ESSAYS IN
MARKETING, ECONOMICS, AND OPTIMIZATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
MANAGEMENT SCIENCE AND ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DING MA

NOVEMBER 2018

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Wesley Hartmann, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Michael Saunders, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Kenneth Judd

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Michael A Saunders

Michael A. Saunders, Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Wesley R. Hartmann, Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Kenneth L Judd

Kenneth L. Judd

Approved for the University Committee on Graduate Studies.

ABSTRACT

This thesis includes four self-contained essays on marketing, economics, and optimization, all sharing a common theme: creating numerical models and algorithms to tackle computationally challenging optimization problems.

The first essay considers geographic sub-branding via manufacture location in marketing. Manufacture location, as part of geographic product identity, is becoming a significant differential factor among a variety of products and a sub-branding element in various markets, but there is little empirical research on how manufacture location influences consumer preference and purchase choices. The first barrier comes naturally from the market. Most of the time, manufacture location is completely correlated with product characteristics. I was fortunately able to acquire both data and a whole year research grant from Ford Motor Company. New car buyer data in the Chinese automobile market makes possible the analysis of manufacture location as a geographic sub-branding element. Hedonic price analysis gives us a quick and intuitive view of how much consumers are willing to pay for geographic sub-branding products, and also motivates our new brand and sub-brand definition for a BLP-type random coefficient discrete choice model. However, using General Method of Moments (GMM) to estimate the variance-covariance matrix of consumer brand taste coefficients poses another challenge for all existing optimization solvers. To prevent indefinite unknown variance-covariance matrices in the constraints of the optimization problem from terminating the solver, I reformulate the optimization program and successfully solve the problem. If the computation becomes more difficult, we can also apply our new algorithm NCL, which is described in details in the third essay. Our results reveal the strong substitution patterns in the market and confirm our definition and framework of brand and geographic identity. Furthermore, our method can be helpful in the analysis of branding and sub-branding in other empirical settings.

The second essay studies optimal income taxation with multidimensional taxpayer types in economics. This engendered the subject of the third essay: stabilized optimization via an NCL algorithm in numerical optimization. The income taxation literature has generally focused on economies where individuals differ only in their productivity, i.e., income. In reality, people differ in many ways. In our

models, we consider people's productivity, basic needs, distaste for work, the elasticity of labor supply, and the elasticity of demand for consumption. We find that extra dimensions give us substantially different and interesting results. In certain cases, high-productivity people may pay negative tax. Therefore, considering income taxation in multiple dimensions is essential, and again is computationally challenging. All existing optimization solvers fail to find optimal solutions. Eventually, we transformed the model, created a new algorithm (NCL), and solved the high-dimension difficult optimization problems. The nonlinearly constrained augmented Lagrangian algorithm (NCL) we created was motivated by the bound-constrained and the linearly constrained augmented Lagrangian algorithms (BCL and LCL). To facilitate implementation, we take advantage of the mathematical programming language AMPL. We did not have to write fifty thousand lines of Fortran code to implement NCL. The third essay on algorithm NCL was published this year in *Numerical Analysis and Optimization*. The taxation part remains as a working paper. I am excited about not only solving the complex income taxation models, but also creating a general algorithm that can be applied to tough mathematical models in different fields. For instance, our algorithm NCL can be easily adapted to nonlinear pricing in economics and marketing.

The fourth essay considers reliable and efficient solution of genome-scale models of metabolism and macromolecular expression. For many years, scientific computing has advanced in two complementary ways: improved algorithms and improved hardware. In order to solve the large and complicated biochemical network of metabolism in systems biology, we made use of improved machine precision and created algorithms utilizing software-simulated quadruple precision arithmetic and successfully solved both original and reformulated optimization models. This essay is published in *Scientific Reports*. Today, our linear and nonlinear quadruple precision solver quadMINOS is supporting the research of systems biologists in their COBRA (COnstraint-Based Reconstruction and Analysis) Toolbox, and it can also help researchers in many other areas. As my advisor Professor Michael Saunders predicts: *"Just as double precision floating-point hardware revolutionized scientific computing in the 1960s, the advent of the quadruple precision data type, even in software, brings us to a new era of greatly improved reliability in optimization solvers."*

ACKNOWLEDGMENTS

My Stanford experience has brought me so much more than the academic achievements that I put into this thesis. I have also learned that the scientific contribution is only one part of a researcher's mission. This work would not be possible without the love and support given me by people in my life. Although I will probably forget to mention some of them in the coming paragraphs, I wish to express my gratitude to all of these people.

First and foremost, my gratitude goes to my advisors, Michael Saunders, Wesley Hartmann, and Kenneth Judd.

Michael, I am so fortunate to have you as my advisor. Advised by you, creator of the industrial-strength optimization solvers MINOS and SNOPT, gives me the privilege to access cutting-edge mathematical models across different fields. I am so grateful that I have the opportunity to work with you on many projects and papers, which led to adventures in amazing places, like Merida of Mexico, Tokyo of Japan, Muscat of Oman, etc. In Oman when I was coding in AMPL, I had David Gay, one of the inventors of AMPL. When I was calling SNOPT, of course I had you, one of the creators of SNOPT. Regardless of your achievements, you remained extremely humble and kind. You are diligent in almost everything, even the details of wording and format of our papers, not to mention our codes. You are so much more than an advisor; you are my supporter, role model, and friend. Michael, thank you for the support and freedom you have given me. For the past six years, I owe it all to you!

Wes, about two years ago, I sat in your class and started my marketing journey with you. We both know that non-marketing students on the marketing job market are not at an advantage. I still remember the time you challenged me to think of my own topic instead of doing a consultation case. I skimmed through the marketing literature. One week later, I came up with a list of potential topics suitable for the data, and you pointed to one of them and told me to dig deeper there. If it were not for you, the marketing paper in this thesis would not have started yet. Wes, it has been a real pleasure to work with you. Thank you for the opportunity, guidance, mentorship you have provided me. They will stay with me as I carry on my marketing journey!

Ken, I could never imagine that I would have the opportunity to work with the legendary Ken Judd. Every time we meet, I am amazed by the depth and breadth of your knowledge. Every minute we talk, I feel like I need to be a sponge absorbing every drop of your wisdom flowing in the air. The unknown part of your legend is that you are sharp but also super encouraging. Sometimes I have to wonder if I deserve your outspoken praise. Ken, your affirmation means a lot to me. Thank you for your insight and guidance. For the past two years, it has been my honor.

In addition to my advisers, Professors Bart Bronnenberg and Trevor Hastie also served on my dissertation committee. They are exemplary people and researchers. I am very thankful for their insightful comments. I was fortunate to have many other brilliant researchers as my co-authors, especially Laurence Yang, Ronan Fleming, and Dominique Orban, from whom I have learned so much. I would also like to thank Yan Fu, Ross Morrow, and Zheng Jiang at Ford for their kind support.

There are so many others I owe credit to that I will probably forget to mention some. If I do, please forgive me. To begin, I wish to thank the MS&E faculty, especially Yinyu, Ross, Itai, and Markus, and the Marketing faculty, especially Pedro, Jim, Sridhar, and Navdeep, for their generous support and guidance. I also wish to thank my colleagues and friends in MS&E: Ruixue, Jing, Lingren, Rob, Nick, Dan, Ehsan, Afshin, Zeyu, Arpit, Hong, Oliver, Hongyang, Anilesh, Stephen, Lin, and Ruoxuan; and in Marketing: Megan, Ilya, Jessica, James, Yingze, and Shreya; and in ICME: Yuekai, Santiago, Jiyan, Chao, and Xiaotong, for all the good and tough times over the past six years.

Especially, I wish to thank my big family, my parents, and my seven-year-old daughter Emma, for their unconditional love and support. They are always there for me, to carry me through ups and downs. They are the essence of my very being. My eternal gratitude is beyond expression.

My work was supported by the National Institute of General Medical Sciences of the National Institutes of Health (award U01GM102098) and the Ford-Stanford Alliance. The content is solely the responsibility of the author and does not necessarily represent the official views of the funding agencies.

CONTENTS

Part I

GEOGRAPHIC SUB-BRANDING VIA MANUFACTURE LOCATION: HOW DOES “MADE IN” CHANGE YOUR PURCHASE?

1

INTRODUCTION

Manufacture Location, as part of geographic product identity, is becoming a significant differential factor among a variety of products and a sub-branding element in various markets, but there is little empirical research on how manufacture location influences consumer preference and their purchase choices. We pursue this topic by analyzing consumer willingness to pay and choice preference in the context of a durable good: automobiles. Using the data from the automobile market in China, hedonic price analysis shows that manufacture location significantly influences prices and consumer willingness to pay. We find the “imported" manufacture location effect regarding the price premium as high as 9.6% with the control of product characteristics. With our definition and framework of the brand and geographic identity, we build a random coefficient discrete choice model to estimate the correlation between consumer taste parameters of different vehicle models that share the same brand identity or geographic identity at the manufacture location level. Our method can also be helpful in the analysis of branding and sub-branding in other empirical settings.

According to the American Marketing Association, a brand is a name, term, design, symbol, or any other feature that identifies one seller's good or service as distinct from those of other sellers.[1] Note that brands also recognize distinct differences between products sold by a single seller. For instance, Volkswagen Group owns 12 brands including VW, Audi, Bentley, Bugatti, Lamborghini, Porsche, etc. [2] The brand of a firm has become an essential part of its assets in financial valuations, especially when merger or acquisition happens. The brand of a product or service unsurprisingly influences consumers' seemingly reasonable yet irrational choices, which in turn steers companies' daily marketing strategy like target marketing and promotions. Farquhar (1989) and Aaker (1991) have studied different perspectives of brand equity. Keller (1993) conceptualizes brand equity from the perspective of the individual consumer and provides a conceptual framework for what consumers know about brands and what such knowledge implies for marketing strategies. Brand knowledge is defined in terms of two components, brand

[1] This definition is from: https://www.ama.org/resources/Pages/Dictionary.aspx?dLetter=B.

[2] A complete list can be found on the company website: http://www.volkswagenag.com/en/brands-and-models.html.

awareness and brand image, where brand image reflects brand associations including product-related or non-product-related attributes; functional, experiential, or symbolic benefits; and overall brand attitudes. These associations are differentiated based on their favorability, strength, and uniqueness. Conceptualizing brand identity from a consumer's perspective gives marketers a broad and detailed view of the effects of marketing activities on consumer preference as well as more traditional outcome measures such as sales. Consistent with Keller (1993), also from a consumer's perspective, we summarize the brand identity as a representation of observable and unobservable (tangible and intangible) product characteristics.[3]

As a trend of globalization, companies have been producing and distributing their products in different countries, and hence brand identity has been extended over the globe. When consumers face similar products under a variety of brand names, the geographic identity of products also serves branding roles and recognizes differences across products. One notion of geographic branding arises when it summarizes common characteristics or qualities of brands originating in the same country or area. Brand origin, also known as country of origin (COO), inevitably plays a significant role in consumers' perception of the product. For example, German automobile companies have known reputations for design and manufacturing expertise that may provide a favorable impression of a German brand, even if the brand is previously unknown to the consumer. Gradually, we also start to pay more attention to the "Made in" label and infer the potential benefit we could get from a product based on its manufacture location, also known as country of manufacture (COM). To investigate this intertwined interaction between the brand identity and geographic identity of a product, we provide a hierarchical framework shown in Figure 1.

While the previous notion of geographic identifiers such as German automobile is viewed hierarchically above a traditional brand name, manufacture location serves hierarchically below brands as a sub-branding element. For instance, Coca-Cola markets its Mexican Coke in parts of the United States, which signifies a different taste arising from the unique formula used to produce Coke in Mexico.[4] Similarly, in China, automobile companies have Chinese generated variants of their vehicles that may differ from their foreign produced counterparts concerning

[3] From consumers' perspectives, tangible characteristics could be unobservable as well. For example, according to Steve Jobs, "A lot of times, people don't know what they want until you show it to them."

[4] Coca-Cola's Mexican produced soda is marketed separately alongside the US produced Coke. Coca-Cola claims that Mexican Coke produced in Mexico and exported to the United States is made with cane sugar instead of high-fructose corn syrup. The classic glass bottled Mexican Coke has become very popular among consumers in the US.

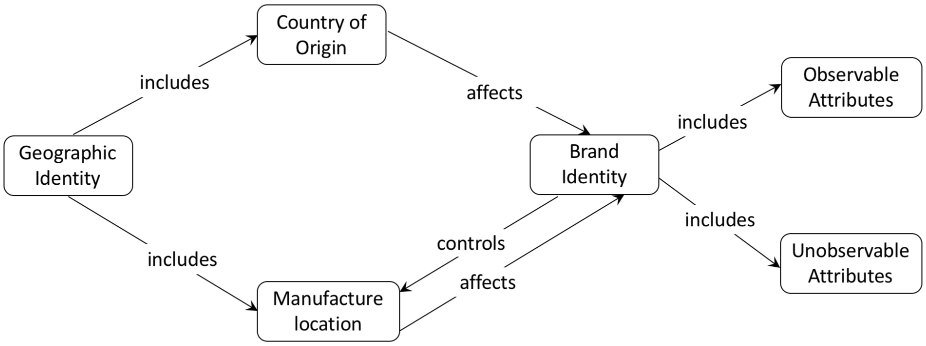

Figure 1: The hierarchical interaction between brand identity and geographic identity.

manufacturing ability, parts sourcing, etc., despite identical designs and model versions. In this study, we seek to understand how geographic identifiers arise as a sub-branding element and the role they play in driving consumer perceptions about what may often be nearly identical offerings by the same parent company brand.

As a matter of fact, research on the top layer of geographic identity, country of origin, started even earlier than branding. Schooler (1965) first studied the country of origin effect in the Central American market and showed that the attitude toward the people of a given country is a factor in existing preconceptions regarding the products of that country. Since then, there has been vibrant literature on the COO effect. Dinnie (2004) gives a comprehensive survey of this literature. More recently, Saridakis and Baltas (2016) weighted COO against the product attributes regarding price premium. COO is gaining more and more leverage over the brand value, especially in international markets.

Meanwhile, manufacture location, the bottom layer of geographic identity in Figure 1, has also become an influential factor in brand identity. It seems that the "Made in" phenomenon emerges with globalization, but manufacture location organically taking on a branding role dates back to the Soviet Union, where manufacturers had no concern about product differentiation or branding. “Production mark" (essentially the manufacture location) became mandatory for the purpose of quality control. Goldman (1960) uses the Soviet Union as an extreme yet great example to show the importance and benefits of product differentiation. In the planned Soviet economy, the geographic identity, a "production mark" on the good or packaging, was probably the only identity of a product, summarizing and assuring quality.

Today, we no longer need to defend product differentiation as Goldman (1960) did. Companies are doing everything to differentiate themselves from their competitors. Intuitively, consumers expect that globalization brings them the same products previously unavailable. However, most firms are not unifying their products to build their global brand. The more prevailing phenomenon is that differences in inputs, production processes, and local taste lead to regional variants of given branded products.

Companies compete in branding and sub-branding across product categories in different locations. Many studies show that consumers have high willingness-to-pay for particular brands, even if the alternatives are very similar (Dekimpe et al., 1997; Ling, Berndt, and Kyle, 2002). Companies extend their brand to produce similar or different types of products, with the belief that the quality perceptions can be transferred across products under the same brand, as investigated in Wernerfelt (1988) and Montgomery and Wernerfelt (1992), and Erdem (1998). Companies often alter the inputs of products across countries or create differences in quality depending on where the product is produced or the service is provided. Manufacture location begins to summarize the country-specific attributes in the product and become a sub-branding factor influencing consumer preference and companies' profit.

In the case of Mexican Coke, the factor of manufacture location (Mexico) is completely correlated with the product attributes (cane sugar and glass bottles). It is very difficult, if not impossible, to measure the effect of manufacture location independently. More difficulties, which have been naturally added to the research of manufacture location effects, are that we don't often see companies distributing the same product built in different locations in the same market. In early 2017, when General Motors were first criticized, they initially denied but admitted later that "A small number of Mexico-made Chevrolet Cruze sedans were produced in 2016 for sale in the US."[5] General Motors obfuscate the country of manufacture, maybe because they view the vehicles of comparable quality. However, the consumer backlash it created is telling a different story. Unfortunately, consumers' preference for production locations of Chevrolet Cruze remains veiled by now.

There is no doubt that consumers care about the geographic identity of a product at the manufacture location level. The question is to what extent manufacture location influences their preference and affects their purchase choices. Remarkably but no longer surprisingly, consumers have already demonstrated strong preference of manufacture locations in the automobile market in China, which provides us a unique empirical setting to untangle the correlation between man-

[5] The quote is from CNNMoney (Lordstown, OH) first published on January 19, 2017. http://money.cnn.com/2017/01/19/news/economy/donald-trump-chevy-cruze-mexico/

ufacture location and product characteristics. Specifically, we consider the case of Chinese-produced automobiles and evaluate the branding role of the Chinese-denoted manufacturing. We seek to understand the extent to which the Chinese sub-brand of an automobile variant summarizes observable and unobservable differences in the tangible and intangible quality of domestic and foreign produced versions of a vehicle.

Based on the models of hedonic price analysis, we find the "imported" manufacture location effect in terms of the price premium as high as 9.6% with the control of product characteristics. Consumers regard the manufacture location as a strong indicator of utility (tangible and intangible) and are willing to pay 9.6% (implicit price) more. The implicit price differences between domestic and imported model versions shrink as we account for more observable characteristics of the product. However, the implicit price differences stop decreasing at a certain point, suggesting consumers still perceive unobservable differences based on the location of manufacture. We conjecture that the geographic sub-brand summarizes both observable and unobservable characteristics for consumers, which motivates our definition of the brand and sub-brand identity as a representation of consumer observable and unobservable (tangible and intangible) product characteristics.

According to our brand definition and our hierarchical framework of the brand and geographic identity, we structure a BLP-type random coefficient discrete choice model. Our model estimates the distribution of consumer taste parameters of different vehicle models and reveals the positive correlations between model versions that share the same brand identity or geographic identity at the manufacture location level. The results of our model further confirm our conjecture of the strong influence of the geographic sub-branding factor and are consistent with our brand definition and hierarchical framework.

The automobile market example of our empirical study highlights the geographic identifier's ability to differentiate products vertically, but the sub-brand could also horizontally differentiate products, as in the Mexican Coke case. Flavored beverages and other products with taste components clearly benefit from sub-branding that characterizes the taste difference, as the list of ingredients is insufficient to fully describe differences. We would expect that our definition and structural framework and model will also apply to other empirical settings and find the same pattern of the geographic brand describing observable and unobservable aspects of the product variants.

The rest of the paper proceeds as follows. Section 2 examines the data patterns and descriptives that motivate modeling decisions. Section 3 presents hedonic price analysis to study the consumer willingness to pay for the sub-branding element, manufacture locations. Section 4 constructs a BLP-type random coefficient discrete choice model to study the effects of geographic identity on consumer preference further. Section 5 concludes by outlining our key findings and discussing directions for future research.

2

DATA AND DESCRIPTIVES

In the automobile market in China, it is quite normal that vehicles with the same brand and model yet different manufacture locations are sold at the same time. We observe that international automobile companies intentionally apply country-specific labels for their vehicle produced in different places. Figure 2 compares the differences between brand logos of vehicles under the same company but manufactured in different locations. The slightly different brand logo design serves as one simple yet efficient way of separate branding based on country of manufacture. The table is not an exhaustive list, but even in the cases where companies don't market with a different brand logo, typically there are labels on the body of a vehicle indicating whether it is made in China. Nevertheless, consumers are well aware of the existence of the different manufacture locations, and in most cases can easily separate an imported vehicle from one that is domestically produced in China.

In the 1980s, China started to open its market to foreign producers. At the time, domestic automobile production was limited both in technology and capacity, so the number of imported vehicles increased dramatically despite the fact that the tariff was as high as 220%.[1] Meanwhile, Chinese government supported state-owned enterprises like Beijing Automobile Works (BAW), Shanghai Automotive Industry Corporation (SAIC) and First Automobile Works (FAW) to form joint ventures with foreign auto companies like American Motors Corporation (AMC, later acquired by Chrysler Corporation) and Volkswagen to absorb the technology in order to advance the Chinese automobile industry. The early joint ventures primarily operated as complete knock-down (CKD) assembly lines of old model versions of foreign products. In the 1990's, more international automobile companies built partnerships with domestic manufacturers. After China entered World Trade Organization (WTO) in 2001, the tariff for imported automobiles gradually decreased and reached 25% in 2006. In step with the fast growth of Chinese economy, the development of the automobile market accelerated in the first ten years of the 21st century. Facing demanding consumers and intense market competition, joint ventures start to offer new models from their foreign parent firms. However,

[1] http://www.chinadaily.com.cn/dfpd/rs10nian/2011-09/23/content_13778708.htm

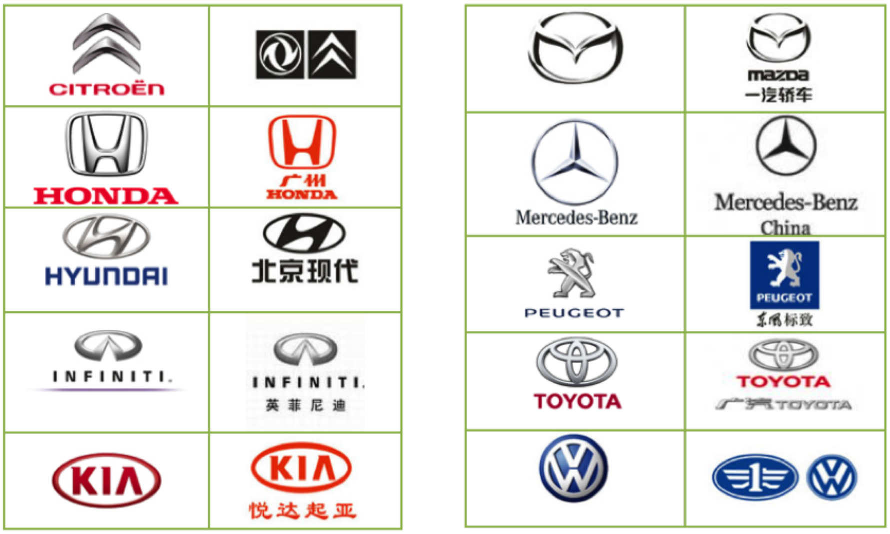

Figure 2: Brand logo comparison for vehicles with different manufacture locations.

unlike the international conglomerates' other subsidiaries, the Chinese joint ventures mainly manufacture vehicles for the Chinese automobile market.

For instance, the BMW Spartanburg Plant was built in South Carolina in the U.S. in 1994. By 2014, Spartanburg Plant had exported to 140 markets around the world. BMW has also built three plants in the city of Shenyang in Northeast China since 2003, but Shenyang plants solely contribute to developing and penetrating the Chinese market. Four series including BMW 5, BMW 3, BMW 2 Tourer and BMW X1 have been produced in Shenyang, whose production and inspection are proclaimed in strict accordance with the unified global standards.[2] Meanwhile, imported BMW 5, BMW 3, and BMW X1 series are observed in our data as well. About 40% of the BMW buyers purchased the imported vehicle, in spite of the much higher price (as shown in Table 1).

2.1 DATA

We use the New Car Buyers Survey (NCBS) data in 2015 provided by Ford-Stanford Alliance. Initially, the survey is conducted to get responses from new car buyers on questions related to purchase motivation, vehicle delivery, driving experience, etc. The survey questions cover a broad range, including new vehicle attributes,

[2] This information is from the official BMW website in China: http://www.bmw-brilliance.cn/cn/en/index.html

Table 1: Consumers of 17 brands with both imported and domestically produced models.

2015	Brand	Imported	Domestic	Total
1	Audi	368	4252	4620
2	BMW	924	1363	2287
3	Buick	19	2165	2184
4	Cadillac	153	341	494
5	Citroen	4	2785	2789
6	Ford	81	3082	3164
7	Hyundai	82	2205	2287
8	Infiniti	143	44	187
9	Kia	150	1821	1974
10	Land Rover	824	117	941
11	Mazda	39	1340	1379
12	Mercedes	550	538	1089
13	Mitsubishi	239	576	815
14	Peugeot	13	2554	2567
15	Toyota	78	3515	3593
16	Volvo	298	558	856
17	VW	532	14763	15295
Total		4497	42019	46516

mileage driven per day, satisfaction with new cars and dealers, price, discount, tax, registration fee, insurance, considered alternatives, previously owned and other vehicles in the household, consumer personality and demographic information, etc.

For our research purpose, after taking out business purchases and respondents failing to provide the price of their new car, we keep 70740 private car owners in the 2015 survey. Among these 70740 consumers, 63283 of them purchased a domestically produced car, and 7457 bought an imported car. Their purchases cover 53 domestically produced brands and 29 imported brands (details are shown in Table 16 in Appendix). Noticeably, many prominent international automobile companies provide both domestically produced and imported vehicles and have gained a massive profit.[3]

[3] In 2015, passenger car sales in China exceeded 20 million, and over 1 million vehicles were imported. By 2016, approximately 163 million privately owned cars were registered in China, and the number is still growing.

The 53 domestically produced brands and 29 imported brands overlap on 17 international companies that provide both domestically produced and imported models in China, as shown in Table 1. For some reason, GM[4] and Nissan[5] only show domestically produced models in the data. It is worth mentioning here that including GM and Nissan, international brands occupy about 70% of the Chinese automobile market in our data, which is consistent with the nationwide statistics over the past decade.

Table 2: 27 Shared imported and domestic models of the 17 brands.

2015	Imported	Freq.	Domestic	Freq.
1	Audi A3 (2012 MY)	49	Audi FAW A3	694
2	Audi A4 Allroad (2009 MY)	10	Audi FAW A4L (2008 MY)	877
3	Audi A6 (2011 MY)	3	Audi FAW A6L (2012 MY)	1155
4	Audi Q3	6	Audi FAW Q3	648
5	Audi Q5	20	Audi FAW Q5	878
6	BMW 3 Series (2012 MY)	3	BMW Brilliance 3 Series (2012 MY)	209
7	BMW 3 Series GT	76	BMW Brilliance 3 Series L (2012 MY)	348
8	BMW 5 Series (2011 MY)	46	BMW Brilliance 5 Series L (2010 MY)	509
9	Citroen C4 Aircross	4	Citroen Dongfeng C4L	472
10	Ford Edge (China, UAE & Saudi Arabia Only)	47	Ford Changan Edge (2015 MY)	80
11	Hyundai Grand Santa Fe	38	Hyundai Beijing Santa Fe (2012 MY)	173
12	Hyundai Santa Fe (2012 MY)	5	Hyundai Hengtong Huatai Santa Fe	25
13	Infiniti Q50	58	Infiniti Dongfeng Q50L (2014 MY)	26
14	Range Rover Evoque	160	Land Rover Chery Range Rover Evoque	117
15	Mazda 5 (2011 MY)	39	Mazda Changan CX-5	271
16	Mercedes C Class W/ S/ C204 (2007 MY)	12	Mercedes Beijing C Class V205 (2014 MY)	146
17	Mercedes E Coupe/ Cabrio C/ A207 (2009 MY)	26	Mercedes Beijing C Class W205 (2014 MY)	3
18	Mercedes G Wagen	3	Mercedes Beijing E Class V212 (2009 MY)	168
19	Mercedes GL X166 (2012 MY)	24	Mercedes Beijing GLA X156	39
20	Mercedes GLA X156	40	Mercedes Beijing GLK X204	151
21	Mitsubishi Outlander (2013 MY)	173	Mitsubishi GAC Pajero	47
22	Mitsubishi Pajero/ Shogun/ Montero (2007 MY)	66	Mitsubishi GAC Pajero Sport (2013 MY)	140
23	Volvo S60 (2010 MY)	12	Volvo S60 (2010 MY)	254
24	Volvo XC60	98	Volvo XC60	271
25	VW Golf (7) (2013 MY)	35	VW FAW Golf (7)	1014
26	VW Passat (2011 MY)/ Passat CC (2012 MY)	44	VW SVW Passat (2011 MY)	1144
27	VW Tiguan	82	VW SVW Tiguan	1335
Total		1179		11194

[4] GM has started producing vehicles in China and exporting to US since 2016. https://nypost.com/2017/07/02/gm-is-producing-popular-car-in-china-exporting-to-us/.

[5] In 2017, Nissan sold more than 1.5 million vehicles in China, including imported, passenger and light commercial vehicles. https://www.automotiveworld.com/news-releases/nissan-show-three-electric-vehicles-auto-china-2018/

All the imported and domestically produced model details for the 17 international automobile companies are listed in Tables 17, 18, and 19 in Appendix. We can easily see that most of the domestically produced models have the same model name as imported models, but with a domestic prefix. For example, domestically produced models use the prefix FAW for Audi, Brilliance for BMW, Changan for Ford, Beijing for Hyundai, etc. These prefixes are symbols for the local plants in China. To further isolate the effect of manufacture locations, we get into specific vehicle attributes. Table 2 shows only the overlapped models within each international brand. Theoretically, the same model domestically produced in China should have exactly the same characteristics as its imported counterpart, as claimed by many local manufacturers and their conglomerates. However, in practice, consumers are willing to pay much higher prices, which indicates that imported models hold higher perceived tangible and intangible quality for consumers.

Among the 17 brands, 12 of them provide the same models for both imported and domestically produced categories, but are they really the same, at least on paper? Do consumers pay extra money only for their beliefs about the manufacture location? Based on the full data set, we build a vehicle dictionary where we can look up all the available car model versions with different attributes and get the corresponding prices. We manually compare the five major attributes: Engine Size, Engine Type, Body Type, Drive Type, Gearbox, of the domestic and imported same models and demonstrate the results in Table 3. Blanks mean that the domestic and imported models match perfectly, i.e., their available choices of that vehicle attribute are the same. A “&" means that the match is not perfect, but choices overlap; a “+" means that the domestic model offers more choices; a “-" means that the imported model provides more choices; a “!" means that the available choices are completely different. Note that for Volvo S60, Engine Size is 1969cc vs. 1984cc, not matched but very close. Hence, we further exclude three models (Citroen C4, Mazda 5, and Mercedes C Class) from the final sample set because consumers probably have to choose their geographic sub-brands with different attributes. Finally, it is more reasonable to assume that consumers in our considered dataset face the same product attributes and availability as later in Assumption 3 of the consumer choice model.

Table 3: Matching status between imported and domestic same models.

2015	Imported	Engine Size	Engine Type	Body Type	Drive Type	Gearbox
1	Audi A3	&				
2	Audi A4	+		!	+	
3	Audi A6	+			+	+
4	Audi Q3	+			+	+
5	Audi Q5		-			+
6	BMW 3 Series	&		-		
7	BMW 5 Series		+	-		
8	Citroen C4	!	+	!	!	+
9	Ford Edge	&	-			
10	Hyundai Santa Fe	&				+
11	Infiniti Q50	-	-			
12	Range Rover Evoque	-	-			
13	Mazda 5	!		!	+	
14	Mercedes C Class	!		!	+	
15	Mercedes E	+		!	+	+
16	Mercedes G	&	-		+	
17	Mitsubishi Outlander/ Pajero	&				+
18	Volvo S60	!				
19	Volvo XC60	&				
20	VW Golf (7)	&	+	-	-	+
21	VW Passat	+	+	!	-	+
22	VW Tiguan	&		+	+	+

[a] Blanks mean that the domestic and imported models match perfectly.
[b] “&" means that the match is not perfect, but choices overlap;
[c] “+" means that the domestic model offers more choices.
[d] “-" means that the imported model provides more choices.
[e] “!" means that the available choices are completely different.

2.2 PRICE PATTERN

For the dataset containing all private purchases, the distribution and statistics of the prices for imported and domestically produced vehicles are shown in Table 4. For the dataset including only common models available in both imported and domestically produced versions, the price range shrinks as shown in Table 5. Both tables illustrate that prices of imported cars are systematically higher. The medians of price distribution of all vehicles, imported cars, and domestic cars are 138K, 400K, and 128.8K yuan, respectively. Hence, consumers will have to budget very

Table 4: Summary of price distributions for imported and domestically produced vehicles containing all private vehicles.

Statistics	Min	1st Quartile	Median	Mean	3rd Quartile	Max
All Vehicles	20000	97000	138000	193400	230000	3000000
Imported	90000	265000	400000	508200	630000	3000000
Domestic	20000	92000	128800	156300	196000	830000

Table 5: Summary of price distributions for imported and domestically produced vehicles that share common models.

Statistics	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Common Models	56800	220000	280000	305200	380000	1700000
Imported	170000	285100	364000	395500	460000	1700000
Domestic	56800	220000	271800	294600	375700	830000

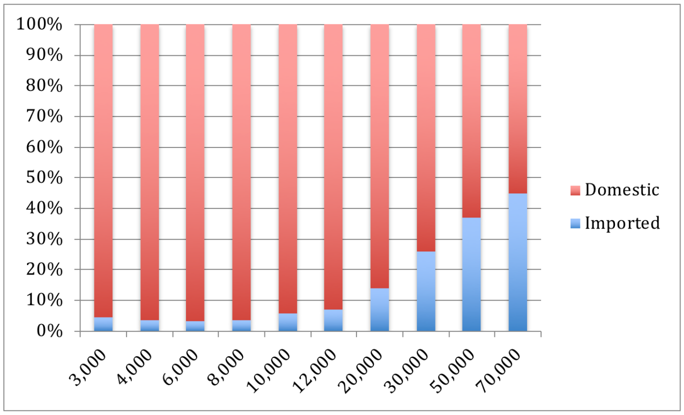

Figure 3: The percentages of domestic and imported vehicle buyers of different income groups.

differently, as high as twice more on average to purchase an imported vehicle in China. Figure 3 shows a monotonic increasing trend of the percentage of imported vehicle buyers, as their incomes increase.[6]

The next section presents hedonic price analysis models to explore the mystery of the imported vehicle prices further and captures the higher consumer willingness to pay induced by the geographic sub-branding element, manufacture location.

[6] The per capita disposable annual income of urban households in China was 31K yuan in 2015. https://www.statista.com/statistics/289186/china-per-capita-disposable-income-urban-households/

3

HEDONIC PRICE ANALYSIS

Although Court (1939) first coined the term, “hedonic price analysis" was not popularized until the early 1960s by Griliches (1961). This method was originally used to quantify different dimensions of quality change of a product, e.g., horsepower, weight, or length for automobiles. Hedonic price analysis uses multivariate regression to derive “implicit prices" per unit of the chosen additional characteristic of the commodity. In the condition of no quality evolution, this procedure could answer the question of what the price of a “new" combination of qualities of a particular product would have been in cases where that particular product was not available, by interpolating or extrapolating the relationship between price and characteristics of available product varieties.

The hedonic hypothesis here is that goods are valued for their utility-bearing attributes or characteristics. Hedonic prices are defined as the implicit prices of product characteristics, which are revealed from observed prices of differentiated products and the specific amounts of characteristics associated with them. The original goal of hedonic price analysis is to find what relationship there exists between the product’s price and its attributes. To estimate the relationship (i.e., function) between prices and product characteristics, we need to make assumptions about the relevance of different characteristics and the ways (i.e., forms of the function) in which they relate to the price.

The semilogarithmic specification of the hedonic function is convenient and becomes popular, as in Eq. (3.1) below. When natural logarithms are used, the β_j coefficient provides an estimate of the percentage increase in price due to a one-unit change of the corresponding characteristic, holding other characteristics constant:

$$\log P_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \varepsilon_i, \tag{3.1}$$

where P_i is the price of the product i, β_0 is the standard regression intercept, β_j’s are the regression coefficients (i.e., implicit prices), x_{ij} is the characteristic j of the product i, and ε_i is the regression error. Under restrictive conditions, the semilogarithmic hedonic function can be derived from an underlying utility function (Rosen, 1974; Diewert, 1961). A microeconomic theoretical interpretation for the hedonic regression can be provided by relating to the demand and supply curves.

3.1 BRAND AND SUB-BRAND HEDONIC PRICE MODELS

We summarize the brand identity as a representation of consumer observable and unobservable (tangible and intangible) product characteristics. Based on this definition of brand identity, brand and sub-brand are utility-bearing and naturally fit into the right-hand side of the hedonic function. Our results in section 3.2 support our understanding of brand identity.

We apply hedonic price analysis to examine the effect of geographic identity on manufacture location level and investigate the interaction between a vehicle's manufacture location and its actual sales prices. Comparing to the original hedonic model, we include not only the physical attributes of a product but also the brand and geographic identities, which convey extra information about aspects of a product that is difficult to quantify, such as reputation, social status representation, and quality perception. These are particularly true in the car market where a brand's origin plays a traditionally important role in purchase decisions, as well as manufacture locations. For example, car quality is often difficult to evaluate prior to purchase, and buyers may have to rely on geographic reputations. To investigate the above hypotheses formally, first, the standard hedonic regression model, Eq. (3.1), is extended to include manufacture location effects. More specifically, we add a binary variable in the hedonic price model:

$$\log P_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \alpha I_i + \varepsilon_i, \tag{3.2}$$

where P_i is the price of the vehicle purchased by consumer i, β_0 is the intercept for the base case vehicle model, β_j's are the regression coefficients (i.e., implicit prices), x_{ij} is characteristic j of vehicle i, I_i is the binary variable representing whether the vehicle is imported or domestically produced in China, and ε_i is the regression error. $I_i = 1$ if a vehicle is imported, otherwise $I_i = 0$. The attribute set J includes all major attributes of a vehicle: Fuel Consumption, Horsepower, Drive Type, Gearbox, Body Type, and Engine Type as control variables.

Furthermore, we include two forms of brand identity, brands and model versions, in the following two models:

$$\log P_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \alpha I_i + \gamma' B_i + \varepsilon_i, \tag{3.3}$$

$$\log P_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \alpha I_i + \theta' M_i + \varepsilon_i, \tag{3.4}$$

where B_i represents the brand name of vehicle i, and M_i represents the model version of vehicle i, which usually contains its brand name. B_i and M_i are vectors with one "1" element indicating which brand or model the vehicle i belongs to, and other elements "0." γ and θ are vectors of the corresponding coefficients regarding different brands and model versions.

The coefficient α in each of the models Eq. (3.2)–(3.4) provides us with the implicit price of the Imported Model attribute, i.e., an estimate of average percentage increase in prices between imported vehicles and ones that are domestically produced in China, holding other vehicle characteristics unchanged.

3.2 HEDONIC MODEL RESULTS

We apply hedonic price analysis as one way to measure the value that manufacture locations contributing to brand equity, in the sense of how much price premium can be achieved when the variations of manufacture locations are taken into consideration, or how the manufacture locations affect consumer willingness to pay. The results of our models on both the whole dataset and the filtered common model dataset have shown that the effect of manufacture location is strongly significant. Precisely, even given the brand model and major vehicle attributes (Fuel Consumption, Horsepower, Drive Type, Gearbox, Body Type, and Engine Type), on average, consumers are still willing to pay 9.6% higher price for switching to the same model but an imported vehicle. These results support our conjecture that manufacture location has become an essential sub-branding factor influencing consumer preference and marketing strategies.

3.2.1 *Results on All Private Purchases*

First, we run the hedonic pricing models on the dataset containing all private vehicles. The results are given in Table 6. As a single independent variable, the binary variable (Imported Model, I) initially contributes 115% of the price increase. As we add a control variable, the brand of a vehicle, the coefficient of manufacture location effect drops to 55.4%, indicating that consumers are willing to pay 55.4% more on average to switch to an imported version of their chosen vehicle brand. When the major attributes of a vehicle (Fuel Consumption, Horsepower, Drive Type, Gearbox, Body Type, and Engine Type) sequentially enter as control variables, the impact of manufacture location on pricing starts to decrease, then bounces back a little, and finally stops at 10.6%, which shows two vehicles with the same core attributes yet different manufacture locations are priced very differently. In the whole market (containing domestic brands in China), the coefficients of manufacture location are significant in the result of all Hedonic Models (1)–(8) in Table 6.

This result confirms the impact of manufacture location broadly. Intuitively, if one product builds its geographic identity in a certain market, the price of similar products of other international and domestic brands can be influenced as well. Generally speaking, the domestic sub-brands of international brands com-

Table 6: Models on the dataset containing All Private Purchases

Hedonic Models	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Imported Model[a]	1.15***	0.554***	0.409***	0.110**	0.105**	0.112**	0.109**	0.106**
Robust SE[b]	0.120	0.087	0.054	0.041	0.038	0.035	0.033	0.039
Robust P-value	< 2e−16	1.93e−10	5.73e−14	0.008	0.005	0.001	0.001	0.007
Brand Name	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Fuel Consumption	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Horsepower	No	No	No	Yes	Yes	Yes	Yes	Yes
4-Wheel Drive	No	No	No	No	Yes	Yes	Yes	Yes
Gearbox	No	No	No	No	No	Yes	Yes	Yes
Body Type	No	No	No	No	No	No	Yes	Yes
Engine Type	No	No	No	No	No	No	No	Yes
Sample Size	70740	70740	70740	70740	70740	70740	70740	70740
R^2	0.294	0.706	0.786	0.903	0.903	0.913	0.917	0.920

[a] Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

[b] Robust standard errors clustered at the brand level are reported.

pete with national brands. More specifically, the new domestic manufacture location sub-branding identity might expand or compete with its original counterpart, which is the focus of the following hedonic models and consumer choice models.

In most of the cases, one vehicle brand has different model versions that are defined by the core attributes of a vehicle along with other unobserved detailed designs. If our definition of brand and sub-brand as a summarization of consumer observed and unobserved (tangible and intangible) attributes of a product is coherent, we should be able to find consistent and more prominent results below on the refined dataset of common models available in both imported and domestically produced versions.

3.2.2 *Brand Name as a Control Variable*

As stated in the Data section, here we only use vehicles with the same model version and different manufacture locations. In Table 7, when we use only "Imported Model" as an independent variable, it reflects 28% price premium, compared to 115% for the whole dataset. Given the brand and core attributes of a vehicle, the manufacture location impact on price premium becomes 8.3%. One reason for the drop in coefficients should be that the refined dataset has a higher base-case price

Table 7: Models on the dataset containing only Common Models available in both imported and domestically produced versions: use Brand Name as a control variable

Hedonic Models	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Imported Model[a]	0.280***	0.208**	0.201***	0.109*	0.104*	0.104*	0.094**	0.083*
Robust SE[b]	0.154	0.071	0.054	0.053	0.042	0.041	0.035	0.037
Robust P-value	0.069	0.004	2e−04	0.039	0.013	0.012	0.008	0.024
Brand Name	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Fuel Consumption	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Horsepower	No	No	No	Yes	Yes	Yes	Yes	Yes
4-Wheel Drive	No	No	No	No	Yes	Yes	Yes	Yes
Gearbox	No	No	No	No	No	Yes	Yes	Yes
Body Type	No	No	No	No	No	No	Yes	Yes
Engine Type	No	No	No	No	No	No	No	Yes
Sample Size	10345	10345	10345	10345	10345	10345	10345	10345
R^2	0.052	0.516	0.628	0.792	0.800	0.800	0.831	0.838

[a] Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

[b] Robust standard errors clustered at the brand level are reported.

because it filtered out the cheaper vehicles of national brands. Even though the ratios are lower than previous models, they remain significant both statistically and economically. Meanwhile, we observe the same pattern as in the whole dataset: the coefficient of manufacture location gradually decreases as product characteristics sequentially enter the analysis. This pattern shows us that the geographic sub-brand is positively correlated with consumer observed tangible attributes, which supports the first half of our definition of brand and sub-brand. Then, how do we demonstrate the second half of our definition? If we zoom in on the car attributes even more, should we naturally expect coefficients to drop even more? Maybe the remarkable impact of manufacture location is purely because we are missing physical characteristics unobservable for econometricians, but known by consumers? The results of the following models confirm our definition of brand identity and the significance of the manufacture location sub-branding effects.

3.2.3 *Model Name as a Control Variable*

With the purpose of including more vehicle attributes that may not be captured by the core characteristics of vehicles, we use Model Name as an independent variable in Table 8, e.g., “Audi A6", “BMW 3 series", “Ford Edge", etc., instead

Table 8: Models on the dataset containing only Common Models available in both imported and domestically produced versions: use Model Name as a control variable

Hedonic Models	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Imported Model[a]	0.280***	0.211***	0.208***	0.152***	0.144***	0.142***	0.104***	0.096***
Robust SE[b]	0.125	0.058	0.056	0.032	0.030	0.031	0.029	0.028
Robust P-value	0.026	2.47e−4	1.23e−4	2.35e−6	1.87e−6	4.00e−6	2.92e−4	7.63e−4
Model Name	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Fuel Consumption	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Horsepower	No	No	No	Yes	Yes	Yes	Yes	Yes
4-Wheel Drive	No	No	No	No	Yes	Yes	Yes	Yes
Gearbox	No	No	No	No	No	Yes	Yes	Yes
Body Type	No	No	No	No	No	No	Yes	Yes
Engine Type	No	No	No	No	No	No	No	Yes
Sample Size	10345	10345	10345	10345	10345	10345	10345	10345
R^2	0.052	0.817	0.825	0.870	0.878	0.878	0.882	0.883

[a] Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1.
[b] Robust standard errors clustered at the model level are reported.

of the general Brand Name. Even though Hedonic Model (2)'s in Table 7 and 8 with two different independent variables, namely Brand Name vs. Model Name, both initially have the coefficients of manufacture location close to 21%, but the R-square value is largely increased by switching from Brand Name to Model Name (0.516 vs. 0.817, R^2 in the second columns of Table 7 and 8). These results unsurprisingly illustrate that Model Name does have stronger explaining power on the price premium than Brand Name. Intuitively, Model Name should be more positively correlated with at least the physical attributes that define the model version of the product.

With all control variables included, we find the coefficient of manufacture location gradually decreases and eventually reaches 9.6%, but did not go below 8.3% of Hedonic Model (8) in Table 7. If the geographic branding factor that we are considering here only represented the observable attributes, the coefficient of manufacture location would have continued decreasing further because Model Name introduces more physical characteristics. Since adding more attributes causes no further reduction in the coefficient, we should scope the representability of product geographic sub-brand identity beyond observable attributes. Part of this geographic sub-brand identity needs to be explained by unobservable attributes, which could be tangible or intangible, and challenging to quantify.

Here we use the vertical distance in price premium or consumer willingness to pay to measure the effect of geographic identity. Manufacture location has become a sub-branding factor of many products. This broadly motives our definition of brand identity as a representation of consumer observable and unobservable (tangible and intangible) product characteristics (as shown in Figure 1). This definition also guides the structure of our consumer choice model and is further confirmed by estimation results.

4

CONSUMER CHOICE MODELING

Hedonic Price Analysis illustrates that consumers have high willingness to pay for “Imported Model," yet at the same time we observe a large amount of "Domestically Produced Model" purchases. The emergence of domestically produced sub-brands seems to provide consumers "cheaper" options, and on the other hand, companies collect more market shares from their sub-branding strategy. In this section, we investigate how geographic identity as a sub-branding element influences consumer purchase choices.

4.1 BLP-TYPE DISCRETE CHOICE MODEL

Based on our framework of Brand and Geographic Identity illustrated in Figure 1, we construct a consumer discrete choice model similar to BLP, one of the major methods for estimating demand of differentiated products (Berry, 1994; Berry, Levinsohn, and Pakes, 1995; Nevo, 2000). Assume we observe $t = 1, \ldots, T$ markets with $i = 1, \ldots, I_t$ consumers choosing from $j = 1, \ldots, J$ different products. The indirect utility of consumer i consuming product j from market t is given by

$$u_{ijt} = \beta_i' B_{jt} + \alpha P_{jt} + \xi_{jt} + \varepsilon_{ijt}, \tag{4.1}$$

where P_{jt} is the price of product j in market t. B_{jt} is a J-dimensional vector representing consumer brand choices, with the jth element equal to “1” and other elements being “0". ξ_{jt} captures the factors that are unobserved by econometricians like systematic shocks to demand. ε_{ijt} is a stochastic term. α represents consumers' marginal utility of wealth. β_i is a J-dimensional vector of individual-specific taste coefficient of brands.

The setup of Eq. (4.1) has integrated various assumptions that result in different implications. BLP classically assumes that producers and consumers observe all product characteristics, but econometricians do not. Because of our definition of brand identity, we have a different version of this fundamental assumption.

Assumption 1. *Consumers make purchase choices according to their perception of observed and unobserved (tangible and intangible attributes), which forms product brand* *identity. In addition, factors beyond brand identity, e.g., directly related here, the preconceptions of country-of-origin of different brands, technology advances, national and international trade policy reformation, etc., also affect consumer purchases.*

This assumption is consistent with our hierarchical structure having brand below country-of-origin but above manufacture location (Figure 1). Since all these factors above the brand identity level are not explicitly quantified, they will be captured by the econometric error term ξ_{jt} in Eq. (4.1). However, producers are exposed to these factors and set pricing strategy accordingly. The econometric problem of endogenous price is still unavoidable, and hence we use a BLP-type model to tackle price endogeneity.

Assumption 2. *The elements captured by ξ_{jt} vertically differentiate products in the same way for all consumers.*

In one domestic market, i.e., the automobile market in China in our estimation, we believe that the vast majority of consumers will share the same perspective of country-of-origin, technology, policy, etc. Even companies' temporary promotional activities will still systematically affect all consumers unless promotions are tailored to target specific individuals. Yet for an automobile purchase, small promotions should have minor effects on prices or consumer decisions.

Assumption 3. *All consumers face the same product attributes, availability, and prices.*

If product prices vary among consumers, averaging prices will lead to measurement error bias. It is another way that prices may correlate with the error term, where the instrumental-variable procedure of BLP shows advantages on price endogeneity. Vehicle characteristics and availability might widely differ. For the research focus in this paper, we therefore select our final samples as described in Chapter 2.

Assumption 4. *The indirect utility in Eq. (4.1) is derived from a quasilinear utility function. Its specification assumes that wealth and prices linearly affect consumer preference.*

For high-cost vehicle purchases, it is not a typical assumption to make. However, for our selected dataset to support our research focus here, this assumption does not seem to drift far from reality. We come back to this point later in the result chapter.

Consumers are assumed to purchase the good that produces the highest utility. Formally, the group of consumers choosing good j is defined by the set

$$A_{jt}(B_{\cdot t}, P_{\cdot t}; \xi_{\cdot t}, \alpha, \beta_i) = \left\{ (\varepsilon_{i1t}, \ldots, \varepsilon_{iJt}, \beta_i) \mid u_{ijt} \geq u_{ilt}, \forall l = 1, \ldots, J \right\}. \tag{4.2}$$

Then we can achieve the market share of product j by the integral over the mass of consumers in region A_{jt}:

$$\begin{aligned} s_{jt}(B_{\cdot t}, P_{\cdot t}; \xi_{\cdot t}, \alpha, \beta_i) &= \int_{A_{jt}} dF(\varepsilon_{ijt}, \beta_i) \\ &= \int_{A_{jt}} dF(\varepsilon_{ijt} \mid \beta_i) dF(\beta_i) \\ &= \int_{A_{jt}} dF(\varepsilon_{ijt}) dF(\beta_i), \end{aligned} \tag{4.3}$$

where $F(\cdot)$ denotes the population distribution. The second and third equality is according to Bayes' rule and the independence assumption of the stochastic error term.

In order to compute the integral in Eq. (4.3), we need to make distributional assumptions on the individual attribute parameters.

Assumption 5. *ε_{ijt}'s are independent and identically distributed according to Type I extreme value distribution.*

Hence, we deduce the market share from Eq. (4.3) to:

$$s_{jt}(B_{\cdot t}, P_{\cdot t}; \xi_{\cdot t}, \alpha, \beta_i) = \int_{A_{jt}} \frac{\exp(\beta_i' B_{jt} + \alpha P_{jt} + \xi_{jt})}{\sum_{k=1}^{J} \exp(\beta_i' B_{kt} + \alpha P_{kt} + \xi_{kt})} dF(\beta_i), \tag{4.4}$$

In addition to the separable additive random shock ε_{ijt}, consumer heterogeneity enters our model through the individual-specific taste coefficient β_i.

Assumption 6. *The random taste coefficient β_i is parametrically distributed under the multivariate normal distribution with mean β and variance-covariance matrix Σ:*

$$\beta_i \overset{iid}{\sim} N(\beta, \Sigma). \tag{4.5}$$

Now instead of ten thousand or more consumer taste coefficients, we only need to estimate the distribution mean vector $\beta \in \mathbb{R}^J$ and variance-covariance matrix $\Sigma \in \mathbb{R}^{J \times J}$. Let $\eta_i \in \mathbb{R}^J$ denote the multivariate standard normal distribution, i.e., $\eta_i \overset{iid}{\sim} N(0, I)$, and Γ denote the upper triangular matrix of the Cholesky decomposition of Σ, i.e., $\Sigma = \Gamma'\Gamma$. Then we can express β_i as

$$\beta_i = \beta + \Gamma' \eta_i. \tag{4.6}$$

We can therefore estimate β and Σ by integrating over η_i.

4.2 ESTIMATION PROCEDURE

Given previous assumptions, we substitute Eq. (4.6) into Eq. (4.1) and have a new expression of utility function:

$$u_{ijt} = (\beta + \Gamma'\eta_i)'B_{jt} + \alpha P_{jt} + \xi_{jt} + \varepsilon_{ijt}. \tag{4.7}$$

Since we have to take the Cholesky decomposition of the symmetric variance-covariance matrix Σ to get Γ, for Σ alone there are $J \times (J+1)/2$ parameters to estimate. For the purpose of reducing the problem dimension to ease estimation and the research focus of this study, we impose a structure on the variance-covariance matrix Σ:

$$\Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_J \end{pmatrix} \begin{pmatrix} 1 & \rho_B & \rho_D & 0 & \rho_D & \dots \\ \rho_B & 1 & 0 & \rho_I & 0 & \\ \rho_D & 0 & 1 & \rho_B & & \\ 0 & \rho_I & \rho_B & 1 & & \\ \rho_D & 0 & & & \ddots & \\ \vdots & & & & & 1 \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_J \end{pmatrix} \equiv SRS, \tag{4.8}$$

where S is a diagonal matrix with the standard deviations $(\sigma_1, \dots, \sigma_J)$ of the distribution of β_i on the diagonal. R is the correlation matrix of β_i with a structure built by the following rules: (1) if two vehicle models share the same brand, their correlation is ρ_B; (2) if two vehicle models are both imported, their correlation is ρ_I; (3) if two vehicle models are both domestically produced, their correlation is ρ_D; (4) otherwise, their correlation equals to 0.

Even though the extreme value distribution of ε_{ijt} in Assumption 5 lets us integrate ε_{ijt}'s analytically, we still have to compute the integral defining the market shares in Eq. (4.4) by simulation. Here we use probably the most common way to approximate the integral:

$$s_{jt}(B_{\cdot t}, P_{\cdot t}; \xi_{\cdot t}, \alpha, \beta_i) = \frac{1}{NS} \sum_{ns=1}^{NS} \frac{\exp((\beta + \Gamma'\eta_{ns})'B_{jt} + \alpha P_{jt} + \xi_{jt})}{\sum_{k=1}^{J} \exp((\beta + \Gamma'\eta_{ns})'B_{kt} + \alpha P_{kt} + \xi_{kt})}, \tag{4.9}$$

where $\eta_{ns}, ns = 1, \dots, NS$, are J dimension i.i.d. draws from the standard normal distribution.

Next, we apply the mathematical program with equilibrium constraints (MPEC) approach to the generalized method of moments (GMM) estimation, and form a nonlinear constraint optimization problem (Su and Judd, 2012; Dubé, Fox, and Su, 2012):

$$\begin{aligned} \min_{\theta} \quad & g(\theta)' W g(\theta) \\ \text{s.t.} \quad & g(\theta) - Z'\xi = 0 \\ & s_{jt}(B_{\cdot t}, P_{\cdot t}; \theta) = S_{jt} \quad \text{for all } j, t, \end{aligned} \tag{4.10}$$

where $s_{jt}(B_{\cdot t}, P_{\cdot t}; \theta)$'s are the market shares estimated by Eq. (4.9), and S_{jt}'s are the observed market shares. $Z = [z_1, \dots, z_M] \in \mathbb{R}^{J \times M}$ is a set of instruments such that the population moments

$$E[\xi(\theta^*)' z_m] = 0, \quad \forall m = 1, \dots, M, \tag{4.11}$$

where θ^* denotes the true values of the parameters. Therefore, we choose an estimate such that the sample analog of the population moments, $\xi(\hat{\theta})' z_m$, is as close to 0 as possible. W is a weight matrix defined by the inverse of the variance-covariance matrix of the moments, ideally a consistent estimate of $E[Z'\xi\xi'Z]^{-1}$. This way we put less weight on the moments that have a higher variance. Finally, we form $\theta = [\alpha; \beta; \sigma_1, \dots, \sigma_J; \rho_B; \rho_D; \rho_I; g; \xi] \in \mathbb{R}^{(2J+4+M+J\times T)}$ as the unknown-parameter vector to estimate.

We solve the nonlinear optimization problem (4.10) in Matlab by calling solvers KNITRO (*KNITRO optimization software*) and SNOPT (Gill, Murray, and Saunders, 2005a) through Tomlab. Two caveats are worth mentioning here.

First, even though we could easily provide the gradient and Hessian of our objective function:

$$\frac{\partial f(\theta)}{\partial \theta} = \begin{pmatrix} 0 \\ 2Wg \\ 0 \end{pmatrix}, \qquad \frac{\partial^2 f(\theta)}{\partial \theta^2} = \begin{pmatrix} 0 & & \\ & 2W & \\ & & 0 \end{pmatrix}, \tag{4.12}$$

where $f(\theta) = g'Wg$ is the objective function in Eq. (4.10), the Jacobian of the nonlinear constraints is no longer analytically achievable because of the Cholesky decomposition of the variance-covariance matrix. Most solvers should be able to use finite differences to approximate the derivatives. Even though it will slow down the estimation, it should not prevent us from reaching the optimal.

The second problem is more challenging for the search algorithms of optimization solvers. The Cholesky decomposition requires the variance-covariance matrix formed by the unknown parameters to be positive definite. There exists no algorithm that can guarantee a positive definite search direction. Instead of MPEC, the first alternative that comes to our mind is the nested fixed point algorithm (NFP) from the BLP paper (Berry, Levinsohn, and Pakes, 1995). However, from the original proof of the contraction mapping in the appendix of BLP, the first property required is that the function be continuously differentiable. Hence, it is not straightforward to prove that the NFP approach would circumvent the difficulty. Therefore, we reformulate optimization problem (4.10) in the following ways:

1. Move the nonlinear constraints to the objective function to gain more freedom to prevent the non-positive definite errors from terminating the algorithm.
2. Add a penalty parameter to drive the additional part to zero and leverage the two parts of the modified objective function.
3. Define a global parameter to keep a record of the updated objective function value with the purpose of trying to make the objective function continuous.
4. Every time we encounter a non-positive definite matrix, we set the objective value back to the previous record.

Eventually, we also move the linear constraints back to the objective function. The final reformulation is

$$\min_{\theta} \quad \xi' Z W Z' \xi + \lambda \| s_{jt}(B_{\cdot t}, P_{\cdot t}; \theta) - S_{jt} \|^2, \tag{4.13}$$

where $\lambda > 0$ is the penalty parameter that we use to tune the optimization problem, and the other variables are the same as in Eq. (4.10). If the estimation becomes more numerically difficult, we expect the augmented Lagrangian algorithm NCL (Ma et al., 2018) to be efficient and reliable.

4.3 IDENTIFICATION

The crucial identification assumption in our algorithm is Eq. (4.11), which requires Z, a set of exogenous instrumental variables. Standard demand-side instruments are the variables that shift cost but are uncorrelated with the demand shock ξ. We use a set of instrumental variables derived by BLP. Excluding price and other potentially endogenous variables, we use the observed product characteristics, the sums of the values of the characteristics of other products provided by the same firm, and the sums of the values of the characteristics of products offered by other firms. To be specific, each model version has its own Horsepower, its same brand yet different manufacture location counterpart's Horsepower, and the sum of Horsepower of the rest of the models, as three different instruments. Similarly, we could create three instrumental variables with each product characteristic.

How well the instrumental variables work really depends on what the econometric error ξ includes. To put it in another way, what we believe goes into the error term decides the type of instrumental variables we need. According to our definition of brand and geographic identity and the structure of our model (especially Assumption 1), it is intuitive that BLP-type instruments are uncorrelated with our structural error ξ, and hence satisfy the moment conditions.

4.4 ESTIMATION RESULTS

Based on our definition of utility in Eq. (4.1) and the simulated market share with Eq. (4.9), the estimates of the model are computed from the reformulated optimization problem Eq. (4.13). The first part of our new objective function, $\xi' Z W Z' \xi$, is the moment condition, which is non-negative by definition. Ideally, we want to make the moment condition close to zero. The second part of the new objective, $\lambda \| s_{jt}(B_{\cdot t}, P_{\cdot t}; \theta) - S_{jt} \|^2$, is a measure of the distance between the estimated market shares and the observed market shares (the square of the ℓ^2-norm), which is also non-negative by definition and desired to be zero. The larger the penalty parameter λ we set, the greater emphasis the optimization solver puts on the second part, to ensure it is really small. Since the two parts share the unknown variable ξ, it is a trade-off the solver has to make. SNOPT (Gill, Murray, and Saunders, 2005a) is able to make the objective value as small as 2.5814, with $\lambda = 10^7$, i.e., the moment condition and the share difference are both satisfied very well. We report the estimation results in Table 9.

In the way described in the previous section, we create three instrumental variables from each product characteristic including Fuel Consumption, Horsepower, Drive Type, Gearbox, etc. However, we did not use tens of instrumental variables. The results in Table 9 are estimated using nine instrumental variables created from Fuel Consumption, Horsepower, and Drive Type. All variables (including vehicle prices and characteristics) are normalized in order to avoid unnecessary numerical difficulties.

Table 9: Estimated parameters and their standard errors of the random coefficient discrete choice model with 20 imported and domestically produced shared vehicle models.

Variables	β	β_{se}	σ	σ_{se}
1 "AudiDomestically Produced Model"	1.3395	0.2293	2.0721	0.9782
2 "AudiImported Model"	1.3638	0.2345	0.2062	0.0983
3 "BMWDomestically Produced Model"	0.7694	0.4608	2.2161	0.4604
4 "BMWImported Model"	1.8081	0.4627	0.1340	0.1585
5 "FordDomestically Produced Model"	-0.8472	0.6194	0.0987	0.3275
6 "FordImported Model"	-0.0641	1.3679	0.2983	0.2284
7 "HyundaiDomestically Produced Model"	0.4898	1.0887	1.2260	0.4557
8 "HyundaiImported Model"	0.5163	1.5416	0.2942	0.3133
9 "InfinitiDomestically Produced Model"	-0.2799	0.8630	0.9742	0.4050
10 "InfinitiImported Model"	-0.1382	0.7083	0.0268	0.4412
11 "Land RoverDomestically Produced Model"	0.0968	1.1398	0.4521	0.1904
12 "Land RoverImported Model"	0.4698	0.9001	0.3327	0.1420
13 "MercedesDomestically Produced Model"	0.7786	0.2790	1.5110	0.6894
14 "MercedesImported Model"	0.7844	0.2593	0.0417	0.4779
15 "MitsubishiDomestically Produced Model"	0.5773	0.6580	1.0779	0.3412
16 "MitsubishiImported Model"	0.7172	0.4496	0.3037	0.1842
17 "VolvoDomestically Produced Model"	-0.2620	0.8448	1.4784	0.4961
18 "VolvoImported Model"	-0.2433	0.8907	0.2781	0.4361
19 "VWDomestically Produced Model"	-0.0012	0.2120	4.9984	0.4974
	α	α_{se}	ρ	ρ_{se}
Price	-0.7465	0.4911		
ρ_D			0.0039	0.1264
ρ_I			0.8097	0.2465
ρ_B			0.4358	0.2290

[a] The base case is VW Imported Model.
[b] Standard errors are calculated by the bootstrap method.

The taste coefficients, β (i.e., the means of the distribution of marginal utilities) and their standard errors are presented in the first two columns. As expected, Audi, BMW, and Mercedes demonstrate higher marginal utilities, which is consistent with their overwhelmingly large market shares despite high prices. Notably, the positive and negative signs of β's generally come in pairs, i.e., two sub-brands sharing similar marginal utilities. That again implies the positive correlation between consumers' taste of sub-brands.

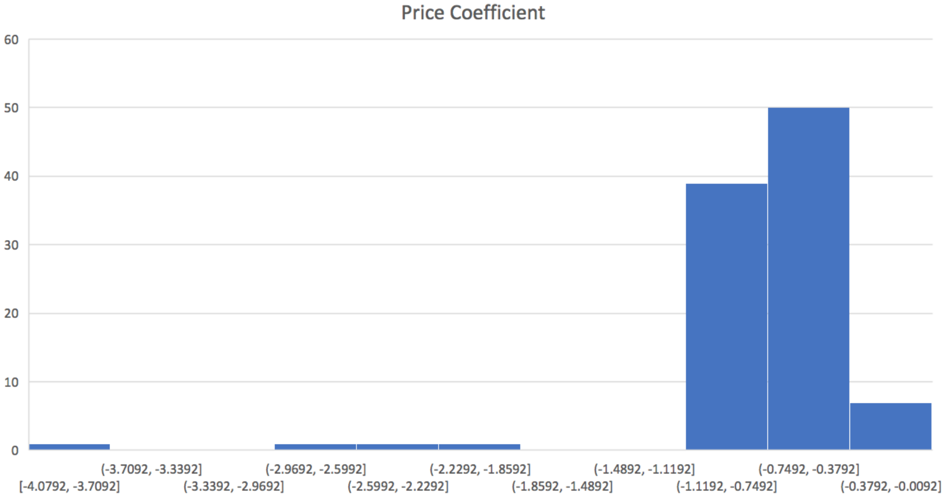

Figure 4: The frequency distribution of price coefficient.

The estimates of standard deviations σ's and their standard errors are shown in the third and fourth columns representing the heterogeneity around the mean β's. The significant standard deviation estimates confirm the existence of substantial consumer heterogeneity.

The price coefficient α is -0.7465. The frequency distribution of price coefficient is shown in Figure 4, illustrating the distribution of the individual price sensitivity. The price coefficient stays negative and does not spread to positive values, probably because the imported extremely luxury cars are excluded from the current dataset because they do not have domestic counterparts.

The estimates of the correlations between consumer taste coefficients and their standard errors are listed in the bottom part of Table 9. Recalling the structure of our correlation matrix R in Eq. (4.8), the results give us: (1) if two vehicle models share the same brand, their correlation is $\rho_B = 0.4358$; (2) if two vehicle models are both imported, their correlation is $\rho_I = 0.8097$; (3) if two vehicle models are both domestically produced, their correlation is $\rho_D = 0.0039$. The mystery of their relatively big standard errors may be caused by our somewhat strict correlation matrix structure to accommodate computational tractability. The estimates are consistent with the data observation and our intuitive understanding of consumer behavior. The taste coefficients for two models are positively correlated if they are both imported or share the same international brand. It is basically saying if consumers prefer one model that is imported, they tend to perceive a higher utility of other cars that are also imported. But the connection is weaker among consumers' preference of domestic models. Also intuitively, if consumers like a car under one automobile brand, they will more likely favor the model under the same brand with a different manufacture location as well.

The estimates of own and cross price elasticities are recorded in Table 10. The element at row i and column j presents the percentage change in market share of model i with a one-percent difference in price of model j. The row and column numbers share the same correspondence with vehicle models in Table 9. The standard errors and significance level of the own and cross price elasticities are available in Table 11 and 12.

The own-elasticities are all negative and tend to be larger than the cross-elasticities in absolute value. In general, the absolute values of all the elasticities are relatively small. It is worth pointing out that a 0.1% change in market share is a significant shift in capital if a market is large. On the other hand, consumers' price sensitivity may be low when it comes to automobile purchases. Specifically, in the dataset consumers spend 300K RMB on average to purchase a vehicle. A 1% price increase (3K RMB) will not easily swing their purchase decisions. It is resonant with the consumer behavior study conducted by two brilliant researchers, Amos Tversky and Daniel Kahneman, referred to in Dan Ariely's book "Predictably Irrational" (Ariely, 2008). A majority of customers would go to another store 15 minutes away to save \$7 of a \$25 pen, but would not take the trip for a \$455 suit. Nevertheless, they are the same pen and suit, and the same \$7 and 15-minute trip.

However, if consumers are open to other choices, the geographic sub-branding alternatives of their originally chosen brands are highly likely to be their second-best options. For instance, if Infiniti Imported Model increases its price by one percent (column 10 of Table 10), its market share naturally decreases by 0.7002% and the market share of Infiniti Domestically Produced Model increases by the largest percentage, 0.0107%. When one of the two Land Rover sub-brands (column 11 and 12) increases its price, the other one proportionally gains the most. We see the similar substitution pattern in almost all vehicle models. This strong substitution pattern demonstrates the positive correlation between geographic sub-brands. Hence, geographic sub-brands not only provide more varieties to consumers and harvest more market shares, but also pick up the swing consumers, which, in a sense, lower the price elasticity of the whole brand.

Because of the positive and relatively big ρ_I, it is also easy to discover another substitution pattern in the elasticity table. For example, when Audi Imported Model increases its price (column 2 in Table 10), all the other imported models enjoy larger percent of market share increase than their domestic counterparts. We see this pattern for each imported model (even number) column of the elas-

Table 10: Own and Cross Price Elasticities.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	-0.2291	0.0044	0.0485	0.0096	0.0035	0.0022	0.0076	0.0021	0.0010	0.0031	0.0079	0.0124	0.0095	0.0075	0.0085	0.0098	0.0256	0.0067	0.0406
2	0.1856	-0.6920	0.1066	0.0295	0.0111	0.0074	0.0156	0.0066	0.0036	0.0093	0.0227	0.0387	0.0300	0.0228	0.0258	0.0310	0.0567	0.0209	0.0438
3	0.1283	0.0066	-0.4241	0.0171	0.0063	0.0039	0.0081	0.0034	0.0023	0.0053	0.0116	0.0191	0.0181	0.0129	0.0109	0.0159	0.0230	0.0110	0.0318
4	0.1726	0.0125	0.1164	-1.0055	0.0115	0.0076	0.0157	0.0067	0.0040	0.0097	0.0228	0.0388	0.0312	0.0236	0.0251	0.0313	0.0572	0.0213	0.0437
5	0.1685	0.0125	0.1148	0.0306	-0.5963	0.0077	0.0158	0.0067	0.0041	0.0099	0.0226	0.0380	0.0325	0.0240	0.0247	0.0310	0.0589	0.0212	0.0449
6	0.1652	0.0130	0.1082	0.0311	0.0119	-0.6563	0.0168	0.0070	0.0040	0.0099	0.0239	0.0409	0.0320	0.0241	0.0268	0.0328	0.0578	0.0220	0.0445
7	0.1966	0.0095	0.0794	0.0224	0.0085	0.0058	-0.4157	0.0056	0.0026	0.0070	0.0171	0.0294	0.0339	0.0171	0.0180	0.0233	0.0480	0.0157	0.0683
8	0.1741	0.0126	0.1062	0.0302	0.0113	0.0077	0.0177	-0.6559	0.0038	0.0095	0.0230	0.0398	0.0314	0.0233	0.0266	0.0320	0.0558	0.0214	0.0451
9	0.1467	0.0125	0.1308	0.0323	0.0126	0.0079	0.0148	0.0068	-0.5981	0.0107	0.0217	0.0381	0.0299	0.0254	0.0204	0.0308	0.0584	0.0225	0.0485
10	0.1733	0.0125	0.1137	0.0303	0.0116	0.0075	0.0154	0.0066	0.0041	-0.7002	0.0227	0.0381	0.0312	0.0238	0.0250	0.0310	0.0587	0.0212	0.0443
11	0.1803	0.0122	0.0999	0.0289	0.0107	0.0073	0.0151	0.0064	0.0034	0.0091	-0.8453	0.0419	0.0290	0.0223	0.0251	0.0312	0.0627	0.0209	0.0462
12	0.1741	0.0128	0.1015	0.0302	0.0111	0.0077	0.0160	0.0069	0.0036	0.0095	0.0258	-1.0022	0.0303	0.0231	0.0271	0.0329	0.0585	0.0219	0.0445
13	0.1508	0.0112	0.1084	0.0274	0.0107	0.0068	0.0208	0.0061	0.0032	0.0087	0.0201	0.0342	-0.6481	0.0218	0.0189	0.0278	0.0434	0.0192	0.0312
14	0.1737	0.0125	0.1133	0.0303	0.0116	0.0075	0.0154	0.0066	0.0040	0.0097	0.0227	0.0382	0.0320	-1.4772	0.0251	0.0310	0.0585	0.0212	0.0440
15	0.1779	0.0127	0.0862	0.0291	0.0108	0.0076	0.0146	0.0068	0.0029	0.0092	0.0230	0.0403	0.0251	0.0227	-0.5367	0.0359	0.0394	0.0200	0.0476
16	0.1721	0.0129	0.1058	0.0305	0.0114	0.0078	0.0159	0.0069	0.0037	0.0096	0.0241	0.0412	0.0309	0.0236	0.0302	-0.5139	0.0551	0.0217	0.0436
17	0.1645	0.0086	0.0557	0.0203	0.0079	0.0050	0.0119	0.0044	0.0025	0.0066	0.0176	0.0267	0.0176	0.0162	0.0121	0.0201	-0.4967	0.0162	0.0570
18	0.1718	0.0126	0.1063	0.0302	0.0113	0.0076	0.0156	0.0067	0.0039	0.0096	0.0235	0.0400	0.0310	0.0233	0.0244	0.0316	0.0645	-0.8034	0.0447
19	0.0633	0.0016	0.0187	0.0038	0.0015	0.0009	0.0041	0.0009	0.0005	0.0012	0.0031	0.0049	0.0031	0.0030	0.0035	0.0038	0.0138	0.0027	-0.0823
20	0.1740	0.0124	0.1142	0.0303	0.0116	0.0075	0.0153	0.0066	0.0041	0.0097	0.0226	0.0378	0.0312	0.0237	0.0248	0.0308	0.0587	0.0210	0.0444

[a] The element at row *i* and column *j* presents the percentage change in market share of model *i* with a 1 % change in price of model *j*.

[b] The row and column numbers correspond to the vehicle models in Table 9.

ticity table. Since ρ_D is small, the similar substitution pattern among domestically produced model is not as clear as the imported models. However, constructing and estimating a correlation matrix helps us structurally reveal and understand the substitution pattern in the market.

Now we use regression of β, σ, and ξ to check and confirm the consistency of our assumptions. We define brand identity as consumers' perception of observed and unobserved product attributes (Assumption 1), which is measured by individual-specific taste coefficients β_i with mean β and standard deviation σ. Intuitively, product characteristics contribute to the mean taste coefficient β, and consumer idiosyncratic factors like demographic information contribute to the variation σ.

The regression results in Table 13 show that almost all the product characteristics significantly correlate with β, i.e., product attributes affect the average of consumers' taste. Regression of the standard deviation of taste coefficient, σ, on consumer demographic information in Table 14 shows that consumers' preference varies less as their ages increase. Different locations and household incomes only affect consumers' taste to some extent. Typically, most of the consumer heterogeneity arguments are based on different incomes, especially about price sensitivity. We do see that more consumers purchase higher priced imported cars as in Figure 3. However, on average, their annual household income is much less than the vehicle price. Hence, we think that the consumer taste heterogeneity is more related to their observed and perceived vehicle attributes than their incomes.[1]

Assumption 1 also says that factors beyond brand identity also affect consumer purchases, which are captured in ξ, the econometric error term of our model. Country of origin related to country reputation is one example of the factors beyond the control of one single brand as illustrated in our hierarchical framework in Figure 1. Therefore, we regress the econometric error ξ on country of origin in Table 15. If we rank the country reputations in Chinese automobile market by comparing their coefficients, we have Germany, Sweden, UK, US, Japan, Korea. This coefficient ranking is consistent with reality. The "counter-intuitive" flipped order of US and Japanese cars could be the effect of vehicle-model subsets of brands and the result of history.[2] Overall, the universally significant coefficient estimates confirm our assumption and hierarchical framework.

[1] It is one reason that the BLP log(income-price) form is not used in our utility function and that our price coefficient is homogeneous among consumers.

[2] According to a study led by Bernstein Research, anti-Japanese sentiment remains an impediment to Japanese car brands in the world's largest market. Half of Chinese consumers say they wouldn't buy a Japanese car. https://blogs.wsj.com/chinarealtime/2014/05/20/half-of-chinese-consumers-say-they-wouldnt-buy-a-japanese-car/

5

CONCLUSION

Manufacture Location, as part of product geographic identity, is becoming a significant differential factor among a variety of products and a sub-branding element in various markets. The automobile market in China makes it possible to isolate the manufacture location effect. Hedonic price analysis shows that manufacture location significantly influences prices and consumer willingness to pay. The “imported" manufacture location effect regarding the price premium could be as high as 9.6% with the control of product characteristics. We define the brand and sub-brand identity as a representation of consumer observable and unobservable (tangible and intangible) product attributes and structure brand identity and geographic identity in a hierarchical framework. Accordingly, we build a random coefficient discrete choice model to approximate the distribution of consumers’ taste. Using the estimates of the consumer taste coefficient, we reveal the underlying substitution patterns among brands and geographic sub-brands and understand the consumer and market behavior in the automobile market in China. Our method can also be helpful for the analysis of branding and sub-branding in other empirical settings and for modeling markets showing strong correlations among brands and products.

It might seem puzzling why geographic identity becomes a relevant branding element when there are many other methods to brand and sub-brand readily available to companies. However, over the past century, the French appellation system has been classifying and labeling French wine by geographic areas.[1] The labels of French wine usually deliver the information of vineyard and even chateau of the wine’s origin, especially for high-quality French wines.[2] Now, the United States, along with many other countries, has adopted a designation or appellation system (American Viticultural Area) similar to the appellation d’origine contrôlée of France. We are not exaggerating to say that in the wine market, geographic identity runs the show.

[1] The wine authorities in Champagne of France have filed lawsuits against other regions or countries naming their sparkling white wine “Champagne".

[2] In addition, the location where the wine was bottled can also be found on the label, as another indicator of quality.

Companies often originate domestically, and therefore by the time they consider international expansions, their COO part of geographic identity is well formed. Zhang (2015) studied an interesting situation where a domestic brand pretends to be a foreign one by picking a foreign sounding name. Our results have clearly shown that COO is significantly correlated with consumer preference of different brands. As little as brands have control of their countries of origin, we do not have much knowledge of the economic magnitude of the interactions between COO and brand identity. On the contrary, as a result of global distribution of production resources, existing companies have more control on geographic identity at a different level, county of manufacture (COM) or manufacture locations, where geographic identity serves hierarchically below a traditional brand name as a sub-branding element, as illustrated in our hierarchical framework.

Driven by cost saving among other incentives like broadening the customer base and sales, companies extend their branches or plants to developing countries like China, India, Mexico, etc. Hence, the brand identity of a firm or a product has been interacting more and more heavily with geographic identity on the level of manufacture location. The impact of manufacture location steers companies' tactics on global production and distribution, which include the choices of new plant locations, new product differentiation, geographic branding, and pricing strategy. As much as companies want to minimize their cost, it is not simple to measure how the change of their geographic identity will impact their brand identity. It is possible that while companies expand their market share, their product loses its pricing privilege, even though it does not seem to be the case in our data. It is also possible that General Motors will gain more profit if they brand and price the Mexican made Chevrolet Cruze differently. Maybe General Motors should embrace Mexican made Chevrolet Cruze as a geographic sub-brand. The connection between brand identity and geographic identity is essential to understand how global allocation and distribution affect consumer preference and geographically change market shares.

6 APPENDIX

6.1 STANDARD ERRORS OF OWN AND CROSS PRICE ELASTICITIES

Table 11: Standard Errors of Own and Cross Price Elasticities.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0.1576	0.0063	0.0180	0.0038	0.0020	0.0021	0.0025	0.0022	0.0015	0.0021	0.0031	0.0047	0.0037	0.0031	0.0032	0.0038	0.0088	0.0026	0.0544
2	0.0871	0.4430	0.0562	0.0115	0.0056	0.0088	0.0062	0.0090	0.0034	0.0115	0.0078	0.0106	0.0113	0.0075	0.0086	0.0124	0.0236	0.0083	0.0318
3	0.0866	0.0120	0.5178	0.0116	0.0056	0.0083	0.0053	0.0077	0.0033	0.0085	0.0055	0.0072	0.0101	0.0057	0.0062	0.0096	0.0209	0.0075	0.0477
4	0.0771	0.0163	0.0694	0.6603	0.0062	0.0098	0.0066	0.0099	0.0036	0.0120	0.0076	0.0105	0.0123	0.0077	0.0082	0.0122	0.0229	0.0086	0.0332
5	0.0846	0.0136	0.0646	0.0120	0.3887	0.0111	0.0066	0.0089	0.0038	0.0089	0.0077	0.0104	0.0140	0.0077	0.0085	0.0116	0.0262	0.0092	0.0441
6	0.0693	0.0153	0.0622	0.0122	0.0080	0.4171	0.0069	0.0106	0.0039	0.0130	0.0084	0.0114	0.0127	0.0078	0.0090	0.0126	0.0256	0.0104	0.0335
7	0.0748	0.0157	0.0610	0.0114	0.0063	0.0097	0.2979	0.0098	0.0037	0.0112	0.0067	0.0088	0.0112	0.0079	0.0077	0.0115	0.0198	0.0083	0.0378
8	0.0712	0.0156	0.0586	0.0119	0.0062	0.0104	0.0083	0.4071	0.0036	0.0144	0.0073	0.0104	0.0124	0.0093	0.0084	0.0127	0.0230	0.0101	0.0356
9	0.0906	0.0130	0.0582	0.0116	0.0060	0.0087	0.0057	0.0080	0.3921	0.0141	0.0071	0.0103	0.0119	0.0075	0.0076	0.0111	0.0265	0.0091	0.0468
10	0.0616	0.0172	0.0624	0.0125	0.0058	0.0113	0.0064	0.0127	0.0050	0.4352	0.0086	0.0119	0.0120	0.0084	0.0082	0.0129	0.0263	0.0122	0.0250
11	0.0941	0.0136	0.0553	0.0109	0.0057	0.0084	0.0058	0.0074	0.0034	0.0088	0.5596	0.0127	0.0107	0.0070	0.0085	0.0117	0.0261	0.0088	0.0479
12	0.0780	0.0165	0.0553	0.0114	0.0057	0.0096	0.0062	0.0099	0.0036	0.0117	0.0099	0.6695	0.0114	0.0076	0.0089	0.0127	0.0247	0.0092	0.0371
13	0.0820	0.0142	0.0660	0.0125	0.0070	0.0095	0.0069	0.0088	0.0037	0.0095	0.0074	0.0104	0.5129	0.0299	0.0080	0.0125	0.0263	0.0092	0.0430
14	0.0758	0.0161	0.0618	0.0130	0.0065	0.0098	0.0068	0.0087	0.0038	0.0103	0.0077	0.0108	0.0319	0.9578	0.0090	0.0131	0.0249	0.0090	0.0334
15	0.0865	0.0145	0.0582	0.0115	0.0060	0.0090	0.0063	0.0087	0.0036	0.0093	0.0083	0.0115	0.0110	0.0088	0.3809	0.0143	0.0217	0.0084	0.0430
16	0.0780	0.0166	0.0587	0.0119	0.0060	0.0095	0.0064	0.0106	0.0037	0.0118	0.0084	0.0114	0.0122	0.0079	0.0107	0.3443	0.0231	0.0088	0.0343
17	0.0785	0.0119	0.0570	0.0096	0.0056	0.0088	0.0051	0.0082	0.0034	0.0106	0.0074	0.0090	0.0107	0.0066	0.0066	0.0108	0.4429	0.0108	0.0487
18	0.0704	0.0147	0.0578	0.0112	0.0064	0.0116	0.0059	0.0122	0.0038	0.0154	0.0083	0.0110	0.0121	0.0078	0.0080	0.0118	0.0352	0.5191	0.0347
19	0.0963	0.0036	0.0129	0.0022	0.0016	0.0018	0.0014	0.0018	0.0013	0.0015	0.0015	0.0016	0.0021	0.0009	0.0016	0.0019	0.0076	0.0016	0.0986
20	0.0769	0.0162	0.0623	0.0118	0.0062	0.0099	0.0063	0.0096	0.0037	0.0124	0.0077	0.0104	0.0120	0.0075	0.0085	0.0122	0.0254	0.0090	0.0342

[a] Standard errors are calculated by the bootstrap method.
[b] The row and column numbers correspond to the vehicle models in Table 9.
[c] The element at row *i* and column *j* presents the standard error of
the percentage change in market share of model *i* with a 1 % change in price of model *j*.

Table 12: Significance of Own and Cross Price Elasticities.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	*	.	**	**	*	*	***	.	.	*	**	**	**	**	**	**	**	**	.
2	**	*	*	**	*	.	**	.	*	.	**	***	**	***	***	**	**	**	*
3	*	.	.	*	*	.	*	.	.	.	**	**	*	**	*	*	*	*	.
4	**	.	*	*	*	.	**	.	*	.	***	***	**	***	***	**	**	**	*
5	*	.	*	**	*	.	**	.	*	*	**	***	**	***	**	**	**	**	*
6	**	.	*	**	*	*	**	.	*	.	**	***	**	***	**	**	**	**	*
7	**	.	*	*	*	.	*	.	.	.	**	***	***	**	**	**	**	*	*
8	**	.	*	**	*	.	**	*	*	.	***	***	**	**	***	**	**	**	*
9	*	.	**	**	**	.	**	.	*	.	***	***	**	***	**	**	**	**	*
10	**	.	*	**	**	.	**	.	.	*	**	***	**	**	***	**	**	*	*
11	*	.	*	**	*	.	**	.	.	*	*	***	**	***	**	**	**	**	.
12	**	.	*	**	*	.	**	.	*	.	**	*	**	***	***	**	**	**	*
13	*	.	*	**	*	.	***	.	.	.	**	***	*	.	**	**	*	**	.
14	**	.	*	**	*	.	**	.	*	.	**	***	*	*	**	**	**	**	*
15	**	.	*	**	*	.	**	.	.	.	**	***	**	**	*	**	*	**	*
16	**	.	*	**	*	.	**	.	*	.	**	***	**	**	**	*	**	**	*
17	**	.	.	**	*	.	**	.	.	.	**	**	*	**	*	*	*	*	*
18	**	.	*	**	*	.	**	.	*	.	**	***	**	***	***	**	*	*	*
19	.	.	*	*	.	.	**	.	.	.	**	***	*	***	**	*	*	*	.
20	**	.	*	**	*	.	**	.	*	.	**	***	**	***	**	**	**	**	*

[a] Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1

[b] The row and column numbers correspond to the vehicle models in Table 9.

[c] The element at row *i* and column *j* presents the significance level of the percentage change in market share of model *i* with a 1 % change in price of model *j*.

6.2 tests of assumptions

Table 13: Regression of the mean of taste coefficient β on product attributes

Coefficients	Estimate	Std. Error	t value	$Pr(>\|t\|)$	
(Intercept)	0.307694	0.134988	2.279	0.022663	*
fc2	-0.316311	0.034053	-9.289	<2e-16	***
horsepower2	0.088808	0.034882	2.546	0.010914	*
drivetype2Yes	0.351422	0.018303	19.200	< 2e-16	***
gearbox2Manual	-0.103948	0.032208	-3.227	0.001253	**
bodytype2Hatchback	0.007384	0.066386	0.111	0.911435	
bodytype2Notchback	0.494258	0.063958	7.728	1.2e-14	***
bodytype2Off Road/ SUV	0.172974	0.064771	2.671	0.007584	**
bodytype2People Carrier	1.583757	0.093054	17.020	< 2e-16	***
enginetype2Petrol (With Turbo)	0.101129	0.112725	0.897	0.369673	
enginetype2Petrol (Without Turbo)	0.267754	0.113667	2.356	0.018511	*
enginetype2Petrol & Electric	-0.821377	0.604721	-1.358	0.174407	
enginetype2Petrol & Electric (With turbo)	0.745610	0.218765	3.408	0.000656	***
enginetype2Petrol & Electric Plug-In (PI) with Turbo	0.185589	0.317621	0.584	0.559024	

[a] Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

[b] Residual standard error: 0.5935 on 10331 degrees of freedom
Multiple R-squared: 0.1292, Adjusted R-squared: 0.1281
F-statistic: 117.9 on 13 and 10331 DF, p-value: < 2.2e-16

Table 14: Regression of the standard deviation of taste coefficient σ on consumer demographic information

Coefficients	Estimate	Std. Error	t value	$Pr(>\|t\|)$	
(Intercept)	2.991695	0.228182	13.111	< 2e-16	***
Age	-0.006596	0.001873	-3.521	0.000432	***
SexMale	-0.057078	0.038493	-1.483	0.138156	
TierTier 2[c]	0.171354	0.039338	4.356	1.34e-05	***
TierTier 3	0.450039	0.061233	7.350	2.14e-13	***
TierTier 4	0.447555	0.083907	5.334	9.81e-08	***
TierTier 5	0.323661	0.225885	1.433	0.151929	
HouseholdIncomeRMB 10,000 - RMB 11,999	0.181698	0.216631	0.839	0.401633	
HouseholdIncomeRMB 12,000 - RMB 19,999	0.025216	0.216610	0.116	0.907329	
HouseholdIncomeRMB 20,000 - RMB 29,999	-0.293868	0.216216	-1.359	0.174131	
HouseholdIncomeRMB 30,000 - RMB 49,999	-0.746461	0.218937	-3.409	0.000653	***
HouseholdIncomeRMB 4,000 - RMB 5,999	0.408235	0.224508	1.818	0.069039	.
HouseholdIncomeRMB 50,000 - RMB 69,999	-0.888078	0.227313	-3.907	9.41e-05	***
HouseholdIncomeRMB 6,000 - RMB 7,999	0.430833	0.223655	1.926	0.054091	.
HouseholdIncomeRMB 70,000 or more	-0.993021	0.225681	-4.400	1.09e-05	***
HouseholdIncomeRMB 8,000 - RMB 9,999	0.319569	0.220329	1.450	0.146973	
HouseholdIncomeUnder RMB 3,999	0.393397	0.240806	1.634	0.102359	

[a] Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

[b] Residual standard error: 1.638 on 10328 degrees of freedom
Multiple R-squared: 0.07586, Adjusted R-squared: 0.07442
F-statistic: 52.98 on 16 and 10328 DF, p-value: < 2.2e-16

[c] Tier systems are used to classify Chinese cities basically based on their GDP.

Table 15: Regression of the econometric structure error ξ on country of origin

Coefficients	Estimate	Std. Error	t value	$Pr(>\|t\|)$	
(Intercept)	1.606632	0.007829	205.219	< 2e-16	***
cooJapan	-1.719904	0.033006	-52.108	< 2e-16	***
cooKorea	-2.149643	0.047297	-45.450	< 2e-16	***
cooSweden	-0.081429	0.029783	-2.734	0.00627	**
cooUK	-1.156919	0.044207	-26.171	< 2e-16	***
cooUS	-1.344947	0.064730	-20.778	< 2e-16	***

[a] Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

[b] Residual standard error: 0.7241 on 10339 degrees of freedom
Multiple R-squared: 0.3457, Adjusted R-squared: 0.3453
F-statistic: 1092 on 5 and 10339 DF, p-value: < 2.2e-16

[c] The base case is Germany.

6.3 additional data information

Table 16: Numbers of vehicles purchased for different brands.

2015	Domestically Produced	Freq.	Imported	Freq.
1	Audi	4252	Audi	368
2	BAIC	156	BMW	924
3	BAIC BAW	7	Buick	19
4	Baoding Great Wall	578	Cadillac	153
5	BMW	1363	Chrysler USA	46
6	Brilliance	262	Citroen	4
7	Buick	2165	Dodge	183
8	BYD	961	Ford	81
9	Cadillac	341	Hyundai	82
10	Changan Auto	1180	Infiniti	143
11	Chery	561	Jaguar	263
12	Citroen	2785	Jeep	693
13	Dongfeng	507	Kia	150
14	Dongfeng Yulong	158	Land Rover	824
15	Everus (Linian)	32	Lexus	451
16	FAW Besturn	449	Lincoln	73
17	FAW Dafa	45	Maserati	39
18	FAW Haima	324	Mazda	39
19	FAW Jilin	16	Mercedes	550
20	FAW Tianjin	104	MINI (From 2002 MY)	212
21	Fiat	526	Mitsubishi	239
22	Ford	3082	Peugeot	13
23	GAC	176	Porsche	317
24	Geely	2906	Renault	331
25	GM Daewoo/ GM Chevrolet	2080	smart	55
26	Haval	916	Subaru	297
27	Honda	2747	Toyota	78
28	Huatai	21	Volvo	298
29	Hyundai	2205	VW	532
30	Infiniti	44		
31	Jianghuai Auto	197		
32	Jiangling (JMC)	120		
33	Kia	1821		
34	Land Rover	117		
35	Lifan	93		
36	Maxus	8		
37	Mazda	1340		
38	Mercedes	538		
39	MG China	110		
40	Mitsubishi	576		
41	Nissan	2249		
42	Peugeot	2554		
43	Qoros	6		
44	Roewe	307		
45	SG Automotive	12		
46	Skoda	1947		
47	Suzuki	525		
48	Toyota	3515		
49	Volvo	558		
50	VW	14763		
51	Wuling	908		
52	Youngman	50		
53	Zotye	20		
Total		63283		7457

Table 17: Comparison between the imported and domestic models of the 17 international brands.

2015	Imported	Freq.	Domestic	Freq.
1	Audi A1	31	Audi FAW A3	694
2	Audi A3 (2012 MY)	49	Audi FAW A4L (2008 MY)	877
3	Audi A4 Allroad (2009 MY)	10	Audi FAW A6L (2012 MY)	1155
4	Audi A5	49	Audi FAW Q3	648
5	Audi A6 (2011 MY)	3	Audi FAW Q5	878
6	Audi A7	19		
7	Audi A8 (2010 MY)	64		
8	Audi Q3	6		
9	Audi Q5	20		
10	Audi Q7 (Pre 2015 MY)	112		
11	Audi TT (2007 MY)	5		
12	BMW 1 Series (2012 MY)	108	BMW Brilliance 3 Series (2012 MY)	209
13	BMW 2 Series	27	BMW Brilliance 3 Series L (2012 MY)	348
14	BMW 2 Series Active Tourer (2015 MY)	41	BMW Brilliance 5 Series L (2010 MY)	509
15	BMW 2 Series Gran Tourer (2015 MY)	4	BMW Brilliance X1 (2009 MY)	297
16	BMW 3 Series (2012 MY)	3		
17	BMW 3 Series GT	76		
18	BMW 4 Series	69		
19	BMW 5 Series (2011 MY)	46		
20	BMW 5 Series GT (2010 MY)	41		
21	BMW 6 Series (2011 MY)	9		
22	BMW 7 Series (2009 MY)	91		
23	BMW X3 (2011 MY)	136		
24	BMW X4	78		
25	BMW X5 (2007 MY)	10		
26	BMW X5 (2014 MY)	136		
27	BMW X6 (2015 MY)	22		
28	BMW X6 (Pre 2015 MY)	17		
29	BMW Z4 (2009 MY)	10		
30	Buick Enclave	19	Buick SGM Encore	267
31			Buick SGM Envision (2014 MY)	216
32			Buick SGM Excelle GT (2010 MY)	223
33			Buick SGM Excelle GT (2015 MY)	231
34			Buick SGM Excelle GT (Pre 2010 MY)	344
35			Buick SGM Excelle XT (2010 MY)	119
36			Buick SGM Firstland GL8 (2005 MY)	77
37			Buick SGM GL8 (2011 MY)	91
38			Buick SGM Lacrosse	278
39			Buick SGM Regal	306
40			Buick SGM Verano	13
41	Cadillac CTS (2008 MY)	13	Cadillac SGM ATS (2012 MY)	168
42	Cadillac SRX (2010 MY)	140	Cadillac SGM XTS	173
43	Citroen C4 Aircross	4	Citroen Changan DS5	105
44			Citroen Changan DS5 LS (2014 MY)	210
45			Citroen Changan DS6 (2014 MY)	148
46			Citroen Dongfeng C-Elysee (2013 MY)	702
47			Citroen Dongfeng C-Quatre	518
48			Citroen Dongfeng C3-XR (2015 MY)	401
49			Citroen Dongfeng C4L	472
50			Citroen Dongfeng C5	229

Table 18

2015	Imported	Freq.	Domestic	Freq.
51	Ford Edge (China, UAE & Saudi Arabia Only)	47	Ford Changan Ecosport	286
52	Ford Explorer	23	Ford Changan Edge (2015 MY)	80
53	Ford Mustang (2015 MY)	11	Ford Changan Escort (2015 MY)	453
54			Ford Changan Fiesta (2008 MY)	259
55			Ford Changan Focus (2012 MY)	802
56			Ford Changan Focus (Pre 2012 MY)	395
57			Ford Changan Kuga	418
58			Ford Changan Mondeo (2007 MY)	64
59			Ford Changan Mondeo (2013 MY)	325
60	Hyundai Genesis (2014 MY)	16	Hyundai Beijing Elantra Langdong	326
61	Hyundai Genesis Coupe (Pre 2014 MY)	3	Hyundai Beijing Elantra Yuedong	198
62	Hyundai Grand Santa Fe	38	Hyundai Beijing ix25 (2014 MY)	203
63	Hyundai Grandeur/ Azera (2010 MY)	2	Hyundai Beijing ix35	229
64	Hyundai Santa Fe (2012 MY)	5	Hyundai Beijing Mistra	261
65	Hyundai Veloster	18	Hyundai Beijing Santa Fe (2012 MY)	173
66			Hyundai Beijing Sonata (2011 MY)	150
67			Hyundai Beijing Sonata LF (2015 MY)	211
68			Hyundai Beijing Tucson (Pre 2015 MY)	85
69			Hyundai Beijing Verna (2011 MY)	344
70			Hyundai Hengtong Huatai Santa Fe	25
71	Infiniti Q50	58	Infiniti Dongfeng Q50L (2014 MY)	26
72	Infiniti Q70 (Was M (2011 MY))	30	Infiniti Dongfeng QX50L (2015 MY)	18
73	Infiniti QX60 (Was JX)	34		
74	Infiniti QX70 (Was FX)	21		
75	Kia Carens (2013 MY)	48	Kia Dongfeng Yueda Forte	155
76	Kia Carnival/ Sedona (2006 MY)	5	Kia Dongfeng Yueda K2	320
77	Kia K7 (S Korea & China Only)	5	Kia Dongfeng Yueda K3	449
78	Kia Mohave	2	Kia Dongfeng Yueda K4 (2014 MY)	249
79	Kia Sorento (2014 MY)	90	Kia Dongfeng Yueda K5	156
80			Kia Dongfeng Yueda KX3 (2015 MY)	141
81			Kia Dongfeng Yueda Soul	52
82			Kia Dongfeng Yueda Sportage	96
83			Kia Dongfeng Yueda Sportage R	203
84	Land Rover Discovery 4 (2010 MY)	165	Land Rover Chery Range Rover Evoque	117
85	Land Rover Discovery Sport (2015 MY)	127		
86	Land Rover Freelander 2 (2007 MY)	88		
87	Range Rover (2013 MY)	134		
88	Range Rover Evoque	160		
89	Range Rover Sport (2014 MY)	150		
90	Mazda 5 (2011 MY)	39	Mazda Changan 3 (2011 MY)	163
91			Mazda Changan 3 (2014 MY)	327
92			Mazda Changan CX-5	271
93			Mazda FAW 6 (2009 MY)	112
94			Mazda FAW 6 (2013 MY)	213
95			Mazda FAW 6 (Pre 2009 MY)	141
96			Mazda FAW 8	4
97			Mazda FAW CX-7 (2010 MY)	73
98	Mercedes A Class W176 (2013 MY)	54	Mercedes Beijing C Class V205 (2014 MY)	146
99	Mercedes B Class W246 (2011 MY)	48	Mercedes Beijing C Class W205 (2014 MY)	3
100	Mercedes C Class W/ S/ C204 (2007 MY)	12	Mercedes Beijing E Class V212 (2009 MY)	168
101	Mercedes CLA C117 (2013 MY)	50	Mercedes Beijing GLA X156	39
102	Mercedes CLS C218/ X218 (2011 MY)	26	Mercedes Beijing GLK X204	151
103	Mercedes E Coupe/ Cabrio C/ A207 (2009 MY)	26	Mercedes Fujian Viano	31
104	Mercedes G Wagen	3		
105	Mercedes GL X166 (2012 MY)	24		
106	Mercedes GLA X156	40		
107	Mercedes M Class W166 (2012 MY)	105		
108	Mercedes R Class W251	68		
109	Mercedes S Class W/ V222/ C217 (2013 MY)	84		
110	Mercedes SLK R172 (2011 MY)	10		

Table 19

2015	Imported	Freq.	Domestic	Freq.
111	Mitsubishi Outlander (2013 MY)	173	Mitsubishi Changfeng Liebao	24
112	Mitsubishi Pajero/ Shogun/ Montero (2007 MY)	66	Mitsubishi GAC ASX	185
113			Mitsubishi GAC Pajero	47
114			Mitsubishi GAC Pajero Sport (2013 MY)	140
115			Mitsubishi Southeast Lancer EX	106
116			Mitsubishi Southeast Lingyue/ V3	74
117	Peugeot 4008	13	Peugeot Dongfeng 2008	443
118			Peugeot Dongfeng 3008	468
119			Peugeot Dongfeng 301	461
120			Peugeot Dongfeng 308 (2014 MY)	60
121			Peugeot Dongfeng 308 (Pre 2014 MY)	472
122			Peugeot Dongfeng 408 (2014 MY)	483
123			Peugeot Dongfeng 508	167
124	Toyota Alphard (Pre 2015 MY)	29	Toyota FAW Corolla (2014 MY)	476
125	Toyota GT86/ 86	26	Toyota FAW Corolla EX	345
126	Toyota Previa (2006 MY)	23	Toyota FAW Crown (2010 MY)	35
127			Toyota FAW Crown (2015 MY)	58
128			Toyota FAW Land Cruiser 200	28
129			Toyota FAW Land Cruiser Prado (2010 MY)	41
130			Toyota FAW Prius	14
131			Toyota FAW RAV4 (2013 MY)	451
132			Toyota FAW Reiz	188
133			Toyota FAW Vios (2013 MY)	377
134			Toyota GAIG Camry (2012 MY)	464
135			Toyota GAIG Camry (Pre 2012 MY)	122
136			Toyota GAIG EZ	64
137			Toyota GAIG Highlander (2015 MY)	129
138			Toyota GAIG Highlander (Pre 2015 MY)	122
139			Toyota GAIG Levin	296
140			Toyota GAIG Yaris L	305
141	Volvo S60 (2010 MY)	12	Volvo Changan S80L	30
142	Volvo V40 (2012 MY)	89	Volvo S60 (2010 MY)	254
143	Volvo V40 Cross Country	14	Volvo XC60	271
144	Volvo V60 (2011 MY)	85	Volvo XC90 (Pre 2015 MY)	3
145	Volvo XC60	98		
146	VW Beetle (2012 MY)	99	VW FAW Bora (2008 MY)	1440
147	VW Eos	1	VW FAW CC	294
148	VW Golf (6) (2009 MY)	10	VW FAW Golf (7)	1014
149	VW Golf (7) (2013 MY)	35	VW FAW Jetta (2013 MY)	1609
150	VW Passat (2011 MY)/ Passat CC (2012 MY)	44	VW FAW Magotan (B7) (2011 MY)	1230
151	VW Passat Alltrack (2012 MY)	8	VW FAW Sagitar (2012 MY)	1676
152	VW Phaeton (2010 MY)	22	VW SVW Gran Lavida	635
153	VW Scirocco (2008 MY)	34	VW SVW Lamando (2015 MY)	310
154	VW Sharan (2010 MY)	39	VW SVW Lavida	1652
155	VW T5 Multivan (Caravelle in Britain) (2010 MY)	24	VW SVW Passat (2011 MY)	1144
156	VW Tiguan	82	VW SVW Polo (2012 MY)	942
157	VW Touareg (2010 MY)	120	VW SVW Santana (2012 MY)	1224
158	VW up!/ e-up!	14	VW SVW Tiguan	1335
159			VW SVW Touran (5) GP2 (2010 MY)	258

Part II

OPTIMAL INCOME TAXATION WITH MULTIDIMENSIONAL TAXPAYER TYPES

7 INTRODUCTION

[1] Beginning with Mirrlees, the optimal taxation literature has generally focused on economies where individuals are differentiated only by their productivity. Here we examine models where individuals are differentiated by up to five characteristics. We examine cases where individuals differ in productivity, elasticity of labor supply, basic needs, levels of distaste for work, and elasticity of demand for consumption. We find that the extra dimensionality produces substantially different results. In particular, we find cases of negative marginal tax rates for some high-productivity taxpayers. In our examples, income becomes a fuzzy signal of who should receive a subsidy under the planner's objective, and the planner chooses less redistribution than in more homogeneous societies. Multidimensional optimal tax problems are difficult nonlinear optimization problems because the linear independence constraint qualification does not hold at all feasible points and often fails to hold at the solution. To solve these nonlinear programs robustly, we initially used SNOPT in elastic mode, which has been shown to be effective for degenerate nonlinear programs. We found SNOPT was not able to solve larger examples, and were therefore motivated to develop algorithm NCL of Part III.

The Mirrlees (1971) optimal tax analysis and much of the literature that followed assumed that people differ only in their productivity while sharing common preferences over consumption and leisure. The world is not so simple. It is not realistic to think people are the same as long as they have the same productivity. A more realistic model would account for multidimensional heterogeneity. For example, some high ability people have low income because they prefer leisure, or the life of a scholar and teacher. In contrast, some low ability people have higher-than-expected income because circumstances, such as having to care for many children, motivate them to work hard. Despite the unrealistic aspects of this one-dimensional analysis, its conclusions have been applied to real tax problems. For example, Saez (2001) says "optimal income tax schedules have few general properties: we know that optimal tax rates must lie between 0 and 1 and that they equal zero at the top and bottom."

[1] This essay is co-authored with Kenneth Judd, Michael Saunders, and Che-Lin Su.

The presence of multidimensional heterogeneity is critically important for optimal taxation. In one-dimensional models, there is often a precise connection between what the government can observe, income, and how much the government wants to help (or tax) the person. Incentive constraints alter this connection, but the solution often involves full revelation of each individual's type. However, this clean connection between income and "merit" is less precise in the presence of multidimensional heterogeneity. If an individual has low income because he has low productivity, then we might want to help him whereas we would not want to help a high-productivity person choosing the same income because of his preference for leisure.

The basic question we ask here is "Does the presence of multidimensional heterogeneity reduce the optimal level of redistribution?" The intuition is clear: a one-dimensional signal like income is a noisy signal of merit, and as the signal to noise ratio falls, we will rely less on it to implement any policy. Our initial results support this conjecture.

There have been some attempts to look at multidimensional tax problems with a continuum of types. For example Mirrlees considers a general formulation, and Wilson (1993) and Wilson (1995) look at similar problems in the context of nonlinear pricing. However, both assume a first-order approach. This approach is justified in one-dimensional cases where the single-crossing property holds and implies that at the solution each type is tempted only by the bundles offered to one of the two neighboring types. This approach leads to a system of partial differential equations. Tarkianen and Tuomala (1999) solves one such example numerically, as does Wilson in the nonlinear pricing context. Unfortunately, no one has found useful assumptions that justify the first-order approach in the multidimensional case. The first-order approach assumes that the only alternatives that are tempting to a taxpayer are the choices made by others who are very similar in their characteristics. The single-crossing property in the one-dimensional case creates a kind of monotonicity that can be exploited to rule out the need to make global comparisons. However, there is no comparable notion of monotonicity in higher dimensions because there is no simple complete ordering of points in multidimensional Euclidean spaces.

The absence of an organizing principle like single-crossing does not alter the general theory; it only makes it harder. The general problem is still a constrained optimization problem: maximize social objective subject to incentive and resource constraints. However, the number of incentive constraints is enormous, and, unlike the single-crossing property in the one-dimensional case, there are no plausible assumptions that allow us to reduce the size of this problem.

There have been several studies that extended the Mirrlees analysis in multidimensional directions. Our initial perusal of that literature indicates that there has been only limited success. Some papers look at cases with a few (such as four) types of people, some consider using other instruments, such as commodity taxation, to sort out types, and some prove theorems of the form "**if** the solution has property A, **then** it also has property B," leaving us with little idea about how plausible Property A is.

Because multidimensional models are clearly more realistic than one-dimensional models, we numerically examine optimal taxation with multidimensional heterogeneity. First, we examine a two-dimensional case with both heterogeneous ability and heterogeneous elasticity of labor supply. This is a particularly useful example since it demonstrates how easy it is to get results different from the one-dimensional case and how any search for simplifying principles like single crossing is probably futile. In particular, we find that the optimal tax rate can be negative for the highest income earners! This contradicts one of the most basic results in the optimal tax literature, and the contradiction is due to the failure of binding incentive constraints to fall into a simple pattern. Second, we look at another two-dimensional model where people differ in ability and basic needs. In this model, income is a bad signal of a person's marginal utility of consumption (which, at the margin, is what the planner cares about) because high income could indicate high wages or an individual with moderate ability but large expenditures, such as medical expenses, that consume resources more than they contribute to utility. Third, we compute the solution to the optimal tax policy for a case of three-dimensional heterogeneity combining heterogeneous ability, elasticity of labor supply, and basic needs.

Our computations not only demonstrate the feasibility of our approach, but they also point towards interesting economic conclusions. When we compare the results of the 1D, 2D, and 3D models, we find that the optimal level of redistribution is significantly less as we add heterogeneity. One response to the reduced ability to redistribute income is to add instruments to tax policy (such as taxing those commodities demanded by those we want to tax) or to gather more facts about a taxpayer (such as his wage). Of course, the marginal value to a planner of making the tax system more complex will have to be balanced against the implementation costs. While it is reasonable to conjecture that we do not want to make the tax code as complex as the world, the exact balance between complexity costs and benefits is not immediately clear. In any case, the approach we take can be used to address this issue.

In all of these cases, we take a numerical approach. The lack of convexity in the incentive constraints implies the possible presence of multiple local optima; we use standard multi-start and other diagnostic techniques to avoid spurious local maxima. A more serious problem is the frequent result that the solution does not satisfy LICQ (linear independence constraint qualification). This makes it difficult for most optimization algorithms to find solutions. We use methods that can deal with moderate failure of LICQ, but many of our problems lie at the frontier of the state of the art in numerical optimization. Hence our development of algorithm NCL.

In this part of the thesis, we examine a small number of examples that highlight the ideas and present some interesting and suggestive initial results. We take the discrete-type approach because it makes no extra assumptions about the solution. The continuous-type approach used in Wilson (1995) and Tarkianen and Tuomala (1999) allows them to use powerful PDE methods but only after they have made strong assumptions about the solution. Since we want to avoid unjustified assumptions about the solution, we stay with models with finite types. This turns out to be justified because we find results that violate standard assumptions. In particular, we find cases where the marginal tax rate on the top income is negative. This appears to violate previous results. In particular, Corollary 6.1 in Guesnerie and Seade (1982) finds that marginal tax rates are always nonnegative in their multidimensional model, but they use "Assumption B", which is essentially a statement that the single-crossing property holds for some ordering of the distinct utility functions. Guesnerie and Seade (1982) admit that this assumption will not hold when there are many types with a multidimensional structure. Indeed, we find rather small examples that violate Assumption B and produce negative marginal tax rates.

The examples show that heterogeneity has substantial impact on optimal tax policy. We also show that computational approaches to optimal tax problems are possible when one uses high-quality optimization software. Further work will examine the robustness of these examples, but the efficiency of our algorithms will allow us to look at a wide variety of specifications for tastes and productivities.

8

TAXATION WITH ONE-DIMENSIONAL TAXPAYER TYPES

We begin with examples of the classical Mirrlees problem. Later we compare them to the optimal tax policies in more hetereogeneous models.

Assume we have N taxpayers ($N > 1$). There are two types of goods: consumption and labor services. Let c_i and l_i denote taxpayer i's consumption and labor supply, with productivity represented by wage rate w_i. We index the taxpayers so that taxpayer i is less productive than taxpayer $i + 1$:

$$0 < w_1 < \cdots < w_N. \tag{8.1}$$

A type i taxpayer has pre-tax income equal to

$$y_i := w_i l_i, \quad i = 1, \ldots, N. \tag{8.2}$$

Individuals have common utility function over consumption and labor supply: $u : R_+ \times R_+ \to R$. We assume that u is a continuously differentiable, strictly increasing, and strictly concave function with $u_c(0, l) = \frac{\partial u}{\partial c} = \infty$, and $\lim_{c \to \infty} u_c(c, l) = 0$. We next define the implied utility function $U^i : R_+ \times R_+ \to R$ over income and consumption:

$$U^i(c_i, y_i) := u(c_i, y_i / w_i), \quad i = 1, \ldots, N. \tag{8.3}$$

For many preferences (such as quasilinear utility) over income and consumption, higher ability individuals have flatter indifference curves, and indifference curves in income-consumption space of different individuals intersect only once; this defines the *single-crossing property*.

An *allocation* is a vector $a := (y, c)$, where $y := (y_1, \ldots, y_N)$ is an *income vector* and $c := (c_1, \ldots, c_N)$ is a *consumption vector*. The *social welfare function* $W : R_+^N \times R_+^N \to R$ is of the weighted utilitarian form:

$$W(a) := \sum_i \lambda_i U^i(c_i, y_i). \tag{8.4}$$

We typically assume that the weights λ_i are positive and nonincreasing in ability. The case where λ_i equals the population frequency of type i is the utilitarian social welfare function. We take the utilitarian approach. We assume Output is proportional to total labor supply, which is the only input. Therefore, technology imposes the constraint

$$\sum_i \lambda_i c_i \leq \sum_i \lambda_i y_i. \tag{8.5}$$

We also assume $c_i \geq 0$ and $y_i \geq 0$.

We assume that the government knows the distribution of wages and the common utility function, that it can measure the pretax income of each taxpayer, but cannot observe a taxpayer's labor supply nor his wage rate. This corresponds to assuming that each taxpayer's tax payment is a function solely of his labor income. We also assume that all taxpayers face the same tax rules. Therefore, each taxpayer can choose any (y_i, c_i) bundle suggested by the government. The government must choose a schedule such that type i taxpayers will choose the (y_i, c_i) bundle; therefore, the allocation must satisfy the incentive-compatibility or self-selection constraint:

$$U^i(c_i, y_i) \geq U^i(c_p, y_p) \text{ for all } i, p, \tag{8.6}$$

which states that each person weakly prefers the consumption and income bundle meant for his type to those for other types of people. If the tax policy satisfies (8.6), then it is common knowledge that an individual with wage w_i will choose (y_i, c_i) from the set $\{(y_1, c_1), \ldots, (y_N, c_N)\}$.

The optimal nonlinear income tax problem is equivalent to the following nonlinear optimization problem, where the government chooses a set of commodity bundles:

$$\begin{array}{rll} \max\limits_{c_i, y_i} & \sum_i \lambda_i U^i(c_i, y_i) & \\ \text{s.t.} & U^i(c_i, y_i) - U^i(c_p, y_p) & \geq 0 \text{ for all } i, p \\ & \sum_i \lambda_i y_i - \sum_i \lambda_i c_i & \geq 0 \\ & c_i, y_i & \geq 0 \text{ for all } i. \end{array} \tag{8.7}$$

8.1 MIRRLEES CASES

We first consider examples of the form

$$u(c,l) = \log c - \frac{l^{1/\eta_0+1}}{1/\eta_0+1} \tag{8.8}$$

with $N = 5$, $w_i \in \{1,2,3,4,5\}$, $\lambda_i = 1$, where different values of η_0 correspond to different examples. The zero tax commodity bundle for type i is the solution to $\max_{l_i} u^i(w_i l_i, l_i)$, which we denote (c_i^*, l_i^*, y_i^*). The zero tax solution here is $l_i^* = 1$ and $c_i^* = w_i$. We compute the solutions for $\eta_0 = 1,\ 1/2,\ 1/3,\ 1/5,\ 1/8$, and report

$$y_i, \quad \frac{y_i - c_i}{y_i} \text{ (average tax rate)}, \quad 1 - \frac{u_l}{w u_c} \text{ (marginal tax rate)}, \quad l_i/l_i^*, \quad c_i/c_i^*$$

for $i = 1, \ldots, N$ in the tables below.

The pattern of the binding incentive-compatibility constraints is the simple monotonic chain to the left property as expected in nonlinear optimal tax problems in one dimension. Note that the results are as expected. Marginal and average tax rates on the types that pay taxes are moderately high, and increase as the elasticity of labor supply falls. The subsidy rates to the poor fall as we move from the high-elasticity world to the low-elasticity world because the high marginal rates the poor face depress their labor supply much more in the high-elasticity world; remember, all people in each of these economies have the same elasticity.

$\eta = 1$					
i	y_i	$\frac{y_i - c_i}{y_i}$	MTR_i	l_i/l_i^*	c_i/c_i^*
1	0.40	-2.87	0.36	0.40	1.56
2	1.31	-0.45	0.38	0.65	0.95
3	2.56	0.03	0.29	0.85	0.83
4	4.01	0.16	0.16	1.00	0.84
5	5.54	0.19	0.00	1.10	0.90

$\eta = 1/2$					
i	y_i	$\frac{y_i - c_i}{y_i}$	MTR_i	l_i/l_i^*	c_i/c_i^*
1	0.60	-2.09	0.31	0.60	1.87
2	1.54	-0.39	0.35	0.77	1.08
3	2.69	0.02	0.29	0.89	0.87
4	3.99	0.17	0.17	0.99	0.82
5	5.41	0.21	0.00	1.08	0.85

$\eta = 1/3$					
i	y_i	$\frac{y_i - c_i}{y_i}$	MTR_i	l_i/l_i^*	c_i/c_i^*
1	0.70	-1.91	0.28	0.70	2.06
2	1.66	-0.38	0.33	0.83	1.15
3	2.77	0.02	0.29	0.92	0.90
4	3.99	0.17	0.18	0.99	0.82
5	5.33	0.23	0.00	1.06	0.82

$\eta = 1/5$					
i	y_i	$\frac{y_i - c_i}{y_i}$	MTR_i	l_i/l_i^*	c_i/c_i^*
1	0.80	-1.84	0.22	0.80	2.29
2	1.78	-0.39	0.29	0.89	1.24
3	2.85	0.02	0.27	0.95	0.93
4	4.01	0.19	0.18	1.00	0.81
5	5.25	0.26	0.00	1.05	0.77

$\eta = 1/8$					
i	y_i	$\frac{y_i - c_i}{y_i}$	MTR_i	l_i/l_i^*	c_i/c_i^*
1	0.87	-1.84	0.17	0.87	2.48
2	1.86	-0.41	0.24	0.93	1.31
3	2.91	0.02	0.23	0.97	0.95
4	4.02	0.20	0.16	1.00	0.80
5	5.19	0.28	0.00	1.03	0.73

9

MULTIDIMENSIONAL HETEROGENEITY

We next consider models with multidimensional heterogeneity. One kind of multidimensional heterogeneity is where people differ in both productivity and elasticity of labor supply, η. We examine that and other types of heterogeneity. More generally, we consider utility functions of the form

$$u(c,l) = \frac{(c-\alpha)^{1-1/\gamma}}{1-1/\gamma} - \psi \frac{l^{1/\eta+1}}{1/\eta+1}, \tag{9.1}$$

where α, γ, ψ, and η are possible taxpayer heterogeneities, in addition to wage w. Each term has a natural economic interpretation. The parameter α represents basic needs—a minimal level of consumption. A high α implies a higher marginal utility of consumption at any c. The parameter γ represents the elasticity of demand for consumption, whereas ψ represents the level of distaste for work. The parameter η represents labor supply responsiveness to the wage. This general specification implies a 5D specification of taxpayer types, and the corresponding 5D nonlinear optimization problem is

$$\begin{aligned}
&\max_{c_{i,j,k,g,h},\, y_{i,j,k,g,h}} \sum_{i,j,k,g,h} \lambda_{i,j,k,g,h} U^{i,j,k,g,h}(c_{i,j,k,g,h}, y_{i,j,k,g,h}) \\
\text{s.t.}\quad & U^{i,j,k,g,h}(c_{i,j,k,g,h}, y_{i,j,k,g,h}) - U^{i,j,k,g,h}(c_{p,q,r,s,t}, y_{p,q,r,s,t}) \geq 0 \quad \forall (i,j,k,g,h),(p,q,r,s,t) \\
& \sum_{i,j,k,g,h} \lambda_{i,j,k,g,h} y_{i,j,k,g,h} - \sum_{i,j,k,g,h} \lambda_{i,j,k,g,h} c_{i,j,k,g,h} \geq 0 \\
& c_{i,j,k,g,h}, y_{i,j,k,g,h} \geq 0 \qquad \forall (i,j,k,g,h),
\end{aligned} \tag{9.2}$$

where

$$\begin{cases}
i,p = 1\!:\!na & (na = \text{number of different wage types}) \\
j,q = 1\!:\!nb & (nb = \text{number of different elasticity of labor supply}) \\
k,r = 1\!:\!nc & (nc = \text{number of different basic need types}) \\
g,s = 1\!:\!nd & (nd = \text{number of different level of distaste for work}) \\
h,t = 1\!:\!ne & (ne = \text{number of different elasticity of demand for consumption}).
\end{cases} \tag{9.3}$$

The following is a fully indexed version of (9.1):

$$U^{i,j,k,g,h}(c_{p,q,r,s,t}, y_{p,q,r,s,t}) = \frac{(c_{p,q,r,s,t} - \alpha_k)^{1-1/\gamma_h}}{1 - 1/\gamma_h} - \psi_g \frac{(\frac{y_{p,q,r,s,t}}{w_i})^{1/\eta_j+1}}{1/\eta_j + 1}. \tag{9.4}$$

Hence, if we assume that taxpayers share the same elasticity of demand for consumption, i.e., $ne = 1$, $\gamma_h = 1, \forall h$, we have a 4D specification of taxpayers' utility function:

$$U^{i,j,k,g}(c_{p,q,r,s}, y_{p,q,r,s}) = \log(c_{p,q,r,s} - \alpha_k) - \psi_g \frac{(\frac{y_{p,q,r,s}}{w_i})^{1/\eta_j+1}}{1/\eta_j + 1}. \tag{9.5}$$

If we further assume that taxpayers share the same level of distaste for work, i.e., $nd = 1, \psi_g = 1, \forall g$, we have a 3D specification of taxpayers' utility function:

$$U^{i,j,k}(c_{p,q,r}, y_{p,q,r}) = \log(c_{p,q,r} - \alpha_k) - \frac{(\frac{y_{p,q,r}}{w_i})^{1/\eta_j+1}}{1/\eta_j + 1}. \tag{9.6}$$

Furthermore, if we add the assumption that taxpayers share the same basic need, i.e., $nc = 1, \alpha_k = 0, \forall k$, we have a 2D specification of taxpayers' utility function:

$$U^{i,j}(c_{p,q}, y_{p,q}) = \log c_{p,q} - \frac{(\frac{y_{p,q}}{w_i})^{1/\eta_j+1}}{1/\eta_j + 1}. \tag{9.7}$$

Finally, if we add the assumption that taxpayers share the same elasticity of labor supply, i.e., $nb = 1, \eta_j = \eta_0, \forall j$, we reach the Mirrlees' case:

$$U^{i}(c_{p}, y_{p}) = \log c_{p} - \frac{(\frac{y_{p}}{w_i})^{1/\eta_0+1}}{1/\eta_0 + 1}. \tag{9.8}$$

10

COMPUTATIONAL DIFFICULTIES AND SOLUTIONS FOR NLPS WITH INCENTIVE CONSTRAINTS

The 5D general taxation nonlinear program (9.2) is a difficult problem to solve numerically. The objective is concave, but there are many constraints. The number of variables is $na \times nb \times nc \times nd \times ne \times 2 := 2|\mathcal{T}|$, and the number of nonlinear constraints is $|\mathcal{T}| * (|\mathcal{T}| - 1)$, the square of the number of variables. This is a feature of all incentive problems; only in some with simplifying principles like single-crossing are we able to significantly reduce the number of constraints.

10.1 INITIALIZE WITH A FEASIBLE SOLUTION

Given the complexity of the taxation optimization problems, it often helps tremendously to provide any optimization solver that you may choose a feasible starting point. In fact, with the solvers and algorithms that we have tried, from 3D and up, the convergence depends on the feasible starting point. For ease of understanding and notation, we use a 1D optimization problem to illustrate the method of finding a feasible point.

To search for a feasible point of any optimization problem, we could solve exactly the same problem with a constant objective function:

$$\begin{array}{rll} \max\limits_{c_i, y_i} & 0 & \\ \text{s.t.} & U^i(c_i, y_i) - U^i(c_p, y_p) & \geq 0 \;\text{ for all } i, p \\ & \sum_i \lambda_i y_i - \sum_i \lambda_i c_i & \geq 0 \\ & c_i, y_i & \geq 0 \;\text{ for all } i, \end{array} \tag{10.1}$$

which has a zero tax special case for each wage type i:

$$\begin{array}{rll} \max\limits_{c_i, y_i} & U^i(c_i, y_i) & \\ \text{s.t.} & y_i - c_i & = 0 \;\text{ for all } i \\ & c_i, y_i & \geq 0 \;\text{ for all } i. \end{array} \tag{10.2}$$

If we define (y^0, c^0) as the solution of the problem (10.2), then

$$(y^0, c^0) = \left[\text{argmax}_{y_i, c_i} \quad U^i(c_i, y_i) \quad \text{s.t.} \quad y_i = c_i\right] \quad \forall i. \tag{10.3}$$

Since different (y_i, c_i) do not interact with each other, maximizing each individual utility is equivalent to maximizing the sum of the individual utilities. Hence, we can find a feasible point by solving one simpler optimization problem:

$$(y^0, c^0) = \text{argmax}_{y_i, c_i} \quad \sum_i U^i(c_i, y_i) \quad \text{s.t.} \quad \left[y_i = c_i, \forall i\right]. \tag{10.4}$$

10.2 GLOBAL EXTENSIONS OF COMMON UTILITY FUNCTIONS

Many utility functions are defined over a bounded domain. For example, neither $\log c$ nor $c^{1-1/\gamma}/(1-1/\gamma)$ is defined for $c < 0$. This can create significant numerical computation difficulty in economics. A common opinion is that there is no difficulty with these functions being undefined for negative consumption. In fact, economists often exploit the Inada condition (that is, $u'[c] = \infty$) to prove interior solutions to optimization problems. However, there are computational and economic reasons to examine utility functions defined over negative consumption.

First, numerical methods will often want to evaluate utility functions at negative values. Even if one imposes a positivity constraint on consumption, some solvers will take that to mean that the solution must satisfy that constraint, not that the objective function is undefined for negative values. Second, economists need to remember that "consumption" generally refers to consumption of market goods. Aspects of real life, such as home production, unreported transactions, and charity imply that people may have zero consumption of market goods but still have well-defined utility in reality. Barter would imply negative consumption of market goods: suppose you bring eggs to a grocery store, sell them to the grocer, and use the proceeds to buy soap. In terms of market goods, you would have negative consumption of eggs and positive consumption of soap. While such transactions may be rare today, they were quite common less than a century ago.

A more common problem is that utility may be undefined even for positive levels of consumption. Examples include utility functions like $(c-\alpha)^{1-1/\gamma}/(1-1/\gamma)$ or $\log(c-\alpha)$ for "minimum" consumption level represented by α. There is no difficulty if $c > \alpha$, but utility will be undefined if $c < \alpha$. We put the term "minimum" in quotations because the α parameter is used to avoid linear Engel curves, not because of some empirical observation about behavior for $c < \alpha$. In general, unless there is no possibility that the solution involves $c \leq \alpha$, the utility function should be defined over all nonnegative consumption levels. Also, in a multi-good context, the utility function should allow negative consumption of individual items.

The need for global extensions of common utility functions may not be extremely clear until we try solving 4D and 5D optimal taxation problems. Because of the high dimensionality, even with additional bounds for the consumption c's, the cross comparison in the incentive constraints could entail evaluating utility functions at points where they are not defined.

Hence, we define alternative utility functions that agree with a standard utility function over most of its domain but are extended to be defined globally. We also want the extended utility functions to satisfy the usual requirements of utility functions, such as monotonicity and concavity.

Here we use a quadratic function to extend the utility function to negative consumption. One can also understand it as taking a Taylor expansion of the original utility function at a point close to 0. For the 3D case, we obtain a piece-wise utility function:

$$U^{i,j,k}(c_{p,q,r}, y_{p,q,r}) = \begin{cases} \log(c_{p,q,r} - \alpha_k) - \frac{(\frac{y_{p,q,r}}{w_i})^{1/\eta_j+1}}{1/\eta_j+1}, \text{ if } c_{p,q,r} > \alpha_k \\ -\frac{1}{2\epsilon^2}(c_{p,q,r} - \alpha_k)^2 + \frac{2}{\epsilon}(c_{p,q,r} - \alpha_k) + \log\epsilon - \frac{3}{2} - \frac{(\frac{y_{p,q,r}}{w_i})^{1/\eta_j+1}}{1/\eta_j+1}, \text{ o.w.} \end{cases} \tag{10.5}$$

For a 4D model, the extended utility function is quite similar, with an additional parameter ψ_g. For a 5D model, with a little bit more algebra, we obtain

$$U^{i,j,k,g,h}(c_{p,q,r,s,t}, y_{p,q,r,s,t}) = \begin{cases} \frac{(c_{p,q,r,s,t}-\alpha_k)^{1-1/\gamma_h}}{1-1/\gamma_h} - \psi_g \frac{(\frac{y_{p,q,r,s,t}}{w_i})^{1/\eta_j+1}}{1/\eta_j+1}, \text{ if } c_{p,q,r,s,t} > \alpha_k \\ -\frac{1}{2\gamma_h}\epsilon^{-1-1/\gamma_h}(c_{p,q,r,s,t} - \alpha_k)^2 + (1 + \frac{1}{\gamma_h})\epsilon^{-1/\gamma_h}(c_{p,q,r,s,t} - \alpha_k) \\ \qquad +(\frac{1}{1-1/\gamma_h} - 1 - \frac{1}{2\gamma_h})\epsilon^{1-1/\gamma_h} - \psi_g \frac{(\frac{y_{p,q,r,s,t}}{w_i})^{1/\eta_j+1}}{1/\eta_j+1}, \text{ o.w.} \end{cases} \tag{10.6}$$

10.3 linear independence constraint qualification (licq)

There is no reason to believe that the constraints are concave. This creates two problems. First, we cannot ignore the possibility of multiple local optima. We deal with this in standard ways, so will not discuss further details.

The second problem is more challenging: the failure of useful constraint qualifications. Recall the structure of constrained optimization problems. Consider the inequality constrained problem

$$\min f(x) \quad \text{s.t.} \quad c(x) \geq 0,$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow R^m$ are assumed smooth. There are many numerical algorithms for solving such problems, but they generally assume that the solution x^* satisfies some constraint qualification. Define the set of binding constraints

$$\mathcal{A}^* = \{i = 1, 2, \ldots, m \mid c_i(x^*) = 0\}.$$

The linear independence constraint qualification (LICQ) states that the gradients $\nabla c_i(x^*)$ of the binding constraints $i \in \mathcal{A}$ are linearly independent. The Mangasarian-Fromovitz constraint qualification (MFCQ) assumes that there is a direction d such that $\nabla c_i(x^*)^T d > 0$ for all $i \in \mathcal{A}^*$.

In 1D models with single-crossing problems, the LICQ will generally hold. However, multidimensional problems can easily run afoul of the LICQ for one simple reason: if the number of binding constraints exceeds the number of variables, as with multidimensional pooling, it is impossible for LICQ to hold because multidimensional problems are not likely to satisfy a simple pattern of binding incentive constraints. In fact, we found many cases where the LICQ could not hold.

LICQ is a sufficient condition for local convergence of many optimization algorithms, but not a necessary condition. However, the failure of LICQ will at least slow down convergence. In many cases the number of major iterations needed was unusually large, sometimes in the order of thousands, even for a 2D problem with 25 total types. The failure of the LICQ is probably the sole source of difficulties in solving these nonlinear programs. Other issues, such as scaling, e.g., the range of input parameters w or η, can also cause difficulties. However, we found LICQ will often fail at solutions of such problems.

Nonlinear programs with constraint qualification failures have been the object of much research in numerical optimization in the past decade. More generally, much progress has been made on a new class of problems called *mathematical programs with equilibrium constraints* (MPECs); see Luo, Pang, and Ralph (1996) and Outrata, Kocvara, and Zowe (1998). One well-known property about MPECs is that the standard CQs fail at every feasible point. Perhaps MPEC methods will be able to solve larger and more complex problems. Instead, we present a new approach in the following section.

10.4 NCL: A ROBUST SOLUTION PROCEDURE

The optimization problems to be solved are of the form

$$\boxed{\begin{array}{lll} \text{NCO} & \underset{x\in\Re^n}{\text{minimize}} & \phi(x) \\ & \text{subject to} & c(x) \geq 0, \quad Ax \geq b, \quad \ell \leq x \leq u, \end{array}}$$

where $\phi(x)$ is a smooth nonlinear function, $c(x) \in \Re^m$ is a vector of smooth nonlinear functions, and $Ax \geq b$ is a placeholder for a set of linear inequality or equality constraints, with x lying between lower and upper bounds ℓ and u. In our case, m greatly exceeds n and many of the contraints in $c(x) \geq 0$ may be essentially active at a solution. General-purpose solvers have difficulty converging because the nonlinear constraints do not satisfy the constraint qualification LICQ.

In Part III, we have derived the NCL (nonlinearly constrained Lagrangian) algorithm for solving problem NCO by solving a sequence of subproblems of the form

$$\boxed{\begin{array}{lll} \text{NC}_k & \underset{x\in\Re^n,\, r\in\Re^m}{\text{minimize}} & \phi(x) + y_k^T r + \frac{1}{2}\rho_k\|r\|^2 \\ & \text{subject to} & c(x) + r \geq 0, \quad Ax \geq b, \quad \ell \leq x \leq u, \end{array}}$$

in which r serves to make the nonlinear constraints independent, y_k estimates the Lagrange multipliers for $c(x) \geq 0$, and ρ_k is a positive penalty parameter. Problem NC$_k$ is solved approximately to give (x_k^*, r_k^*). If $\|r_k^*\|$ is sufficiently small ($\|r_k^*\| \leq \tilde{\eta}_k$), the multiplier estimate is updated ($y_{k+1} = y_k + \rho_k r_k^*$). Otherwise, the penalty parameter is increased ($\rho_{k+1} > \rho_k$).

Key properties of NCL are that the subproblems can be solved inexactly (with tightening optimality tolerance $\omega_k \searrow 0$), "sufficiently small" becomes more demanding (with tightening feasibility tolerance $\eta_k \searrow 0$), and ρ_k increases only finitely often, as illustrated by the table below for a 4D example with $na = 11$, $nb = 3$, $nc = 3$, $nd = 2$, giving $m = 39007$, $n = 396$. Problem NCO and algorithm NCL were formulated in the AMPL modeling language (Fourer, Gay, and Kernighan, 2002). The solvers SNOPT (Gill, Murray, and Saunders, 2005a) and

IPOPT (Wächter and Biegler, 2006) were unable to solve NCO itself, but algorithm NCL with IPOPT as solver gave successful results as follows:

k	ρ_k	$\tilde{\eta}_k$	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	Itns	Time
1	1.0e+02	1.0e-02	3.1e-03	-2.1478532e+01	125	42.8
2	1.0e+02	1.0e-03	1.3e-03	-2.1277587e+01	18	6.5
3	1.0e+03	1.0e-03	6.6e-04	-2.1177152e+01	27	9.1
4	1.0e+03	1.0e-04	5.5e-04	-2.1110210e+01	31	10.8
5	1.0e+04	1.0e-04	2.9e-04	-2.1066664e+01	57	24.3
6	1.0e+05	1.0e-04	6.5e-05	-2.1027152e+01	75	26.8
7	1.0e+05	1.0e-05	5.2e-05	-2.1018896e+01	130	60.9
8	1.0e+06	1.0e-05	9.3e-06	-2.1015295e+01	159	81.8
9	1.0e+06	1.0e-06	2.0e-06	-2.1014808e+01	139	70.0
10	1.0e+07	1.0e-06	2.1e-07	-2.1014800e+01	177	97.6

The optimality tolerance for IPOPT was $\omega_k = 10^{-6}$ throughout, and warm starts were specified for $k \geq 2$ (options warm_start_init_point=yes, mu_init=1e-4). Itns refers to IPOPT's primal-dual interior point method, and time is seconds on an Apple iMac with 2.93 GHz Intel Core i7.

11

NUMERICAL EXAMPLES

We now examine economies with multiple dimensions of heterogeneity.

11.1 WAGE-LABOR SUPPLY ELASTICITY HETEROGENEITY

We first consider the case where taxpayers differ in terms of their wage and elasticity of labor supply (w and η). We assume that $\alpha_k = 0$, $\gamma_h = 1$ (log utility), and $\psi_g = 1$. We consider an optimal nonlinear income tax problem with 2D types of taxpayers:

$$\begin{array}{rll} \max\limits_{c_{i,j},y_{i,j}} & \sum_{i,j} \lambda_{i,j} U^{i,j}(c_{i,j},y_{i,j}) & \\ \text{s.t.} & U^{i,j}(c_{i,j},y_{i,j}) - U^{i,j}(c_{p,q},y_{p,q}) & \geq 0 \ \ \forall (i,j),(p,q) \\ & \sum_{i,j} \lambda_{i,j} y_{i,j} - \sum_{i,j} \lambda_{i,j} c_{i,j} & \geq 0 \\ & c_{i,j}, y_{i,j} & \geq 0 \ \ \forall (i,j), \end{array} \tag{11.1}$$

where $U^{i,j}(c_{i,j},y_{i,j})$ and $U^{i,j}(c_{p,q},y_{p,q})$ are defined in (9.7). We choose the following parameters: $na = nb = 5$, $w_i \in \{1,2,3,4,5\}$, $\eta_j \in \{1,\ 1/2,\ 1/3,\ 1/5,\ 1/8\}$, and $\lambda_{ij} = 1$.

We use the zero-tax solution (c^*, y^*) from section 10.1 as a starting point for SNOPT. We report numerical results in Tables 20 and 21. There are several points to emphasize.

1. All taxpayers with wage type $w_i = 4$ are pooled. Table 21 shows that there are many more binding constraints than there are variables. This tells us that our worries about failures of LICQ are well-founded.

2. This example violates the general result in 1D optimal tax theory that marginal tax rates lie between zero and one. Consider taxpayers with wage rate $w_i = 5$. In particular, the taxpayers with low labor supply elasticity η tend to work less and make less income. However, they pay more tax than those taxpayers with high labor supply elasticity η, who have higher income. We also find negative marginal tax rate for the high-productivity types with $w = 5$. These

results are quite different from the results for 1D-type taxpayers in section 8.1 as well as general conclusions in optimal income taxation literature.

3. All high-productivity types are better off in the heterogeneous world. We are not surprised that the low-elasticity high-productivity types are better off, because their low elasticity was exploited in the world where all had low labor supply elasticity. In a heterogeneous world, the average elasticity of labor is higher, and so there should be lower taxes on high-productivity workers. The surprise is that the high-elasticity, high-productivity workers also gain by hiding in a heterogeneous world. The reason, as seen in Table 21, is that these workers do respond to incentives and find that it is tempting to join the pool at $w = 4$. This case is also an example of where the binding constraints are not local, because the highest income type is tempted to pretend to be workers with much less income.

4. Heterogeneity reduces redistribution. This is related to point 3 above, and is highlighted in Figures 5 and 6, where we see that the tax schedule and the average tax rates are almost uniformly lower in the heterogeneous world than in any of the individual Mirrlees economies. Hence, redistribution in the heterogeneous world is not just the average of redistribution in the simpler worlds, but instead is substantially less.

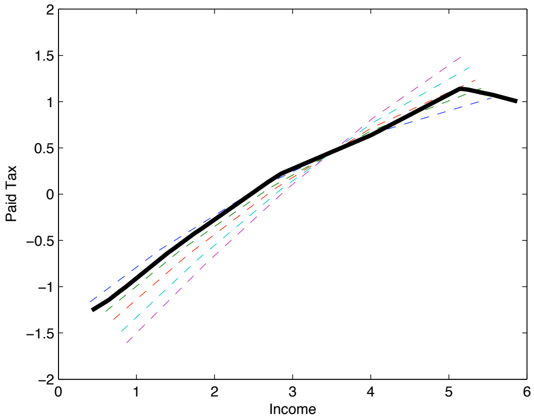

Figure 5: Income vs. Paid Tax for the 2D heterogeneity example. Thin lines represent taxes when all have same elasticity of labor supply. The thick line is the 2D heterogeneity case.

Table 20: $\eta = (1, 1/2, 1/3, 1/5, 1/8)$, $w = (1, 2, 3, 4, 5)$

(i,j)	c_{ij}	y_{ij}	$\Delta TR_{i,j}$	$MTR_{i,j}$	$ATR_{i,j}$	l_{ij}/l^*_{ij}	c_{ij}/c^*_{ij}	Utility	
								Judd et al.	Mirrlees
$(1,1)$	1.68	0.42		0.28	-2.92	0.42	1.68	0.4294	.3641
$(1,2)$	1.77	0.62	0.51	0.32	-1.86	0.62	1.77	0.4952	.3138
$(1,3)$	1.79	0.65	0.54	0.51	-1.75	0.65	1.79	0.5378	.6601
$(1,4)$	1.83	0.77	0.66	0.50	-1.37	0.77	1.83	0.5700	.7830
$(1,5)$	1.86	0.86	0.62	0.43	-1.16	0.86	1.86	0.5940	.8760
$(2,1)$	1.86	0.86		0.60	-1.16	0.43	0.93	0.5308	.3751
$(2,2)$	2.03	1.39	0.68	0.50	-0.45	0.69	1.01	0.5973	.6180
$(2,3)$	2.07	1.50	0.60	0.56	-0.38	0.75	1.03	0.6512	.7189
$(2,4)$	2.16	1.74	0.62	0.46	-0.24	0.87	1.08	0.7006	.8181
$(2,5)$	2.20	1.83	0.55	0.46	-0.20	0.91	1.10	0.7413	.9085
$(3,1)$	2.20	1.83		0.55	-0.20	0.61	0.73	0.6053	.5496
$(3,2)$	2.47	2.49	0.59	0.43	0.00	0.83	0.82	0.7157	.7269
$(3,3)$	2.47	2.49		0.53	0.00	0.83	0.82	0.7878	.8158
$(3,4)$	2.55	2.68	0.59	0.52	0.04	0.89	0.85	0.8520	.9057
$(3,5)$	2.62	2.85	0.54	0.42	0.07	0.95	0.87	0.8965	.9672
$(4,1)$	3.36	4.00	0.36	0.16	0.15	1.00	0.84	0.7127	.7090
$(4,2)$	3.36	4.00	–	0.16	0.15	1.00	0.84	0.8794	.8664
$(4,3)$	3.36	4.00	–	0.15	0.15	1.00	0.84	0.9627	.9402
$(4,4)$	3.36	4.00	–	0.15	0.15	1.00	0.84	1.0461	1.0080
$(4,5)$	3.36	4.00		0.15	0.15	1.00	0.84	1.1017	1.0476
$(5,5)$	4.00	5.14	0.44	0	0.22	1.02	0.80	1.2439	1.1487
$(5,4)$	4.11	5.24	-0.10	-0.05	0.21	1.04	0.82	1.1928	1.1331
$(5,3)$	4.34	5.43	-0.17	-0.12	0.20	1.08	0.86	1.1188	1.0877
$(5,2)$	4.49	5.56	-0.17	-0.11	0.19	1.11	0.89	1.0428	1.0286
$(5,1)$	4.87	5.87	-0.22	-0.15	0.17	1.17	0.97	0.8933	.8901

Table 21: Binding IC[(i,j), (i',j')]

(i,j)	(p,q)	(i,j)	(p,q)
		$(4,1)$	$(3,2)$, $(3,3)$, $(3,5)$, $(4,2)$, $(4,3)$, $(4,4)$, $(4,5)$
$(1,2)$	$(1,1)$	$(4,2)$	$(4,1)$, $(4,3)$, $(4,4)$, $(4,5)$
$(1,3)$	$(1,2)$	$(4,3)$	$(4,1)$, $(4,2)$, $(4,4)$, $(4,5)$
$(1,4)$	$(1,3)$	$(4,4)$	$(4,1)$, $(4,2)$, $(4,3)$, $(4,5)$
$(1,5)$	$(1,4)$, $(2,1)$	$(4,5)$	$(4,1)$, $(4,2)$, $(4,3)$, $(4,4)$
$(2,1)$	$(1,4)$, $(1,5)$	$(5,1)$	$(4,1)$, $(4,2)$, $(4,3)$, $(4,4)$, $(4,5)$
$(2,2)$	$(1,5)$, $(2,1)$	$(5,2)$	$(4,1)$, $(4,2)$, $(4,3)$, $(4,4)$, $(4,5)$, $(5,1)$
$(2,3)$	$(2,2)$	$(5,3)$	$(5,2)$
$(2,4)$	$(2,3)$	$(5,4)$	$(5,3)$
$(2,5)$	$(2,4)$, $(3,1)$	$(5,5)$	$(5,4)$
$(3,1)$	$(2,3)$, $(2,5)$		
$(3,2)$	$(2,5)$, $(3,1)$, $(3,3)$		
$(3,3)$	$(3,2)$		
$(3,4)$	$(3,2)$, $(3,3)$		
$(3,5)$	$(3,4)$		

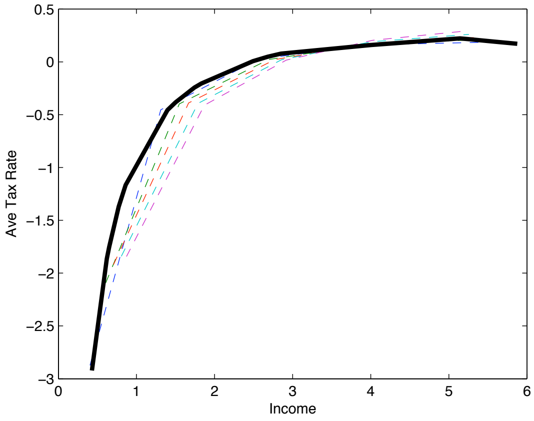

Figure 6: Income vs. Average Tax Rate for the 2D heterogeneity example.

11.2 3D HETEROGENEITY

We next consider a case of heterogeneity in wages, basic needs, and labor supply elasticity

$$\begin{array}{rlll} \max\limits_{c_{i,j,k},y_{i,j,k}} & \sum_{i,j,k} \lambda_{i,j,k} U^{i,j,k}(c_{i,j,k}, y_{i,j,k}) & & \\ \text{s.t.} & U^{i,j,k}(c_{i,j,k}, y_{i,j,k}) - U^{i,j,k}(c_{p,q,r}, y_{p,q,r}) & \geq 0 & \forall (i,j,k),(p,q,r) \\ & \sum_{i,j,k} \lambda_{i,j,k} y_{i,j,k} - \sum_{i,j,k} \lambda_{i,j,k} c_{i,j,k} & \geq 0 & \\ & c_{i,j,k}, y_{i,j,k} & \geq 0 & \forall (i,j,k), \end{array} \tag{11.2}$$

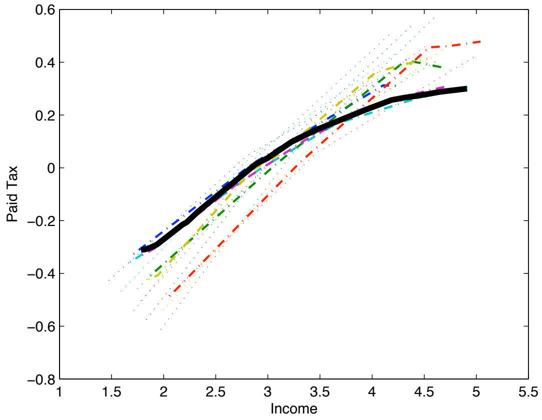

Figure 7: Income vs. Paid Tax for the 3D heterogeneity example.

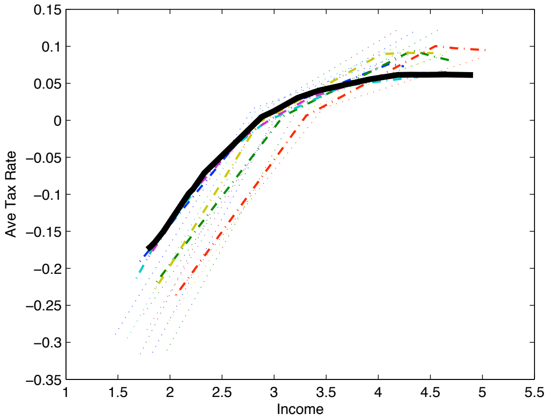

Figure 8: Income vs. Average Tax Rate for the 3D heterogeneity example.

where $U^{i,j,k}(c_{i,j,k}, y_{i,j,k})$ and $U^{i,j,k}(c_{p,q,r}, y_{p,q,r})$ are defined in (9.6). We choose parameters $na = nb = nc = 3$, $w_i \in \{2, 3, 4\}$, and $\lambda_{ijk} = 1/(na \times nb \times nc)$. We use the zero tax solution (c^*, y^*) as a starting point for the NLP solver SNOPT and compute solutions for $\eta_j \in \{1/2, 1, 2\}$ and $a_k \in \{0, 1, 2\}$.

Figures 7 and 8 illustrate the solution. The dotted lines are average tax rates if wage is only heterogeneity. Thin lines represent cases of 2D heterogeneity, both $w - \eta$ heterogeneity and $w - a$ heterogeneity. The solid line is one economy with 3D heterogeneity: taxpayers differing in wages, basic needs, and labor supply elasticity. The patterns are clear. The most redistributive economies are those where wage is the only heterogeneity. As we add heterogeneity in either a or η, redistribution is less. The final case where there is heterogeneity in wages, basic needs, and elasticity, has the least redistribution.

12

CONCLUSIONS

Here we examine models of optimal taxation in economies with multiple dimensions of heterogeneities. We analyze cases where individuals differ in productivity, elasticity of labor supply, basic needs, levels of distaste for work, and elasticity of demand for consumption. These examples show that many results from the basic 1D Mirrlees model no longer hold. In particular, we find cases of negative marginal tax rates for some high-productivity taxpayers. The examples also indicate that redistribution is less in economies with multidimensional heterogeneity, probably because income is a noisier signal of a taxpayer's type.

Multidimensional optimal tax problems are difficult nonlinear optimization problems because the linear independence constraint qualification does not hold at all feasible points and often fails to hold at the solution, as we have seen in our numerical examples. We found SNOPT was not able to solve larger examples. To solve these nonlinear programs robustly, we present a new approach, the NCL algorithm, developed in the next part of this thesis.

Part III

STABILIZED OPTIMIZATION VIA AN NCL ALGORITHM

13

INTRODUCTION

[1] For optimization problems involving many nonlinear inequality constraints, we extend the bound-constrained (BCL) and linearly constrained (LCL) augmented-Lagrangian approaches of LANCELOT and MINOS to an algorithm that solves a sequence of nonlinearly constrained augmented Lagrangian subproblems whose nonlinear constraints satisfy the LICQ everywhere. The NCL algorithm is implemented in AMPL and tested on large instances of a tax policy model that could not be solved directly by any of the state-of-the-art solvers that we tested due to degeneracy. Algorithm NCL with IPOPT as subproblem solver proves to be effective, with IPOPT achieving warm starts on each subproblem.

We consider constrained optimization problems of the form

$$\text{NCO} \qquad \begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) \geq 0, \quad Ax \geq b, \quad \ell \leq x \leq u, \end{array}$$

where $\phi(x)$ is a smooth nonlinear function, $c(x) \in \mathbb{R}^m$ is a vector of smooth nonlinear functions, and $Ax \geq b$ is a placeholder for a set of linear inequality or equality constraints, with x lying between lower and upper bounds ℓ and u.

In some applications where $m \gg n$, there may be more than n constraints that are essentially active at a solution. The constraints do not satisfy the linear independence constraint qualification (LICQ), and general-purpose solvers are likely to have difficulty converging. Some form of regularization is required. We achieve this by adapting the augmented Lagrangian algorithm of the general-purpose optimization solver LANCELOT (Conn, Gould, and Toint, 1991; Conn, Gould, and Toint, 1992; *LANCELOT optimization software* 1991) to derive a sequence of regularized subproblems denoted in the next chapter by NC$_k$.

[1] This essay is co-authored with Kenneth Judd, Dominique Orban and Michael Saunders, and published in *Numerical Analysis and Optimization (2018)*.

14

BCL, LCL, AND NCL METHODS

The theory for the large-scale solver LANCELOT is best described in terms of the general optimization problem

$$\boxed{\begin{array}{lll} \text{NECB} & \underset{x\in\mathbb{R}^n}{\text{minimize}} & \phi(x) \\ & \text{subject to} & c(x)=0, \quad \ell\le x\le u \end{array}}$$

with *nonlinear equality constraints* and bounds. We let x^* denote a local solution of NECB and (y^*, z^*) denote associated multipliers. LANCELOT treats NECB by solving a sequence of *bound-constrained subproblems* of the form

$$\boxed{\begin{array}{lll} \text{BC}_k & \underset{x}{\text{minimize}} & L(x,y_k,\rho_k)=\phi(x)-y_k^Tc(x)+\tfrac{1}{2}\rho_k\|c(x)\|^2 \\ & \text{subject to} & \ell\le x\le u, \end{array}}$$

where y_k is an estimate of the Lagrange multipliers y^* for the equality constraints. This was called a bound-constrained Lagrangian (BCL) method by Friedlander and Saunders (2005), in contrast to the LCL (linearly constrained Lagrangian) methods of Robinson (1972) and MINOS (Murtagh and Saunders, 1982), whose subproblems LC_k contain bounds as in BC_k and also linearizations of the equality constraints at the current point x_k (including linear constraints).

In order to treat NCO with a sequence of BC_k subproblems, we convert the nonlinear inequality constraints to equalities to obtain

$$\boxed{\begin{array}{lll} \text{NCO}' & \underset{x,s}{\text{minimize}} & \phi(x) \\ & \text{subject to} & c(x)-s=0, \quad Ax\ge b, \quad \ell\le x\le u, \quad s\ge 0 \end{array}}$$

with corresponding subproblems (including linear constraints)

$$\boxed{\begin{array}{lll} \text{BC}_k{}' & \underset{x,s}{\text{minimize}} & L(x,y_k,\rho_k)=\phi(x)-y_k^T(c(x)-s)+\tfrac{1}{2}\rho_k\|c(x)-s\|^2 \\ & \text{subject to} & Ax\ge b, \quad \ell\le x\le u, \quad s\ge 0. \end{array}}$$

We now introduce variables $r = -(c(x) - s)$ into $\mathrm{BC}_k{}'$ to obtain the *nonlinearly constrained Lagrangian* (NCL) subproblem

$$\boxed{\begin{array}{lll} \mathrm{NC}_k & \underset{x,r}{\text{minimize}} & \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k \|r\|^2 \\[6pt] & \text{subject to} & c(x) + r \geq 0, \quad Ax \geq b, \quad \ell \leq x \leq u, \end{array}}$$

in which r serves to make the nonlinear constraints independent. Assuming existence of finite multipliers and feasibility, for $\rho_k > 0$ and larger than a certain finite value, the NCL subproblems should cause y_k to approach y^* and most of the solution $(x_k^*, r_k^*, y_k^*, z_k^*)$ of NC_k to approach (x^*, y^*, z^*), with r_k^* approaching zero.

Problem NC_k is analogous to Friedlander and Orban's formulation for convex quadratic programs (Friedlander and Orban, 2012, Eq. (3.2)). See also Arreckx and Orban (2016), where the motivation is the same as here, achieving reliability when the nonlinear constraints don't satisfy LICQ.

Note that for general problems NECB, the BCL and LCL subproblems contain linear constraints (bounds only, or linearized constraints and bounds). Our NCL formulation retains nonlinear constraints in the NC_k subproblems, but simplifies them by ensuring that they satisfy LICQ. On large problems, the additional variables $r \in \mathbb{R}^m$ in NC_k may be detrimental to active-set solvers like MINOS or SNOPT (Gill, Murray, and Saunders, 2005a) because they increase the number of degrees of freedom (superbasic variables). Fortunately they are easily accommodated by interior methods, as our numerical results show for IPOPT (Wächter and Biegler, 2006; *IPOPT open source NLP solver* 2006) and for KNITRO (Byrd, Nocedal, and Waltz, 2006; *KNITRO optimization software* 2006).

We also show that the sequence of NCL sub-problems NC_k can be efficiently handled by warm-starting both IPOPT and KNITRO, in spite of the folklore that interior methods cannot be warm-started.

15

THE BCL ALGORITHM

The LANCELOT BCL method is summarized in Algorithm BCL. Each subproblem BC$_k$ is solved with a specified optimality tolerance ω_k, generating an iterate x_k^* and the associated Lagrangian gradient $z_k^* \equiv \nabla L(x_k^*, y_k, \rho_k)$. If $\|c(x_k^*)\|$ is sufficiently small, the iteration is regarded as "successful" and an update to y_k is computed from x_k^*. Otherwise, y_k is not altered but ρ_k is increased.

Key properties are that the subproblems are solved inexactly, the penalty parameter is increased only finitely often, and the multiplier estimates y_k need not be assumed bounded. Under certain conditions, all iterations are eventually successful, the ρ_k's remain constant, the iterates converge superlinearly, and the algorithm terminates in a finite number of iterations (Conn, Gould, and Toint, 1991).

Algorithm 1 BCL (Bound-Constrained Lagrangian Method for NECB)

```
 1: procedure BCL($x_0$, $y_0$, $z_0$)
 2:     Set penalty parameter $\rho_1 > 0$, scale factor $\tau > 1$, and constants $\alpha, \beta > 0$ with $\alpha < 1$.
 3:     Set positive convergence tolerances $\eta_*, \omega_* \ll 1$ and infeasibility tolerance $\eta_1 > \eta_*$.
 4:     $k \leftarrow 0$, converged $\leftarrow$ false
 5:     repeat
 6:         $k \leftarrow k + 1$
 7:         Choose optimality tolerance $\omega_k > 0$ such that $\lim_{k\to\infty} \omega_k \le \omega_*$.
 8:         Find $(x_k^*, z_k^*)$ that solves BC$_k$ to within $\omega_k$.
 9:         if $\|c(x_k^*)\| \le \max(\eta_*, \eta_k)$ then
10:             $y_k^* \leftarrow y_k - \rho_k c(x_k^*)$
11:             $x_k \leftarrow x_k^*,\ y_k \leftarrow y_k^*,\ z_k \leftarrow z_k^*$
12:             if $(x_k, y_k, z_k)$ solves NECB to within $\omega_*$, converged $\leftarrow$ true
13:             $\rho_{k+1} \leftarrow \rho_k$
14:             $\eta_{k+1} \leftarrow \eta_k / (1 + \rho_{k+1}^{\beta})$
15:         else
16:             $\rho_{k+1} \leftarrow \tau \rho_k$
17:             $\eta_{k+1} \leftarrow \eta_0 / (1 + \rho_{k+1}^{\alpha})$
18:         end if
19:     until converged
20:     $x^* \leftarrow x_k,\ y^* \leftarrow y_k,\ z^* \leftarrow z_k$
21: end procedure
```

Note that at step 8 of Algorithm BCL, the inexact minimization would be typically carried out from the initial guess (x_k^*, z_k^*). However, other initial points are possible. At step 12, we say that (x_k, y_k, z_k) solves NECB to within ω_* if the largest dual infeasibility is smaller than ω_*.

16

THE NCL ALGORITHM

To derive a stabilized algorithm for problem NCO, we modify Algorithm BCL by introducing r and replacing the subproblems BC_k by NC_k. The resulting method is summarized in Algorithm NCL. The update to y_k becomes $y_k^* \leftarrow y_k - \rho_k(c(x_k^*) - s_k^*) = y_k + \rho_k r_k^*$, the value satisfied by an optimal y_k^* for subproblem NC_k. Step 8 of Algorithm NCL would typically use $(x_k^*, r_k^*, y_k^*, z_k^*)$ as initial guess, and that is what we use in our implementation below.

Algorithm 2 NCL (Nonlinearly Constrained Lagrangian Method for NCO)

```
 1: procedure NCL($x_0$, $r_0$, $y_0$, $z_0$)
 2:     Set penalty parameter $\rho_1 > 0$, scale factor $\tau > 1$, and constants $\alpha, \beta > 0$ with $\alpha < 1$.
 3:     Set positive convergence tolerances $\eta_*, \omega_* \ll 1$ and infeasibility tolerance $\eta_1 > \eta_*$.
 4:     $k \leftarrow 0$, converged $\leftarrow$ false
 5:     repeat
 6:         $k \leftarrow k + 1$
 7:         Choose optimality tolerance $\omega_k > 0$ such that $\lim_{k\to\infty} \omega_k \le \omega_*$.
 8:         Find $(x_k^*, r_k^*, y_k^*, z_k^*)$ that solves $\mathrm{NC}_k$ to within $\omega_k$.
 9:         if $\|r_k^*\| \le \max(\eta_*, \eta_k)$ then
10:             $y_k^* \leftarrow y_k + \rho_k r_k^*$
11:             $x_k \leftarrow x_k^*,\ r_k \leftarrow r_k^*,\ y_k \leftarrow y_k^*,\ z_k \leftarrow z_k^*$
12:             if $(x_k, y_k, z_k)$ solves NCO to within $\omega_*$, converged $\leftarrow$ true
13:             $\rho_{k+1} \leftarrow \rho_k$
14:             $\eta_{k+1} \leftarrow \eta_k / (1 + \rho_{k+1}^{\beta})$
15:         else
16:             $\rho_{k+1} \leftarrow \tau \rho_k$
17:             $\eta_{k+1} \leftarrow \eta_0 / (1 + \rho_{k+1}^{\alpha})$
18:         end if
19:     until converged
20:     $x^* \leftarrow x_k,\ r^* \leftarrow r_k,\ y^* \leftarrow y_k,\ z^* \leftarrow z_k$
21: end procedure
```

16.1 an application: optimal tax policy

Some challenging test cases arise from the tax policy models described in Judd et al. (2017). With $x = (c, y)$, they take the form

$$\boxed{\begin{array}{llll} \text{TAX} & \underset{c,y}{\text{maximize}} & \sum_i \lambda_i U^i(c_i, y_i) & \\ & \text{subject to} & U^i(c_i, y_i) - U^i(c_j, y_j) \geq 0 & \text{for all } i, j \\ & & \lambda^T(y - c) \geq 0 & \\ & & c,\ y \geq 0, & \end{array}}$$

where c_i and y_i are the consumption and income of taxpayer i, and λ is a vector of positive weights. The utility functions $U^i(c_i, y_i)$ are each of the form

$$U(c, y) = \frac{(c - \alpha)^{1-1/\gamma}}{1 - 1/\gamma} - \psi \frac{(y/w)^{1/\eta+1}}{1/\eta + 1},$$

where w is the wage rate and α, γ, ψ and η are taxpayer heterogeneities. More precisely, the utility functions are of the form

$$U^{i,j,k,g,h}(c_{p,q,r,s,t}, y_{p,q,r,s,t}) = \frac{(c_{p,q,r,s,t} - \alpha_k)^{1-1/\gamma_h}}{1 - 1/\gamma_h} - \psi_g \frac{(y_{p,q,r,s,t}/w_i)^{1/\eta_j+1}}{1/\eta_j + 1},$$

where (i, j, k, g, h) and (p, q, r, s, t) run over na wage types, nb elasticities of labor supply, nc basic need types, nd levels of distaste for work, and ne elasticities of demand for consumption, with na, nb, nc, nd, ne determining the size of the problem, namely $m = T(T-1)$ nonlinear constraints, $n = 2T$ variables, with $T := na \times nb \times nc \times nd \times ne$.

16.2 results for ncl/ipopt

Table 22 summarizes results for a 4D example ($ne = 1$ and $\gamma_1 = 1$). The first term of $U(c, y)$ becomes $\log(c - \alpha)$, the limit as $\gamma \to 1$. Problem NCO and Algorithm NCL were formulated in the AMPL modeling language (Fourer, Gay, and Kernighan, 2002). The solvers SNOPT (Gill, Murray, and Saunders, 2005a) and IPOPT (Wächter and Biegler, 2006) were unable to solve NCO itself, but Algorithm NCL was successful with IPOPT solving the subproblems NC$_k$. We use a default configuration of IPOPT with MUMPS (Amestoy et al., 2001) as symmetric indefinite solver to compute search directions. We set the optimality tolerance

Table 22: NCL results on a 4D example with $na, nb, nc, nd = 11, 3, 3, 2$, giving $m = 39006$, $n = 395$. Itns refers to IPOPT's primal-dual interior point method, and Time is seconds on an Apple iMac with 2.93 GHz Intel Core i7.

k	ρ_k	η_k	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	Itns	Time
1	10^2	10^{-2}	3.1e-03	-2.1478532e+01	125	42.8
2	10^2	10^{-3}	1.3e-03	-2.1277587e+01	18	6.5
3	10^3	10^{-3}	6.6e-04	-2.1177152e+01	27	9.1
4	10^3	10^{-4}	5.5e-04	-2.1110210e+01	31	10.8
5	10^4	10^{-4}	2.9e-04	-2.1066664e+01	57	24.3
6	10^5	10^{-4}	6.5e-05	-2.1027152e+01	75	26.8
7	10^5	10^{-5}	5.2e-05	-2.1018896e+01	130	60.9
8	10^6	10^{-5}	9.3e-06	-2.1015295e+01	159	81.8
9	10^6	10^{-6}	2.0e-06	-2.1014808e+01	139	70.0
10	10^7	10^{-6}	2.1e-07	-2.1014800e+01	177	97.6

for IPOPT to $\omega_k = \omega_* = 10^{-6}$ throughout, and specified warm starts for $k \geq 2$ using options warm_start_init_point=yes and mu_init=1e-4. These options greatly improved the performance of IPOPT on each subproblem compared to cold starts, for which mu_init=0.1. It is helpful that only the objective function of NC$_k$ changes with k.

For this example, problem NCO has $m = 39007$ nonlinear inequality constraints and one linear constraint in $n = 396$ variables $x = (c, y)$, and nonnegativity bounds. Subproblem NC$_k$ has 39007 constraints and 39402 variables when r is included. Fortunately r does not affect the complexity of each IPOPT iteration, but greatly improves stability. In contrast, active-set methods like MINOS and SNOPT are very inefficient on the NC$_k$ subproblems because the large number of inequality constraints leads to thousands of minor iterations, and the presence of r (with no bounds) leads to thousands of superbasic variables. About $3.2n$ constraints were within 10^{-6} of being active.

Table 23 summarizes results for a 5D example. The NC$_k$ subproblems have $m = 32220$ nonlinear constraints and $n = 360$ variables, leading to 32581 variables including r. Again the options warm_start_init_point=yes and mu_init=1e-4 for $k \geq 2$ led to good performance by IPOPT on each subproblem. About $3n$ constraints were within 10^{-6} of being active.

For much larger problems of this type, we found that it was helpful to reduce mu_init more often, as illustrated in Table 24. The NC$_k$ subproblems here have $m = 570780$ nonlinear constraints and $n = 1512$ variables, leading to 572292 vari-

Table 23: NCL results on a 5D example with $na, nb, nc, nd, ne = 5, 3, 3, 2, 2$, giving $m = 32220$, $n = 360$.

k	ρ_k	η_k	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	Itns	Time
1	10^2	10^{-2}	7.0e-03	-4.2038075e+02	95	41.1
2	10^2	10^{-3}	4.1e-03	-4.2002898e+02	17	7.2
3	10^3	10^{-3}	1.3e-03	-4.1986069e+02	20	8.1
4	10^4	10^{-3}	4.4e-04	-4.1972958e+02	48	25.0
5	10^4	10^{-4}	2.2e-04	-4.1968646e+02	43	20.5
6	10^5	10^{-4}	9.8e-05	-4.1967560e+02	64	32.9
7	10^5	10^{-5}	6.6e-05	-4.1967177e+02	57	26.8
8	10^6	10^{-5}	4.2e-06	-4.1967150e+02	87	46.2
9	10^6	10^{-6}	9.4e-07	-4.1967138e+02	96	53.6

Table 24: NCL results on a 5D example with $na, nb, nc, ne, ne = 21, 3, 3, 2, 2$, giving $m = 570780$, $n = 1512$.

k	ρ_k	η_k	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	mu_init	Itns	Time
1	10^2	10^{-2}	5.1e-03	-1.7656816e+03	10^{-1}	825	7763.3
2	10^2	10^{-3}	2.4e-03	-1.7648480e+03	10^{-4}	66	472.8
3	10^3	10^{-3}	1.3e-03	-1.7644006e+03	10^{-4}	106	771.3
4	10^4	10^{-3}	3.8e-04	-1.7639491e+03	10^{-5}	132	1347.0
5	10^4	10^{-4}	3.2e-04	-1.7637742e+03	10^{-5}	229	2450.9
6	10^5	10^{-4}	8.6e-05	-1.7636804e+03	10^{-6}	104	1096.9
7	10^5	10^{-5}	4.9e-05	-1.7636469e+03	10^{-6}	143	1633.4
8	10^6	10^{-5}	1.5e-05	-1.7636252e+03	10^{-7}	71	786.1
9	10^7	10^{-5}	2.8e-06	-1.7636196e+03	10^{-7}	67	725.7
10	10^7	10^{-6}	5.1e-07	-1.7636187e+03	10^{-8}	18	171.0

ables including r. Note that the number of NCL iterations is stable ($k \leq 10$), and IPOPT performs well on each subproblem with decreasing mu_init. This time about $6.6n$ constraints were within 10^{-6} of being active.

Note that the LANCELOT approach allows early subproblems to be solved less accurately (Conn, Gould, and Toint, 1991). It may save time to set $\omega_k = \eta_k$ (say) rather than $\omega_k = \omega_*$ throughout.

16.3 results for knitro and ncl/knitro

Like IPOPT, KNITRO is an interior-point solver for linear and nonlinear optimization. Here we investigate the performance of both solvers on problems of increasing size. We applied IPOPT and KNITRO to the original problems and to the sequence of subproblems generated by algorithm NCL. As test problems, we used the tax policy models with increasing values of na but fixed values $nb = 3$, $nc = 3$, $nd = 2$, $ne = 2$.

Table 25 shows that IPOPT by itself could solve only the smallest problem. KNITRO was reliable on larger problems, but the solve time increased rather rapidly. With warm starts for most NCL iterations, NCL/IPOPT performed well. With cold starts for each NCL iteration, NCL/KNITRO was reliable but extremely slow.

Table 26 shows again that KNITRO was reliable but rather slow on larger problems, but with warm starts for most NCL iterations, NCL/KNITRO performed extremely well.

We conclude that, contrary to the folklore of interior methods, IPOPT and KNITRO can be reliably warm-started on a sequence of related problems such as those arising in algorithm NCL. The computational savings are highly significant.

Table 25: Comparison of IPOPT, KNITRO, NCL. The problem size increases with na. For IPOPT, * means the dual variables diverged and the linear solver MUMPS kept requesting more memory. ! means IPOPT went into a loop. Cold starts were used everywhere except for NCL/IPOPT, which reduced the initial barrier parameter μ for NCL iterations 2, 4, 6, 8 as coded in section 18.4.

na = increasing $nb = 3$ $nc = 3$ $nd = 2$ $ne = 2$

			IPOPT		KNITRO		NCL/IPOPT		NCL/KNITRO	
na	m	n	itns	time	itns	time	itns	time	itns	time
5	32220	360	449	217	168	53	322	146	2320	8.0mins
9	104652	648	> 98*	> 360*	928	825	655	1023	9697	1.9hrs
11	156420	792	> 87*	∞!	2769	4117	727	1679	26397	7.0hrs
17	373933	1224			2598	11447	1021	6347		
21	570780	1512					1761	17218	45039	1.9 days
							Warm starts		Cold starts	

Table 26: NCL/KNITRO with Warm Starts. The same as Table 25 except the last column, where NCL/KNITRO used warm-start options for NCL iterations 2, 4, 6, 8 as coded in section 18.5.

na = increasing $nb = 3$ $nc = 3$ $nd = 2$ $ne = 2$

			IPOPT		KNITRO		NCL/IPOPT		NCL/KNITRO	
na	m	n	itns	time	itns	time	itns	time	itns	time
5	32220	360	449	217	168	53	322	146	339	63
9	104652	648	> 98*	> 360*	928	825	655	1023	307	239
11	156420	792	> 87*	∞!	2769	4117	727	1679	383	420
17	373933	1224			2598	11447	1021	6347	486	1200
21	570780	1512					1761	17218	712	2880
							Warm starts		Warm starts	

17

CONCLUSIONS

This work has been illuminating in several ways as we sought to improve our ability to solve examples of problem TAX.

- Small examples of the tax model solve efficiently with MINOS and SNOPT, but eventually fail to converge as the problem size increases.

- IPOPT also solves small examples efficiently, but eventually starts requesting additional memory for the MUMPS sparse linear solver. The solver may freeze, or the iterations may diverge.

- The NC_k subproblems are not suitable for MINOS or SNOPT because of the large number of variables (x, r) and the resulting number of superbasic variables (although warm-starts are natural).

- It is often said that interior methods cannot be warm-started. Nevertheless, IPOPT has several runtime options that have proved to be extremely helpful for implementing Algorithm NCL. For the results obtained here, it has been sufficient to say that warm starts are wanted for $k > 1$, and that the IPOPT barrier parameter should be initialized at decreasing values for later k (where only the objective of subproblem NC_k changes with k). Analogous runtime options were determined for KNITRO.

- The numerical examples of section 16.1 had $3n$, $3n$ and $6.6n$ constraints essentially active at the solution, yet were solved successfully. They suggest that the NCL approach with an interior method as subproblem solver can overcome LICQ difficulties on problems that could not be solved directly.

18

Appendix AMPL Models, Data, and Scripts

Algorithm NCL has been implemented in the AMPL modeling language (Fourer, Gay, and Kernighan, 2002) and tested on problem TAX. The following sections list each relevant file. The files are available from *NCL*.

18.1 Tax model

File `pTax5Dncl.mod` codes subproblem NC_k for problem TAX with five parameters w, η, α, ψ, γ, using $\mu := 1/\eta$. Note that for $U(c, y)$ in the objective and constraint functions, the first term $(c-\alpha)^{1-1/\gamma}/(1-1/\gamma)$ is replaced by a piecewise-smooth function that is defined for all values of c and α (see Judd et al., 2017).

Primal regularization $\frac{1}{2}\delta\|(c, y)\|^2$ with $\delta = 10^{-8}$ is added to the objective function to promote uniqueness of the minimizer. The vector r is called `R` to avoid a clash with subscript `r`.

```
# pTax5Dncl.mod

# Define parameters for agents (taxpayers)
param na;                # number of types in wage
param nb;                # number of types in eta
param nc;                # number of types in alpha
param nd;                # number of types in psi
param ne;                # number of types in gamma
set A := 1..na;          # set of wages
set B := 1..nb;          # set of eta
set C := 1..nc;          # set of alpha
set D := 1..nd;          # set of psi
set E := 1..ne;          # set of gamma
set T = {A,B,C,D,E};     # set of agents

# Define wages for agents (taxpayers)
param wmin;              # minimum wage level
param wmax;              # maximum wage level
param w {A};             # i, wage vector
param mu{B};             # j, mu = 1/eta# mu vector
param mu1{B};            # mu1[j] = mu[j] + 1
param alpha{C};          # k, ak vector for utility
param psi{D};            # g
param gamma{E};          # h
param lambda{A,B,C,D,E}; # distribution density
param epsilon;
param primreg     default 1e-8;  # Small primal regularization

var c{(i,j,k,g,h) in T} >= 0.1;  # consumption for tax payer (i,j,k,g,h)
var y{(i,j,k,g,h) in T} >= 0.1;  # income      for tax payer (i,j,k,g,h)
var R{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} >= -1e+20, <= 1e+20;

param kmax      default 20;         # limit on NCL itns
param rhok      default 1e+2;       # augmented Lagrangian penalty parameter
param rhofac    default 10.0;       # increase factor
param rhomax    default 1e+8;       # biggest rhok
param etak      default 1e-2;       # opttol for augmented Lagrangian loop
param etafac    default  0.1;       # reduction factor for opttol
param etamin    default 1e-8;       # smallest etak
param rmax      default    0;       # max r (for printing)
param rmin      default    0;       # min r (for printing)
param rnorm     default    0;       # ||r||_inf
param rtol      default 1e-6;       # quit if biggest |r_i| <= rtol

param nT        default    1;       # nT = na*nb*nc*nd*ne
param m         default    1;       # nT*(nT-1) = no. of nonlinear constraints
param n         default    1;       # 2*nT      = no. of nonlinear variables

param ck{(i,j,k,g,h) in T} default 0;        # current variable c
param yk{(i,j,k,g,h) in T} default 0;        # current variable y
param rk{(i,j,k,g,h) in T, (p,q,r,s,t) in T: # current variable r = - (c(x) - s)
   !(i=p and j=q and k=r and g=s and h=t)} default 0;
param dk{(i,j,k,g,h) in T, (p,q,r,s,t) in T: # current dual variables (y_k)
   !(i=p and j=q and k=r and g=s and h=t)} default 0;

minimize f:
   sum{(i,j,k,g,h) in T}
   (
      (if c[i,j,k,g,h] - alpha[k] >= epsilon then
          - lambda[i,j,k,g,h] *
               ((c[i,j,k,g,h] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
                - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j])
       else
          - lambda[i,j,k,g,h] *
         (-   0.5/gamma[h] * epsilon^(-1/gamma[h]-1) * (c[i,j,k,g,h] - alpha[k])^2
          + ( 1+1/gamma[h])* epsilon^(-1/gamma[h]  ) * (c[i,j,k,g,h] - alpha[k])
          + (1/(1-1/gamma[h]) - 1 - 0.5/gamma[h]) * epsilon^(1-1/gamma[h])
                - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j])
      )
   + 0.5 * primreg * (c[i,j,k,g,h]^2 + y[i,j,k,g,h]^2)
   )
 + sum{(i,j,k,g,h) in T, (p,q,r,s,t) in T: !(i=p and j=q and k=r and g=s and h=t)}
       (dk[i,j,k,g,h,p,q,r,s,t] * R[i,j,k,g,h,p,q,r,s,t]
                   + 0.5 * rhok * R[i,j,k,g,h,p,q,r,s,t]^2);

subject to

Incentive{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
          !(i=p and j=q and k=r and g=s and h=t)}:
   (if c[i,j,k,g,h] - alpha[k] >= epsilon then
      (c[i,j,k,g,h] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
       - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
    else
       -  0.5/gamma[h] *epsilon^(-1/gamma[h]-1)*(c[i,j,k,g,h] - alpha[k])^2
       + (1+1/gamma[h])*epsilon^(-1/gamma[h]  )*(c[i,j,k,g,h] - alpha[k])
       + (1/(1-1/gamma[h]) - 1 - 0.5/gamma[h])*epsilon^(1-1/gamma[h])
       - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
   )
 - (if c[p,q,r,s,t] - alpha[k] >= epsilon then
      (c[p,q,r,s,t] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
       - psi[g]*(y[p,q,r,s,t]/w[i])^mu1[j] / mu1[j]
    else
       -  0.5/gamma[h] *epsilon^(-1/gamma[h]-1)*(c[p,q,r,s,t] - alpha[k])^2
       + (1+1/gamma[h])*epsilon^(-1/gamma[h]  )*(c[p,q,r,s,t] - alpha[k])
       + (1/(1-1/gamma[h]) - 1 - 0.5/gamma[h])*epsilon^(1-1/gamma[h])
       - psi[g]*(y[p,q,r,s,t]/w[i])^mu1[j] / mu1[j]
   )
 + R[i,j,k,g,h,p,q,r,s,t] >= 0;

Technology:
   sum{(i,j,k,g,h) in T} lambda[i,j,k,g,h]*(y[i,j,k,g,h] - c[i,j,k,g,h]) >= 0;
```

18.2 tax model data

File `pTax5Dncl.dat` provides data for a specific problem.

```
# pTax5Dncl.dat

data;
```

```
let na := 5;
let nb := 3;
let nc := 3;
let nd := 2;
let ne := 2;

# Set up wage dimension intervals
let wmin := 2;
let wmax := 4;
let {i in A}  w[i]   := wmin + ((wmax-wmin)/(na-1))*(i-1);

data;

param  mu :=
    1   0.5
    2   1
    3   2 ;

# Define mu1
let {j in B} mu1[j] := mu[j] + 1;

data;

param alpha :=
    1   0
    2   1
    3   1.5;

param psi :=
    1   1
    2   1.5;

param gamma :=
    1   2
    2   3;

# Set up 5 dimensional distribution
let {(i,j,k,g,h) in T} lambda[i,j,k,g,h] := 1;

# Choose a reasonable epsilon
let epsilon := 0.1;
```

18.3 initial values

File `pTax5Dinitial.run` solves a simplified model to compute starting values for Algorithm NCL. The nonlinear inequality constraints are removed, and $y = c$ is enforced. This model solves easily with MINOS or SNOPT on all cases tried. Solution values are output to file `p5Dinitial.dat`.

```
# pTax5Dinitial.run

# Define parameters for agents (taxpayers)
param na := 5;        # number of types in wage
param nb := 3;         # number of types in eta
param nc := 3;         # number of types in alpha
param nd := 2;         # number of types in psi
param ne := 2;         # number of types in gamma
set A := 1..na;        # set of wages
set B := 1..nb;        # set of eta
set C := 1..nc;        # set of alpha
set D := 1..nd;      # set of psi
set E := 1..ne;        # set of gamma
set T = {A,B,C,D,E};     # set of agents

# Define wages for agents (taxpayers)
param  wmin := 2;         # minimum wage level
param  wmax := 4;         # maximum wage level
param  w {i in A} := wmin + ((wmax-wmin)/(na-1))*(i-1);  # wage vector

# Choose a reasonable epsilon
param epsilon := 0.1;

# mu vector
param mu {B};             # mu = 1/eta
param mu1{B};             # mu1[j] = mu[j] + 1
param alpha {C};
param gamma {E};
param psi {D};

var c {(i,j,k,g,h) in T} >= 0.1;
var y {(i,j,k,g,h) in T} >= 0.1;

maximize f: sum{(i,j,k,g,h) in T}
   if c[i,j,k,g,h] - alpha[k] >= epsilon then
     (c[i,j,k,g,h] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
      -  psi[g] * (y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
   else
      -  0.5/gamma[h] *epsilon^(-1/gamma[h]-1)*(c[i,j,k,g,h] - alpha[k])^2
      + (1+1/gamma[h])*epsilon^(-1/gamma[h])  *(c[i,j,k,g,h] - alpha[k])
      + (1/(1-1/gamma[h]) -1 - 0.5/gamma[h])*epsilon^(1-1/gamma[h])
      -  psi[g] * (y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j];

subject to
   Budget {(i,j,k,g,h) in T}: y[i,j,k,g,h] - c[i,j,k,g,h] = 0;

let {(i,j,k,g,h) in T} y[i,j,k,g,h] := i+1;
let {(i,j,k,g,h) in T} c[i,j,k,g,h] := i+1;

data;

param  mu :=
    1   0.5
    2   1
    3   2 ;

# Define mu1
let {j in B} mu1[j] := mu[j] + 1;

data;

param alpha :=
    1   0
    2   1
    3   1.5;

param psi :=
    1   1
    2   1.5;

param gamma :=
    1   2
    2   3;

option solver snopt;
option show_stats 1;

option snopt_options  ’ \
   summary_file=6       \
   print_file=9         \
   scale=no             \
   print_level=0        \
   major_iterations=2000\
   iterations=50000     \
   optimality_tol=1e-7  \
  *penalty=100.0        \
   superbasics_limit=3000\
   solution=yes         \
  *verify_level=3       \
';


display na,nb,nc,nd,ne;
solve;
display na,nb,nc,nd,ne;
display y,c >p5Dinitial.dat;
close p5Dinitial.dat;
```

18.4 ncl implementation with ipopt

File `pTax5Dnclipopt.run` uses files

```
pTax5Dinitial.run
pTax5Dncl.mod
pTax5Dncl.dat
pTax5Dinitial.dat
```

to implement Algorithm NCL. Subproblems NC$_k$ are solved in a loop until $\|r_k^*\|_\infty \le$ `rtol = 1e-6`, or η_k has been reduced to parameter `etamin = 1e-8`, or ρ_k has been increased to parameter `rhomax = 1e+8`. The loop variable k is called `K` to avoid a clash with subscript `k` in the model file.

Optimality tolerance $\omega_k = 10^{-6}$ is used throughout to ensure that the solution of the final subproblem NC$_k$ will be close to a solution of the original problem if $\|r_k^*\|_\infty$ is small enough for the final k ($\|r_k^*\|_\infty \le$ `rtol = 1e-6`).

IPOPT is used to solve each subproblem NC$_k$, with runtime options set to implement increasingly warm starts.

```
# pTax5Dnclipopt.run

reset;
model pTax5Dinitial.run;
reset;
model pTax5Dncl.mod;
data  pTax5Dncl.dat;
data; var include p5Dinitial.dat;
```

```
model;
option solver ipopt;
option show_stats 1;

option ipopt_options ’\
 dual_inf_tol=1e-6    \
 max_iter=5000        \
’;

option opt2 $ipopt_options ’ warm_start_init_point=yes’;

# NCL method.
# kmax, rhok, rhofac, rhomax, etak, etafac, etamin, rtol
# are defined in the .mod file.

printf "NCLipopt log for pTax5D\n" > 5DNCLipopt.log;
display na, nb, nc, nd, ne, primreg  > 5DNCLipopt.log;
printf "   k      rhok      etak     rnorm        Obj\n" > 5DNCLipopt.log;

for {K in 1..kmax}
{  display na, nb, nc, nd, ne, primreg, K, kmax, rhok, etak;
   if K == 2 then {option ipopt_options $opt2 ’ mu_init=1e-4’};
   if K == 4 then {option ipopt_options $opt2 ’ mu_init=1e-5’};
   if K == 6 then {option ipopt_options $opt2 ’ mu_init=1e-6’};
   if K == 8 then {option ipopt_options $opt2 ’ mu_init=1e-7’};
   if K ==10 then {option ipopt_options $opt2 ’ mu_init=1e-8’};

   display $ipopt_options;
   solve;

   let rmax := max({(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} R[i,j,k,g,h,p,q,r,s,t]);
   let rmin := min({(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} R[i,j,k,g,h,p,q,r,s,t]);
   display na, nb, nc, nd, ne, primreg, K, rhok, etak, kmax;
   display K, kmax, rmax, rmin;
   let rnorm := max(abs(rmax), abs(rmin));   # ||r||_inf

   printf "%4i %9.1e %9.1e %9.1e %15.7e\n", K, rhok, etak, rnorm, f >> 5DNCLipopt.log;
   close 5DNCLipopt.log;

   if rnorm <= rtol then
   { printf "Stopping: rnorm is small\n"; display K, rnorm; break; }

   if rnorm <= etak then # update dual estimate dk; save new solution
   {let {(i,j,k,g,h) in T, (p,q,r,s,t) in T:
         !(i=p and j=q and k=r and g=s and h=t)}
            dk[i,j,k,g,h,p,q,r,s,t] :=
            dk[i,j,k,g,h,p,q,r,s,t] + rhok*R[i,j,k,g,h,p,q,r,s,t];
    let {(i,j,k,g,h) in T} ck[i,j,k,g,h] := c[i,j,k,g,h];
    let {(i,j,k,g,h) in T} yk[i,j,k,g,h] := y[i,j,k,g,h];
    display K, etak;
    if  etak == etamin then { printf "Stopping: etak = etamin\n"; break; }
    let etak := max(etak*etafac, etamin);
    display etak;
   }
   else # keep previous solution; increase rhok
   { let {(i,j,k,g,h) in T} c[i,j,k,g,h] := ck[i,j,k,g,h];
     let {(i,j,k,g,h) in T} y[i,j,k,g,h] := yk[i,j,k,g,h];
     display K, rhok;
     if  rhok == rhomax then { printf "Stopping: rhok = rhomax\n"; break; }
     let rhok := min(rhok*rhofac, rhomax);
     display rhok;
   }
}

display c,y;  display na, nb, nc, nd, ne, primreg, rhok, etak, rnorm;

# Count how many constraint are close to being active.
data;
let nT   := na*nb*nc*nd*ne;   let m := nT*(nT-1);   let n := 2*nT;
let etak := 1.0001e-10;
printf "\n m = %8i\n n = %8i\n", m, n >> 5DNCLipopt.log;
printf "\n Constraints within tol of being active\n\n" >> 5DNCLipopt.log;
printf "    tol      count    count/n\n" >> 5DNCLipopt.log;

for {K in 1..10}
{
 let kmax := card{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
                  !(i=p and j=q and k=r and g=s and h=t)
                  and Incentive[i,j,k,g,h,p,q,r,s,t].slack <= etak};
 printf "%9.1e %8i %8.1f\n", etak, kmax, kmax/n >> 5DNCLipopt.log;
 let etak := etak*10;
}
printf "Created 5DNCLipopt.log\n";
```

18.5 ncl implementation with knitro

Here is an analogous file for solving the NC$_k$ subproblems with KNITRO, with increasingly warm starts every second subproblem.[1]

```
# pTax5Dnclknitro5.run

reset;
commands pTax5Dinitial.run;
reset;
model pTax5Dncl.mod;
data  pTax5Dncl.dat;

param initialDatFile symbolic := "p5Dinitial_" & sprintf("%i", na) & ".dat";

data; commands (initialDatFile);

model;
option solver knitro;
option show_stats 1;

param logFile symbolic := "5DNCLknitro_" & sprintf("%i", na) & ".log";
param finalMsg symbolic := sprintf("Created %s", logFile);

option knitro_options ’\
    algorithm=1       \
    bar_directinterval=0 \
    bar_initpt=2      \
    maxit=90000       \
    feastol=1e-6      \
    opttol=1e-6       \
    outlev=3          \
’;

#   cg_maxit=0        \
#   algorithm=1       \   # Use the Interior/Direct algorithm
#   algorithm=5       \   # Run all algorithms, possibly in parallel
#   bar_feasible=1    \   # Stay satisfying inequality constraints


#option opt2  $ipopt_options ’ warm_start_init_point=yes’;
 option opt2 $knitro_options ’ bar_murule=1’;
```

[1] We thank Richard Waltz for his help in choosing runtime options for KNITRO.

```
# NCL method.
# kmax, rhok, rhofac, rhomax, etak, etafac, etamin, rtol
# are defined in the .mod file.

printf "NCLknitro log for pTax5D\n" > (logFile);
display na, nb, nc, nd, ne, primreg > (logFile);
printf "   k      rhok      etak     rnorm        Obj\n" > (logFile);

for {K in 1..kmax}
{  display na, nb, nc, nd, ne, primreg, K, kmax, rhok, etak;
   if K == 2 then {option knitro_options $opt2 ’ bar_initmu=1e-4 bar_slackboundpush=1e-4’};
   if K == 4 then {option knitro_options $opt2 ’ bar_initmu=1e-5 bar_slackboundpush=1e-5’};
   if K == 6 then {option knitro_options $opt2 ’ bar_initmu=1e-6 bar_slackboundpush=1e-6’};
   if K == 8 then {option knitro_options $opt2 ’ bar_initmu=1e-7 bar_slackboundpush=1e-7’};
   if K ==10 then {option knitro_options $opt2 ’ bar_initmu=1e-8 bar_slackboundpush=1e-8’};

   display $knitro_options;
   solve;

   let rmax := max({(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} R[i,j,k,g,h,p,q,r,s,t]);
   let rmin := min({(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} R[i,j,k,g,h,p,q,r,s,t]);
   display na, nb, nc, nd, ne, primreg, K, rhok, etak, kmax;
   display K, kmax, rmax, rmin;
   let rnorm := max(abs(rmax), abs(rmin));   # ||r||_inf

   printf "%4i %9.1e %9.1e %9.1e %15.7e\n", K, rhok, etak, rnorm, f >> (logFile);
   close (logFile);

   if rnorm <= rtol then
   { printf "Stopping: rnorm is small\n"; display K, rnorm; break; }

   if rnorm <= etak then # update dual estimate dk; save new solution
   {let {(i,j,k,g,h) in T, (p,q,r,s,t) in T:
         !(i=p and j=q and k=r and g=s and h=t)}
            dk[i,j,k,g,h,p,q,r,s,t] :=
            dk[i,j,k,g,h,p,q,r,s,t] + rhok*R[i,j,k,g,h,p,q,r,s,t];
    let {(i,j,k,g,h) in T} ck[i,j,k,g,h] := c[i,j,k,g,h];
    let {(i,j,k,g,h) in T} yk[i,j,k,g,h] := y[i,j,k,g,h];
    display K, etak;
    if  etak == etamin then { printf "Stopping: etak = etamin\n"; break; }
    let etak := max(etak*etafac, etamin);
    display etak;
   }
   else # keep previous solution; increase rhok
   { let {(i,j,k,g,h) in T} c[i,j,k,g,h] := ck[i,j,k,g,h];
     let {(i,j,k,g,h) in T} y[i,j,k,g,h] := yk[i,j,k,g,h];
     display K, rhok;
     if  rhok == rhomax then { printf "Stopping: rhok = rhomax\n"; break; }
     let rhok := min(rhok*rhofac, rhomax);
     display rhok;
   }
}

display c,y;  display na, nb, nc, nd, ne, primreg, rhok, etak, rnorm;

# Count how many constraint are close to being active.
data;
let nT   := na*nb*nc*nd*ne;   let m := nT*(nT-1);   let n := 2*nT;
let etak := 1.0001e-10;
printf "\n m = %8i\n n = %8i\n", m, n >> (logFile);
printf "\n Constraints within tol of being active\n\n" >> (logFile);
printf "    tol      count    count/n\n" >> (logFile);

for {K in 1..10}
{
 let kmax := card{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
                  !(i=p and j=q and k=r and g=s and h=t)
                  and Incentive[i,j,k,g,h,p,q,r,s,t].slack <= etak};
 printf "%9.1e %8i %8.1f\n", etak, kmax, kmax/n >> (logFile);
 let etak := etak*10;
}
printf "total time spent in solver: %f\n", _total_solve_time >> (logFile);
printf "%s\n", finalMsg;
```

Part IV

RELIABLE AND EFFICIENT SOLUTION OF GENOME-SCALE MODELS OF METABOLISM AND MACROMOLECULAR EXPRESSION

19

INTRODUCTION

[1] Constraint-Based Reconstruction and Analysis is currently the only methodology that permits integrated modeling of Metabolism and macromolecular Expression (ME) at genome-scale. Linear optimization computes steady-state flux solutions to ME models, but flux values are spread over many orders of magnitude. Data values also have greatly varying magnitudes. Standard double-precision solvers may return inaccurate solutions or report that no solution exists. Exact simplex solvers based on rational arithmetic require a near-optimal warm start to be practical on large problems (current ME models have 70,000 constraints and variables and will grow larger). We have developed a quadruple-precision version of our linear and nonlinear optimizer MINOS, and a solution procedure (DQQ) involving Double and Quad MINOS that achieves reliability and efficiency for ME models and other challenging problems tested here. DQQ will enable extensive use of large linear and nonlinear models in systems biology and other applications involving multiscale data.

Constraint-Based Reconstruction and Analysis (COBRA) (Palsson, 2006) has been applied successfully to predict phenotypes for a range of genome-scale biochemical processes. The popularity of COBRA is partly due to the efficiency of the underlying optimization algorithms, permitting genome-scale modeling at a particular timescale using readily available open source software (Schellenberger et al., 2011; Ebrahim et al., 2013) and industrial quality optimization algorithms (*Gurobi optimization system for linear and integer programming* 2014; *IBM ILOG CPLEX optimizer* 2014; *MOSEK Optimization Software* 2014). A widespread application of COBRA is the modeling of steady states in genome-scale Metabolic models (M models). COBRA has also been used to model steady states in macromolecular Expression networks (E models), which stoichiometrically represent the transcription, translation, post-translational modification and formation of all protein complexes required for macromolecular biosynthesis and metabolic reaction catalysis (Thiele et al., 2009; Thiele et al., 2011). COBRA of metabolic networks or expression networks depends on numerical optimization algorithms to compute solutions to certain model equations, or to determine that no solution exists. Our

[1] This essay is co-authored with Laurence Yang, Ronan Fleming, Ines Thiele, Bernhard Palsson, and Michael Saunders, and published in *Scientific Reports*.

purpose is to discuss available options and to demonstrate an approach that is reliable and efficient for ever larger networks.

Metabolism and macromolecular Expression (ME) models have opened a whole new vista for predictive mechanistic modeling of cellular processes, but their size and multiscale nature pose a challenge to standard linear optimization (LO) solvers based on 16-digit double-precision floating-point arithmetic. Standard LO solvers usually apply scaling techniques (Fourer, 1982; Tomlin, 1975) to problems that are not already well scaled. The scaled problem typically solves more efficiently and accurately, but the solver must then unscale the solution, and this may generate significant primal or dual infeasibilities in the original problem (the constraints or optimality conditions may not be accurately satisfied).

A *lifting* approach (Sun et al., 2013) has been implemented to alleviate this difficulty with multiscale problems. Lifting reduces the largest matrix entries by introducing auxiliary constraints and variables. This approach has permitted standard (double-precision) LO solvers to find more accurate solutions, even though the final objective value is still not satisfactory. Another approach to increasing the precision is to use an exact solver (Dhiflaoui et al., 2003). An exact simplex solver QSopt_ex (Applegate et al., 2007; Applegate et al., 2008) has been used for a ME model of *Thermotoga maritima* (Lerman et al., 2012) (model TMA_ME) representing a network with about 18,000 metabolites and reactions. The solution time was about two weeks, compared to a few minutes for a standard double-precision solver, but the latter's final objective value had only one correct digit. QSopt_ex has since been applied to a collection of 98 metabolic models by Chindelvitch et al. (Chindelevitch et al., 2014) via their MONGOOSE toolbox. Most of the 98 models have less than 1000 metabolites and reactions. QSopt_ex required about a day to solve all models (Chindelevitch et al., 2014), compared to a few seconds in total for a standard solver.

To advance COBRA for increasingly large biochemical networks, solvers that perform more efficiently than exact solvers and also perform more reliably than standard LO solvers are definitely needed. Gleixner et al. (Gleixner, Steffy, and Wolter, 2012; Gleixner, Steffy, and Wolter, May 2015; Gleixner, 2015; Gleixner, Steffy, and Wolter, 2016) have addressed this need, and Chapter 4 of Gleixner (2015) is devoted to multiscale metabolic networks, showing significant improvement relative to CPLEX (*IBM ILOG CPLEX optimizer* 2014). Our work is complementary and confirms that the simplex solver in the Gleixner et al. references should be enhanced to employ quadruple-precision computation, as we have done here.

We use Single, Double, and Quad to denote the main options for floating-point arithmetic (with precision around 7, 16, and 34 digits respectively). For many years, scientific computation has advanced in two complementary ways: improved *algorithms* and improved *hardware*. Compilers have typically evaluated expressions using the same arithmetic as the variables' data type. Most scientific codes apply Double variables and Double arithmetic throughout (16 significant digits stored in 64-bit words). The floating-point hardware sometimes has *slightly extended precision* (80-bit registers). Kahan (Kahan, 2011) notes that early C compilers generated Double instructions for all floating-point computation *even for program variables stored in single precision*. Thus for a brief period, C programs were serendipitously more reliable than typical Fortran programs of the time. (For Single variables a and b, Fortran compilers would use Single arithmetic to evaluate the basic expressions $a \pm b$, $a{*}b$, a/b, whereas C compilers would transfer a and b to longer registers and operate on them using Double arithmetic.) Most often, the C compiler's extra precision was not needed, but occasionally it did make a critical difference. Kahan calls this the *humane* approach to debugging complex numerical software. Unfortunately, Quad hardware remains very rare and for the foreseeable future will be simulated on most machines by much slower software. Nevertheless, we believe the time has come to produce Quad versions of key sparse-matrix packages and large-scale optimization solvers for multiscale problems.

Here, we report the development and biological application of Quad MINOS, a quadruple-precision version of our general-purpose, industrial-strength linear and nonlinear optimization solver MINOS (Murtagh and Saunders, 1978; Murtagh and Saunders, 1982). We also developed a Double-Quad-Quad MINOS procedure (DQQ) that combines the use of Double and Quad solvers in order to achieve a balance between efficiency in computation and accuracy of the solution. We extensively tested this DQQ procedure on 83 genome-scale metabolic network models (M models) obtained from the UCSD Systems Biology repository (UCSD Systems Biology Research Group, 2015; *Multiscale Systems Biology Collaboration* 2016) and 78 from the BiGG database (King et al., 2016). We also applied DQQ to ME models of *Thermotoga maritima* (Lerman et al., 2012) (about 18,000 metabolites and reactions) and *E. coli* K12 MG1655 (Thiele et al., 2012) (about 70,000 metabolites and reactions). For M models, we find that Double MINOS alone is sufficient to obtain non-zero steady-state solutions that satisfy feasiblility and optimality conditions with a tolerance of 10^{-7}. For ME models, application of our DQQ procedure resulted in non-zero steady-state solutions that satisfy feasibility and optimality conditions with a tolerance of 10^{-20}. The largest ME model required 4.5 hours, mostly in step D of DQQ because of conservative runtime options. Qsopt_ex would not be practical on such a large model unless warm-started at a near-optimal solution. The SoPlex80bit solver (Wunderling, 1996; *SoPlex: The sequential object-oriented simplex solver* 2016) has performed very efficiently on large ME models with the help of rational arithmetic at a near-optimal solution, but had difficulty on some other challenging problems that DQQ solved accurately (see Gleixner (2015, Ch. 4), *problematic* models below, and Supplementary Information).

Thus, we expect our DQQ procedure to be a robust and efficient tool for the increasingly detailed study of biological processes, such as metabolism and macromolecular synthesis, and for challenging optimization problems arising in other scientific fields.

OVERVIEW. A preliminary version of this work appeared in Ma and Saunders (2015). Here we name the approach DQQ and report experiments with an analogous but cheaper DRR procedure based on conventional iterative refinement of all linear equations arising in the simplex method (see Methods section and Supplementary Information). We also became aware of the work of Gleixner, Steffy, and Wolter (2012), Gleixner, Steffy, and Wolter (May 2015), Gleixner (2015), and Gleixner, Steffy, and Wolter (2016) and their thorough and successful implementation of iterative refinement in SoPlex80bit. However, we learned that DRR may lose ground during periodic refactorizations of the simplex basis matrix B, if the current B is nearly singular and "basis repair" becomes necessary. Our DQQ and DRR experience points to the need for an optional Quad version of the basic SoPlex solver to ensure maximum reliability of the refinement approach in the Gleixner et al. references. Meanwhile, DQQ will be effective on a wide range of problems as long as step D finishes naturally or is limited to a reasonable number of iterations before steps Q1 and Q2 take over.

20

RESULTS

We discuss Double and Quad implementations of MINOS applied to *linear optimization* (LO) problems of the form

$$\min_{v}\ c^T v \ \text{ s.t. }\ Sv = 0,\ \ \ell \le v \le u, \tag{20.1}$$

where $S \in R^{m\times n}$. To achieve reliability and efficiency on multiscale problems, we developed the following 3-step procedure.

DQQ PROCEDURE

Step D Apply the Double solver with scaling and somewhat strict runtime options.
Step Q1 Warm-start the Quad solver with scaling and stricter options.
Step Q2 Warm-start the Quad solver with no scaling but stricter options.

DQQ is described further in Algorithm 3, where loop 1 is the primal simplex method, P is a permutation matrix, and δ_1, δ_2 are Feasibility and Optimality tolerances. MINOS terminates loop 1 when the (possibly scaled) bounds on v are satisfied to within δ_1, and the sign of $z_j/(1+\|y\|_\infty)$ is correct to within δ_2. Table 27 shows the default runtime options for Double MINOS and preferred options for each step of DQQ. Scale specifies whether the problem data should be scaled before the problem is solved (and unscaled afterward). Tolerances δ_1, δ_2 specify how well the primal and dual constraints of the (possibly scaled) problem should be satisfied. Expand frequency controls the MINOS anti-degeneracy procedure (Gill et al., 1989). The LU tolerances balance stability and sparsity when LU factors of B are computed and updated.

Steps D and Q1 are usually sufficient, but Q2 costs little more and ensures that the tolerances δ_1 and δ_2 apply to the original (unscaled) problem. For conventional solvers it is reasonable to set δ_1 and δ_2 to 10^{-6} or perhaps as small as 10^{-9}. For Quad MINOS, we set them to 10^{-15} to be sure of capturing variables v_j as small as $O(10^{-10})$.

SMALL M MODELS. Of the 98 metabolic network models in the UCSD Systems Biology repository (UCSD Systems Biology Research Group, 2015), A. Ebrahim was able to parse 83 models (Ebrahim, 2015a) and compute solutions with a range

Algorithm 3 DQQ procedure

```
Data: Linear optimization problem (20.1)
Result: Flux vector $v^*$, basis partition $SP = (B\ N)$, one of three states: optimal, infeasible, or unbounded (possible if infinite $\ell_j$, $u_j$ exist)
Step D: use Double MINOS with scaling
repeat
  Find a nonsingular basis matrix $B$ from the columns of $S$ so that $SP = (B\ N)$
  Find $v = P(v_B, v_N)$ with $Sv \equiv Bv_B + Nv_N = 0$
  Partition $c$ accordingly as $c = P(c_B, c_N)$
  Solve $B^T y = c_B$
  Set $z_N \leftarrow c_N - N^T y;\ \tau \leftarrow (1 + \|y\|_\infty)\delta_2$
until $\forall j \in N$, $z_j \le \tau$ if $v_j = \ell_j$, and $z_j \ge -\tau$ if $v_j = u_j$ (optimal); or fail to find $\ell - \delta_1 \le v \le u + \delta_1$ (infeasible); or fail to find a new $B$ (unbounded)
Step Q1: use Quad MINOS with scaling  Start with the saved $B$ from Step D to run the loop to find a new $B$
Step Q2: use Quad MINOS without scaling  Start with the saved $B$ from Step Q1 to run the loop to reach a final $B$
```

Table 27: Runtime options for MINOS in each step of the DQQ procedure.

	Default	Step D	Step Q1	Step Q2
Precision	Double	Double	Quad	Quad
Scale	Yes	Yes	Yes	No
Feasibility tol δ_1	1e-6	1e-7	1e-15	1e-15
Optimality tol δ_2	1e-6	1e-7	1e-15	1e-15
Expand frequency	10000	100000	100000	100000
LU Factor tol	100.0	1.9	10.0	5.0
LU Update tol	10.0	1.9	10.0	5.0

of solvers (Ebrahim, 2015b). We constructed MPS files for the 83 models (*Multiscale Systems Biology Collaboration* 2016) and solved them via DQQ. Most models have less than 1000 metabolites and reactions. Almost all models solved in less than 0.08 seconds, and many in less than 0.01 seconds. The total time was less than 3 seconds. In contrast, Chindelevitch et al. (2014) reports that the exact solver Qsopt_ex needed a day.

LARGE ME MODELS. COBRA can be used to stoichiometrically couple metabolic and macromolecular expression networks with single nucleotide resolution at genome-scale (Thiele et al., 2012; Lerman et al., 2012). The corresponding Metabolic and macromolecular Expression models (ME models) explicitly represent catalysis by macromolecules, and in turn, metabolites are substrates in macromolecular synthesis reactions. These reconstructions lead to the first multi-timescale and genome-scale stoichiometric models, as they account for multiple cellular functions operating on widely different timescales and typically account for about 40 percent of a prokaryote's open reading frames. A typical M model might be represented by 1000 reactions generated by hand (Feist et al., 2007). In contrast, ME models can have more than 50,000 reactions, most of which have been generated algorithmically from template reactions (defined in the literature) and omics data (Thiele et al., 2012; Lerman et al., 2012). Typical net metabolic reaction rates are 6 orders of magnitude faster than macromolecular synthesis reaction rates (millimole/gDW vs nanomole/gDW, gDW = gram dry weight), and the number of metabolic moieties in a macromolecule can be many orders of magnitude larger than in a typical metabolite. The combined effect is that the corresponding ME models have biochemically significant digits over many orders of magnitude. When Flux Balance Analysis (FBA) is augmented with coupling constraints (Thiele et al., 2010) that constrain the ratio between catalytic usage of a molecule and synthesis of the same molecule, the corresponding linear optimization problem is multiscale in the sense that both data values and solution values have greatly varying magnitudes. For a typical ME model, input data values (objective, stoichiometric or coupling coefficients, or bounds) differ by 6 orders of magnitude, and biochemically meaningful solution values can be as large as 10^8 or as small as 10^{-10}.

The results of DQQ on three large ME models are shown in Tables 28–29, including the model dimensions m and n, the number of nonzeros in S, the norms of the optimal primal and dual variables (v^*, y^*), the iterations and runtime for each step, the final objective value, and log10 of the primal and dual infeasibilities (Pinf and Dinf). The constraints in (20.1) are satisfied to within Pinf, and $z_j/(1+\|y^*\|_\infty)$ has the correct sign to within Dinf, where $B^T y = c_B$ for the optimal basis B, and $z = c - S^T y$.

TMA_ME developed by Lerman et al. (Lerman et al., 2012) has some large entries $|S_{ij}|$ and many small solution values v_j that have meaning to systems biologists. For example, transcription and translation rates can have values $O(10^{-10})$ or less, which is much smaller than metabolic reactions. These small values are linked to large matrix entries arising from building large macromolecules from smaller constituents (Thiele et al., 2012). The ME part of the model also contains small $|S_{ij}|$. For instance, enzyme levels are estimated in ME models by dividing certain metabolic fluxes by "effective rate constants." Because these constants are typically large (e.g., 234,000 h^{-1}), the matrix entries (the inverse of the rate con-

Table 28: Three large ME biochemical network models TMA_ME, GlcAerWT, GlcAlift (Lerman et al., 2012; Thiele et al., 2012; Sun et al., 2013). Dimensions of $m \times n$ constraint matrices S, size of the largest optimal primal and dual variables v^*, y^*, number of iterations and runtimes in seconds for each step, and the total runtime of each model.

ME model	TMA_ME	GlcAerWT	GlcAlift
m	18210	68300	69529
n	17535	76664	77893
nnz(S)	336302	926357	928815
max $\lvert S_{ij} \rvert$	2.1e+04	8.0e+05	2.6e+05
$\lVert v^* \rVert_\infty$	5.9e+00	6.3e+07	6.3e+07
$\lVert y^* \rVert_\infty$	1.1e+00	2.4e+07	2.4e+07
D itns	21026	47718	93857
D time	350.9	10567.8	15913.7
Q1 itns	597	4287	1631
Q1 time	29.0	1958.9	277.3
Q2 itns	0	4	1
Q2 time	5.4	72.1	44.0
Total time	385	12599	16235

stants) become small. In step D, most iterations were needed to find a feasible solution, with the objective then having the correct order of magnitude (but only one correct digit). Step Q1 improved the accuracy, and step Q2 provided confirmation. Note that the efficiency advantage of our approach is also evident: 385 seconds solve time for DQQ (Total time in Table 28) compared to 2 weeks using exact arithmetic (Lerman et al., 2012).

Two slightly different versions of this model provided welcome empirical evidence that the optimal objective and solution values do not change significantly when the problem data are perturbed by $O(10^{-6})$ (see Supplementary Information).

GlcAerWT is a ME model from the study by Thiele et al. (Thiele et al., 2012) After 33,000 iterations in step D, MINOS began to report singularities following updates to the basis factors (71 times during the next 15,000 iterations). After 47,718 iterations (D itns in Table 28), step D terminated with maximum primal and dual infeasibilities $O(10^{-4})$ and $O(1)$ (Pinf and Dinf in Table 29). These were small enough to be classified "Optimal", but we see that the final objective value $-6.7687\text{e+}05$ had no correct digits compared to $-7.0382\text{e+}05$ in steps Q1 and Q2. For large models, step Q1 is important. It required significant work: 4,287 iterations costing 1958.9 seconds (Q1 itns and time in Table 28). Step Q2 soon confirmed the final objective value. The total time (12,599 seconds ≈ 3.5 hours) is

Table 29: Three large ME biochemical network models TMA_ME, GlcAerWT, GlcAlift (Lerman et al., 2012; Thiele et al., 2012; Sun et al., 2013). Optimal objective value of each step, Pinf and Dinf = final maximum primal and dual infeasibilities ($\log_{10}$ values tabulated, except – means 0). Bold figures show the final *(step Q2)* Pinf and Dinf.

ME model	Step	Objective	Pinf	Dinf
TMA_ME	D	8.3789966820e−07	−06	−05
	Q1	8.7036315385e−07	−25	−32
	Q2	8.7036315385e−07	**–**	**−32**
GlcAerWT	D	−6.7687059922e+05	−04	+00
	Q1	−7.0382449681e+05	−07	−26
	Q2	−7.0382449681e+05	**−21**	**−22**
GlcAlift	D	−5.3319574961e+05	−03	−01
	Q1	−7.0434008750e+05	−08	−22
	Q2	−7.0434008750e+05	**−18**	**−23**

modest compared to an expected time of months for the exact solver approach of Chindelevitch et al. (2014).

GlcAlift was generated because of difficulties that TMA_ME and GlcAerWT presented to Double solvers. The lifting technique of Sun et al. (2013) was applied to GlcAerWT to reduce some of the large matrix values. The aim of lifting is to remove the need for scaling (and hence magnified errors from unscaling), but with DQQ we do activate scaling in step D because steps Q1 and Q2 follow. Our experience is that lifting improves accuracy for Double solvers but substantially increases the simplex iterations. On GlcAlift, Double MINOS again reported frequent singularities following basis updates (235 times starting near iteration 40,000). It took 93,857 iterations (D itns in Table 28), twice as many as GlcAerWT, with only a slight improvement in $\max\{\text{Pinf}, \text{Dinf}\}$ (Table 29). Double MINOS with scaling on the lifted model couldn't reach agreement with the final objective −7.0434008750e+05 in steps Q1 and Q2, and the total solve time increased (4.5 hours), mostly in step D. The objective for both GlcA models is to maximize v_{60069}. The fact that there are no correct digits in the step D objectives illustrates the challenge that these models present, but steps Q1 and Q2 are accurate and efficient. The Q2 objectives for GlcAerWt and GlcAlift should be the same, but limited precision in the data files could explain why there is just 3-digit agreement.

The Tomlab interface (*TOMLAB optimization environment for* MATLAB 2015) and CPLEX were used Thiele et al. (2012) to improve the results for standard Double solvers. On the NEOS server (*NEOS server for optimization* 2016), Gurobi was unable to solve GlcAerWT with default parameters (numeric error after nearly

600,000 iterations). It performed considerably better on GlcAlift (about 46,000 iterations) but terminated with a warning of unscaled primal/dual residuals 1.07 and 1.22e-06. As shown above, our DQQ procedure saves researchers' effort on lifting the model, and is able to solve the original model faster (3.5 hours vs 4.5 hours).

Further tests of the DQQ procedure on challenging LO problems are reported in **Methods**. As for the ME models, the simplex method in Double MINOS usually gives a good starting point for the same simplex method in Quad MINOS. Hence, much of the work can be performed efficiently with conventional 16-digit floating-point hardware to obtain near-optimal solutions. For Quad MINOS, 34-digit floating-point operations are implemented in the compiler's Quad math library via software (on today's machines). Each simplex iteration is therefore considerably slower than with floating-point hardware, but the reward is high accuracy. Of interest is that Quad MINOS usually achieves *much more accurate solutions than requested* (see bold figures in Table 29). This is a favorable empirical finding.

21

DISCUSSION

Exact solvers compute exact solutions to LO problems involving rational data. Although stoichiometric coefficients for chemical reactions are in principle integers, most genome-scale metabolic models have non-integer coefficients where the stoichiometry is known to only a few digits, e.g., a coefficient in a biomass reaction. Such a stoichiometric coefficient should not be considered exact data (to be converted into a rational number for use with an exact solver). This casts doubt on any effort to compute an exact solution for a particular FBA problem.

Exact solvers employ rational arithmetic, and have been applied to important problems (Koch, 2004; Applegate et al., 2007; Applegate et al., 2008; Lerman et al., 2012; Gleixner, Steffy, and Wolter, 2012; Gleixner, Steffy, and Wolter, May 2015; Gleixner, 2015; Gleixner, Steffy, and Wolter, 2016). Quad precision and variable-precision floating-point have also been mentioned (Koch, 2004; Applegate et al., 2007). Here, we exploit Quad precision more fully on a range of larger problems, knowing that current genome-scale models will continue to grow even larger.

While today's commercial solvers, including CPLEX, Gurobi, Mosek, and Xpress (*IBM ILOG CPLEX optimizer* 2014; *MOSEK Optimization Software* 2014; *Gurobi optimization system for linear and integer programming* 2014; *FICO Xpress Optimization Suite* 2015), are effective on a wide range of linear and mixed integer optimization models, the work of Thiele et al. (2012) calls for greater reliability in solving FBA and ME models in systems biology. Our DQQ procedure has demonstrated that warm starts with Quad solvers are efficient, and that the accuracy achieved exceeds requirements by a very safe margin. Kahan (Kahan, 2011) has noted that "*carrying somewhat more precision in the arithmetic than twice the precision carried in the data and available for the result will vastly reduce embarrassment due to roundoff-induced anomalies*" and that "*default evaluation in Quad is the humane option,*" as opposed to coding specialized tests for each application. The `real(16)` datatypes in today's Fortran compilers provide a humane method for converting existing Double code to Quad. The `float128` datatype in some C++ compilers makes it possible to switch from Double to Quad at runtime within a single code, making code maintenance even more humane.

Warm starts are essential for steps Q1 and Q2 of DQQ. Exact simplex solvers can also be warm-started, as noted by Gleixner et al. (Gleixner, Steffy, and Wolter, May 2015; Gleixner, 2015). We could envisage a DE procedure: Double solver followed by Exact solver. However, for the GlcA problems in Table 28 (and for the gen problems in the Mészáros *problematic* set below), we see that step Q1 performs a significant number of iterations. Thus, warm-starting an exact solver on large models may not be practical when the Double solver is not reliable.

Looking ahead, we note that metabolic reconstructions of the form (20.1) may need to be processed before they can be treated as stoichiometrically consistent models. As discussed in (Fleming et al., 2016), certain rows of S may need to be deleted according to the solution ℓ of the problem $\max \|\ell\|_0$ s.t. $S^T\ell = 0$, $\ell \geq 0$. This problem can be approximated by the linear problem

$$\begin{aligned} \max_{z,\ell} \quad & \mathbf{1}^T z \\ \text{s.t.} \quad & S^T\ell = 0, \quad z \leq \ell, \quad 0 \leq z \leq \mathbf{1}\alpha, \quad 0 \leq \ell \leq \mathbf{1}\beta, \end{aligned} \tag{21.1}$$

where scalars α, β are proportional to the smallest molecular mass considered nonzero and the largest molecular mass allowed (e.g., $\alpha = 10^{-4}$, $\beta = 10^4$). Note that problem (21.1) involves S^T and is larger than the FBA problem (20.1) itself. We could not design consistent FBA models in this way unless we were sure of being able to solve (21.1) effectively. Our work here offers assurance of such capability.

We believe that reliable solutions are now readily available for large, multiscale applications such as FBA and flux variability analysis (FVA) in systems biology (Palsson, 2006; Orth, Thiele, and Palsson, 2010; Thiele et al., 2012; Gudmundsson and Thiele, 2010; Thiele et al., 2010), and that our DQQ procedure will allow biologists to build increasingly large models to explore metabolism and macromolecular synthesis. Combined use of Double and Quad solvers will help other areas of computational science involving multiscale optimization problems. We have also treated nonlinear constraints directly with the nonlinear algorithms in Quad MINOS (Murtagh and Saunders, 1982; Yang et al., 2016).

22

METHODS

MULTISCALE CONSTRAINT-BASED MODELING. Consider a network of biochemical reactions, represented by a stoichiometric matrix $S \in \mathbb{R}^{m \times n}$ with each row and column corresponding to a molecular species and biochemical reaction, respectively. S_{ij} respresents the *stoichiometry* of molecular species i participating as a substrate (negative) or product (positive) in reaction j. The evolution of molecular species concentrations with respect to time (t) is given by the ordinary differential equation

$$\frac{dx(t)}{dt} = Sv(x(t)), \tag{22.1}$$

where $x(t) \in \mathbb{R}^m_{\geq 0}$ is a vector of time-dependent concentrations and $v(x(t)) : \mathbb{R}^m_{\geq 0} \rightarrow \mathbb{R}^n$ is a nonlinear function of concentrations that depends on the kinetic mechanism of each reaction.

If one assumes that species concentrations are time-invariant, then the set of all steady-state reaction rates, satisfying $Sv(x) = 0$, may be approximated by the linear *steady-state constraint* $Sv = 0$, where $v \in \mathbb{R}^n$ is a vector of reaction fluxes. Thermodynamic principles and experimental data can also be used to specify lower and upper *bound constraints* on reaction fluxes $\ell \leq v \leq u$. Biochemical relationships between the rates of macromolecular synthesis and utilization can be approximated by coupling of the corresponding reaction fluxes (Thiele et al., 2010), e.g., pyruvate kinase reaction flux and the synthesis flux of pyruvate kinase in a ME model (Thiele et al., 2012). Flux coupling can be represented by bounding the ratio between two reaction fluxes with two coupling coefficients:

$$\sigma_{\min} \leq \frac{v_i}{v_j} \leq \sigma_{\max}, \tag{22.2}$$

where v_i and v_j are a pair of non-negative fluxes. This nonlinear constraint can be reformulated into a pair of linear *coupling constraints*

$$\sigma_{\min} v_j \leq v_i, \quad v_i \leq \sigma_{\max} v_j, \tag{22.3}$$

Table 30: Three pilot models from Netlib (*Netlib collection of LP problems in MPS format* 1988) and eight *problematic* problems from Mészáros (Mészáros, 2004). Dimensions of $m \times n$ constraint matrices S, size of the largest nonzero in S, and norm of the optimal primal and dual variables v^*, y^*.

model	m	n	nnz(S)	max $\lvert S_{ij} \rvert$	$\lVert v^* \rVert_\infty$	$\lVert y^* \rVert_\infty$
pilot4	411	1000	5145	2.8e+04	9.6e+04	2.7e+02
pilot	1442	3652	43220	1.5e+02	4.1e+03	2.0e+02
pilot87	2031	4883	73804	1.0e+03	2.4e+04	1.1e+01
de063155	853	1488	5405	8.3e+11	3.1e+13	6.2e+04
de063157	937	1488	5551	2.3e+18	2.3e+17	6.2e+04
de080285	937	1488	5471	9.7e+02	1.1e+02	2.6e+01
gen1	770	2560	64621	1.0e+00	3.0e+00	1.0e+00
gen2	1122	3264	84095	1.0e+00	3.3e+00	1.0e+00
gen4	1538	4297	110174	1.0e+00	3.0e+00	1.0e+00
l30	2702	15380	64790	1.8e+00	1.0e+09	4.2e+00
iprob	3002	3001	12000	9.9e+03	3.1e+02	1.1e+00

or more generally a set of linear inequalities $Cv \leq d$. In addition to the aforementioned physicochemical and biochemical contraints, one may hypothesize a biologically motivated objective. For example, in modeling a growing cell, one may hypothesize that the objective is to maximize the rate of a biomass synthesis reaction. Typically, a biomass synthesis reaction is created with experimentally determined stoichiometric coefficients, each of which represents the relative composition of a cellular biomass constituent. Optimization of a linear combination of reaction fluxes $c^T v$ leads to linear optimization problems: (20.1). Flux balance analysis of a ME model with coupling constraints results in an ill-scaled instance of this problem because the stoichiometric coefficients and coupling coefficients vary over many orders of magnitude.

MINOS IMPLEMENTATION. MINOS (Murtagh and Saunders, 1978; Murtagh and Saunders, 1982) is a linear and nonlinear optimization solver implemented in Fortran 77 to solve problems of the form

$$\min_{v} \; c^T v + \varphi(v) \quad \text{s.t.} \quad \ell \leq \begin{pmatrix} v \\ Sv \\ f(v) \end{pmatrix} \leq u, \tag{22.4}$$

where $\varphi(v)$ is a smooth nonlinear function and $f(v)$ is a vector of smooth nonlinear functions (see Supplementary Information).

Table 31: Iterations and runtimes in seconds for steps D, Q1, Q2 on the problems of Table 30. Pinf and Dinf = final maximum primal and dual infeasibilities ($\log_{10}$ values tabulated, except – means 0). Problem iprob is infeasible. Bold figures show Pinf and Dinf at the end of step Q2. Note that Pinf/$\|v^*\|_\infty$ and Dinf/$\|y^*\|_\infty$ are $O(10^{-30})$ or smaller, even though only $O(10^{-15})$ was requested.

model	Itns	Times	Final objective	Pinf	Dinf
pilot4	1464	0.1	−2.5811392619e+03	−05	−12
	7	0.0	−2.5811392589e+03	−52	−31
	0	0.0	−2.5811392589e+03	**–**	−**29**
pilot	16060	9.0	−5.5739887685e+02	−06	−03
	29	0.3	−5.5748972928e+02	–	−32
	0	0.1	−5.5748972928e+02	**–**	−**32**
pilot87	19340	22.6	3.0171038489e+02	−08	−06
	32	0.9	3.0171034733e+02	–	−32
	0	0.6	3.0171034733e+02	**–**	−**33**
de063155	973	0.1	1.8968895791e+10	−14	+03
	90	0.1	9.8830944565e+09	–	−27
	0	0.0	9.8830944565e+09	**–**	−**24**
de063157	1473	0.1	2.6170359397e+12	–	+08
	286	0.2	2.1528501109e+07	−29	−12
	0	0.0	2.1528501109e+07	**–**	−**12**
de080285	418	0.0	1.4495817688e+01	−09	−02
	132	0.1	1.3924732864e+01	−35	−32
	0	0.0	1.3924732864e+01	**–**	−**32**
gen1	303212	156.9	−8.1861282705e−08	−06	−13
	216746	3431.2	1.2939275026e−06	−12	−31
	8304	112.5	1.2953925804e−06	−**46**	−**31**
gen2	45905	60.0	3.2927907833e+00	−04	−12
	2192	359.9	3.2927907840e+00	–	−29
	0	10.4	3.2927907840e+00	**–**	−**32**
gen4	38111	151.3	−1.2724113149e−07	−07	−12
	58118	6420.2	2.8932557999e−06	−12	−31
	50	4.3	2.8933064888e−06	−**53**	−**30**
l30	1302602	805.6	9.5266141670e−01	−08	−09
	500000	6168.8	−4.5793509329e−26	−25	−00
	16292	204.4	−6.6656750251e−26	−**25**	−**31**
iprob	1087	0.2	2.6891551285e+03	+02	−11
	0	0.0	2.6891551285e+03	+02	−30
	0	0.0	2.6891551285e+03	+02	−**28**

FURTHER TESTS OF DQQ. We report results from the primal simplex solvers in Double and Quad MINOS on two sets of challenging LO problems shown in Table 30. As with the M and ME models, we used an Apple iMac with 2.93 GHz quad-core Intel i7 and gfortran compiler with -O flag (GNU Fortran 5.2.0). The input files were in the MPS format of commercial mathematical programming systems (*Input format for LP data* 1960) with 12-character fields for data values.

The pilot problems. These are economic models developed by Professor George Dantzig in the Systems Optimization Laboratory at Stanford University during the 1980s. They have been used in other computational studies (e.g., Koch (2004)) and are available from Netlib (*Netlib collection of LP problems in MPS format* 1988). We use three examples of increasing size: pilot4, pilot, pilot87. In Table 31, three lines for each problem show the results of steps D, Q1, Q2 of the DQQ procedure.

For pilot, line 1 shows that step D (cold start and scaling) required 16060 iterations and 9 CPU seconds. The unscaled solution v satisfied the constraints in (20.1) to within $O(10^{-6})$ and the dual solution y satisfied the optimality conditions to within $O(10^{-3})$. Line 2 shows that step Q1 needed only 29 further Quad iterations and 0.3 seconds to obtain a very accurate solution. Line 3 shows that the "insurance" step Q2 with no scaling gave an equally good solution (with maximum infeasibilities 0.0 and $O(10^{-32})$). The final Double and Quad objective values differ in the 4th significant digit, as suggested by the $O(10^{-3})$ dual infeasibility in step D. For pilot4 and pilot87 the results are analogous.

The Mészáros problematic problems. Our DQQ procedure was initially developed for this set of LO problems collected by Mészáros (2004), who named them *problematic* and noted that "*modeling mistakes made these problems "crazy," but they are excellent examples to test numerical robustness of a solver*." The first two problems have large entries in S. The step D objective value for de063155 has only 1 digit of precision, and none for de063157. Nevertheless, the infeasibilities Pinf and Dinf for steps Q1 and Q2 are small when the solution norms are taken into account.

The gen problems arise from image reconstruction. There are no large entries in S, v, y, but the primal solutions v are highly degenerate. For gen1, 60% of the step D and Q1 iterations made no improvement to the objective, and 30% of the basic variables in the final solution are on their lower bound. Step Q1 gave an almost feasible initial solution (253 basic variables outside their bounds by more than 10^{-15} with a sum of infeasibilities of $O(10^{-8})$), yet over 200,000 iterations were needed to reach optimality. Evidently Quad precision does not remove the need for a more rigorous anti-degeneracy procedure (such as Wolfe's method as advocated by Fletcher, 2014) or steepest-edge pricing (Forrest and Goldfarb, 1992) to reduce the total number of iterations. Problems gen1 and gen4 show that step Q2 is sometimes needed to achieve high accuracy.

Problem l30 behaved similarly (80% degenerate iterations in steps D and Q1). Since the objective value is essentially zero, we can't expect the Q1 and Q2 objectives to agree. The Q1 iterations were inadvertently limited to 500,000, but step Q2 did not have much further to go.

Problem iprob is artificial and intended to be feasible with a very ill-conditioned optimal basis, but the MPS file contained low-precision data such as 0.604 or 0.0422. The Double and Quad runs determine that the problem is infeasible. This is an example of Quad removing doubt that would inevitably arise with just Double.

Table 31 shows that Quad MINOS usually achieves much greater accuracy than requested (the primal and dual infeasibilities are almost always much smaller than 10^{-15}). Thus our procedure for handling the *problematic* problems has seemed appropriate for the systems biology M and ME models. Like the gen problems, the ME models showed many degenerate iterations in step D, but fortunately not so many total iterations in step Q1 (see Table 28). This is important for FVA and for ME models with nonlinear constraints, which involve multiple warm starts.

ME models (FBA with coupling constraints). In these models, coupling constraints are often functions of the organism's growth rate μ. Thus, O'Brien et al. (2013) consider growth-rate optimization nonlinearly, with μ entering as the objective in (20.1) instead of via a linear biomass objective function. Nonlinear constraints of the form

$$v_i \geq \mu \textstyle\sum_j v_j / k_{i,j}^{\text{eff}} \tag{22.5}$$

are added to (20.1), where v_i, v_j, μ are all variables, and $k_{i,j}^{\text{eff}}$ is an effective rate constant. If μ is fixed at a specific value μ_k, the constraints (22.5) become linear. O'Brien et al. (2013) implemented a binary search on a discrete set of values within an interval $[\mu_{\min}, \mu_{\max}]$ to find the largest $\mu_k \equiv \mu^*$ that keeps the associated linear problem feasible. The procedure required reliable solution of a sequence of LO problems.

Flux Variability Analysis (FVA). After FBA (20.1) returns an optimal objective value $c^T v^* = Z_0$, FVA examines how much a flux v_j can vary within the feasible region without much change to the optimal objective:

$$\min_v \ \pm v_j \ \text{ s.t. } \ Sv = 0, \ \ c^T v \geq \gamma Z_0, \ \ l \leq v \leq u, \tag{22.6}$$

where $0 < \gamma < 1$ and $\gamma \approx 1$. Potentially $2n$ LO problems (22.6) must be solved if all reactions are of interest. Warm starts are used when j is increased to $j+1$

(Gudmundsson and Thiele, 2010). For such a sequence of problems it would be simplest to warm-start each problem in Quad, but warm-starting in Double and then Quad might be more efficient.

CONVENTIONAL ITERATIVE REFINEMENT. A Double simplex solver would be more reliable with the help of iterative refinement (Wilkinson, 1965) on each linear system involving the basis matrix B or its transpose, but we found this inadequate for the biology models (see DRR procedure in Chapter 23).

THE ZOOM STRATEGY. A step toward warm-starting interior methods for optimization was proposed in Saunders and Tenenblat (2006) to take advantage of the fact that a low-accuracy solution (x_1, y_1) for a general problem

$$\min\ c^T x \ \text{ s.t. } \ Ax = b, \ \ \ell \le x \le u \tag{22.7}$$

can be obtained relatively cheaply when an iterative solver for linear systems is used to compute each search direction. (The iterative solver must work harder as the interior method approaches a solution.) If (x_1, y_1) has at least some correct digits, the primal residual $r_1 = b - Ax_1$ will be somewhat small ($\|r_1\| = O(1/\sigma)$ for some $\sigma \gg 1$) and the dual residual $d_1 = c - A^T y_1$ will be comparably small in the elements associated with the final B. If we define

$$\begin{aligned} b_2 &= \sigma r_1, & c_2 &= \sigma d_1, \\ \ell_2 &= \sigma(\ell - x_1), & u_2 &= \sigma(u - x_1), \\ x &= x_1 + \tfrac{1}{\sigma} x_2, & y &= y_1 + \tfrac{1}{\sigma} y_2, \end{aligned} \tag{22.8}$$

and note that the problem is equivalent to

$$\min\ c^T x - y_1^T(Ax - b) \ \text{ s.t. } \ Ax = b, \ \ell \le x \le u \tag{22.9}$$

with dual variable $y - y_1$, we see that x_2 solves

$$\min\ c_2^T x_2 \ \text{ s.t. } \ Ax_2 = b_2, \ \ell_2 \le x_2 \le u_2 \tag{22.10}$$

with dual variable y_2. Importantly, with σ chosen carefully we expect (x_2, y_2) in this "*zoomed in*" problem to be of order 1. Hence we can solve the problem with the same solver as before (as solvers use absolute tolerances and assume that A and the solution are of order 1). If the computed (x_2, y_2) has at least some digits of accuracy, the correction $x_1 \leftarrow x_1 + \frac{1}{\sigma} x_2$, $y_1 \leftarrow y_1 + \frac{1}{\sigma} y_2$ will be more accurate than before. The process can be repeated. With repeated zooms (named *refinement rounds* in Gleixner, Steffy, and Wolter (May 2015) and Gleixner (2015)), the residuals (r_1, d_1) must be computed with increasingly high precision. Subject to the expense of using rational arithmetic for this purpose, Gleixner, Steffy, and Wolter (May 2015) gives extensive results for over 1000 challenging problems and shows that exceptional accuracy can be obtained in reasonable time: only 3 or 4 refinements to achieve 10^{-50} precision, and less than 20 refinements to achieve 10^{-250}. SoPlex80bit (Wunderling, 1996; *SoPlex: The sequential object-oriented simplex solver* 2016) is used for each refinement round with feasibility and optimality tolerances set to 10^{-9}. In Gleixner, Steffy, and Wolter (May 2015) the authors recognize that much depends on the robustness of the simplex solver used for the original problem and each refinement. The potential difficulties are the same as in each step of our DRR procedure, where Double MINOS is on the brink of failure on the Glc problems because B is frequently near-singular when it is refactorized every 100 iterations. A practical answer for Gleixner, Steffy, and Wolter (May 2015) is to use a more accurate floating-point solver such as Quad MINOS or Quad versions of SoPlex or SNOPT (Gill, Murray, and Saunders, 2005b) for all refinement rounds.

DQQ SERVES THE CURRENT PURPOSE. In the context of ME models whose non-integer data is accurate to only 4 or 5 digits, we don't need 10^{-50} precision. Tables 29 and 31 show that our DQQ procedure achieves more accuracy than necessary on all tested examples. For models where the Double solver is expected to encounter difficulty, step D can use a reasonable iteration limit. Step Q1 will perform more of the total work with greatly improved reliability. Step Q2 provides a small but important improvement at negligible cost, ensuring small residuals for the original (unscaled) problem.

THE NEED FOR QUAD PRECISION. To summarize why a conventional Double solver may not be adequate for multiscale problems (even with iterative refinement on systems $Bp = a$ and $B^T y = c_B$ each iteration), we note that the current basis matrix B must be factorized at regular intervals. If B appears to be nearly singular, a "basis repair" procedure replaces some columns of B by appropriate unit vectors (thus making certain slack variables basic). The new B is better conditioned, but the solution obtained after recomputing the basic variables from $Bv_B + Nv_N = 0$ may have an objective value $c^T v$ that is unpredictably less optimal than before. The preceding iterations would make progress, but basis repair allows loss of ground. Basis repair is unlikely to happen if Quad precision is used for all storage and computation, as it is in steps Q1 and Q2 of DQQ.

DATA AND SOFTWARE AVAILABILITY. Double and Quad Fortran 77 implementations of MINOS are included within the Cobra toolbox (Schellenberger et al., 2011; Heirendt et al., 2018). MPS or JSON files for all models discussed are available from *Multiscale Systems Biology Collaboration* (2016). Python code for running Double and Quad MINOS on the BiGG JSON files is also available from *Multiscale Systems Biology Collaboration* (2016).

23

SUPPLEMENTARY INFORMATION

23.1 INTRODUCTION

Figure 9 summarizes our DQQ procedure for achieving reliability and efficiency for multiscale optimization problems.

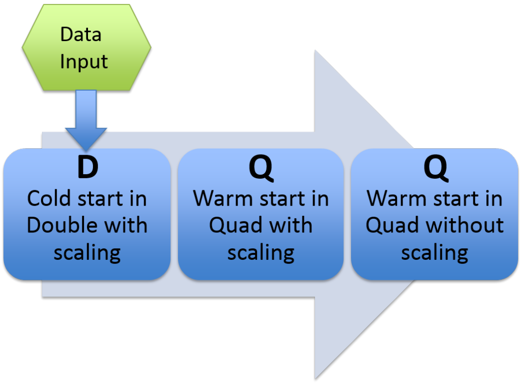
Data Input
D
Cold start in Double with scaling
Q
Warm start in Quad with scaling
Q
Warm start in Quad without scaling

Figure 9: Flowchart for the 3-step DQQ procedure.

The previous chapters report application of DQQ to three large ME models (TMA_ME, GlcAerWT, GlcAlift) and to some other challenging linear optimization problems (the pilot economic models and the Mészáros *problematic* set). Below we provide the following supplementary information:

- Solution of 78 Metabolic models by Double and Quad MINOS, verifying that the Double solver gives reliable results.
- Solution of two slightly different forms of the TMA_ME model, showing robustness of solution values with respect to $O(10^{-6})$ relative perturbations of the data.
- Some details of the Double and Quad MINOS implementations.
- Experiments with conventional iterative refinement (DRR procedure).
- Results with Gurobi on the ME models.
- Results with SoPlex80bit on the ME and *problematic* models.

23.2 Metabolic models with Quad solvers admit biomass synthesis

COBRA models of metabolic networks assume the existence of at least one steady-state flux vector that satisfies the imposed constraints and admits a non-zero optimal objective. Where the objective is to maximize a biomass synthesis reaction, the corresponding FBA problem should admit a nonzero biomass synthesis rate. It is established practice to solve monoscale metabolic FBA problems with Double solvers, so one may ask: do biomass synthesis predictions from metabolic models hold when higher precision solvers are applied to the same FBA problem? We tested 78 M models derived from the BiGG database (King et al., 2016) using Double and Quad MINOS. We downloaded these models in the JSON format and parsed them using the JSON reader in cobrapy (Ebrahim et al., 2013). The models were not modified after loading, so all constraints, bounds, and objective coefficients were used as in the original files. All models were feasible using both Double and Quad, and all but five models had an optimal objective value greater than zero. Of these five models, four simply had all-zero objective coefficients, while the remaining (RECON1) model maximized a single reaction (S6T14g) but its optimal value was zero. The maximum difference in objective value between Double and Quad was 2.6×10^{-12}. The additional precision provided by Quad MINOS enabled us to conclude efficiently and effectively that the 78 metabolic models could be solved reliably using a Double solver. This conclusion is consistent with previous findings by **Ebrahim2015**

23.3 Robustness of solution values for TMA_ME

TMA_ME (Lerman et al., 2012) was the first ME model that we used for Quad experiments. The data S, c, ℓ, u came as a Matlab structure with $c_j = 0$, $\ell_j = 0$, $u_j = 1000$ for most j, except four variables had smaller upper bounds, the last variable had moderate positive bounds, and 64 variables were fixed at zero. The objective was to maximize flux v_{17533}. We output the data to a plain text file. Most entries of S were integers (represented exactly), but about 5000 S_{ij} values were of the form 8.037943687315e-01 or 3.488862338191e-06 with 13 significant digits. The text data was read into Double and Quad versions of a prototype Fortran 90 implementation of SQOPT (Gill, Murray, and Saunders, 2005b).

For the present work, we used the same Matlab data to generate an MPS file for input into MINOS. Since this is limited to 6 significant digits, the values in the preceding paragraph were rounded to 8.03794e-01 and 3.48886e-06 and in total

Table 32: TMA_ME model. Robustness of objective values computed by four high-accuracy solvers for two slightly different versions of the problem with 13-digit and 6-digit data (from Matlab and MPS data respectively).

	Optimal objective	
SoPlex80bit	8.703671403e-07	Matlab data
QSopt_ex	8.703646169e-07	Matlab data
Quad SQOPT	8.703646169e-07	Matlab data
Quad MINOS	8.703631539e-07	MPS data

Table 33: TMA_ME model. Robustness of small solution values v_j and w_j computed by Quad MINOS for two slightly different versions (Matlab and MPS data respectively).

j	107	201	302
v_j	2.336815e-06	8.703646e-07	1.454536e-11
w_j	2.336823e-06	8.703632e-07	1.454540e-11

about 5000 S_{ij} values had $O(10^{-6})$ relative perturbations of this kind. This was a fortuitous limitation for the ME models. We have been concerned that such data perturbations could alter the FBA solution greatly because the final basis matrices could have condition number as large as 10^6 or even 10^{12} (as estimated by LUSOL (**LUSOL**) each time SQOPT or MINOS factorizes the current basis B). However, in comparing Quad SQOPT and Quad MINOS with SoPlex (Wunderling, 1996; *SoPlex: The sequential object-oriented simplex solver* 2016) and the exact simplex solver QSopt_ex (Applegate et al., 2008), we observe in Table 32 that the final objective values for TMA_ME in Matlab data reported by QSopt_ex and Quad SQOPT match in every digit. Moreover, the objective value achieved by Quad MINOS on the perturbed data in MPS format agrees to 5 digits of the results from the exact solver QSopt_ex on the “accurate" data. These results show the robustness of the TMA_ME model and our 34-digit Quad solvers.

More importantly, for the most part *even small solution values* are perturbed in only the 5th or 6th significant digit. Let v and w be the solutions obtained on slightly different data. Some example values are given in Table 33. Among all j for which $\max(v_j, w_j) > \delta_1 = 10^{-15}$ (the feasibility tolerance), the largest relative difference $|v_j - w_j| / \max(v_j, w_j)$ was less than 10^{-5} for all but 31 variables. For 22 of these pairs, either v_j or w_j was primal or dual degenerate (meaning one of them was zero and there are alternative solutions with the same objective value). The remaining 9 variables had v_j, w_j values shown in Table 34.

Table 34: TMA_ME model. The values of 9 fluxes v_j, w_j computed by Quad MINOS for two slightly different versions of the problem, revealing robustness of all 9 solution pairs. These values have 1 digit of agreement. Almost all 17535 pairs of values agree to 5 or more digits.

j	v_j	w_j	Relative difference
16383	6.07e-07	2.04e-06	0.70
16459	1.71e-06	2.18e-06	0.22
16483	2.47e-06	5.99e-07	0.76
16730	1.44e-06	7.87e-07	0.46
17461	1.71e-06	2.18e-06	0.22
17462	2.47e-06	5.99e-07	0.76
17478	6.07e-07	2.04e-06	0.70
17507	1.44e-06	7.87e-07	0.46
17517	8.70e-07	2.97e-06	0.71

We see that the values are small (the same magnitude as the data perturbation) but for each of the nine pairs there is about 1 digit of agreement. We could expect thousands of small solution pairs to differ more, yet for almost *all* 17535 pairs at least 5 digits agree.

Although these observations do not prove robustness of FBA models in general (because we analyzed only one perturbation to one model), they are welcome empirical evidence that the solutions are not extremely unstable. Quad solvers can help evaluate the robustness of future (increasingly large) models of metabolic networks by enabling similar comparison of high-accuracy solutions for slightly different problems.

23.4 MINOS IMPLEMENTATION

MINOS (Murtagh and Saunders, 1978; Murtagh and Saunders, 1982) is a linear and nonlinear optimization solver implemented in Fortran 77 to solve problems of the form

$$\min_v \; c^T v + \varphi(v) \quad \text{s.t.} \quad \ell \le \begin{pmatrix} v \\ Sv \\ f(v) \end{pmatrix} \le u, \tag{23.1}$$

where $\varphi(v)$ is a smooth nonlinear function and $f(v)$ is a vector of smooth nonlinear functions. The matrix S and the Jacobian of $f(v)$ are assumed to be sparse.

Let Single/Double/Quad denote the floating-point formats defined in the 2008 IEEE 754 standard (**IEEE754**) with about 7/16/34 digits of precision, respectively. Single is not useful in the present context, and Double may not ensure adequate accuracy for multiscale problems. This is the reason for our work. Since release 4.6 of the GCC C and Fortran compilers (**GCC**), Quad has been available via the `long double` and `real(16)` data types. Thus, we have made a Quad version of Double MINOS using the GNU gfortran compiler (GNU Fortran 5.2.0).

On today's machines, Double is implemented in hardware, while Quad (if available) is typically implemented in a software library, in this case GCC libquadmath (**GCC-libquadmath**).

For Double MINOS, floating-point variables are declared `real(8)` ($\approx$ 16 digits). For Quad MINOS, they are `real(16)` ($\approx$ 34 digits) with the data S, c, ℓ, u stored in Quad even though they are not known to that precision. This allows operations such as Sv and S^Ty to be carried out directly on the elements of S and the Quad vectors v, y. If S were stored in Double, such products would require each entry S_{ij} to be converted from Double to Quad at runtime (many times).

The primal simplex solver in MINOS includes geometric mean scaling (Fourer, 1982), the EXPAND anti-degeneracy procedure (Gill et al., 1989), and partial pricing (but no steepest-edge pricing, which would generally reduce total iterations and time). Basis LU factorizations and updates are handled by LUSOL (**LUSOL**). Cold starts use a Crash procedure to find a triangular initial basis matrix. Basis files are used to preserve solutions between runs and to enable warm starts.

Scaling is commonly applied to linear programs to make the scaled data and solution values closer to 1. Feasibility and optimality tolerances can be chosen more easily for the scaled problem, and LU factors of the basis matrix are more likely to be sparse. For geometric mean scaling, several passes are made through the columns and rows of S to compute a scale factor for each column and row. A difficulty is that the scaled problem may solve to within specified feasibility and optimality tolerances, but when the solution is unscaled it may lie significantly outside the original (unscaled) bounds.

EXPAND tries to accommodate consecutive "degenerate" simplex iterations that make no improvement to the objective function. The problem bounds are effectively expanded a tiny amount each iteration to permit nonzero improvement. Convergence is usually achieved but is not theoretically guaranteed (**HallMcKinnon2004**). Progress sometimes stalls for long sequences of iterations.

LUSOL bounds the subdiagonals of L when the current basis matrix B is factorized as $P_1BP_2 = LU$ with some permutations P_1, P_2. It also bounds off-diagonal elements of elementary triangular factors L_j that update L in product form each simplex iteration. (The diagonals of L and each L_j are implicitly 1.) Maximum numerical stability would be achieved by setting the LU Factor and Update tolerances to be near 1.0, but larger values are typically chosen to balance stability with sparsity. For safety, we specify 1.9 in step D of DQQ. This value guards against unstable factorization of the deceptive matrix tridiag$(-1\ 2\ 1)$, and improves the reliability of Double MINOS in the present context.

23.5 CONVENTIONAL ITERATIVE REFINEMENT

For the biology models, our aim is to satisfy Feasibility and Optimality tolerances of 10^{-15} (close to Double precision). It is reasonable to suppose that this could be achieved within a Double simplex solver by implementing iterative refinement (Wilkinson, 1965) for every linear system involving the basis matrix B or B^T. This is a more sparing use of Quad precision than our DQQ procedure. For example, each time the current B is factorized directly (typically a new sparse LU factorization every 100 iterations), the constraints $Sv = 0$ can be satisfied more accurately by computing the primal residual $r = 0 - Sv$ from the current solution v, solving $B\Delta v_B = r$, and updating $v_B \leftarrow v_B + \Delta v_B$. In general, the new v will not be significantly more accurate unless r is computed in Quad. (If B is nearly singular, more than one refinement may be needed.) Similarly for solving $B^Ty = c_B$ after refactorization, and for two systems of the form $Bp = a$ and $B^Ty = c_B$ each iteration of the simplex method.

By analogy with DQQ, we implemented the following procedure within a test version of Double MINOS. Note that "iterative refinement" in steps R1, R2 means a single refinement for each B or B^T system, with residuals $-Sv$, $a - Bp$, $c_B - B^Ty$ computed in Quad as just described.

DRR PROCEDURE

Step D Apply Double MINOS with scaling and moderately strict runtime options.
Step R1 Warm-start Double MINOS with scaling, stricter options, and iterative refinement.
Step R2 Warm-start Double MINOS without scaling but with stricter options and iterative refinement.

Step D is the same as for DQQ (with no refinement). The runtime options for each step are the same as for DQQ, except in steps R1, R2 the tolerances 1e-15 were relaxed to 1e-9.

Table 35: DRR procedure on three ME models. Iterations and runtimes in seconds for step D (Double MINOS with scaling) and steps R1, R2 (Double MINOS with iterative refinement, with and without scaling). Pinf and Dinf = final maximum primal and dual infeasibilities ($\log_{10}$ values tabulated). Bold figures show Pinf and Dinf at the end of step R2. The fourth line for each model shows the correct objective value (from step Q2 of DQQ).

model	Itns	Times	Final objective	Pinf	Dinf
TMA_ME	21026	350.9	8.3789966820e−07	−06	−05
	422	25.4	8.6990918717e−07	−08	−07
	71	0.0	8.7035701805e−07	**−10**	**−10**
			8.7036315385e−07		
GlcAerWT	47718	10567.8	−6.7687059922e+05	−04	+00
	907	1442.7	−7.0344344753e+05	−04	−04
	157	151.2	−7.0344342883e+05	**−10**	**−02**
			−7.0382449681e+05		
GlcAlift	19340	15913.7	−5.3319574961e+05	−03	−01
	447	198.8	−7.0331052509e+05	−03	−03
	460	0.6	−7.0330602383e+05	**−06**	**−10**
			−7.0434008750e+05		

In Table 35 we see that this simplified (cheap) form of iterative refinement is only partially successful, with step R2 achieving only 4, 3, and 2 correct digits in the final objective. For GlcAerWT, steps R1 and R2 encountered frequent near-singularities in the LU factors of B (requiring excessive refactorizations and alteration of B), and in step R2, the single refinement could not always achieve full Double precision accuracy for each system. Additional refinements would improve the final Pinf and Dinf, but would not reduce the excessive factorizations. We conclude that on the bigger ME problems, a Double solver is on the brink of failure even with the aid of conventional (Wilkinson-type) iterative refinement of each system involving B and B^T. We conclude that our DQQ procedure is a more expensive but vitally more robust approach.

23.6 RESULTS WITH NEOS/GUROBI

For large linear models, commercial solvers have reached a high peak of efficiency. It would be ideal to make use of them to the extent possible. For example, their Presolve capability allows most of the optimization to be performed on a greatly reduced form of any typical model.

Table 36: Performance of Gurobi with default options on three ME models. Note that "switch to quad" means switch to 80-bit floating-point (not to IEEE Quad precision). This did not help GLcAerWT. For GlcAlift2, the options were NumericFocus 3, no Presolve, and no scaling.

TMA_ME	Presolve	18209 × 17535 → 2386 × 2925
Optimal	Iterations	1703
0.5 secs	Objective	9.6318438361e-07
	True obj	8.7036315385e-07
GlcAerWT	Presolve	68299 × 76664 → 18065 × 26157
	Warning	switch to quad (itns ≈ 14000)
Numeric error	Iterations	593819
3715 secs	Objective	3.2926249e+07
	True obj	-7.0382449681e+05
GlcAlift	Presolve	69528 × 77893 → 18063 × 26155
	Warning	switch to quad (itns ≈ 10000)
Optimal	Iterations	45947
109 secs	Objective	-7.043390954e+05
	True obj	-7.0434008750e+05
	Warning	unscaled primal/dual residuals:
		1.07, 1.22e-06
GlcAlift2		
Optimal	Iterations	128596
844 secs	Objective	-7.043415774e+05
	True obj	-7.0434008750e+05
	Warning	unscaled primal residual:
		1.05e-05

Table 36 summarizes the performance of Gurobi (*Gurobi optimization system for linear and integer programming* 2014) on three large ME models via the NEOS server (*NEOS server for optimization* 2016). The first three results used Gurobi's default runtime options, including Presolve, Dual simplex, and Scaling (with default FeasibilityTol = OptimalityTol = 1e-6). TMA_ME seemed to solve successfully, but from the Quad MINOS solution we know that Gurobi's final objective value has no correct digits. GlcAerWT failed with "Numeric error" after many expensive iterations using 80-bit floating-point. GlcAlift also switched to 80-bit floating-point. The scaled problem seemed to solve successfully, but unscaling damaged the primal residual and this casts significant doubt on the final solution. (This is the reason for our research.)

For GlcAlift2 we specified NumericFocus 3 with no Presolve and no scaling. These options are appropriate for lifted models (Sun et al., 2013). Gurobi did not switch to 80-bit arithmetic, yet achieved 5 correct digits in the objective. This helps confirm the value of the lifting strategy of Sun et al. (2013), and would provide a good starting point for steps Q1 and Q2 of DQQ. However, DQQ permits us to solve the original model GlcAerWT directly (without the lifting transformation).

23.7 RESULTS WITH NEOS/SOPLEX80BIT

Table 37 summarizes the performance of SoPlex80bit (*SoPlex: The sequential object-oriented simplex solver* 2016) on the three large ME models via NEOS with default options, except the simplifier and lifting options were turned off to ensure that SoPlex80bit was iterating on the same problems as MINOS.

SoPlex80bit performed extremely well on all ME models (Table 37). The first floating-point solves achieved maximum primal and dual feasibilities of order 1e-7 or less, with no sign of scaling or potentially troublesome unscaling, and three rounds of iterative refinement reduced the infeasibilities to order 1e-44(!). The optimal objective values agreed to the 11 digits printed by Quad MINOS. Analogous excellent performance by SoPlex80bit on large ME models is described by Gleixner (2015, Ch. 4).

On the *problematic* set (Table 38), SoPlex80bit solved most problems solved accurately, but with some anomalies. On de063157, the first floating-point solve achieved 5 significant digits in the objective function but with primal and dual infeasibilities of 4e+2 and 1e+4. The first refinement reduced the latter to 2e+1 and 3e-12, and the second refinement achieved 4e-15 and 3e-12. This should have

Table 37: Performance of SoPlex80bit on three large ME models (default options except no simplifier or lifting).

TMA_ME		18209 × 17535
Optimal	Iterations	19563 (3 refinements)
90.9 secs	Objective	`8.7036315385e-07`
	True obj	`8.7036315385e-07`
GlcAerWT		68299 × 76664
Optimal	Iterations	86366 (3 refinements)
1059 secs	Objective	`-7.0382449681e+05`
	True obj	`-7.0382449681e+05`
GlcAlift		69528 × 77893
Optimal	Iterations	83941 (3 refinements)
889 secs	Objective	`-7.0434008750e+05`
	True obj	`-7.0434008750e+05`

been acceptable, but a further 100 refinements were conducted (at negligible cost) before the run was terminated with no final solution available.

On gen2 and gen4, the first floating-point solves were very efficient and accurate (41 and 82 seconds respectively). Three refinements achieved primal and dual infeasibilities of order 1e-11 or less. A final rational factorization proved expensive and accounted for 99% of the total times (6016 and 7132 seconds respectively), but confirmed optimality.

On l30, the first floating-point solve performed many iterations but achieved primal and dual infeasibilities of 4e-9 and 1e-10 with objective value −2.5e-11, which should have been acceptable. The first refinement reported numerical troubles after 2702 iterations. It continued to about 154000 iterations and computed an unbounded ray. Nearly 4000 refinements followed (each doing no iterations) before numerical trouble was reported. One final solve performed 3000 iterations before increasing the Markowitz threshold and terminating with no solution.

Details of this nature will change, but some of them hint at the need for higher precision in the floating-point solver to facilitate SoPlex's iterative refinement.

23.8 looking ahead

The large-scale optimizer SNOPT (Gill, Murray, and Saunders, 2005b) is maintained as a Fortran 77 solver `snopt7` (**UCSDsoftware**) suitable for step D of the DQQ procedure. An accompanying Fortran 2003 version `snopt9` has also been developed, for which Double and Quad libraries can be built with only one line

Table 38: Performance of SoPlex80bit on the *problematic* models (default options except no simplifier or lifting).

de063155		852 × 1488
Optimal	Iterations	1766 (3 refinements)
0.3 secs	Objective	9.8830944565e+09
	True obj	9.8830944565e+09
de063157		936 × 1488
Optimal	Iterations	3828 before refinement
0.1 secs	Objective	2.15277062e+07
	True obj	2.1528501109e+07
de080285		936 × 1488
	Iterations	804 (2 refinements)
0.1 secs	Objective	1.3924732864e+01
	True obj	1.3924732864e+01
gen1		769 × 2560
	Iterations	12850 (3 refinements)
186.5 secs	Objective	1.2953925804e-06
	True obj	1.2953925804e-06
gen2		1121 × 3264
Optimal	Iterations	12079 (2 refinements)
6016 secs	Objective	3.2927907840e+00
	True obj	3.2927907840e+00
gen4		1537 × 4297
Optimal	Iterations	14358 (3 refinements)
7132 secs	Objective	2.8933064888e-06
	True obj	2.8933064888e-06
l30		2701 × 15380
	Iterations	3400093 before refinement
11552 secs	Objective	-2.54658516e-11
	True obj	-6.6.......e-26
iprob		3001 × 3001
Infeasible	Iterations	3001 (2 refinements)
0.6 secs	Objective	1.0e+100

of source code changed. They are ideal for applying DQQ to future multiscale linear and nonlinear optimization models, as long as step D can be terminated early enough when numerical difficulties arise. Quad enhancements to the SoPlex floating-point solver also promise reliability and extreme accuracy for future challenging models.

BIBLIOGRAPHY

Aaker, David A. *Managing Brand Equity*. New York: The Free Press, 1991.

Amestoy, Patrick R., Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling." *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 15–41. DOI: 10.1137/S0895479899358194.

Applegate, D., W. Cook, S. Dash, and M. Mevnkamp. *QSopt_ex: A simplex solver for computing exact rational solutions to LP problems*. (Date of access: 13/10/2016). 2008. URL: http://www.math.uwaterloo.ca/~bico/qsopt/ex.

Applegate, D. L., W. Cook, S. Dash, and D. G. Espinoza. "Exact solutions to linear programming problems." *Operations Res. Lett.* 35 (2007), pp. 693–699.

Ariely, Dan. *Predictably Irrational*. HarperCollins Publishers, 2008.

Arreckx, S. and D. Orban. *A regularized factorization-free method for equality-constrained optimization*. Technical Report GERAD G-2016-65. Montréal, QC, Canada: GERAD, 2016. DOI: 10.13140/RG.2.2.20368.00007.

Berry, Steven, James Levinsohn, and Ariel Pakes. "Automobile Prices in Market Equilibrium." *Econometrica* 63.4 (1995), pp. 841–890.

Berry, Steven T. "Estimating Discrete-Choice models of Product Differentiation." *The RAND Journal of Economics* 25.2 (1994), pp. 242–262.

Byrd, Richard H., Jorge Nocedal, and Richard A. Waltz. "Knitro: An Integrated Package for Nonlinear Optimization." In: *Large-Scale Nonlinear Optimization*. Ed. by G. Di Pillo and M. Roma. Boston, MA: Springer US, 2006, pp. 35–59. DOI: 10.1007/0-387-30065-1_4.

Chindelevitch, L., J. Trigg, A. Regev, and B. Berger. "An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models." *Nat. Commun.* 5.4893 (2014), 9 pp.

Conn, A. R., N. I. M. Gould, and Ph. L. Toint. "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds." *SIAM J. Numer. Anal.* 28 (1991), pp. 545–572. DOI: 10.1137/0728030.

— *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*. Lecture Notes in Computation Mathematics 17. Berlin, Heidelberg, New York, London, Paris and Tokyo: Springer Verlag, 1992. ISBN: 3-540-55470-X 65-04.

Court, Andrew T. “Hedonic Price Indexes with Automotive examples.” *The Dynamics of Automobile Demand* (1939), pp. 98–119.

Dekimpe, Marnik G., Jan-Benedict E.M. Steenkamp, Martin Mellens, and Piet Vanden Abeele. “Decline and variability in brand loyalty.” *International Journal of Research in Marketing* 14.5 (1997), pp. 405–420.

Dhiflaoui, M. et al. “Certifying and repairing solutions to large LPs: How good are LP-solvers?” In: *Proceedings of the 14th annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’03)*. Baltimore, MD, 2003, pp. 255–256.

Diewert, W. Erwin. “Hedonic Regressions. A Consumer Theory Approach.” *National Bureau of Economic Research* Scanner Data and Price Indexes (1961), pp. 317–348.

Dinnie, Keith. “Country-of-origin 1965-2004: A literature review.” *Journal of Customer Behavior* 3.2 (2004), pp. 165–213.

Dubé, Jean-Pierre, Jeremy T. Fox, and Che-Lin Su. “Improving the numerical performance of static and dynamic aggregate discrete choice random coefficients demand estimation.” *Econometrica* 80.5 (2012), pp. 2231–2267.

Ebrahim, A. *Generation of 83 models from UCSD repository*. (Date of access: 10/06/2016). 2015. URL: https://github.com/opencobra/m_model_collection/blob/master/load_models.ipynb.

— *Solution of 83 models from UCSD repository*. (Date of access: 10/06/2016). 2015. URL: https://github.com/opencobra/m_model_collection/blob/master/exact_solving_models.ipynb.

Ebrahim, Ali, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. “COBRApy: COnstraints-Based Reconstruction and Analysis for Python.” *BMC Syst Biol* 7 (2013), p. 74.

Erdem, Tülin. “An Empirical Analysis of Umbrella Branding.” *Journal of Marketing Research* 35.3 (1998), pp. 339–351.

Farquhar, Peter H. “Managing Brand Equity.” *Marketing Research* (1989), pp. 24–33.

Feist, A. M. et al. “A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information.” *Mol Syst Biol* 3.1 (2007). (Date of access: 05/03/2016), e121. DOI: 10.1038/msb4100155. URL: http://www.hubmed.org/display.cgi?uids=17593909.

FICO Xpress Optimization Suite. (Date of access: 27/04/2014). 2015. URL: http://www.fico.com/en/products/fico-xpress-optimization-suite/.

Fleming, R. M. T., N. Vlassis, I. Thiele, and M. A. Saunders. "Conditions for duality between fluxes and concentrations in biochemical networks." *J. Theoretical Biology* 409 (2016), pp. 1–10.

Fletcher, R. "On Wolfe's method for resolving degeneracy in linearly constrained optimization." *SIAM J. Optim.* 24.3 (2014), pp. 1122–1137.

Forrest, J. J. and D. Goldfarb. "Steepest-edge simplex algorithms for linear programming." *Math. Program.* 57 (1992), pp. 341–374.

Fourer, R. "Solving staircase linear programs by the simplex method, 1: Inversion." *Math. Program.* 23 (1982), pp. 274–313.

Fourer, R., D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. second. Pacific Grove: Brooks/Cole, 2002.

Friedlander, M. P. and D. Orban. "A primal–dual regularized interior-point method for convex quadratic programs." *Math. Prog. Comp.* 4.1 (2012), pp. 71–107. DOI: 10.1007/s12532-012-0035-2.

Friedlander, M. P. and M. A. Saunders. "A globally convergent linearly constrained Lagrangian method for nonlinear optimization." *SIAM J. Optim.* 15.3 (2005), pp. 863–897. DOI: 10.1137/S1052623402419789.

Gill, P. E., W. Murray, and M. A. Saunders. "SNOPT: An SQP algorithm for large-scale constrained optimization." *SIAM Review* 47.1 (2005). SIGEST article, pp. 99–131. DOI: 10.1137/S0036144504446096.

— "SNOPT: An SQP algorithm for large-scale constrained optimization." *SIAM Review* 47.1 (2005). SIGEST article, pp. 99–131.

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright. "A practical anti-cycling procedure for linear and nonlinear programming." *Math. Program.* 45 (1989), pp. 437–474.

Gleixner, A. M., D. E. Steffy, and K. Wolter. "Improving the accuracy of linear programming solvers with iterative refinement." In: *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*. Grenoble, 2012, pp. 187–194.

— "Iterative refinement for linear programming." *INFORMS J. Computing* 28.3 (2016), pp. 449–464.

— *Iterative refinement for linear programming*. ZIB Report 15-15. Berlin, Germany: Konrad-Zuse-Zentrum für Informationstechnik Berlin, May 2015.

Gleixner, Ambros M. "Exact and Fast Algorithms for Mixed-Integer Nonlinear Programming." PhD thesis. Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Technical University of Berlin, 2015.

Goldman, Marshall I. “Product Differentiation and Advertising: Some Lessons from Soviet Experience.” *Journal of Political Economy* 68.4 (1960), pp. 346–357.

Griliches, Zvi. “Hedonic Price Indexes for Automobiles: An Econometric Analysis of Quality Change.” *National Bureau of Economic Research* The Price Statistics of the Federal Government (1961), pp. 173–196.

Gudmundsson, S. and I. Thiele. “Computationally efficient flux variability analysis.” *BMC Bioinformatics* 11.489 (2010).

Guesnerie, R. and J. Seade. “Nonlinear pricing in a finite economy.” *Journal of Public Economics* 17 (1982), pp. 157–179.

Gurobi optimization system for linear and integer programming. (Date of access: 01/02/2016). 2014. URL: http://www.gurobi.com.

Heirendt et al. “Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3. 0.” *Nature Protocols* forthcoming (2018).

IBM ILOG CPLEX optimizer. (Date of access: 27/04/2014). 2014. URL: http://www.ibm.com/software/commerce/optimization/cplex-optimizer/.

Input format for LP data. (Date of access: 27/04/2014). 1960. URL: http://lpsolve.sourceforge.net/5.5/mps-format.htm.

IPOPT open source NLP solver. https://projects.coin-or.org/Ipopt.

Judd, K. L., D. Ma, M. A. Saunders, and C.-L. Su. *Optimal income taxation with multidimensional taxpayer types*. Working paper, Hoover Institution, Stanford University. 2017.

Kahan, W. *Desperately needed remedies for the undebuggability of large floating-point computations in science and engineering*. IFIP/SIAM/NIST Working Conference on Uncertainty Quantification in Scientific Computing, Boulder CO. (Date of access: 16/12/2013). 2011. URL: http://www.eecs.berkeley.edu/~wkahan/Boulder.pdf.

Keller, Kevin Lane. “Conceptualizing, Measuring, and Managing Customer-Based Brand Equity.” *Journal of Marketing* 57.1 (1993), pp. 1–22.

King, Z. A. et al. “BiGG Models: A platform for integrating, standardizing, and sharing genome-scale models.” *Nucl. Acids Res.* 44 (2016), pp. D515–522. DOI: doi:10.1093/nar/gkv1049.

KNITRO optimization software. https://www.artelys.com/tools/knitro_doc/2_userGuide.html.

Koch, T. “The final NETLIB-LP results.” *Operations Research Letters* 32 (2004), pp. 138–142.

LANCELOT optimization software. http://www.numerical.rl.ac.uk/lancelot/blurb.html.

Lerman, J. A. et al. “In silico method for modelling metabolism and gene product expression at genome scale.” *Nat. Commun.* 3.929 (2012), 10 pp.

Ling, Davina C., Ernst R. Berndt, and Margaret K. Kyle. “Deregulating direct-to-consumer marketing of prescription drugs: Effects on prescription and over-the-counter product sales.” *Journal of Law & Economics* XLV (2002), pp. 691–723.

Luo, Z.-Q., J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, UK, 1996.

Ma, D. and M. A. Saunders. “Solving multiscale linear programs using the simplex method in quadruple precision.” In: *Numerical Analysis and Optimization, NAO-III, Muscat, Oman, January 2014*. Ed. by M. Al-Baali, L. Grandinetti, and A. Purnama. Vol. 134. Springer Proceedings in Mathematics and Statistics. Springer International Publishing Switzerland, 2015.

Ma, D., K. L. Judd, D. Orban, and M. A. Saunders. “Stabilized optimization via an NCL algorithm.” *Numerical Analysis and Optimization* 235 (2018), pp. 173–191.

Mészáros, C. *A collection of challenging LP problems*. (Date of access: 19/03/2014). 2004. URL: http://www.sztaki.hu/~meszaros/public_ftp/lptestset/problematic.

Mirrlees, J.A. “An exploration in the theory of optimum income taxation.” *Review of Economic Studies* 38 (1971), pp. 175–208.

Montgomery, Cynthia A. and Birger Wernerfelt. “Risk Reduction and Umbrella Branding.” *The Journal of Business* 65.1 (1992), pp. 31–50.

MOSEK Optimization Software. (Date of access: 27/04/2014). 2014. URL: http://www.mosek.com/.

Multiscale Systems Biology Collaboration. (Date of access: 13/09/2016). 2016. URL: http://stanford.edu/group/SOL/multiscale/models.html.

Murtagh, B. A. and M. A. Saunders. “A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints.” *Math. Program. Study* 16 (1982), pp. 84–117.

— “Large-scale linearly constrained optimization.” *Math. Program.* 14 (1978), pp. 41–72.

NCL. http://stanford.edu/group/SOL/multiscale/models/NCL/.

NEOS server for optimization. (Date of access: 02/01/2016). 2016. URL: http://www.neos-server.org/neos/.

Netlib collection of LP problems in MPS format. (Date of access: 27/04/2014). 1988. URL: http://www.netlib.org/lp/data.

Nevo, Aviv. "A Practitioner's Guide to estimation of random-coefficients logit models of demand." *Journal of Economics & Management Strategy* 9.4 (2000), pp. 513–548.

O'Brien, E. J., J. A. Lerman, R. L. Chang, D. R. Hyduke, and B. O. Palsson. "Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction." *Mol Syst Biol* 9.1 (2013), p. 693.

Orth, Jeffrey D., Ines Thiele, and Bernhard O. Palsson. "What is flux balance analysis?" *Nature Biotechnology* 28.3 (2010), pp. 245–248.

Outrata, J., M. Kocvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications, and Numerical Results*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

Palsson, B. O. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, NY, 2006.

Robinson, S. M. "A quadratically-convergent algorithm for general nonlinear programming problems." *Math. Program.* 3 (1972), pp. 145–156. DOI: 10.1007/BF01584986.

Rosen, Sherwin. "Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition." *The Journal of Political Economy* 82.1 (1974), pp. 34–55.

Saez, Emmanuel. "Using Elasticities to Derive Optimal Income Tax Rates." *Review of Economic Studies* 68 (2001), pp. 205–229.

Saridakis, Charalampos and George Baltas. "Modeling price-related consequences of the brand origin cue: An empirical examination of the automobile market." *Mark Lett* 27 (2016), pp. 77–87.

Saunders, M. A. and L. Tenenblat. *The Zoom strategy for accelerating and warm-starting interior methods*. Talk at INFORMS Annual Meeting, Pittsburgh, PA, USA. (Date of access: 10/06/2016). 2006. URL: http://stanford.edu/group/SOL/talks/saunders-tenenblat-INFORMS2006.pdf.

Schellenberger, J. et al. "Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0." *Nature Protocols* 6.9 (2011), pp. 1290–1307.

Schooler, Robert D. "Product Bias in the Central American Common Market." *Journal of Marketing Research* II.November (1965), pp. 394–398.

SoPlex: The sequential object-oriented simplex solver. (Date of access: 16/06/2016). 2016. URL: http://soplex.zib.de.

Su, Che-Lin and Kenneth L. Judd. "Constrained optimization approaches to estimation of structural models." *Econometrica* 80.5 (2012), pp. 2213–2230.

Sun, Y., R. M. T. Fleming, I. Thiele, and M. A. Saunders. “Robust flux balance analysis of multiscale biochemical reaction networks.” *BMC Bioinformatics* 14.240 (2013).

Tarkianen, R. and M. Tuomala. “Optimal Nonlinear Income Taxation with a two-dimensional population: A computational approach.” *Computational Economics* 13 (1999), pp. 1–16.

Thiele, I., R. M. T. Fleming, A. Bordbar, R. Que, and B. O. Palsson. “A systems biology approach to the evolution of codon use pattern.” *Nature Precedings* (2011). (Date of access: 13/10/2016). URL: http://dx.doi.org/10.1038/npre.2011.6312.1.

Thiele, I., R. M. T. Fleming, A. Bordbar, J. Schellenberger, and B. O. Palsson. “Functional characterization of alternate optimal solutions of *Escherichia coli*’s transcriptional and translational machinery.” *Biophysical J.* 98.10 (2010), pp. 2072–2081.

Thiele, I., N. Jamshidi, R. M. T. Fleming, and B. O. Palsson. “Genome-scale reconstruction of *E. coli*’s transcriptional and translational machinery: A knowledge-base, its mathematical formulation, and its functional characterization.” *PLoS Comput Biol* 5.3 (2009), e1000312.

Thiele, I., R. M. T. Fleming, R. Que, A. Bordbar, D. Diep, and B. O. Palsson. “Multiscale modeling of metabolism and macromolecular synthesis in *E. coli* and its application to the evolution of codon usage.” *PLOS ONE* 7.9 (2012), 18 pp.

TOMLAB optimization environment for MATLAB. (Date of access: 03/05/2016). 2015. URL: http://tomopt.com.

Tomlin, J. A. “On scaling linear programming problems.” In: *Computational Practice in Mathematical Programming*. Vol. 4. Mathematical Programming Studies. Springer, 1975, pp. 146–166.

UCSD Systems Biology Research Group. *Available predictive genome-scale metabolic network reconstructions*. (Date of access: 03/05/2016). 2015. URL: http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms.

Wächter, A. and L. T. Biegler. “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming.” *Math. Program.* 106.1 (2006). DOI: 10.1007/s10107-004-0559-y.

Wernerfelt, Birger. “Umbrella Branding as a Signal of New Product Quality: An Example of Signaling by Posting a Bond.” *RAND Journal of Economics* 19.3 (1988), pp. 458–466.

Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Oxford University Press, 1965.

Wilson, R. *Nonlinear Pricing*. Oxford University Press, New York, 1993.

Wilson, R. “Nonlinear pricing and mechanism design.” In: *Handbook of Computational Economics*. Ed. by H. Amman, D. Kendrick, and J. Rust. Vol. 1. Elsevier Science Publishers, B.V., Amsterdam, 1995, pp. 205–229.

Wunderling, R. “Paralleler und objektorientierter Simplex-Algorithmus.” PhD thesis. Technische Universität Berlin, 1996.

Yang, L., D. Ma, A. Ebrahim, C. J. Lloyd, M. A. Saunders, and B. O. Palsson. “solveME: fast and reliable solution of nonlinear ME models.” *BMC Bioinformatics* 17.391 (2016).

Zhang, Kaifu. “Breaking Free of a Stereotype: Should a Domestic Brand Pretend to Be a Foreign One?” *Marketing Science* 34.4 (2015), pp. 539–554.