



Universidade Federal de Itajubá
Instituto de Matemática e Computação

Renan Barbosa de Moraes

Solucionando o *Inverse Power Index Problem*

ITAJUBÁ
2021

Renan Barbosa de Moraes

Solucionando o *Inverse Power Index Problem*

Monografia apresentada como trabalho final de graduação, requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Pedro Henrique Del Bianco Hokama
Coorientador: Prof. Dr. Mário César San Felice

ITAJUBÁ
2021

R393 Morais, Renan Barbosa de
 Solucionando o Inverse Power Index Problem / Renan Barbosa
de Morais. – Itajubá-MG, 2021.
 52 f.: il.; PDF.

Trabalho Final de Graduação (Bacharelado em Ciência da
Computação) - Universidade Federal de Itajubá, Campus Itajubá-
MG, 2021.

Orientador: Prof. Dr. Pedro Henrique Del Bianco Hokama.

Inclui referências.

1. Inverse Power Index Problem. 2. Heurística Evolutiva. 3.
Câmara dos Deputados. I. Título. II. Hokama, Pedro Henrique Del
Bianco. III. Universidade Federal de Itajubá.

CDD: 519.3

Men fear thought as they fear nothing else
on earth—more than ruin, more even than
death.

(Bertrand Russell)

Agradecimentos

Aos meus pais pelo apoio e confiança que sempre me deram durante toda a minha vida. Aos amigos e familiares que ao longo desta etapa me encorajaram e me apoiaram. Ao meu orientador Pedro Henrique Del Bianco Hokama pelo apoio contínuo ao meu estudo, pela dedicação e paciência. Aos professores Mário César San Felice e Gabriel Ávila Casalecchi pelas valiosas contribuições dadas durante todo o trabalho. À Universidade Federal de Itajubá, essencial no meu processo de formação acadêmica e profissional. A todos aqueles que contribuíram, de alguma maneira, para a realização deste trabalho.

Resumo

Em muitos problemas de tomada de decisão, alguma forma de votação é aplicada. Em algumas delas, os atores envolvidos podem possuir quantidade de votos distintos, o qual chamamos de pesos. Esse tipo de votação, conhecido como jogo de votação ponderada, é usado em muitos cenários para que cada ator possua uma representação justa ao decidir sobre algum assunto. Porém, muitas vezes o peso atribuído à eles não corresponde à influência real na votação. Este trabalho busca estudar o poder efetivo dos agentes nesse tipo de jogo, através dos power indices, e desenvolver algoritmos computacionais para encontrar jogos de votação ponderada que sejam justos, isto é, em que cada jogador possua uma importância que reflita algum critério objetivo. Em especial, é feito um estudo de caso sobre o poder efetivo na Câmara dos Deputados brasileira, além de encontrar jogos considerados justos.

Abstract

In many decision-making problems, some form of voting is applied. In some of them, the actors involved can have different number of votes, which we call weights. This type of voting, known as weighted voting game, is used in many scenarios so that each actor has a fair representation when deciding on an issue. However, often the weight assigned to them does not correspond to the real influence on the vote. This work seeks to study the voting power of agents in this type of game, through power indices, and develop computational algorithms to find weighted voting games that are fair, ie, in which each player has an importance that reflects some objective criterion. In particular, a case study is carried out on voting power in the Brazilian Chamber of Deputies, in addition to finding games considered fair.

Lista de Figuras

3.1	Comparação entre peso normalizado e Shapley-Shubik power index na distribuição atual.	20
3.2	Comparação entre população normalizada e Shapley-Shubik power index na distribuição atual.	21
3.3	Comparação entre população normalizada e Shapley-Shubik power index na distribuição proporcional à população.	23
4.1	Exemplo da evolução da população k para $k + 1$	32
4.2	Exemplo de cruzamento para produção de um novo cromossomo.	33
5.1	Comparação entre população normalizada e Shapley-Shubik power index na distribuição proposta.	41
5.2	Comparação da razão entre o Shapley-Shubik power index e a população normalizada nos três cenários descritos anteriormente.	42

Lista de Tabelas

2.1	Exemplo de coalizões e jogadores decisivos	15
3.1	Distribuição atual de deputados na Câmara dos Deputados.	19
3.2	Distribuição de deputados proporcional à população em 2019, sem teto ou piso.	22
4.1	Parâmetros necessários para o BRKGA.	32
5.1	População em milhares para os 27 estados membros da UE em 2017.	35
5.2	Instâncias UE_2 a UE_5 para a UE em 2012.	35
5.3	População em milhares para os 25 estados membros da UE em 2004.	36
5.4	Instância UE_{jc} para a UE em 2004.	36
5.5	População para as 27 UFs do Brasil em 2019.	37
5.6	Instância para a Câmara dos Deputados do Brasil.	38
5.7	Resultados para as instâncias UE_3 a UE_8 utilizando PLI.	38
5.8	Resultado para a instância da Câmara dos Deputados utilizando a heurística evolutiva.	39
5.9	Distribuição de deputados obtidos utilizando a heurística evolutiva baseado na <i>linear Shapley rule</i>	40
5.10	Resultados para as instâncias UE_3 a UE_{11} utilizando a heurística evolutiva.	43
5.11	Resultado para a instância UE_{jc} utilizando a heurística evolutiva.	44
5.12	Comparação dos resultados para as instâncias UE_3 a UE_{11}	44
5.13	Comparação dos resultados para a instância UE_{jc}	45

Sumário

1	Introdução	11
2	Fundamentação Teórica	13
2.1	Jogos	13
2.2	Power index	14
2.2.1	Banzhaf power index	14
2.2.2	Shapley-Shubik power index	15
2.2.3	Calculando power indices	15
2.3	Problema inverso	16
3	Descrição do Problema	17
3.1	Poder na Câmara nos Deputados	18
3.1.1	Distribuição atual	19
3.1.2	Distribuição proporcional à população atual	22
3.1.3	Alternativas de distribuição	24
4	Algoritmos	25
4.1	Power index	25
4.1.1	Banzhaf	27
4.1.2	Shapley-Shubik	28
4.2	Problema Inverso	29
4.2.1	Formulação do problema inverso em PLI	30
4.2.2	Heurística evolutiva com BRKGA	31
5	Resultados Experimentais	34
5.1	Instâncias	34
5.1.1	Conselho da União Europeia	34
5.1.2	Câmara dos Deputados do Brasil	36
5.2	Programação Linear Inteira	38
5.3	Heurística Evolutiva	39
5.3.1	Câmara dos Deputados	39
5.3.2	União Europeia	42
5.4	Comparação dos resultados	44
6	Conclusões	47

Capítulo 1

Introdução

Em muitos problemas de tomada de decisão que envolvem diversos agentes, algum tipo de votação é aplicada. Em diversas delas, existe a necessidade que os diferentes atores envolvidos possuam diferentes quantidade de votos, o qual chamamos de pesos. Esse tipo de votação, conhecido como jogo de votação ponderada, é usado em muitos cenários, como eleições, decisões de stakeholders em organizações e deliberações de instituições políticas.

Um das principais razões para o uso de jogos de votação ponderada é que cada agente (chamado de jogador) possua uma representação justa ao decidir sobre algum assunto. Porém, muitas vezes o peso atribuído a eles não corresponde à influência real na votação.

Para mensurar a influência dos jogadores em jogos de votação ponderada, utilizamos o conceito de power index. O power index de um jogador corresponde ao poder que o mesmo tem de afetar o resultado de uma votação de acordo com seu peso. Assim, é possível quantificar as discrepâncias de representatividade existentes entre o peso e seu poder. Muito se estuda sobre esse tema, principalmente em relação ao Colégio Eleitoral nos Estados Unidos da América [Banzhaf III, 1968, Miller, 2013] e à composição do Conselho da União Europeia [Felsenthal e Machover, 1997, Laruelle e Widgrén, 1998]. No Brasil, Lima e Ramos [2010] utilizam power indices para estudar os estados e regiões brasileiras, além de discutir as implicações econômicas.

Mostraremos que os pesos dos jogadores em um jogo muitas vezes não representam o seu real poder, surgindo outra questão: como definir o peso que resulta em um power index desejado? Para isso precisamos resolver o Problema Inverso dos Power Indices (IPIP, do inglês Inverse Power Index Problem).

Esse problema é importante em cenários onde desejamos definir um jogo em que cada jogador possua uma representação considerada justa. Uma inspiração para esta pesquisa, Zyczkowski e Slomczynski [2004] possuem importante trabalho no desenvolvimento de um sistema de votação para o Conselho Europeu na qual utilizam outra estratégia para desenvolver um jogo justo pela ausência de como resolver claramente o problema inverso.

Até onde sabemos pouco trabalho já foi feito para resolver esse problema. Existem soluções propostas para o problema inverso que obtêm soluções sem garantia de otimalidade [Leech, 2002a, Aziz et al., 2007, Fatima et al., 2008] e também algoritmos exatos [de Keijzer, 2009, Kurz, 2012].

O objetivo desse trabalho é investigar esse problema e desenvolver algoritmos que possam encontrar jogos em que os jogadores possuam power indices o mais próximo possível

de valores desejados. Desenvolvemos dois algoritmos para solucionar o problema inverso: um em programação linear inteira e outro através de uma heurística evolutiva com o framework BRKGA. Note que para resolver o problema inverso também precisamos calcular os power indices e, por isso, também desenvolvemos algoritmos para esse problema. Além disso, também estudamos o que pode ser definido como um jogo justo, isto é, como definimos power indices que desejamos que os jogadores de um jogo possuam.

Outro propósito importante desse trabalho é o estudo sobre a representação das unidades federativas na Câmara dos Deputados do Brasil. Nesse contexto, um dos principais pontos de debate é o da não-proporcionalidade entre a população de uma circunscrição eleitoral e o seu número de representantes na Câmara dos Deputados, estudado por diversos autores da ciência política [Nicolau, 1997, Soares e Lourenço, 2004, de Oliveira, 2004, Carmo et al., 2012]. Sob a ótica dos power indices, contribuímos para este debate calculando o power index de cada estado da federação brasileira considerando a atual configuração da Câmara dos Deputados, bem como uma distribuição proporcional à população em 2019, na qual os limitantes superior e inferior sejam abolidos. Diante disso, desenvolvemos uma nova distribuição através da solução do problema inverso que leva em consideração os fatores examinados.

O trabalho está dividido em 5 capítulos além desta Introdução. No Capítulo 2, apresentamos o conceito de jogos, mais precisamente os de votação ponderada, os power indices e o problema inverso. No Capítulo 3, descrevemos o problema relacionado ao desenvolvimento de jogos de votação ponderada e, em especial, pormenorizamos o problema da proporcionalidade na Câmara dos Deputados. Em seguida apresentamos no Capítulo 4 os algoritmos desenvolvidos para o power index e para o problema inverso. O Capítulo 5 apresenta os resultados experimentais obtidos com a aplicação dos algoritmos e suas comparações. Por fim, concluímos o trabalho no Capítulo 6 e discutimos trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo apresentamos os conceitos necessários para entendimento deste trabalho. As próximas seções descrevem o conceito de jogo, em especial o jogo de votação ponderada, além dos power indices e o problema inverso dos power indices.

2.1 Jogos

A teoria dos jogos pode ser definida como o estudo de modelos matemáticos de conflito e cooperação entre tomadores de decisão racionais e inteligentes [Myerson, 2013]. A teoria dos jogos é importante para analisar situações estratégicas, onde existem indivíduos tomando decisões que afetam uns aos outros. Essa situação estratégica é chamada de **jogo**, e os agentes racionais que participam dela são chamados de **jogadores**.

Estamos interessados nos chamados **jogos cooperativos**, onde estudamos o que acontece quando os jogadores colaboram, e como deveriam colaborar para atingir um resultado desejado.

Assim, definimos um jogo cooperativo como um par (N, v) , onde N é um conjunto finito de n jogadores e $v : 2^N \rightarrow \mathbb{R}^+$ é a função característica.

Como tratamos de jogos cooperativos, um conceito importante é o da **coalizão**. Um subconjunto $S \subseteq N$ é chamado de coalizão e $v(S)$ é o ganho da coalizão S quando coopera. O conjunto N é chamado de grande coalizão.

Jogos simples são um subconjunto dos jogos cooperativos, onde cada coalizão somente ganha ou perde, tendo $v(S) = 1$ caso seja uma coalizão vencedora e $v(S) = 0$ se perdedora. O conjunto de coalizões vencedoras \mathcal{W} contém todas as coalizões se $v(S) = 1$.

Jogos monótonos são os jogos simples em que uma coalizão vencedora não pode se tornar perdedora ao adicionar um jogador e de modo semelhante, uma coalizão perdedora não se torna vencedora ao se retirar um jogador.

Por fim, tratamos dos **jogos de votação ponderada**, que são os comumente utilizados em situações de tomadas de decisão. Definimos um jogo de votação ponderada (N, v, W, q) como um jogo monótono em que $W = \{w_1, w_2, \dots, w_n\}$, no qual cada jogador i possui peso $w_i \in \mathbb{R}$, e uma quota $q \in \mathbb{R}$ define se uma coalizão está perdendo ou

ganhando através de sua função característica $v : 2^N \rightarrow \{0, 1\}$, em que:

$$v(S) = \begin{cases} 1 & \text{se } \sum_{i \in S} w_i \geq q, \\ 0 & \text{caso contrário.} \end{cases} \quad (2.1)$$

2.2 Power index

Uma das principais razões para o uso de jogos de votação ponderada é que cada jogador possua uma representação justa ao decidir sobre algum assunto. Porém, muitas vezes o peso atribuído à eles não corresponde à influência real na votação.

Um exemplo simples para ilustrar que o peso de um jogador muitas vezes não é igual ao seu poder, é um jogo onde a única coalizão vencedora é a grande coalizão. Isto é, dentre todas as coalizões, cada jogador só faz parte de uma coalizão vencedora: a que contém todos os outros jogadores. Portanto, o poder de todos eles é equivalente, embora o peso não necessariamente também seja igual.

Diferentes power indices surgiram desta necessidade de se medir a influência de um jogador em um jogo. Dois dos principais power indices são o Banzhaf Power Index [Banzhaf III, 1964] e o Shapley-Shubik Power Index [Shapley e Shubik, 1954]. Também existem na literatura outros power indices menos difundidos como os de Coleman, Deegan-Packel e Holler, para citar alguns exemplos.

É importante dizer que os power indices não fazem suposições sobre as preferências e opiniões dos jogadores em um jogo. De modo geral, podemos dizer que a probabilidade de um jogador fazer parte de uma coalizão é igual entre todas elas. Além disso, embora neste trabalho o power index seja aplicado em jogos de votação ponderada, podemos também utilizá-los no superconjunto dos jogos cooperativos simples.

Nas próximas subseções definimos os power indices de Banzhaf e Shapley-Shubik, os quais utilizamos nesse trabalho.

2.2.1 Banzhaf power index

O Banzhaf power index calcula a quantidade de coalizões em que um jogador é decisivo. Isto ocorre se uma coalizão é vencedora com este jogador e deixa de ser retirando-o deste conjunto. O Banzhaf *index* β'_i para cada jogador i em um jogo é dado por:

$$\beta'_i = |\{S \subseteq N - \{i\} : v(S) = 0 \wedge v(S \cup \{i\}) = 1\}|. \quad (2.2)$$

Em busca de se representar de uma forma mais natural, o Banzhaf power index pode ainda ser normalizado [Dubey e Shapley, 1979], definido como:

$$\beta_i = \frac{\beta'_i}{\sum_{j=1}^n \beta'_j}. \quad (2.3)$$

Tomemos como exemplo um jogo com 3 jogadores, com $w_1 = 3, w_2 = 2, w_3 = 1$, e $q = 4$. Enumeramos todas as coalizões possíveis nesse jogo e verificamos os jogadores decisivos. A Tabela 2.1 descreve as coalizões, mostrando a soma dos pesos e os jogadores decisivos em cada coalizão.

Coalizão	Pesos	Soma	Decisivos
S ₁	{w ₁ }	3	
S ₂	{w ₂ }	2	
S ₃	{w ₃ }	1	
S ₄	{w ₁ , w ₂ }	3+2=5	{w ₁ , w ₂ }
S ₅	{w ₁ , w ₃ }	3+1=4	{w ₁ , w ₃ }
S ₆	{w ₂ , w ₃ }	2+1=3	
S ₇	{w ₁ , w ₂ , w ₃ }	3+2+1=6	{w ₁ }

Tabela 2.1: Exemplo de coalizões e jogadores decisivos

A operação 2.2 é realizada para todas as coalizões e para cada jogador presente nela. Então, exemplificando, podemos verificar os jogadores da coalizão S_4 , removendo em um primeiro momento o peso w_1 da soma, obtendo uma soma igual a 3, menor que a quota. Em seguida, fazemos o mesmo para o peso w_2 , alcançando soma 2, também menor que a quota. Logo, nessa coalizão temos dois jogadores decisivos, somando uma unidade à β'_i para cada um deles. Fazemos essa operação para todas as coalizões, e no final, normalizamos o power index com a equação 2.3, obtendo os power indices $\beta_1 = 0.6$, $\beta_2 = 0.2$ e $\beta_3 = 0.2$.

2.2.2 Shapley-Shubik power index

Seja Π o conjunto de permutações de n jogadores em um jogo, $\pi \in \Pi$ uma permutação e $\Upsilon_i \subseteq \pi$ o subconjunto que precede o jogador i em π . Para cada $\pi \in \Pi$, um jogador i é o pivô se ele é o primeiro a tornar a coalizão vencedora. Assim, φ_i é a quantidade de vezes em que isso ocorre, dado por:

$$\varphi_i = \sum_{\pi \in \Pi} v(\Upsilon_i \cup \{i\}) - v(\Upsilon_i). \quad (2.4)$$

φ_i também é chamado de *raw Shapley-Shubik index*. O Shapley-Shubik power index do jogador i é dado por:

$$\phi_i = \frac{\varphi_i}{n!}. \quad (2.5)$$

Uma maneira mais conveniente para calcular esse power index é dado por:

$$\phi_i = \sum_{S \notin \mathcal{W}: S \cup i \in \mathcal{W}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}. \quad (2.6)$$

2.2.3 Calculando power indices

Uma implementação simples para calcular os power indices de Banzhaf e Shapley-Shubik é, respectivamente, a enumeração de todas possíveis coalizões e todas permutações de jogadores, em ambos os casos métodos exponenciais. Para o Banzhaf index, o número de possíveis coalizações é exponencial no número de jogadores, enquanto para o seguinte

seriam $n!$ permutações a se enumerar. [Bachrach et al., 2010]. As implementações tem complexidade de tempo $O(n^2 2^n)$ e $O(n 2^n)$ [de Keijzer, 2008].

Uma melhor abordagem proposta por vários autores [Mann e Shapley, 1962, Tannenbaum, 1997, Bilbao et al., 2000], faz uso de funções geradoras. Ainda Matsui e Matsui [2000] propõem um outro algoritmo usando programação dinâmica e o método de Monte Carlo para resolução.

A melhor implementação conhecida é a de Klinz e Woeginger [2005], que consiste em uma derivação do método de enumeração direta, com uso de particionamento. Possui complexidade de tempo $O(n^2(\sqrt{2})^n)$ para Banzhaf index e $O(n(\sqrt{2})^n)$ para Shapley-Shubik index.

Prasad e Kelly [1990], Matsui e Matsui [2001] provam que ambos os problemas, tanto o Banzhaf power index quanto o Shapley-Shubik power index são problemas NP-completos quando

$$\frac{1}{2} \sum_{i=1}^n w_i < q \leq \sum_{i=1}^n w_i. \quad (2.7)$$

2.3 Problema inverso

Como descrito anteriormente, dado um jogo (N, v, W, q) podemos calcular os power indices a partir dos pesos dos jogadores. Esse problema é chamado de *forward problem*. Mas e se quisermos a partir de power indices desejados encontrar um jogo (N, v, W, q) que o represente?

Para encontrar pesos e quota que correspondem aos power indices desejados, precisamos resolver o Problema Inverso dos Power Indices. Seja n o número de jogadores e $D = \{d_1, \dots, d_n\}$ o conjunto de power indices desejado, com $d_i \geq 0$ e $\sum_{i=1}^n d_i = 1$. Queremos encontrar um conjunto de pesos $W = \{w_1, \dots, w_n\}$ e uma quota q , com $q > 0$ e $w_i \geq 0$, onde minimizamos uma função pretendida. Nesse trabalho, utilizamos

$$\sum_{i=1}^n |d_i - p_i| \quad (2.8)$$

para todo jogador $1 \leq i \leq n$, na qual cada p_i representa o power index do jogador de peso w_i . É importante notar que nem sempre existe um jogo que resulta nos power indices desejados.

Existem soluções propostas para o problema inverso que obtêm soluções sem garantia de otimalidade [Leech, 2002a, Aziz et al., 2007, Fatima et al., 2008]. Até onde sabemos, somente dois autores propõem algoritmos exatos [de Keijzer, 2009, Kurz, 2012]. Note que, uma solução ótima para esse problema corresponde a uma configuração que leva aos power indices desejados ou, se estes não são alcançáveis, que minimiza a distância até eles.

Capítulo 3

Descrição do Problema

Os jogos de votação ponderada são de fundamental importância no mundo atual, em especial utilizados em muitos órgãos políticos decisórios. Dois dos exemplos mais famosos onde esse tipo de jogo é aplicado são as eleições presidenciais nos Estados Unidos da América e o Conselho da União Europeia. Além disso, um jogo de votação ponderada também está muito presente em organizações onde sócios ou acionistas possuem poder de voto distintos.

Sabemos que os pesos dos jogadores em um jogo muitas vezes não representam o seu real poder. Um popular exemplo onde isso ocorre é o caso do Conselho da Comunidade Econômica Europeia, formado em 1957, composto por Alemanha, França, Itália, Bélgica, Holanda e Luxemburgo.

Nesse conselho, cada um dos três primeiros países possuíam peso 4, Bélgica e Holanda tinham peso 2 cada e Luxemburgo tinha peso 1. Para alguma resolução ser aprovada, era necessário um total de 12. Embora esta distribuição de votos possa parecer justa e proporcional ao tamanho, poder ou população de cada país, Luxemburgo não possuía nenhuma influência nas decisões. Isto porque em qualquer situação seu voto não tinha o poder de alterar o resultado da votação, isto é, qualquer coalizão que ganhava ou perdia sem Luxemburgo continuava a ganhar ou perder com Luxemburgo.

Neste trabalho, em vez de analisar o poder de cada agente em um jogo, focamos no problema ao qual chamamos de problema inverso dos power indices. Nesse problema, nosso objetivo é encontrar jogos em que os power indices dos jogadores sejam o mais próximo possível de power indices desejados.

Para isso, desenvolvemos dois algoritmos para a solução do problema inverso: um exato, baseado no trabalho de Kurz [2012] e outro sem garantia de otimalidade. Além disso, também implementamos soluções para o cálculo do power index, fundamental para a resolução do problema inverso.

Parte importante desse trabalho é o estudo de caso realizado no contexto da Câmara dos Deputados, onde tratamos o discutido problema da desproporcionalidade de representantes em relação à população das unidades federativas no Brasil.

3.1 Poder na Câmara nos Deputados

A proporcionalidade na representação política é tema importante para a Ciência Política, particularmente para a literatura que pesquisa a democracia e os sistemas políticos. No Brasil, um dos principais pontos de debate é o da não-proporcionalidade entre a população de uma circunscrição eleitoral e o seu número de representantes na Câmara dos Deputados. Em outras palavras, o número de deputados de um estado não é exatamente proporcional ao número de habitantes, o que fere a máxima “uma cabeça, um voto”.

Existem duas formas mais comuns de violação da proporcionalidade entre habitantes e representantes [Nicolau, 1997]: 1) a ausência de revisão periódica do número de representantes de cada circunscrição eleitoral em virtude de mudanças demográficas da população e 2) a inclusão de limites mínimos e máximos de deputados por circunscrição. A inclusão dos limites está no Art. 45, §1º, da Constituição Federal: *"O número total de Deputados, bem como a representação por Estado e pelo Distrito Federal, será estabelecido por lei complementar, proporcionalmente à população, procedendo-se aos ajustes necessários, no ano anterior às eleições, para que nenhuma daquelas unidades da Federação tenha menos de oito ou mais de setenta Deputados."*

O caso mais utilizado para ilustrar a distorção entre número de representantes e população é comparar os estados de São Paulo e Roraima. No primeiro, o número de habitantes por deputado é de quase 656 mil, enquanto no segundo é próximo de 76 mil pessoas. Ou seja, em São Paulo cada deputado representa aproximadamente 8,6 vezes mais pessoas que em Roraima (de acordo com a população em 2019 [IBGE, 2019]). Essa disparidade tem dividido os estudiosos da Ciência Política. Por um lado, o principal argumento a favor de uma representação estritamente proporcional em termos de população e cadeiras parlamentares seria o seu caráter mais democrático. Como visto no exemplo acima, ele infringe diretamente o princípio igualitário da democracia, no qual cada indivíduo tem o mesmo valor em termos de voto.

Por outro lado, a principal objeção a esse argumento é que a democracia não se limita ao princípio da igualdade e a modelos majoritários de decisão. De acordo com Soares e Lourenço [2004], é possível defender uma outra concepção “que privilegie a inclusão e o consenso, dando expressão a interesses relevantes presentes na sociedade, mesmo que minoritários”. A introdução de limites seria positiva, pois garantiria a representatividade efetiva de estados e populações minoritários, que de outra forma não teriam condições de fazer valer suas demandas [King, 1993, Lijphart, 2012].

As discussões em torno da desproporcionalidade têm motivado estudiosos a desenvolver pesquisas empíricas que visem não só identificar as causas e consequências do fenômeno, como também analisar o impacto que a regra introduz no processo político [de Oliveira, 2004, Carmo et al., 2012].

Preocupações semelhantes, envolvendo encontrar uma representação justa em um cenário com diversos agentes, motivaram estudos utilizando o conceito do power index [Penrose, 1946, Banzhaf III, 1966, 1968, Miller, 2013]. Em particular, diversos desses estudos são relacionados à composição do Conselho Europeu [Laruelle e Widgrén, 1998, Leech, 2002b]. No Brasil, Lima e Ramos [2010] utilizaram power índices para estudar os estados e regiões brasileiras, além de discutir as implicações econômicas. Neste trabalho, estamos

interessados em estudar o poder na Câmara dos Deputados sob a perspectiva dos power indices. Então, considerando a representação de cada unidade federativa na Câmara dos Deputados, mensuramos o poder efetivo de cada uma delas por meio dos power indices.

3.1.1 Distribuição atual

A Tabela 3.1 mostra o número atual de cadeiras para cada estado da federação. Note que, a atual distribuição leva em conta a população de 1993, conforme estabelecido na Lei Complementar nº 78 de 30/12/1993.

UF	#Dep.	UF	#Dep.	UF	#Dep.
Acre	8	Maranhão	18	Rio de Janeiro	46
Alagoas	9	Mato Grosso	8	Rio Grande do Norte	8
Amapá	8	Mato Grosso do Sul	8	Rio Grande do Sul	31
Amazonas	8	Minas Gerais	53	Rondônia	8
Bahia	39	Pará	17	Roraima	8
Ceará	22	Paraíba	12	Santa Catarina	16
Distrito Federal	8	Paraná	30	São Paulo	70
Espírito Santo	10	Pernambuco	25	Sergipe	8
Goiás	17	Piauí	10	Tocantins	8

Tabela 3.1: Distribuição atual de deputados na Câmara dos Deputados.

A partir dos pesos, representados pelo número de deputados, e da quota, representado pelo número de 257 deputados necessários para aprovação de uma lei complementar, ou de um projeto de lei e medida provisória com quórum máximo, podemos calcular o poder efetivo de cada estado. A Figura 3.1 apresenta a comparação entre o Shapley-Shubik power index de cada unidade federativa e seu peso normalizado.

Notamos que o número de representantes (peso) que cada estado possui não corresponde a seu poder efetivo. Os quatro estados com maior população (São Paulo, Minas Gerais, Rio de Janeiro e Bahia) possuem poder efetivo maior do que seu peso, enquanto todos os outros têm poder menor.

Podemos então comparar os power indices com a população, conforme ilustra a Figura 3.2. Essa comparação nos mostra que apesar do estado de São Paulo possuir um poder efetivo superior a seu número de deputados normalizado, seu power index ainda é desproporcionalmente pequeno em relação a sua população. Por outro lado, os estados atingidos pelo piso apresentam um poder efetivo bem maior em relação a sua população.

Assim, notamos que a presença de um teto e um piso, com o objetivo de restringir a influência das unidades federativas com população muito grande e de ampliar a participação daquelas com população pequena causa diversas distorções. Será que, abolindo o teto e o piso conseguiríamos uma representação mais justa?

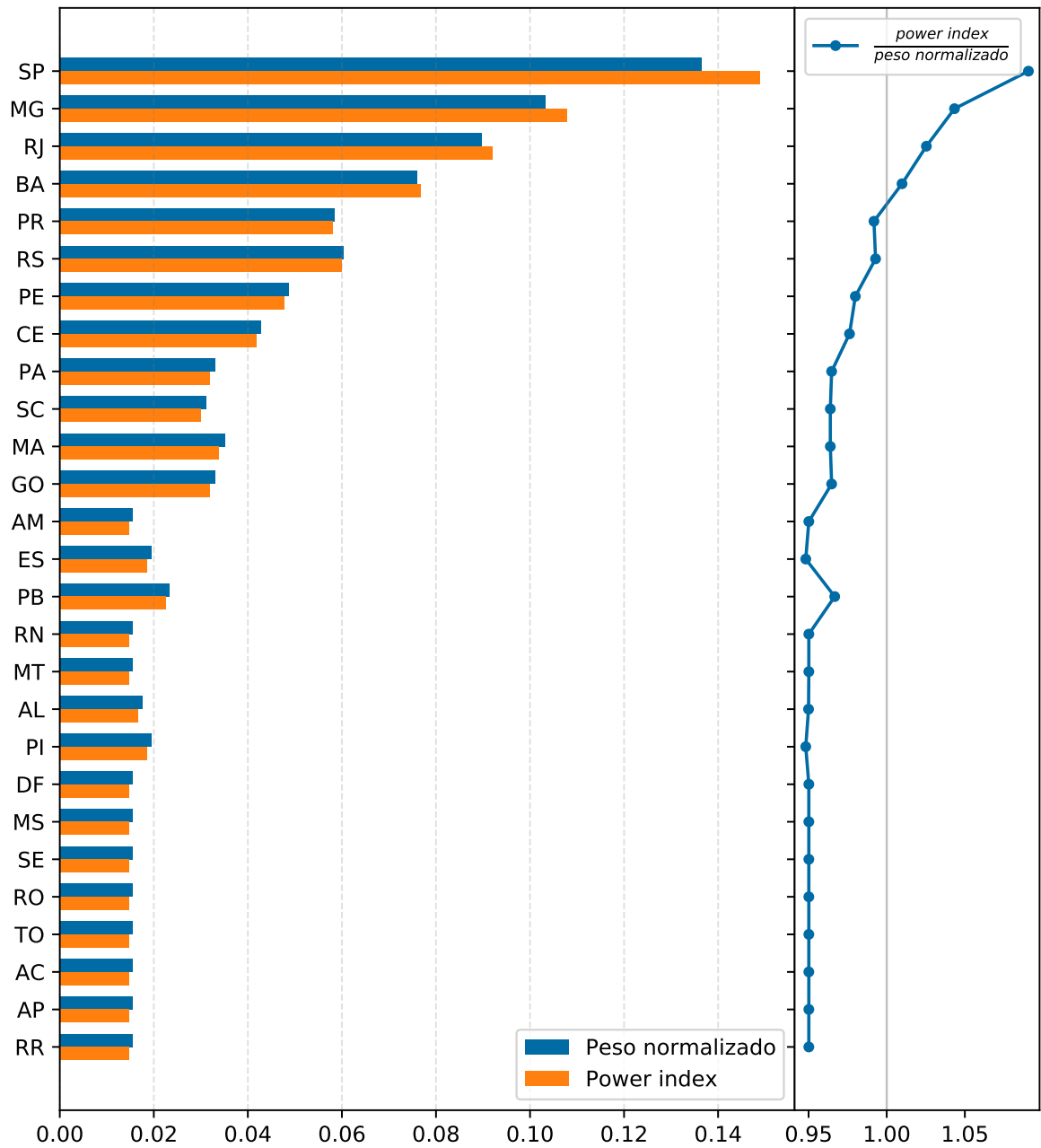


Figura 3.1: Comparação entre peso normalizado e Shapley-Shubik power index na distribuição atual.

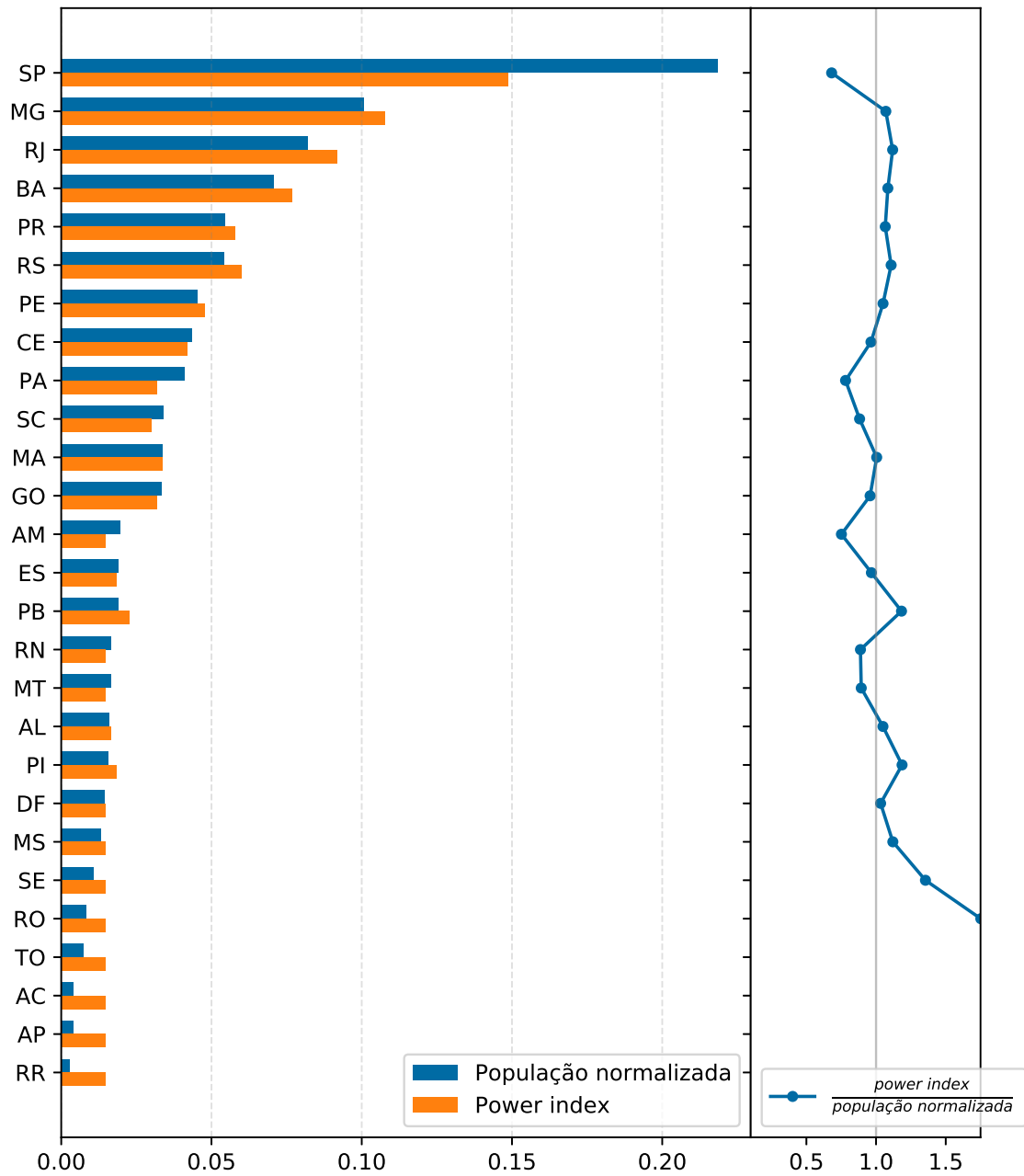


Figura 3.2: Comparação entre população normalizada e Shapley-Shubik power index na distribuição atual.

3.1.2 Distribuição proporcional à população atual

Uma das críticas frequentes ao atual sistema de distribuição é que os habitantes em São Paulo não valem tanto quanto os dos outros estados. Analisamos então uma distribuição em que o número de representantes é estritamente proporcional à população de cada estado, sem limitantes inferiores e superiores. Mostraremos que essa distribuição também possui problemas. Mantendo o número total de 513 deputados, a Tabela 3.2 ilustra a quantidade de parlamentares por estado, considerando a população de 2019.

Estado	Diferença p/		Estado	Diferença p/	
	#Dep.	Atual		#Dep.	Atual
Acre	2	-6	Paraíba	10	-2
Alagoas	8	-1	Paraná	28	-2
Amapá	2	-6	Pernambuco	23	-2
Amazonas	10	+2	Piauí	8	-2
Bahia	36	-3	Rio de Janeiro	42	-4
Ceará	22		Rio Grande do Norte	9	+1
Distrito Federal	7	-1	Rio Grande do Sul	28	-3
Espírito Santo	10		Rondônia	4	-4
Goiás	17		Roraima	1	-7
Maranhão	17	-1	Santa Catarina	18	+2
Mato Grosso	9	+1	São Paulo	112	+42
Mato Grosso do Sul	7	-1	Sergipe	6	-2
Minas Gerais	52	-1	Tocantins	4	-4
Pará	21	+4			

Tabela 3.2: Distribuição de deputados proporcional à população em 2019, sem teto ou piso.

É importante observar que o produto da população normalizada por 513 não resulta em números exatos. Para resolver isso, arredondamos-os para o inteiro mais próximo. Ao arredondar obtivemos um total de 512 cadeiras, e distribuimos a última cadeira ao estado que estava mais próximo de ser aproximado para cima, o estado de Santa Catarina.

Grande parte das mudanças se deve à diferença da população em 2019 comparada a de 1993. Como exemplo, os estados do Amazonas e da Paraíba, que possuem atualmente 8 e 12 deputados respectivamente, teriam atualizações em sentidos opostos, se igualando no número de 10 deputados.

Vemos que o estado de São Paulo com folga é o que mais possui alteração no número de deputados, ganhando 42 novas cadeiras. Os estados beneficiados pelo piso sofrem reduções significativas, com Roraima possuindo apenas uma cadeira nessa nova configuração. Mas será que essa distribuição ficou justa?

A Figura 3.3 ilustra a relação entre o Shapley-Shubik power index de cada estado com sua população normalizada. Intuitivamente, a distribuição estritamente proporcional traria um cenário mais justo, igualando o poder dos cidadãos de cada estado. No entanto, analisando o poder efetivo vemos que ainda existe uma grande disparidade. Em particular, São Paulo possui aproximadamente 26,15% do poder enquanto sua população representa

21,85% da população brasileira. Em verdade, todos os outros estados têm poder efetivo inferior a suas populações.

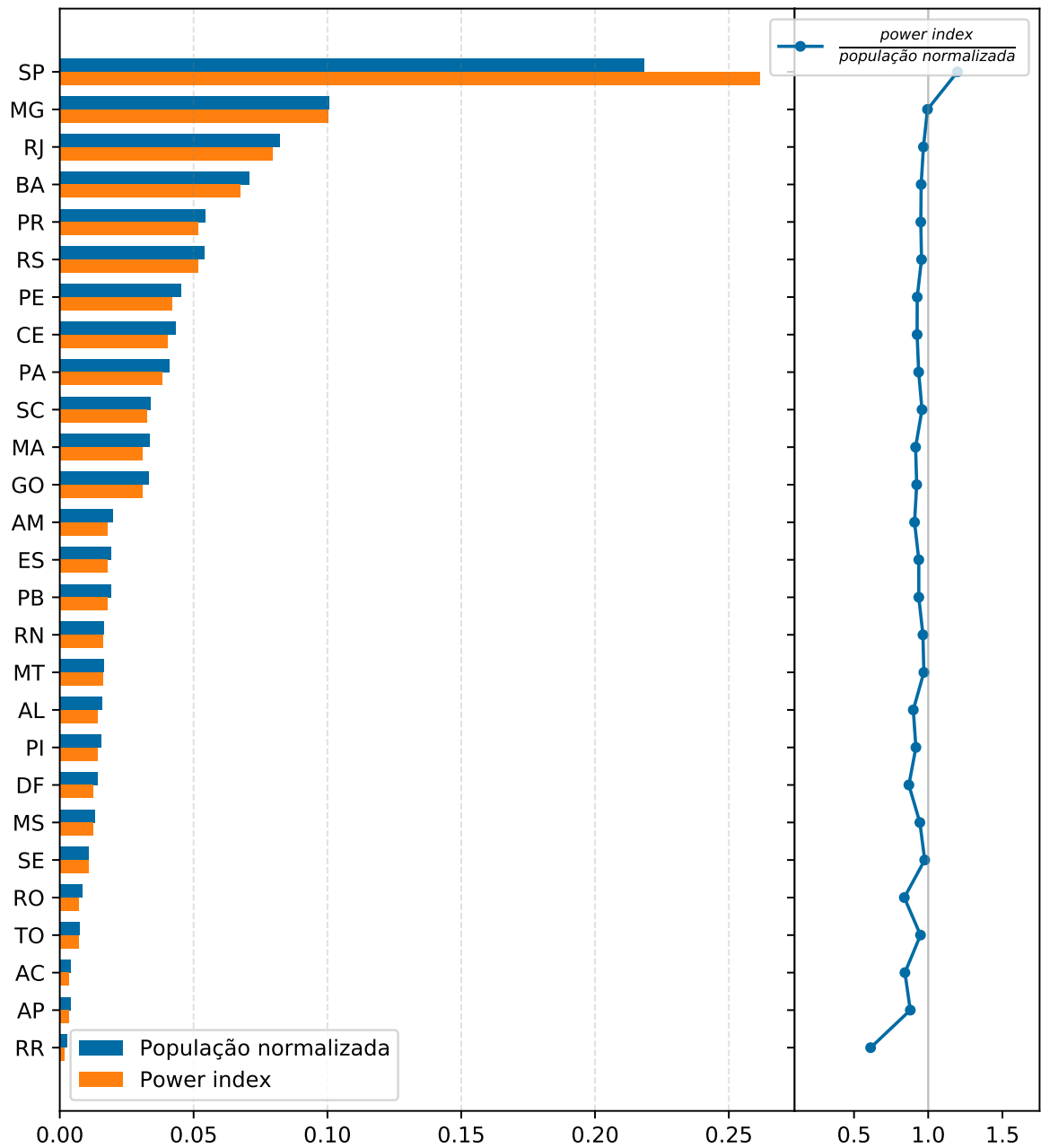


Figura 3.3: Comparação entre população normalizada e Shapley-Shubik power index na distribuição proporcional à população.

Aqui não comparamos os valores dos power índices com os pesos, como fizemos na Figura 3.1. Como o número de deputados é proporcional à população de cada unidade federativa, a relação entre peso e poder efetivo é semelhante à mostrada na Figura 3.3 (entre população e poder efetivo), com pequenas divergências causadas por arredondamentos.

3.1.3 Alternativas de distribuição

Vimos que, atualmente a Câmara dos Deputados possui uma distribuição de cadeiras com muitas deturpações. Porém, uma proposta estritamente proporcional à população causaria distorções muito significativas no poder efetivo dos estados, principalmente em favor de São Paulo.

No parlamento existem várias propostas que, ao propor a redução do número de deputados, acabam alterando as distorções pelo teto e piso e corrigindo a desatualização populacional, como as PEC 12/2019, PEC 431/2018, PEC 405/2018 e PEC 106/2015, para citar alguns exemplos. Em relação à atualização populacional, a Resolução 23.389/2013 do Tribunal Superior Eleitoral redefiniu a distribuição de deputados considerando a população da época, porém acabou não vigorando após ser julgada inconstitucional pelo Supremo Tribunal Federal.

Existe, na literatura, diversas discussões sobre alternativas de distribuição na Câmara dos Deputados que seriam mais justas. Propostas baseadas na proporcionalidade regressiva [Carmo et al., 2012], na *penrose square root rule* [Lima e Ramos, 2010] ou até mesmo no aumento do número de deputados [de Oliveira, 2004] são algumas das soluções propostas. No contexto dos power indices, a *penrose square root rule* é muito utilizada em cenários para obter uma distribuição de poder justa. Nessa regra, de forma resumida, um agente deve ter uma probabilidade de ser decisivo em uma votação proporcional à raiz quadrada de sua população.

No Capítulo 5 propomos uma distribuição alternativa, que contempla os problemas de poder efetivo já apresentados, onde nos baseamos na *linear Shapley rule* [Kurz et al., 2017], onde uma distribuição justa se dá quando os power indices são proporcionais a população.

Capítulo 4

Algoritmos

Neste capítulo apresentamos os algoritmos desenvolvidos neste trabalho. Primeiro, descrevemos os algoritmos para o cálculo dos power indices de Banzhaf e Shapley-Shubik. A seguir, expomos as abordagens de solução do problema inverso, mostrando a formulação em Programação Linear Inteira e, logo após, a heurística evolutiva desenvolvida.

4.1 Power index

Os algoritmos desenvolvidos para cálculo do power index, tanto Banzhaf quanto Shapley-Shubik, são muito semelhantes. Nesse problema temos como entrada um conjunto de jogadores, com seus pesos, e uma quota, originados a partir de um jogo (N, v, W, q) . A saída é um conjunto de power indices. É importante notar que o conjunto contendo os pesos não é em todo caso um conjunto ordenado.

De maneira trivial, podemos enumerar todas as coalizões possíveis nesse jogo e então encontrar os índices de Banzhaf e Shapley-Shubik a partir delas. No algoritmo, para ambos os power indices, iteramos sobre o número de jogadores que podem existir nas coalizões do jogo e geramos as combinações possíveis para este número de jogadores, isto é, iteramos do número de jogadores de uma coalizão vazia (sem nenhum jogador) até o número de jogadores em uma grande coalizão (todos os jogadores) e geramos todas as combinações para cada número.

Entretanto, essa operação é extremamente custosa, de crescimento exponencial. Enquanto um jogo de 4 jogadores possui 16 coalizões, um jogo com 20 jogadores possui mais de 1 milhão. Mas, como definido na Seção 2.2, sabemos que as coalizões que de fato definem os power indices são aquelas onde $S \notin \mathcal{W}$ e $S \cup i \in \mathcal{W}$, para qualquer jogador i . Por isso, precisaria-se somente enumerar essas coalizões. Se limitar somente à essas coalizões na enumeração é um desafio matemático e computacional por si só, que foge do escopo desse trabalho. Entretanto, conseguimos desenvolver métodos para eliminar boa parte das coalizões que sabemos não serem necessárias na enumeração.

Para isso, se encontra a coalizão vencedora S_{nmax} , com peso mais próximo da quota, composta exclusivamente pelos jogadores de menor peso. A partir disso, percebe-se que todas as coalizões S' em que $|S'| > |S_{nmax}|$ serão coalizões em que $v(S') = 1$ e $v(S' - \{i\}) = 1, \forall i \in S'$. Isso significa que qualquer coalizão S' não influencia no power index, e portanto

não precisamos enumerá-las.

Para encontrar $|S_{nmax}|$, se soma os pesos dos jogadores, do menor para o maior, e a execução é interrompida assim que é identificada a primeira oportunidade em que esse valor é maior do que a quota, encontrando o valor desejado. Essa busca está ilustrada no Algoritmo 1.

Algoritmo 1: MaxCoalizao

Entrada: O conjunto de pesos W e a quota q .

Saída: O número máximo de jogadores em uma coalizão.

```

1 ordena  $W$  em ordem crescente;
2  $max \leftarrow 0$ ;
3  $tw \leftarrow 0$ ;
4 para todo  $(w,i) \in W$  faça
5    $tw \leftarrow tw + w$ ;
6   se  $tw \geq q$  então
7      $max \leftarrow i$ ;
8     encerra;
9 devolva  $max$ ;
```

Importante notar que o peso somado de todos os jogadores em um jogo é no mínimo igual a quota, pois por definição, nossos jogos têm $v(N) = 1$.

De forma semelhante, é possível encontrar a coalizão vencedora S_{nmin} , com peso mais próximo da quota, composta exclusivamente pelos jogadores de maior peso. Analogamente, todas as coalizões S'' em que $|S''| < |S_{nmin}|$ serão coalizões em que $v(S'') = 0$ e, pela definição de jogos monótonos, $v(S'' - \{i\}) = 0, \forall i \in S''$. Então, não precisamos enumerar qualquer coalizão S'' , já que também não influenciam no power index.

Algoritmo 2: MinCoalizao

Entrada: O conjunto de pesos W e a quota q .

Saída: O número mínimo de jogadores em uma coalizão.

```

1 ordena  $W$  em ordem decrescente;
2  $min \leftarrow 0$ ;
3  $tw \leftarrow 0$ ;
4 para todo  $(w,i) \in W$  faça
5    $tw \leftarrow tw + w$ ;
6   se  $tw \geq q$  então
7      $min \leftarrow i$ ;
8     encerra;
9 devolva  $min$ ;
```

No Algoritmo 2, os pesos são ordenados em ordem decrescente, e somando os pesos um a um, ao identificar a primeira coalizão vencedora (em que a soma dos pesos é maior que a quota) a execução é interrompida, encontrando $|S_{nmin}|$.

As subseções 4.1.1 e 4.1.2 mostram os algoritmos para os power indices de Banzhaf e Shapley-Shubik, com as devidas diferenças existentes.

4.1.1 Banzhaf

No cálculo do power index de Banzhaf, enumeramos todos as coalizões com tamanhos entre $|S_{nmin}|$ e $|S_{nmax}|$ com a ajuda dos métodos descrito anteriormente, e para cada uma das coalizões utilizamos o cálculo descrito na Seção 2.2.1, onde um jogador é decisivo se ao retirá-lo de uma coalizão vencedora, essa coalizão passa a ser perdedora. O Algoritmo 3 apresenta o cálculo do Banzhaf power index β_i de todo jogador i .

Algoritmo 3: Banzhaf power index

Entrada: O conjunto de pesos W e a quota q .

Saída: O conjunto de power index encontrado.

```

1  $\beta \leftarrow \{\beta_1 = 0, \beta_2 = 0, \dots, \beta_n = 0\}$ ;
2  $min \leftarrow MinCoalizao(W, q)$ ;
3  $max \leftarrow MaxCoalizao(W, q)$ ;
4 ordena  $W$  em ordem decrecente;
5 para  $i \leftarrow min$  até  $max$  faça
6    $S_n \leftarrow Combinacoes(W, i)$ ;
7   para todo  $S \in S_n$  faça
8      $cw \leftarrow \sum_{w \in S} w$ ;
9     se  $cw \geq q$  então
10      para todo  $w \in S$  faça
11        se  $cw - w < q$  então
12           $\beta_i \leftarrow \beta_i + 1$ ;
13        senão
14          encerra;
15  $total \leftarrow \sum_{\beta_i \in \beta} \beta_i$ ;
16 para todo  $\beta_i \in \beta$  faça
17    $\beta_i \leftarrow \beta_i / total$ 
18 devolva  $\beta$ ;
```

As linhas 2 e 3 do algoritmo fazem uso da função descrita anteriormente, que calculam o número máximo $|S_{nmax}|$ e mínimo $|S_{nmin}|$ que precisamos iterar para gerar as coalizões. A seguir, o algoritmo ordena os pesos em ordem decrescente na linha 4.

O loop na linha 5 itera entre os valores $|S_{nmin}|$ e $|S_{nmax}|$ encontrados. A linha 6 gera as coalizões através de combinações com um número de jogadores i e todos os jogadores presentes. Note que aqui não detalhamos os métodos computacionais utilizados para gerar as combinações (a geração é simples, e a maioria das linguagens de programação possuem bibliotecas que fazem isso de maneira bem otimizada).

Em seguida, na linha 7 se itera para cada uma das coalizões S , somando o peso total dela na linha 8. Então, é verificado na linha 9 se o peso total é maior ou igual que a quota. Se não, não é necessário verificar se os jogadores são decisivos, pois a coalizão já é perdedora. Se sim, é verificado para todos os jogadores dessa coalizão (linha 10) se eles são decisivos, isto é, é subtraído o seu peso do total e comparamos com a quota, na linha 11. Caso o novo peso seja menor que a quota, significa que a coalizão se tornou

perdedora e esse jogador é um jogador decisivo, sendo somado uma unidade ao seu power index, na linha 12.

Note aqui a importância da ordenação dos pesos realizada anteriormente. Isso possibilita que ao verificar se um jogador é decisivo em uma coalizão e ele não o for, podemos abandonar essa verificação pois garantimos que nenhum outro jogador nesta coalizão será decisivo.

Ao final da iteração, para normalizar é somado o total de vezes em que os jogadores são decisivos, na linha 15, e na linha 17 é dividido o valor do power index de cada jogador por esse total. Assim, é encontrado o power index β_i para cada jogador i neste jogo.

4.1.2 Shapley-Shubik

Para o power index de Shapley-Shubik, o procedimento possui poucas diferenças em relação ao outro método de Banzhaf descrito previamente. Aqui, é calculado o valor de ϕ para cada jogador i conforme definido na Seção 2.2.2. Para poupar tempo computacional, é calculado o valor das fatoriais presentes na soma em ϕ antecipadamente. O Algoritmo 4 ilustra o cálculo da expressão

$$\frac{|S|!(|N| - |S| - 1)!}{|N|!}. \quad (4.1)$$

Algoritmo 4: Fats

Entrada: O número total de jogadores $n = |N|$

Saída: Cálculo da expressão (4.1) para todo $|S|$ de 0 a $|N|$.

```

1  $F \leftarrow \{f_1, f_2, \dots, f_n\}$ ;
2 para  $s \leftarrow 0$  até  $n - 1$  faça
3    $r \leftarrow 1$ ;
4    $np \leftarrow n$ ;
5   para  $d \leftarrow 1$  até  $s$  faça
6      $r \leftarrow r * np$ ;
7      $np \leftarrow np - 1$ ;
8      $r \leftarrow r/d$ ;
9    $r \leftarrow r * np$ ;
10   $f_{s+1} \leftarrow 1/r$ ;
11 devolva  $F$ ;
```

Como sabemos os valores que $|S|$ e $|N|$ podem assumir desde antes da execução do algoritmo para descobrir os power indices, se pode armazenar o valor a ser utilizado na soma de ϕ , quando é verificado se as coalizões que ganham e deixam de ganhar ao se retirar um jogador.

Veja que a expressão (4.1) (ou mais precisamente o inverso dela) é muito similar ao coeficiente binomial $\binom{n}{k}$ e, por isso, nosso método se baseia em um cálculo iterativo de um coeficiente binomial, onde utilizamos a recorrência $\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1}$ para encontrar o resultado.

Então, para cada tamanho s de coalizão é calculado a expressão (4.1) e o resultado salvo no respectivo índice na lista de resultados.

Assim, é possível calcular o power index de Shapley-Shubik. O Algoritmo 5 ilustra o cálculo desse power index.

Algoritmo 5: Shapley-Shubik power index

Entrada: O conjunto de pesos W e a quota q .

Saída: O conjunto de power index encontrado.

```

1  $\phi \leftarrow \{\phi_1 = 0, \phi_2 = 0, \dots, \phi_n = 0\}$ ;
2  $F \leftarrow Fats(W, q)$ ;
3  $min \leftarrow MinCoalizao(W, q)$ ;
4  $max \leftarrow MaxCoalizao(W, q)$ ;
5 ordena  $W$  em ordem decrecente;
6 para  $i \leftarrow min$  até  $max$  faça
7    $Sn \leftarrow Combinacoes(W, i)$ ;
8   para todo  $S \in Sn$  faça
9      $cw \leftarrow \sum_{w \in S} w$ ;
10    se  $cw \geq q$  então
11      para todo  $w \in S$  faça
12        se  $cw - w < q$  então
13           $\phi_i \leftarrow \phi_i + f_i$ ;
14        senão
15          encerra;
16 devolva  $\phi$ ;

```

Nesse algoritmo, a única diferença se dá no cálculo do próprio valor do power index. O primeiro passo é obter os valores da expressão 4.1 na linha 2, de 0 ao número total de jogadores do jogo. Depois, ao atribuir o valor dos power indices, diferentemente do Banzhaf, que soma um valor unitário ao power index, aqui é somado os valores calculados pela função mostrada acima.

Note também que depois de iteradas todas as coalizões, não é preciso alterar o power index, pois o valor ϕ obtido já está forma normalizada.

4.2 Problema Inverso

Realizamos duas abordagens distintas para resolver o IPIP: uma formulação em programação linear inteira, solucionada com o resolvidor CPLEX, e uma heurística evolutiva baseada no framework BRKGA. Destacamos que resolver este problema inverso é particularmente difícil, pois mesmo para avaliar a qualidade de uma solução, ou seja, calcular o power index de uma das distribuição de pesos, temos que resolver um problema NP-completo.

4.2.1 Formulação do problema inverso em PLI

Baseado no trabalho de Kurz [2012] desenvolvemos uma formulação modificada em PLI para resolver o IPIP utilizando jogos de votação ponderada, considerando ambos Banzhaf e Shapley-Shubik power indices.

Cabe ressaltar que utilizamos na PLI o Shapley-Shubik power index, devido à maior simplicidade comparado ao Banzhaf power index.

Seja n o número de jogadores e $D = \{d_1, \dots, d_n\}$ o conjunto de power indices desejado, com $d_i \geq 0$ e $\sum_{i=1}^n d_i = 1$. Definimos a coalizão $U \subseteq N$ com o subconjunto de jogadores que votam juntos, onde N é o conjunto com todos os jogadores. Utilizamos as variáveis binárias x_U para indicar se a coalizão U vence e $y_{i,U}$ para dizer se o jogador i é um jogador decisivo na coalizão U , isto é, se a coalizão passa a perder quando i muda o seu voto.

Os power indices dos jogadores são representados por p_i e usamos a variável $\delta_i = |d_i - p_i|$ para mensurar o erro de cada jogador. As variáveis w_i e q correspondem, respectivamente, ao peso dos jogadores e à quota. Para minimizar a soma dos erros, utilizamos a formulação 6. As inequações (4.3) e (4.4) são utilizadas para garantir que δ_i não é negativo. A

Algoritmo 6: Formulação em PLI para o Problema Inverso.

$$\min \sum_{i=1}^n \delta_i \quad (4.2)$$

$$\text{s.a } \delta_i \geq p_i - d_i \quad 1 \leq i \leq n, \quad (4.3)$$

$$\delta_i \geq d_i - p_i \quad 1 \leq i \leq n, \quad (4.4)$$

$$p_i = \frac{1}{n!} \cdot \sum_{U \subseteq N \setminus \{i\}} (|U|! \cdot (n - |U| - 1)!) \cdot y_{i,U} \quad 1 \leq i \leq n, \quad (4.5)$$

$$y_{i,U} = x_{U \cup \{i\}} - x_U \quad 1 \leq i \leq n, U \subseteq N \setminus \{i\}, \quad (4.6)$$

$$x_U \geq x_{U \setminus \{j\}} \quad \forall \emptyset \neq U \subseteq N, j \in U, \quad (4.7)$$

$$q - (1 - x_U) \cdot M - \sum_{i \in U} w_i \leq 0 \quad \forall U \subseteq N, \quad (4.8)$$

$$-x_U \cdot M + \sum_{i \in U} w_i \leq q - 1 \quad \forall U \subseteq N, \quad (4.9)$$

$$x_\emptyset = 0 \quad (4.10)$$

$$x_N = 1 \quad (4.11)$$

$$x_U \in \{0,1\} \quad \forall U \subseteq N, \quad (4.12)$$

$$y_{i,U} \in \{0,1\} \quad 1 \leq i \leq n, U \subseteq N \setminus \{i\}, \quad (4.13)$$

$$w_i \geq 0 \quad 1 \leq i \leq n, \quad (4.14)$$

$$q > 0. \quad (4.15)$$

equação (4.5) corresponde ao Shapley-Shubik power index para o jogador i e usamos (4.6) para indicar que i é um votante decisivo na coalizão U . Nos restringimos a pesos inteiros para decidir se $\sum_{i \in U} w_i \geq q$ ou $\sum_{i \in U} w_i < q$ na formulação em PLI. Para conectar x_U

com w_i e q empregamos (4.8) e (4.9), onde M é uma constante suficientemente grande maior que $\sum_{i=1}^n w_i$. Além disso, usamos a inequação (4.7) para garantir que o jogo é monótono e as equações (4.10) e (4.11) para impor que uma coalizão vazia perca e uma grande coalizão vença.

4.2.2 Heurística evolutiva com BRKGA

Propomos uma heurística evolutiva baseada no framework BRKGA [Toso e Resende, 2015], que encontra soluções para instâncias do IPIP de tamanho razoável. Desenvolvemos esse algoritmo como alternativa ao implementado anteriormente, devido aos resultados e também à aplicação ao nosso estudo de caso, conforme será exposto no Capítulo 5.

A ideia principal de um algoritmo evolutivo pode ser enunciado como: dada uma população de vários indivíduos, são selecionados dentre eles os que são mais aptos, que geram descendentes e integram uma nova população. Para isso, através de uma função para ser maximizada, pode-se mensurar a aptidão, também chamada de *fitness*, de cada indivíduo.

Nossa heurística é baseada no framework BRKGA, implementado por Toso e Resende [2015]. O BRKGA (biased random-key genetic algorithm) é uma metaheurística de busca para encontrar soluções ótimas ou quase ótimas para difíceis problemas de otimização combinatória.

Na heurística evolutiva desenvolvida, parte fundamental é a geração aleatória de cromossomos, cujos alelos tem valor no intervalo $[0,1)$. Precisamos decodificar os cromossomos afim de resolver nosso problema de otimização: o Problema Inverso. Isso é necessário pois utilizamos os cromossomos para construir a solução do nosso problema de otimização, calculando a função objetivo a partir deles.

Para nosso cenário, os alelos de um cromossomo são equivalentes ao peso w de cada jogador i . A quota q pode ser representada de duas maneiras: por um cromossomo além desses que representam o peso ou como uma função dos pesos. No primeiro caso, temos uma quota desconhecida, enquanto no segundo já sabemos qual valor a quota deve possuir, no caso de uma votação com maioria simples por exemplo, onde a quota será

$$\frac{\sum_{1 \leq i \leq n} w_i}{2}. \quad (4.16)$$

Um cromossomo decodificado para nosso problema de otimização precisa ter seu *fitness* calculado para se tornar o que chamamos de indivíduo. O *fitness* de um indivíduo é calculado através da nossa função objetivo, dado por:

$$\sum_{i=1}^n |d_i - p_i|. \quad (4.17)$$

Onde d_i é o power index desejado para o jogador i enquanto p_i é o power index encontrado para o jogador i , que é descoberto calculando os power indices para o jogo representado por cada cromossomo, onde cada alelo possui um valor que representa o peso w_i .

O cálculo do power index pode ser realizado independentemente da nossa decodificação

dos cromossomos para um conjunto de pesos. Por isso, podemos utilizar qualquer power index desejado.

Além da decodificação dos cromossomos, ilustramos na Tabela 4.1 os parâmetros necessários para especificar e entender o BRKGA. Dos parâmetros, o número de alelos por cromossomo é definido justamente pelo número de jogadores em um jogo. Os outros parâmetros, p , p_e , p_m e p_o são definidos de acordo com outras condições, como por exemplo a instância do problema, o tempo de execução desejado e o número de gerações desejadas.

Parâmetro	Descrição
n	Número de alelos por cromossomo
p	Número de cromossomos em uma população
p_e	Tamanho do conjunto elite
p_m	Número de mutantes a serem introduzidos na população em cada geração
p_o	Probabilidade de um alelo ser herdado do ascendente da elite

Tabela 4.1: Parâmetros necessários para o BRKGA.

O algoritmo funciona da seguinte forma: inicia com uma população de p cromossomos, cada um com n alelos, que evolui até atingir um critério de parada. Para cada geração k , são executados os seguintes passos:

1. Se calcula o *fitness* de cada indivíduo através da decodificação descrita anteriormente.
2. A população é dividida em dois grupos: uma elite contendo p_e indivíduos com os melhores valores de aptidão e uma não-elite com os indivíduos restantes.
3. A geração $k + 1$ será composta de:
 - (a) p_e indivíduos da elite copiados da geração k .
 - (b) p_m cromossomos gerados aleatoriamente (mutantes).
 - (c) $p_o = p - p_m - p_e$ descendentes produzidos através do cruzamento entre dois pais da geração k .

A Figura 4.1 mostra um exemplo de evolução da geração k para a geração $k + 1$, onde temos $p = 10$, $p_e = 0.2$, $p_m = 0.2$ e $p_o = 0.6$.

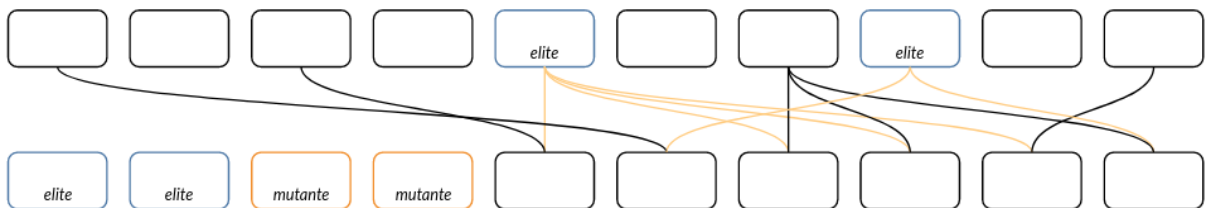


Figura 4.1: Exemplo da evolução da população k para $k + 1$.

O cruzamento realizado para gerar um novo cromossomo se dá selecionando aleatoriamente dois pais da geração atual, um da elite e um da não-elite. Para definir o i -ésimo

alelo de um descendente, é gerado um número uniformemente aleatório $r \in [0,1)$. Se $r \leq \rho_e$ com $\rho_e > 0.5$, o descendente herda o alelo i do pai elite; caso contrário, herda o alelo i do pai não-elite.

A Figura 4.2 ilustra um exemplo de cruzamento entre dois cromossomos com $\rho_e = 0.6$, um elite e um não elite, na produção de um novo cromossomo.

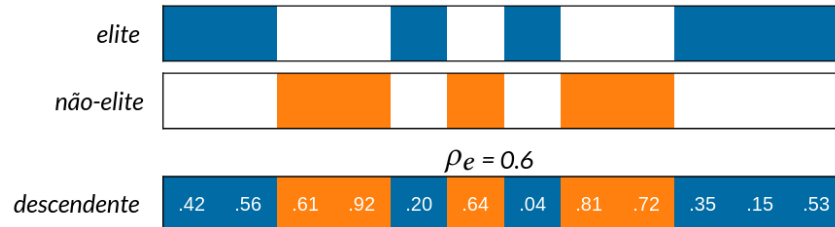


Figura 4.2: Exemplo de cruzamento para produção de um novo cromossomo.

A execução do algoritmo finaliza quando se atinge o critério de parada definido. Utilizamos dois critérios de parada no desenvolvimento do trabalho: atingir o número máximo de gerações definido ou atingir um erro menor que 0.00001. Ao final da execução, se tem um indivíduo de melhor *fitness* que contém o conjunto de pesos w .

O BRKGA ainda dispõe do uso de populações independentes, que permite que várias populações evoluam independentemente, substituindo os indivíduos de pior aptidão com os indivíduos da elite de outra população, com a ação sendo realizada em um intervalo de gerações definidos.

Além disso, o BRKGA permite a redefinição de uma população com novos cromossomos gerados aleatoriamente, o que é útil para evitar mínimos locais. Embora seja uma estratégia válida, nos nossos experimentos não a implementamos pelo número de populações reduzido.

Capítulo 5

Resultados Experimentais

Neste capítulo discutimos os resultados obtidos na implementação dos algoritmos descritos na Seção 4.2 e sua execução para a solução do problema inverso. Vale lembrar que nesse problema de otimização, estamos interessados em minimizar o erro $\sum_{i=1}^n |d_i - p_i|$ nas duas soluções apresentadas, além de mensurar o tempo de execução de cada um deles.

Nas próximas seções apresentemos primeiramente algumas instâncias que utilizamos, seguido dos resultados para o algoritmo em PLI e heurística evolutiva. Para o último, aprofundamos os resultados no nosso estudo na Câmara dos Deputados e, por fim, comparamos nossos resultados.

5.1 Instâncias

Para resolver o problema inverso precisamos de dados de entrada, que para esse problema são conjuntos de power indices. Poderíamos utilizar qualquer conjunto de números reais que respeitasse as propriedades de um power index, mas preferimos optar por utilizar instâncias mais próximas de cenários reais.

É importante apontar que a definição das instâncias é um problema por si só, isto é, determinar um conjunto de power indices que seja desejado é um estudo além de apenas resolver o problema inverso dos power indices. Por isso, nos baseamos em alguns conceitos disponíveis na literatura usando a *penrose square root law* [Penrose, 1946] e a *linear Shapley rule* [Kurz et al., 2017] para a definição das instâncias. Essas regras serão exibidas posteriormente.

Um ponto relevante é que a performance dos algoritmos varia muito conforme a instância do problema, o que pode favorecer algumas abordagens em detrimento de outras, como mostraremos nas próximas seções.

5.1.1 Conselho da União Europeia

Como dito anteriormente, um dos cenários mais famosos e mais estudados onde há jogos de votação ponderada é a União Europeia. Por isso, consideramos uma excelente escolha de instância para solucionar o problema inverso. Utilizaremos as instâncias apresentadas no trabalho de Kurz [2012] e de Zyczkowski e Slomczynski [2004]. Importante observar que as instâncias mudam tanto de tamanho (existem mais membros em 2012 do que em

2004) e de valores (as populações mudam com o tempo). Słomczyński e Życzkowski [2016] ainda expõe uma distribuição para o conselho com os mesmos 27 membros de Kurz [2012], porém com a população de 2007, mas preferimos usar o cenário em 2004 para variarmos a instância.

Em nosso primeiro cenário para a União Europeia, mostramos as instâncias definidas por Kurz [2012]. A Tabela 5.1 apresenta a população dos 27 países da UE em 2012, em ordem decrescente, conforme exposta no trabalho.

Membro	População	Membro	População	Membro	População
Alemanha	82500	Chéquia	10500	Finlândia	5200
Reino Unido	60600	Bélgica	10400	Irlanda	4100
França	60000	Hungria	10200	Letônia	3400
Itália	58500	Portugal	10100	Lituânia	2300
Espanha	43000	Suécia	9000	Eslovênia	2000
Polônia	38200	Áustria	8200	Estônia	1300
Romênia	21700	Bulgaria	7800	Chipre	700
Países Baixos	16300	Dinamarca	5400	Luxemburgo	500
Grécia	11100	Eslováquia	5400	Malta	400

Tabela 5.1: População em milhares para os 27 estados membros da UE em 2017.

Então, construímos instâncias a partir de composições destes países, do mais populoso para o menos populoso. Nosso objetivo é construir instâncias para cenários figurados onde existe um número de países de 2 a 27.

Para cada uma dessas composições, aplicamos a *penrose square root law* para definirmos o power index desejado, gerando as instâncias pretendidas. Nessa regra, calculamos o power index desejado d_i da seguinte forma, onde pop_i é a população do jogador i :

$$d_i = \frac{\sqrt{pop_i}}{\sum_{j=1}^n \sqrt{pop_j}}. \quad (5.1)$$

A Tabela 5.2 ilustra algumas das instâncias, com os power indices aproximados para a quarta casa decimal.

Instância	Membros	Power indices
UE ₂	Alemanha, Reino Unido	[0.5385, 0.4615]
UE ₃	Alemanha, Reino Unido, França	[0.3690, 0.3163, 0.3147]
UE ₄	Alemanha, Reino Unido, França, Itália	[0.2815, 0.2413, 0.2401, 0.2371]
UE ₅	Alemanha, Reino Unido, França, Itália, Espanha	[0.2340, 0.2005, 0.1995, 0.1970, 0.1689]
⋮	⋮	⋮

Tabela 5.2: Instâncias UE₂ a UE₅ para a UE em 2012.

No outro cenário, utilizamos o trabalho de Życzkowski e Słomczyński [2004], em que os autores propõem uma distribuição de pesos para a União Europeia baseada na *penrose square root rule*. Essa proposta, apelidada de *Jagiellonian Compromise*, ganhou relevância na mídia e chegou a ser brevemente discutida na reforma do Conselho da União Europeia em 2007.

Diferentemente do nosso trabalho, onde solucionamos o problema inverso dos power indices, os autores chegam a uma distribuição de pesos de outra forma. Por isso, é muito significativo que possamos resolver o mesmo problema solucionando o IPIP. A Tabela 5.3 apresenta a população dos 25 países da UE em 2004, em ordem decrescente, conforme expressa no trabalho e utilizada na *Jagiellonian Compromise*.

Membro	População	Membro	População	Membro	População
Alemanha	82540	Bélgica	10360	Lituânia	3460
França	59630	Chéquia	10200	Letônia	2330
Reino Unido	59330	Hungria	10140	Eslovênia	2000
Itália	57320	Suécia	8940	Estônia	1360
Espanha	40680	Áustria	8070	Chipre	720
Polônia	38220	Dinamarca	5380	Luxemburgo	450
Países Baixos	16190	Eslováquia	5380	Malta	400
Grécia	11020	Finlândia	5210		
Portugal	10410	Irlanda	3960		

Tabela 5.3: População em milhares para os 25 estados membros da UE em 2004.

A partir dessa população com os 25 membros, construímos a instância. Aqui também aplicamos a *penrose square root law* para definirmos o power index desejado, gerando as instâncias pretendidas. O cálculo é feito conforme a equação 5.1.

A Tabela 5.4 ilustra a instância UE_{jc} construída, com os power indices aproximados para a quarta casa decimal.

Membro	Power index	Membro	Power index	Membro	Power index
Alemanha	0.1036	Bélgica	0.0367	Lituânia	0.0212
França	0.0881	Chéquia	0.0364	Letônia	0.0174
Reino Unido	0.0879	Hungria	0.0363	Eslovênia	0.0161
Itália	0.0864	Suécia	0.0341	Estônia	0.0133
Espanha	0.0728	Áustria	0.0324	Chipre	0.0097
Polônia	0.0705	Dinamarca	0.0265	Luxemburgo	0.0077
Países Baixos	0.0459	Eslováquia	0.0265	Malta	0.0072
Grécia	0.0379	Finlândia	0.0260		
Portugal	0.0368	Irlanda	0.0227		

Tabela 5.4: Instância UE_{jc} para a UE em 2004.

5.1.2 Câmara dos Deputados do Brasil

Como descrito no Capítulo 3 parte importante deste trabalho é o estudo da desproporcionalidade na Câmara dos Deputados. Então, de maneira similar à União Europeia, geramos nossa instância a partir da população das Unidades Federativas no Brasil, conforme a população em 2019 [IBGE, 2019], que está ilustrada na Tabela 5.5.

Aqui, diferentemente do descrito na seção anterior, construímos as instâncias apenas para as 27 unidades da federação. Embora também seja possível resolver o problema

UF	População	UF	População	UF	População
São Paulo	45.919.049	Santa Catarina	7.164.788	Piauí	3.273.227
Minas Gerais	21.168.791	Maranhão	7.075.181	Distrito Federal	3.015.268
Rio de Janeiro	17.264.943	Goiás	7.018.354	Mato Grosso do Sul	2.778.986
Bahia	14.873.064	Amazonas	4.144.597	Sergipe	2.298.696
Paraná	11.433.957	Espírito Santo	4.018.650	Rondônia	1.777.225
Rio Grande do Sul	11.377.239	Paraíba	4.018.127	Tocantins	1.572.866
Pernambuco	9.557.071	Rio Grande do Norte	3.506.853	Acre	881.935
Ceará	9.132.078	Mato Grosso	3.484.466	Amapá	845.731
Pará	8.602.865	Alagoas	3.337.357	Roraima	605.761

Tabela 5.5: População para as 27 UFs do Brasil em 2019.

inverso com as instâncias de variados tamanhos, nesse caso estamos interessados em contribuir para o problema descrito na Seção 3.1 e portanto, usamos somente a instância com 27 jogadores.

Além disso, neste caso estendemos a discussão para o que caracterizamos como power index justo. Intuitivamente, pode-se argumentar que uma distribuição de poder mais justa se dá quando os power indices, e não os pesos, são proporcionais à população.

Na literatura, a *penrose square root rule* [Penrose, 1946] é objeto de estudo de muitos autores em diferentes cenários [Felsenthal e Machover, 2004, Zyczkowski e Slomczynski, 2004, Lima e Ramos, 2010, Miller, 2013]. Penrose [1946] argumenta que em um sistema binário de decisão, isto é, os votantes apenas votam *sim* ou *não*, e onde todos os indivíduos possuem decisões estatisticamente independentes e igualmente prováveis, um agente deve ter uma probabilidade de ser pivô proporcional à raiz quadrada de sua população.

Kurz et al. [2017] estendem a análise binária para um cenário de decisões onde os agentes têm preferências em um intervalo e possuem a chamada preferência de pico único. Isso quer dizer que em vez de terem preferências que envolvem somente *sim* ou *não*, os agentes possuem preferências distribuídas em um intervalo, com um ponto onde há o melhor resultado desejado e o resultado piora conforme se distancia desse ponto.

Essa análise expandida é importante pois as decisões binárias não englobam conjunturas onde possa haver negociação e barganha. No contexto da Câmara dos Deputados isto é de fundamental importância por exemplo, em situações onde é discutido a distribuição de recursos para as unidades federativas.

Os autores chegam à conclusão que, em cenários onde há um alto nível de heterogeneidade nos interesses dos agentes, uma distribuição linear é mais adequada, a qual alcunham de *linear Shapley rule*. Nesse trabalho, estamos particularmente interessados nas votações em que os deputados decidem de forma homogênea dentro de suas UFs, e de forma heterogênea entre UFs. Em virtude da homogeneidade interna, podemos aproximar o número de deputados por um peso. Além disso, a heterogeneidade entre unidades federativas torna adequada a escolha da *linear Shapley rule*.

Nessa regra, calculamos o power index desejado d_i da seguinte forma, onde pop_i é a população do jogador i :

$$d_i = \frac{pop_i}{\sum_{j=1}^n pop_j}. \quad (5.2)$$

Então, buscamos encontrar uma distribuição (resolver o problema inverso) para a Câmara dos Deputados na qual cada estado possui poder efetivo proporcional a sua população. Para isso, utilizamos a *linear Shapley rule* para construir os power indices, representados na Tabela 5.6, aproximados para a quarta casa decimal.

UF	Power index	UF	Power index	UF	Power index
São Paulo	0.1015	Santa Catarina	0.0401	Piauí	0.0271
Minas Gerais	0.0689	Maranhão	0.0399	Distrito Federal	0.0260
Rio de Janeiro	0.0623	Goiás	0.0397	Mato Grosso do Sul	0.0250
Bahia	0.0578	Amazonas	0.0305	Sergipe	0.0227
Paraná	0.0507	Espírito Santo	0.0300	Rondônia	0.0200
Rio Grande do Sul	0.0505	Paraíba	0.0300	Tocantins	0.0188
Pernambuco	0.0463	Rio Grande do Norte	0.0281	Acre	0.0141
Ceará	0.0453	Mato Grosso	0.0280	Amapá	0.0138
Pará	0.0439	Alagoas	0.0274	Roraima	0.0117

Tabela 5.6: Instância para a Câmara dos Deputados do Brasil.

5.2 Programação Linear Inteira

Implementamos a formulação em Programação Linear Inteira descrita no Algoritmo 6 para a solução do problema inverso utilizando o ILOG CPLEX 12.9 como resolvidor de PLI. Vale reforçar que embora a formulação desenvolvida esteja utilizando o Shapley-Shubik power index, também conseguimos utilizar o Banzhaf power index, como descrito em Kurz [2012].

Utilizamos as instâncias do Conselho da União Europeia descritas na Seção 5.1.1 e ilustradas na Tabela 5.2. Assim, resolvemos o problema inverso em jogos a partir da instância UE_3 (com 3 jogadores).

Na Tabela 5.7 mostramos os resultados com o erro encontrado, o tempo de execução, e um jogo encontrado para as instâncias resolvidas. Note que a partir de 8 jogadores o tempo cresce consideravelmente, e com 9 jogadores o tempo passa ser inviável.

Instância	Erro	Tempo	Quota; Pesos
UE_3	0.0713791	0.02s	[1; 1, 1, 1]
UE_4	0.0630734	0.03s	[1; 1, 1, 1, 1]
UE_5	0.0690250	0.05s	[1; 1, 1, 1, 1, 1]
UE_6	0.0540185	0.51s	[115; 57, 57, 56, 56, 3, 55]
UE_7	0.0375077	32.76s	[556; 218, 115, 113, 114, 111, 110, 108]
UE_8	0.0178177	52.24m	[1434; 411, 409, 409, 408, 407, 208, 206, 205]

Tabela 5.7: Resultados para as instâncias UE_3 a UE_8 utilizando PLI.

Não é difícil perceber que para instâncias maiores e principalmente para uma instância de tamanho 27, como é o caso da Câmara dos Deputados (e também da UE) essa abordagem se torna inviável. Embora nossa formulação não possua muitas otimizações, a explosão do tempo de execução com jogos entre 8 e 9 jogadores torna difícil resolver problemas reais onde existam muitos deles.

Por isso, optamos por desenvolver uma estratégia original e que não é um algoritmo exato para encontrar jogos de votação ponderada resolvendo o problema inverso dos power indices.

5.3 Heurística Evolutiva

Implementamos a heurística evolutiva descrita na Seção 4.2.2 utilizando o framework BRKGA. Aqui, o principal objetivo é solucionar o problema inverso considerando o tema descrito na Seção 3.1 e que leva em conta os problemas de poder efetivo apresentados nela, além de sanar os problemas enfrentados pelo algoritmo em PLI apresentado anteriormente, e também solucionar as instâncias do Conselho da União Europeia.

5.3.1 Câmara dos Deputados

Utilizamos a instância da Câmara dos Deputados descritas na Seção 5.1.2 e ilustradas na Tabela 5.6, onde nos baseamos na *linear Shapley rule*, em que uma distribuição justa se dá quando os power indices são proporcionais a população. Ou seja, queremos que o power index de cada estado seja proporcional a sua população.

Apresentamos na Tabela 5.8 os resultados encontrados solucionando o problema inverso para a instância descrita. Exibimos os resultados com os pesos somando o valor de 513 e uma quota de 257, porém nosso algoritmo produz uma saída escalar à esses valores.

UF	Peso	UF	Peso	UF	Peso
São Paulo	96.9087	Santa Catarina	18.5239	Piauí	8.4007
Minas Gerais	51.5560	Maranhão	18.2664	Distrito Federal	7.7960
Rio de Janeiro	42.7700	Goiás	16.8354	Mato Grosso do Sul	7.3369
Bahia	36.9520	Amazonas	10.9714	Sergipe	6.0386
Paraná	28.8636	Espírito Santo	10.7269	Rondônia	4.8979
Rio Grande do Sul	28.6501	Paraíba	10.4984	Tocantins	4.0876
Pernambuco	24.0575	Rio Grande do Norte	9.9475	Acre	2.3562
Ceará	23.3330	Mato Grosso	9.1272	Amapá	2.1576
Pará	22.0626	Alagoas	8.4895	Roraima	1.3881
Quota = 257					

Tabela 5.8: Resultado para a instância da Câmara dos Deputados utilizando a heurística evolutiva.

Assim, transformarmos os pesos w_i de cada jogador i , que estão no intervalo $[0,1)$, nos pesos w'_i , que estão ilustrados na Tabela 5.8 com os valores dos pesos aproximados para a quarta casa decimal, da seguinte forma:

$$w'_i = \frac{513 * w_i}{\sum_{j=1}^n w_j}. \quad (5.3)$$

Executamos nosso algoritmo com uma população $p = 10$, uma fração $p_e = 0.1$ da população que representa o conjunto elite, uma fração $p_m = 0.2$ de mutantes, uma fração $p_o = 0.7$ de cromossomos gerados a partir de cruzamento e uma probabilidade $\rho_e = 0.7$ de um descendente herdar da elite.

Além disso, executamos com um total de 3 populações independentes, em um máximo de 3000 gerações. Também trocamos a cada 25 gerações os 2 melhores indivíduos de cada população. Outro ponto a ser ressaltado é a possibilidade de execução em paralelo proporcionado pelo BRKGA, que nos possibilitou executar os algoritmos utilizando 10 *threads*, melhorando a performance.

Essa solução encontrada possui erro $\sum_{i=1}^n |d_i - p_i| = 0.0095140$, com tempo de execução de aproximadamente 526 minutos. É importante observar que para obter um resultado que se aplique na Câmara dos Deputados, é preciso arredondar o resultado, assim como fizemos em 3.1.2. Os números de deputados para as unidades federativas estão ilustrados na Tabela 5.9.

Estado	Diferença p/			Estado	Diferença p/		
	#Dep.	Atual	Prop.		#Dep.	Atual	Prop.
Acre	2	-6		Paraíba	11	-1	+1
Alagoas	8	-1		Paraná	29	-1	+1
Amapá	2	-6		Pernambuco	24	-1	+1
Amazonas	11	+3	+1	Piauí	8	-2	
Bahia	37	-2	+1	Rio de Janeiro	43	-3	+1
Ceará	23	+1	+1	Rio Grande do Norte	10	+2	+1
Distrito Federal	8		+1	Rio Grande do Sul	29	-2	+1
Espírito Santo	11	+1	+1	Rondônia	5	-3	+1
Goiás	17			Roraima	1	-7	
Maranhão	18		+1	Santa Catarina	19	+3	+1
Mato Grosso	9	+1		São Paulo	97	+27	-15
Mato Grosso do Sul	7	-1		Sergipe	6	-2	
Minas Gerais	52	-1		Tocantins	4	-4	
Pará	22	+5	+1				

Tabela 5.9: Distribuição de deputados obtidos utilizando a heurística evolutiva baseado na *linear Shapley rule*.

Analisando o resultado encontrado com as análises realizadas nas Seções 3.1.1 e 3.1.2, observamos que a proporção de cadeiras se mantém parecida com o caso estritamente proporcional. Apenas São Paulo possui uma diferença expressiva, com os outros estados tendo uma mudança de no máximo uma cadeira.

Agora, comparando o poder efetivo com a população em cada UF, obtemos uma diferença muito pequena, causada pelo erro encontrado na solução e, principalmente, pelo arredondamento. A Figura 5.1 permite observar essa comparação, bem como notar

que a distribuição proposta corrige tanto as distorções da distribuição atual quanto da estritamente proporcional.

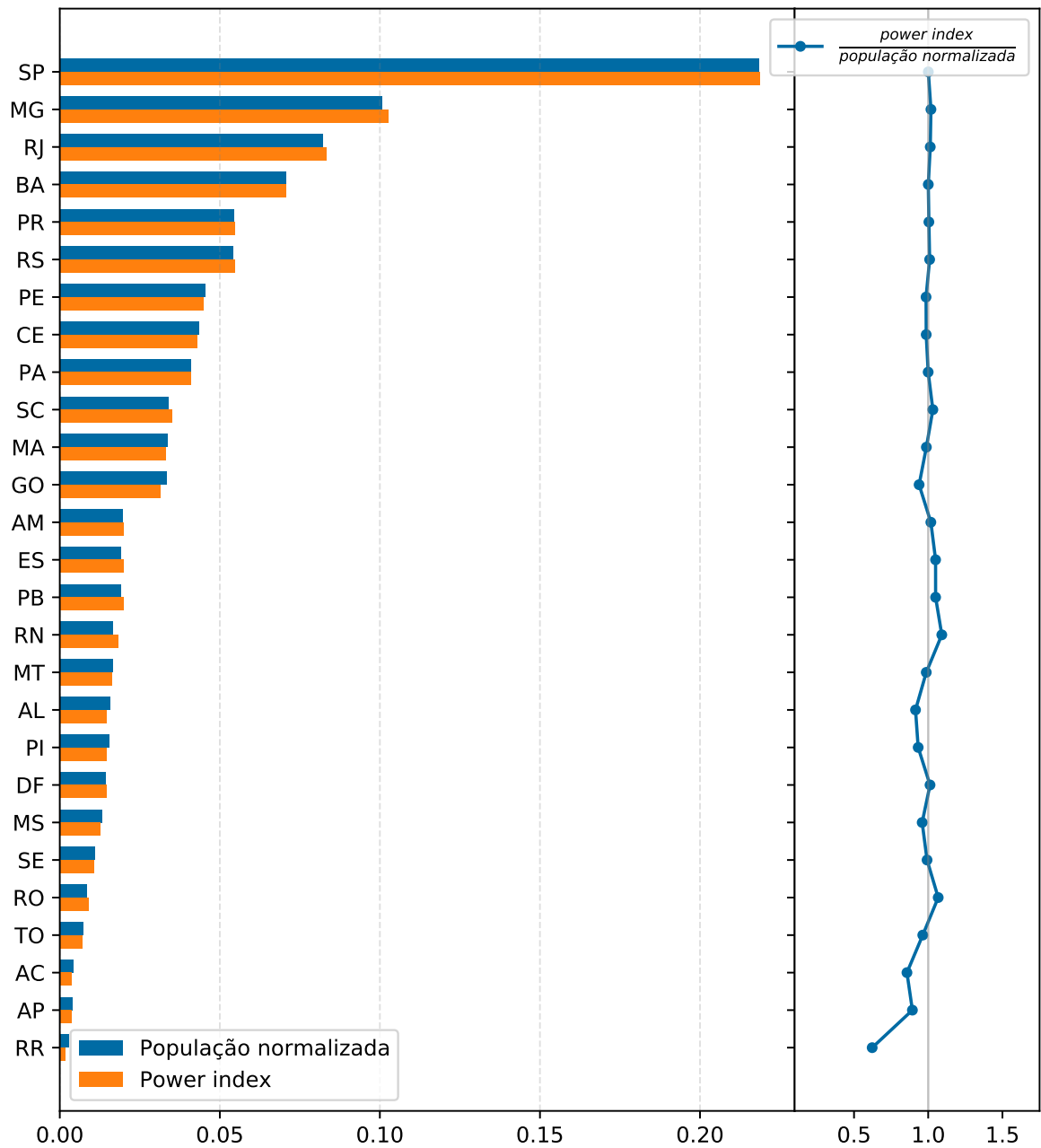


Figura 5.1: Comparação entre população normalizada e Shapley-Shubik power index na distribuição proposta.

Podemos comparar ainda a razão entre a população normalizada de cada estado com seu respectivo power index, para cada cenário analisado, como mostra a Figura 5.2. Note que a solução encontrada possui razão mais próxima a 1 em relação aos outros cenários, principalmente se confrontado com a distribuição atual.

É importante observar que, conforme a população decresce, há maior dificuldade em a igualar com o power index. Isso ocorre por conta do arredondamento, que afeta o número de deputados e, conseqüentemente, o power index.

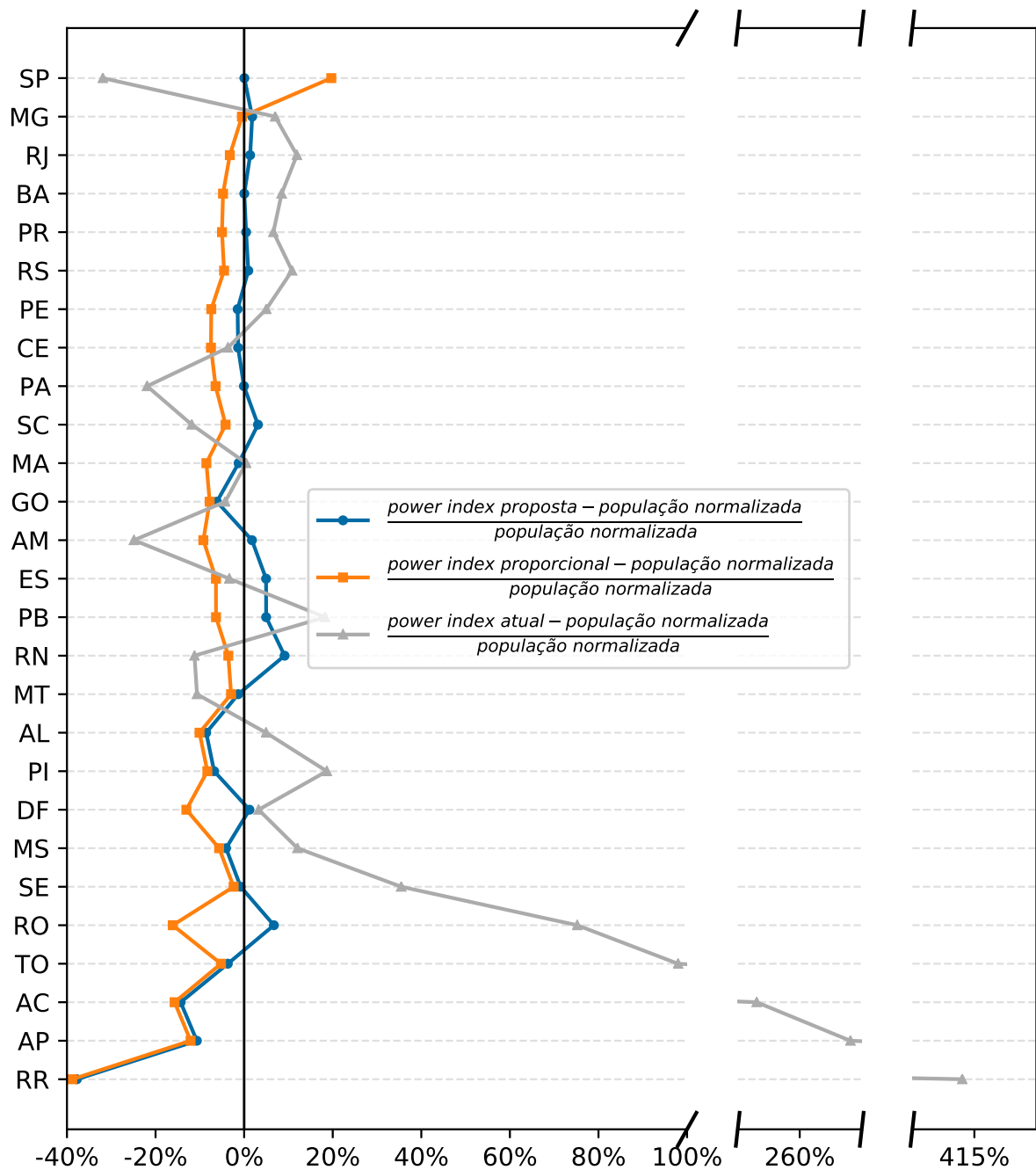


Figura 5.2: Comparação da razão entre o Shapley-Shubik power index e a população normalizada nos três cenários descritos anteriormente.

5.3.2 União Europeia

Assim como na seção que apresenta os resultados da PLI, utilizamos as instâncias da União Europeia vistas na Seção 5.1.1 e ilustradas na Tabela 5.2. Executamos nosso algoritmo para as instâncias UE_3 a UE_{11} , com os mesmos parâmetros que usamos nos experimentos com a instância da Câmara dos Deputados, com exceção do tamanho da população e número de populações independentes, que aumentamos de 10 para 50 e 3 para 20, respectivamente.

Além disso, também solucionamos o problema inverso para instância UE_{jc} afim de

encontrar um conjunto de pesos que se aproxime daqueles retratados na *Jagiellonian Compromise*.

Aqui, diferentemente do caso da Câmara dos Deputados, não temos uma quota pré-definida e por isso, precisamos descobri-la na solução do problema inverso. Para isso, calculamos a quota como a soma do valor dos alelos dos pesos multiplicado pelo alelo que representa a quota.

Na Tabela 5.10 mostramos os resultados com o erro encontrado para as instâncias UE₃ a UE₁₁, o tempo de execução até o ótimo, e o número de gerações passadas para atingir a solução. Não são exibidos os tempos para as instâncias de 3 a 6 jogadores pois a solução foi encontrada já na geração de criação dos cromossomos e os tempos são imprecisos.

É importante dizer que resolvemos essa instância a fim de comparação. O tempo de execução é até a solução apresentada encontrada, a qual já sabemos ser a ótima (conforme o outro algoritmo executado), e não o tempo de execução total do algoritmo, o qual depende da quantidade de gerações definidas ou do erro definido (0.00001).

Instância	Erro	Tempo até o ótimo	Gerações
UE ₃	0.0713791	-	1
UE ₄	0.0630734	-	1
UE ₅	0.0690250	-	1
UE ₆	0.0540185	-	1
UE ₇	0.0375077	0.018s	2
UE ₈	0.0178177	0.022s	6
UE ₉	0.0068989	0.616s	98
UE ₁₀	0.0053565	8.903s	1175
UE ₁₁	0.0036279	4.45m	16767

Tabela 5.10: Resultados para as instâncias UE₃ a UE₁₁ utilizando a heurística evolutiva.

Um ponto importante é que nossa heurística encontra jogos com pesos reais, e não inteiros, como a PLI. Entretanto, podemos transformar esses valores reais em inteiros se assim desejarmos.

Para a instância UE_{jc} utilizamos os mesmos parâmetros que usamos com a instância da Câmara dos Deputados. A única diferença é o número de gerações: enquanto usamos 3000 no caso do Brasil, aqui utilizamos 10000 gerações, dado o número de jogadores ligeiramente menor nessa instância (mais a quota, que também faz parte dos cromossomos nesse caso). A Tabela 5.11 mostra os valores dos pesos e quota encontrados, aproximados para a quarta casa decimal e transformados da seguinte forma:

$$w'_i = \frac{w_i}{\sum_{j=1}^n w_j}, q' = \frac{q}{\sum_{j=1}^n w_j}. \quad (5.4)$$

Além dos pesos, também encontramos a quota $q' = 0.6373$. A solução encontrada possui erro $\sum_{i=1}^n |d_i - p_i| = 0.0016453$, com tempo de execução de aproximadamente 387 minutos.

Membro	Peso	Membro	Peso	Membro	Peso
Alemanha	0.1053	Bélgica	0.0365	Lituânia	0.0212
França	0.0887	Chéquia	0.0362	Letônia	0.0175
Reino Unido	0.0882	Hungria	0.0362	Eslovênia	0.0161
Itália	0.0867	Suécia	0.0339	Estônia	0.0130
Espanha	0.0727	Áustria	0.0322	Chipre	0.0095
Polônia	0.0701	Dinamarca	0.0263	Luxemburgo	0.0077
Países Baixos	0.0458	Eslováquia	0.0263	Malta	0.0072
Grécia	0.0377	Finlândia	0.0259		
Portugal	0.0366	Irlanda	0.0226		
Quota = 0.6373					

Tabela 5.11: Resultado para a instância UE_{jc} utilizando a heurística evolutiva.

5.4 Comparação dos resultados

Para avaliar nossos resultados, comparamos os resultados apresentados por Kurz [2012] nas instâncias UE_3 a UE_{11} com o que foi apresentado nas seções anteriores. A Tabela 5.12 apresenta a comparação entre as três soluções: A PLI de Kurz [2012], a PLI que implementamos e a heurística evolutiva que desenvolvemos.

Instância	Erro ótimo $\sum_{i=1}^n d_i - p_i $			Tempo até o ótimo		
	Kurz	PLI	Heurística	Kurz	PLI	Heurística
UE_3	0.0713793	0.0713791	0.0713791	0.01s	0.02s	-
UE_4	0.0630740	0.0630734	0.0630734	0.02s	0.03s	-
UE_5	0.0690250	0.0690250	0.0690250	0.04s	0.05s	-
UE_6	0.0540190	0.0540185	0.0540185	0.17s	0.51s	-
UE_7	0.0375078	0.0375077	0.0375077	0.40s	32.76s	0.02s
UE_8	0.0178178	0.0178177	0.0178177	2.86s	52.24m	0.02s
UE_9	0.0068992	N/A	0.0068989	49.65s	N/A	0.62s
UE_{10}	0.0053562	N/A	0.0053565	28.68m	N/A	8.90s
UE_{11}	0.0036285	N/A	0.0036279	69.16h	N/A	4.45m

Tabela 5.12: Comparação dos resultados para as instâncias UE_3 a UE_{11} .

Comparando o erro encontrado, notamos que se aproximam muito. Entre os algoritmos desenvolvidos por nós, os erros são precisamente os mesmos. A diferença em relação ao trabalho de Kurz parece ser mais um problema de arredondamento do que uma solução diferente, dado que nossa PLI é baseado no próprio trabalho, e a solução encontrada pela heurística também coincide.

Encontrar soluções próximas ao valor ótimo em nossa heurística parece ser viável, mas possui uma forte dependência dos parâmetros que usamos na execução do algoritmo. Se explorarmos mais as otimizações possíveis na heurística, podemos reduzir ainda mais o tempo de execução dessas instâncias.

Também comparamos o resultado encontrado por Zyczkowski e Slomczynski [2004], o qual forma a *Jagiellonian Compromise*, com a solução encontrada por nosso heurística

evolutiva. Aqui, o importante é comparar o power index de cada membro, que buscamos aproximar do desejado pela *penrose square root rule*. A Tabela 5.13 apresenta a comparação entre power indices e pesos para a *Jagiellonian Compromise* (JC) e a heurística evolutiva (HE).

Membro	Peso			Power index		
	JC	HE	Diferença	JC	HE	Diferença
Alemanha	0.1037	0.1053	$-16 \cdot 10^{-4}$	0.1036	0.1036	$\approx 0 \cdot 10^{-4}$
França	0.0881	0.0887	$-6 \cdot 10^{-4}$	0.0882	0.0883	$-1 \cdot 10^{-4}$
Reino Unido	0.0878	0.0882	$-4 \cdot 10^{-4}$	0.0879	0.0878	$1 \cdot 10^{-4}$
Itália	0.0864	0.0867	$-3 \cdot 10^{-4}$	0.0865	0.0864	$1 \cdot 10^{-4}$
Espanha	0.0728	0.0727	$1 \cdot 10^{-4}$	0.0729	0.0727	$2 \cdot 10^{-4}$
Polônia	0.0705	0.0701	$4 \cdot 10^{-4}$	0.0706	0.0702	$4 \cdot 10^{-4}$
Países Baixos	0.0459	0.0458	$1 \cdot 10^{-4}$	0.0459	0.0460	$-1 \cdot 10^{-4}$
Grécia	0.0379	0.0377	$2 \cdot 10^{-4}$	0.0379	0.0379	$\approx 0 \cdot 10^{-4}$
Portugal	0.0368	0.0366	$2 \cdot 10^{-4}$	0.0368	0.0368	$\approx 0 \cdot 10^{-4}$
Bélgica	0.0367	0.0365	$2 \cdot 10^{-4}$	0.0367	0.0367	$\approx 0 \cdot 10^{-4}$
Chéquia	0.0364	0.0362	$2 \cdot 10^{-4}$	0.0364	0.0364	$\approx 0 \cdot 10^{-4}$
Hungria	0.0363	0.0362	$1 \cdot 10^{-4}$	0.0363	0.0364	$-1 \cdot 10^{-4}$
Suécia	0.0341	0.0339	$2 \cdot 10^{-4}$	0.0341	0.0341	$\approx 0 \cdot 10^{-4}$
Áustria	0.0324	0.0322	$2 \cdot 10^{-4}$	0.0324	0.0323	$1 \cdot 10^{-4}$
Dinamarca	0.0265	0.0263	$2 \cdot 10^{-4}$	0.0265	0.0265	$\approx 0 \cdot 10^{-4}$
Eslováquia	0.0265	0.0263	$2 \cdot 10^{-4}$	0.0265	0.0264	$1 \cdot 10^{-4}$
Finlândia	0.0260	0.0259	$1 \cdot 10^{-4}$	0.0259	0.0260	$-1 \cdot 10^{-4}$
Irlanda	0.0227	0.0226	$1 \cdot 10^{-4}$	0.0227	0.0227	$\approx 0 \cdot 10^{-4}$
Lituânia	0.0212	0.0212	$\approx 0 \cdot 10^{-4}$	0.0212	0.0213	$-1 \cdot 10^{-4}$
Letônia	0.0175	0.0175	$\approx 0 \cdot 10^{-4}$	0.0175	0.0176	$-1 \cdot 10^{-4}$
Eslovênia	0.0161	0.0161	$\approx 0 \cdot 10^{-4}$	0.0161	0.0162	$-1 \cdot 10^{-4}$
Estônia	0.0133	0.0130	$3 \cdot 10^{-4}$	0.0133	0.0131	$2 \cdot 10^{-4}$
Chipre	0.0097	0.0095	$2 \cdot 10^{-4}$	0.0097	0.0095	$2 \cdot 10^{-4}$
Luxemburgo	0.0076	0.0077	$-1 \cdot 10^{-4}$	0.0076	0.0077	$-1 \cdot 10^{-4}$
Malta	0.0072	0.0072	$\approx 0 \cdot 10^{-4}$	0.0072	0.0072	$\approx 0 \cdot 10^{-4}$
Quota JC = 0.6200						
Quota HE = 0.6373						

Tabela 5.13: Comparação dos resultados para a instância UE_{jc} .

A solução encontrada se aproxima bastante da apresentada na *Jagiellonian Compromise*. A quota calculada possui valor bem próximo, com uma diferença de $0.62 - 0.6373 = -0.0173$. Além disso, os power indices das soluções possuem power indices extremamente próximos, com a maior disparidade ocorrendo nos pesos (a Alemanha, por exemplo, possui diferença de 0.0016).

A fim de comparação, calculamos o erro $\sum_{i=1}^n |d_i - p_i|$ para a *Jagiellonian Compromise*, apesar dos autores não utilizarem essa métrica. O valor do erro é 0.0007, sendo $0.0016453 - 0.0007 = 0.0009453$ menor do que a nossa solução.

A melhor solução encontrada possui quota bem próxima da *Jagiellonian Compromise*,

mas também encontramos soluções com erro bem próximo à apresentada aqui que possuíam quota bem diferente. Por exemplo, encontramos uma solução que possui erro 0.0005 pior que a melhor encontrada, em que a quota dessa solução tinha valor aproximado de 0.43, bem distante do 0.62 da *Jagiellonian Compromise* e do 0.6373 da melhor solução que encontramos, apesar dos pesos serem bem próximos. A partir dos resultados obtidos, parece ser interessante definir uma boa função para codificar a quota, ou até mesmo elaborar outra maneira dela ser definida.

Nos resultados notamos que em algumas gerações em que há melhora na solução somente a quota é alterada, o que se aproxima do trabalho realizado por Zyczkowski e Slomczynski [2004] nesse cenário e de outros na literatura, que consiste na manipulação da quota para obter melhores jogos.

Capítulo 6

Conclusões

Conseguimos neste trabalho desenvolver dois diferentes algoritmos para solucionar o problema inverso dos power indices, um algoritmo exato e outro sem garantia de otimalidade. Para nossa implementação do algoritmo em programação linear inteira, derivado do trabalho de Kurz [2012], encontramos limitações quanto ao tamanho da instância, o que nos levou ao desenvolvimento de outra abordagem para a solução desse problema.

Nossa heurística evolutiva implementada através do framework BRKGA teve grande sucesso ao conseguir solucionar problemas de tamanho razoável e além disso, obteve soluções ótimas em tempo menor se comparados com as encontradas nos algoritmos exatos apresentados.

É importante ressaltar que nossa heurística tem a flexibilidade de poder solucionar o problema inverso para o power index que desejarmos. De modo semelhante, nossa abordagem se adequa facilmente a regras diferentes da *linear Shapley rule* e a *penrose square root rule* citadas nesse trabalho.

Em relação ao estudo de caso realizado sobre a proporcionalidade na Câmara dos Deputados, obtemos resultados que nos permitem dizer que o power index de cada unidade federativa no sistema atual de proporcionalidade apresenta grande distorção em relação a sua população, devido à presença dos limitantes superiores e inferiores. Analisando um sistema onde não há limites, demonstramos que, embora as UFs afetados pelo piso passem a ter um poder efetivo próximo a sua população, o estado de São Paulo passa a ter poder efetivo muito superior às outras UFs.

Uma distribuição baseada na *linear shapley rule* encontrada resolvendo o IPIP aproxima os power indices das unidades federativas de sua população, proporcionando uma alocação mais justa. Neste caso, o arredondamento causa algumas distorções, principalmente nas UFs de menor população, mas menores se comparados com o cenário estritamente proporcional.

Embora encontramos soluções satisfatórias para as instâncias utilizadas, ainda existe muito espaço para aprimoramento, seja ele na heurística evolutiva ou na PLI. Um maior estudo das propriedades de jogos para podermos compreender melhor as propriedades das coalizões pode ser importante para melhores resultados. Além disso, o aprofundamento nos melhores algoritmos para o cálculo dos power indices é de muita importância, já que é um fator fundamental na solução do IPIP.

Em relação ao PLI, aprofundar nas questões apresentadas no trabalho base e expandir

o estudo para o *branch and bound* pode oferecer uma melhora relevante no algoritmo apresentado, além de expandir no próprio artigo base de nossa implementação.

Em nossa heurística alterar a geração aleatória dos cromossomos do framework parece ser um caminho promissor no aperfeiçoamento do algoritmo, além de ser importante também estudar nossa função objetivo. Nos casos onde resolvemos o problema inverso sem ter uma quota previamente definida parece ser bastante importante determinar essa quota de maneira fundamentada.

Do ponto de vista da Câmara dos Deputados, a melhora da solução e a conseqüente redução da distância entre os power indices e a população normalizada poderia trazer maior confiança no cenário proposto, e também nos garantir que a distorção causada pelo arredondamento seja menor. O estudo de outras propostas de distribuição de poder pode trazer melhor discussão para este problema, dado que existem poucas investigações sobre o tema.

Como sugestões de trabalhos futuros para este tema da proporcionalidade, analisar o poder efetivo das regiões brasileiras nos parece interessante. Ademais, analisar eleições anteriores com as diferentes propostas de distribuição pode resultar em descobertas sobre a representação eleitoral nas unidades federativas sub e sobre-representados tanto do ponto de vista partidário como parlamentar, principalmente naquelas UFs afetados pelo piso.

Outro aspecto sobre a distribuição apresentada em que se pode agir é na alteração do critério de arredondamento ou até mesmo de outras restrições em relação ao número de deputados para cada unidade federativa, dado a relevância política desse caso.

Referências Bibliográficas

- H. Aziz, M. Paterson, e D. Leech. Efficient algorithm for designing weighted voting games. In *2007 IEEE International Multitopic Conference*, pages 1–6. IEEE, 2007.
- Y. Bachrach, E. Markakis, E. Resnick, A. D. Procaccia, J. S. Rosenschein, e A. Saberi. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20(2):105–122, 2010.
- J. F. Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317, 1964.
- J. F. Banzhaf III. Multi-member electoral districts. do they violate the "one man, one vote" principle. *The Yale Law Journal*, 75(8):1309–1338, 1966.
- J. F. Banzhaf III. One man, 3.312 votes: a mathematical analysis of the electoral college. *Vill. L. Rev.*, 13:304, 1968.
- J. Bilbao, J. Fernandez, A. J. Losada, e J. Lopez. Generating functions for computing power indices efficiently. *Top*, 8(2):191–213, 2000.
- E. F. Carmo, E. C. Rocha, e D. B. Figueiredo Filho. Aplicando o modelo de proporcionalidade degressiva na representação dos estados na câmara dos deputados. *Revista do Direito Público*, 7(3):191–204, 2012.
- B. de Keijzer. A survey on the computation of power indices. *Delft University of Technology*, 2008.
- B. de Keijzer. On the design and synthesis of voting games: Exact solutions for the inverse problems. 2009.
- A. F. de Oliveira. *Sobre a representação dos Estados na Câmara dos Deputados*. Senado Federal, Consultoria Legislativa, 2004.
- P. Dubey e L. S. Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979.
- S. Fatima, M. Wooldridge, e N. R. Jennings. An anytime approximation method for the inverse shapley value problem. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 935–942. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

- D. S. Felsenthal e M. Machover. The weighted voting rule in the eu's council of ministers, 1958–1995: Intentions and outcomes. *Electoral Studies*, 16(1):33–47, 1997.
- D. S. Felsenthal e M. Machover. Analysis of qm rules in the draft constitution for europe proposed by the european convention, 2003. *Social Choice and Welfare*, 23(1):1–20, 2004.
- IBGE. Tabelas de estimativas populacionais para os municípios e para as unidades da federação brasileiros em 01.07.2019. ftp://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2019/estimativa_dou_2019.ods, 2019. Acessado em 04 de Junho de 2020.
- P. King. Federation and representation. *Comparative Federalism and Federation*, 95, 1993.
- B. Klinz e G. J. Woeginger. Faster algorithms for computing power indices in weighted voting games. *Mathematical Social Sciences*, 49(1):111–116, 2005.
- S. Kurz. On the inverse power index problem. *Optimization*, 61(8):989–1011, 2012.
- S. Kurz, N. Maaser, e S. Napel. On the democratic weights of nations. *Journal of Political Economy*, 125(5):1599–1634, 2017.
- A. Laruelle e M. Widgrén. Is the allocation of voting power among eu states fair? *Public Choice*, 94(3-4):317–339, 1998.
- D. Leech. Computation of power indices. 2002a.
- D. Leech. Designing the voting system for the council of the european union. *Public Choice*, 113(3-4):437–464, 2002b.
- A. Lijphart. *Patterns of democracy: Government forms and performance in thirty-six countries*. Yale University Press, 2012.
- A. C. d. C. Lima e F. d. S. Ramos. Há desigualdade de poder entre os estados e regiões do brasil? uma abordagem utilizando o índice de poder de banzhaf e a penrose square root law. *Economia Aplicada*, 14(2):225–249, 2010.
- I. Mann e L. S. Shapley. Values of large games, vi. 1962.
- T. Matsui e Y. Matsui. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan*, 43(1):71–86, 2000.
- Y. Matsui e T. Matsui. Np-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science*, 263(1-2):305–310, 2001.
- N. R. Miller. A priori voting power and the us electoral college. In *Power, Voting, and Voting Power: 30 Years After*, pages 411–442. Springer, 2013.
- R. B. Myerson. *Game theory*. Harvard university press, 2013.

- J. M. Nicolau. As distorções na representação dos estados na câmara dos deputados brasileira. *Dados*, 40(3), 1997.
- L. S. Penrose. The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, 109(1):53–57, 1946.
- K. Prasad e J. S. Kelly. Np-completeness of some problems concerning voting games. *International Journal of Game Theory*, 19(1):1–9, 1990.
- L. S. Shapley e M. Shubik. A method for evaluating the distribution of power in a committee system. *American political science review*, 48(3):787–792, 1954.
- W. Słomczyński e K. Życzkowski. Jagiellonian compromise—an alternative voting system for the council of the european union. In *Institutional design and voting power in the European Union*, pages 67–82. Routledge, 2016.
- M. M. Soares e L. C. Lourenço. A representação política dos estados na federação brasileira. *Revista Brasileira de Ciências Sociais*, 19(56):113–127, 2004.
- P. Tannenbaum. Power in weighted voting systems. *Mathematica Journal*, 7(1):58–63, 1997.
- R. F. Toso e M. G. Resende. A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93, 2015.
- K. Zyczkowski e W. Slomczynski. Voting in the european union: The square root system of penrose and a critical point. *arXiv preprint cond-mat/0405396*, 2004.