

Otimização da distância mínima entre mesas e pessoas em estabelecimentos do ramo da alimentação

Luciana Y. Tsuchiya¹, Emerson V. Castelani², Francisco N. Calmon Sobral²

¹Instituto Federal do Paraná (IFPR)
CEP 87703-536 – Paranavaí – PR – Brazil

²Departamento de Matemática - Universidade Estadual de Maringá (UEM)
CEP 87020-900 – Maringá – PR – Brazil

luciana.tsuchiya@ifpr.edu.br, evcastelani@uem.br, fncsobral@uem.br

Abstract. *As one of the sectors most affected by the crisis caused by the coronavirus, the food business establishments will have to adapt to strict rules to promote the gradual return of its activities. Among some precautions, it is recommended that food establishments reorganize their space, ensuring a safe distance between people. In this work, we solved the problem of maximizing the minimum distance between tables and people in an establishment defined by a convex region. The problem was modeled as a nonlinear programming problem and implemented in the Julia programming language. Finally, a user interface was created.*

Resumo. *Como um dos setores mais afetados pela crise causada pelo corona vírus, o ramo de alimentação terá que se adaptar a rígidas regras para promover o retorno gradual de suas atividades. Dentre várias medidas, recomenda-se que os estabelecimentos deste ramo reorganizem seus espaços, de modo que seja possível uma distância segura entre as pessoas. Neste trabalho, resolvemos o problema de maximizar a distância mínima entre mesas e pessoas em um estabelecimento definido por uma região convexa. O problema foi modelado como um problema de programação não-linear e implementado na linguagem de programação Julia. Por fim, foi criada uma interface do usuário para que donos de estabelecimentos do ramo possam manipular.*

1. Introdução

O ano de 2020 trouxe um complicado cenário mundial. A pandemia da covid-19, causada pelo novo corona vírus, impôs o isolamento social que alterou por completo a dinâmica de vida nas cidades. As escolas pararam de funcionar de forma presencial, diversos trabalhos passaram a ser executados de forma remota e a economia praticamente parou. No que diz respeito a serviços de alimentação, inicialmente, a pandemia provocou a suspensão do serviço de alimentação fora do lar, fazendo com que muitos empreendimentos fechassem temporariamente suas portas ou que tivessem que se reinventar buscando novas formas de atendimento.

Com o passar dos dias, mesmo sem sinais de que a pandemia estava controlada, os governos passaram a flexibilizar as medidas restritivas impostas para conter a transmissão do corona vírus. Os governos dos 26 estados brasileiros e do Distrito Federal criaram

planos que definiam a retomada gradativa das atividades econômicas. Na maioria dos estados, esses planos resultaram em reabertura de diversos setores, mas com especial atenção aos cuidados para enfrentamento da doença. Passou-se a discutir então, diversas regras e medidas que pudessem garantir a reabertura dos setores com segurança.

Como um dos setores mais afetados pela crise da covid-19, o ramo de alimentação terá que se adaptar a regras rígidas para promover o retorno gradual de suas atividades. O objetivo é garantir que a população possa voltar a frequentar bares, lanchonetes, restaurantes e toda a cadeia da alimentação com condições higiênico-sanitárias seguras e distanciamento entre as pessoas.

Fundamentada em orientações da Organização Mundial da Saúde [OMS and FAO 2020], o Sebrae publicou uma cartilha com orientações para a retomada segura das atividades em bares, lanchonetes e restaurantes [Sebrae 2020]. Dentre várias medidas contidas na cartilha, uma recomenda que estes estabelecimentos reorganizem seus espaços, de modo que seja possível uma separação mínima de um metro entre as cadeiras (de clientes desconhecidos) e dois metros entre as mesas do salão. O Sebrae ainda ressalta que é fundamental levar em consideração as decisões da autoridade pública local que regulamentam a reabertura em cada estado, região e município. E que caso haja divergências, omissões ou imprecisões nas regras editadas para o território em que o empreendimento atue, que se opte sempre por seguir a regra mais rigorosa.

Para donos de bares, lanchonetes e restaurantes surge então um questão. Como posicionar as mesas no salão de forma a garantir a distância segura recomendada entre as pessoas?

Essa questão pode ser vista e resolvida como um problema de otimização. Mais especificamente, podemos pensar no seguinte problema:

Dado um estabelecimento definido por uma região convexa e dado um número N de mesas permitidas, encontrar a posição destes objetos de forma a maximizar a menor distância entre eles.

Esse problema foi proposto no curso *Otimização em ação!* ministrado pelo Departamento de Matemática da Universidade Estadual de Maringá. Apresentaremos aqui, uma modelagem matemática para este problema, falaremos brevemente da linguagem de programação na qual o modelo foi implementado, o *solver* utilizado para resolvê-lo e apresentaremos uma interface para o usuário (dono do estabelecimento) utilizar.

2. Problemas de Otimização

A otimização é uma área da matemática aplicada onde se desenvolvem métodos que têm por objetivo selecionar a melhor opção entre um conjunto de alternativas disponíveis, respeitando-se alguns critérios predefinidos. Mais especificamente, dado um certo problema que permita algum tipo de melhoramento ou que envolva uma tomada de decisão, a otimização está interessada em fornecer a resposta que irá gerar este melhoramento ou que auxiliará a tomada de decisão de forma ótima.

Problemas de otimização buscam maximizar ou minimizar uma função definida sobre um determinado domínio, sujeita a certas restrições. Em geral, demandam métodos numéricos que são estratégias implementáveis no computador utilizando alguma linguagem de programação escolhida.

De modo geral, um problema de otimização [Luenberger and Ye 1984] pode ser definido como :

$$\begin{aligned} &\text{Minimizar} && f(x) \\ &\text{sujeito a} && h_i(x) = 0, \quad i = 1, \dots, m \\ & && g_j(x) \leq 0, \quad j = 1, \dots, r \\ & && x \in S \end{aligned}$$

onde, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é chamada *função objetivo*, $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma restrição de desigualdade para cada $j = 1, \dots, r$ e $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ é uma restrição de igualdade para cada $i = 1, \dots, m$. S também é um conjunto de restrição e é arbitrário. Geralmente, as restrições impostas por S são restrições sobre o tipo de variável x .

O problema que nos propomos a resolver será modelado como um problema de otimização, mais precisamente, como um problema de otimização não linear contínuo, onde as funções envolvidas são diferenciáveis e algumas delas são não lineares.

3. A Modelagem do Problema

3.1. Dados do Problema

Para a modelagem matemática do problema foram feitas algumas considerações.

1. Para simplificar, consideramos N mesas redondas de raio igual a R metros, com quatro cadeiras posicionadas em torno dela a uma mesma distância uma da outra, formando um quadrado. Veja a Figura 1.

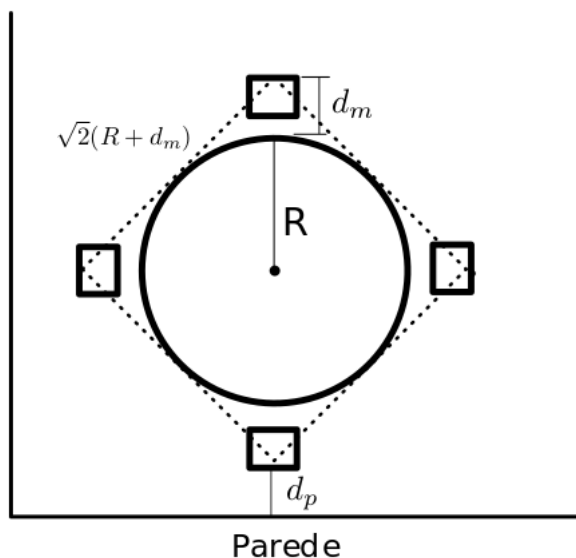


Figura 1. Configuração das mesas

2. O encosto de cada cadeira foi posicionado à uma distância d_m igual a 0,40 metros da borda da mesa, de forma que uma pessoa tenha espaço para sentar entre a mesa e o encosto. Assim, a distância de uma cadeira a outra cadeira oposta é de $2(R + d_m)$ metros e a distância entre cadeiras consecutivas é de $\sqrt{2}(R + d_m)$, conforme a Figura 1.

3. A distância entre as costas do encosto de uma cadeira e as paredes do estabelecimento deve ser no mínimo igual a $d_p = 0,30$ metros, para que seja possível uma pessoa empurrar a cadeira para trás e sair da mesa.
4. O salão do estabelecimento onde as mesas serão posicionadas, é uma região convexa poligonal de M lados. Matematicamente, isto significa que dados quaisquer dois pontos na região convexa, o segmento formado por eles está inteiramente contido na região.
5. A distância mínima entre as cadeiras de mesas distintas será denotada por D e o objetivo é maximizar ela. D deve ser no mínimo igual a $D_{min} = 1$ metro, conforme recomenda o Sebrae.

Observe que este problema poderia ser modelado de duas formas. Maximizar a distância mínima entre as mesas, isto é, entre os centros dos círculos que representam as mesas ou maximizar a distância mínima entre as cadeiras de mesas distintas, ou seja, entre as pessoas de mesas distintas.

De acordo com o item 5 acima, optamos pela segunda forma, embora o problema se torne mais complexo, pois uma vez que a distância mínima entre as cadeiras é maximizada, a distância entre as mesas também será maximizada.

Além disso, a maximização entre as cadeiras permite soluções em que as cadeiras de uma mesa estejam rotacionadas em relação a disposição das cadeiras de outra mesa, podendo aumentar ainda mais a distância entre as cadeiras, ou permitindo mais mesas no estabelecimento de forma a ainda manter a distância segura entre as pessoas.

Na Figura 2, por exemplo, as mesas A e B e as mesas C e D estão à mesma distância uma da outra. As mesas A e B possuem o mesmo padrão de distribuição das cadeiras ao redor delas. Já a mesa D, em relação à mesa C, está com suas cadeiras rotacionadas. Veja que ao rotacionar as cadeiras foi possível aumentar a distância entre as cadeiras mais próximas de mesas distintas.

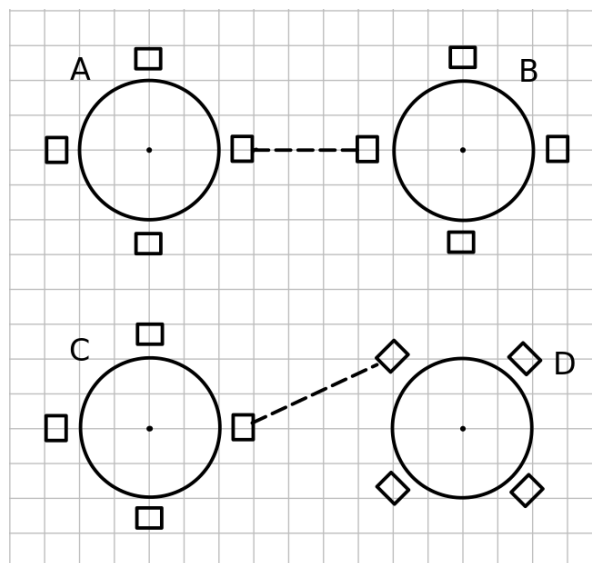


Figura 2. Comparação da distância entre as cadeiras com distanciamento igual entre centros das mesas

Por outro lado, na Figura 3, as distâncias entre as cadeiras mais próximas das

mesas A e B e das cadeiras mais próximas das mesas C e D são iguais a d , mas ao permitir a rotação da disposição das cadeiras na mesa D, foi possível aproximar as mesas, otimizando assim o espaço do salão, que poderá comportar mais mesas, ainda mantendo uma distância desejada entre as pessoas.

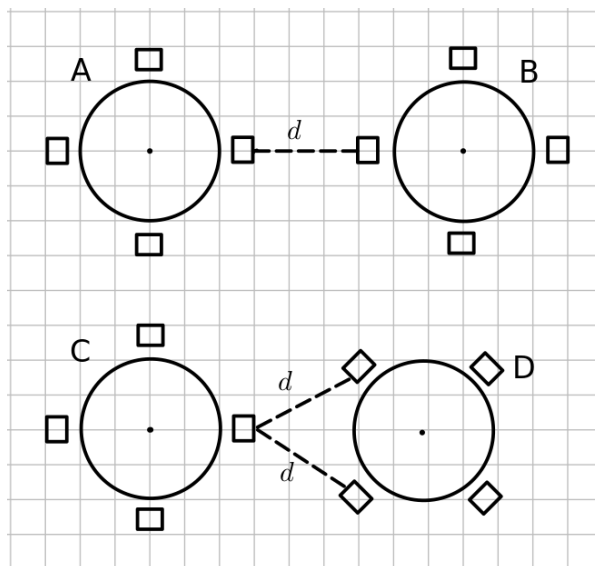


Figura 3. Comparação da distância entre os centros das mesa com distanciamento igual entre cadeiras próximas

3.2. O modelo matemático

Para o modelo matemático consideramos as seguintes questões:

1. *Posição das cadeiras:* independente de onde uma mesa esteja alocada, suas cadeiras devem estar em torno dela a uma distância $d_m = 0,40$ de sua borda, de maneira que elas formem um quadrado (Veja a Figura 1). Os centros das mesas serão denotados por C_i , $i = 1, \dots, N$. Os vértices (cadeiras) que formam o quadrado serão denotado por X_{ij} , com $i = 1, \dots, N$ e $j = 1, 2, 3, 4$. Assim, X_{ij} representa a cadeira j da mesa i . Temos então as equações.

$$\|X_{i1} - C_i\|^2 = (R + d_m)^2, \quad i = 1, \dots, N \quad (1)$$

$$\|X_{i2} - C_i\|^2 = (R + d_m)^2, \quad i = 1, \dots, N \quad (2)$$

$$\|X_{i3} - C_i\|^2 = (R + d_m)^2, \quad i = 1, \dots, N \quad (3)$$

$$\|X_{i4} - C_i\|^2 = (R + d_m)^2, \quad i = 1, \dots, N \quad (4)$$

$$\frac{X_{i1} + X_{i3}}{2} = C_i, \quad \frac{X_{i2} + X_{i4}}{2} = C_i, \quad i = 1, \dots, N \quad (5)$$

$$\|X_{i1} - X_{i2}\|^2 = 2 \cdot (R + d_m)^2, \quad i = 1, \dots, N \quad (6)$$

As equações (1), (2), (3) e (4) impõem que os vértices (cadeiras) do quadrado estejam a uma distância de 0,40 da borda da respectiva mesa.

As equações (5) asseguram que os vértices X_{i1} e X_{i2} serão opostos em relação ao centro da mesa, aos vértices X_{i3} e X_{i4} , respectivamente.

As equações (6) garantem que os vértices X_{i1} e X_{i2} não sejam coincidentes e estejam distantes entre si a uma distância correspondente ao lado de um quadrado de diagonal $2(R + d_m)$. Consequentemente, por causa das equações (5), os outros vértices também não serão coincidentes e estarão, consecutivamente, a uma mesma distância um do outro, formando assim um quadrado.

2. *Distância mínima entre as pessoas:* impomos que a distância entre cadeiras de mesas diferentes sejam maior ou igual a D_{min} por meio das equações (7) e (8) abaixo:

$$\|X_{ik} - X_{jl}\|^2 \geq D^2 \quad (7)$$

$$i \neq j, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad k = 1, \dots, 4, \quad l = 1, \dots, 4$$

$$D \geq D_{min} \quad (8)$$

3. *Alocando as mesas dentro do estabelecimento:* a região convexa de M lados que representa o salão do estabelecimento onde as mesas serão alocadas é formado pela intersecção de M semiplanos. Veja a Figura 4.

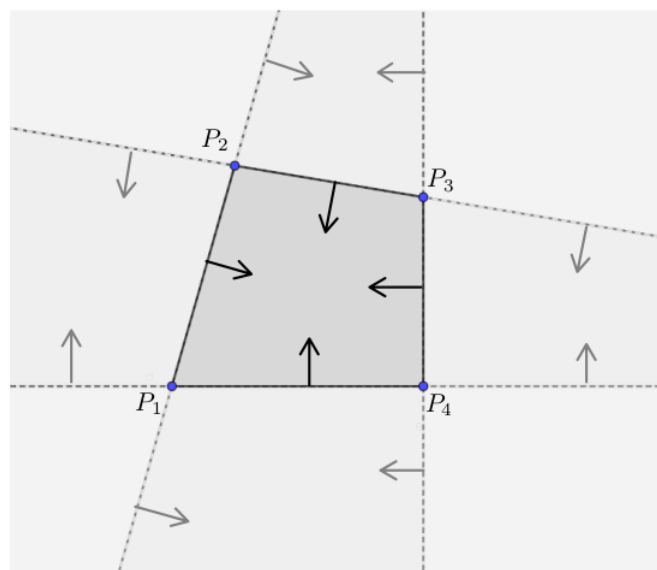


Figura 4. Região Convexa

Dados P_i e P_{i+1} dois vértices consecutivos da região convexa é possível determinar a reta $a_i x + b_i y + c_i = 0$ que passa por P_i e P_{i+1} de forma que seu vetor normal (a_i, b_i) esteja apontando para dentro da região convexa.

Considerando que a distância do centro $C_j = (C_{j1}, C_{j2})$ da mesa j ao encosto da cadeira é $(R + d_m)$ e a distância entre o encosto da cadeira e a parede deve ser no mínimo d_p , o centro da mesa deve estar a uma distância mínima de $(R + d_m + d_p)$ do lado da região convexa formado por P_i e P_{i+1} . Para que isso ocorra, “somamos”

às coordenadas de C_i , as respectivas coordenadas do vetor $(R + d_m + d_p) \frac{(a_i, b_i)}{\|(a_i, b_i)\|}$.

Assim, para assegurar que as mesas e as cadeiras serão alocadas dentro do estabelecimento, isto é, na intersecção dos semiplanos que formam a região convexa, impomos as seguintes restrições:

$$a_i \left(C_{j1} + \frac{a_i(R + d_m + d_p)}{\|(a_i, b_i)\|} \right) + b_i \left(C_{j2} + \frac{b_i(R + d_m + d_p)}{\|(a_i, b_i)\|} \right) + c_i \geq 0, \quad (9)$$

$$i = 1, \dots, M \text{ e } j = 1, \dots, N$$

Observe que optamos por considerar a distância dos centros das mesas às paredes, ao invés da distância das cadeiras às paredes, pois neste último caso, aumentaríamos em 4 vezes as restrições das equações 9, tornando o problema ainda mais complexo.

4. *Evitando que as mesas se sobreponham:* para evitar a sobreposição de mesas na solução do problema impomos a seguinte restrição:

$$\|C_i - C_j\|^2 \geq (2R + 2d_m)^2, \quad \text{para } i = 1, \dots, N, j = 1, \dots, N, i \neq j \quad (10)$$

Juntando todas as equações com a função objetivo, obtemos o modelo:

maximizar D

sujeito a

$$\|X_{i1} - Ci\|^2 = (R + d_m)^2, \quad i = 1, \dots, N$$

$$\|X_{i2} - Ci\|^2 = (R + d_m)^2, \quad i = 1, \dots, N$$

$$\|X_{i3} - Ci\|^2 = (R + d_m)^2, \quad i = 1, \dots, N$$

$$\|X_{i4} - Ci\|^2 = (R + d_m)^2, \quad i = 1, \dots, N$$

$$\frac{X_{i1} + X_{i3}}{2} = C_i, \quad \frac{X_{i2} + X_{i4}}{2} = C_i, \quad i = 1, \dots, N,$$

$$\|X_{i1} - X_{i2}\|^2 = 2 \cdot (R + d_m)^2, \quad i = 1, \dots, N,$$

$$\|X_{ik} - X_{jl}\|^2 \geq D^2,$$

$$i \neq j, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \quad k = 1, \dots, 4, \quad l = 1, \dots, 4,$$

$$D \geq D_{min},$$

$$a_i \left(C_{j1} + \frac{a_i(R + d_m + d_p)}{\|(a_i, b_i)\|} \right) + b_i \left(C_{j2} + \frac{b_i(R + d_m + d_p)}{\|(a_i, b_i)\|} \right) + c_i \geq 0,$$

$$i = 1, \dots, M \text{ e } j = 1, \dots, N,$$

$$\|C_i - C_j\|^2 \geq (2R + 2d_m)^2, \quad \text{para } i = 1, \dots, N, j = 1, \dots, N, i \neq j.$$

Cabe mencionar, que da forma como foi modelado, o problema poderia se encaixar na classe de problemas de empacotamento [E. G. Birgin and Ronconi 2005], que de um modo geral, são de natureza NP-completos [R.J. Fowler and S.L 1981]. Contudo, ele não é de natureza puramente combinatória, uma vez que precisamos também maximizar distâncias, o que dificulta uma exposição neste momento sobre a amplitude do problema.

4. Ponto Inicial

O problema que modelamos será resolvido utilizando um método de pontos interiores primal-dual [Luenberger and Ye 1984], que é um método iterativo cuja eficiência e robustez depende de um bom ponto inicial.

Para resolver o problema é necessário então, fornecer bons pontos iniciais para o centro das mesas e das cadeiras que serão alocados no estabelecimento.

Para encontrar o ponto inicial dos centros das mesas resolvemos o problema que consiste em maximizar a menor distância entre os centros das mesas. Note que este é uma simplificação do problema original. Embora este problema também demande bons pontos iniciais, heurísticas clássicas podem ser utilizadas, pois seu conjunto de restrições é mais simples que o problema original. Noutros termos, a solução do problema de encontrar os centros é obtida de forma mais robusta. Ressaltamos ainda, que utilizamos a solução deste problema, apenas como ponto inicial para o problema original.

Uma vez determinados os pontos iniciais para os centros das mesas, os pontos iniciais para a posição das cadeiras podem ser tomados de forma que os segmentos formados por cadeiras opostas de uma mesma mesa, sejam paralelos, cada um a um dos eixos coordenados.

Para a mesa i com centro inicial $C_i = (C_{i1}, C_{i2})$, por exemplo, a posição inicial das cadeiras será

$$\begin{aligned} X_{i1} &= (C_{i1}, C_{i2} + R + d_m) \\ X_{i2} &= (C_{i1} + R + d_m, C_{i2}) \\ X_{i3} &= (C_{i1}, C_{i1} - (R + d_m)) \\ X_{i4} &= (C_{i1} - (R + d_m), C_{i2}) \end{aligned}$$

O ponto inicial do problema simplificado é obtido de forma randômica.

5. Adaptação do modelo para outros casos

Podemos aproveitar a mesma modelagem para determinar o posicionamento de cadeiras de mesas quadradas de lado L . Neste caso, podemos pensar nas mesas quadradas inscritas em circunferências de raio $L/\sqrt{2}$.

Também é possível adaptar o modelo para mesas quadradas ou circulares com lados ou raio distintos, bastando para isso, construir um vetor com as medidas dos raios ou lados e modificar o modelo de forma que, nas equações, as medidas relacionadas a mesas distintas se alterem.

6. Implementação

O modelo matemático foi implementado na linguagem de programação *Julia*, versão 1.0.5, que é uma linguagem de propósito geral e código aberto que visa um bom desempenho e produtividade [Castelluccia 2017].

Mais especificamente, utilizamos a biblioteca *JuMP.jl*. A *JuMP* (Julia for Mathematical Programming) é uma linguagem de modelagem embutida na *Julia* e que suporta mais de 10 *solvers* para problemas de otimização. Uma de suas principais características

é a independência do *solver* a ser utilizado. Isto significa que um modelo pode ser implementado em JuMP independentemente do *solver* que será utilizado para sua solução.

Além disso, já visando a criação de uma interface com o usuário (dono do estabelecimento), onde ocorrerão trocas de dados, utilizamos arquivos com os dados do problema no formato *JSON* (JavaScript Object Notation).

Uma vez implementados os modelos, para solucioná-los, utilizamos o *solver Ipopt* (Interior Point OPTimizer)¹ [Wachter and Biegler 2006] que possui rotinas que implementam um método de pontos interiores primal-dual para programação não-linear [Luenberger and Ye 1984] e para tanto, utilizamos junto à *JuMP* a *Ipopt.jl*², que é uma interface da Julia para o solver Ipopt <https://ipoptjl.readthedocs.io/en/latest/ipopt.html>.

Para desenhar a solução apresentada utilizamos o pacote *TikZ* do *Latex*.

7. Resultados

A seguir apresentamos algumas configurações do problema e as soluções obtidas para elas com a nossa implementação.

As mesas são representadas pelos círculos maiores azuis e as cadeiras pelos 4 pontos em torno delas. Os círculos tracejados em cinza possuem o raio igual a metade de D , a distância mínima maximizada, e estão centrados nas cadeiras. Quando os círculos tracejados de duas cadeiras se tangenciam é porque elas estão à essa distância D uma da outra.

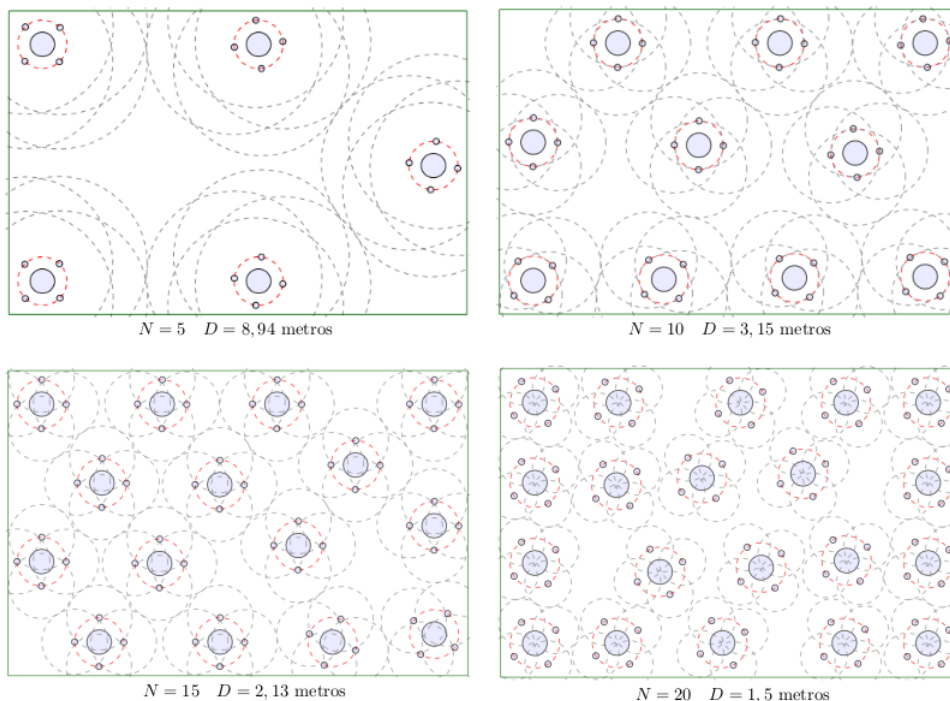


Figura 5. Mesas alocadas em uma região retangular de tamanho 15×10

¹Aos leitores interessados em maiores detalhes sobre o solver Ipopt, recomendamos <https://coin-or.github.io/Ipopt/index.html>

²Mais detalhes podem ser vistos em <https://ipoptjl.readthedocs.io/en/latest/ipopt.html>

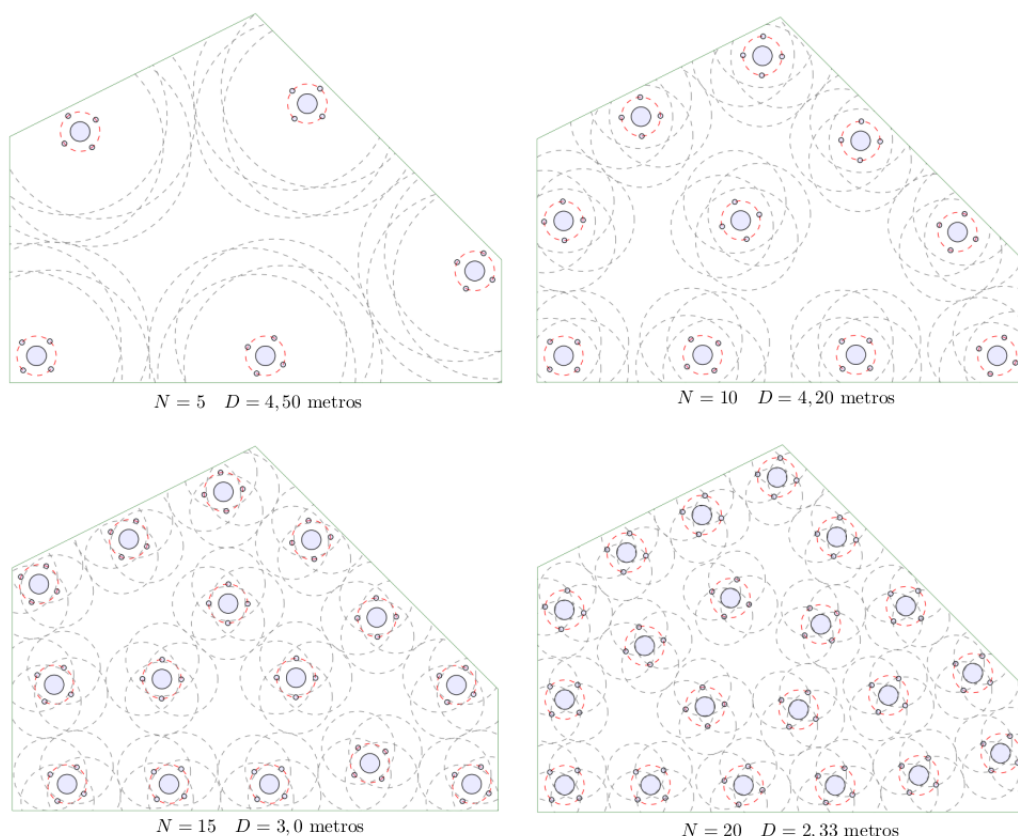


Figura 6. Mesas alocadas em uma região convexa irregular

A Figura 5 mostra a disposição de mesas redondas com raio igual a 0,4m em um salão retangular de tamanho 15×10 . São apresentados os resultados para 5, 10, 15 e 20 mesas. Para 20 mesas obtivemos uma distância mínima de 1,5 metros, mantendo assim uma distância acima da recomendada. Para esta região retangular e este tamanho de mesa, o problema teve solução até 30 mesas, com distância mínima de 1,02m. Para 31 mesas o problema é infactível, fornecendo uma distância mínima de 0,95m, abaixo da recomendada.

A Figura 6 exibe a disposição de mesas redondas de raio igual a 0,4m em um salão convexo irregular cujos vértices são os pontos (0, 10), (10, 15), (20, 5), (20, 0) e (0, 0). São apresentados os resultados para 5, 10, 15 e 20 mesas. Com 20 mesas foi obtida uma distância mínima de 2,33 metros, mantendo também uma distância acima da recomendada.

		5 mesas	10 mesas	15 mesas	20 mesas
região retangular	Média	0.119	0.543	2.001	13.231
	Desvio Padrão	0.021	0.018	0.054	0.305
região convexa	Média	0.263	1.268	4.607	21.990
	Desvio Padrão	0.016	0.001	0.075	0.437

Tabela 1. Média de 3 tempos de resolução

A Tabela 1 exibe o tempo médio de resolução de cada caso, obtidos por meio de

três repetições.

8. Interface do usuário

Para manipulação do usuário foi criada a página exibida na Figura 7. Nos campos adequados o usuário insere a quantidade de mesas e suas dimensões. Na malha quadriculada desenha a região do seu estabelecimento, criando os pontos que a delimitam por meio do clique do mouse. Ao clicar em “Calcular”, o problema é processado e uma vez resolvido, é fornecido para o usuário o valor da distância mínima e exibida na malha quadriculada as posições das mesas e cadeiras que maximizam a distância mínima entre as pessoas, como mostra a Figura 8.

Crie os pontos que delimitem seu estabelecimento:

Número de mesas: 5

Tamanho das mesas (LxC): 1 x 1

Calcular

Figura 7. Interface com o usuário

Crie os pontos que delimitem seu estabelecimento:

Número de mesas: 7

Tamanho das mesas (LxC): 0.8 x 0.8

Calcular

2.374877087028346

Figura 8. Interface com o usuário

Observe que a página oferece campos para inserir dimensões de mesas retangulares. Se as mesas forem quadradas, o raio dos círculos que circunscrevem as mesas são calculados conforme descrito na seção 5. Se as mesas forem retangulares, para a solução do problema, os retângulos também são inscritos em círculos, mas neste caso, se perde

um pouco de espaço na otimização, pelo fato de sobrar mais espaço não ocupado pelo retângulo dentro do círculo.

Para a comunicação entre a interface do usuário e o modelo matemático implementado foram utilizadas as bibliotecas *HTTP.jl* e *sockets.jl* disponíveis em *Julia*. Mais especificamente, estas bibliotecas são utilizadas para subir um servidor Web que responda às requisições feitas por uma chamada Javascript.

A comunicação entre a interface do usuário e *Julia* é feita utilizando o objeto *JSON*. Para que seja possível ler o objeto e convertê-lo em um objeto na linguagem *Julia* e de forma inversa, converter um objeto na linguagem *Julia* para um objeto *JSON*, utilizamos a biblioteca em *Julia* denominada *JSON2.jl*.

Quando o usuário clica em “Calcular”, o *JavaScript* envia uma requisição para o servidor HTTP, que por sua vez chama o modelo em *Julia*.

O programa em *Julia* resolve o problema e devolve a solução que é desenhada utilizando o objeto *Canvas* manipulado por meio de *JavaScript*.

9. Considerações Finais

Propomos neste trabalho uma modelagem e implementação para o problema de maximizar a distância mínima entre as pessoas, e conseqüentemente entre as mesas, em um estabelecimento do ramo da alimentação.

Melhorias ainda podem ser feitas no modelo proposto, de forma a contemplar de maneira ótima mesas retangulares que não são quadradas e regiões que não sejam convexas. Este problema também poderia ser adaptado para qualquer cômodo em que se deseje maximizar a distância entre objetos, por exemplo, as carteiras de uma sala de aula.

Referências

- Castelluccia, P. B. (2017). Tutorial convidado julia e jump: Novas ferramentas para programação matemática. In *Pesquisa Operacional para o Desenvolvimento*, volume 11, pages 48–61.
- E. G. Birgin, J. M. and Ronconi, D. (2005). Optimizing the packing of cylinders into a rectangular container: a nonlinear approach. *European Journal of Operational Research*, 160:19–33.
- Luenberger, D. G. and Ye, Y. (1984). *Linear and Nonlinear Programming*, volume 67.
- OMS and FAO (2020). Covid-19 e segurança dos alimentos: Orientações para as autoridades competentes responsáveis pelos sistemas de controle da segurança dos alimentos. World Health Organization and the Food and Agriculture Organization of the United Nations.
- R.J. Fowler, M. P. and S.L, T. (1981). Optimal packing and covering in the plane are np-complete. *Information Processing Letters*, 12:133–137.
- Sebrae (2020). Orientações para a retomada segura das atividades - bares - lanchonetes - restaurantes. pages 48–61. Serviço Brasileiro de Apoio às Micro e Pequenas Empresas.
- Wachter, A. and Biegler, L. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57.