulty?

2. Given a difficulty, how does one generate a puzzle of this difficulty?

While generating a valid Sudoku puzzle is not too complex, the non-local and unclear process of deduction makes determining or specifying a difficulty much more complicated.

Traditional approaches involve rating a puzzle by the strategies necessary to find the solution, while other approaches have been proposed by Caine and Cohen [2006] and Emery [2007]. A genetic algorithms approach found some correlation with human-rated difficulties [Mantere and Koljonen 2006], and Simonis presents similar findings with a constraint-based rating [2005]. However, in both cases, the correlation is not clear.

Puzzle generation seems to be more difficult. Most existing generators use complete search algorithms to add numbers systematically to cells in a grid until a unique solution is found. To generate a puzzle of a given difficulty, this process is repeated until the desired difficulty is achieved. This is the approach found in Mantere and Koljonen [2006], while Simonis [2005] posits both this and a similar method based on removal of cells from a completed board. Felgenhauer and Jarvis [2005] calculate the number of valid Sudoku puzzles.

We present a new approach. We create `hsolve`, a program to simulate how a human solver approaches a puzzle, and present a new difficulty metric based upon `hsolve`'s simulation of human solving behavior. We propose two methods based on `hsolve` to generate puzzles of varying difficulties.

Problem Setup

Difficulty Metric

We create an algorithm that takes a puzzle and returns a real number that represents its abstract "difficulty" according to some metric. We base our definition of difficulty on the following general assumptions:

1. The amount of time for a human to solve a puzzle increases monotonically with difficulty.
2. Every solver tries various strategies. To avoid the dependence of our results on a novice's ignorance of strategies and to extend the range of measurable puzzles, we take our hypothetical solver to be an expert.

Hence, we define the *difficulty* of a Sudoku puzzle to be *the average amount of time that a hypothetical Sudoku expert would spend solving it*.

Puzzle Generation

Our main goal in puzzle generation is to produce a valid puzzle of a given desired difficulty level that has a unique solution. We take a sample of 1,000 Sudoku puzzles and assume that they are representative of the difficulty distribution of all puzzles. We also endeavor to minimize the complexity of the generation algorithm, measured as the expected execution time to find a puzzle of the desired difficulty level.

A Difficulty Metric

Assumptions and Metric Development

To measure the time for an expert Sudoku solver to solve a puzzle, there are two possibilities:

1. Model the process of solving the puzzle.
2. Find some heuristic for board configurations that predicts the solve time.

There are known heuristics for difficulty of a puzzle—for example, puzzles with a small number of initial givens are somewhat harder than most. However, according to Hayes [2006], the overall correlation is weak.

Therefore, we must model the process of solving. We postulate the following assumptions for the solver:

1. Strategies can be ranked in order of difficulty, and the solver always applies them from least to most difficult. This assumption is consistent with the literature. We use a widely accepted ranking of strategies described in the **Appendix**.
2. During the search for a strategy application, each ordering of possible strategy applications occurs with equal probability. There are two components of a human search for a possible location to apply a strategy: complete search and intuitive pattern recognition. While human pattern recognition is extremely powerful (see, for example, Cox et al. [1997]), it is extremely difficult to determine its precise consequences, especially due to possible differences between solvers. Therefore, we do not consider any intuitive component to pattern recognition and restrict our model to a complete search for strategy applications. Such a search will proceed among possible applications in the random ordering that we postulate.

We define a *possible application* of a strategy to be a configuration on the board that is checked by a human to determine if the given strategy can be applied; a list of exactly which configurations are checked varies by strategy and is given in the **Appendix**. We model our solver as following the algorithm `HumanSolve` defined as follows:

Algorithm `HumanSolve` repeats the following steps until there are no remaining empty squares:

1. Choose the least difficult strategy that has not yet been searched for in the current board configuration.
2. Search through possible applications of any of these strategies for a valid application of a strategy.
3. Apply the first valid application found.

We take the difficulty of a single run of HumanSolve to be *the total number of possible applications that the solver must check*; we assume that each check takes the same amount of time. Multiple runs of this method on the same puzzle may have different difficulties, due to different valid applications being recognized first.

For a board B, its *difficulty metric* $m(B)$ is *the average total number of possible applications checked by the solver* while using the `HumanSolve` algorithm.

`hsolve` and Metric Calculation

To calculate $m(B)$, we use `hsolve`, a program in Java 1.6 that simulates `HumanSolve` and calculates the resulting difficulty:

1. Set the initial difficulty $d = 0$.
2. Repeat the following actions in order until B is solved or the solver cannot progress:
   (a) Choose the tier of easiest strategies S that has not yet been searched for in the current board configuration.
   (b) Find the number p of possible applications of S.
   (c) Find the set V of all valid applications of S and compute the size v of V.
   (d) Compute $E(p, v)$, the expected number of possible applications that will be examined before a valid application is found.
   (e) Increment d by $E(p, v) \times t$, where t is the standard check time. Pick a random application in V and apply it to the board.
3. Return the value of d and the final solved board.

While `hsolve` is mostly a direct implementation of `HumanSolve`, it does not actually perform a random search through possible applications; instead, it uses the expected search time $E(p, v)$ to simulate this search. The following lemma gives an extremely convenient closed-form expression for $E(p, v)$ that we use in `hsolve`.

Lemma. *Assuming that all search paths through p possible approaches are equally likely, the expected number $E(p, a)$ of checks required before finding one of v valid approaches is given by*

$$E(p, v) = \frac{p+1}{v+1}.$$

Proof: For our purposes, to specify a search path it is enough to specify the v indices of the valid approaches out of p choices, so there are $\binom{p}{v}$ possible search paths. Let I be the random variable equal to the smallest index of a valid approach. Then, we have

$$E(p, v) = \sum_{i=1}^{p-v+1} iP(I = i) = \sum_{i=1}^{p-v+1} \sum_{j=i}^{p-v+1} P(I = j) = \sum_{i=1}^{p-v+1} P(I \geq i)$$
$$= \frac{1}{\binom{p}{v}} \sum_{i=1}^{p-v+1} \binom{p+1-i}{v} = \frac{1}{\binom{p}{v}} \sum_{j=0}^{p-v} \binom{v+j}{v} = \frac{\binom{p+1}{v+1}}{\binom{p}{v}} = \frac{p+1}{v+1},$$

where we've used the "hockeystick identity" [AoPS Inc. 2007]. □

Given a puzzle B, we calculate $m(B)$ by running `hsolve` several times and take the average of the returned difficulties. Doing 20 runs per puzzle gives a ratio of standard deviation to mean of $\frac{\sigma}{\mu} \approx \frac{1}{10}$, so we use 20 runs per puzzle.

Analysis

Our evaluation of `hsolve` consists of three major components:

1. Checking that `hsolve`'s conception of difficulty is correlated with existing conceptions of difficulty.
2. Comparing the distribution of difficulties generated by `hsolve` to established distributions for solve time.
3. Finding the runtime of the algorithm.

Validation Against Existing Difficulty Ratings

For each of the difficulty ratings in {`supereasy`,`veryeasy`,`easy`,`medium`, `hard`,`harder`,`veryhard`,`superhard`}, we downloaded a set of 100 puzzles from Hanssen [n.d.]. No other large datasets with varying difficulty ratings were available.

We ran `hsolve` on each puzzle 20 times and recorded the average difficulty for each board. We classified boards by difficulty on a ranking scale, with 8 groups of 100 puzzles. **Table 1** shows the results.

Table 1.

Results: $\chi^2 = 6350$ (df $= 49$), $\gamma = 0.82$.

Difficulty	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
`supereasy`	81	19	0	0	0	0	0	0
`veryeasy`	19	68	12	1	0	0	0	0
`easy`	0	8	38	33	18	2	1	0
`medium`	0	2	26	29	22	17	4	0
`hard`	0	2	10	19	20	30	11	8
`harder`	0	0	5	7	22	26	36	4
`veryhard`	0	1	9	7	16	13	27	27
`superhard`	0	0	0	4	2	12	21	61

A χ^2-test for independence gives $\chi^2 = 6350$ ($p < 0.0001$). Thus, there is a statistically significant deviation from independence.

Furthermore, the Goodman-Kruskal coefficient is $\gamma = 0.82$ is relatively close to 1, indicating a somewhat strong correlation between our measure of difficulty and the existing metric. This provides support for the validity of our metric; more precise analysis seems unnecessary because we are only checking that our values are close to those of others.

Validation of Difficulty Distribution

When run 20 times on each of 1,000 typical puzzles from Lenz [n.d.], `hsolve` generates the distribution for measured difficulty shown in **Figure 3**. The distribution is sharply peaked near 500 and has a long tail towards higher difficulty.

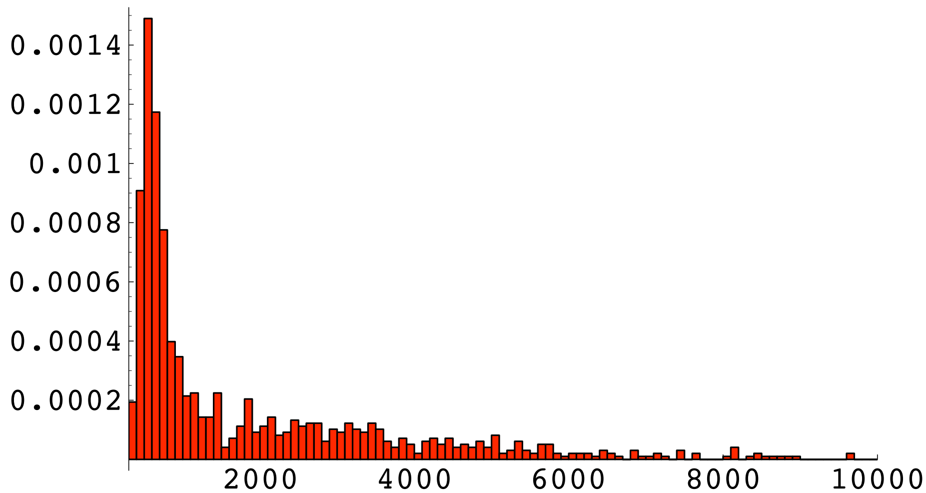


Figure 3. Histogram of measured difficulty for 1,000 typical puzzles.

We compare this difficulty distribution plot with the distributions of times required for visitors to `http://www.websudoku.com` to solve the puzzles available there [Web Sudoku n.d.]. This distribution, generated by the solution times of millions of users, is shown in **Figure 4**.

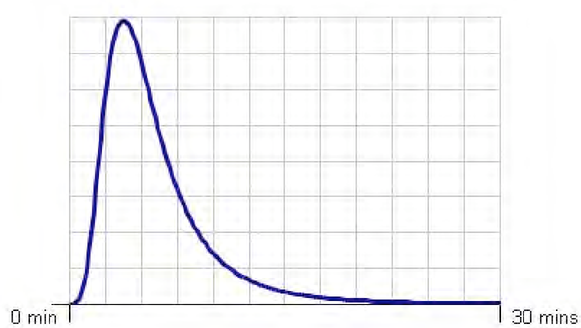


Figure 4. A distribution plot of the time to solve Easy-level puzzles on `www.websudoku.com`; the mean is 5 min 22 sec.

The two graphs share a peak near 0 and are skewed to the right.

Runtime

With running 20 iterations of `hsolve` per puzzle, rating 100 puzzles requires 13 min, or about 8 sec per puzzle, on a 2 Ghz Centrino Duo processor with 256 MB of Java heap space. While this runtime is slower than existing difficulty raters, we feel that `hsolve` provides a more detailed evaluation of difficulty that justifies the extra time.

Generator

Our choice of using a solver-based metric for difficulty has the following implications for puzzle generation:

- It is impossible to make a very accurate prediction of the difficulty of the puzzle in the process of generating it, before all of the numbers on the puzzle have been determined. This is because adding or repositioning a number on the board can have a profound impact on which strategies are needed to solve the puzzle.

  Thus, given a difficulty, we create a puzzle-generating procedure that generates a puzzle of approximately the desired difficulty and then runs `hsolve` on the generated puzzle to determine if the actual difficulty is the same as the desired difficulty. This is the approach that we take in both the generator and pseudo-generator described below.

- There is an inevitable trade-off between the ability to generate consistently difficult puzzles and the ability to generate truly random puzzles. A generator that creates puzzles with as randomized a process as possible is unlikely to create very difficult puzzles, since complex strategies would not be employed very often.

  Hence, for a procedure that consistently generates hard puzzles, we must either reduce the randomness in the puzzle-generating process or limit the types of puzzles that can result.

- The speed at which puzzles can be generated depends upon the speed of `hsolve`.

We describe two algorithms for generating puzzles: a standard generator and a pseudo-generator.

Standard Generator

Our standard puzzle generator follows this algorithm:

1. Begin with an empty board and randomly choose one number to fill into one cell.
2. Apply `hsolve` to make all logical deductions possible. (That is, after every step of generating a puzzle, keep track of the Sudoku Solution Graph for all cells of the board.)
3. Repeat the following steps until either a contradiction is reached or the board is completed:

- Randomly fill an unoccupied cell on the board with a candidate for that cell's SSG.
- Apply `hsolve` to make all logical deductions (which will fill in naked and hidden singles and adjust the SSG accordingly)
- If a contradiction occurs on the board, abort the procedure and start the process again from an empty board.

If no contradiction is reached, then eventually the board must be completely filled, since a new cell is filled in manually at each iteration.

The final puzzle is the board with all of the numbers that were filled in manually at each iteration of the algorithm (i.e., the board without the numbers filled in by `hsolve`).

Guaranteeing a Unique Solution with Standard Generator

For this algorithm to work, a small modification must be made in our backtracking strategy. If the backtracking strategy makes a guess that successfully completes the puzzle, we treat it as if this guess does not complete the puzzle but rather comes to a dead end. Thus, the backtracking strategy only makes a modification to the board if it makes a guess on some square that results in a contradiction, in which case it fills in that square with the other possibility. With this modification, we easily see that if our algorithm successfully generates a puzzle, then the puzzle must have a unique solution, because all of the cells of the puzzle that are not filled in are those that were determined at some point in the construction process by `hsolve`. With this updated backtracking strategy, `hsolve` makes a move only if the move follows logically and deterministically from the current state of the board; so if `hsolve` reaches a solution, it must be the unique one.

Pseudo-Generator

Our pseudo-generator takes a completed Sudoku board and a set of cells to leave empty at beginning of a puzzle, called the *reserved cells*. The idea is to guarantee the use of a high-level strategy, such as Swordfish or Backtracking, by ensuring that a generated puzzle cannot be completed without such a strategy. Call the starting puzzle the *seed board* and the solution the *completed seed board*. To use the pseudo-generator, we must first prepare a list of reserved cells, found as follows:

1. Take a seed board that `hsolve` cannot solve using strategies only up to tier k, but `hsolve` can solve with strategies up to tier $k+1$ (see Appendix for the different tiers of strategies we use).
2. Use `hsolve` to make all possible deductions (i.e. adjusting the SSG) using only strategies up to tier k.
3. Create a list of cells that are still empty.

We then pass to the pseudo-generator the completed seed board and this list of reserved cells. The pseudo-generator iterates the algorithm below, starting with an empty board, until all the cells except the reserved cells are filled in:

1. Randomly fill an unoccupied, unreserved cell on the board with the number in the corresponding cell of the completed seed board.
2. Apply `hsolve` to make logical deductions and to complete the board as much as possible.

Differences From Standard Generator

The main differences between the pseudo-generator and the standard generator are:

1. When filling in an empty cell, the standard generator uses the number in the corresponding cell of the completed puzzle, instead of choosing this number at random from the cell's SSG.
2. When selecting which empty cell to fill in, the pseudo-generator never selects one of the reserved cells.
3. `hsolve` is equipped with strategies only up to tier k.
4. The pseudo-generator terminates not when the board is completely filled in but rather when all of the unreserved cells are filled in.

The pseudo-generator is only partially random. It provides enough clues so that the unreserved cells of the board can be solved with strategies up to tier k, and the choice of which of these cells to reveal as clues is determined randomly. However, the solution of the generated puzzle is independent of these random choices and must be identical to the completed seed board. For the same reason as in the standard generator, the solution must be unique.

The pseudo-generator never provides clues for reserved cells; hence, when `hsolve` solves a puzzle, it uses strategies of tiers 0 through k to fill in the unreserved cells, and then is forced to use a strategy in tier $k + 1$ to solve the remaining portion of the board.

Pseudo-Generator Puzzle Variability

The benefit of the pseudo-generator over the standard generator is generating puzzles in which a strategy of tier $k + 1$ must be used, thus guaranteeing a high level of difficulty (if k is high). The drawback is that the pseudo-generator cannot be said to generate a puzzle at random, since it starts with a puzzle already generated in the past and constructs a new puzzle (using some random choices) out of its solution.

We implement the pseudo-generator by first randomly permuting the rows, columns, and numbers of the given completed puzzle, so as to create an illusion that it is a different puzzle. Ideally, we should have a large database of difficult puzzles to choose from (together with the highest tier strategy needed to solve each puzzle and its list of reserved cells that cannot be filled with strategies of lower tiers).

Difficulty Concerns

"Difficulty level" is not well-defined: In a system of three difficulty levels, how difficult is a medium puzzle, as compared to a hard or easy puzzle? In the previous correlation analysis in which we divided 800 puzzles into eight difficulty levels, we forced each difficulty level to contain 100 puzzles.

Generating Puzzles with a Specific Difficulty

Figure 5 shows the measured difficulty of 1,000 puzzles generated by the standard generator. We can divide the puzzles into intervals of difficulty, with equal numbers of puzzles in each interval. To create a puzzle of given difficulty level using the standard generator, we iterate the generator until a puzzle is generated whose difficulty value falls within the appropriate interval.

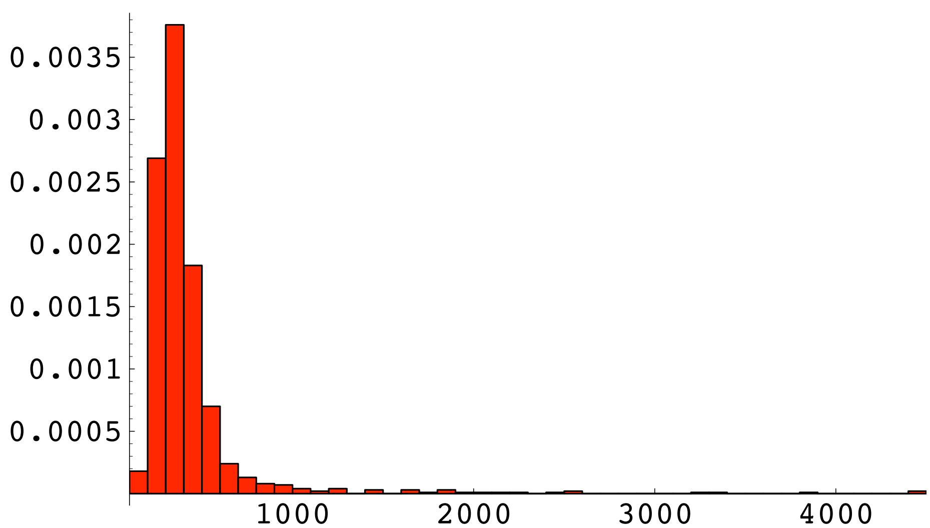


Figure 5. A histogram of the measured difficulty of 1,000 puzzles generated by the standard generator.

Standard Generator Runtime

It took 3 min to generate 100 valid boards (and 30 invalid boards) and 12 min to determine the difficulties of the 100 valid boards. Thus, 100 boards take a total 15 min to run, or an average of about 9 sec per valid board.

From the difficulty distribution in **Figure 5**, we can obtain an expected runtime estimate for each level of difficulty. For four levels, the expected number of boards that one needs to construct to obtain a board of level 1 is a geometric random variable with parameter $p = \frac{598}{1000}$, so the expected runtime to obtain a board of level 1 is $0.15 \times \frac{1000}{598} = 0.25$ min. Similarly, the expected runtimes to obtain boards of level 2, level 3, and level 4 are 3.1, 4.7, and 30 min.

Using Pseudo-Generator to Generate Difficult Puzzles

To generate large numbers of difficult boards, it would be best to employ the pseudo-generator. We fed the pseudo-generator a puzzle ("Riddle of Sho") that can be solved only by using the tier-5 backtracking strategy [Sudoku Solver n.d.]. The difficulty of the puzzle was determined to be 8,122, while the average difficulty of 20 derived puzzles generated using this puzzle was 7,111. Since all puzzles derived from a puzzle fed into the pseudo-generator must share application of the most difficult strategy, the difficulties of the derived puzzles are approximately the same as that of the original puzzle.

With a database of difficult puzzles, a method of employing the pseudo-generator is to find the midpoint of the difficulty bounds of the desired level, choose randomly a puzzle whose difficulty is close to this midpoint, and generate a derived puzzle. If the difficulty of the derived puzzle fails to be within our bounds, we continue choosing an existing puzzle at random and creating a derived puzzle until the bound condition is met. The average generation time for a puzzle is 9 sec, the same as for the standard generator. For difficult boards, there is a huge difference between the two strategies in the expected number of boards that one needs to construct, and the pseudo-generator is much more efficient.

Conclusion

Strengths

Our human solver `hsolve` models how a human Sudoku expert would solve a sudoku puzzle by posing Sudoku as a search problem. We judge the relative costs of each strategy by the number of verifications of possible strategy applications necessary to find it and thereby avoid assigning explicit numerical difficulty values to specific strategies. Instead, we allow the difficulty of a strategy to emerge from the difficulty of finding it, giving a more formal treatment of what seems to be an intuitive notion. This derivation of the difficulty provides a more objective metric than that used in most existing difficulty ratings.

The resulting metric has a Goodman-Kruskal γ-coefficient of 0.82 with an existing set of hand-rated puzzles, and it generates a difficulty distribution that corresponds to one empirically generated by millions of users. Thus, we have some confidence that this new metric gives an accurate and reasonably fast method of rating Sudoku puzzle difficulties.

We produced two puzzle generators, one able to generate original puzzles that are mostly relatively easy to solve, and one able to modify preexisting hard puzzles to create ones of similar difficulty. Given a database of difficult puzzles, our pseudo-generator is able to reliably generate many more puzzles of these difficulties.

Weaknesses

It was difficult to test the difficulty metric conclusively because of the dearth of available human-rated Sudoku puzzles. Hence, we could not conclusively establish what we believe to be a significant advantage of our difficulty metric over most existing ones.

While our puzzle generator generated puzzles of all difficulties according to our metric, it experienced difficulty creating very hard puzzles, as they occurred quite infrequently. Although we attempted to address this flaw by creating the pseudo-generator, it cannot create puzzles with entirely different final configurations.

Because of the additional computations required to calculate the search space for human behavior, both the difficulty metric and the puzzle generator have relatively slow runtimes compared to other raters and generator.

Appendix: Sudoku Strategies

Most (but not all) Sudoku puzzles can be solved using a series of logical deductions [What is Sudoku? n.d.]. These deductions have been organized into a number of common patterns, which we have organized by difficulty. The strategies have been classed into *tiers* between 0 and 5 based upon the general consensus of many sources on their level of complexity (for example, see Johnson [n.d.] and Sudoku Strategy [n.d.]).

In this work, we have used what seem to be the most commonly occurring and accessible strategies together with some simple backtracking. There are, of course, many more advanced strategies, but since our existing strategies suffice to solve almost all puzzles that we consider, we choose to ignore the more advanced ones.

0. Tier 0 Strategies
   - **Naked Single:** A Naked Single exists in the cell (i, j) if cell (i, j) on the board has no entry, but the corresponding entry (i, j) on the Sudoku Solution Graph has one and only one possible value. For example, in **Figure A1**. We see that cell $(2, 9)$ is empty. Furthermore,

1	2	3	4	5	6	7	8	?
v								

Figure A1. Example for Naked Single strategy.

the corresponding Sudoku Solution Graph entry in $(2, 9)$ can only contain the number 9, since the numbers 1 through 8 are already assigned to cells in row 2. Therefore, since cell $(2, 9)$ in the corresponding Sudoku Solution Graph only has one (naked) value, we can assign that value to cell $(2, 9)$ on the sudoku board.

Application Enumeration: Since a Naked Single could occur in any empty cell, this is just the number of empty cells, since checking if any empty cell is a Naked Single requires constant time.

- **Hidden Single:** A Hidden Single occurs in a given cell (i, j) when:

  (a) (i, j) has no entry on the Sudoku board
  (b) (i, j) contains the value k (among other values) on the Sudoku Solution Graph
  (c) No other cell in the same group as (i, j) has k as a value in its Sudoku Solution Graph

  Once we find a hidden single in (i, j) with value k, we assign k to (i, j) on the Sudoku board. The logic behind hidden singles is that given any group, all numbers 1 through 9 must appear exactly once. If we know cell (i, j) is the only cell that could contain the value k in a given row, then we know that it must hold value k on the actual Sudoku board. We can consider the example in **Figure A2**.

  We look at cell $(1, 1)$. First, $(1, 1)$ does not have an entry, and we can see that its corresponding entry in the Sudoku Solution Graph contains $\{1, 2, 7, 8, 9\}$. However, we see that the other cells in region 1 that don't have values assigned, i.e. cells $(1, 2), (1, 3), (2, 1)$ and $(3, 1)$, do not have the value 1 in their corresponding Sudoku Solution Graph cells; that is, none of the other four empty cells in the board besides $(1, 1)$ can hold the value 1, and so we can assign 1 to the cell $(1, 1)$.

?								
	3	4	1					
	5	6				1		
	1							
		1						

Figure A2. Example for Hidden Single strategy.

Application Enumeration: Since a Hidden Single could occur in any empty cell, this is just the number of empty cells, since checking if any empty cell is a Hidden Single requires constant time (inspecting other cells in the same group).

1. Tier 1 Strategies

- **Naked Double:** A Naked Double occurs when two cells on the board in the same group g do not have values assigned, and both their corresponding cells in the Sudoku Solution Graph have only the same two values k_1 and k_2 assigned to them. A naked double in (i_1, j_1) and (i_2, j_2) does not immediately give us the values contained in either (i_1, j_1) or (i_2, j_2), but it does allows us to eliminate k_1 and k_2 from the Sudoku Solution Graph of all cells in g beside (i_1, j_1) and (i_2, j_2).

  Application Enumeration: For each row, column, and region, we sum up $\binom{n}{2}$ where n is the number of empty cells in each group, since a Naked Double requires two empty cells in the same group.

- **Hidden Double:** A Hidden Double occurs in two cells (i_1, j_1) and (i_2, j_2) in the same group g when:

  (a) (i_1, j_1) and (i_2, j_2) have no values assigned on the board
  (b) (i_1, j_1) and (i_2, j_2) share two entries k_1 and k_2 (and contain possibly more) in the Sudoku Solution Graph
  (c) k_1 and k_2 do not appear in any other cell in group g on the Sudoku Solution Graph

  A hidden double does not allow us to immediately assign values to (i_1, j_1) or (i_2, j_2), but it does allow us to eliminate all entries other than k_1 and k_2 in the Sudoku Solution Graph for cells (i_1, j_1) and (i_2, j_2).

  Application Enumeration: For each row, column, and region, we sum up $\binom{n}{2}$ where n is the number of empty cells in each group, since a Hidden Double requires two empty cells in the same group.

- **Locked Candidates:** A Locked Candidate occurs if we have cells (for simplicity, suppose we only have two: (i_1, j_1) and (i_2, j_2)) such that:

  (a) (i_1, j_1) and (i_2, j_2) have no entries on the board
  (b) (i_1, j_1) and (i_2, j_2) share two groups, g_1 and g_2 (i.e. both cells are in the same row and region, or the same column and region)
  (c) (i_1, j_1) and (i_2, j_2) share some value k in the Sudoku Solution Graph
  (d) $\exists g_3$, a group of the same type as g_1, $g_1 \neq g_3$, such that k occurs in cells of $g_2 \cap g_3$
  (e) k does not occur elsewhere in g_3 besides $g_3 \cap g_2$
  (f) k does not occur in g_2 aside from $(g_2 \cap g_1) \cup (g_2 \cap g_3)$

  Then, since k must occur at least once in g_3, we know k must occur in $g_2 \cap g_3$. However, since k can only occur once in g_2, then k cannot occur in $g_2 \cap g_1$, so we can eliminate k from the Sudoku Solution Graph cells corresponding to (i_1, j_1) and (i_2, j_2). A locked candidate can also occur with three cells.

  Application Enumeration: For every row i, we examine each three-cell subset rs_{ij} formed as the intersection with some region j; there are twenty-seven such subsets. Out of those twenty-seven, we denote the number of subsets that have two or three empty cell as r_l. We define c_l for columns analogously, so this is just the sum $r_l + c_l$.

2. Tier 2 Strategies

- **Naked Triple:** A Naked Triple occurs when three cells on the board, $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3), in the same group g do not have values assigned, and all three of their corresponding cells in the Sudoku Solution Graph share only the same three possible values, k_1, k_2 and k_3. However, each cell of a Naked Triple does not have to have all three values, e.g. we can have (i_1, j_1) have values k_1, k_2 and k_3, (i_2, j_2) have k_2, k_3 and (i_3, j_3) have k_1 and k_3 on the Sudoku Solution Graph. We can remove k_1, k_2 and k_3 from all cells except for $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) in the Sudoku Solution Graph that are also in group g; the logic is similar to that of the Naked Double strategy.

  Application Enumeration: For each row, column, and region, we sum up $\binom{n}{3}$ where n is the number of empty cells in each group, since a Naked Triple requires three empty cells in the same group.

- **Hidden Triple:** A Hidden Triple is similar to a Naked Triple the way a Hidden Double is similar to a Naked Double, and occurs in cells $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) sharing the same group g when:

  (a) $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) contain no values on the Sudoku Board

(b) Values k_1, k_2 and k_3 appear among $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) in their SSG

(c) k_1, k_2 and k_3 do not appear in any other cells of g in the SSG

Then, we can eliminate all values beside k_1, k_2 and k_3 in the SSG of cells $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3). The reasoning is the same as for the Hidden Double strategy.

Application Enumeration: For each row, column, and region, we sum up $\binom{n}{3}$ where n is the number of empty cells in each group, since a Hidden Triple requires three empty cells in the same group.

- **X-Wing:** Given a value k, an X-Wing occurs if:

  (a) $\exists$ two rows, r_1 and r_2, such that the value k appears in the SSG for exactly two cells each of r_1 and r_2

  (b) $\exists$ distinct columns c_1 and c_2 such that k only appears in rows r_1 and r_2 the Sudoku Solution Graph in the set $(r_1 \cap c_1) \cup (r_1 \cap c_2) \cup (r_2 \cap c_1) \cup (r_2 \cap c_2)$

  Then, we can eliminate the value k as a possible value for all cells in c_1 and c_2 that are not also in r_1 and r_2, since k can only appear in each of the two possible cells of in each row r_1 and r_2 and k. Similarly, the X-Wing strategy can also be applied if we have a value k that is constrained in columns c_1 and c_2 in exactly the same two rows.

  Application Enumeration: For each value k, 1 through 9, we count the number of rows that contain k exactly twice in the SSG of its empty cells, r_k. Since we need two such rows to form an X-Wing for any one number, we take $\binom{r_k}{2}$. We also count the number of columns that contain k exactly twice in the SSG of its cells, c_k, and similarly take $\binom{c_k}{2}$. We sum over all values k, so this value is $\sum_k \binom{r_k}{2} + \binom{c_k}{2}$.

3. Tier 3 Strategies

- **Naked Quad:** A Naked Quad is similar to a Naked Triple; it occurs when four unfilled cells in the same group g contain only elements of set K of at most four possible values in their SSG. In this case, we can remove all values in K from all other cells in group g, since the values in K must belong only to the four unfilled cells.

  Application Enumeration: For each row, column, and region, we sum up $\binom{n}{4}$ where n is the number of empty cells in each group, since a Naked Quad requires three four empty cells in the same group.

- **Hidden Quad:** A Hidden Quad is analogous to a Hidden Triple. It occurs when we have four cells $(i_1, j_1), (i_2, j_2), (i_3, j_3)$ and (i_4, j_4) in the same group g such that:

  (a) $(i_1, j_1), (i_2, j_2), (i_3, j_3)$ and (i_4, j_4) share (among other elements) elements of the set K of at most four possible values in their SSG

  (b) No values of K appear in the SSG of any other cell in g

Then we can eliminate all values that cells $(i_1, j_1), (i_2, j_2), (i_3, j_3)$ and (i_4, j_4) take on other than values in K from their corresponding cells in the Sudoku Solution Graph. The reasoning is analogous to the Hidden Triple strategy.

Application Enumeration: For each row, column, and region, we sum up $\binom{n}{4}$ where n is the number of empty cells in each group, since a Hidden Quad requires three four empty cells in the same group.

- **Swordfish:** The Swordfish Strategy is the three-row analogue to the X-Wing Strategy. Suppose we have three rows, r_1, r_2 and r_3, such that the value k has not been assigned to any cell in r_1, r_2 or r_3. If the cells of r_1, r_2 and r_3 that have k as a possibility in their corresponding SSG are all in the same three columns c_1, c_2 and c_3, then no other cells in c_1, c_2 and c_3 can take on the value k, so we may eliminate the value k from the corresponding cells in the SSG. (This strategy can also be applied if we have columns that restrict the occurrence of k to three rows).

  Application Enumeration: For each value k, 1 through 9, we count the number of rows that contain k exactly two or three times in the SSG of its empty cells, r_k. Since we need three such rows to form a Swordfish for any one number we take $\binom{r_k}{3}$. We also count the number of columns that contain k two or three times in the SSG of its cells, c_k, and similarly take $\binom{c_k}{3}$. We sum over all values k, so this value is $\sum_k \binom{r_k}{3} + \binom{c_k}{3}$.

4. Tier 4 Strategies

   - **Jellyfish:** The Jellyfish Strategy is analogous to the Swordfish and X-Wing strategies. We apply similar reasoning to four rows r_1, r_2, r_3 and r_4 in which some value k is restricted to the same four columns c_1, c_2, c_3 and c_4. If the appearance of k in cells of r_1, r_2, r_3 and r_4 in the Sudoku Solution Graph is restricted to four specific columns, then we can eliminate k from any cells in c_1, c_2, c_3 and c_4 that are not in one of r_1, r_2, r_3 or r_4. Like the Swordfish strategy, the Jellyfish strategy may also be applied to columns instead of rows.

     Application Enumeration: For each value k, 1 through 9, we count the number of rows that contain k exactly two, three or four times in the SSG of its empty cells, r_k. Since we need four such rows to form a Jellyfish for any one number k, we take $\binom{r_k}{4}$. We also count the number of columns that contain k two, three or four times in the SSG of its cells, c_k, and similarly take $\binom{c_k}{4}$. We sum over all values k, so this value is $\sum_k \binom{r_k}{4} + \binom{c_k}{4}$.

5. Tier 5 Strategies

   - **Backtracking:** Backtracking in the sense that we use is a limited version of complete search. When cell (i, j) has no assigned value, but exactly 2 possible values k_1, k_2 in its SSG, the solver will assign a test value (assume k_1) to cell (i, j) and continue solving the puzzle using only Tier 0 strategies.

There are three possible results. If the solver arrives at a contradiction, he deduces that k_2 is in cell (i, j). If the solver completes the puzzle using the test value, this is the unique solution and the puzzle is solved. Otherwise, if the solver cannot proceed further but has not solved the puzzle completely, backtracking has failed and the solver must start a different strategy.

Application Enumeration: Since we only apply Backtracking to cells with exactly two values in its SSG, this is just the number of empty cells that have exactly two values in their SSG.

References

AoPS Inc. 2007. Combinatorics and sequences. `http://www.artofproblemsolving.com/Forum/viewtopic.php?t=88383`.

Caine, Allan, and Robin Cohen. 2006. MITS: A Mixed-Initiative Intelligent Tutoring System for Sudoku. *Advances in Artificial Intelligence* 550–561.

Cox, Kenneth C., Stephen G. Eick, Graham J. Wills, and Ronald J. Brachman. 1997. Brief application description; visual data mining: Recognizing telephone calling fraud. *Data Mining and Knowledge Discovery* 225–331.

Emery, Michael Ray. 2007. Solving Sudoku puzzles with the COUGAAR agent architecture. Thesis. `http://www.cs.montana.edu/techreports/2007/MichaelEmery.pdf`.

Eppstein, David. 2005. Nonrepetitive paths and cycles in graphs with application to Sudoku. `http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0507053`.

Felgenhauer, Bertram, and Frazer Jarvis. 2005. Enumerating possible Sudoku grids. `http://www.afjarvis.staff.shef.ac.uk/sudoku/sudoku.pdf`.

Goodman, Leo A., and William H. Kruskal. 1954. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49 (December 1954) 732–764.

GraphPad Software. n.d. QuickCalcs: Online calculators for scientists. `http://www.graphpad.com/quickcalcs/PValue1.cfm`.

Hanssen, Vegard. n.d. Sudoku puzzles. `http://www.menneske.no/sudoku/eng/`.

Hayes, Brian. 2006. Unwed numbers: The mathematics of Sudoku, a puzzle that boasts "No math required!" *American Scientist Online* `http://www.americanscientist.org/template/AssetDetail/assetid/48550?print=yes`.

Johnson, Angus. n.d. Solving Sudoku. `http://www.angusj.com/sudoku/hints.php`.

Knuth, Donald Ervin. 2000. Dancing links. In *Millennial Perspectives in Computer Science: Proceedings of the 1999 Oxford-Microsoft Symposium in Honour of Professor Sir Antony Hoare*, edited by Jim Davies, Bill Roscoe, and Jim Woodcock, 187–214. Basingstoke, U.K.: Palgrave Macmillan. `http://www-cs-faculty.stanford.edu/~uno/preprints.html`.

Lenz, Moritz. n.d. Sudoku Garden. `http://sudokugarden.de/en`.

Lewis, Rhyd. 2007. Metaheuristics can solve sudoku puzzles. *Journal of Heuristics* 13 (4): 387–401.

Lynce, Inês, and Joël Ouaknine. 2006. Sudoku as a SAT Problem. `http://sat.inesc-id.pt/~ines/publications/aimath06.pdf`.

Mantere, Timo, and Janne Koljonen. 2006. Solving and rating Sudoku puzzles with genetic algorithms. In *Proceedings of the 12th Finnish Artificial Intelligence Conference STeP*. `http://www.stes.fi/scai2006/proceedings/step2006-86-solving-and-rating-sudoku-puzzles.pdf`.

Simonis, Helmut. 2005. Sudoku as a constraint problem. In *Modelling and Reformulating Constraint Satisfaction*, edited by Brahim Hnich, Patrick Prosser, and Barbara Smith, 13–27. `http://homes.ieu.edu.tr/~bhnich/mod-proc.pdf#page=21`.

Sudoku. n.d. *Times Online*. `http://entertainment.timesonline.co.uk/tol/arts_and_entertainment/games_and_puzzles/sudoku/`.

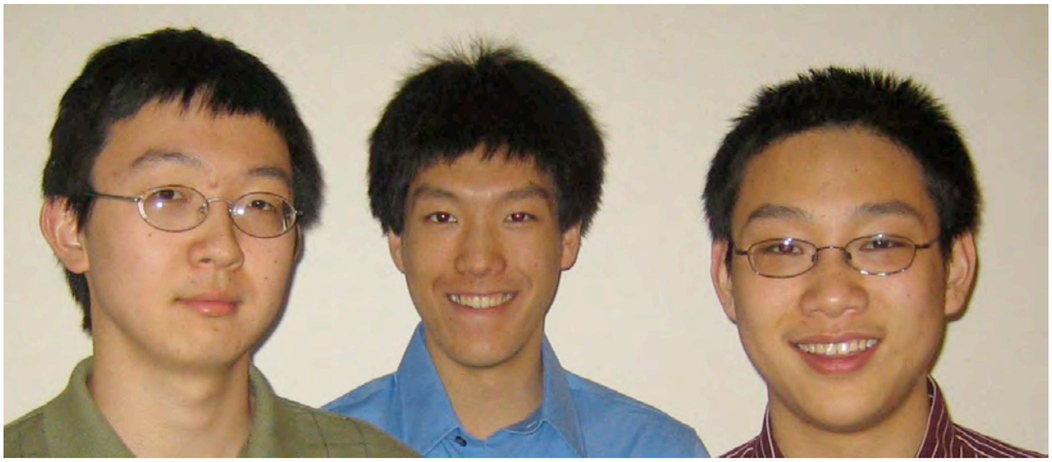
Sudoku solver. `http://www.scanraid.com/sudoku.htm`.

Sudoku strategy. n.d. *Sudoku Dragon*. `http://www.sudokudragon.com/sudokustrategy.htm`.

Web Sudoku. n.d. URL: `http://www.websudoku.com/`.

What is Sudoku? n.d. `http://www.sudokuaddict.com`.

Yato, Takayuki. 2003. Complexity and completeness of finding another solution and its application to puzzles. Thesis, January 2003. `http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/MasterThesis.pdf`.



Zhou Fan, Christopher Chang, and Yi Sun.

Taking the Mystery Out of Sudoku Difficulty: An Oracular Model

Sarah Fletcher
Frederick Johnson
David R. Morrison

Harvey Mudd College
Claremont, CA

Advisor: Jon Jacobsen

Summary

In the last few years, the 9-by-9 puzzle grid known as Sudoku has gone from being a popular Japanese puzzle to a global craze. As its popularity has grown, so has the demand for harder puzzles whose difficulty level has been rated accurately.

We devise a new metric for gauging the difficulty of a Sudoku puzzle. We use an oracle to model the growing variety of techniques prevalent in the Sudoku community. This approach allows our metric to reflect the difficulty of the puzzle itself rather than the difficulty with respect to some particular set of techniques or some perception of the hierarchy of the techniques. Our metric assigns a value in the range $[0, 1]$ to a puzzle.

We also develop an algorithm that generates puzzles with unique solutions across the full range of difficulty. While it does not produce puzzles of a specified difficulty on demand, it produces the various difficulty levels frequently enough that, as long as the desired score range is not too narrow, it is reasonable simply to generate puzzles until one of the desired difficulty is obtained. Our algorithm has exponential running time, necessitated by the fact that it solves the puzzle it is generating to check for uniqueness. However, we apply an algorithm known as Dancing Links to produce a reasonable runtime in all practical cases.

The UMAP Journal 29 (3) (2008) 327–341.

Introduction

The exact origins of the Sudoku puzzle are unclear, but the first modern "Sudoku" puzzle showed up under the name "Number Place" in a 1979 puzzle magazine put out by Dell Magazines. Nikoli Puzzles introduced the puzzle to Japan in 1984, giving it the name "Suuji wa dokushin ni kagiru," which was eventually shortened to the current "Sudoku." In 1986, Nikoli added two new constraints to the creation of the puzzle: There should be no more than 30 clues (or givens), and these clues must be arranged symmetrically. With a new name and a more esthetically-pleasing board, the game immediately took off in Japan. In late 2004, Sudoku was introduced to the London *Times*; and by the summer of 2005, it had infiltrated many major American newspapers and become the latest puzzle craze [Wikipedia 2008b].

Sudopedia is a Website that collects and organizes electronic information on Sudoku, including solving techniques, from how do deal with "Fishy Cycles" and "Squirmbags" to identifying "Skyscrapers" and what to do if you discover that you have a "Broken Wing." It even explains the possibilities for what has happened if you find yourself hopelessly buried in a "Bivalue Universal Grave." Some techniques are more logically complex than others, but many of similar complexity seem more natural to different players or are more powerful in certain situations. This situation makes it difficult to use specific advanced techniques in measuring the difficulty of a puzzle.

Our goal is a metric to rate Sudoku puzzles and an algorithm to generate them. A useful metric should reflect the difficulty as perceived by humans, so we analyze how humans approach the puzzle and use the conclusions as the basis for the metric. In particular, we introduce the concept of an "oracle" to model the plethora of complicated techniques. We also devise a normalized scoring technique, which allows our metric to be extended to a variety of difficulty levels.

We devise a generation algorithm to produce puzzles with unique solutions that span all difficulty levels, as measured by our metric. To ensure uniqueness, our generation algorithm must solve the puzzle (multiple times) to check for extra solutions. Since solving a Sudoku puzzle is an NP-complete problem [Wikipedia 2008b], our algorithm has exponential running time at best.

Terminology

- A **completed Sudoku board** is a 9×9 grid filled with $\{1,\ldots,9\}$ such that every row, column and 3×3 subgrid contains each number exactly once.
- A **Sudoku puzzle** or **Sudoku board** is a completed Sudoku board from

which some of the cell contents have been erased.

- A **cell** is one of the 81 squares of a 9×9 grid.
- The nine 3×3 subgrids that appear by dividing the board into thirds are called **blocks**.
- A **house** refers to any row, column or block of a 9×9 grid.
- A **hint** is a cell that has already been filled in a Sudoku puzzle.
- A **candidate** is a number that is allowed to go in a given cell. Initially, any empty cell has the candidate set $\{1, \ldots, 9\}$. Candidates can be eliminated when a number can already be found in a house containing the cell and by more complicated techniques.
- **Singles** is a solving technique in which a cell is determined by one of two basic methods:

  Naked Singles: If a cell has only one remaining candidate, then that cell can be filled with that candidate.

  Hidden Singles: If there is only one cell in a given house that has a certain candidate, then that cell can be filled with that candidate.

Assumptions

- *Every Sudoku puzzle has a unique solution.*
- *There are no restrictions on the locations of the hints in a Sudoku puzzle.* When the Japanese puzzle company Nikoli adapted the puzzle in 1986, it added the constraint that clues should be arranged symmetrically. We do not consider this esthetic touch to be important to the structure of the puzzle and hence ignore this constraint.
- *The singles solving techniques are sufficiently basic that the typical player uses them.* The logic for these techniques derives directly from the definition of the game.
- *The naked singles technique is "easier" than the hidden singles technique.* When we look for hidden singles first and move to naked singles only when hidden singles no longer produces new information, we can solve a puzzle in many fewer steps. On the other hand, if we first look for naked singles and then move to hidden singles, we could oscillate between the methods repeatedly. We do not claim that all human solvers find naked singles easier than hidden singles. However, hidden singles appears to be more powerful and thus in some sense "harder."
- *The difficulty of a puzzle cannot be based on any specific set of techniques.* There are many different techniques beyond the singles, and we cannot assume

that a player will use any particular one. A list of such techniques with explanations can be found at Sudopedia [2008]. Different puzzles will succumb more easily to different techniques and will thus seem easier (or harder) to different people, depending on what approaches they tend to use.

- *The difficulty of a puzzle does not scale linearly with the number of applications of higher-level techniques* There is an obvious jump in difficulty when a puzzle requires more than just the singles techniques, since then a player must use strategies that cannot be read directly from the rules. On the other hand, having to use the same or a similar higher technique repeatedly does not require any extra leap of logic.

Sudoku Difficulty Metric

Objectives

Our first task is to develop a metric, or scoring system, to determine the difficulty of an arbitrary Sudoku grid. However, the starting configuration of a puzzle is often quite deceptive about the level of difficulty; so we must analyze the difficulty by looking at both the starting configuration and the completed board.

Additionally, we want our metric to be extensible to varied difficulty levels and player abilities. That is, we would like those who disagree with our metric to be able to adjust it and produce a metric that they agree with. Finally, our metric should be representative of the *perceived* difficulty of a puzzle by a human solver, regardless of how "simple" it is for a computer to solve.

A Trip to the Oracle

We assume that a typical player starts solving a Sudoku puzzle begin by filling in cells that can be determined by the singles techniques. When the player can determine no more cells via those techniques, the player will begin to employ one or more higher-level methods and combine the new information with the singles techniques until solving the puzzle or getting stuck again.

We can exploit this observation to develop a metric that rates the difficulty of a puzzle simply by determining the number of different methods used to solve it. In particular, the more complicated the methods, the more challenging the puzzle is. However, due to the complicated nature of the more than 50 solving techniques [Sudopedia 2008], it is hard to say which are "more challenging" than others. Additionally, many humans approach a puzzle differently, applying different techniques at different stages of the puzzle. To avoid becoming bogged down in a zoo of Fish and X-Wings, we introduce the concept of an oracle.

The *oracle* is a being that knows the solution to all puzzles and can communicate a solution to a player, as long as the player knows how to ask properly. We can think of the oracle as if it were another player who uses a more-advanced solving technique. When a player gets stuck, the player goes to the oracle for help, and the oracle reveals the value of a cell in the grid. The usefulness of the revealed cell depends on the manner in which the player phrases the question.

In slightly less mystical terms, we use an oracle to represent the fact that the player uses higher-level techniques to solve a puzzle. Doing so allows us to model the difficulty of a puzzle without knowing anything about specific difficulty levels of solving methods. The oracle can be any method beyond singles that the player uses to fill in additional cells, and we represent this in our metric by randomly filling in some cell in the matrix. The perceived difficultly level of a puzzle increases as more higher-level techniques are used, but this increase is not linear in the number of techniques.

A Sudoku Difficulty Metric

Based on our above assumptions about how a human being approaches a Sudoku puzzle, we developed **Algorithm 1** to rate a puzzle's difficulty.

Algorithm 1 SUDOKU METRIC

```
procedure SCORE(InitialGrid, Solution)
    for All trials do
        Board = InitialGrid
        while Board is unsolved do
            Find all singles                          ▷ Iterative process
            if stuck then
                Ask ORACLE for help
            end if
        end while
        Count singles and ORACLE visits
    end for
    Compute average counts
    SCOREFUNCTION(singlesCounts, oracleCounts)
    return score                                  ▷ Use tanh to scale
end procedure
```

First, we search for naked singles until there are no more to be filled in. Then, we perform one pass looking for hidden singles. We repeat this process until the board is solved or until we can get no further using singles. (The order in which we consider the singles techniques accords with our assumption that naked singles are "easier" than hidden singles). When we can get no further, we consult the oracle. The algorithm keeps track of the the number of iterations, naked singles, hidden singles, and oracle visits, and presents this information to a scoring function, which combines the values and scales them to between 0 and 1, the "normalized" difficulty of the puzzle. The details of the scoring function are discussed below.

The above description does not take into consideration that because we are using a random device to reveal information, separate runs may produce different difficulty values and hence the revealed cell may provide either more or less aid. To smooth out the impacts of this factor, we run the test many times and average the scores of the trials.

Scoring with $\tanh$

A normalized score allows one puzzle to be compared with another. While on average we do not expect the unscaled score to be large, the number of oracle visits could be very high, thus sometimes producing large variability in the weighted sum of naked single, hidden single, and oracle visit counts. Simple scaling by an appropriate factor would give undue influence to outlier trials. Consequently, we pass the weighted sum through a sigmoid function that weights outliers on both the high and low ends similarly and gives the desired range of variability in the region that we expect most boards to fall into. We use a $\tanh$ function to accomplish this.

We would also like to model our assumption that each successive oracle visit is likely to provide less information than the previous one. We do this by passing the number of oracle visits through an inverse exponential function before scaling. Since we run a large number of trials to compute each score, we use the average number of oracle visits over all the trials in this exponential function, as doing so makes our scores fluctuate much less than if we average after applying the inverse exponential.

We arrive at the following equation for the unscaled score:

$$s = \alpha N + \beta H + \gamma \left(1 - e^{\delta \cdot (\overline{O} - \sigma)}\right), \tag{1}$$

where

- the Greek letters are user-tunable parameters (we discuss their significance shortly),
- N and H are the average number of naked singles and hidden singles found per scan through the board, and
- $\overline{O}$ is the average number of oracle visits per trial.

Note that N and H are averaged over a single trial, and together represent how many singles you can expect to find at a given stage in solving the puzzle; $\overline{O}$ is averaged over all trials and represents how many times you can expect to use higher-order techniques to solve the puzzle.

Finally, we pass this unscaled score through an appropriately shifted sigmoid:

$$\text{ScaledScore} = \frac{1 + \tanh\big[A(s - B)\big]}{2},$$

where s is the unscaled score from **(1)**, and A and B are user-tunable parameters. We shift the function up by 1 and scale by 1/2 to produce a range of values between 0 and 1 (**Figure 1**). We also smear the function out over a wide area to capture the differences in unscaled scores.

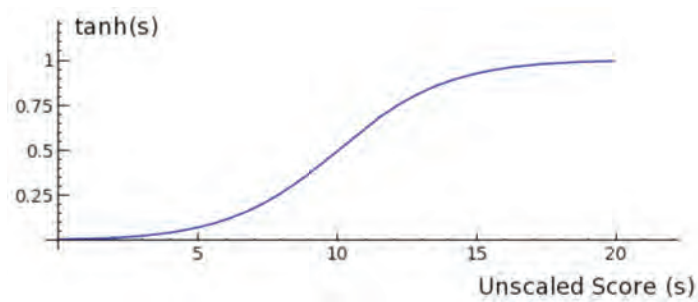


Figure 1. The scaled hyperbolic tangent that produces our final score. We shift the function up by 1 and scale by 1/2 to produce a range of values between 0 and 1. We also smear the function out over a wide area to capture the differences in unscaled scores.

A Zoo of Parameters

Our parameters can be divided into two groups:

- those that represent some intrinsic notion of how challenging a Sudoku board is

  α: Represents the difficulty of finding naked singles in a puzzle. It allows us to scale the observed number of naked singles in the puzzle based on how challenging we think they are to find.

  β: Weights the difficulty of finding hidden singles in a board. To agree with our earlier assumptions, we assume that $\alpha < \beta$.

  γ: Gives the weighting function for the number of oracle visits. This parameter will in general be quite high, as we believe that oracle visits should be the primary determination of difficulty level. In actuality, γ is a function of $\overline{O}$, since we don't want the exponential function in (1) to contribute negatively to the score. Thus, we have that

  $$\gamma = \begin{cases} 0, & \text{for } \overline{O} < 1; \\ G, & \text{otherwise,} \end{cases}$$

  for some large constant G.

- those that allow us to scale and shift a graph around:
  - δ: Controls the steepness of the exponential function.
  - σ: Controls the x-intercept of the exponential function.
  - A: Controls the spread of the $\tanh$ function.
  - B: Controls the shift of the $\tanh$ function.

The shift parameters are designed to allow for greater differentiation between puzzle difficulties, not to represent how difficult a puzzle actually is. Therefore, we believe that the first three parameters are the important ones. That is, those should be adjusted to reflect puzzle difficulty, and the last four should be set to whatever values allow for maximum differentiation among difficulty levels for a given set of puzzles. We discuss our choice of parameter values later. First, we turn to the problem of board generation.

Generation Algorithm

Objectives

It is natural to require that a Sudoku generator:

- always generate a puzzle with a unique solution (in keeping with our assumptions about valid Sudoku boards), and
- can generate any possible completed Sudoku board. (As it turns out, our algorithm does not actually generate all possible completed boards, but it should be able to if we expand the search space slightly.)

We also would like our generator be able to create boards across the spectrum of difficulty defined by our metric; but we do not demand that it be able to create a puzzle of a *specified* difficulty level, since small changes in a Sudoku board can have wide-reaching effects on its difficulty. However, our generator can be turned into an "on demand" generator by repeatedly generating boards until one of the desired difficulty level is produced.

Uniqueness and Complexity

It is easy to generate a Sudoku puzzle: Start with a completely filled-in grid, then remove numbers until you don't feel like removing any more. This method is quite fast but does not guarantee uniqueness. What is worse, the number of cells that you have to erase before multiple solutions can occur is alarmingly low! For example, if all of the 6s and 7s are removed from a Sudoku puzzle, there are now two possible solutions: the original solution, and one in which all positions of 6 and 7 are reversed. In fact, there is an even worse configuration known as a "deadly rectangle" [Sudopedia 2008] that can result in non-unique solutions if the wrong four cells of some

Sudoku boards are emptied. If we cannot guarantee uniqueness of solutions when only four numbers have been removed, how can we possibly guarantee it when more than 50 have?

The natural solution to this problem is to check the board for uniqueness after each cell is removed. If a removal causes the board to have a non-unique solution, replace it and try again. Unfortunately, there is no known fast algorithm for determining if the board has a unique solution. The only way is to enumerate all possible solutions, and this requires exponential time in the size of the board [Wikipedia 2008b].

The good news is that nevertheless there are fast algorithms, including Donald Knuth's Dancing Links algorithm [2000]. Dancing Links, also know as DLX, is an optimized brute-force enumeration algorithm for a problem known as *Exact Cover*. Exact Cover is an NP-complete problem dealing with membership in a certain collection of sets. By formulating the constraints on a Sudoku grid as sets, we can turn a Sudoku problem into an Exact Cover problem. While DLX is still an exponential algorithm, it outperforms most other such algorithms for similar problems. For our purposes, it is more than sufficient, since it solved the most challenging Sudoku problems we could find in 0.025 second.

DLX affords us an algorithm to generate puzzles with unique solutions: Simply remove cells from the completed board until no more cells can be removed while maintaining a unique solution, and use DLX at every stage to guarantee that the solution is still unique.

We now return to the issue of creating a completed Sudoku grid, since this is the one unfinished point in the algorithm. It again turns out to be quite difficult to generate a completed Sudoku grid, since doing so is akin to solving the puzzle. In theory, we could start with an empty grid and apply DLX to enumerate every possible solution—but there are 6.671×10^{21} completed boards.

Alternatively, many Websites suggest the following approach:

- Start with a completed Sudoku board.
- Permute rows, columns, and blocks (or other such operations that maintain the validity of the board).
- Output new Sudoku board.

This approach has two significant flaws:

- It assumes that we already have a valid Sudoku board (which is the very problem we're trying to solve); and
- it drastically limits the space of possible generated boards, since any single starting board can generate through these permutations only $3,359,232$ of the 6.617×10^{21} possible Sudoku boards [Wikipedia 2008a].

Thus, to perform our initial grid generation, we employ a combination of the two techniques that is quite fast and does not overly limit the size of the search space. We generate three random permutations of $1, \ldots, 9$ and fill these in along the diagonal blocks of an empty grid. We then seed the DLX algorithm with this board and ask it to find the first N solutions to the board, for N a large number (in our case, 100). Finally, we randomly select one of these boards to be our final board.

In principle, if N is sufficiently large, this method can generate any valid Sudoku board, since the seed to DLX is random, even if DLX itself is deterministic. With $(9!)^3$ seeds and 100 boards per seed, we can generate 4.778×10^{18} Sudoku boards, assuming that each seed has at least 100 solutions.

Use of the DLX algorithm makes this method fast enough for our purposes; we took advantage of Python code written by Antti Anjanki Ajanki [2006] that applies DLX to Sudoku puzzles. **Algorithm 2** gives the pseudocode for our generation algorithm.

Algorithm 2 SUDOKU GENERATOR

```
procedure GENERATEBOARD
    for i = 1, ..., 3 do                          ▷ Seed DLX
        PERMUTE {1, ..., 9}
        Fill diagonal block i
    end for
    DLX(seed, numToGenerate)
    Select randomly generated board
    repeat
        Remove random cells
        Check uniqueness (DLX)
    until No more cells can be removed
end procedure
```

Results and Analysis

To test both the utility of our metric and the effectiveness of our generation algorithm, we generated and scored 1,000 boards, and as a baseline also scored some independently-generated grids from the Internet. Our results match up well with "accepted" levels of difficulty, though there are some exceptions (**Table 1**). Our generator is biased towards generating easy puzzles but can generate puzzles with quite high difficulty; we believe that this performance is a consequence of the fact that difficult Sudoku puzzles are hard to create.

According to our metric, the most important factor in the difficulty of a puzzle is the number of oracle visits. The easiest puzzles are ones that can be solved entirely by singles techniques and thus do not visit the oracle at

Table 1.
Results from our Sudoku generator.

Difficulty	Number generated	Average score	Average # oracle visits
Cinch	740	0.06	0.1
Confusing	273	0.32	0.6
Challenging	53	0.69	1.5
Crazy	34	0.88	2.3

all. In general, this type of puzzles has a score of less than 0.18, which we use as our first dividing region.

The next level of difficulty is produced by puzzles that visit the oracle once on average; these puzzles produce a scaled score of between 0.18 and 0.6, so we use this as our second dividing region. Puzzles that require two or three hints from the oracle are scored between 0.6 and 0.8, and the absolute hardest puzzles have scores ranging from 0.8 to 1.0, due to needing three or more oracle visits. We show the output function of our metric, and mark the four difficulty regions, in **Figure 2**.

A significant number of generated boards have very high scores but no oracle visits, perhaps because of a large number of hidden singles.

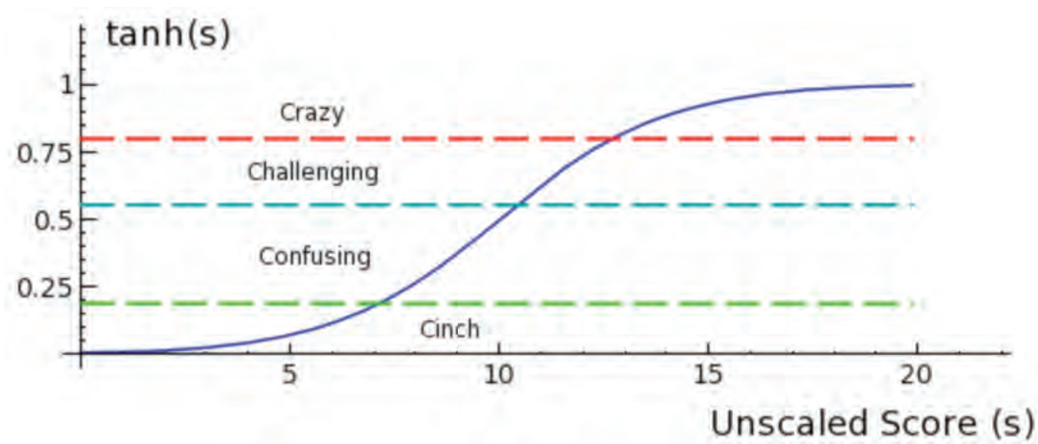


Figure 2. Our four difficulty regions plotted against the sigmoid scoring function.

For your solving pleasure, in **Figure 3** we included a bonus: four sample boards generated by our algorithm, ranging in difficulty level from "Cinch" (solvable by singles alone) to "Crazy" (requiring many advanced techniques). Try to solve them and see if you agree!

Our algorithm can generate puzzles across the spectrum but fewer of more difficult boards. This behavior appears to reflect the fact that Sudoku is very sensitive to seemingly minor modifications: Small changes to the layout of the hints or of the board can lead to vast changes in difficulty. In fact, a number of independently-generated boards that were rated by others as quite difficult turn out to be solvable by singles techniques alone,

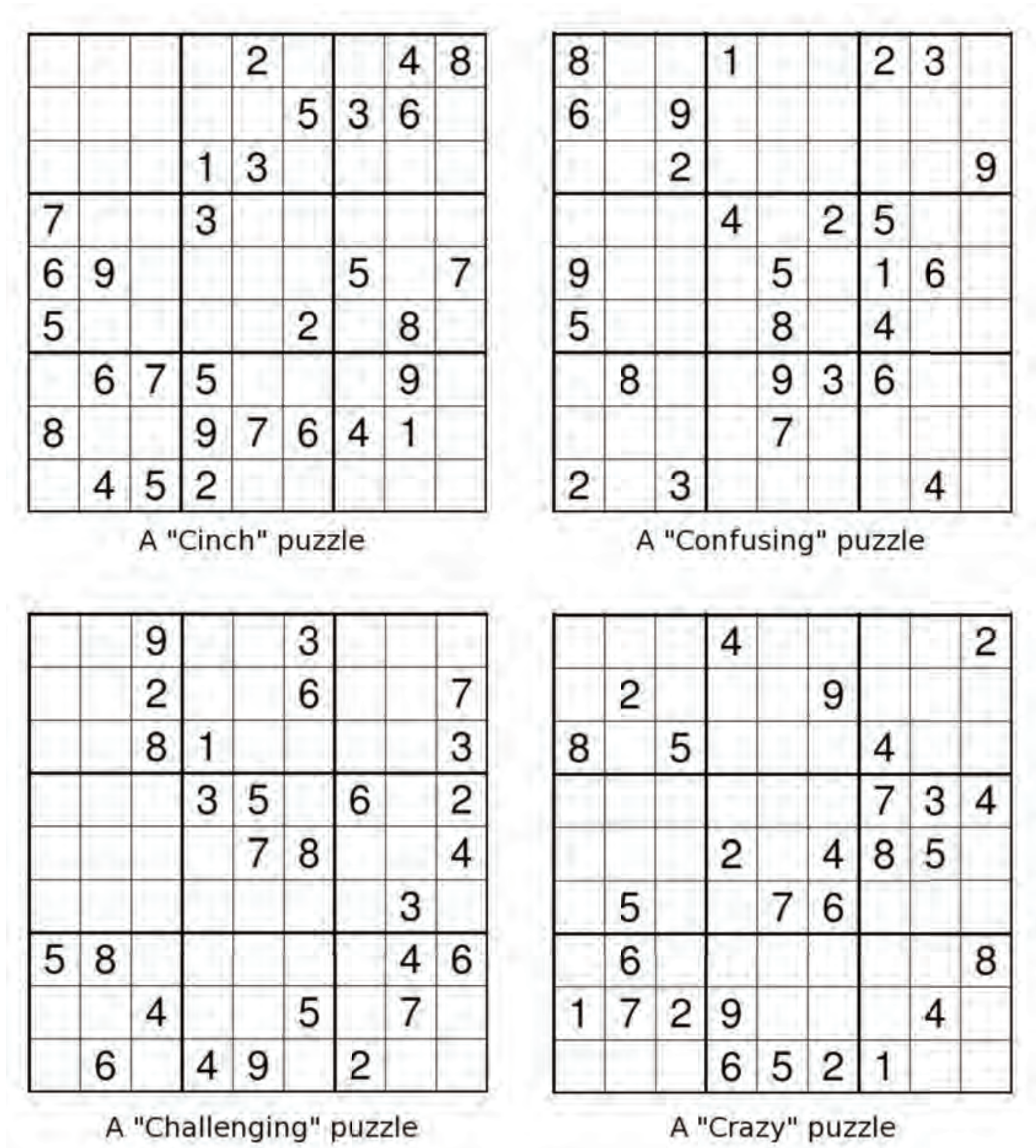


Figure 3. Four puzzles created by our generator, increasing in difficulty from left to right and top to bottom.

leading us to believe that creating hard Sudoku puzzles is hard.

We found a large number of very challenging Sudoku puzzles at Websites, which we tested with our metric. In particular, a Website called SudoCue [Werf 2007] offers some of the most challenging boards that we could find. Its "Daily Nightmare" section scored above 0.8 in almost all of our tests, with many puzzles above 0.9.

Our metric is highly sensitive to its parameter values. This is good, since it allows different users to tweak the metric to reflect their individual difficulty levels; but it is bad, because two different parameter sets can lead to vastly different difficulty scores. In **Table 2**, we show the parameter values that we use; they were empirically generated. We maintain that they are informative but acknowledge that they could be changed to yield different outcomes.

Table 2.

Parameter values chosen for our metric. The first three represent the difficulty of the puzzle, and the last four are scaling and shift parameters.

Parameter	α	β	γ	δ	σ	A	B
Value	0.1	0.5	15	0.5	1	0.25	10

In analyzing our metric, we found some puzzles that are scored reasonably and some that are scored outrageously. Our generator gave us a puzzle rated .85 that had no oracle visits. Suspicious of our metric, one member of our team tried solving the puzzle. He did so in less than 7.5 min, even though a photographer interrupted him to take pictures of the team. This discovery caused us to realize that perhaps our parameters do not mean what we thought they did. Our intent in weighting naked singles positively was to make our metric say that nearly-completely-filled-in puzzles are easier than sparsely-filled-in puzzles. We considered the average number of singles per scan rather than the total number because we wanted to somehow include the number of iterations in our metric. However, with our current parameter values, our metric says that a puzzle with a high number of singles per scan is easier than a puzzle with a low number of singles per scan, which is wrong and not what we intended. Parameters α and β should probably be negative, or else we should change what N and H mean.

Future Work

Any generator that has to solve the puzzle to check for the uniqueness of the solution will inherently have an exponential running time (assuming P$\neq$NP), since solving Sudoku has been shown to be NP-complete. Thus, to produce a generator with a better runtime, it would be necessary to find some other means of checking the uniqueness of the solution. One possible approach would be to analyze the configurations that occur when a puzzle does not have a unique solution. Checking for such configurations could result in a more efficient method of checking for solution uniqueness and thus would potentially allow for generators with less than exponential running time.

While we are excited by the potential that the oracle brings in rating Sudoku puzzles, we recognize that our metric as it stands is not as effective as it could be. We have a variety of ideas as to how it could be improved:

- The simplest improvement would be to run more experiments to determine better parameter values for our scoring function and the best places to insert difficulty breaks. A slightly different scoring function, perhaps one that directly considers the number of iterations needed to solve the Sudoku puzzle, could be a more accurate measure.

- Alternatively, we would like to devise a scoring function that distinguishes puzzles that can be solved using only the singles techniques vs. others.
- There is a much larger set of techniques that Sudopedia refers to as the Simple Sudoku Technique Set (SSTS) that advanced solvers consider trivial. If we were to add another layer to our model, so that we first did as much as possible with singles, then applied the rest of the SSTS and only went to the oracle when those techniques had been exhausted, it could give a better delineation of "medium-level" puzzles. If there were a threshold score dividing puzzles solvable by SSTS and puzzles requiring the oracle, it would allow advanced solvers to determine which puzzles they would find interesting.
- Another extension would be altering the oracle to eliminate possible cell candidate(s) rather than reveal a new cell. This alteration could potentially allow for greater differentiation among the hardest puzzles.

Conclusion

We devised a metric which uses an oracle to model techniques employed by the Sudoku community. This approach has the advantage of not depending on a specific set of techniques or any particular hierarchy of them. The large number of parameters in our scoring function leaves it open to adjustment and improvement.

In addition, we developed an algorithm to generate puzzles with unique solutions across a wide range of difficulties. It tends towards creating easier puzzles.

There is an increasing demand for more Sudoku puzzles, different puzzles, and harder puzzles. We hope that we have contributed insights into its levels of difficulty.

References

Ajanki, Antti. 2006. Dancing links Sudoku solver. Source code available online at `http://users.tkk.fi/~aajanki/sudoku/index.html`.

Eppstein, David. 2005. Nonrepetitive paths and cycles in graphs with application to sudoku. `http://arxiv.org/abs/cs/0507053v1`.

Felgenhauer, Frazer, and B. Jarvis. 2006. Mathematics of Sudoku I. `http://www.afjarvis.staff.shef.ac.uk/sudoku/`.

Knuth, Donald E. 2000. Dancing links. In *Millenial Perspectives in Computer Science*, 187–214. `http://www-cs-faculty.stanford.edu/~uno/preprints.html`.

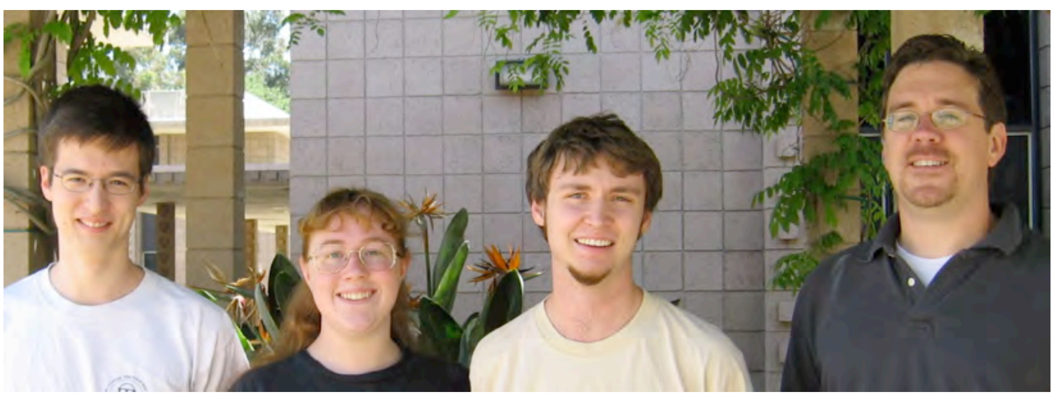
Simonis, Helmut. 2005. Sudoku as a constraint problem. In *Modelling and Reformulating Constraint Satisfaction*, edited by Brahim Hnich, Patrick Prosser, and Barbara Smith, 13–27. `http://homes.ieu.edu.tr/~bhnich/mod-proc.pdf#page=21`.

Solving technique. `http://www.sudopedia.org`. Accessed 16 Feb 2008, then last modified 11 Jan 2008.

van der Werf, Ruud. 2007. SudoCue—home of the Sudoku addict. `http://www.sudocue.net`.

Wikipedia. 2008a. Mathematics of Sudoku. `http://en.wikipedia.org/wiki/Mathematics_of_Sudoku`.

________. 2008b. Sudoku. `http://en.wikipedia.org/wiki/Sudoku`.



Frederick Johnson, Sarah Fletcher, David Morrison, and advisor Jon Jacobsen.

Difficulty-Driven Sudoku Puzzle Generation

Martin Hunt
Christopher Pong
George Tucker

Harvey Mudd College
Claremont, CA

Advisor: Zach Dodds

Summary

Many existing Sudoku puzzle generators create puzzles randomly by starting with either a blank grid or a filled-in grid. To generate a puzzle of a desired difficulty level, puzzles are made, graded, and discarded until one meets the required difficulty level, as evaluated by a predetermined difficulty metric. The efficiency of this process relies on randomness to span all difficulty levels.

We describe generation and evaluation methods that accurately model human Sudoku-playing. Instead of a completely random puzzle generator, we propose a new algorithm, *Difficulty-Driven Generation*, that guides the generation process by adding cells to an empty grid that maintain the desired difficulty.

We encapsulate the most difficult technique required to solve the puzzle and number of available moves at any given time into a *rounds* metric. A round is a single stage in the puzzle-solving process, consisting of a single high-level move or a maximal series of low-level moves. Our metric counts the numbers of each type of rounds.

Implementing our generator algorithm requires using an existing metric, which assigns a puzzle a difficulty corresponding to the most difficult technique required to solve it. We propose using our rounds metric as a method to further simplify our generator.

The UMAP Journal 29 (3) (2008) 343–362.

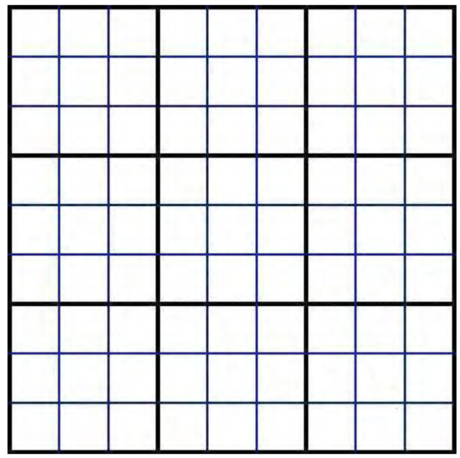


Figure 1. A blank Sudoku grid.

Introduction

Sudoku first appeared in 1979 in *Dell Pencil Puzzles & Word Games* under the name "Number Place" [Garns 1979]. In 1984, the puzzle migrated to Japan where the *Monthly Nikolist* began printing its own puzzles entitled "Suuji Wa Dokushin Ni Kagiru" or "the numbers must be single"—later shortened to Sudoku [Pegg Jr. 2005]. It was not until 2005, however, that this puzzle gained international fame.

We propose an algorithm to generate Sudoku puzzles at specified difficulty levels. This algorithm is based on a modified solver, which checks to ensure one solution to all generated puzzles and uses metrics to quantify the difficulty of the puzzle. Our algorithm differs from existing algorithms in using human-technique-based modeling to guide puzzle construction. In addition, we offer a new grading algorithm that measures both the most difficult moves and the number of available moves at each stage of the solving process. We ran basic tests on both algorithms to demonstrate their feasibility. Our work leads us to believe that combining our generation algorithm and metrics would result in a generation algorithm that creates puzzles on a scale of difficulty corresponding to actual perceived difficulty.

Terminology

Figure 1 shows a blank 9×9 cell Sudoku grid. In this grid, there are nine rows, nine columns, and nine blocks (3×3 disjoint cell groups defined by the thicker black lines). We use the terms:

- **Cell**: A single unit square in the Sudoku grid that can contain exactly one digit between 1 and 9 inclusive.
- **Adjacent Cell**: A cell that is in the same row, column or block as some other cell(s).

- **Hint**: One of the digits initially in a Sudoku puzzle.
- **Puzzle**: The 9×9 cell Sudoku grid with some cells containing digits (hints). The remaining cells are intentionally left empty.
- **Completed puzzle**: The 9×9 cell Sudoku grid with all cells containing digits.
- **Well-posed puzzle**: A Sudoku puzzle with exactly one solution (a unique completed puzzle). A Sudoku puzzle that is not well-posed either has no solutions or has multiple solutions (different completed puzzles can be obtained from the same initial puzzle).
- **Proper puzzle**: A Sudoku puzzle that can be solved using only logical moves—guessing and checking is not necessary. All proper puzzles are well-posed. Some newspaper or magazine problems are proper puzzles We concern ourselves with proper Sudoku puzzles only.
- **Candidate**: A number that can potentially be placed in a given cell. A cell typically has multiple candidates. A cell with only one candidate, for example, can simply be assigned the value of the candidate.

How to Play

The object of the game is to place the digits 1 through 9 in a given puzzle board such that every row, column and block contains each digit exactly once. An example puzzle with 28 hints is shown in **Figure 2a**. All of the digits can be placed using the logical techniques described subsequently. **Figure 2b** shows the completed solution to this problem. A well-posed Sudoku problem has only one solution.

9		8		2	3			
		3				7	9	
4					9			3
		6	5				8	4
			6					
						5		
	1			6	4			
	4				8		1	
8		9	7					

a. Puzzle example.

9	7	8	1	2	3	4	5	6
1	2	3	4	5	6	7	9	8
4	6	5	8	7	9	1	2	3
7	9	6	5	3	1	2	8	4
2	5	4	6	8	7	9	3	1
3	8	1	9	4	2	5	6	7
5	1	2	3	6	4	8	7	9
6	4	7	2	9	8	3	1	5
8	3	9	7	1	5	6	4	2

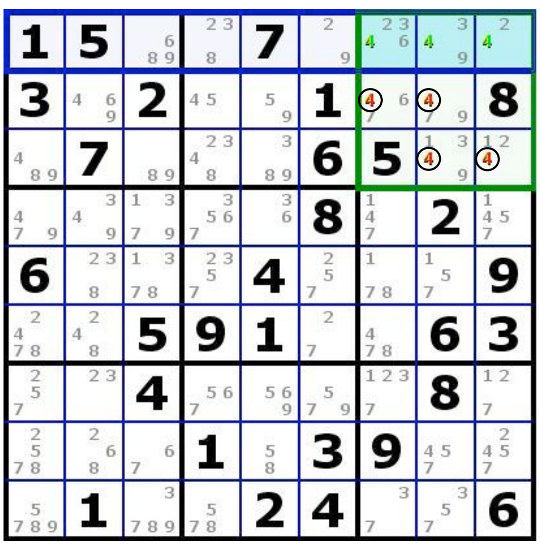
b. Puzzle solution.

Figure 2. Example of a Sudoku puzzle and its solution.

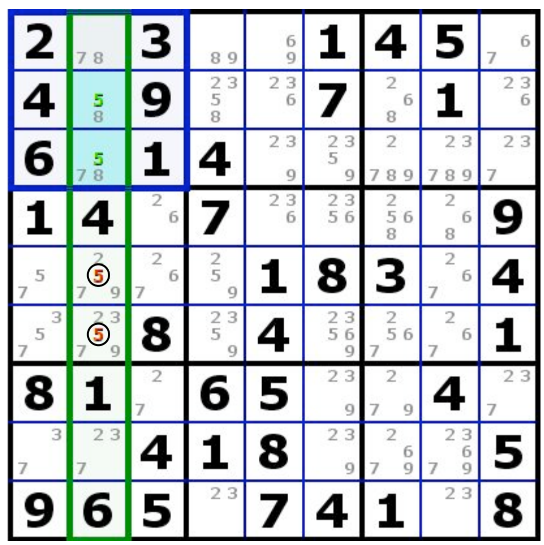
Techniques

We provide a list of commonly-used techniques; they require that the solver know the possible candidates for the relevant cells.

- **Naked Single**: A cell contains only one candidate, therefore it must be that number. This is called naked because there is only one candidate in this cell.
- **Hidden Single**: A cell contains multiple candidates but only one is possible given a row/column/block constraint, therefore it must be that number. This is called hidden because there are multiple candidates in the cell, but only one of them can be true due to the constraints.
- **Claiming**: This occurs when a candidate in a row/column also only appears in a single block. Since the row must have at least one of the candidate, these candidates "claim" the block. Therefore, all other instances of this candidate can be eliminated in the block. In **Figure 3a**, the circled 4s are the candidates that can be eliminated.
- **Pointing**: This occurs when a block has a candidate that appears only in a row/column. These candidates "point" to other candidates along the row/column that can be eliminated. In **Figure 3b**, the circled 5s are the candidates that can be eliminated.



a. Claiming.

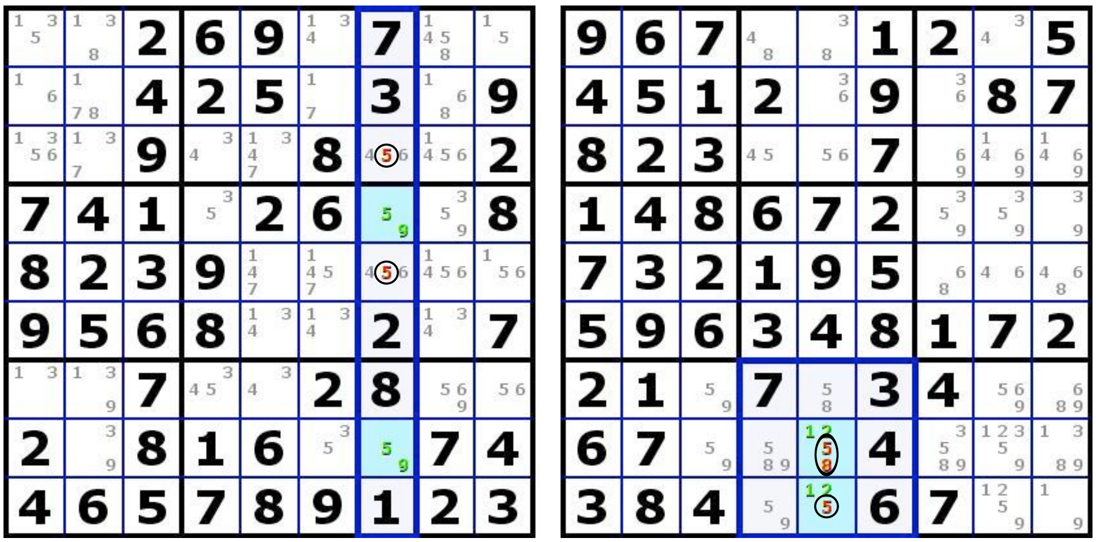


b. Pointing.

Figure 3. Examples of pointing and claiming techniques.

- **Naked Double**: In a given row/column/block, there are two cells that have the same two and only two candidates. Therefore, these candidates can be eliminated in any other adjacent cells. In **Figure 4a**, the circled 5s are the candidates that can be eliminated.

- **Hidden Double**: In a given row/column/block, two and only two cells can be one of two candidates due to the row/column/block constraint. Therefore any other candidates in these two cells can be eliminated. In **Figure 4b**, the circled 5s and 8 are the candidates that can be eliminated.



a. Naked Double. **b.** Hidden Double.

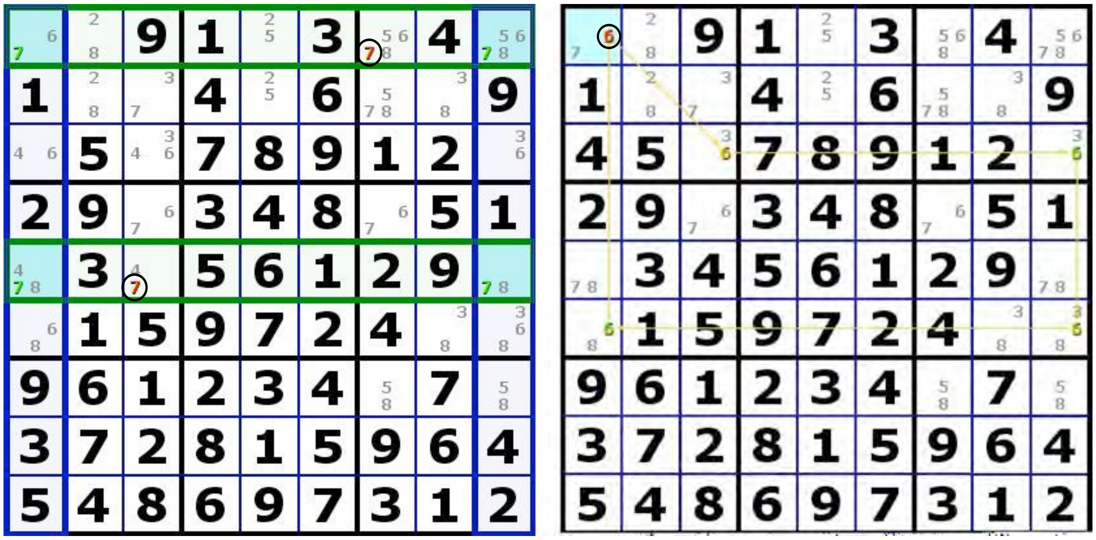
Figure 4. Examples of naked double and hidden double techniques.

- **Naked Subset (Triple, Quadruple, etc.)**: This is an extension of the naked double to higher numbers of candidates and cells, n. Because these n candidates must appear in the n cells, these candidates to be eliminated from adjacent cells.

- **Hidden Subset (Triple, Quadruple, etc.)**: This is an extension of the hidden double to higher numbers of candidates and cells, n. Due to row/column/block constraints, the n candidates must occupy n cells and other candidates in these cells can be eliminated.

- **X-Wing**: The X-Wing pattern focuses on the intersection of two rows and two columns. If two rows contain contain a single candidate digit in exactly two columns, then we can eliminate the candidate digit from all of the cells in those columns. The four cells of interest form a rectangle and the candidate digit must occupy cells on alternate corners, hence the name X-Wing. Note that the rows and columns can be interchanged. In **Figure 5a**, the circled 7s can be eliminated. X-Wings naturally lead to extensions such as XY-Wings, XYZ-Wings, Swordfish, Jellyfish, and Squirmbags. (Other advanced techniques are discussed at Stuart [2008].)

- **Nishio**: A candidate is guessed to be correct. If this leads to a contradiction, that candidate can be eliminated. In **Figure 5b**, the circled 6 can be eliminated.



a. X-Wing. **b.** Nishio.

Figure 5. Examples of X-Wing and Nishio techniques.

Generators

Solving Sudoku has been well-investigated. but generating puzzles is without much exploration. There are 6,670,903,752,021,072,936,960 possible completed puzzles [Felgenhauer and Jarvis 2006]. There are even more possible puzzles, since there are many ways of removing cells from completed puzzles to produce initial puzzles, though this number cannot be easily defined since the uniqueness constraint of well-formed puzzles makes them difficult to count.

We studied several generators, including Sudoku Explainer, jlib, Isanaki, PsuedoQ, SuDoKu, SudoCue, Microsoft Sudoku, and Gnome Sudoku. All rely on Sudoku solvers to verify that the generated puzzles are unique and sufficiently difficult. The two fastest generators work by randomly generating puzzles with brute-force solvers.

The most powerful solver that we encountered, Sudoku Explainer [Juillerat 2007], uses a brute-force solver to generate a completed puzzle. After producing the completed puzzle, it randomly removes cells and uses the same solver to verify that the puzzle with randomly removed cells still contains a unique solution. Finally, it uses a technique-based solver to evaluate the difficulty from a human perspective. Due to the variety of data structures and techniques already implemented in Sudoku Explainer, we use it as a framework for our code. Since generators rely heavily on solvers, we begin with a discussion of existing solver methods.

Modeling Human Sudoku Solving

Humans and the fastest computer algorithms solve Sudoku in very different ways. For example, the Dancing Links (DLX) Algorithm Knuth [2000] solves Sudoku by using a depth-first brute-force search in mere milliseconds. Backtracking and efficient data structures (2D linked lists) allow this algorithm to operate extremely quickly relative to other methods, although the Sudoku problem for a general $N \times N$ grid is NP-complete.

Because DLX is a brute-force solver, it can also find multiple solutions to Sudoku puzzles that are not well-posed (so from a computer's perspective, solving any Sudoku problem is a simple task!). However, most Sudoku players would not want to solve a puzzle in this manner because it requires extensive guessing and back-tracking. Instead, a player uses a repertoire of techniques gained from experience.

To determine the difficulty of a Sudoku puzzle from the perspective of a human by using a computer solver, we must model human behavior.

Human Behavior

We first consider Sudoku as a formal problem in constraint satisfaction [Simonis 2005], where we must satisfy the following four constraints:

1. **Cell constraint**: There can be only one number per cell.
2. **Row constraint**: There can be only one of each number per row.
3. **Column constraint**: There can be only one of each number per column.
4. **Block constraint**: There can be only one of each number per block.

We can realize these constraints in the form of a binary matrix. The rows represent the candidate digits for a cell; the columns represent constraints. In a blank Sudoku grid, there are 729 rows (nine candidates in each of the 81 cells) and 324 columns (81 constraints for each of the four constraints). A 1 is placed in the constraint matrix wherever a candidate satisfies a constraint. For example, the possibility of a 1 in the $(1, 1)$ entry of the Sudoku grid is represented as a row in the constraint matrix. The row has a 1 in the column corresponding to the constraint that the first box contains a 1. Therefore, each row contains exactly four 1s, since each possibility satisfies four constraints; and each column contains exactly nine 1s, since each constraint can be satisfied by nine candidates. An example of an abbreviated constraint matrix is in **Table 1**.

A solution is a subset of the rows such that each column has a single 1 in exactly one row of this subset. These rows correspond to a selection of digits such that every constraint is satisfied by exactly one digit, thus to a solution to the puzzle. The problem of finding in a general binary matrix a

Table 1.
Example of an abbreviated constraint matrix.

	One # in (2,3)	One # in (2,4)	#1 in Row 2	#2 in Row 2	#2 in Col. 4
#1@(2,3)	1	.	1	.	.
#2@(2,3)	1	.	.	1	.
#3@(2,3)	1	.	.	.	.
#1@(2,4)	.	1	1	.	.
#2@(2,4)	.	1	.	1	1
#3@(2,4)	.	1	.	.	.

subset of rows that sum along the columns to a row of 1s is known as the *exact-cover problem*.

The normal solution procedure for an exact-cover problem is to use DLX. The constraint-matrix form allows us to spot easily hidden or naked singles by simply looking for columns with a single 1 in them, which is already a step of the DLX algorithm. Moreover, the formalities of the constraint matrix allow us to state a general technique.

Theorem [Constraint Subset Rule]. *Suppose that we can form a set of n constraints A such that every candidate that satisfies a constraint of A only satisfies a single constraint of A and a set of n constraints B such that each candidate of A also satisfies a constraint in B. Then we can eliminate any candidate of B that is not a candidate of A.*

Proof: The assumption that every candidate that satisfies a constraint of A only satisfies a single constraint of A ensures that we must choose n candidates to satisfy the constraints in A (or else we could not possibly satisfy the n constraints of A). Each of those candidates satisfies at least one constraint in B, by assumption. Suppose for the sake of contradiction that we select a candidate that satisfies a constraint in B that does not satisfy any constraints A. Then B will have at most $n-1$ remaining constraints to be satisfied and the rows chosen to satisfy A will oversatisfy B (recall that each constraint must be satisfied once and only once). This is a contradiction, proving the claim. □

With specific constraints, the theorem translates naturally into many of the techniques discussed earlier. For example, if we a choose a singe block constraint as set A and a single row constraint as set B, then the Constraint Subset Rule reduces to the pointing technique.

Human Model

Using the constraint rule, we model the behavior of a human solver as:

1. Search for and use hidden and naked singles.

2. Search for constraint subsets of size one.
3. If a constraint subset is found, return to step 1. If none are found, return to step 2 and search for constraint subsets of one size larger.

Assumptions

- **Human solvers do not guess or use trial and error.** In certain puzzles, advanced Sudoku players use limited forms of trial and error, but extensive guessing trivializes the puzzle. So we consider only proper puzzles, which do not require guessing to solve.
- **Human solvers find all of the singles before moving on to more advanced techniques.** Sometimes players use more-advanced techniques while looking for singles; but for the most part, players look for the easiest moves before trying more-advanced techniques.
- **Human solvers search for subsets in order of increasing size.** The constraint subset rule treats all constraints equally, which allows us to generalize the advanced techniques easily. In practice, however, not all constraints are equal. Because the cells in a block are grouped together, for example, it is easier to focus on a block constraint than a row or column constraint. Future work should take this distinction into account.

Strengths

- **Simplicity.** By generalizing Sudoku to an exact cover problem, we avoid having to refer to specific named techniques and instead can use a single rule to govern our model.
- The model fairly accurately approximates how someone would go about solving a puzzle.

Weaknesses

- The constraint subset rule does not encompass all of the techniques that a human solver would employ; some advanced techniques, such as Nishio, do not fall under the subset rule. However, the constraint subset rule is a good approximation of the rules that human solvers use. Future work would incorporate additional rules to govern the model's behavior.

Using this model, we proceed to define metrics to assess the difficulty of puzzles.

Building the Metrics

One might assume that the more initial hints there are in a puzzle, the easier the puzzle is. Many papers follow this assumption, such as Tadei and Mancini [2006]. Lee [2006] uses a difficulty level based on a weighted sum of the techniques required to solve a puzzle, showing from a sample of 10,000 Sudoku puzzles that there is a correlation between the number of initially empty cells and the difficulty level.

There are many exceptions to both of these metrics that make them impractical to use. For example, puzzle A may only start with 22 hints but can be solved entirely with naked and hidden singles. Puzzle B may start with 50 hints but require the use of an advanced technique such as X-Wing to complete the puzzle. Puzzle C could start with 40 hints and after filling in 10 hidden singles be equivalent to Puzzle B (requiring an X-Wing). In practice, most people would find puzzles C and B equally difficult, while puzzle A would be significantly easier ($A < B = C$). The number of initial hints metric would classify the difficulties as $A > B > C$. The weighted-sum metric would do better, classifying $A < B < C$. Examples such as these show that counting the number of initial hints or weighting required techniques does not always accurately measure the difficulty of a puzzle.

We restrict our attention to evaluating the process involved in solving the puzzle. Our difficulty metric measures the following aspects:

- **Types of techniques required to solve the puzzle**: A more experienced Sudoku player will use techniques that require observing interactions between many cells.
- **The number of difficult techniques**: A puzzle with two X-Wings should be harder than a puzzle with one X-Wing.
- **The number of available moves**: It is easier to make progress in the puzzle when there are multiple options available, as opposed to a puzzle with only one logical move available.

To rate the difficulty of puzzles, Sudoku Explainer assigns each technique a numerical difficulty; the difficulty of the puzzle is defined to be the most difficult technique [Juillerat 2007]. The first problem with this type of implementation is that every technique must be documented and rated. The second problem is that many "medium" and even "hard" puzzles can be solved by finding singles, which have a low difficulty rating. The difference is that in difficult puzzles, the number of options at any given point is small. Sudoku Explainer's method of assigning difficulty has low granularity in the easy-to-hard category, which is the most important range of difficulties for most Sudoku players. The advanced techniques are not known by the general public and are difficult to use without a computer (one must write out all candidates for each cell).

We want our difficulty metric to distinguish between puzzles even when the puzzle requires only basic techniques. One problem is that different orders of eliminating singles may increase or decrease the number of options available. So we measure difficulty by defining a *round*.

> A **round** is performing either every possible single move (hidden or naked) that we can see from our current board state *or* exactly one higher-level move.

Our metric operates by performing rounds until it either solves the puzzle or cannot proceed further. We classify puzzles by the number of rounds, whether they use higher-level constraint sets, or if they cannot be solved using these techniques.

Difficulty Levels

Our difficulty levels are:

1. **Easy**: Can be solved using hidden and naked singles.
2. **Medium**: Can be solved using constraint subsets of size one and hidden and naked singles.
3. **Hard**: Can be solved using constraint subsets of size two and easier techniques.
4. **Fiendish**: Cannot be solved using constraint subsets of size two or easier techniques.

Within each category, the number of rounds can be used to rank puzzles.

Strengths

- **Accounts for the number of available options.**
- **Provides finer granularity at the easier levels.** Within each category, the number of rounds is way to quantify which puzzles are harder.
- **Generalizes many of the named techniques** by expressing them in terms of eliminations in the exact cover matrix.
- **Conceptually very simple.** By formulating the human solver's behavior in terms of the constraint matrix, we avoid having to use and rate specific techniques. The Constraint Subset Rule naturally scales in difficulty as the subset size increases.

Weaknesses

- **Does not take into account the differences among row/column/block constraints.** It is easier to find a naked single in a block than in a row, because the cells are grouped together.

- **Computationally slower.** Our metric searches for more basic types of moves than the Sudoku Explainer does. The benefits of expressing the rules in terms of the exact cover matrix far outweigh the decrease in speed.
- **Lack of regard for advanced techniques.** This metric does not consider many of the advanced techniques. Though most Sudoku players do not use these techniques, our basic implementation cannot classify puzzles with comparable granularity in the Hard and Fiendish categories. However, expanding the metric to include additional techniques is relatively simple using the constraint matrix formulation.

Results

Our metrics were tested with 34 Sudoku puzzles that appeared in the Los Angeles *Times* from 15 January 2008 to 17 February 2008. The results are tabulated in **Table 2**. We see that the terms "Gentle" and "Moderate" puzzles would both fall under our "Easy" category. These two difficulty levels would be distinguished within our "Easy" level because we can see that on average "Gentle" puzzles have fewer rounds than "Moderate" puzzles. However, there is some overlap. Perhaps our algorithm can distinguish between these categories better than the algorithm used by the *Times*, or the *Times*'s algorithm accounts for the fact that some singles may be easier to find than others. For example, a hidden single in a block might be easier to spot than a hidden single in a row or column; our algorithm treats all of these equally.

The *Times*'s Sudoku puzzles skip over our defined difficulty level of "Medium"—there are no puzzles with constraint subsets of size one. According to our metrics, the jump from "Moderate" to "Tough" is much larger than is justified. For example, there should be a level of difficulty between puzzles that use only naked and hidden singles and puzzles that require an X-Wing, consisting of puzzles that use pointing and claiming. In effect, our algorithm defines a higher granularity than that used by the Los Angeles *Times*.

Two "Tough" puzzles could not be solved using our Constraint Subset Rule. In fact, when checked with Sudoku Explainer, these puzzles require the use of Nishio and other advanced techniques that border on trial-and-error. Our metric ensures that puzzles such as these will not fall under the "Hard" level but will instead be moved up to the "Fiendish" level.

Building the Generator

Existing computer-based puzzle generators use one of two as random generation (RG) techniques, both of which amount to extending the func-

Table 2.
Results of metrics tested against Los Angeles *Times* Sudoku puzzles.

Difficulty	# of Rounds	# of Constraint Subsets of Size 1	# of Constraint Subsets of Size 2
Gentle	8	-	-
Gentle	7	-	-
Gentle	8	-	-
Gentle	10	-	-
Gentle	11	-	-
Gentle	9	-	-
Gentle	10	-	-
Gentle	7	-	-
Gentle	8	-	-
Moderate	8	-	-
Moderate	12	-	-
Moderate	9	-	-
Moderate	14	-	-
Moderate	9	-	-
Moderate	11	-	-
Moderate	10	-	-
Moderate	14	-	-
Moderate	9	-	-
Moderate	13	-	-
Moderate	12	-	-
Moderate	11	-	-
Moderate	11	-	-
Moderate	10	-	-
Moderate	10	-	-
Tough	12	8	1
Tough	11	5	1
Tough	14	3	1
Tough	Failed to solve		
Tough	Failed to solve		
Diabolical	Failed to solve		
Diabolical	Failed to solve		
Diabolical	Failed to solve		
Diabolical	Failed to solve		
Diabolical	Failed to solve		

tionality of a solver to guide the grid construction. (An introduction to making Sudoku grids by hand can be found at Time Intermedia Corporation [2007].)

RG1 (bottom-up generation). Begin with a blank grid.

(a) Add in a random number in a random spot in the grid

(b) Solve puzzle to see if there is a unique solution

   i. If there is a unique solution, proceed to next step

   ii. If there are multiple solutions, return to step (a)

(c) Remove unnecessary hints (any hints whose removal does not change the well-posed nature of the puzzle)

(d) Assess the difficulty of the puzzle, restarting generation if the difficulty is not the desired difficulty.

RG2 (top-down generation). Begin with a solved grid. There are many methods for constructing a solved grid. The one we primarily used builds grids by using a random brute-force solver with the most basic human techniques, placing numbers in obvious places for hidden and naked singles and randomly choosing numbers when no numbers can be placed logically (basic backtracking allows the algorithm to retry random number placement in the cases where it fails to generate a valid solution).

(a) Take out a number in a random spot

(b) Solve puzzle to determine if there is still a unique solution

   i. If there is a unique solution, go to step (a)

   ii. If there are multiple solutions, undo the last removal so that the grid again only has a single solution.

(c) Assess the difficulty of the puzzle, restarting generation if the difficulty is not the desired difficulty.

The first method is best implemented with a depth-first brute-force solver, because when the grid is nearly empty, multiple solutions need to be detected quickly. DLX is a natural choice for generators of this type.

The two methods are very similar: Steps RG1c and RG1d are essentially the same as steps RG2b and RG2c. The second method, however, runs the solver mostly on puzzles with unique solutions, while the first method runs the solver mostly on puzzles with multiple solutions. Solvers based on human techniques are slower and expect to use logic to deduce every square, operating best on multiple solutions. Thus, generators relying on solvers based on human techniques tend to favor the second method over the first.

These methods are driven primarily by random numbers, which is beneficial because random techniques theoretically offer fairly unbiased access to the entire domain of possible puzzles. However, no research that we are aware of has yet proved that a particular generation method is bias-free. Due to the size of the domain, we are not too worried about small biases. More importantly, these techniques are popular because they operate very quickly, converging for most difficulty levels to valid puzzles within a few seconds. Extremely hard puzzles cannot be dependably generated easily using these techniques, since they are very rare.

Difficulty-Driven Generation

There are two drawbacks to the previously mentioned methods:

- They do not operate with any notion of difficulty, hoping to stumble upon difficult puzzles by chance.
- They require many calls to the solver (on the order of hundreds) due both to the difficulty requirement and the backtracking in case random placements or removals fail. Particularly when using a human-solver, these calls can be very expensive. We do not concern ourselves too much with the runtime, since even generators that require several seconds to generate hard puzzles are tolerable to human users. Online puzzles, puzzles in magazines and newspapers, and puzzles on handheld devices are frequently pre-generated anyway.

We propose a new method of generating puzzles to address the first concern. Our goal is to develop a method that can produce a puzzle of a specified difficulty by guiding the placement of numbers in cells. We call this method - *Generation (DDG)*. DDG is based on merging the human-technique based solvers frequently found in the top-down RG2 method with the bottom-up RG1 method.

1. Begin with an empty grid and a desired difficulty, d. (We use real numbers for our difficulty levels.) Look for a puzzle difficulty $d' \in \mathbb{R}$ with $|d' - d| < t$ for some threshold $t \in \mathbb{R}$.
2. Fill in some number of cells to initialize the grid (well-posed Sudoku puzzles with fewer than 17 cells have not been found, so initializing the empty grid with some number of cells less than 17 is feasible).
3. Solve forward logically with the human-technique solver to remove any obvious cells from consideration.
4. For $i = 1$ to n: (n controls how much the search behaves like a depth-first search and how much it behaves like a breadth-first search)
   (a) Pick a random cell c_i.
   (b) Compute the possible values for c_i, choose a value v_i at random, and fill c_i with v_i.
   (c) Solve forward logically as far as possible. Record the difficulty, d_i.

(d) If we have solved the puzzle and $|d_i - d| < t$, stop and return the grid.

(e) Unfill c_i.

5. Find the value j that minimizes $|d_j - d|$.

6. Recursively call this procedure (enter at step 3) with c_j filled with v_j.

Implementing DDG is highly dependent on the choice of metrics, because it makes the following assumption:

> **DDG Assumption:** During bottom-up grid generation, choosing a cell that makes the puzzle better fit to the desired difficulty at a given iteration will make the final puzzle closer to the overall difficulty.

The construction of the DDG algorithm aims to make this assumption true. Ideally, the choice of later cells in the DDG approach will not significantly change the difficulty of the puzzle, since all cells in the DDG algorithm are subject to the same difficulty constraints. This assumption is not entirely true, particularly due to the complexity of the interactions in Sudoku. However, once we decide on particular metrics, we can alter DDG to make the assumption hold more often.

DDG with the Most Difficult Required Technique

As an example, we consider the metric of the *Most Difficult Required Technique (MDRT)*. Regardless of how many easy moves a player makes when solving a puzzle, the player must be able to perform the most difficult required technique to solve the puzzle. To code DDG with the MDRT, we began with the human-technique based solver of Sudoku Explainer. This solver cannot handle multiple solutions, so we modified it to return partial solutions when it could no longer logically deduce the next move. The solver works by checking a database of techniques (in order of increasing difficulty) against the puzzle to see if any are applicable. When applied, the technique returns both the areas affected by the technique and the specific cells affected (either potential candidates are removed from cells or cells are forced to certain values). Some techniques use the assumption of well-posed puzzle uniqueness to make deductions, which can lead to the solver falsely reporting the puzzle was solved entirely with logical deduction when in reality the puzzle has multiple solutions. To counter this, we use a fast brute-force solver to identify two solutions when there are multiple solutions.

To help ensure the DDG Assumption, we limit the possible locations where cells can be filled in Step 4a. During each recursive step, we find the most difficult technique used in the previous solve step and try to ensure that technique or similar techniques will continue to be required by the puzzle. Sudoku Explainer techniques are ranked using floating point numbers between 1.0 and 11.0 depending on their complexity in the particular instance when they are applied (for example, hidden singles in blocks are ranked as easier than hidden singles in rows or columns). During Step 4a, no cells or regions affected by the most difficult technique in the previous solve can be filled with values.

Additionally, we use the intersection of the two solutions found by the brute-force solver to indicate which cells are most likely to make the puzzle converge to a unique solution, limiting our cell choices in Step 4a to those cells that were not common to the two solutions. The brute-force solver we use computes the two solutions by depth-first searching in the 'opposite' direction, picking candidates in the search starting at 1 for one solution and 9 for the the other solution, guaranteeing maximal difference between the solutions.

Improvements to Difficulty-Driven Generation

DDG currently can produce a range of solutions for various difficulty levels. However, it operates much slower than the random generation solvers. To optimize the algorithm, we recommend the following adjustments. These adjustments significantly increase the complexity of the algorithm far beyond a guided breadth/depth first search.

- **Tune DDG's desired difficulty levels** at different stages of recursion. The DDG Assumption is entirely valid in practice due to the fact that fewer cells are available to choose from as the algorithm progresses. We recommend that the desired difficulty level start initially high with a high tolerance for the acceptable range of difficulties d_i that should be tried in a recursive step. As more cells are added to the puzzle, the difficulty naturally declines, so the generator should aim higher at the beginning. As the generator progresses, a process similar to simulated annealing should occur with the difficulty level being 'cooled' at the appropriate rate to converge to the desired difficulty level at the same time the puzzle converges to a unique solution of the same difficulty.
- **Make better use of the metrics** to determine exactly which cells can be added at each step with the dual goal of maximizing closeness to uniqueness and minimizing changes to the previously found most difficult required technique. This approach may also allow such algorithms to look for particular techniques. One could imagine that a player desiring extra practice with forcing chains of a certain length could generate several puzzles explicitly preserving forcing chains found during the search process.
- **Sample cell values from a known solution grid** rather than a random distribution. Our experimental results indicate that certain solution grids have a maximum difficulty regardless of how much searching is performed over the grid for a good initial grid. Thus, if this suggestion

is used, after a certain amount of the solution space is searched, the maximum difficulty for that solution grid should be estimated. If the estimated difficulty does not meet the desired difficulty, the solution grid should be regenerated.

DDG Complexity

The DDG algorithm is more complex than RG algorithms, and we believe that the potential benefits outweigh the additional complexity. For generating puzzles with targeted difficulty levels, DDG has the potential to be much more efficient than RG in terms of expected running time and search complexity (number of iterations and branches).

Overall complexity of any generator is highly dependent on the solver used. By using the human technique-based solver, our runtime significantly exceeds the runtime of RG methods using brute-force solvers. We believe that our work on DLX-based difficulty assessment could replace the reliance on the human technique-based solver, significantly accelerating the generator to make it competitive with RG methods in terms of running time. We hypothesize that an optimized DDG will outperform RG for extremely high difficulty levels since the probability of complicated structures arising at random is very small. RG requires several seconds to compute extremely difficult puzzles. (We consider "extremely difficult" puzzles to be those rated over 8.0 on the Sudoku Explainer scale.)

Conclusion

We cast Sudoku as an exact cover problem. Then we present a model of human solving strategies, which allows us to define a natural difficulty metric on Sudoku puzzles. This metric provides four difficulty levels: Easy, Medium, Hard, and Fiendish, each with an additional granularity determined by the number of rounds to complete the puzzle. This metric was tested against puzzles from the Los Angeles *Times*. Our 'Easy' metric encompasses both the 'Gentle' and 'Moderate', but is able to distinguish between puzzles in the category by the number of rounds it takes to complete the puzzle. It also offers additional granularity at the higher end by defining another level that contains constraint sets of size one, which the *Times* lacks. This metric also provides additional accuracy by separating out puzzles from the "Hard" level into a "Fiendish" level that require much more advanced techniques such as Nishio.

Our Difficulty-Driven Generation algorithm customizes to various definitions of difficulty. It builds from existing ideas of random generation algorithms, combining the bottom-up approach with human-technique based solvers to generate puzzles of varying difficulties. Though the generator requires additional tuning to make it competitive with current generators, we have demonstrated its ability to generate a range of puzzle difficulties.

References

Felgenhauer, Bertram, and Frazer Jarvis. 2006. Mathematics of sudoku I. `http://www.afjarvis.staff.shef.ac.uk/sudoku/felgenhauer_jarvis_spec1.pdf` .

Garns, Howard. 1979. Number place. *Dell Pencil Puzzles & Word Games* #16 (May 1979): 6.

Juillerat, Nicolas. 2007. Sudoku Explainer. `http://diuf.unifr.ch/people/juillera/Sudoku/Sudoku.html` .

Knuth, Donald E. 2000. Dancing links. Knuth, Donald Ervin. 2000. Dancing links. In *Millennial Perspectives in Computer Science: Proceedings of the 1999 Oxford-Microsoft Symposium in Honour of Professor Sir Antony Hoare*, edited by Jim Davies, Bill Roscoe, and Jim Woodcock, 187–214. Basingstoke, U.K.: Palgrave Macmillan. `http://www-cs-faculty.stanford.edu/~uno/preprints.html` .

Lee, Wei-Meng. 2006. Programming Sudoku. Berkeley, CA: Apress.

Mancini, Simona. 2006. Sudoku game: Theory, models and algorithms. Thesis, Politecnico di Torino. `http://compalg.inf.elte.hu/~tony/Oktatas/Rozsa/Sudoku-thesis/tesi_Mancini_Simona%2520SUDOKU.pdf` .

Pegg, Ed, Jr. 2005. Sudoku variations. Math Games. `http://www.maa.org/editorial/mathgames/mathgames_09_05_05.html`.

Simonis, Helmut. 2005. Sudoku as a constraint problem. In *Modelling and Reformulating Constraint Satisfaction*, edited by Brahim Hnich, Patrick Prosser, and Barbara Smith, 13–27. `http://homes.ieu.edu.tr/~bhnich/mod-proc.pdf#page=21` .

Stuart, Andrew. 2008. Strategy families. `http://www.scanraid.com/Strategy_Families` .

Time Intermedia Corporation. 2007. Puzzle generator Japan. `http://puzzle.gr.jp/show/English/LetsMakeNPElem/01`.

Chris Pong, Martin Hunt, and George Tucker.

Ease and Toil: Analyzing Sudoku

Seth B. Chadwick
Rachel M. Krieg
Christopher E. Granade

University of Alaska Fairbanks
Fairbanks, AK

Advisor: Orion S. Lawlor

Abstract

Sudoku is a logic puzzle in which the numbers 1 through 9 are arranged in a 9×9 matrix, subject to the constraint that there are no repeated numbers in any row, column, or designated 3×3 square.

In addition to being entertaining, Sudoku promises insight into computer science and mathematical modeling. Since Sudoku-solving is an NP-complete problem, algorithms to generate and solve puzzles may offer new approaches to a whole class of computational problems. Moreover, Sudoku construction is essentially an optimization problem.

We propose an algorithm to construct unique Sudoku puzzles with four levels of difficulty. We attempt to minimize the complexity of the algorithm while still maintaining separate difficulty levels and guaranteeing unique solutions.

To accomplish our objectives, we develop metrics to analyze the difficulty of a puzzle. By applying our metrics to published control puzzles with specified difficulty levels, we develop classification functions. We use the functions to ensure that our algorithm generates puzzles with difficulty levels analogous to those published. We also seek to measure and reduce the computational complexity of the generation and metric measurement algorithms.

Finally, we analyze and reduce the complexity involved in generating puzzles while maintaining the ability to choose the difficulty level of the puzzles generated. To do so, we implement a profiler and perform statistical hypothesis-testing to streamline the algorithm.

The UMAP Journal 29 (3) (2008) 363–379.

Introduction

Goals

Our goal is to create an algorithm to produce Sudoku puzzles that:

- creates only valid puzzle instances (no contradictions, unique solution);
- can generate puzzles at any of four different difficulty levels;
- produces puzzles in a reasonable amount of time.

We explicitly do not aim to:

- "force" a particular solving method upon players,
- be the best available algorithm for the making exceedingly difficult puzzles, or
- impose symmetry requirements.

Rules of Sudoku

Sudoku is played on a 3×3 grid of blocks, each of which is a 3×3 grid of *cells*. Each cell contains a *value* from 1 through 9 or is empty. Given a partially-filled grid called a *puzzle*, the object is to place values in all empty cells so that the constraints (see below) are upheld. We impose the additional requirement that a puzzle admit exactly one solution.

The constraints are that in a solution, no row, column, or block may have two cells with the same value.

Terminology and Notation

Assignment A tuple (x, X) of a value and a cell. We say that X has the value x, X maps to x, or $X \mapsto x$.

Candidates Values that can be assigned to a square. The set of candidates for a cell X is denoted $X?$.

Cell A single square, which may contain a value between 1 and 9. We denote cells by uppercase italic serif letters: X, Y, Z.

Block One of the nine 3×3 squares in the puzzle. The boundaries of blocks are denoted by thicker lines on the puzzle's grid. No two blocks overlap (share common cells).

Grouping A set of cells in the same row, column or block. We represent groupings by uppercase boldface serif letters: $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$. We refer unambiguously to the row groupings $\mathbf{R}_i$, the column groupings $\mathbf{C}_j$ and the block groupings $\mathbf{B}_c$. The set of all groupings is $\mathbb{G}$.

Metric A function m from the set of valid puzzles to the reals.

Puzzle A 9×9 matrix of cells with at least one empty and at least one filled cell. We impose the additional requirement that a puzzle have exactly one solution. We denote puzzles by boldface capital serif letters: $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$. We refer to cells belonging to a puzzle: $X \in \mathbf{P}$.

Representative of a block The upper-left cell in the block.

Restrictions In some cases, it is more straightforward to discuss which values a cell cannot have than to discuss the candidates. The restrictions set $X!$ for a cell X is $\mathbb{V} \backslash X?$.

Rule An algorithm that accepts a puzzle $\mathbf{P}$ and produces either a puzzle $\mathbf{P}'$ representing strictly more information (more restrictions have been added via logical inference or cells have been filled in) or some value that indicates that the rule failed to advance the puzzle towards a solution.

Solution A set of assignments to all cells in a puzzle such that all groupings have exactly one cell assigned to each value.

Value A symbol that may be assigned to a cell. All puzzles here use the traditional numeric value set $\mathbb{V} = \{1, \ldots, 9\}$. A value is denoted by a lowercase sans serif letter: x, y, z.

Indexing

By convention, all indices start with zero for the first cell or block.

$$
\begin{aligned}
c &: \text{ block number} \\
k &: \text{ cell number within a block} \\
i, j &: \text{ row number, column number} \\
i', j' &: \text{ representative row number, column number}
\end{aligned}
$$

These indicies are related by the following functions:

$$
\begin{aligned}
&c\,(i, j) = \frac{j}{3} + 3\left\lfloor \frac{i}{3} \right\rfloor, \\
&i\,(c, k) = 3\left\lfloor \frac{c}{3} \right\rfloor + \left\lfloor \frac{k}{3} \right\rfloor, \qquad j\,(c, k) = (c \bmod 3) \cdot 3 + (k \bmod 3)\,, \\
&i'\,(c) = 3\left\lfloor \frac{c}{3} \right\rfloor, \qquad j'\,(c) = (c \bmod 3) \cdot 3, \\
&i'\,(i) = 3\left\lfloor \frac{i}{3} \right\rfloor, \qquad j'\,(j) = 3\left\lfloor \frac{j}{3} \right\rfloor.
\end{aligned}
$$

Figure 1 demonstrates how the rows, columns and blocks are indexed, as well as the idea of a block representative. In the third Sudoku grid, the representatives for each block are denoted with an "r".

0								
1								
2								
3								
4								
5								
6								
7								
8								

0	1	2	3	4	5	6	7	8

r			r			r		
	0			1			2	
r			r			r		
	3			4			5	
r			r			r		
	6			7			8	

Figure 1. Demonstration of indexing schemes.

Rules of Sudoku

We state formally the rules of Sudoku that restrict allowable assignments, using the notation developed thus far:

$$(\forall \mathbf{G} \in \mathbb{G}\ \forall X \in G) \qquad X \mapsto \mathsf{v} \Rightarrow \nexists Y \in \mathbf{G} : Y \mapsto \mathsf{v}.$$

Applying this formalism will allow us to make strong claims about solving techniques.

Example Puzzles

The rules alone do not explain what a Sudoku puzzle looks like, and so we have included a few examples of well-crafted Sudoku puzzles. **Figure 2** shows a puzzle ranked as "Easy" by WebSudoku [Greenspan and Lee n.d.].

						7		8
3			2			4	5	
8	7	4	5	9	3		1	
			8	1				
	9	2	3		5	8	4	
				7	9			
	4		6	3	1	9	8	5
	8	1			4			6
6		9						

Figure 2. Puzzle generated by WebSudoku (ranked as "Easy").

			1	2	4			
	8					4		
6				8	3	9		
3		1	4	5	2			7
	2		3		8	1	5	4
4	5	8		1		3		2
		9	2	4	1	5		6
		5	8	3	6		4	9
X			9	7	5	Y		

Figure 3. Example of the Naked Pair rule.

Background

Common Solving Techniques

In the techniques below, we assume that the puzzle has a unique solution. These techniques and examples are adapted from Taylor [2008] and Astraware Limited [2005].

Naked Pair

If, in a single row, column or block grouping $\mathbf{A}$, there are two cells X and Y each having the same pair of candidates $X? = Y? = \{p, q\}$, then those candidates can be eliminated from all other cells in $\mathbf{A}$. To see that we can do this, assume for the sake of contradiction that there exists some cell $Z \in \mathbf{A}$ such that $Z \mapsto \mathsf{p}$, then $X \not\mapsto \mathsf{p}$, which implies that $X \mapsto \mathsf{q}$. This in turn means that $Y \not\mapsto \mathsf{q}$, but we have from $Z \mapsto \mathsf{p}$ that $Y \not\mapsto \mathsf{p}$, leaving $Y? = \varnothing$. Since the puzzle has a solution, this is a contradiction, and $Z \not\mapsto \mathsf{p}$.

As an example of this arrangement is shown in **Figure 3**. The cells marked X and Y satisfy $X? = Y? = \{2, 8\}$, so we can remove both 2 and 8 from all other cells in $\mathbf{R}_8$. That is, $\forall Z \in (\mathbf{R}_8 \setminus \{X, Y\}) : 2, 8 \notin Z?$.

Naked Triplet

This rule is analogous to the Naked Pair rule but involves three cells instead of two. Let $\mathbf{A}$ be some grouping (row, column or block), and let $X, Y, Z \in \mathbf{A}$ such that the candidates for X, Y and Z are drawn from $\{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$. Then, by exhaustion, there is a one-to-one set of assignments from $\{X, Y, Z\}$ to $\{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$. Therefore, no other cell in $\mathbf{A}$ may map to a value in $\{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$.

An example is in **Figure 4**. We have marked the cells $\{X, Y, Z\}$ accordingly and consider only block 8. In this puzzle, $X? = \{3, 7\}$, $Y? = \{1, 3, 7\}$ and $Z? = \{1, 3\}$. Therefore, we must assign 1, 3, and 7 to these cells, and can remove them as candidates for cells marked with an asterisk.

Hidden Pair

Informally, this rule is conjugate to the Naked Pair rule. We again consider a single grouping $\mathbf{A}$ and two cells $X, Y \in \mathbf{A}$, but the condition is that there exist two values u and v such that at least one of $\{\mathsf{u}, \mathsf{v}\}$ is in each of $X?$ and $Y?$, but such that for any cell $Q \in (\mathbf{A} \setminus \{X, Y\})$, $\mathsf{u}, \mathsf{v} \notin Q?$. Thus, since $\mathbf{A}$ must contain a cell with each of the values, we can force $X?, Y? \subseteq \{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$.

An example of this is given in **Figure 5**. We focus on the grouping $\mathbf{R}_8$, and label X and Y in the puzzle. Since X and Y are the only cells in $\mathbf{R}_8$ whose candidate lists contain 1 and 7, we can eliminate all other candidates for these cells.

		4				9	1	8
6	5	2	8				2	
8		9	1	3	2			5
5	1	2						4
	9		4	7	5	1	6	2
6	7	4	2	8	1	5	3	9
	4		6	2		X	5	Y
	3	5			8	2	*	6
2	6	7				*	*	Z

Figure 4. Example of the Naked Triplet rule.

		4	9		5		8	6
6	5	2	7		8		3	
8		9		3	6		5	
		8			4		2	7
	2	6		5	7			
7	4		8	9	2	1	6	
	8			7	9	6		2
2	9				1	3		
4	6	X			3		Y	

Figure 5. Example of the Hidden Pair rule.

Hidden Triplet

As with the Naked Pair rule, we can extend the Hidden Pair rule to apply to three cells. Let $\mathbf{A}$ be a grouping, and let $X, Y, Z \in \mathbf{A}$ be cells such that at least one of $\{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$ is in each of $X?$, $Y?$ and $Z?$ for some values t, u and v. Then, if for any other cell $Q \in (\mathbf{A} \setminus \{X, Y, Z\})$, $\mathsf{t}, \mathsf{u}, \mathsf{v} \notin Q?$, we claim that we can force $X?, Y?, Z? \subseteq \{\mathsf{t}, \mathsf{u}, \mathsf{v}\}$.

An example is shown in **Figure 6**, where in the grouping $\mathbf{R}_5$, only the cells marked X, Y and Z can take on the values of 1, 2 and 7. We would thus conclude that any candidate of X, Y or Z that is not either 1, 2, or 7 may be eliminated.

8	9	5		4	X	6	2	3
1	6	3	2			5	4	7
2	7	4		5		1	9	8
	8		4		Y			5
	5	2		3		4		1
4	3				5		6	2
9	1	7	5	6		2		4
3	2	8			4	7	5	6
5	4	6			Z		1	9

Figure 6. Example of the Hidden Triplet rule.

*	*	9		3		6		
*	3	6		1	4		8	9
1			8	6	9		3	5
*	9	*				8		
*	1	*					9	
*	6	8		9		1	7	
6	*	1	9		3			2
9	7	2	6	4		3		
*	*	3		2		9		

Figure 7. Example of the Multi-Line rule.

Multi-Line

We develop this technique for columns, but it works for rows with trivial modifications. Consider three blocks $\mathbf{B}_a$, $\mathbf{B}_b$, and $\mathbf{B}_c$ that all intersect the columns $\mathbf{C}_x$, $\mathbf{C}_y$, and $\mathbf{C}_z$. If for some value v, there exists at least one cell X in each of $\mathbf{C}_x$ and $\mathbf{C}_y$ such that $\mathsf{v} \in X?$ but that there exists no such $X \in \mathbf{C}_z$, then we know that the cell $V \in \mathbf{B}_c$ such that $V \mapsto \mathsf{v}$ satisfies $V \in \mathbf{C}_z$. Were this not the case, then we would not be able to satisfy the requirements for $\mathbf{B}_a$ and $\mathbf{B}_b$.

An example of this rule is shown in **Figure 7**. In that figure, cells that we are interested in, and for which 5 is a candidate, are marked with an asterisk. We will be letting $a = 0$, $b = 6$, $c = 3$, $x = 0$, $y = 1$ and $z = 2$. Then, note that all of the asterisks for blocks 0 and 6 are located in the first two columns. Thus, in order to satisfy the constraint that a 5 appear in each of these blocks, block 0 must have a 5 in either column 0 or 1, while block 6 must have a 5 in the other column. This leaves only column 2 open for block 3, and so we can remove 5 from the candidate lists for all of the cells in column 0 and block 3.

Previous Work

Sudoku Explainer

The Sudoku Explainer application [Juillerat 2007] generates difficulty values for a puzzle by trying each in a battery of solving rules until the puzzle is solved, then finding out which rule had the highest difficulty value. These values are assigned arbitrarily in the application.

QQWing

The QQWing application [Ostermiller 2007] is an efficient puzzle generator that makes no attempt to analyze the difficulty of generated puzzles beyond categorizing them into one of four categories. QQWing has the unique feature of being able to print out step-by-step guides for solving given puzzles.

GNOME Sudoku

Included with the GNOME Desktop Environment, GNOME Sudoku [Hinkle 2006] is a Python application for playing the game; since it is distributed in source form, one can directly call the generator routines.

The application assigns a difficulty value between 0 and 1 each puzzle. Rather than tuning the generator to requests, it simply regenerates any puzzle outside of a requested difficulty range. Hence, it is not a useful model of how to write a tunable generator, but it is very helpful for quickly generating large numbers of control puzzles. We used a small Python script to extract the puzzles.

Metric Design

Overview

The metric that we designed to test the difficulty of puzzles was the *weighted normalized ease function* (WNEF). It is was based on the calculation of a *normalized choice histogram*.

As the first step in calculating this metric, we count the number of choices for each empty cell's value and compile these values into a histogram with nine bins. Finally, we multiply these elements by empirically-determined weights and sum the result to obtain the WNEF.

Assumptions

The design of the WNEF metric is predicated on two assumptions:

- There exists some objective standard by which we can rank puzzles in order of difficulty.
- The difficulty of a puzzle is roughly proportional to the number of choices that a solver can make without directly contradicting any basic constraint.

In addition, in testing and analyzing this metric, we included a third assumption:

- The difficulties of individual puzzles are independently and identically distributed over each source.

Mathematical Basis for WNEF

We start by defining the choice function of a cell $c\left(X\right)$:

$$c\left(X\right) = \left|X?\right|,$$

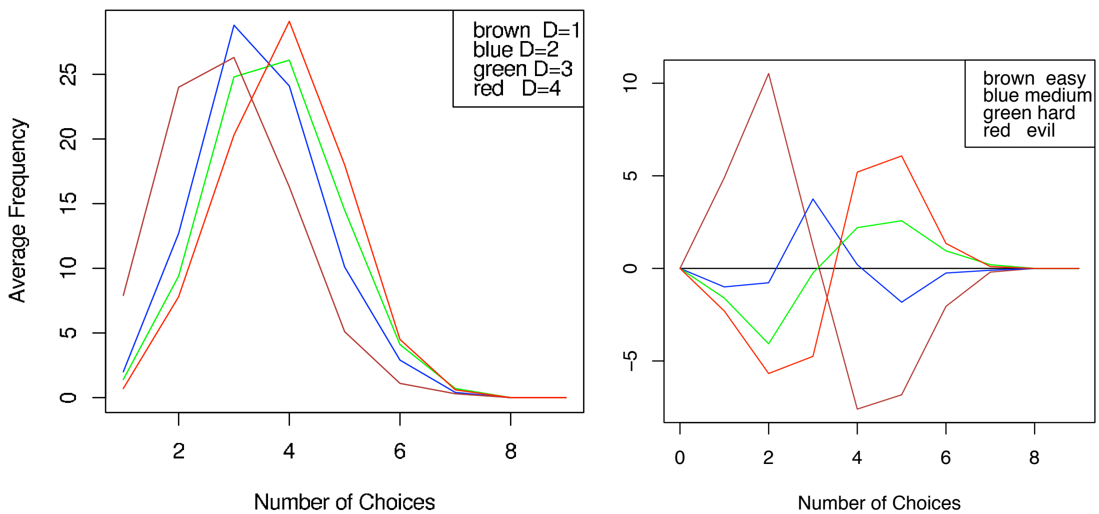
the number of choices available. This function is useful only for empty cells. We denote all empty cells in a puzzle $\mathbf{P}$ by $\mathbf{P}$

$$E\left(\mathbf{P}\right) = \left\{X \in P \mid \forall \mathsf{v} \in \mathbb{V} : X \not\mapsto \mathsf{v}\right\}.$$

By binning each empty cell based on the choice function, we obtain the choice histogram $\vec{c}\left(\mathbf{P}\right)$ of a puzzle $\mathbf{P}$.

$$c_n\left(\mathbf{P}\right) = \left|\left\{X \in \mathbf{P} \mid c\left(X\right) = n\right\}\right| = \left|\left\{X \in \mathbf{P} \mid \left|X?\right| = n\right\}\right|. \qquad \textbf{(1)}$$

Examples of histograms with and without the mean control histogram (obtained from control puzzles) are in **Figures 8a** and **b**.



(a) Original histograms. (b) Histograms with mean removed.

Figure 8. Examples of choice histograms.

From the histogram, we obtain the value $\text{wef}(\mathbf{P})$ of the (unnormalized) weighted ease function by convoluting the histogram with a weight function $w(n)$:

$$\text{wef}(\mathbf{P}) = \sum_{n=1}^{9} w(n) \cdot c_n(\mathbf{P}),$$

where $c_n(\mathbf{P})$ is the nth value in the histogram $\vec{c}(\mathbf{P})$. This function, however, has the absurd trait that removing information from a puzzle results in more empty cells, which in turn causes the function to strictly increase. We therefore calculate the weighted and *normalized* ease function:

$$\text{wnef}(\mathbf{P}) = \frac{\text{wef}(\mathbf{P})}{w(1) \cdot |E(\mathbf{P})|}.$$

This calculates the ratio of the weighted ease function to the maximum value that it can have (which is when all empty cells are completely determined but have not been filled in). We experimented with three different weight functions before deciding upon the *exponential weight function*.

Complexity

The complexity of finding the WNEF is the same as for finding the choice histogram (normalized or not). To do that, we need to find the direct restrictions on each cell by examining the row, column, and block in which it is located. Doing so in the least efficient way that is still reasonable, we look at each of the 8 other cells in those three groupings, even though some are checked multiple times, resulting in 24 comparisons per cell. For a total of 81 cells, this results in 1,944 comparisons. Of course, we check only when the cell is empty; so for any puzzle, the number of comparisons is fewer. Hence, finding the WNEF is a constant-time operation.

Metric Calibration and Testing

Control Puzzle Sources

In calibrating and testing the metrics, we used published puzzles from several sources with levels of difficulty as labeled by their authors, including:

- WebSudoku [Greenspan and Lee n.d.]: 10 each of Easy, Medium, Hard, and Evil puzzles
- Games World of Sudoku [Ganni 2008]: 10 each of puzzles with 1, 2, 3, and 4 stars
- GNOME Sudoku [Hinkle 2005]: 2000 Hard puzzles.
- "top2365" from Stertenbrink [2005]: 2365 Evil puzzles.

Testing Method

Defining Difficulty Ranges

We separated our control puzzles into four broad ranges of difficulty: easy, medium, hard, and evil, denoted by indices 1, 2, 3 and 4.

Information Collection

We calculated the metrics for each control puzzle. The information collected included:

- $|E(\mathbf{P}_i)|$, the total number of empty cells in $\mathbf{P}_i$;
- $C(\mathbf{P}_i) = \sum_{X \in \mathbf{P}_i} X?$, the number of possible choices for all cells; and
- the choice histogram $\vec{c}$ defined in **1**.

Statistical Analysis of Control Puzzles

The number of empty cells and the number of total choices lack any association with difficulty. In easier puzzles, there seem to be more cells with fewer choices than in more difficult puzzles (**Figure 8**).

We found a negative correlation between the difficulty level and WNEF for the control puzzles (lowest curve in **Figure 9**). This leads us to consider the mean WNEF for control puzzles of difficulty d, $d = 1, 2, 3, 4$. We tested the hypotheses that this mean is different for d and $d + 1$, for $d = 1, 2, 3$, using the mean WNEF, its standard deviation, and the t-test for difference of means. We concluded that the WNEF produces distinct difficulty levels, at significance level $\alpha = 0.0025$, for each of $d = 1, 2, 3$.

Choice of Weight Function.

We tried three different weighting functions for specifying WNEF values: exponential, quadratic and linear.

$$\begin{aligned} w_{\text{exp}}(n) &= 2^{9-n}, \\ w_{\text{sq}}(n) &= (10-n)^2, \\ w_{\text{lin}}(n) &= (10-n), \end{aligned}$$

where n is the number of choices for a cell. For all three, the graphs of WNEF vs. difficulty all looked very similar (**Figure 9**).

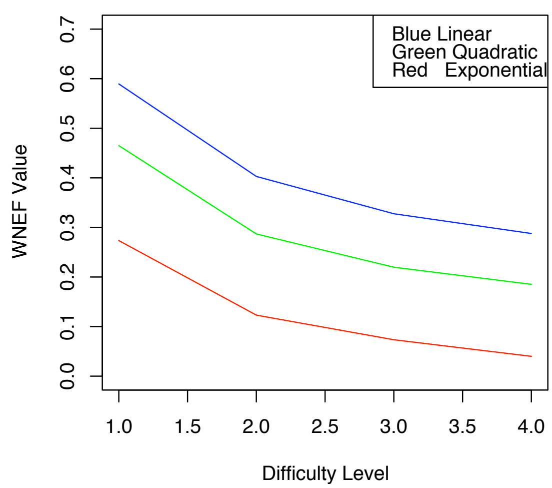


Figure 9. WNEF vs. difficulty level, for various weighting functions.

We concluded that we could choose any of the three weighting functions. We arbitrarily chose w_{exp}.

Generator Algorithm

Overview

The generator algorithm works by first creating a valid solved Sudoku board, then "punching holes" in the puzzle by applying a *mask*. The solved puzzle is created via an efficient backtracking algorithm, and the masking is performed via application of various *strategies*. A strategy is simply an algorithm that outputs cell locations to attempt to remove, based on some goal. After a cell entry is removed, the puzzle is checked to ensure that it still has a unique solution. If this test succeeds, another round is started. Otherwise, the board's mask is reverted, and a different strategy is consulted. Once all strategies have been exhausted, we do a final "cleanup" phase in which additional cells are removed systematically, then return the completed puzzle. For harder difficulties, we introduce *annealing*.

Completed Puzzle Generation

Completed puzzles are generated via backtracking. A solution is generated via some systematic method until a contradiction is found. At this point the algorithm reverts back to a previous state and attempts to solve the problem via a slightly different method. All methods are tried in a systematic manner. If a valid solution is found, then we are done.

Backtracking can be slow. To gain efficiency, we take the 2D Sudoku board and view it as a 1D list of rows. The problem reduces to filling rows with values; if we cannot, we backtrack to the previous row. We are finished if we complete the last row.

This recasting of the problem also simplifies the constraints; we need concern ourselves only with the values in each column and in the three clusters (or blocks) that the current row intersects. These constraints can be maintained by updating them each time a new value is added to a row.

Cell Removal

To change a puzzle to one that is entertaining to solve, we perform a series of removals that we call *masking*. One or more cells are removed from the puzzle (*masked out* of the puzzle), and then the puzzle is checked to ensure that it still has a unique solution. If this is not the case, the masking action is undone (or the cells are added back into the puzzle).

In *random masking*, every cell is masked in turn but in random order. Every cell that can be removed is, resulting in a minimal puzzle. This procedure is very fast and has potential to create any possible minimal puzzle, though with differing probability.

Tuned masking is slower and cannot create a puzzle any more difficult than random masking can. The idea is to tune the masking to increase the probability that a given type of puzzle is generated.

To create a board with a given WNEF, we apply strategies to reduce the WNEF. If we reach a minimum WNEF that is not low enough, we use a method from mathematical optimization, *simulated annealing*: We add some number of values back into the board and then optimize from there, in hope that doing so will result in a lower minimum. State-saving lets us to revert to the board with the lowest WNEF. Annealing allowed us to produce puzzles with lower WNEF values than we could have without it.

Uniqueness Testing

To ensure we generate boards with only one solution, we must test if this condition is met.

The fast solution uses Hidden Single and Naked Single: A cell with only one possible value can be filled in with that value, and any cell that is the only cell in some reference frame (such as its cluster, row, or column) with the potential of some value can be filed in with that value. These two logic processes are performed on a board until either the board is solved (indicating a unique solution) or no logic applies (which indicates the need to guess and hence a high probability that the board has multiple solutions). This test can produce false negatives, rejecting a board that has a unique solution.

The slow solution is to try every valid value in some cell and ask if the board is unique for each. If more than one value produces a unique result, the board has more then one solution. This solution calls itself recursively to determine the uniqueness of the board with the added values. The advantage of this approach is that it is completely accurate, and will not result in false negatives.

We used a hybrid method. It proceeds with the slow solution when the fast one fails. A further optimization restricts the number of times that the slow solution is applied to a board. This is similar to saying, "If we have to guess more then twice, we reject the board."

Complexity Analysis

Parameterization

We measure the time complexity t for generating a puzzle of difficulty d $t\left(d\right) = f\left(d\right) \cdot t_0$, where f is a function that we will find through our analysis and t_0 is the time complexity for generating a puzzle randomly.

Complexity of Completed Puzzle Generation

The puzzle generation algorithm works on each line of the Sudoku and potentially does so over all possible boards. In the worst case, we have the 9 possible values times the 9 cells in a line times 9 shifts, all raised to the 9 lines power, or $\left(9 \times 9 \times 9\right)^9 \approx 5.8 \times 10^{25}$. While this is a constant, it is prohibitively large. The best case is 81, where all values work on the first try.

However, in the average case, we not only do not cover all possible values, or cover all possible shifts, but we also do not recurse all possible times. So let us keep the same value for the complexity of generating a line (we have to try all 9 values, in all 9 cells, and perform all 9 shifts) but let us assume that we only do this once per line, getting $9^4 = 6561$; the actual value may be less or slightly more. We have a very high worst case but a very reasonable average case. In practice, the algorithm runs in time that is negligible in comparison to the masking algorithms.

Complexity of Uniqueness Testing and Random Filling

In the worst case, the "fast" uniqueness algorithm examines each of the 81 cells and compares it to each of the others. Thus, the uniqueness test can be completed in $81 \times 81 = 6,561$ operations. For the hybrid algorithm with brute-force searching, in the worst case we perform the fast test for each allowed guess plus one more time before making a guess at all. Therefore, the hybrid uniqueness testing algorithm has complexity linear in the number of allowed guesses.

The random filling algorithm does not allow any guessing when it calls the uniqueness algorithm and it performs the uniqueness test exactly once per cell. So it performs exactly $81^3 = 531,441$ comparisons.

Profiling Method

To collect empirical data on the complexity of puzzle generation, we implemented a profiling utility class in PHP. We remove dependencies on our particular hardware by considering only the normalized time $\hat{t} = t/t_0$, where t_0 is the mean running time for the random fill generator.

WNEF vs. Running Time

For the full generator algorithm, we can no longer make deterministic arguments about complexity, since there is a dependency on random variables. Hence, we rely on our profiler to gather empirical data about the complexity of generating puzzles. In particular, **Figure 10** shows the normalized running time required to generate a puzzle as a function of the obtained WNEF after annealing is applied. To show detail, we plot the normalized time on a logarithmic scale (base 2).

This plot suggests that even for the most difficult puzzles that our algorithm generates, the running time is no worse than about 20 times that of the random case. Also, generating easy puzzles can actually be faster than generating via random filling.

Testing

WNEF as a Function of Design Choices

The generator algorithm is fairly generic. We thus need some empirical way to specify parameters, such as how many times to allow cell removal

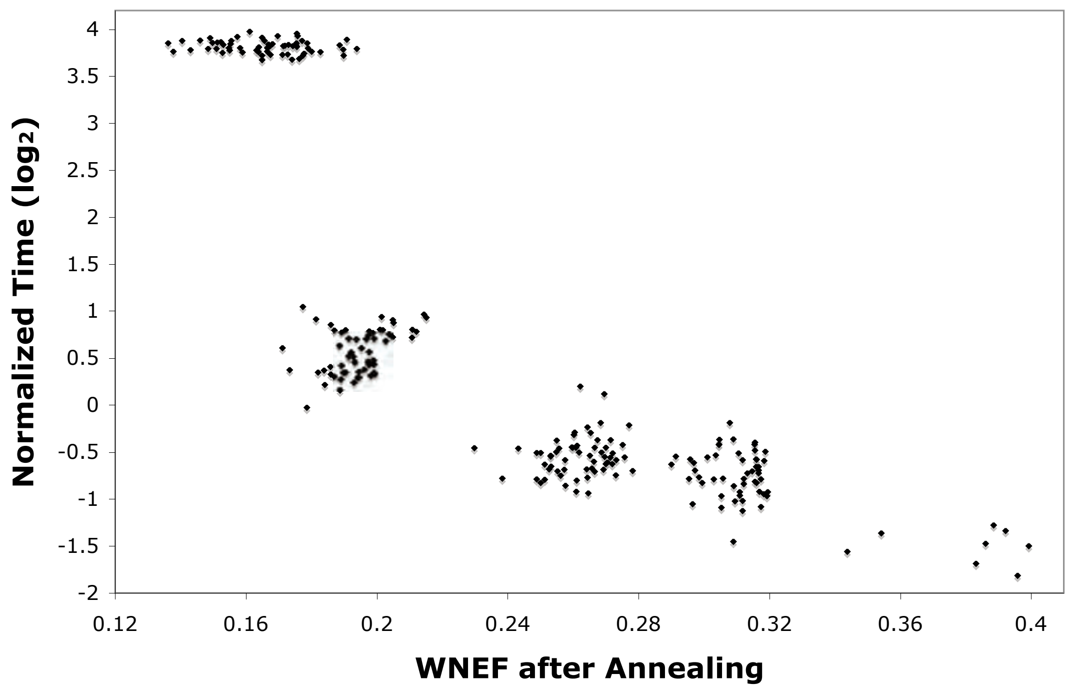


Figure 10. Log_2 of running time vs. WNEF.

to fail before concluding that the puzzle is minimal. We thus plotted the number of failures that we permitted vs. the WNEF produced, shown in **Figure 11**. This plot shows us both that we need to allow only a very small number of failures to enjoy small WNEF values, and that annealing reduces the value still further, even in the low-failure scenario.

Hypothesis Testing

Effectiveness of Annealing To show that the annealing resulted in lower WNEF values, and was thus useful, we tested the hypothesis that it was effective vs. the null hypothesis that it was not, using the mean WNEF for puzzles produced with annealing and the mean WNEF for those produced without it. A t-test at level of $\alpha = 0.0005$ concluded that annealing lowered the WNEF values.

Distinctness of Difficulty Levels To determine whether the difficulty levels of our puzzle generator are unique, we performed a t-test using the mean WNEF of puzzles produced by our generator algorithm with d as the target difficulty vs. those produced with target $d + 1$. For $d = 1, 2, 3$, concluded that the difficulty levels are indeed different.

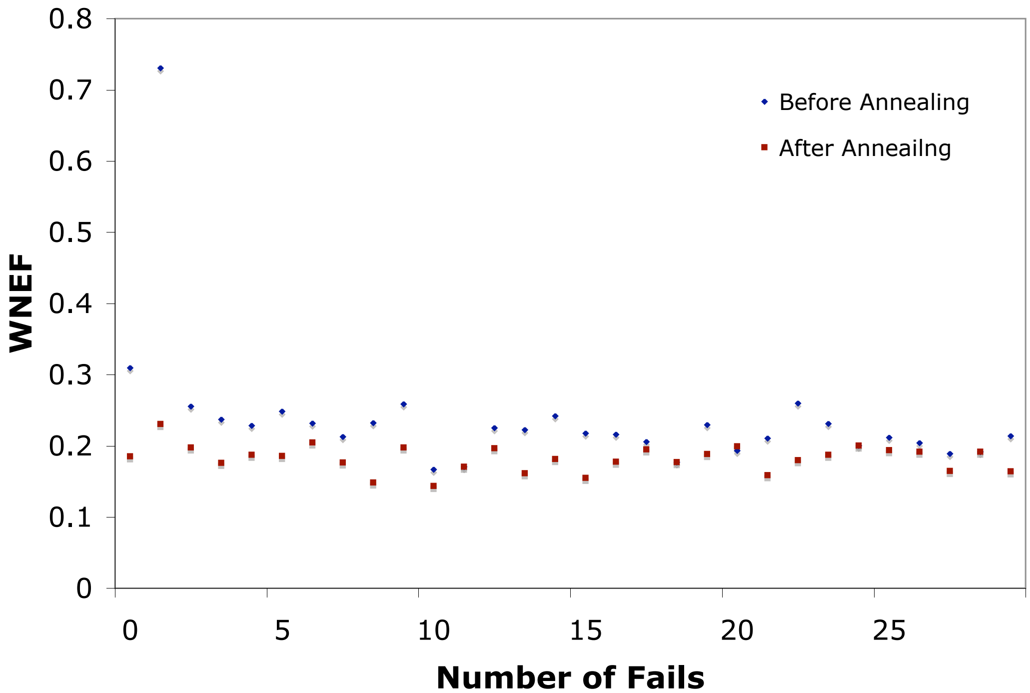


Figure 11. WNEF as a function of allowed failures.

Strengths and Weaknesses

It is possible to increase the difficulty of a puzzle without affecting its WNEF, by violating the assumption that all choices present similar difficulty to solvers. In particular, puzzles created with more-esoteric solving techniques, such as Swordfish and XY-Wing, can be crafted so that their WNEF is higher than easier puzzles. Thus, there is a limited regime over which the WNEF metric is useful.

On the other hand, the WNEF offers the notable advantage of being quick to calculate and constant for any puzzle difficulty, allowing us to make frequent evaluations of the WNEF while tuning puzzles.

Our generator algorithm seems to have difficulty generating puzzles with a WNEF lower than some floor, hence our decision to make our Evil difficulty level somewhat easier than published puzzles. The reason is that our tuning algorithm is still inherently a random algorithm and the probability of randomly creating a puzzle with a small WNEF value is very low.

The generator algorithm creates difficult puzzles quickly. Its method is similar to randomly generating puzzles until one of the desired difficulty is found (a method that is subject to the same disadvantage as ours), except that we can do so without generating more than one puzzle. We can generate a difficult puzzle in less time than it would take to generate multiple puzzles at random and discard the easy ones.

Conclusions

We introduce a metric, the weighted normalized ease function (WNEF), to estimate the difficulty of a Sudoku puzzle. We base this metric on the observation that the essential difficulty encountered in solving comes from ambiguities that must be resolved. The metric represents how this ambiguity is distributed across the puzzle.

Using data from control puzzles, we find that the WNEF shows a strong negative association with the level of difficulty (the harder the puzzle, the lower the WNEF). WNEF values of different difficulty levels are distinct. The choice of weighting function does not change this association.

We also designed an algorithm that employs these insights to create puzzles of selectable difficulty. It works by employing back-tracking and annealing to optimize the WNEF metric toward a desired level. Annealing leads to better results, and that the generator successfully produces puzzles falling into desired ranges of difficulty.

References

Aaronson, Lauren. 2006. Sudoku science. `http://spectrum.ieee.org/feb06/2809`.

Astraware Limited. 2005. Techniques for solving Sudoku. `http://www.sudokuoftheday.com/pages/techniques-overview.php`.

Ganni, J. 2008. *Games World of Sudoku* (April 2008). Blue Bell, PA: Kappa Publishing Group.

Greenspan, Gideon, and Rachel Lee. n.d. Web Sudoku. `http://www.websudoku.com`.

Hinkle, Tom. 2006. GNOME Sudoku. `http://gnome-sudoku.sourceforge.net/`.

Juillerat, N. 2007. Sudoku Explainer. `http://diuf.unifr.ch/people/juillera/Sudoku/Sudoku.html`.

Nikoli Puzzles. n.d. Sudoku tutorials. `http://www.nikoli.co.jp/en/puzzles/sudoku`.

Ostermiller, Stephen. 2007. QQwing—Sudoku generator and solver. `http://ostermiller.org/qqwing`.

Stertenbrink, Guenter. 2005. Sudoku. `magictour.free.fr/sudoku.htm`.

Taylor, A.M. 2008. *Dell Sudoku Challenge* (Spring 2008). New York: Dell Magazines.

Pp. 381–394 can be found on the *Tools for Teaching 2008* CD-ROM.

Cracking the Sudoku: A Deterministic Approach

David Martin
Erica Cross
Matt Alexander

Youngstown State University
Youngstown, OH

Advisor: George T. Yates

Summary

We formulate a Sudoku-puzzle-solving algorithm that implements a hierarchy of four simple logical rules commonly used by humans. The difficulty of a puzzle is determined by recording the sophistication and relative frequency of the methods required to solve it. Four difficulty levels are established for a puzzle, each pertaining to a range of numerical values returned by the solving function.

Like humans, the program begins solving each puzzle with the lowest level of logic necessary. When all lower methods have been exhausted, the next echelon of logic is implemented. After each step, the program returns to the lowest level of logic. The procedure loops until either the puzzle is completely solved or the techniques of the program are insufficient to make further progress.

The construction of a Sudoku puzzle begins with the generation of a solution by means of a random-number-based function. Working backwards from the solution, numbers are removed one by one, at random, until one of several conditions, such as a minimum difficulty rating and a minimum number of empty squares, has been met. Following each change in the grid, the difficulty is evaluated. If the program cannot solve the current puzzle, then either there is not a unique solution, or the solution is beyond the grasp of the methods of the solver. In either case, the last solvable puzzle is restored and the process continues.

Uniqueness is guaranteed because the algorithm never guesses. If there

The UMAP Journal 29 (3) (2008) 381–394.

is not sufficient information to draw further conclusions—for example, an arbitrary choice must be made (which must invariably occur for a puzzle with multiple solutions)—the solver simply stops. For obvious reasons, puzzles lacking a unique solution are undesirable. Since the logical techniques of the program enable it to solve most commercial puzzles (for example, most "evil" puzzles from Greenspan and Lee [2008]), we assume that demand for puzzles requiring logic beyond the current grasp of the solver is low. Therefore, there is no need to distinguish between puzzles requiring very advanced logic and those lacking unique solutions.

Introduction

The development of an algorithm to construct Sudoku puzzles of varying difficulty entails the preceding formulation of a puzzle-solving algorithm. Our program (written in C++) contains a function that attempts to generate the solution to a given puzzle. Four simple logical rules encompass the reasoning necessary to solve most commercially available Sudoku puzzles, each more logically complex than the previous. The varying complexity establishes a somewhat natural system by which to rate the difficulty of a puzzle. Each technique is given a weight proportional to its complexity; then, difficulty is determined by a weighted average of the methods used. Our algorithm places each puzzle in one of four categories that we identify as Easy, Medium, Hard, and Very Hard.

The lowest level of logic is the most fundamental method used by our program (and humans!) in an attempt to solve a Sudoku puzzle. When a level of reasoning can no longer be used, the next level of logic is prompted. A successful attempt at this new level is followed by a regression back to the lowest level of logic employed. A failed attempt at the new stage initiates a further advance in logic. The procedure loops until the problem is completely solved or no more progress can be made. Consistency is guaranteed by the use of a check function, which verifies that each row, column, and box contains each of the digits 1 to 9 without duplication. If the techniques are inadequate to solve a puzzle, the loop terminates.

Our algorithm constructs Sudoku puzzles in a somewhat "backward" manner. First, a completed Sudoku is formulated using a simple random-number-based function, similar to many brute-force methods of solving the puzzles. Before puzzle generation begins, the user enters restrictions such as desired difficulty level and the maximum number of cells that are initially given. Creating a puzzle begins by randomly eliminating digits from one cell at a time. The elimination process continues until the conditions specified are met. After each removal, the program attempts to solve the existing puzzle.

A Sudoku puzzle cannot be solved in one of two scenarios:

- The puzzle no longer has a unique solution. The algorithm is deterministic and only draws conclusions that follow directly from the current state of the puzzle. In such a case, because an arbitrary decision must be made, the algorithm simply terminates.

- The logical methods available to the solver are not sufficient to solve the puzzle.

In either circumstance the program restores the last solvable puzzle and resumes the process.

Due to the undesirable nature of both ambiguous puzzles and puzzles that require guessing, the algorithm never guesses. If there exists a unique solution for a given puzzle, then failure to solve implies that the puzzle requires logical methods higher than those written into the program. This conclusion is appropriate, since demand is low for Sudoku puzzles requiring extremely sophisticated logical methods. Thus, our algorithm does not distinguish between puzzles with no solution and those requiring more-advanced logic.

Definitions

Cell: A location on a Sudoku grid identified by the intersection of a row and a column, which must contain a single digit.

Row: A horizontal alignment of 9 cells in the Sudoku grid.

Column: A vertical alignment of 9 cells in the Sudoku grid.

Box: One of the nine 3×3 square groups of cells that together comprise the Sudoku grid.

Group: A row, column, or box on the Sudoku grid that must contain each digit from 1-9 exactly once.

Given: A cell whose answer is provided at the beginning of the puzzle.

Candidate: A possible solution for a cell that was not given.

Method: The technique used to eliminate candidates as possibilities and solve cells.

Unique: The puzzle is considered unique when it has a unique solution.

Difficulty: The level of skill needed to solve the puzzle, based on the complexity and frequency of the methods required to solve it.

Assumptions

- We work only with the classic Sudoku grid consisting of a 9×9 square matrix.

- Guessing, while a form of logic, is not a deterministic method. Demand is low for Sudoku puzzles that require nondeterministic logic. All puzzles at Greenspan and Lee [2008] can be solved without guessing (or so the site claims).
- Demand is low for puzzles requiring extremely complicated logical methods. Our algorithm solves all Easy, Medium, Hard, and some Very Hard puzzles.
- The difficulty of a puzzle can be calculated as a function of the sophistication and frequency of the logical methods demanded.
- The ordering of a given set of puzzles by difficulty will be the same for the program as for humans, because the solver uses the same techniques employed by humans.

Model Design

The Solver

The program is based on simple logical rules, utilizing many of the same methods employed by humans. Like humans, the program begins solving each puzzle with the lowest level of logic necessary. When all lower methods have been exhausted, the next echelon of logic is implemented. After each step, the program returns to the lowest level of logic, so always to use the lowest possible level of logic. The procedure loops until either the problem is completely solved or the logical techniques of the program are insufficient to make further progress. The following techniques are included in the algorithm.

1. **Naked Single:** a cell for which there exists a unique candidate based on the circumstance that its groups contain all the other digits [Davis 2007]. In **1**, the number 1 is clearly the only candidate for the shaded cell.

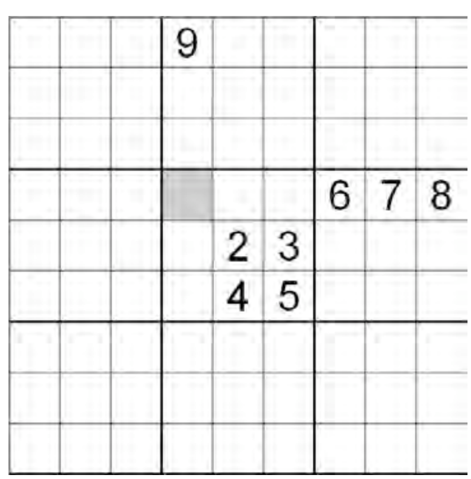


Figure 1. Naked Single.

2. **Hidden Single:** a cell for which there exists a unique candidate based on the constraint that no other cell in one of its groups can be that number [Davis 2007]. In **Figure 2**, the shaded cell must be a 1.

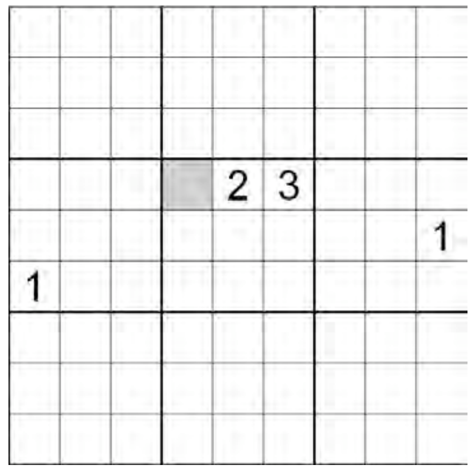


Figure 2. Hidden Single.

3. **Locked Candidate:**

   A. A method of elimination for which a number within a box is restricted to a specific row or column and therefore can be excluded from the remaining cells in the corresponding row or column outside of the selected box [Davis 2007]. In **Figure 3**, none of the shaded cells can be a 1.

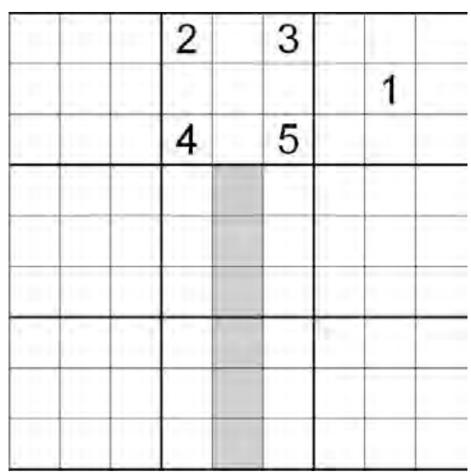


Figure 3. Locked Candidate (box).

   B. A method of elimination for which a number within a row or column is restricted to a specific box and therefore can be excluded from the remaining cells within the box. In **Figure 4**, again, none of the shaded cells can be a 1.

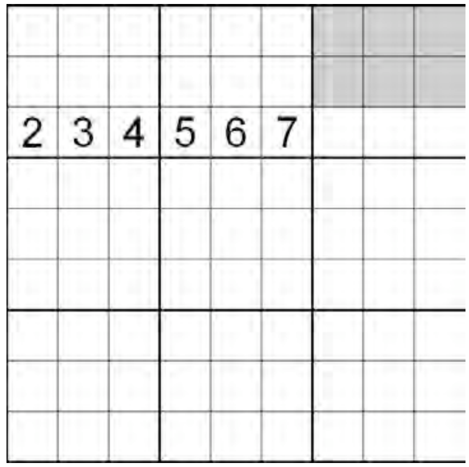


Figure 4. Locked Candidate (rows and columns).

4. **Naked Pairs:** This method of elimination pertains to the situation in which two numbers are candidates in exactly two cells of a given group. Consequently, those two numbers are eliminated as candidates in all other cells within the group [Davis 2007]. In **Figure 5**, none of the shaded cells can contain either a 1 or 2.

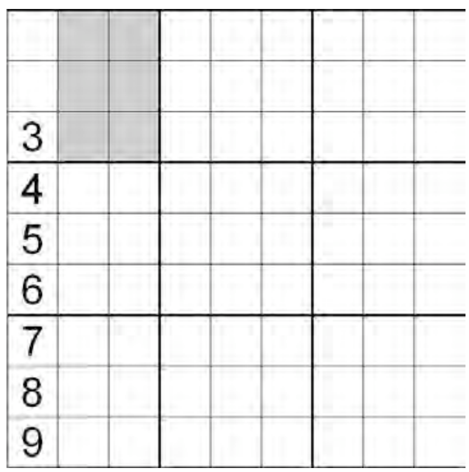


Figure 5. Naked Pairs.

The overall algorithm is represented by the diagram in **Figure 6**.

Difficulty

The algorithm is based on techniques commonly employed by humans; so, for a given set of puzzles, the ordering by difficulty will be about the same for the program as for humans, making the difficulty rating produced by the program of practical value.

As the solver works on a puzzle, it keeps track of the number of times that it uses each level of logic. Let $i \in \{1, 2, 3, 4\}$ correspond to a logic

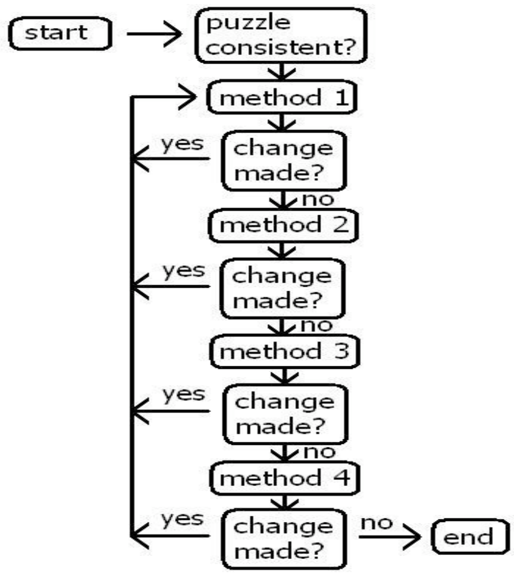


Figure 6. Puzzle Solver.

level discussed above (naked single, hidden single, locked candidate, naked pairs). Let n_i be the number of times that technique i is used. The difficulty rating can then be calculated by means of a simple formula:

$$D(n_1, n_2, n_3, n_4) = \frac{\sum w_i n_i}{\sum n_i},$$

where w_i is a difficulty weight assigned to each method. Naturally, the weight should increase with the complexity of the logic used in a technique.

As the proportion of changes $n_i / \sum n_i$ made by a method increases, the difficulty value approaches the weight assigned to that technique. In practical application, higher methods are used extremely rarely. Therefore, seemingly disproportionately high weights should be assigned to the higher methods for them to have an appreciable effect on difficulty. The choice of these values is somewhat arbitrary, and small changes are not likely to have an appreciable effect on the ordering by difficulty of a set of puzzles.

For our purposes, we used $w_1 = 1$, $w_2 = 3$, $w_3 = 9$, and $w_4 = 27$. In application, these values provide a nice spectrum ranging from 1 (only the first level of logic is used) to about 4 (the higher levels are used frequently). One of the hardest puzzles generated by the program required the use of techniques one, two, three, and four 42, 11, 10, and 2 times, respectively with corresponding difficulty rating

$$D = \frac{42 \cdot 1 + 11 \cdot 3 + 10 \cdot 9 + 2 \cdot 27}{42 + 11 + 10 + 2} \approx 3.37.$$

Difficulty categories can be determined by partitioning the interval [1,4] into any four subintervals. We determined the reasonable subintervals:

Easy , $D \in [1, 1.5)$. A typical Easy puzzle with a rating of 1.25 requires use of the second level of logic 7 times.

Medium , $D \in [1.5, 2)$. A typical Medium puzzle with a rating of 1.7 requires the use of the second level of logic 17 times and the third level once.

Hard , $D \in [2, 3)$. A typical Hard puzzle with a rating of 2.5 requires the use of the second level of logic 17 times, of the third level 4 times, and of the fourth level once.

Very Hard , $D \in [3, 4]$. The aforementioned puzzle, with 3.37, required the use of the second method 11 times, of the third method 10 times, and of the fourth method twice.

The Puzzle Creator

Rather than starting with an empty grid and adding numbers, the program begins with a completed Sudoku, produced by a random-number-based function within the program. The advantage is that rather than hoping to stumble upon a puzzle with a unique solution, the program begins with a puzzle with a unique solution and maintains the uniqueness.

Once a completed Sudoku grid has been created, the puzzle is developed by working backwards from the solution, removing numbers one by one (at random) until one of several conditions has been met. These conditions include a minimum difficulty rating (to ensure that the puzzle is hard enough) and a minimum number of empty squares (to ensure that the puzzle is far from complete). Following each change in the grid, the difficulty is evaluated as the program solves the current puzzle. If the program cannot solve the current puzzle, then either the puzzle does not have a unique solution or the solution is beyond the grasp of the logical methods of the algorithm. In either case, the last solvable puzzle is restored and the process continues (see **Figure 7**).

In theory, a situation may occur in which removing any number will yield a puzzle that is not solvable (by the algorithm) but has a unique solution. In such a case, the puzzle creator has reached a "dead end" and cannot make further progress toward a higher difficulty rating. To overcome this obstacle, the program, rather than generating a single puzzle as close as possible to a given difficulty rating, generates 1,000 puzzles and sorts them by difficulty. In this manner, the program produces a virtual continuum of difficulties ranging from 1.0 to whatever difficulty was requested (within the limits of the program, which cannot produce puzzles that are harder than about 4).

Uniqueness

Uniqueness is guaranteed because the algorithm never guesses.

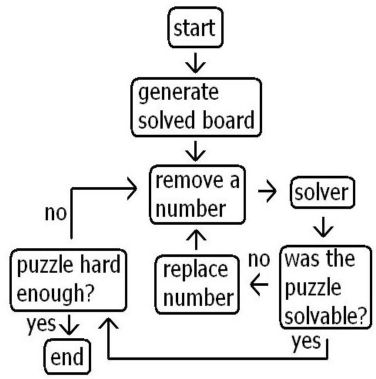


Figure 7. Puzzle Creator.

Difficulty

Since the logical techniques possessed by the program enable it to solve most commercial puzzles, we assume that demand for puzzles requiring logic beyond the current grasp of the solver is low. Therefore, there is no need to distinguish between puzzles requiring very advanced logic and those lacking unique solutions.

Model Testing (Relevance of the Difficulty Rating)

To determine the relevance of the algorithm to real-world Sudoku puzzles, we set our program loose on 48 randomly selected puzzles from three popular Websites [Greenspan and Lee 2008; ThinkFun Inc. 2007; and www.LearnToUseComputers.com]. Four puzzles were selected from each of four difficulty categories for each source. The difficulty levels assigned by our program and a summary of our results are in **Tables 1** and **2**.

All puzzles labeled by the source as Easy, Medium, and Hard (or an equivalent word) were solved successfully and rated. Some—but not all—of the Very Hard puzzles were solved successfully and rated; those beyond the grasp of our program were simply given a rating of 4.0, the maximum rating in the scale. Although the algorithm was able to crack exactly one half of the Very Hard puzzles attempted from both Greenspan and Lee [2008] and www.LearnToUseComputers.com [2008], it solved none of the Very Hard puzzles from ThinkFun Inc. [2007].

Under the suggested partition ($[1, 1.5)$, $[1.5, 2)$, $[2, 3)$, and $[3, 4]$), all of the puzzles labeled by the source as Easy (or equivalent) were awarded the same rating by our program. Agreement was excellent with ThinkFun Inc. [2007], with which our program agreed on 13 of the 16 puzzles tested (the four puzzles from that source with a rating of Very Hard were not solved

Table 1. Performance summary.

Label Awarded by Source	Websudoku	Puzzles	Sudoku Puzzles Online
Easy	Easy	Easy	Easy
	Easy	Easy	Easy
	Easy	Easy	Easy
	Easy	Easy	Easy
Medium	Easy	Easy	Easy
	Easy	Medium	Easy
	Easy	Medium	Easy
	Easy	Medium	Medium
Hard	Easy	Easy	Medium
	Easy	Medium	Medium
	Medium	Hard	Easy
	Medium	Hard	Medium
Very Hard	Hard	Not Solved	Hard
	Very Hard	Not Solved	Very Hard
	Not Solved	Not Solved	Not Solved
	Not Solved	Not Solved	Not Solved

Table 2. Difficulty rating.

		Websudoku	Puzzles	Sudoku Puzzles Online
Easy	Average Difficulty	1.0	1.1	1.0
	Proportion Solved	4 / 4	4 / 4	4 / 4
Medium	Average Difficulty	1.2	1.6	1.3
	Proportion Solved	4 / 4	4 / 4	4 / 4
Hard	Average Difficulty	1.6	1.9	1.7
	Proportion Solved	4 / 4	4 / 4	4 / 4
Very Hard	Average Difficulty	3.5	4.0	3.5
	Proportion Solved	2 / 4	0 / 4	2 / 4

by the algorithm and received a difficulty rating of 4.0 by default). The three puzzles for which the algorithm and ThinkFun Inc. [2007] disagreed were given a lower difficulty rating by the program. The program successfully solved four Very Hard puzzles from the other two sources but was apparently not too impressed, awarding half of those solved a mere Hard rating. In the Medium and Hard categories, puzzles from Greenspan and Lee [2008] and www.LearnToUseComputers.com [2008] were consistently awarded lower difficulty ratings than those suggested by the source.

Model Analysis

Complexity Analysis

The Solver

The puzzle-solving algorithm is surprisingly short, relying on a simple topology connecting a hierarchy of just four logical methods. At first glance, one might suspect that most puzzles would be beyond the scope of the four logical operations available to the solver. However, as seen above, the algorithm does not meet its match until it is presented with puzzles labeled as Very Hard.

The Puzzle Creator

At the start of the process, a mysterious procedure randomly creates a solved puzzle. This procedure is not complicated, and can be summarized as follows.

Going from the top of the puzzle to the bottom, and from the left to the right, a random digit is inserted into each empty box. The program checks for consistency. If insertion of the digit violates a constraint of the Sudoku puzzle, then another digit is attempted. After a fixed number of attempts have failed at a given cell (in fact, no digit may be possible if there remain no candidates for a given cell), the program removes both of the digits involved in the last contradiction. This allows the program to dismantle relationships that make a puzzle unsolvable. The process loops until the puzzle is both consistent and complete (no empty spaces).

The rest of the puzzle creation process is largely the inverse of the above procedure, except that rather than inserting numbers and checking for consistency and completeness, the program removes numbers and checks for solvability and uniqueness (which are equivalent for reasons discussed above) as well as constraints pertaining to difficulty rating and number of givens.

Sensitivity Analysis

The primary source of arbitrariness in the model is the method by which difficulty ratings are established. It requires the user to assign to each logical technique a weight proportional to its complexity and its contribution to the overall difficulty of the puzzle. We assigned weights of 1, 3, 9, and 27 to the levels of logic. The exact values are relatively unimportant, evidenced by the fact that two additional sets of weights produced exactly the same ordering by difficulty of a set of eight typical puzzles created by the program. **Table 3** summarizes the relative independence of the ordering on weight values.

Table 3. Sensitivity analysis.

LOGIC LEVEL	WEIGHTING SYSTEM 4^N	3^N	2^N	2N+1
1	1	1	1	1
2	4	3	2	3
3	16	9	4	5
4	64	27	8	7

PUZZLE	4^N	3^N	2^N	2N+1
A	1.00	1.00	1.00	1.00
B	1.47	1.32	1.16	1.16
C	1.96	1.61	1.29	1.27
D	2.66	2.00	1.45	**1.40**
E	3.52	2.29	1.48	**1.35**
F	3.87	2.61	1.66	**1.51**
G	5.38	3.06	1.70	**1.46**
H	5.78	3.28	1.78	1.52

Although all three of the exponential weight systems produce the same difficulty rating, the linear system does not. Because the featured system, which uses the weights of 1, 3, 9, and 27, agrees so well with ThinkFun Inc. [2007], it seems safe to say that the exponential weighting system makes much more sense (at least with the current hierarchy of logical techniques) than the linear.

Shortcomings of the Model

- We assume that four levels of logic are sufficient to solve any Sudoku puzzle, though other techniques exist.
- The model lacks the capacity to solve some "evil" puzzles featured on Greenspan and Lee [2008], due to the absence of more-complex methods within the program.
- Our model reports an error for Sudoku puzzles that either have no solution or multiple solutions but does not differentiate between the two.

Strengths of the Model

- Our model considers the fact that once a higher level of logic is used and a cell is filled, a human will return to attempting a solution with the simplest method of logic, and therefore so does our program.
- Utilizing a functional program, we were able to construct and evaluate the difficulty of a thousand Sudoku puzzles in a matter of minutes.
- The program uses deterministic logic in each method featured in the program and does not resort to guessing.
- The code can be easily expanded to include more advanced levels of logic such as naked triplets and quads, x-wings and swordfish, or coloring.
- The code could also be easily modified to do other types of Sudoku puzzles such as a 16×16 grid and other rules for the puzzle.

Conclusion

In spite of the seemingly small scope of the four logical operations available to the solver, the algorithm solved all Easy, Medium, and Hard puzzles from three popular Internet sources and one-third of their Very Hard puzzles. Therefore, a small set of logical rules is sufficient to solve nearly all commercially available Sudoku puzzles.

To expand the scope of the solver, the overall complexity of the algorithm need not be increased. Simply adding another logical technique to the loop can increase the solving power. A mere two or three additional methods would probably suffice to enable the program to solve all commercially available puzzles that do not require guessing.

Just For Fun

The puzzle in **Figure 8**, created by the program and given a difficulty rating of 3.52 (Very Hard), requires the use of methods one, two, three, and four 45, 12, 9, and 3 times, respectively. Have fun!

						1		
4					6	7		9
	6		2	1				
			5		3		1	
	5	8		7				3
			6					
9				8				
		2	4				5	
7			3					2

Figure 8. Difficulty 3.52.

References

Davis, Tom. 2007. The mathematics of Sudoku. `http://www.geometer.org/mathcircles/sudoku.pdf` . Accessed 15 February 2008.

Greenspan, Gideon, and Rachel Lee. 2008. Web Sudoku. `http://www.websudoku.com/` .

ThinkFun Inc. 2007. Sudoku 12. `http://www.puzzles.com/PuzzleLinks/Sudoku12.htm` .

www.LearnToUseComputers.com. 2008. MPS-Sudoku 2008. `http://www.sudokupuzzlesonline.com`, which defaults to `http://www.ltucshop.com/prodtype.asp?strParents=70&CAT_ID=71&numRecordPosition=1` .

Matt Alexander, George Yates (advisor), Erica Cross, and David Martin.