

UNIVERSIDADE FEDERAL DO ABC  
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO



---

**Problemas de Balanceamento de Carga em Teoria dos  
Jogos Algorítmica**

---

Guilherme Naziozeno Barreto  
Profa. Dra. Carla Negri Lintzmayer

29 de Setembro de 2020

## **Resumo**

A Teoria dos Jogos Algorítmica é uma área relativamente nova e de grande crescimento em Teoria da Computação, que nos permite modelar situações onde vários agentes atuam guiados por interesses próprios, muito presentes em ambientes descentralizados como por exemplo a Internet. Esse projeto tem por objetivo a introdução do candidato à pesquisa em Algoritmos e Otimização Combinatória por meio do estudo de conceitos de Teoria dos Jogos, com foco no problema de Balanceamento de Carga, variante do clássico problema de Escalonamento de Tarefas. Este relatório faz um resumo dos principais tópicos que foram estudados no período.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Otimização Combinatória</b>	<b>5</b>
2.1	Algoritmos de Aproximação . . . . .	6
2.2	Problema de Empacotamento . . . . .	7
2.2.1	Algoritmo Guloso . . . . .	8
2.2.2	APTAS . . . . .	8
2.3	Balanceamento de Carga . . . . .	10
2.3.1	Algoritmo Guloso . . . . .	11
2.3.2	PTAS . . . . .	11
<b>3</b>	<b>Teoria dos Jogos Algorítmica</b>	<b>14</b>
3.1	Conceitos de Solução de Jogos . . . . .	15
3.2	Teorema de Nash . . . . .	16
3.3	Conceitos de Eficiência em Jogos . . . . .	20
3.4	Jogos Potenciais . . . . .	21
3.5	Jogos de Congestão . . . . .	22
<b>4</b>	<b>Jogos de Balanceamento de Carga</b>	<b>24</b>
4.1	Conceitos de Eficiência . . . . .	26
4.2	Mecanismos de Coordenação . . . . .	27
4.2.1	Política de Ativação . . . . .	27
4.2.2	Taxação . . . . .	28
4.3	Modelos Determinísticos . . . . .	28
4.3.1	Máquinas Idênticas . . . . .	28
4.3.2	Convergência . . . . .	30
4.3.3	Máquinas Uniformemente Relacionadas . . . . .	31
4.4	Convergência . . . . .	35
4.5	Modelos Aleatórios . . . . .	35
4.5.1	Máquinas Idênticas . . . . .	36
4.5.2	Máquinas Uniformemente Relacionadas . . . . .	37
<b>5</b>	<b>Conclusão e Perspectivas</b>	<b>42</b>
	<b>Referências</b>	<b>43</b>
<b>A</b>	<b>Probabilidade</b>	<b>47</b>

# 1 Introdução

*“Interactive Decision Theory would perhaps be a more descriptive name for the discipline usually called Game Theory.”*

---

*Robert Aumann*

Estratégia é um mecanismo vital em qualquer conflito de interesses, e aparece nas mais adversas situações, seja em economia, política, biologia, computação, etc. Em termos gerais, pode-se dizer que o cerne da Teoria dos Jogos é o estudo de tais interações estratégicas. Assim, busca-se compreender matematicamente o comportamento global de agentes coerentes (i.e., agentes que tomam as melhores decisões com base em suas informações) em interações estratégicas visando otimizar seus anseios em certo sentido.

Um jogo pode ser visto como uma tupla composta por um conjunto (finito ou não) de agentes, um perfil de estratégias (ou decisões) admissíveis e uma função que associa a cada estratégia um custo. É necessário tomar também diversas escolhas sobre a estrutura do jogo, as quais estão sujeitas às mais diversas variações como por exemplo jogos sequenciais, jogos de informação incompleta ou jogos cooperativos.

Noções do que viria a se tornar a Teoria dos Jogos podem ser encontradas na antiguidade, por exemplo nos textos do Talmud (como no problema do contrato de casamento) ou nos textos de Platão, como na situação a seguir [1].

*“Consider a soldier at the front, waiting with his comrades to repulse an enemy attack. It may occur to him that if the defense is likely to be successful, then it isn’t very probable that his own personal contribution will be essential. But if he stays, he runs the risk of being killed or wounded – apparently for no point. On the other hand, if the enemy is going to win the battle, then his chances of death or injury are higher still, and now quite clearly to no point, since the line will be overwhelmed anyway. Based on this reasoning, it would appear that the soldier is better off running away regardless of who is going to win the battle. Of course, if all of the soldiers reason this way – as they all apparently should, since they’re all in identical situations – then this will certainly bring about the outcome in which the battle is lost. Of course, this point, since it has occurred to us as analysts, can occur to the soldiers too. Does this give them a reason for staying at their posts? Just the contrary: the greater the soldiers’ fear that the battle will be lost, the greater their incentive to get themselves out of harm’s way. And the greater the soldiers’ belief that the battle will be won, without the need of any*

*particular individual's contributions, the less reason they have to stay and fight. If each soldier anticipates this sort of reasoning on the part of the others, all will quickly reason themselves into a panic, and their horrified commander will have a rout on his hands before the enemy has even engaged."*

*Platão, em "Laches"*

No decorrer da história, diversas situações de conflito foram pensadas nos termos hoje compreendidos pela Teoria dos Jogos, podendo ser encontradas também em René Descartes, William Shakespeare, Thomas Hobbes e Adam Smith, para citar alguns. Embora houvessem trabalhos tangenciais de autores como Augustin Cournot (1838), Francis Edgeworth (1881) e Emile Borel (1921), sua formalização matemática veio a ocorrer apenas em 1944, com a publicação de "*Theory of Games and Economic Behavior*" por John von Neumann e Oskar Morgenstern, que resultou também em desenvolvimentos em microeconomia e em teoria de utilidade [2, 3, 4]. Este desenvolvimento póstumo possivelmente está associado ao desenvolvimento tardio de aplicações da matemática nas ciências sociais. É interessante notar que os trabalhos de Neumann em Computação e Teoria dos Jogos viriam posteriormente a convergir para uma área comum.

Para elucidar uma situação compreendida pela teoria, consideremos o seguinte exemplo: um conjunto de viajantes deseja partir de carro simultaneamente de uma origem  $o$  para um destino final comum  $d$ , cada qual buscando o caminho mais rápido. Suponhamos que existem apenas dois caminhos de  $o$  para  $d$ , um superior  $u$  e um inferior  $\ell$ . Pelo caminho superior, passa-se pelo ponto  $v$ , enquanto que pelo inferior por  $w$ . Assim, os dois caminhos possíveis são  $u : o \rightarrow v \rightarrow d$  e  $\ell : o \rightarrow w \rightarrow d$ . Sabe-se que os tempos de viagem  $\mathcal{C}(a, b)$  de  $a$  para  $b$  são:  $\mathcal{C}(o, v) = x$ ,  $\mathcal{C}(v, d) = 1$ ,  $\mathcal{C}(o, w) = 1$  e  $\mathcal{C}(w, d) = x$ , em que  $x$  é a porcentagem de agentes que utiliza este caminho em relação ao número total de agentes. Assim, o custo total tanto no superior quanto no inferior é dado por  $\mathcal{C}(u) = 1 + x_u$  e  $\mathcal{C}(\ell) = 1 + x_\ell$ , sendo  $x_u + x_\ell = 1$ . É de se esperar, então, que  $x_u = x_\ell = 0.5$ , e os agentes se dividam igualmente entre os dois caminhos. Assim, o tempo total percorrido é de uma hora e meia para todos os agentes, independentemente da escolha.

No entanto, na expectativa de melhorar o tempo de viagem, criou-se uma conexão entre  $v$  e  $w$ , de tal forma que  $\mathcal{C}(v, w) \approx 0$ . O que deveria ocorrer então? Ora, existem quatro possibilidades:  $c_1 : o \rightarrow v \rightarrow d$ ,  $c_2 : o \rightarrow v \rightarrow w \rightarrow d$ ,  $c_3 : o \rightarrow w \rightarrow d$  e  $c_4 : o \rightarrow w \rightarrow v \rightarrow d$ . Claramente, a pior decisão é escolher  $c_4$ , uma vez que o tempo é fixo em 2 horas. Por experiência no trânsito, os viajantes individualmente devem chegar à conclusão de que é mais prudente escolher  $c_2$ , uma vez que os caminhos  $o \rightarrow v$  e  $w \rightarrow d$  possuem tempo de viagem variável. Assim, todos optam pelo mesmo caminho. E acabam gastando 2 horas da

mesma maneira!

Este problema é conhecido como paradoxo de Braess, e mostra que o comportamento egoísta dos viajantes pode acabar sendo agravado se este não for levado em consideração no projeto de mecanismos [5]. Tais configurações muitas vezes levam os agentes para situações sub-ótimas, que poderiam ser evitadas pelo altruísmo.

Este estado de estabilidade alcançado pelos agentes é conhecido como equilíbrio de Nash em jogos não-cooperativos. Genericamente, conceitos de solução são os métodos utilizados para se estudar jogos, podendo ser definidos grosso modo como funções que recebem um jogo e devolvem um conjunto de estratégias ótimas em certo sentido. O equilíbrio de Nash é um exemplo de conceito de solução, que retorna as estratégias de equilíbrio, isto é, estratégias em que nenhum agente pode unilateralmente melhorar sua satisfação. Conceitos de solução são, assim, uma tentativa de se descrever a dinâmica de jogos.

O surgimento e crescimento da Internet trouxe consigo diversas interações descentralizadas, tornando altamente relevante a compreensão de sua dinâmica e a busca por políticas mais eficientes. Em 1999, as publicações de Nisan e Ronen [6] e Koutsoupias e Papadimitriou [7] viriam a se tornar a fundação da Teoria Algorítmica dos Jogos, com a conceitualização dos conceitos de projeto de mecanismos e preço de anarquia. Estes, juntamente com Roughgarden e Tardos [8] ganharam o prêmio Gödel em 2012 por suas contribuições para o desenvolvimento da área. Nas palavras de Nisan e Ronen [6]:

*“A large part of research in computer science is concerned with protocols and algorithms for inter-connected collections of computers. The designer of such an algorithm or protocol always makes an implicit assumption that the participating computers will act as instructed – except, perhaps, for the faulty or malicious ones. With the emergence of the Internet as the platform of computation, this assumption can no longer be taken for granted. Computers on the Internet belong to different persons or organizations and will likely do what is most beneficial to their owners. We cannot simply expect each computer on the Internet to faithfully follow the designed protocols or algorithms. It is more reasonable to expect that each computer will try to manipulate it for its owners’ benefit. Such an algorithm or protocol must therefore be designed in advance for this kind of behavior!”*

A Teoria dos Jogos Algorítmica é um campo emergente e de grande crescimento nos últimos 21 anos que estuda a competitividade no contexto computacional. Esta difere em essência da Teoria Clássica de Jogos principalmente porque modela aplicações através de problemas de otimização concretos, buscando soluções ótimas, resultados de impossibilidade, limitantes inferiores e superiores em aproximações e considera questões de complexidade computacional como restrições na viabilidade do comportamento dos participantes. Estes

resultados são importantes não apenas no contexto computacional, uma vez que a dificuldade no tratamento deve também ser experienciada por um grupo de agentes em uma situação análoga.

O *problema de balanceamento de carga* é clássico em redes e sistemas distribuídos. De forma geral, temos um conjunto de tarefas que deve ser executado em um conjunto de máquinas e é preciso balancear a carga entre as máquinas de forma a explorá-las eficientemente. Tradicionalmente, procura-se definir um algoritmo que calcule a distribuição das tarefas. Na abordagem de Teoria dos Jogos, supomos que não existe uma forma de controle geral que possa fazer tal distribuição. Ao invés disso, cada tarefa é controlada por um agente, que provavelmente irá querer atribuir sua tarefa à máquina que possuir menos carga.

Este relatório descreve um estudo dos conceitos básicos de Teoria dos Jogos Algorítmica e, em particular, do problema de balanceamento de carga nesse contexto. A Seção 2 trata de alguns fundamentos de Otimização Combinatória, e descreve alguns dos problemas clássicos em balanceamento de carga. A Seção ?? visa dar uma introdução aos temas de Teoria dos Jogos e seus resultados principais necessários para o relatório. A Seção ?? visa aplicar os conceitos desenvolvidos ao longo do texto para estudar os jogos de balanceamento de carga. Por fim, a Seção 5 apresenta nossas considerações finais.

## 2 Otimização Combinatória

“It is a profound and necessary truth that the deep things in science are not found because they are useful; they are found because it was possible to find them.”

---

*Robert Oppenheimer*

Diversos problemas em Computação podem ser tratados como problemas de maximização (ou minimização), em que se quer encontrar uma solução ótima a partir de determinado critério, tendo-se determinadas restrições. Em domínio discreto, estes problemas possuem natureza combinatorial. Embora se tenha um conjunto finito de possíveis soluções, são necessários diversos métodos para se tratar grande parte destes problemas, pois a quantidade de combinações possíveis em geral é enorme.

A noção de algoritmo é básica em toda a Teoria de Computação, e aqui a tomaremos como intuitiva. Grosso modo, um algoritmo é uma sequência bem definida de instruções para se encontrar uma solução a determinado problema, dadas suas restrições e estruturas. A rigor, todo algoritmo clássico pode ser visto como uma máquina de estados abstratos [9].

Um problema de otimização  $\Pi$  é caracterizado por um conjunto de instâncias (entradas)  $\mathcal{I}$ , um conjunto de todas as possíveis soluções  $S(I)$  para uma instância  $I \in \mathcal{I}$  e uma função  $f : S(I) \rightarrow \mathbb{R}$  que atribui um valor para cada solução. Assim, um problema de maximização é definido da seguinte forma.

**Definição 1.** *Dada uma instância  $I \in \mathcal{I}$ , o problema de maximização se resume a encontrar  $s' \in S(I)$  tal que*

$$\forall s \in S(I), f(s') \geq f(s).$$

No decorrer do texto, designaremos a solução ótima de um problema  $\Pi$  para uma dada instância  $I$  por  $\text{opt}(I)$ .

Pode-se representar toda instância de um problema computacional como uma sequência de símbolos descritos por algum alfabeto. Assim, a noção de tamanho de uma instância está bem definida, e pode ser vista como o número de elementos descritos por esta estrutura (seja um conjunto, uma sequência, etc.). Em geral, existem diversos algoritmos possíveis para um mesmo problema computacional. Assim, é necessária uma medida de eficiência, como forma de se comparar e certificar a melhor forma de tratamento. Para tanto, utilizaremos a notação

de Bachmann-Landau como forma de descrever o comportamento assintótico dos algoritmos. Intuitivamente, quanto maior o tamanho da entrada, mais difícil e mais demorado se é para encontrar uma solução, uma vez que serão necessárias mais operações a se realizar. Assim, não existe algoritmo que possua um tempo de execução que decresça com o tamanho da entrada.

Algoritmos são considerados eficientes quando seu tempo de execução pode ser descrito como uma função polinomial no tamanho da instância de entrada. Infelizmente, existem vários problemas de otimização para os quais não se conhece algoritmos eficientes que os resolvam. Em geral, tais problemas são *NP*-difíceis, e encontrar um algoritmo eficiente para um deles significaria encontrar algoritmos eficientes para todos eles, provando que  $P = NP$ .

## 2.1 Algoritmos de Aproximação

Alguns problemas de otimização *NP*-difíceis permitem aproximação a um grau arbitrário. Formalizaremos esta noção a seguir. Seja  $\Pi$  um problema de otimização com função objetivo  $f_\Pi$ . Diremos que um algoritmo  $\mathcal{A}$  é um esquema de aproximação para  $\Pi$  se para uma entrada  $(I, \varepsilon)$  com  $I$  sendo uma instância de  $\Pi$  e  $\varepsilon > 0$  um parâmetro de erro, sua saída é uma solução  $s$  de tal modo que  $f_\Pi(I, s) \leq (1 + \varepsilon)\text{opt}(I)$  se  $\Pi$  é um problema de minimização. Denotaremos por  $\mathcal{A}(I)$  o valor de  $f_\Pi(I, s)$  da solução obtida por  $\mathcal{A}$ .

Denotaremos o tamanho de uma instância  $I$  por  $|I|$ , e por  $|I_u|$  o tamanho unário. Diz-se que um dado algoritmo para um problema  $\Pi$  é eficiente se para uma dada instância  $I$ , seu tempo de execução é limitado polinomialmente por  $|I|$ . Se este for limitado polinomialmente por  $|I_u|$ , o designaremos como algoritmo pseudo-polinomial.

**Definição 2.** Um algoritmo  $\mathcal{A}$  será dito esquema de aproximação em tempo polinomial (PTAS) se, para  $\varepsilon > 0$  fixo, seu tempo de execução é limitado por um polinômio no tamanho da instância  $I$ , para qualquer instância  $I$ .

**Definição 3.** Seja  $\mathcal{A}$  um algoritmo de aproximação de um dado problema de otimização  $\Pi$ . A razão de performance  $R_{\mathcal{A}}(I)$  de  $\mathcal{A}$  para a instância  $I$  é definida como

$$R_{\mathcal{A}}(I) = \frac{\mathcal{A}(I)}{\text{opt}(I)}.$$

**Definição 4.** Chama-se de taxa de desempenho absoluta  $R_{\mathcal{A}}$  a expressão dada como

$$R_{\mathcal{A}} = \inf\{r > 1 : R_{\mathcal{A}}(I) \leq r, \forall I \in D_\Pi\}.$$

**Definição 5.** Define-se a taxa de desempenho assintótico  $R_{\mathcal{A}}^\infty$  de um algoritmo de apro-

aproximação  $\mathcal{A}$  para um problema de otimização  $\Pi$  como

$$R_{\mathcal{A}}^{\infty} = \inf\{r > 1 : \exists N_0, R_{\mathcal{A}}(I) \leq r; \forall I \in D_{\Pi} : \text{opt}(I) \geq N_0\}.$$

**Definição 6.** Um esquema assintótico de aproximação em tempo polinomial (APTAS) é uma família de algoritmos  $\{A_{\varepsilon} : \varepsilon > 0\}$  tal que cada  $A_{\varepsilon}$  executa em tempo polinomial com o comprimento da instância e  $R_{A_{\varepsilon}}^{\infty} \leq 1 + \varepsilon$ .

**Definição 7.** Um algoritmo  $\mathcal{A}$  é chamado de esquema de aproximação totalmente em tempo polinomial (FPTAS) se o tempo de execução de  $\mathcal{A}$  é limitado por  $|I|$  e  $1/\varepsilon$ , para qualquer instância  $I$  e  $\varepsilon > 0$ .

**Teorema 1.** Seja  $p$  um polinômio e  $\Pi$  um problema de minimização NP-difícil tal que a função de objetivo  $f_{\Pi}$  é inteira e em qualquer instância  $I$  se tem  $\text{opt}(I) < p(|I_u|)$ . Se  $\Pi$  admite um FPTAS, então também admite um algoritmo pseudo-polinomial.

*Demonstração.* Suponha que existe um FPTAS para  $\Pi$  com tempo de execução em uma instância  $I$  e parâmetro  $\varepsilon$  igual a  $q(|I|, 1/\varepsilon)$ , onde  $q$  é um polinômio. Na instância  $I$ , façamos  $\varepsilon = 1/p(|I_u|)$  e executemos o FPTAS. Agora, a solução produzida terá valor no máximo igual a

$$(1 + \varepsilon)\text{opt}(I) < \text{opt}(I) + \varepsilon p(|I_u|) = \text{opt}(I) + 1.$$

Com este parâmetro  $\varepsilon$ , o FPTAS produz uma solução ótima. O tempo de execução será  $q(|I|, p(|I_u|))$ , i. e., polinomial em  $|I_u|$ . Assim, obtemos um algoritmo pseudo-polinomial para  $\Pi$ . ■

**Corolário 1.** Seja  $\Pi$  um problema de otimização NP-difícil satisfazendo as restrições do Teorema 1. Se  $\Pi$  é fortemente NP-difícil, então  $\Pi$  não admite um FPTAS assumindo  $P \neq NP$ .

*Demonstração.* Se  $\Pi$  admite um FPTAS, então também admite um algoritmo pseudo-polinomial pelo Teorema 1. Assim, não é fortemente NP-difícil, assumindo  $P \neq NP$ , levando a uma contradição. ■

## 2.2 Problema de Empacotamento

Suponha que se tenha  $n$  itens, cada qual com um dado tamanho, e alguns pacotes de tamanhos iguais. Gostaríamos de alocar estes itens utilizando a menor quantidade possível de pacotes. O problema pode assim ser formulado da seguinte forma.

**Definição 8** (Problema de Empacotamento). *Dada uma lista de números não-negativos  $a_1, \dots, a_k \leq 1$ , encontrar um número  $k \in \mathbb{N}$  e uma alocação  $f : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$  com  $\sum_{i:f(i)=j} a_i \leq 1$  para todo  $j \in \{1, \dots, k\}$  tal que  $k$  é mínimo.*

O problema de empacotamento é um dos problemas mais tradicionais em alocação de recursos, e diversos problemas mais complexos podem ser reduzidos nele. Claramente, tem-se como limitante inferior  $\lceil \sum_i a_i \rceil$ . O problema de empacotamento é fortemente NP-difícil, de tal forma que discutiremos a seguir alguns algoritmos de aproximação.

### 2.2.1 Algoritmo Guloso

Como primeira abordagem, é intuitivo buscar uma estratégia gulosa para o problema, de onde se tem o seguinte algoritmo.

#### Algoritmo First-Fit

1. Iterativamente, tente empacotar um item em algum dos pacotes já existentes.
2. Se não for possível, crie um novo pacote e coloque o item nele.

**Teorema 2.** *O algoritmo guloso first-fit possui fator de aproximação 2 para o problema de empacotamento.*

*Demonstração.* Seja  $I$  uma dada instância do problema de empacotamento. Note que, se o algoritmo utiliza  $m$  pacotes, então deve-se ter que ao menos  $m - 1$  pacotes estão preenchidos em pelo menos metade. Assim,

$$\text{opt}(I) > \sum_{i=1}^n a_i > \frac{m-1}{2}.$$

E, portanto,  $m \leq 2\text{opt}(I)$ . ■

### 2.2.2 APTAS

**Teorema 3.** *Para qualquer  $\varepsilon > 0$ , não existe algoritmo de aproximação que tenha fator  $3/2 - \varepsilon$  para o problema de empacotamento assumindo  $P \neq NP$ .*

*Demonstração.* Digamos que houvesse um algoritmo do tipo. Mostraremos que, se fosse o caso, este resolveria de modo ótimo um problema NP-difícil, o que causa contradição. Consideremos o problema da partição, em que são dados  $n$  números não-negativos  $a_1, \dots, a_n$

e queremos decidir se existem dois conjuntos que somem, cada um,  $\frac{1}{2} \sum_i a_i$ . Dada uma instância  $I$  do problema de partição, pode-se construir uma instância  $I' = (I, S = \frac{1}{2} \sum_i a_i)$  para o problema do empacotamento. Assim, se existe uma  $(3/2 - \varepsilon)$ -aproximação  $A(I')$  para o problema do empacotamento, pode-se escrever  $\text{opt}(I') \leq A(I') \leq (3/2 - \varepsilon)\text{opt}(I')$ . Se a instância  $I$  é *sim*, então  $\text{opt}(I') = 2$ . Logo,  $2 \leq A(I') \leq 3 - 2\varepsilon$ . Se, em contrapartida,  $I$  é *não*, então  $A(I') \geq 3$ . Pode-se então construir um algoritmo ótimo que resolva o problema de partição analisando tão somente se  $A(I') \geq 3$ . ■

**Lema 1.** *Seja  $\varepsilon > 0$  e  $K$  um inteiro não-negativo. Considere a restrição ao problema de empacotamento em que cada item tem tamanho de pelo menos  $\varepsilon$  e existem  $K$  tamanhos distintos. Existe um algoritmo de tempo polinomial que resolve este problema restrito.*

*Demonstração.* O número de pacotes está limitado por  $M = \lfloor 1/\varepsilon \rfloor$ , em que o número de combinações distintas possíveis de tipos de empacotamentos está limitado por  $R = \binom{M+K}{M}$ , sendo o número total de pacotes de no máximo  $n$ . Também, o número de possíveis empacotamentos está limitado por  $P = \binom{R+n}{R}$ . Note que  $\varepsilon$ ,  $M$  e  $R$  são constantes. Enumerando este conjunto e escolhendo o melhor empacotamento, garante-se então uma solução ótima em tempo polinomial. ■

**Lema 2.** *Seja  $\varepsilon > 0$ . Considere a restrição ao problema de empacotamento em que cada item tem tamanho pelo menos  $\varepsilon$ . Existe um algoritmo em tempo polinomial que resolve este problema restrito com fator de  $(1 + \varepsilon)$ .*

*Demonstração.* Seja  $I$  a instância dada. Ordenemos os  $n$  itens de modo crescente, e particionemos eles em  $K = \lceil 1/\varepsilon^2 \rceil$  grupos cada qual com  $Q = \lfloor n\varepsilon^2 \rfloor$  itens, exceto talvez por um deles. Seja a instância  $J$  construída arredondando para cima o tamanho de cada item em um grupo, para o tamanho do item com maior tamanho neste grupo. Assim, a instância  $J$  possui no máximo  $K$  itens de tamanhos distintos. Portanto, pelo Lema 1, podemos encontrar um empacotamento ótimo para  $J$ . Como o tamanho dos itens foram arredondamos para cima, este empacotamento também é válido para a instância original. Mostraremos que  $\text{opt}(J) \leq (1 + \varepsilon)\text{opt}(I)$ , finalizando a demonstração.

Seja  $J'$  outra instância, agora onde o tamanho de todos os itens são arredondados para o tamanho do menor item no grupo. Claramente,  $\text{opt}(J') \leq \text{opt}(I)$ , uma vez que os itens em  $J'$  são menores. Observe que uma solução para a instância  $J^*$  criada removendo-se os  $Q$  maiores elementos de  $J$  pode ser construída a partir de uma solução de  $J'$ . Assim,

$$\text{opt}(J^*) = \text{opt}(J) - Q \leq \text{opt}(J') \leq \text{opt}(I).$$

Uma vez que cada item em  $I$  possui tamanho de pelo menos  $\varepsilon$ , tem-se que  $\text{opt}(I) \geq n\varepsilon$ . Assim,  $Q = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon\text{opt}(I)$ . Portanto,  $\text{opt}(J) \leq (1 + \varepsilon)\text{opt}(I)$ . ■

**Teorema 4.** *Para qualquer  $\varepsilon \in (0, 1/2]$ , existe um algoritmo  $A_\varepsilon$  polinomial que encontra um empacotamento de até  $(1 + 2\varepsilon)\text{opt}(I) + 1$  pacotes.*

*Demonstração.* Seja  $I$  uma dada instância, e  $I'$  a instância obtida descartando os itens pequenos, com tamanho menor que  $\varepsilon$ . Pelo Lema 2, podemos encontrar um empacotamento de  $I'$  utilizando no máximo  $(1 + \varepsilon)\text{opt}(I')$  pacotes. Assim, colocaremos os itens pequenos de forma gulosa nos pacotes de  $I'$ , em que se abrem novos pacotes caso os itens não caibam nestes.

Assim, se pacotes adicionais não são necessários, tem-se um empacotamento de  $(1 + \varepsilon)\text{opt}(I') \leq (1 + \varepsilon)\text{opt}(I)$  pacotes. Caso contrário, seja  $M$  o número total de pacotes utilizados. Claramente, todos os pacotes menos um precisam estar preenchidos em no mínimo  $1 - \varepsilon$ . Portanto, a soma de todos os itens de  $I$  é de no mínimo  $(M - 1)(1 - \varepsilon)$ . Portanto, uma vez que este é um limitante inferior de  $\text{opt}(I)$ ,

$$M \leq \frac{\text{opt}(I)}{1 - \varepsilon} + 1 \leq (1 + 2\varepsilon)\text{opt}(I) + 1.$$

■

#### Algoritmo $A_\varepsilon$

1. Remova todos os itens cujo tamanho é menor que  $\varepsilon$ .
2. Arredonde o tamanho dos itens para obter um número constante de tamanhos distintos (Lema 1)
3. Encontre um empacotamento ótimo ao problema restrito (Lema 2)
4. Utilize este empacotamento para os itens originais
5. Coloque os itens removidos pelo algoritmo First-Fit.

## 2.3 Balanceamento de Carga

**Definição 9** (Problema de Balanceamento de Carga). *Dadas  $n$  tarefas com pesos  $\omega_1, \dots, \omega_n$  e  $m$  máquinas idênticas, encontrar uma alocação das tarefas nas máquinas  $A : N \rightarrow M$  de forma que o tempo de processamento  $\ell$  (também chamado de makespan) é o menor possível.*

O problema de balanceamento de carga também pode ser visto como um problema de

programação linear, da forma a seguir [10]:

$$\begin{array}{ll}
 \text{minimize} & \ell \\
 \text{sujeito a} & \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \\
 & \sum_{j \in N} x_{ij} p_{ij} \leq \ell \quad \forall i \in M \\
 & x_{ij} \in \{0, 1\} \quad \forall i \in M, j \in N
 \end{array}$$

As variáveis  $x_{ij}$  indicam se a tarefa  $j$  está alocada na máquina  $i$  e o objetivo é minimizar o *makespan*  $t$ . A primeira restrição indica que cada tarefa  $j$  está alocada em apenas uma máquina, e a segunda restringe a carga nas máquinas.

**Proposição 1.** *O valor  $LB = \max \left\{ \frac{1}{m} \sum_{i=1}^n \omega_i, \max_i \{\omega_i\} \right\}$  é um limite inferior para o *makespan* ótimo.*

### 2.3.1 Algoritmo Guloso

**Teorema 5.** *O algoritmo first-fit para o problema de balanceamento de carga tem fator de aproximação 2.*

*Demonstração.* Seja  $m_i$  a máquina com maior carga na última iteração do algoritmo e  $j$  o índice da última tarefa alocada nesta máquina. Seja  $t_j$  o tempo em que a tarefa  $j$  começa a ser executada na máquina  $m_i$ . Como o algoritmo aloca as tarefas na máquina com menor carga, todas as máquinas estão ocupadas até  $t_j$ . Assim,  $t_j \leq LB \leq \text{opt}(I)$ . Também,  $\omega_j \leq \text{opt}(I)$ . Portanto, o *makespan* é  $t_j + \omega_j \leq 2\text{opt}(I)$ . ■

#### Algoritmo First-Fit

1. Aloque as  $m$  primeiras tarefas em ordem arbitrária.
2. Aloque as próximas tarefas sequencialmente nas máquinas com menor carga.

### 2.3.2 PTAS

Construiremos um esquema de aproximação em tempo polinomial para o problema de balanceamento de carga a seguir.

**Teorema 6.** *O problema de balanceamento de carga é fortemente NP-difícil.*

Pelo Corolário 1, portanto, não se pode construir um *FPTAS*. Dada uma instância  $I$  do problema de balanceamento de carga, observe que é possível criar um paralelo com o problema de empacotamento buscando empacotar as tarefas em pacotes de tamanho  $t$  cada. Seja  $B(I, t)$  o menor número de pacotes de tamanho  $t$  requeridos. O menor *makespan*  $\ell$  é então dado por  $\min\{t : B(I, t) \leq m\}$ .

Como mostramos,  $LB$  e  $2LB$  são os limitantes inferior e superior de  $\ell$ . Assim, podemos determinar o *makespan* ótimo através de uma busca binária neste intervalo.

Apresentaremos um algoritmo de programação dinâmica que encontra uma solução ótima em tempo polinomial para instâncias do problema de empacotamento onde existem  $k$  tamanhos distintos para os itens. Seja  $k$  um número fixo de tamanhos distintos, em que todos os pacotes possuem capacidade 1. Fixemos uma ordem para o tamanho dos objetos. Uma instância do problema de empacotamento de tarefas pode ser descrita por uma  $k$ -tupla  $(i_1, \dots, i_k)$  especificando o número de objetos de cada tamanho. Seja  $\text{BINS}(i_1, \dots, i_k)$  o número mínimo de pacotes necessários para abranger todos estes itens.

Para uma dada instância  $(n_1, \dots, n_k)$  com  $\sum_{i=1}^k n_i = n$ , computaremos inicialmente  $\mathcal{Q}$ , o conjunto de todas as  $k$ -tuplas  $(q_1, \dots, q_k)$  tal que  $\text{BINS}(q_1, \dots, q_k) = 1$  e  $0 \leq q_i \leq n_i$ ,  $1 \leq i \leq k$ . Percebe-se que  $|\mathcal{Q}|$  é de no máximo  $\mathcal{O}(n^k)$  elementos. Assim, computaremos todas as entradas na tabela  $k$ -dimensional  $\text{BINS}(i_1, \dots, i_k)$  para todo  $(i_1, \dots, i_k) \in \times_{i=1}^k \{0, \dots, n_i\}$ . A tabela é inicializada fazendo-se  $\text{BINS}(q) = 1$  para todo  $q \in \mathcal{Q}$ . Então, usasse a seguinte recorrência para se calcular as entradas faltantes

$$\text{BINS}(i_1, \dots, i_k) = 1 + \min_{q \in \mathcal{Q}} \text{BINS}(i_1 - q_1, \dots, i_k - q_k).$$

Computar cada entrada leva tempo de  $\mathcal{O}(n^k)$ . Assim, a tabela inteira pode ser computada em  $\mathcal{O}(n^{2k})$ , determinando  $\text{BINS}(n_1, \dots, n_k)$ .

A ideia da redução do problema de balanceamento de carga ao problema de empacotamento de tarefas é permitir um erro para computar o menor *makespan*. Este erro pode ser feito tão pequeno quanto se queira, às custas do tempo de execução. Seja assim  $\varepsilon$  o parâmetro de erro e  $t \in [LB, 2LB]$ . Dizemos que um objeto é *queno* se seu tamanho é menor do que  $t\varepsilon$ . Retiraremos estes objetos a primeiro momento. Assim, todos os itens  $\omega_j \in [t\varepsilon(1 + \varepsilon)^i, t\varepsilon(1 + \varepsilon)^{i+1})$  são substituídos por  $\omega'_j = t\varepsilon(1 + \varepsilon)^i$ , em que  $i \geq 0$ . Estes itens resultantes podem então assumir  $k = \lceil \log_{1+\varepsilon} \frac{1}{\varepsilon} \rceil$  valores distintos. Utilizamos, assim, o algoritmo  $\mathcal{A}_\varepsilon$  para determinar um empacotamento ótimo. Uma vez que tal processo reduz o tamanho dos itens por um fator de  $(1 + \varepsilon)$ , se considerarmos os itens originais o empacotamento determinado é válido para um pacote de tamanho  $t(1 + \varepsilon)$ . Permitindo este tamanho de pacote, colocamos então os itens pequenos de maneira gulosa, abrindo novos pacotes caso

necessário. Claramente, sempre que um novo pacote é aberto, todos os outros pacotes devem estar preenchidos em pelo menos  $t$ . Denotaremos por  $\alpha(I, t, \varepsilon)$  o número de pacotes de tamanho  $t(1 + \varepsilon)$  usados por este algoritmo.

**Lema 3.**  $\alpha(I, t, \varepsilon) \leq B(I, t)$ .

*Demonstração.* Se o algoritmo não abre novos pacotes para os itens pequenos, então o lema claramente vale, uma vez que todas as tarefas arredondadas para baixo foram empacotadas de modo ótimo em pacotes de tamanho  $t$ . Caso contrário, todos os pacotes menos o último possuem carga de pelo menos  $t$ . Assim, um empacotamento ótimo para  $I$  deve usar ao menos  $\alpha(I, t, \varepsilon)$  pacotes, uma vez que todos já estão preenchidos em pelo menos  $t$ . ■

Uma vez que  $\text{opt}(I) = \min\{t : B(I, t) \leq m\}$ , o Lema 3 resulta no corolário abaixo.

**Corolário 2.**  $\min\{t : \alpha(I, t, \varepsilon) \leq m\} \leq \text{opt}(I)$ .

Se  $\min\{t : \alpha(I, t, \varepsilon) \leq m\}$  pode ser determinado sem erro adicional durante a busca binária, então claramente pode-se utilizar o algoritmo para obter uma alocação cujo *makespan* é de  $(1 + \varepsilon)\text{opt}(I)$ . A busca binária é realizada no intervalo  $[LB, 2LB]$ . Assim, o comprimento do intervalo disponível é  $LB$  no começo da busca, e se reduz por um fator de 2 a cada iteração. A busca é realizada até que se tenha um intervalo disponível de comprimento  $\varepsilon LB$ . Isto requer  $\lceil \log_2 \frac{1}{\varepsilon} \rceil$  iterações. Seja  $T$  o ponto mais à direita do intervalo final. Vale então o lema a seguir.

**Lema 4.**  $T \leq (1 + \varepsilon)\text{opt}(I)$ .

*Demonstração.* Por construção, tem-se que  $\min\{t : \alpha(I, t, \varepsilon) \leq m\} \in [T - \varepsilon LB, T]$ . Portanto,

$$T \leq \min\{t : \alpha(I, t, \varepsilon) \leq m\} + \varepsilon LB.$$

Utilizando o Corolário 2 e utilizando-se do fato de  $LB \leq \text{opt}(I)$ , o lema segue. ■

**Teorema 7.** *O tempo de execução do algoritmo como um todo é  $\mathcal{O}(n^{2k} \lceil \log_2 \frac{1}{\varepsilon} \rceil)$ , em que  $k = \lceil \log_{1+\varepsilon} \frac{1}{\varepsilon} \rceil$ .*

*Demonstração.* O teorema segue trivialmente dos resultados anteriores. Como visto, o *makespan* produzido é de pelo menos  $(1 + \varepsilon)^2 \text{opt}(I) \leq (1 + 3\varepsilon) \text{opt}(I)$ . ■

### 3 Teoria dos Jogos Algorítmica

“An equilibrium is not always an optimum; it might not even be good. This may be the most important discovery of game theory.”

---

*Ivar Ekeland*

A Teoria dos Jogos Algorítmica é um campo de grande crescimento em Teoria de Computação e Economia, que tem surgido principalmente com a importância e transformação que a Internet tem trazido na sociedade contemporânea. Existem diversos problemas de grande interesse aplicado que são compreendidos pela área, como no comércio eletrônico e em sistemas distribuídos. Questões como quantificar a ineficiência inerente à falta de coordenação, complexidade de equilíbrio e o design de mecanismos são alguns pontos centrais que a área busca desenvolver [11].

A abordagem computacional aos problemas tradicionais em Teoria dos Jogos se justifica por seu poder de simulação, pois não há esperança de se descrever conceitos de solução a jogos e esperar que agentes cheguem a um certo estado de equilíbrio se uma máquina não for capaz de encontrá-lo. Isto é, não basta apenas determinar a convergência da dinâmica de um jogo se esta não ocorrer de modo eficiente.

É interessante notar a conexão entre Teoria dos Jogos e Sistemas Dinâmicos [12, 13]. Essencialmente, um jogo evolui no tempo e estamos interessados em entender como se dá sua dinâmica, de forma que diversos pontos compreendidos pelo formalismo de Sistemas Dinâmicos como equilíbrio, recorrência e invariância fornecem uma visão alternativa ao tema.

Os jogos podem ser estruturados das mais diversas formas, podendo por exemplo ser contínuos, infinitos, cooperativos, etc. Aqui, estamos interessados em estudar problemas de natureza combinatorial em que os agentes são guiados por interesses próprios. Assim, podemos definir um jogo estratégico da seguinte forma.

**Definição 10.** *Um jogo estratégico (finito e discreto) é uma tripla  $G = \langle N, \mathcal{S}, C \rangle$ , onde  $N$  é o conjunto de agentes com  $|N| \in \mathbb{N}$ ,  $\mathcal{S} = \{S_i\}_{i \in N}$  é o conjunto com todas estratégias  $S_i$  de cada agente  $i$  e  $C : S \rightarrow \mathbb{R}^{|N|}$  é a função combinada de custos dentro do perfil estratégico  $S = \times_{i \in N} S_i$ , onde  $C_i(s) \in \mathbb{R}$  é o custo do perfil de estratégias  $s \in S$  quando o agente  $i$  escolhe essa estratégia.*

Pode-se dizer que um jogo é uma interação estratégica em que seus participantes possuem comportamento racional desejando individualmente minimizar seus custos. De forma

alternativa, pode-se definir que a pretensão de um agente é a de maximizar sua utilidade  $U : S \rightarrow \mathbb{R}^{|N|}$  com  $U(s) = -C(s), \forall s \in S$ . Tanto custo quanto utilidade aqui possuem o papel de modelar a preferência em determinadas estratégias por parte dos agentes.

Computacionalmente, para especificar a dinâmica do jogo é necessário especificar também quais serão suas regras de execução, que serão omitidas neste capítulo.

**Definição 11.** Denotaremos por  $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_{|N|})$  o perfil estratégico composto por  $s$  sem o  $i$ -ésimo elemento. Assim, se escreve  $s = (s_i, s_{-i}), \forall i \in N$  e  $\forall s \in S$ .

Note que a melhor estratégia  $s_i$  que um agente pode escolher depende das estratégias  $s_{-i}$  de todos os outros agentes.

**Definição 12.** Diremos que uma estratégia  $s \in S$  é uma estratégia dominante se, para todo jogador  $i$  e todo perfil estratégico  $s' \in S$ , tem-se que

$$C_i(s_i, s'_i) < C_i(s'_i, s'_{-i}).$$

Uma estratégia dominante não necessariamente possui a maior utilidade para um determinado agente. Um exemplo bem conhecido desta situação é a tragédia dos comuns, em que os agentes são forçados a escolher estratégias de forma a impedir que outros agentes se coloquem em vantagem.

### 3.1 Conceitos de Solução de Jogos

Para se descrever o comportamento de jogos, buscamos estruturar conceitos de solução, isto é, metodologias para se descrever a interação entre os agentes. Inicialmente, podemos pensar em estados de equilíbrio, em que o jogo se torna estacionário. O conceito de solução mais comum em Teoria dos Jogos é o equilíbrio puro de Nash.

**Definição 13.** Seja  $G$  um jogo. Um perfil estratégico  $s \in S$  é um equilíbrio puro de Nash (PNE) de  $G$  se, para todo agente  $i$ , e todo desvio unilateral (isto é, nenhum agente troca de estratégia cooperativamente ou simultaneamente)  $s'_i \in S_i$ , vale

$$C_i(s) \leq C_i(s'_i, s_{-i}).$$

Isto é, um perfil estratégico é um PNE se nenhum agente consegue individualmente minimizar sua função de custo. Nem todo jogo  $G$  possui PNE. De fato, encontrar PNE é um problema NP-difícil [14]. Por isso, faz-se necessário a construção de conceitos de solução

menos restritivos e mais tratáveis computacionalmente. É importante observar que intratabilidade não é o único problema de um conceito de solução, mas também questões como unicidade, uma vez se perde poder preditivo.

**Definição 14.** *Distribuições  $\sigma_1, \dots, \sigma_k$  em estratégias  $S_1, \dots, S_k$  de um jogo  $G$  constituem um equilíbrio misto de Nash se para todo agente  $i \in N$  e todo desvio unilateral  $s'_i \in S_i$  vale*

$$\mathbb{E}_{s \in \sigma}[C_i(s)] \leq \mathbb{E}_{s \in \sigma}[C_i(s'_i, s_{-i})],$$

onde  $\sigma = \times_{i=1}^k \sigma_i$ .

**Definição 15.** *Uma distribuição  $\sigma$  em um perfil estratégico  $S_1 \times \dots \times S_k$  de  $G$  é um equilíbrio correlacionado se para todo agente  $i \in N$ , estratégia  $s_i \in S_i$  e desvio  $s'_i \in S_i$ ,*

$$\mathbb{E}_{s \in \sigma}[C_i(s)|s_i] \leq \mathbb{E}_{s \in \sigma}[C_i(s'_i, s_{-i})|s_i].$$

Vale então a seguinte relação

$$PNE \subset MNE \subset CE.$$

**Definição 16** ([15]). *Um equilíbrio  $k$ -robusto ( $k$ -SE) é um PNE que é resiliente a desvios de coalizão de tamanho máximo  $k$ .*

**Conjectura 1** ([16]). *O equilíbrio misto completo de Nash  $F$  é o pior equilíbrio de Nash. Isto é, é um equilíbrio  $F$  tal que  $\mathcal{C}(F) \geq \mathcal{C}(P)$  para qualquer equilíbrio de Nash  $P$ .*

Nem todo jogo converge para um equilíbrio de Nash, mesmo quando a existência deste é garantida. Pode-se, por exemplo, construir um jogo que é a junção de dois jogos com mesma estrutura mas estratégias diferentes: um com equilíbrio de Nash e outro não. Assim, a dinâmica do jogo vai ficar completamente restrita às condições iniciais, e pertencerá tão exclusivamente a um dos jogos. De fato, esta é uma das maiores críticas ao equilíbrio de Nash enquanto representativo da dinâmica dos jogos. Outras abordagens são descritas em termos de sistemas dinâmicos.

## 3.2 Teorema de Nash

O teorema de Nash é um dos resultados principais e mais gerais em Teoria dos Jogos, por demonstrar a existência de estados em equilíbrio para todo jogo estratégico. Essencialmente, o teorema cria uma conexão entre procurar um equilíbrio e encontrar um ponto fixo [17].

Assim, pode-se estender estes resultados de diversas maneiras como em Geometria Diferencial [18].

Surpreendentemente, o resultado de existência vem a ocorrer apenas no caso em que se permite aos agentes aleatorizar suas estratégias. Também, em sua versão discreta, esta fornece uma relação estreita com os problemas de coloração em grafos.

**Definição 17.** *Seja  $X$  um conjunto e  $f : X \rightarrow X$  uma função. Um ponto  $x \in X$  é chamado de ponto fixo de  $f$  em  $X$  se  $f(x) = x$ .*

**Lema 5** (Lema do Aperto de Mãos). *A soma dos graus de todos os vértices de um grafo não direcionado  $G$  é duas vezes o número de arestas de  $G$*

$$\sum_{u \in V(G)} d(u) = 2|E(G)|.$$

**Definição 18.** *Um triângulo tricromático é um triângulo em que cada um de seus vértices possui uma coloração diferente.*

**Definição 19.** *Uma coloração própria de um  $n$ -simplexo subdividido é uma atribuição de  $n + 1$  cores aos vértices da subdivisão de tal modo que cada subdivisão possui vértices com cores diferentes.*

**Teorema 8** (Lema de Sperner). *Para toda coloração própria de um triângulo subdividido, existe um número ímpar de triângulos tricromáticos.*

*Demonstração.* Seja  $G$  um grafo que possui um vértice em cada triângulo interno, mais um vértice externo que corresponde à região por fora do triângulo maior. O grafo  $G$  possui uma aresta para cada par de triângulos internos que compartilham um lado com um ponto verde e um ponto vermelho. Todo triângulo tricromático corresponde a um vértice de grau um de  $G$ . Observa-se que, com exceção do vértice externo, todos os outros vértices do grafo possuem grau menor ou igual a dois. Por construção, o grau do vértice externo deve ser um número ímpar. Pelo Lema 5, a soma dos graus de todos os vértices de grau ímpares deve ser um número par. Portanto, deve-se ter um número ímpar de vértices com grau 1, e portanto, um número ímpar de triângulos tricromáticos.

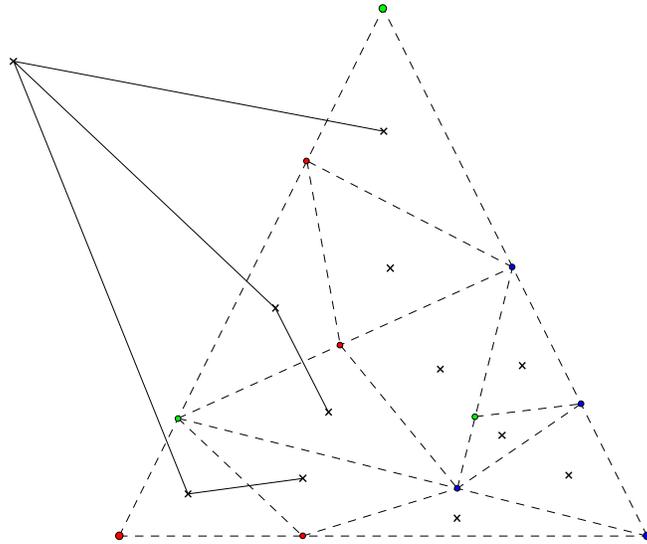


Figura 1: Representação do grafo  $G$ .

■

**Teorema 9** (Ponto fixo de Brouwer). *Seja  $X$  um conjunto do  $\mathbb{R}^n$  não vazio, compacto e convexo. Se  $f : X \rightarrow X$  é uma função contínua, então existe  $x \in X$  tal que  $x$  é um ponto fixo de  $f$ .*

*Demonstração.* Considere o caso especial em que  $C$  é um simplex em  $\mathbb{R}^2$ . Subdivida  $C$  em triângulos menores, de modo arbitrário. Colore-se um vértice  $x$  do triângulo de verde se  $f(x)$  está mais distante do vértice esquerdo de  $C$  do que de  $x$ ; vermelho se  $f(x)$  está mais distante do vértice superior de  $C$  do que de  $x$  e azul se  $f(x)$  está mais distante do vértice direito de  $C$  do que  $x$ , de forma análoga à figura 2. Se duas destas condições se aplicam a  $x$ , qualquer cor pode ser utilizada. Note que, se nenhuma delas se aplica,  $x$  é um ponto fixo. Este procedimento é uma coloração própria, e pelo Lema de Sperner deve-se ter ao menos um triângulo tricromático, representando um triângulo com vértices levados a direções diferentes por  $f$ . Tomando assim uma sequência de mais subdivisões, tem-se uma sequência decrescente de triângulos tricromáticos. Como  $C$  é compacto, os centros destes triângulos contêm uma sequência que converge para um ponto  $x^*$  em  $C$ . Como  $f$  é contínua, no limite,  $f(x^*)$  é tão distante dos três vértices que  $x^*$ . Assim,  $x^*$  é um ponto fixo de  $f$ .

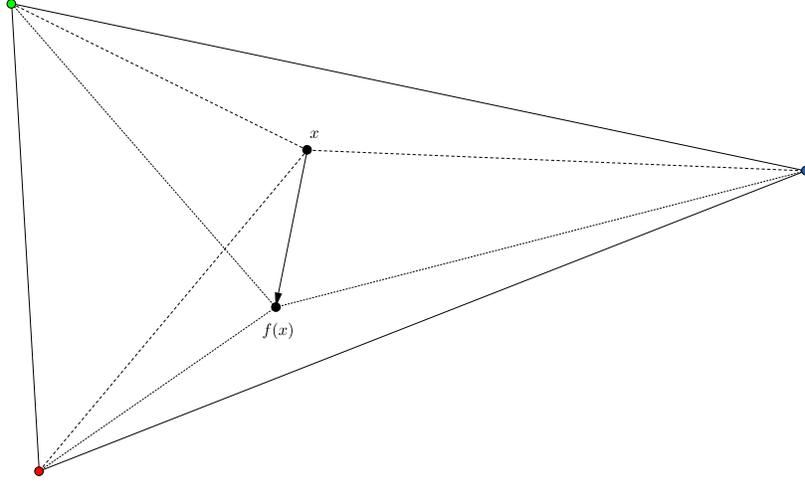


Figura 2: Esquema da triangularização.

■

**Definição 20.** O  $n$ -simplex regular  $\Delta_n$  é definido como

$$\Delta_n = \left\{ y \in \mathbb{R}^{n+1} \mid \sum_{i=0}^n y_i = 1, \forall i = 0, \dots, n; y_i \geq 0 \right\}.$$

Grosso modo, um  $n$ -simplex é uma generalização de um triângulo de dimensão  $n$  imerso em um espaço de dimensão  $n + 1$ . Uma estratégia mista de um jogador  $i$  em um jogo finito  $G$  é uma distribuição de probabilidades no conjunto  $S_i$  de estratégias puras. Formalmente, se define

$$\Delta(S_i) = \left\{ s_{ik, k \in S_i} \in \mathbb{R}^{|S_i|} : \sum_k s_{ik} = 1 \right\}.$$

**Teorema 10** (Nash [19]). *Todo jogo estratégico finito possui ao menos um equilíbrio misto de Nash.*

*Demonstração.* Faremos a seguir um rascunho da demonstração do Teorema de Nash. Sua versão completa pode ser encontrada em diversos textos, como em [20]. Considere um jogo com  $k$  agentes, com perfis estratégicos  $S_1, \dots, S_k$  e funções de custo  $\pi_1, \dots, \pi_k$ . Queremos definir uma função contínua  $f : C \rightarrow C$ , onde  $C$  é um conjunto convexo e compacto definido como  $C = \prod_{i=1}^k \Delta(S_i)$ , onde  $\Delta(S_i)$  é um simplex representando o conjunto de estratégias mistas em  $S_i$ ; de tal forma que os pontos fixos de  $f$  são um equilíbrio misto de Nash. Definiremos  $f$  separadamente para cada componente  $f_i : C \rightarrow \Delta(S_i)$ . Uma ideia natural é definir  $f_i$  como melhor resposta do agente  $i$  para os perfis estratégicos de todos os outros agentes.

Desta forma, no entanto, não se tem uma função bem definida. Assim, faz-se

$$f_i(x_i, x_{-i}) = \operatorname{argmax}_{x'_i \in \Delta(S_i)} g_i(x'_i, x),$$

em que

$$g_i(x'_i, x) = \mathbb{E}_{s_i \in x'_i} [\pi_i(s)] - \|x'_i - x_i\|^2.$$

O primeiro termo da função  $g_i$  encoraja a melhor resposta enquanto o último desencoraja grandes mudanças na estratégia  $x_i$  do agente  $i$ . Como a função  $f_i$  é estritamente côncava em  $x'_i$ ,  $f_i$  está bem definida. A função  $f = (f_1, \dots, f_k)$  é contínua. Por construção, todo equilíbrio misto de Nash de um dado jogo é um ponto fixo de  $f$ . Para verificar este resultado, suponha que  $x$  não é um equilíbrio, com o agente  $i$  podendo melhorar seu retorno esperado desviando de  $x_i$  para  $x'_i$ . Mostra-se que, para qualquer  $\varepsilon > 0$  suficientemente pequeno,  $g_i((1 - \varepsilon)x_i + \varepsilon x'_i, x) > g_i(x_i, x)$ . Portanto que  $x$  não é um ponto fixo de  $f$ . ■

**Teorema 11** (Glicksberg [21]). *Todo jogo contínuo possui um equilíbrio misto de Nash.*

**Definição 21.** *A função  $\beta_i : S \rightarrow 2^{S_i}$  definida como*

$$\beta_i(s) = \operatorname{argmin}_{k \in S_i} C_i(k, s_{-i})$$

*é chamada de função de melhor resposta para o agente  $i$ .*

**Proposição 2.** *Um perfil estratégico  $s$  é um equilíbrio de Nash se e somente se  $s \in \beta(s)$ , onde  $\beta(s) = \times_{i \in N} \beta_i(s)$ .*

Diz-se que um agente  $i$  está insatisfeito (ou em condição sub-ótima) se  $s_i \notin \beta_i(s)$ . Dado um perfil estratégico inicial  $s^0$ , uma sequência de respostas de melhoria é uma sequência  $s^0, s^1, \dots$ , em que para cada  $\tau$  existe um agente  $i$  de tal forma que  $s^{\tau+1} = (s'_i, s_i^\tau)$ , onde  $s'_i \in \beta(s_i^\tau)$ .

### 3.3 Conceitos de Eficiência em Jogos

**Definição 22.** *O preço de anarquia de um jogo  $G$  é a maior razão entre o valor de um perfil de estratégias em equilíbrio e o valor de um perfil de estratégias sociais ótimo. Isto é,*

$$PoA(G) = \max_{s \in \text{Nash}(S)} \frac{C(s)}{\text{opt}(G)}.$$

**Definição 23.** *O preço de estabilidade de um jogo estratégico  $G$  é a menor razão entre o valor de um perfil de estratégias em equilíbrio e o valor de um perfil de estratégias sociais*

ótimo.

$$PoS(G) = \min_{s \in Nash(S)} \frac{C(s)}{\text{opt}(G)}.$$

Uma regra de desvio  $D$ , como definida por [22], é um mapa que associa  $S$  a  $N$  de tal forma que, dado um perfil estratégico  $s$ , se escolhe um agente dentre todos os agentes insatisfeitos em  $s$ , o qual muda sua estratégia via melhor resposta.

**Definição 24.** A ineficiência  $\alpha$  de uma regra de desvio é, entre todos os perfis estratégicos iniciais  $s^0$ , definida como a pior razão entre o custo social do pior equilíbrio de Nash alcançado por  $D$  e o custo social do melhor equilíbrio de Nash alcançado por  $s^0$ . Isto é,

$$\alpha = \max_{s^0 \in S} \max_{s \in Nash(S)} \frac{C(s)}{C(s_0^*)},$$

onde  $s_0^* = \text{argmin}_{s \in Nash(s^0)} C(s)$ .

**Teorema 12.** Para todo jogo  $G$  e toda regra de desvio  $D$ , a ineficiência de  $D$  é de no mínimo 1 e limitada inferiormente pelo preço de anarquia.

### 3.4 Jogos Potenciais

**Definição 25.** Uma função  $\Phi : S \rightarrow \mathbb{R}$  é uma função potencial para um jogo estratégico  $G$  se, para todo  $i \in N$  e todo  $s \in S$  tem-se que

$$C_i(s'_i, s_{-i}) - C_i(s) \geq 0 \Leftrightarrow \Phi(s'_i, s_{-i}) - \Phi(s) \geq 0,$$

para todo  $s'_i \in S_i$ . Uma função potencial é dita exata se

$$C_i(s'_i, s_{-i}) - C_i(s) = \Phi(s'_i, s_{-i}) - \Phi(s).$$

**Corolário 3.** Seja  $G$  um jogo potencial finito. O perfil de estratégias que minimiza sua função potencial é um equilíbrio de Nash.

*Demonstração.* Se  $\Phi$  é uma função potencial de  $G$  e  $s$  um perfil estratégico que minimiza  $\Phi$ , tem-se que  $\Phi(s'_i, s_{-i}) \geq \Phi(s)$  para qualquer estratégia  $s'_i$ . Por definição, então, vale que  $C_i(s'_i, s_{-i}) \geq C_i(s)$ , o que mostra que  $s$  é um equilíbrio. ■

**Teorema 13.** Um jogo potencial finito sempre possui um equilíbrio puro de Nash.

*Demonstração.* Se existe função potencial, então existe um perfil estratégico que a minimiza, uma vez que o conjunto é finito. Logo, existe Equilíbrio Puro de Nash. ■

**Teorema 14.** *Considere um jogo potencial  $G$ , com função potencial  $\Phi$ . Se, para todo  $s \in S$ ,*

$$\frac{1}{\alpha}C(s) \leq \Phi(s) \leq \beta C(s),$$

*para constantes  $\alpha, \beta > 0$ , então o preço de estabilidade é de no máximo  $\alpha\beta$ .*

### 3.5 Jogos de Congestão

Os jogos de congestão foram introduzidos em 1973 por Robert Rosenthal, e descrevem em geral jogos em que se possui algum tipo de saturação [23]. Tais jogos são de particular interesse por suas propriedades, como mostrado a seguir [24, 25]. De fato, até mesmo o dilema do prisioneiro pode ser formulado como um jogo de congestão. Em todas as situações neste texto, tratamos apenas os casos (sem explicitamente mencionar) em que todos os agentes decidem suas estratégias simultaneamente. Jogos por turno são também de grande relevância na modelagem de problemas reais [26].

**Definição 26.** *Um jogo de congestão  $G = \langle N, M, \mathcal{S}, C \rangle$  é um jogo estratégico em que se tem Um conjunto  $N$  de agentes, um conjunto  $M$  de recursos, para cada agente  $i \in N$ , um conjunto não-vazio de ações ou estratégias  $S_i$ , em que cada estratégia  $s_i \in S_i$  é um subconjunto de  $M$  e uma função custo  $c_r(k)$  para cada agente que faz uso do recurso  $r$  dado que existe um número  $k$  de agentes utilizando  $r$ .*

O custo total que o agente  $i$  paga quando escolhe a estratégia  $s_i$  é a soma dos custos dos recursos em  $s_i$ . Em outras palavras, se  $k_j$  denota a quantidade de agentes que usam o recurso  $j$ , então  $i \in N$  paga

$$\sum_{j \in s_i} c_j(k_j).$$

Assim, a função custo do agente  $i$  dada uma estratégia global  $A$  é dada por

$$\pi_i(A) = \sum_{a \in A_i} c_a(\sigma_a(A)),$$

em que se tem  $\sigma_a(A) = |\{i \in N : a \in A_i\}|$ .

O seguinte é um exemplo de jogo de congestão. Em uma dada rede formada por diversas estradas, um conjunto de pessoas deve viajar (simultaneamente) de modo a sair de determinadas origens com objetivo de chegar a seus destinos. O tempo que se gasta para atravessar qualquer estrada é função no número de pessoas que viaja por esta estrada.

**Teorema 15.** A função  $\Phi : A \rightarrow \mathbb{R}$  dada por

$$\Phi(a) = \sum_{j=1}^{|M|} \sum_{\ell=1}^{\sigma_j(a)} \pi_j(\ell)$$

é uma função potencial para os jogos de congestão. Chama-se  $\Phi$  de potencial de Rosenthal.

*Demonstração.* Seja  $n = |M|$ . Suponhamos, sem perda de generalidade, que o agente  $n$  muda sua estratégia  $a$  para  $a' = (a_n, a_{-n})$ . Devemos notar que, quando  $j = n$ , tem-se que  $\sum_{\ell=1}^{\sigma_n(a)} \pi_j(\ell) = c_n(a)$ . Assim, se escreve

$$\Phi(a) = \sum_{\ell=1}^{\sigma_1(a)} \pi_1(\ell) + \cdots + c_n.$$

Portanto, vale que

$$\begin{aligned} \Phi(a) - \Phi(a') &= \left( \sum_{\ell=1}^{\sigma_1(a)} \pi_1(\ell) + \cdots + c_n(a) \right) - \left( \sum_{\ell=1}^{\sigma_1(a')} \pi_1(\ell) + \cdots + c_n(a') \right) \\ &= c_n(a) - c_n(a'). \end{aligned}$$

■

**Lema 6.** Seja  $S$  um estado qualquer de um jogo de congestão. Suponha que este estado muda para  $S'$  realizando-se um passo incremental que diminui o custo do agente  $i$  por  $\Delta$ . Então vale que  $\Phi(S') = \Phi(S) - \Delta$ .

**Definição 27.** Dois jogos  $G = \langle N, \mathcal{S}, C \rangle$  e  $G' = \langle N, \mathcal{S}', C' \rangle$  são isomorfos se para todo  $i \in N$ , existe uma bijeção  $\phi : S_i \rightarrow S'_i$  de tal modo que  $C_i(\times_{i \in N} S_i) = C'_i(\times_{i \in N} \phi_i(s_i))$ .

**Teorema 16.** Todo jogo potencial exato é isomorfo a um jogo de congestão.

**Corolário 4.** Todo jogo de congestão possui um equilíbrio puro de Nash.

*Demonstração.* Segue que, como o número de estados de perfis estratégicos possíveis é finito, deve-se haver algum estado que minimiza a função potencial. ■

## 4 Jogos de Balanceamento de Carga

“A arte de viver consiste em tirar o maior bem do maior mal.”

---

*Machado de Assis*

Suponhamos que, diferentemente do problema de balanceamento de carga, não mais se tenha um controle sobre como as alocações são realizadas, mas que estas fossem distribuídas de forma descentralizada entre diversos agentes, que possuem individualmente suas próprias intenções. Claramente, nestas condições, estaríamos tratando um problema completamente diferente. Em uma situação competitiva do gênero, buscar *insights* na Teoria dos Jogos é certamente intuitivo, e não só se aplica para modelar o comportamento humano como também para entender a interação entre máquinas e desenhar mecanismos que tornem esta interação mais eficiente.

Vale notar que jogos de balanceamento de carga são um caso especial de jogos de congestão, em que o conjunto de estratégias dos agentes são unitários. Trataremos apenas dos jogos não-cooperativos e de informação completa, mas existem diversas variações possíveis na construção destes jogos [27, 28, 29, 30, 31]. Formalizaremos o problema a seguir.

Um jogo de balanceamento de carga  $G$  é uma tupla  $\langle M, s, N, \omega \rangle$ , onde  $M = \{1, \dots, m\}$  é um conjunto de  $m$  máquinas de velocidades  $s = \{s_1, \dots, s_m\}$  e  $N = \{1, \dots, n\}$  é um conjunto de  $n$  tarefas com tamanhos  $\omega = \{\omega_1, \dots, \omega_n\}$ , cada qual controlada por um agente, com  $n \geq m$ . Assim, cada  $i \in N$  será tratado como tarefa ou agente, de forma indistinta. Cada agente  $i \in N$  tem como estratégia alocar sua tarefa em uma máquina  $j \in M$ , e a mesma será processada por  $j$  com peso  $\omega_{ij} = \omega_i/s_j$ . As alocações dos agentes definem, a cada instante, um mapa  $A : N \rightarrow M$  de como todas as tarefas são alocadas no sistema de máquinas. Denotaremos por  $A_j = \{i \in N : A(i) = j\}$  o conjunto de tarefas alocadas à máquina  $j$ . Cada agente  $i \in N$  atribui um custo  $C_i(j) = \ell_j(A)$  à máquina  $j$ , em que  $\ell_j(A)$  é a carga da máquina  $j$  em  $A$ , definida como

$$\ell_j(A) = \sum_{i \in A_j} \omega_{ij}.$$

Individualmente, os agentes visam minimizar seus custos, alocando seus processos nas máquinas de menor carga. A alocação  $A$  de agentes nas máquinas estará em equilíbrio quando nenhum agente conseguir unilateralmente realizar um passo incremental, isto é, quando não houver estratégias melhores para nenhum agente.

**Proposição 3.** *Seja  $G = \langle M, s, N, \omega \rangle$  uma instância do jogo de balanceamento de carga. Um mapa  $A: N \rightarrow M$  é um equilíbrio de Nash se e somente se  $\forall i \in N : \forall j \in M : c_i(A(i)) \leq c_i(j)$ .*

Para que o jogo esteja bem estruturado, precisamos definir suas regras, isto é, definir como os turnos, as alocações e os processamentos se darão. Um jogo por turno é um jogo em que, a cada instante de tempo, um agente decide sua estratégia.

**Definição 28.** *Dado um jogo de balanceamento de carga por turno  $G = \langle M, s, N, \omega \rangle$ , uma política de ativação é a política que, a cada rodada, define quais agentes trocarão suas estratégias.*

**Definição 29.** *Dado um jogo de balanceamento de carga  $G = \langle M, s, N, \omega \rangle$ , uma política de alocação  $P: N \rightarrow M$  é a política que cada agente escolhe para decidir suas estratégias.*

Dada uma alocação  $A$ , tomaremos as políticas de alocação dos agentes como  $P(i) = \operatorname{argmin}_{j \in M} \ell_j(A)$ , isto é, o custo dos agentes é tão somente a carga na máquina  $j$ . Outras variantes, no entanto, são interessantes porque em problemas reais os agentes geralmente não buscam exaustivamente a melhor solução, uma vez que existe um custo também para tanto.

Assim, a dinâmica de um jogo de balanceamento fica bem definida, e pode ser escrita no algoritmo a seguir.

### Dinâmica do Jogo de Balanceamento de Carga

1. Dada uma determinada política de ativação, escolhe-se iterativamente um agente  $i \in N$ .
2. Cada agente visa minimizar seu custo, alocando sua tarefa em uma máquina com base em uma política de alocação.

O custo social (também chamado de *makespan*) é dado por  $\mathbf{C}(A) = \max_{j \in M} \ell_j(A)$ . A partir deste, surgem diversas métricas para se avaliar como o comportamento individualista dos agentes leva a estados sub-ótimos.

Como veremos na proposição a seguir, um jogo de balanceamento de carga sempre alcança um estado de equilíbrio para máquinas uniformemente relacionadas (isto é, máquinas cujo custo de processamento é dado por  $\omega_{ij} = \omega_i/s_j, s_j \neq 1$ ).

**Proposição 4.** *Em qualquer instância do jogo de balanceamento de carga, existe ao menos um equilíbrio de Nash.*

*Demonstração.* Seja  $\langle M, s, N, \omega \rangle$  uma instância do jogo. Se um dado mapa de alocações  $A: N \rightarrow M$  não está em equilíbrio de Nash, então existe um agente  $i$  em uma máquina  $p$  que deseja mudar sua tarefa para, digamos, a máquina  $q$ , isto é,  $\ell_p(A) - \omega_{ip} > \ell_q(A)$ . Assim, o agente pode reduzir seu custo alocando sua tarefa em  $q$ . Seja  $A'$  uma alocação que diferencia de  $A$  apenas pela alocação de  $i$ . Então  $|\ell_p(A') - \ell_q(A')| < |\ell_p(A) - \ell_q(A)|$ , isto é, em  $A'$  a diferença entre as cargas de  $p$  e  $q$  se torna menor. Assim, tem-se

$$\sum_{j,k \in M} |\ell_j(A') - \ell_k(A')| < \sum_{j,k \in M} |\ell_j(A) - \ell_k(A)|,$$

e portanto a diferença de carga entre o sistema como um todo diminui. Mas, como o conjunto de possíveis alocações é finito e o sistema é limitado inferiormente (i.e., existe um mínimo de carga), então em algum momento se alcança uma alocação em que nenhum agente consegue individualmente reduzir seu custo, alcançando um equilíbrio de Nash. ■

Logo, vale dizer que os estados seguem de modo linear no tempo  $A^0 \rightarrow A^1 \rightarrow \dots \rightarrow A^t$  (ou seja, não existem ciclos) com a seguinte relação de ordem de satisfatibilidade  $A^t \succ A^{t-1} \succ \dots \succ A^0$ , uma vez que um determinado agente não mudaria sua alocação caso esta não reduzisse seu custo.

## 4.1 Conceitos de Eficiência

Uma das maneiras mais intuitivas de se buscar compreender o comportamento dos jogos de balanceamento (e jogos algorítmicos no geral) é buscar por alguma métrica de eficiência. Isto é, buscar uma métrica que indique quão distante de um estado ótimo encontra-se o sistema. Como, em grande parte dos jogos, a dinâmica converge para um equilíbrio de Nash, é interessante buscar por algo que una tais coeficientes. Assim, uma possível construção é formulada como se segue.

**Definição 30.** Para um dado inteiro  $m$ , seja  $G(m)$  o conjunto de todos os jogos de balanceamento com  $m$  máquinas. Para um desses jogos  $G \in G(m)$ , considere  $Nash(G)$  como o conjunto de todos os equilíbrios puros de Nash e  $\text{opt}(G)$  o custo da solução ótima para  $G$ , em que o custo social é minimizado. Assim, define-se o preço de anarquia como

$$PoA(m) = \max_{G \in G(m)} \max_{A \in Nash(G)} \frac{C(A)}{\text{opt}(G)}.$$

Isto é, o preço de anarquia é um modo de quantificar como o comportamento egoísta dos agentes prejudica o custo social. A formulação aqui pode ser reconstruída, no entanto, a depender do conceito de solução que se for mais propício considerar, tornando o conceito mais

representativo a depender do comportamento do sistema. Também, é comumente de interesse a ordem de complexidade de tempo para se alcançar o equilíbrio, uma vez que é uma forma de se estimar em quanto tempo os agentes alcançam o equilíbrio. Para qualquer variante do jogo de balanceamento, o caso estudado em que os agentes buscam por soluções ótimas para minimizar individualmente seu custo limita superiormente a ordem de complexidade. Limitantes para o *preço de anarquia* serão vistos mais adiante.

O resultado a seguir mostra que existe um *equilíbrio de Nash* que possui custo social ótimo, enquanto o Teorema 17 mostra que não é fácil obtê-lo.

**Proposição 5.** *O preço de estabilidade do jogo de balanceamento de carga é 1.*

*Demonstração.* Dada qualquer solução ótima (isto é, uma alocação que minimiza o *makespan*) do jogo de balanceamento de carga, pode-se construir um equilíbrio de Nash a partir de passos de incremento (alocações que minimizam o custo dos agentes), uma vez que estes nunca aumentam o custo social. ■

**Teorema 17** ([32]). *Encontrar um equilíbrio de Nash de custo social mínimo para o jogo de balanceamento de carga com duas máquinas uniformes é um problema NP-difícil.*

## 4.2 Mecanismos de Coordenação

Embora os jogos estratégicos sejam inerentemente sistemas distribuídos, existem metodologias que possibilitam se alcançar um certo controle das decisões dos agentes. Discutiremos sobre algumas dessas nas próximas seções.

### 4.2.1 Política de Ativação

Políticas de ativação são um dos possíveis mecanismos de coordenação da dinâmica dos jogos algorítmicos. Elas descrevem a forma como as alocações são processadas para as máquinas. Algumas políticas de ativação podem ser definidas, por exemplo, como escolher primeiro o agente com tarefa de maior tamanho ( $\max \omega_i$ ), escolher o agente com tarefa de menor tamanho ( $\min \omega_i$ ) ou escolher arbitrariamente.

Digamos que as tarefas sejam alocadas de tal forma que seus pesos estejam ordenados de modo decrescente,  $\omega_1 \geq \dots \geq \omega_n$ , seguindo a política  $\max \omega_i$ . Nesta configuração, quando entram no sistema, cada tarefa imediatamente seleciona sua melhor resposta.

**Lema 7.** *Em todo instante, a política de ativação  $\max \omega_i$  numa configuração inicial nula permanece em um equilíbrio de Nash.*

De forma análoga, pode-se incrementar políticas de modo a utilizá-las para descrever a forma como as máquinas processam as tarefas. Assim, por exemplo, uma possível política é fazer com que as máquinas processem a tarefa com maior peso primeiro. Até o momento, estivemos assumindo que a ordem de processamento não é relevante para o problema, mas esta seria uma possível generalização que claramente modificaria o comportamento dos agentes. Assim os processos com maior peso teriam prioridade nas máquinas mais rápidas.

### 4.2.2 Taxação

Como maneira de forçar o comportamento do sistema a convergir para uma solução mais próxima do ótimo, pode-se atribuir taxas nas estratégias dos agentes, como descrito por Caragiannis, Kaklamanis e Kanellopoulos [33]. Dado um jogo  $G = \langle M, s, N, \omega \rangle$ , uma taxa é uma função  $\delta : M \times \mathbb{Q}^+ \rightarrow \mathbb{Q}^+$  que atribui um valor  $\delta(j, w)$  para cada agente de peso  $w$  que deseja utilizar a máquina  $j \in M$ . Além disso, pode-se considerar que cada agente é sensível à taxa de uma maneira diferente, com base em um coeficiente de sensibilidade  $\gamma_i > 0$ . Assim, os agentes buscam minimizar o custo total  $C_i(j) + \gamma_i \delta(j, \omega_i)$ . Diz-se que uma taxa é  $\rho$ -eficiente se o custo social de qualquer equilíbrio puro de Nash é ao menos  $\rho$  vezes o custo social da alocação ótima. Caragiannis, Kaklamanis e Kanellopoulos [33] demonstraram a existência de uma taxa que é 1.618-eficiente.

Problemas de taxa de possuem grande interesse em economia e engenharia de transportes, que se utilizam majoritariamente de modelos de congestão não-atômica, em que a ação de um agente não influencia a performance do sistema como um todo [34].

## 4.3 Modelos Determinísticos

As próximas seções apresentam resultados elementares para os jogos de balanceamento de carga em que as estratégias são decididas de forma determinística.

### 4.3.1 Máquinas Idênticas

Consideremos o problema de balanceamento em que todas as máquinas possuem a mesma velocidade  $s_1 = \dots = s_m = 1$ . Note que  $\omega_{ij} = \omega_i$ . O preço de anarquia é então dado como a seguir.

**Teorema 18** ([14]). *Considere uma instância  $G = \langle M, s, N, \omega \rangle$  do jogo de balanceamento de carga com  $n$  tarefas com tamanhos  $\omega_1, \dots, \omega_n$  e  $m$  máquinas idênticas. Seja  $A : N \rightarrow M$  um*

mapa de alocações em equilíbrio de Nash. Então, vale que

$$\mathfrak{C}(A) \leq \left(2 - \frac{2}{m+1}\right) \text{opt}(G).$$

*Demonstração.* Analisaremos primeiro o que ocorre com os custos das máquinas no equilíbrio de Nash. Seja  $j^*$  a máquina com maior carga em um equilíbrio de Nash  $A$ , e  $i^*$  o processo com menor peso nesta máquina. Sem perda de generalidade, pode-se afirmar que existem ao menos dois processos em  $j^*$ , uma vez que se não se tivesse valeria que  $\mathfrak{C}(A) = \text{opt}(G)$ , o que está em conformidade com o enunciado acima. Assim,  $\omega_{i^*} \leq \frac{1}{2} \mathfrak{C}(A)$ , caso contrário  $i^*$  não seria o processo com menor peso. Note que  $j^*$  define o custo social, uma vez que é a máquina que possui maior carga.

Suponhamos que existe uma máquina  $j \neq j^*$  com carga menor que  $\ell_{j^*}(A) - \omega_{i^*}$ , isto é, com carga menor do que  $j^*$  retirando-se a contribuição do menor processo. Assim, poderíamos mover  $i^*$  de  $j^*$  para  $j$ , reduzindo portanto o custo do processo. Mas  $A$  é um equilíbrio de Nash, logo não pode existir tal máquina  $j$ .

Portanto, pode-se escrever

$$\ell_j(A) \geq \ell_{j^*}(A) - \omega_{i^*} \geq \mathfrak{C}(A) - \frac{1}{2}\mathfrak{C}(A) = \frac{1}{2}\mathfrak{C}(A).$$

Observe, agora, que o custo de uma alocação ótima não pode ser menor do que a média de todas as cargas nas máquinas. Assim, pelo resultado acima,

$$\text{opt}(G) \geq \sum_{i \in M} \frac{\ell_i(A)}{m} \geq \frac{\mathfrak{C}(A) + \frac{1}{2}\mathfrak{C}(A)(m-1)}{m}.$$

E, portanto,

$$\mathfrak{C}(A) \leq \left(2 - \frac{2}{m+1}\right) \text{opt}(G).$$

■

Deve-se notar que quando  $m \rightarrow \infty$ , o preço de anarquia converge para 2. E, no caso trivial em que  $m = 1$ , o custo social sempre é o melhor possível e portanto o preço de anarquia é 1. Assim, o  $PoA$  está sempre entre estes dois valores. Vamos mostrar, na Seção 4.3.3, que no problema mais genérico de máquinas uniformemente relacionadas, o preço de anarquia não mais é limitado.

### 4.3.2 Convergência

Além do critério de existência, é importante compreender quanto tempo os agentes levam para convergir para um equilíbrio de Nash. A política de maior peso nos mostra que existe uma condição em que o equilíbrio é atingido em até  $n$  passos, onde  $n$  é o número de agentes. Essa política ativa sequencialmente os agentes preferenciando os processos com maior peso entre aqueles que não estão em situações ótimas.

**Proposição 6.** *Para máquinas idênticas, um passo incremental sempre aumenta a menor carga entre as máquinas que compõe o sistema.*

*Demonstração.* Suponhamos que a proposição não fosse satisfeita. Ora, então existe um agente que em algum momento estava na máquina com menor carga e mudou sua alocação para outra máquina. Como já se estava na máquina de menor carga, esta nova máquina possui uma carga maior. Mas os agentes sempre visam minimizar seu custo, uma contradição. ■

**Proposição 7.** *Seja  $G = \langle M, s, N, \omega \rangle$  uma instância do jogo de balanceamento e  $A: N \rightarrow M$  uma alocação. Um dado agente  $i$  está satisfeito em  $A$  se e somente se a carga  $\ell_{A(i)}(A) - \omega_i$  é mínima.*

**Teorema 19.** *Seja  $G = \langle M, s, N, \omega \rangle$  uma instância do jogo de balanceamento e  $A: N \rightarrow M$  uma alocação. A partir de  $A$ , pode-se alcançar um equilíbrio de Nash com todos os agentes sendo ativados ao menos uma vez pela política de peso máximo.*

*Demonstração.* Suponhamos que, sempre que cada agente for ativado pela política de peso máximo, ele nunca mais fica insatisfeito. Isto é, a alocação final é ótima permanentemente. Se assim for o caso, o teorema é satisfeito. Como consequência das Proposições 6 e 7, a única forma de um agente ficar insatisfeito é um outro agente colocar seu processo na mesma máquina em que ele está alocado. Suponha que um agente  $k$ , ativado depois do agente  $i$ , aloca sua carga na mesma máquina de  $i$ , chamada de  $j^*$ . Como a alocação de  $k$  é a melhor possível e vale  $\omega_k \leq \omega_i$  pela política de peso máximo, tem-se que, para qualquer máquina  $j$ ,

$$\ell_{j^*}(A) \leq \ell_j(A) + \omega_k \leq \ell_j(A) + \omega_i.$$

Ou seja,

$$\ell_j(A) \geq \ell_{j^*}(A) - \omega_i,$$

e portanto o agente  $i$  permanece satisfeito, uma vez que nenhuma outra máquina possui carga que reduziria seu custo.

Portanto, em todo passo incremental dada a política de peso máximo os agentes nunca ficam insatisfeitos e assim se alcança o equilíbrio de Nash quando todos os agentes são ativados. ■

É importante notar que a ordem de ativação pode tornar a análise consideravelmente diferente, como descrito na Seção 4.2.1. Por exemplo, se a ativação for realizada nas tarefas com menor peso, pode-se ocorrer situações em que a convergência ocorre em tempo exponencial [35].

**Teorema 20** ([32]). *Existe uma instância do jogo de balanceamento de carga com  $n$  tarefas e  $m$  máquinas uniformes onde a política de resposta ótima das tarefas mais leves requer pelo menos  $(n/(m-1)^2)^{m-1}$  passos para atingir um equilíbrio de Nash.*

### 4.3.3 Máquinas Uniformemente Relacionadas

Consideremos agora o caso mais geral em que as máquinas possuem velocidades arbitrárias, e que o peso  $\omega_{ij}$  da tarefa  $i$  na máquina  $j$  é processado como  $\omega_i/s_j$ . Para o cálculo do preço de anarquia, as seguintes proposições serão úteis.

**Proposição 8.** *Vale que  $\frac{\log m}{\log \log m} > \sqrt{\log m}$ , para  $m > 1$  suficientemente grande.*

*Demonstração.* Dado  $m$  real positivo, basta fazer a transformação  $m \rightarrow e^{e^n}$ , para algum  $n$ , de onde se tem que  $e^n > n^2$ . ■

**Proposição 9.** *Sendo  $\Gamma$  a função Gamma definida como  $\Gamma(m) = (m-1)!$  para  $m$  inteiro positivo, vale que*

$$\Gamma^{-1}(m) = \Theta\left(\frac{\log m}{\log \log m}\right).$$

*Demonstração.* Seja  $\Gamma^{-1}(m) = k$ , de modo que  $k!$  é o maior fatorial menor ou igual a  $m$ . Vale assim que  $(\frac{k}{2})^{k/2} \leq k! \leq m$ . Portanto,  $\frac{k}{2} \log(\frac{k}{2}) \leq \log m$  e  $k \leq \frac{2 \log m}{\log k - 1} \leq \frac{4 \log m}{\log k}$  para  $k \geq 4$ . Logo,  $k$  é  $\mathcal{O}(\frac{\log m}{\log k})$ .

Também, tem-se que  $m \leq (k+1)! \leq (k+1)k^k$ . Assim, vale que  $\log m \leq \log(k+1) + k \log k \leq 2k \log k$ . Logo,  $k \geq \frac{\log m}{2 \log k}$  e portanto  $k$  é  $\Omega(\frac{\log m}{\log k})$ .

Como  $k \leq \frac{4 \log m}{\log k} \leq \frac{4 \log m}{\log(\frac{\log m}{2 \log k})}$ , tem-se que  $k \frac{4 \log m}{\log \log m - \log(2 \log k)} \leq c \cdot \frac{\log m}{\log \log m}$  para alguma constante  $c$ , uma vez que  $k$  é muito menor do que  $m$ . Portanto,  $k$  é  $\mathcal{O}(\frac{\log m}{\log \log m})$ . De forma análoga se mostra que  $k$  é  $\Omega(\frac{\log m}{\log \log m})$ , o que demonstra portanto a proposição. ■

**Teorema 21.** *Considere uma instância  $G = \langle M, s, N, \omega \rangle$  do jogo de balanceamento com  $n$  tarefas com pesos  $\omega_1, \dots, \omega_n$  e  $m$  máquinas com velocidades  $s_1, \dots, s_m$ . Seja  $A : N \rightarrow M$  um mapa de alocações em equilíbrio de Nash. Então, vale que*

$$\mathbf{c}(A) = \mathcal{O} \left( \frac{\log m}{\log \log m} \right) \mathbf{opt}(G).$$

*Demonstração.* Seja  $c = \lfloor \mathbf{c}(A) / \mathbf{opt}(G) \rfloor$ . Mostraremos que  $c \leq \Gamma^{-1}(m)$ , e então o resultado segue pela Proposição 9. Sem perda de generalidade, vamos assumir que  $s_1 \geq \dots \geq s_m$ , e  $L = (1, \dots, m)$  é a lista de máquinas em ordem não-crescente de velocidade. Para  $k \in \{0, \dots, c-1\}$ , seja  $L_k = (1, 2, \dots, p)$  a sequência das  $p$  primeiras máquinas em ordem de velocidade de tal modo que para todo  $q \leq p$  vale que  $\ell_q(A) \leq k \mathbf{opt}(G)$ . Mostraremos que vale a seguinte recorrência:

$$|L_k| \geq \begin{cases} (k+1)|L_{k+1}| & \text{se } 0 \leq k \leq c-2 \\ 1 & \text{se } k = c-1. \end{cases}$$

Resolvendo a recorrência, tem-se que  $L_0 \geq (c-1)! = \Gamma(c)$ . Observe que  $L_0 = L$  e, portanto  $|L_0| = m$ . Assim,  $m \geq \Gamma(c)$ , o que demonstra o teorema.

Dada uma alocação  $A$  em equilíbrio de Nash, mostraremos primeiro que  $|L_{c-1}| \geq 1$ . Suponhamos por absurdo que  $L_{c-1} = \emptyset$ . Então, a carga na máquina 1 é menor que  $(c-1)\mathbf{opt}(G)$ . Seja  $i$  uma tarefa na máquina de maior carga, que é de ao menos  $c\mathbf{opt}(G)$ . Movendo  $i$  para a máquina 1, valeria que

$$(c-1)\mathbf{opt}(G) + \frac{\omega_i}{s_1} \leq (c-1)\mathbf{opt}(G) + \mathbf{opt}(G) \leq c\mathbf{opt}(G),$$

em que vale  $\frac{\omega_i}{s_1} \leq \mathbf{opt}(G)$ , uma vez que  $\mathbf{opt}(G)$  é maior ou igual ao peso de qualquer tarefa individualmente. Assim, o agente  $i$  consegue unilateralmente reduzir seu custo, contradizendo a hipótese de que  $A$  está em equilíbrio. Uma vez que é garantida a existência de  $A$ ,  $|L_{c-1}| \geq 1$ .

Agora, mostraremos que  $|L_k| \geq (k+1)|L_{k+1}|$ . Seja  $A^*$  uma alocação ótima. A afirmação a seguir relaciona a alocação em equilíbrio de Nash  $A$  com a alocação ótima  $A^*$ .

**Afirmção 1.** *Suponha que  $i$  é uma tarefa com  $A(i) \in L_{k+1}$ . Então  $A^*(i) \in L_k$ .*

*Proof.* Se  $L \setminus L_k = \emptyset$ , a afirmação é satisfeita. Seja  $q$  a máquina mais rápida em  $L \setminus L_k$ . Por definição, a carga  $\ell_q(A)$  é menor que  $k\mathbf{opt}(G)$ . Como  $x := A(i) \in L_{k+1}$ ,  $\ell_x(A) \geq (k+1)\mathbf{opt}(G)$ . Vamos assumir, por contradição, que  $\frac{\omega_i}{s_q} < \mathbf{opt}(G)$ . Então, movendo a tarefa

$i$  para a máquina  $q$  iria reduzir o custo de  $i$  para

$$\ell_q(A) + \frac{\omega_i}{s_q} < k \text{opt}(G) + \text{opt}(G) \leq \ell_x(A),$$

o que contradiz a hipótese de que  $A$  é um equilíbrio de Nash. Assim, todo processo  $i$  com  $A(i) \in L_{k+1}$  satisfaz  $\omega_i > s_q \text{opt}(G)$ .

Seja  $j = A^*(i)$  e suponha, por contradição, que  $j \in L \setminus L_k$ . Então a carga em  $j$  seria ao menos

$$\frac{\omega_i}{s_j} > \frac{s_q \text{opt}(G)}{s_j} \geq \text{opt}(G),$$

uma vez que  $s_j \leq s_q$ . Isso contradiz o fato de que  $A^*$  é ótimo e, portanto,  $A^*(i) \in L_k$ .  $\square$

Por definição de  $L_{k+1}$ , a soma das cargas que  $A$  atribui para a máquina  $j \in L_{k+1}$  é de ao menos  $(k+1)s_j \text{opt}(G)$ . Logo, a soma total de peso nas máquinas em  $L_{k+1}$  é de ao menos  $\sum_{j \in L_{k+1}} (k+1)s_j \text{opt}(G)$ . Pela Afirmação 1, uma alocação ótima deve alocar todo este peso nas máquinas de  $L_k$  de tal modo que a carga em todas as máquinas é de no máximo  $\text{opt}(G)$ . Como consequência,

$$\sum_{j \in L_{k+1}} (k+1)s_j \text{opt}(G) \leq \sum_{j \in L_k} s_j \text{opt}(G).$$

Portanto,

$$\sum_{j \in L_{k+1}} k s_j \leq \sum_{j \in L_k \setminus L_{k+1}} s_j.$$

Seja  $s^*$  a velocidade da máquina mais lenta em  $L_{k+1}$ , isto é,  $s^* = s_{|L_{k+1}|}$ . Para todo  $j \in L_{k+1}$ ,  $s_j \geq s^*$  e, para todo  $j \in L_k \setminus L_{k+1}$ ,  $s_j \leq s^*$ . Assim, vale que

$$\sum_{j \in L_{k+1}} k s^* \leq \sum_{j \in L_k \setminus L_{k+1}} s^*,$$

o que implica que  $|L_{k+1}| \cdot k \leq |L_k \setminus L_{k+1}| = |L_k| - |L_{k+1}|$ . Assim,  $|L_k| \geq (k+1) \cdot |L_{k+1}|$ .  $\blacksquare$

O resultado a seguir mostra que o fator do *preço de anarquia* do Teorema 21 é justo.

**Teorema 22.** *Para todo  $m \in \mathbb{N}$ , existe uma instância  $G = \langle M, s, N, \omega \rangle$  do jogo de balanceamento de carga com  $m$  máquinas e  $n \leq m$  tarefas que possui um equilíbrio de Nash  $A$  com*

$$\mathbf{c}(A) = \Omega\left(\frac{\log m}{\log \log m}\right) \text{opt}(G).$$

*Demonstração.* Iremos construir uma instância do jogo de balanceamento de carga  $G$  com

um mapa  $A$  em equilíbrio de Nash satisfazendo

$$\mathfrak{C}(A) \geq \frac{1}{2}(\Gamma^{-1}(m) - 2 - o(1))\text{opt}(G),$$

e o restante seguirá da Proposição 9. Nossa construção se utiliza de  $q + 1$  grupos disjuntos de máquinas  $G_0, \dots, G_q$  com  $q = \lfloor \Gamma^{-1}(m/3) - 1 \rfloor \geq \Gamma^{-1} - 2 - o(1)$ . Para  $0 \leq k \leq q$ , o grupo  $G_k$  consiste de  $q!/k!$  máquinas de velocidade  $2^k$  cada em que são alocadas  $k$  tarefas com peso  $2^k$ . O número total de máquinas é dado por

$$\sum_{k=0}^q |G_k| = q! \sum_{k=0}^q \frac{1}{k!} \leq 3\Gamma(q+1) \leq m.$$

Uma vez que  $\sum_{k=0}^q \frac{1}{k!} \leq 3$ , e pela definição de  $q$ , temos  $3\Gamma(q+1) \leq m$ . Assim, pode haver máquinas que não pertencem a nenhum destes grupos. Vamos assumir que todas estas máquinas possuem velocidade 1 e  $A$  não aloca tarefas nelas.

Mostraremos agora que  $A$  é um equilíbrio de Nash. Um agente  $k$  com tarefa alocada no grupo  $G_k$  possui custo  $k$ . Não se pode reduzir seu custo movendo a tarefa para uma máquina do grupo  $G_j$  com  $j \geq k$  uma vez que estas máquinas possuem carga de pelo menos  $k$ , nem para  $j < k$ , uma vez que se assim o fosse seu custo seria

$$j + \frac{2^k}{2^j} = j + 2^{k-j} \geq j + (k - j + 1) = k + 1,$$

uma vez que  $2^t \geq t + 1$ , para todo  $t \geq 1$ . Logo,  $A$  é um equilíbrio de Nash.

Note que o custo social da alocação  $A$  é  $q$ . Mostraremos que  $\text{opt}(G) \leq 2$ . Para  $k \in \{0, \dots, q\}$ , as tarefas mapeadas por  $A$  no grupo  $G_k$  são agora alocadas em  $G_{k-1}$ . Observe que o número total de tarefas que  $A$  mapeia em  $G_k$  é

$$k \cdot |G_k| = k \cdot \frac{q!}{k!} = |G_{k-1}|.$$

Logo, se pode alocar as tarefas de tal modo que cada máquina em  $G_{k-1}$  recebe exatamente uma tarefa que  $A$  mapeia. Como estes processos possuem peso de  $2^k$  e a velocidade das máquinas é de  $2^{k-1}$ , a carga do sistema nesta alocação é de no máximo 2. Portanto, como  $\frac{1}{2}\text{opt}(G) \leq 1$ , tem-se que

$$\mathfrak{C}(A) = q \geq \frac{1}{2}(\Gamma^{-1}(m) - 2 - o(1))\text{opt}(G),$$

finalizando assim a demonstração. ■

## 4.4 Convergência

**Conjectura 2.** *Existe uma política de resposta ótima que sempre converge para um equilíbrio de Nash em um número polinomial de respostas de melhoria para o jogo de balanceamento de carga em máquinas uniformemente relacionadas.*

**Teorema 23.** *O algoritmo LPT produz uma atribuição em equilíbrio de Nash para máquinas uniformemente relacionadas.*

*Demonstração.* Sejam as tarefas numeradas de 1 a  $n$  por ordem de inserção. Seja  $t \in \{0, \dots, n\}$  o tempo denotando o instante em que a  $t$ -ésima tarefa foi inserida. Mostraremos por indução que o mapa de alocações parcial  $A^t$  computado pelo algoritmo LPT no tempo  $t$  é um equilíbrio de Nash.

Por hipótese, as tarefas  $\{1, \dots, t-1\}$  estão satisfeitas no tempo  $t-1$ , isto é, nenhum desses agentes consegue reduzir unilateralmente seus custos. Quando a tarefa  $t$  é inserida, ela pode ser mapeado em uma máquina  $j$  que já possui outras tarefas. Mostraremos que estas tarefas em  $A_j^{t-1}$  não ficam insatisfeitas pelo acréscimo de carga resultante por  $t$ .

Seja  $i < t$  uma tarefa em  $j$ . Como a tarefa  $t$  minimiza o custo do agente  $t$  na máquina  $j$  e, pela política de alocação do algoritmo,  $\omega_t \leq \omega_i$ , vale

$$\frac{\ell_j}{s_j} \leq \frac{\ell_{j^*} + \omega_t}{s_{j^*}} \leq \frac{\ell_{j^*} + \omega_i}{s_{j^*}},$$

para toda máquina  $j^* \in M$ . Logo, no instante  $t$  o agente  $i$  continua satisfeito na máquina  $j$ , uma vez que não consegue reduzir mais seu custo. ■

O algoritmo LPT não apenas produz um equilíbrio de Nash como também aproxima o *makespan* ótimo em uma razão de  $\frac{5}{3}$  para máquina uniformemente relacionadas [36] e  $\frac{4}{3} - \frac{1}{m}$  para máquinas idênticas [37].

**Teorema 24** ([38]). *Dado qualquer perfil estratégico puro, o algoritmo de Nashificação computa um equilíbrio de Nash sem aumentar o custo social performando no máximo  $(m+1)n$  movimentos em ordem  $\mathcal{O}(m^2n)$ .*

## 4.5 Modelos Aleatórios

Considerando o teorema de Nash, é relevante estudar o caso em que os agentes possuem estratégias mistas. No que segue, seja  $G = \langle M, s, N, \omega \rangle$  uma instância do jogo de balanceamento de carga,  $A : N \rightarrow M$  uma alocação e  $p_i^j = \mathbb{P}[A(i) = j]$  a probabilidade do agente  $i$  alocar sua tarefa na máquina  $j$ . Um perfil estratégico  $P = (p_i^j)_{i \in N, j \in M}$  especifica um mapa

aleatório entre as tarefas e as máquinas em todo o sistema. Seja  $x_i^j$  uma variável aleatória que toma o valor 1 caso  $A(i) = j$  e 0 caso contrário. Assim, a carga esperada da máquina  $j$  com perfil estratégico  $P$  é dada por

$$\mathbb{E}[\ell_j(A)] = \mathbb{E} \left[ \sum_{i \in N} \frac{\omega_i x_i^j}{s_j} \right] = \sum_{i \in N} \frac{\omega_i \mathbb{E}[x_i^j]}{s_j} = \sum_{i \in N} \frac{\omega_i p_i^j}{s_j}.$$

O custo social  $\mathfrak{C}(P)$  do perfil estratégico  $P$  é definido como o *makespan* máximo esperado da forma

$$\mathfrak{C}(P) = \mathbb{E}[\mathfrak{C}(A)] = \mathbb{E} \left[ \max_{j \in M} \ell_j(A) \right].$$

Assumimos, também, que todo agente visa minimizar seu custo. Do ponto de vista do agente  $i$ , o custo esperado na máquina  $j$  é dado por  $C_i(j) = \mathbb{E}[\ell_j(A) | A(i) = j]$ . Para qualquer  $P$ , vale que

$$C_i(j) = \frac{\sum_{k \neq i} \omega_k p_k^j + \omega_i}{s_j} = \mathbb{E}[\ell_j(A)] + (1 - p_i^j) \frac{\omega_i}{s_j}.$$

Os agentes se encontram em um equilíbrio misto de Nash se não há incentivo para individualmente mudarem sua probabilidade de alocação. Isto é, vale a seguinte proposição.

**Proposição 10.** *Um perfil estratégico  $P$  é um equilíbrio de Nash se, e somente se, para qualquer  $i \in N$  e  $j \in M$ , tem-se que não existe  $k \in M$  tal que  $C_i(j) \geq C_i(k)$ .*

#### 4.5.1 Máquinas Idênticas

No decorrer dessa seção, toma-se que as velocidades das máquinas são iguais  $s_1 = \dots = s_m = 1$ . O próximo resultado será útil para obtenção do preço de anarquia do sistema.

**Teorema 25** ([14]). *Considere  $G = \langle M, s, N, \omega \rangle$  uma instância do jogo de balanceamento. Para qualquer máquina  $j \in M$ , vale que a carga esperada  $\mathbb{E}[\ell_j(A)]$  possui limitante superior da forma*

$$\mathbb{E}[\ell_j(A)] \leq \left( 2 - \frac{2}{m+1} \right) \text{opt}(G).$$

*Demonstração.* A demonstração é semelhante à do Teorema 18. ■

**Teorema 26** ([14]). *Considere uma instância  $G = \langle M, s, N, \omega \rangle$  do jogo de balanceamento com  $n$  tarefas com pesos  $\omega_1, \dots, \omega_n$  e  $m$  máquinas idênticas. Sendo  $P$  um perfil estratégico qualquer em equilíbrio de Nash, vale que*

$$\mathfrak{C}(P) = \mathcal{O} \left( \frac{\log m}{\log \log m} \right) \text{opt}(G).$$

*Demonstração.* Fixemos  $j \in M$ , e seja  $\omega_Z$  o maior peso no sistema. Aplicando o Lema 9, vale que para todo  $t$

$$\mathbb{P}[\ell_j(A) \geq t] \leq \min \left\{ 1, \left( \frac{e\mathbb{E}[\ell_j(A)]}{t} \right)^{t/\omega_Z} \right\} \leq \left( \frac{2e\text{opt}(G)}{t} \right)^{t/\text{opt}(G)}.$$

Uma vez que  $\mathbb{E}[\ell_j(A)] \leq 2\text{opt}(G)$  e  $\omega_Z \leq \text{opt}(G)$ . Seja  $\tau = 2\frac{\log m}{\log \log m}\text{opt}(G)$ . Então, para todo  $x \geq 0$ ,

$$\begin{aligned} \mathbb{P}[\ell_j(A) \geq \tau + x] &\leq \left( \frac{e \log \log m}{\log m} \right)^{2 \log m / \log \log m + x / \text{opt}(G)} \\ &\leq \left( \frac{1}{\sqrt{\log m}} \right)^{2 \log m / \log \log m} e^{-x / \text{opt}(G)} \\ &= m^{-1} e^{-x / \text{opt}(G)}, \end{aligned} \tag{1}$$

onde se utilizou a Proposição 8 na primeira passagem. Uma vez que para uma variável aleatória  $X$  se tem  $\mathbb{E}[X] = \int_0^\infty \mathbb{P}[X \geq t] dt$ , então

$$\mathfrak{C}(P) = \mathbb{E} \left[ \max_{j \in M} \ell_j(A) \right] = \int_0^\infty \mathbb{P} \left[ \max_{j \in M} \ell_j(A) \geq t \right] dt.$$

Substituindo  $t$  por  $\tau + x$  e aplicando a desigualdade de Boole,

$$\mathfrak{C}(P) \leq \tau + \int_0^\infty \mathbb{P} \left[ \max_{j \in M} \ell_j(A) \geq \tau + x \right] dx \leq \tau + \int_0^\infty \sum_{j \in M} \mathbb{P}[\ell_j(A) \geq \tau + x] dx.$$

Por fim, pelo resultado na Equação 1,

$$\mathfrak{C}(P) \leq \tau + \int_0^\infty e^{-x/\text{opt}(G)} dx = \tau + \text{opt}(G) = \mathcal{O} \left( \frac{\log m}{\log \log m} \right) \text{opt}(G).$$

O preço de anarquia do sistema assim cresce com o número de máquinas pela expressão obtida. ■

#### 4.5.2 Máquinas Uniformemente Relacionadas

Estudaremos, por fim, o caso mais genérico dentro dos jogos de balanceamento abarcados.

**Teorema 27** ([14]). *Considere uma instância  $G$  do jogo de balanceamento com  $n$  tarefas*

com pesos  $\omega_1, \dots, \omega_n$  e  $m$  máquinas com velocidades  $s_1, \dots, s_n$ . Sendo  $P$  um perfil estratégico qualquer em equilíbrio de Nash, vale que

$$\mathfrak{C}(P) = \mathcal{O}\left(\frac{\log m}{\log \log \log m}\right) \text{opt}(G).$$

*Demonstração.* Primeiramente iremos encontrar um limitante superior para a maior carga esperada  $\max_{j \in M} \mathbb{E}[\ell_j(A)]$ , ao invés da carga esperada máxima  $\mathbb{E}[\max_{j \in M} \ell_j(A)]$ . Como simplificação da notação, façamos  $\text{opt}(G) = 1$ , o que é obtido reescalando os pesos apropriadamente. Seja  $c = \lfloor \max_{j \in M} \mathbb{E}[\ell_j(A)] \rfloor$ , e façamos  $s_1 \geq \dots \geq s_m$ . Seja  $L$  a lista de máquinas em ordem não-decrescente de velocidade. Para  $k \in \{0, \dots, c-1\}$ , seja  $L_k$  o prefixo de comprimento máximo tal que a carga esperada em cada máquina em  $L_k$  é de no máximo  $k$ . De forma análoga à demonstração do Teorema 21, mostra-se a recorrência  $|L_k| \geq (k+1) \cdot |L_{k+1}|$ , para  $0 \leq k \leq c-2$ , e  $|L_{c-1}| \geq 1$ . Resolvendo a recorrência,  $|L_0| \geq (c-1)! = \Gamma(c)$ . Assim,  $|L_0| = m$  implica  $c \leq \Gamma^{-1}(m) = \Theta(\ln m / \ln \ln m)$ . Seja

$$C = \max\left\{c+1, \frac{\ln m}{\ln \ln m}\right\} = \Theta\left(\frac{\ln m}{\ln \ln m}\right).$$

Mostraremos que o *makespan* esperado da alocação em equilíbrio pode exceder  $C$  por um fator de no máximo  $\ln \ln m / \ln \ln \ln m$  tal que o *makespan* esperado é  $\mathcal{O}(\ln m / \ln \ln \ln m)$ , o que prova o teorema, pois  $\text{opt}(G) = 1$ .

Para uma máquina  $j \in M$ , seja  $T_j^{(1)}$  o conjunto de tarefas  $i$  com  $p_i^j \geq 1/4$  e  $T_j^{(2)}$  o conjunto de todas as outras. Sejam  $\ell_j^{(1)}$  e  $\ell_j^{(2)}$  as cargas na máquina  $j$  devido aos processos em  $T_j^{(1)}$  e  $T_j^{(2)}$ , respectivamente. Note que  $\ell_j(A) = \ell_j^{(1)} + \ell_j^{(2)}$ . Para as tarefas em  $T_j^{(1)}$ , vale que

$$\ell_j^{(1)} = \sum_{i \in T_j^{(1)}} \frac{\omega_i}{s_j} \leq 4 \sum_{i \in T_j^{(1)}} \frac{\omega_i p_i^j}{s_j} = 4\mathbb{E}[\ell_j^{(1)}] \leq 4C.$$

Para encontrar um limitante superior para  $\ell_j^{(2)}$ , utilizaremos o limitante de Chernoff do Lema 9. Para  $0 \leq k \leq c-2$ , seja  $G_k = L_k \setminus L_{k+1}$  e  $G_{c-1} = L_{c-1}$ . Para  $0 \leq k \leq c-1$ , seja  $s(k)$  a velocidade da máquina mais rápida em  $G_k$ . Note que  $s(c-1) \geq s(c-2) \geq \dots \geq s(0)$ . Mostraremos que esta sequência decresce geometricamente.

**Afirmção 2.** Para  $0 \leq k \leq c-4$ ,  $s(k+2) \geq 2s(k)$ .

*Proof.* Observe que existe uma tarefa  $j^*$  com  $\omega_{j^*} \leq s(k+2)$  com probabilidade positiva de estar em uma máquina em  $L_{k+3}$ . Isto porque uma alocação ótima deve mover uma parte da carga esperada das máquinas em  $L_{k+3}$  para as máquinas em  $L \setminus L_{k+3}$  que possuem peso menor

que a maior velocidade entre o conjunto, que é  $s(k+2)$ . Agora, suponha que  $s(k) > s(k+2)/2$ . A carga esperada na máquina mais rápida em  $G_k$  é de no máximo  $k+1$  por definição. Então, o custo esperado de  $j^*$  na máquina mais rápida de  $G_k$  é de até

$$k + 1 + \frac{\omega_{j^*}}{s(k)} < k + 1 + \frac{2\omega_{j^*}}{s(k+2)} \leq k + 3,$$

o que contradiz que o custo esperado de  $j^*$  é de no mínimo  $k+3$  e possui probabilidade positiva em alguma máquina de  $L_{k+3}$ .  $\square$

Utilizaremos a Afirmação 2 para encontrar um limitante superior para os pesos das máquinas em  $T_j^{(2)}$ .

**Afirmação 3.** Para todo  $j \in M$  e  $i \in T_j^{(2)}$ ,  $\omega_i \leq 12s_j$ .

*Proof.* Seja  $i$  uma tarefa em  $T_j^{(2)}$  e  $j \in G_k$  para  $0 \leq k \leq c-1$ . O custo esperado de  $i$  em  $j$  é de

$$c_i^j = \mathbb{E}[\ell_j(A)] + (1 - p_i^j) \frac{\omega_i}{s_j} \geq k + \frac{3\omega_i}{4s_j}.$$

Suponha que  $k \geq c-3$ . Neste caso,  $\omega_i > 12s_j$  implica que  $c_i^j > k+9 \geq c+6$ , o que contradiz que em um equilíbrio de Nash o custo esperado de qualquer tarefa na máquina mais rápida é de  $c+1$ . Assim, a afirmação é válida para  $k \geq c-3$ . Agora suponha que  $k \leq c-4$ . Seja  $q$  a máquina mais rápida em  $G_{k+2}$ . Pela Afirmação 2 vale  $s_q = s(k+2) \geq 2s(k) \geq 2s_j$ . Assim, o custo esperado de  $i$  em  $q$  é de

$$c_i^q = \mathbb{E}[\ell_q(A)] + (1 - p_i^q) \frac{\omega_i}{s_q} \leq k + 3 + \frac{\omega_i}{2s_j}.$$

Como  $p_i^j > 0$ , pela condição de  $P$  estar em equilíbrio de Nash vale que  $c_i^j \leq c_i^q$ . Assim,

$$k + \frac{3\omega_i}{4s_j} \leq k + 3 + \frac{\omega_i}{2s_j},$$

o que implica que  $\omega_i \leq 12s_j$ , e portanto finalizando a demonstração.  $\square$

Seja  $z$  a maior carga em  $T_j^{(2)}$ , isto é,  $z = \max_{i \in T_j^{(2)}} \omega_i/s_j$ . A Afirmação 3 implica que  $z \leq 12$ . Agora, aplicando o Lema 9, vale que para todo  $\alpha > 0$ ,

$$\mathbb{P}[\ell_j^{(2)} \geq \alpha C] \leq \left( \frac{e \cdot \mathbb{E}[\ell_j^{(2)}]}{\alpha C} \right)^{\alpha C/z} \leq \left( \frac{e}{\alpha} \right)^{\alpha C/12},$$

uma vez que  $\mathbb{E}[\ell_j^{(2)}] \leq C$ . Seja  $\tau = 24C \ln \ln m / \ln \ln \ln m$ . Como  $C$  é de ordem  $\ln m / \ln \ln m$ , segue que  $\tau$  é de ordem  $\ln m / \ln \ln \ln m$ . Seja  $x \geq 0$ . Substituindo  $\alpha C$  por  $\tau + x$ , vale então

$$\begin{aligned} \mathbb{P}[\ell_j^{(2)} \geq \tau + x] &\leq \left( \frac{eC}{\tau + x} \right)^{(\tau+x)/12} \\ &\leq \left( \frac{e \ln \ln \ln m}{24 \ln \ln m} \right)^{2C \ln \ln m / \ln \ln \ln m + x/12}. \end{aligned}$$

Observe que  $24 \ln \ln m / (e \ln \ln \ln m)$  é limitado inferiormente por  $\sqrt{\ln \ln m}$  e também por  $e^2$ . Também,  $C \geq \ln m / \ln \ln m$ . Assim,

$$\mathbb{P}[\ell_j^{(2)} \geq \tau + x] \leq \left( \frac{1}{\sqrt{\ln \ln m}} \right)^{2 \ln m / \ln \ln \ln m} \cdot e^{-x/6} = m^{-1} \cdot e^{-x/6}.$$

Conseqüentemente,

$$\begin{aligned} \mathbb{E} \left[ \max_{j \in M} \ell_j^{(2)} \right] &= \int_0^\infty \mathbb{P} \left[ \max_{j \in M} \ell_j^{(2)} \geq t \right] dt \\ &\leq \tau + \int_0^\infty \mathbb{P} \left[ \max_{j \in M} \ell_j^{(2)} \geq \tau + x \right] dx \\ &\leq \tau + \int_0^\infty \sum_{j \in M} \mathbb{P}[\ell_j^{(2)} \geq \tau + x] dx. \end{aligned}$$

Utilizando o limitante superior,

$$\mathbb{E} \left[ \max_{j \in M} \ell_j^{(2)} \right] \leq \tau + \int_0^\infty e^{-x/6} dx = \tau + 6.$$

Finalmente, combinando as expressões, tem-se

$$c(P) = \mathbb{E} \left[ \max_{j \in M} \ell_j^{(2)} \right] \leq 4C + \tau + 6 = \mathcal{O} \left( \frac{\log m}{\log \log \log m} \right).$$

■

O resultado a seguir mostra que o anterior é justo.

**Teorema 28** ([14]). *Para todo  $m \in \mathbb{N}$ , existe uma instância  $G$  do jogo de balanceamento de carga com  $m$  máquinas e  $n \leq m$  tarefas que possui um perfil estratégico em equilíbrio de*

Nash  $P$  com

$$c(P) = \Omega\left(\frac{\log m}{\log \log \log m}\right) \text{opt}(G).$$

*Demonstração.* Construiremos uma instância que é continuidade da instância do Teorema 22. Utilizaremos estratégias mistas apenas nos grupos  $G_k$  com  $k = \lfloor q/2 \rfloor$ . Seja  $M = |G_k| = q!/k! \geq (q/2)^{\lfloor q/2 \rfloor}$ . Observe que  $\log M = \Theta(q \log q) = \Theta(\log m)$ .

Seja  $T$  o número de tarefas mapeadas por  $A$  para uma das máquinas em  $G_k$ , todas com peso  $2^k$ . Cada uma destas tarefas é atribuída aleatoriamente para uma das máquinas em  $G_k$ , com  $p_i^j = 1/M$ . Para todas as outras tarefas em  $P$  se permanece com as alocações de  $A$ . O custo esperado de uma tarefa  $i \in T$  em uma máquina  $j \in G_k$  é, então,

$$c_i^j = \mathbb{E}[\ell_j(A)] + (1 - p_i^j) \frac{\omega_i}{s_j} = k + \left(1 - \frac{1}{m}\right) < k + 1.$$

Na demonstração do Teorema 22, mostramos que o custo da tarefa  $i$  de peso  $2^k$  em uma máquina do grupo  $G_j$ , com  $j \neq k$ , é de pelo menos  $k + 1$ . Logo,  $P$  é um equilíbrio de Nash, uma vez que os agentes em  $G_k$  encontram-se satisfeitos.

As modificações não afetam o custo social  $\text{opt}(G) = 2$ . Mostraremos um limitante inferior de  $c(P)$ . Este custo é limitado inferiormente pelo maior número de tarefas em uma nova máquina em  $G_k$ . Aplicando o Teorema 32 com  $m$  caixas e  $n = km$  bolas vale então que o *makespan* esperado é

$$\Theta\left(\frac{\ln m}{\ln(1 + m/n \log m)}\right) = \Omega\left(\frac{\log m}{\log \log \log m}\right),$$

uma vez que  $k = \Theta(\log m / \log \log m)$  e  $\log M = \Theta(\log m)$ . ■

**Teorema 29** ([39]). *Considere uma instância  $G = \langle M, s, N, \omega \rangle$  do jogo de balanceamento de carga com  $n$  tarefas com pesos  $\omega_1, \dots, \omega_n$  em que se tem  $S = \sum_{k=1}^n \omega_k > (n-1)\omega_i, \forall i \in N$ , e  $m$  máquinas com velocidades  $s_1 = \dots = s_{m-1} = 1$  e  $s_m = s > 0$ . Então existe um único equilíbrio misto completo de Nash se*

$$s \in \left( \frac{(m-1)S - (m-1)(n-1) \min_i \omega_i}{(m-1)S + (2-m)(n-1) \max_i \omega_i}, \frac{S}{S + (1-n) \min_i \omega_i} \right).$$

## 5 Conclusão e Perspectivas

“The Internet is an equilibrium, we just have to identify the game.”

---

*Scott Shenker*

Os problemas de balanceamento de carga e o de escalonamento de tarefas são clássicos em alocação de recursos, não apenas dada sua simplicidade mas também seu potencial de ser generalizado para problemas mais complexos. Sua variante descentralizada é interessante por diversas razões apresentadas no decorrer do texto, embora sejam tradicionalmente tratados como problemas de otimização.

Os modelos apresentados de jogos de balanceamento de carga podem ser vistos como uma descrição grosseira dos problemas que surgiram para entender a Internet. Os resultados são positivos, demonstrando eficiência sub-ótima, embora de certa forma paradoxal. Este caráter se torna mais complexo à medida que o número de agentes interagem na rede.

Os conceitos de Teoria dos Jogos são interessantes por seu potencial em descrever interações entre agentes num contexto geral. Pode-se ver na crise do SARS-CoV-2 resultados de seus conceitos, no conflito entre posicionamentos nas intervenções não-farmacêuticas (NPIs) e seus efeitos globais na transmissão do vírus.

A Teoria Algorítmica dos Jogos é uma área ainda em crescimento e longe de estar completa, tendo grandes perspectivas na utilização de simulações computacionais para descrever situações mais realistas.

## Referências

- [1] Don Ross. Game theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2019 edition, 2019.
- [2] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton paperbacks. Princeton University Press, 1980.
- [3] Maurice Fréchet. Commentary on the three notes of emile borel. *Econometrica*, 21(1):118–124, 1953.
- [4] Omar Raoof and Hamed Al-raweshidy. *Theory of Games: an Introduction*, chapter 1, pages 1–10. IntechOpen, 09 2010.
- [5] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. On a paradox of traffic planning. *Transportation science*, 39(4):446–450, 2005.
- [6] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 129–140, New York, NY, USA, 1999. Association for Computing Machinery.
- [7] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413. Springer, 1999.
- [8] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259, 2002.
- [9] Yuri Gurevich. The sequential asm thesis. *Current Trends in Theoretical Computer Science: Entering the 21st Century*, page 363, 2001.
- [10] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
- [11] Tim Roughgarden. An algorithmic game theory primer. In *Proceedings of the 5th IFIP International Conference on Theoretical Computer Science (TCS). An invited survey*, 2008.
- [12] Christos Papadimitriou and Georgios Piliouras. From nash equilibria to chain recurrent sets: Solution concepts and topology. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 227–235, 2016.
- [13] Jason R Marden and Jeff S Shamma. Game theory and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:105–134, 2018.

- [14] Eva Tardos Noam Nisan, Tim Roughgarden and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [15] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. *Games and Economic Behavior*, 65(2):289–317, 2009.
- [16] Simon Fischer and Berthold Vöcking. On the structure and complexity of worst-case equilibria. *Theoretical Computer Science*, 378(2):165–174, 2007.
- [17] Sehie Park. A history of the nash equilibrium theorem in the fixed point theory. *RIMS Kôkyûroku Bessatsu*, 2011.
- [18] Brian Swenson, Ryan Murray, and Soumya Kar. On best-response dynamics in potential games, 2017.
- [19] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.
- [20] Albert Xin Jiang and Kevin Leyton-Brown. A tutorial on the proof of the existence of nash equilibria. *University of British Columbia Technical Report TR-2007-25*, 2009.
- [21] I. L. Glicksberg. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.
- [22] Michal Feldman, Yuval Snappir, and Tami Tamir. The efficiency of best-response dynamics. In *International Symposium on Algorithmic Game Theory*, pages 186–198. Springer, 2017.
- [23] Robert W Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
- [24] Ron Holzman and Nissan Law-Yone. Strong equilibrium in congestion games. *Games and economic behavior*, 21(1-2):85–101, 1997.
- [25] Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.
- [26] Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure nash equilibria in concurrent deterministic games. *arXiv preprint arXiv:1503.06826*, 2015.
- [27] Cong Chen and Yinfeng Xu. Selfish load balancing for jobs with favorite machines. *Operations Research Letters*, 47(1):7–11, 2019.

- [28] Qingyu Li, Panlong Yang, Shaojie Tang, Chaocan Xiang, and Fan Li. Many is better than all: Efficient selfish load balancing in mobile crowdsourcing systems. In *2015 Third International Conference on Advanced Cloud and Big Data*, pages 1–6. IEEE, 2015.
- [29] Bibhudatta Sahoo. *Dynamic load balancing strategies in heterogeneous distributed system*. PhD thesis, National Institute of Technology, Rourkela, 2013.
- [30] Elias Koutsoupias, Panagiota N Panagopoulou, and Paul G Spirakis. Selfish load balancing under partial knowledge. In *International Symposium on Mathematical Foundations of Computer Science*, pages 609–620. Springer, 2007.
- [31] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul W Goldberg, Zengjian Hu, and Russell Martin. Distributed selfish load balancing. *SIAM Journal on Computing*, 37(4):1163–1181, 2007.
- [32] F. K. Miyazawa R. C. S. Schouery, O. Lee and E. C. Xavier. *Tópicos da Teoria dos Jogos em Computação*. Instituto de Matemática Pura e Aplicada, 2015.
- [33] Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Improving the efficiency of load balancing games through taxes. In *International Workshop on Internet and Network Economics*, pages 374–385. Springer, 2008.
- [34] He Zhang, Yuelong Su, Lihui Peng, and Danya Yao. A review of game theory applications in transportation analysis. *Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application*, 12 2010.
- [35] Eyal Even-Dar, Alex Kesselman, and Yishay Mansour. Convergence time to nash equilibria. In *International Colloquium on Automata, Languages, and Programming*, pages 502–513. Springer, 2003.
- [36] Donald K Friesen. Tighter bounds for lpt scheduling on uniform processors. *SIAM Journal on Computing*, 16(3):554–560, 1987.
- [37] Ronald L Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45(9):1563–1581, 1966.
- [38] Rainer Feldmann, Martin Gairing, Thomas Lücking, Burkhard Monien, and Manuel Rode. Nashification and the coordination ratio for a selfish routing game. In *International Colloquium on Automata, Languages, and Programming*, pages 514–526. Springer, 2003.

- [39] Cuixia Miao and Yuzhong Zhang. Fully mixed nash equilibria for the load balancing games on uniform parallel machines. In *International Mathematical Forum*, volume 5, pages 1607–1612. Citeseer, 2010.
- [40] Mark Burgin. Interpretations of negative probabilities. *arXiv preprint arXiv:1008.1287*, 2010.
- [41] Michael Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

# A Probabilidade

**Definição 31.** Um espaço de probabilidade possui três componentes.

1. Um conjunto universo  $\Omega$  de todas as possíveis ocorrências de eventos;
2. Uma família de conjuntos  $\mathcal{F}$  representando os eventos factíveis, onde  $\mathcal{F} \subset \Omega$ ;
3. Uma função de probabilidade  $\mathbb{P} : \mathcal{F} \rightarrow \mathbb{R}$ , satisfazendo a Definição 32.

**Definição 32.** Uma função de probabilidade é qualquer função  $\mathbb{P} : \mathcal{F} \rightarrow \mathbb{R}$  que satisfaz as seguintes condições:

1. Para qualquer evento  $A$ ,  $0 \leq \mathbb{P}[A] \leq 1$ ;
2.  $\mathbb{P}[\Omega] = 1$ ;
3. Para quaisquer sequência de conjuntos contáveis de eventos mutuamente disjuntos  $A_1, A_2, \dots, A_n$ ,

$$\mathbb{P} \left[ \bigcup_{i=1}^n A_i \right] = \sum_{i=1}^n \mathbb{P}[A_i].$$

Embora sempre se definam probabilidades não-negativas, estas podem vir a surgir de forma utilitária em certas situações [40].

**Proposição 11.** A ocorrência de um evento de probabilidade zero em um espaço contínuo não é impossível.

**Definição 33.** A esperança  $\mathbb{E}[X]$  de uma variável aleatória contínua  $X$  é dada por

$$\mathbb{E}[X] = \int_{\Omega} \tau \mathbb{P}[\tau] d\tau.$$

**Proposição 12.** O operador de esperança  $\mathbb{E}[\cdot]$  é linear, isto é, para quaisquer variáveis aleatórias  $X$  e  $Y$  e uma constante  $a$  qualquer vale que  $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$  e  $\mathbb{E}[aX] = a\mathbb{E}[X]$ .

**Lema 8.** Seja  $X$  uma variável aleatória contínua. Então

$$\mathbb{E}[X] = \int_{\Omega} \mathbb{P}[X \geq \tau] d\tau.$$

**Teorema 30** (Desigualdade de Markov). *Seja  $X : S \rightarrow \mathbb{R}$  uma variável aleatória contínua não-negativa. Então, para qualquer  $a > 0$ ,*

$$\mathbb{P}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

**Teorema 31** (Desigualdade de Boole). *Para um conjunto contável de eventos  $A_1, \dots, A_n$  vale que*

$$\mathbb{P}\left(\bigcup_{i=1}^n A_i\right) \leq \sum_{i=1}^n \mathbb{P}(A_i).$$

*Em particular,  $\mathbb{P}(A_k) \leq \sum_{i=1}^n \mathbb{P}(A_i)$  para qualquer  $k$ .*

**Lema 9** (Limitante de Chernoff). *Seja  $X_1, \dots, X_n$  variáveis aleatórias independentes com valor entre  $[0, z]$  para algum  $z$  positivo, e seja  $X = \sum_i X_i$ . Então vale que*

$$\mathbb{P}[X \geq t] \leq \left(\frac{e\mathbb{E}[X]}{t}\right)^{t/z}.$$

**Teorema 32** ([41]). *Suponha que  $n$  bolas são colocadas independentemente e uniformemente de modo aleatório em  $m$  caixas, e seja  $|M_i|$  o número de bolas na caixa  $i$ . Então a ocupância máxima esperada em uma caixa é*

$$\mathbb{E}\left[\max_{i \in M} |M_i|\right] = \Theta\left(\frac{\ln m}{\ln \ln m}\right).$$