

PROPOSTA DE UM SCRIPT EM LINGUAGEM R PARA APRENDIZAGEM E RESOLUÇÃO DO PROBLEMA DO TRANSPORTE

¹Adilson Vilarinho Terra; ¹adilsonvilarinho@id.uff.br; ¹Universidade Federal Fluminense;
²Marcos dos Santos; ²marcosdossantos@ime.eb.br; ²Instituto Militar de Engenharia;
³Miguel Ângelo Lellis Moreira; ³miguellellis@id.uff.br; ³Universidade Federal Fluminense.

RESUMO: *O Problema do Transporte é um dos problemas de otimização combinatória mais estudados da Pesquisa Operacional. O objetivo desse trabalho é apresentar um script desenvolvido em linguagem R capaz de determinar como transportar mercadorias de maneira otimizada, além de servir como recurso instrucional, facilitando a compreensão e aprendizado da Programação Linear. Tal script está fundamentado no método Simplex e na modelagem matemática clássica do problema de transporte, onde as restrições são os limites de oferta dos pontos de origem e as demandas dos pontos de destino. Para elucidação do funcionamento do script, foi utilizado um exemplo numérico de transporte de respiradores de 5 fábricas até 5 hospitais. Aplicado em tal exemplo fictício, o script foi capaz de encontrar a solução ótima do problema de maneira intuitiva, cumprindo com sua proposta educativa.*

PALAVRAS-CHAVE: *Problema do Transporte; Programação Linear; Linguagem R.*

ABSTRACT: *The Transportation Problem is one of the most studied combinatorial optimization problems in Operational Research. The objective of this work is to present a script developed in R language capable of determining how to transport goods in an optimized way, besides serving as an instructional resource, facilitating the understanding and learning of Linear Programming. Such a script is based on the Simplex method and the classical mathematical modeling of the transport problem, where constraints are the supply limits of the source points and the demands of the destination points. To elucidate the functioning of the script, a numerical example of transport of respirators from 5 factories to 5 hospitals was used. Applied in such a fictitious example, the script was able to find the optimal solution of the problem intuitively, fulfilling its educational proposal.*

KEYWORDS: *Transportation Problem; Linear Programming; R Language.*

1. Introdução

O ambiente corporativo, caracterizado por grande competitividade entre as organizações e por disputas crescentes por maiores fatias de mercado, exige que os gestores estejam preparados para tomar decisões que representem melhorias efetivas no desempenho das empresas. Diante de tal cenário, é comum o uso de técnicas e ferramentas presentes na Pesquisa Operacional (PO), em prol de viabilizar a otimização de recursos, redução de custos e, conseqüentemente, uma melhor compreensão, análise e solução de problemas complexos (MOREIRA *et al.*, 2020).

Segundo Santos *et al.* (2017), a PO em forma de ciência é composta por inúmeras técnicas e modelos com base na otimização de sistemas produtivos. De forma complementar, Maciel *et al.* (2020) afirmam que a PO é uma das áreas de conhecimento de grande relevância da engenharia, viabilizando o apoio a tomada de decisão mediante a implementação de métodos e algoritmos científicos.

No entanto, apesar de ser uma ciência de suma importância na tomada de decisões estratégicas, o ensino da pesquisa operacional não é trivial, pois usa estatística, matemática e problemas de fato complexos e variados, o que acaba se tornando um desafio para o professor e o aluno (HIRATA *et al.*, 2020). Com o propósito de trazer um recurso instrucional que facilite o aprendizado e ensino de PO, o presente artigo propõe um script intuitivo e de fácil compreensão, desenvolvido em linguagem R, como ferramenta de aprendizagem e resolução de Problemas de Transporte em sua modelagem clássica.

O Problema do Transporte, de acordo com Hillier e Lieberman (2013) recebeu essa denominação em virtude de várias de suas aplicações envolverem determinar como transportar mercadorias de maneira otimizada, é um dos mais utilizados no início da aprendizagem de PO. Ele é caracterizado por atribuir quantidades de um produto que será transportado de um ponto de origem (por exemplo, fábricas) para um ponto de destino (por exemplo, depósitos). Para esse fim, é necessário obedecer a limitação da capacidade de oferta dos produtos de forma a atender toda a demanda existente (BELFIORE e FÁVERO, 2013).

Por último, existe o custo associado ao transporte das unidades do produto, que está ligado ao gasto com combustível, pedágio e toda logística envolvida. O principal objetivo do problema reside em minimizar os custos envolvidos no transporte, respeitando as restrições de oferta e demanda. Arenales *et al.* (2015) ressaltam que o Problema do Transporte é um dos principais tipos de problemas da Programação Linear, e sua principal contribuição nesse caso é minimizar o custo de transporte de produtos para os seus destinos.

O recurso computacional utilizado para a resolução do Problema de Transporte foi o script desenvolvido em linguagem R apresentado na seção 2.4. Para demonstrar seu funcionamento, foi utilizado na seção 3 um exemplo numérico com dados fictícios para minimização dos custos de transporte de respiradores utilizados no combate contra a pandemia de COVID-19, das fábricas até os hospitais. O método utilizado no script para a resolução do Problema de Transporte é o algoritmo Simplex, auxiliando na minimização dos custos de transporte.

2. Fundamentação teórica

Conforme apresentado pela Sociedade Brasileira de Pesquisa Operacional (SOBRAPO), a Pesquisa Operacional (PO), é uma das áreas de conhecimento de grande relevância da engenharia, por ser uma ferramenta que mediante métodos científicos e matemáticos auxiliam na tomada de decisão (SOBRAPO, 2021). Santos *et al.* (2015) complementam afirmando que

a PO através de seus modelos matemáticos tem por objetivo a resolução de problemas reais, apresentando um caráter altamente multidisciplinar.

2.1. Programação linear

A Programação Linear (PL) é um método matemático que surgiu durante a Segunda Guerra Mundial com a finalidade de otimizar o uso dos escassos recursos disponíveis. Por isso, A PL é considerada um dos maiores avanços do século XX (HILLIER;LIEBERMAN, 2013). Segundo Bressan e Ramos (2020) a Programação Linear pode ser definida como uma ciência que aplica métodos científicos em problemas envolvendo o controle e a otimização de problemas.

As aplicações de modelos de PL podem ser diversas e ilimitadas, como por exemplo: problema da dieta, otimização de portfólios, telecomunicações, manufatura e transporte (ABOELMAGD 2018). Ainda de acordo com Aboelmagd (2018), a formulação de problemas de PL segue os três passos a seguir:

- i. Definição das variáveis de decisão, que são expressas como símbolos, por exemplo x_i ;
- ii. Definição das restrições em função das variáveis de decisão;
- iii. Definição da função objetivo, em função das variáveis de decisão, que será otimizada (maximização ou minimização).

Dessa forma, um modelo de programação linear busca a otimização da F_0 com base em um conjunto de variáveis x_i , sendo y_i a quantidade ótima a ser alocada para cada variável em análise, dentro do conjunto de restrições M , por exemplo. Como forma de exemplificação e facilitação da compreensão, a equação (1) apresenta um modelo genérico de PL, cuja função objetivo é de minimização.

$$\text{Min } Z = \sum_{i=1}^m x_i \cdot y_i \quad (1)$$

Restrito a:

$$\begin{aligned} 0 &\leq x_i \leq 1 \\ y_i &\in \mathbb{R} \\ \sum_{i=1}^m x_i \cdot c_i &\leq M \end{aligned}$$

2.2. Formulação matemática do problema do transporte

Para obter a formulação do problema, assume-se que existem m pontos de partida e n destinos, cada um dos quais a fonte i tem um limite máximo de oferta, chamado o_i , e cada destino j tem uma demanda mínima a ser suprida d_j . Ademais, o custo de transporte de uma unidade da mercadoria de i a j é igual a c_{ij} . A função objetivo Z é responsável por minimizar os custos totais de distribuição das unidades da mercadoria. Já as restrições têm o papel de garantir que cada origem i não exceda sua capacidade de fornecimento e que cada destino j receba o mínimo requerido. Na figura 1 encontra-se a representação em forma de rede do problema de transporte descrito.

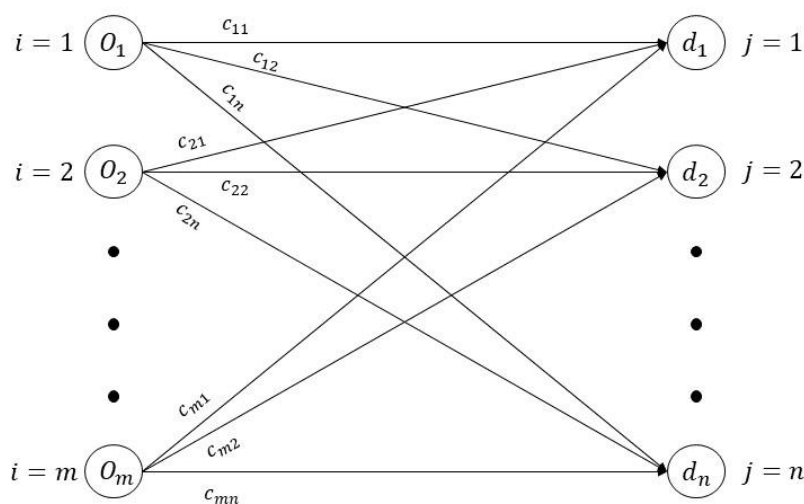


FIGURA 1 – Representação em forma de rede. Fonte: Adaptado de Hillier e Liberman (2013).

A variável de decisão que vai indicar a quantidade de mercadorias a ser transportada da origem ao destino com o menor custo possível é representada por x_{ij} . Por fim, a última equação de restrição representa a factibilidade (ou equilíbrio) do problema, pois ela faz com que cada destino j receba a quantidade mínima (ou superior a mínima) de unidades de mercadoria da origem i . A formulação clássica do problema de transporte é descrita assim:

$$\text{Min } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij}$$

Restrito a:

$$\sum_{j=1}^n x_{ij} \leq o_i \quad \forall i = 1, 2, \dots, m$$

$$\sum_{i=1}^m x_{ij} \geq d_j \quad \forall j = 1, 2, \dots, n$$

$$\sum_{i=1}^n o_i \geq \sum_{j=1}^m d_j$$

$$x_{ij} \geq 0 \quad \forall i = 1, 2, \dots, m; j = 1, 2, \dots, n$$

2.3. Proposta de script em linguagem R

Conforme discutido na seção de introdução, apresenta-se um script baseado no tratamento de dados para resolução de problemas de transporte em sua formulação matemática clássica. Dado modelo apresenta a integração, coleta e tratamento de dados, juntamente com a exposição de informação em base numérica.

Desenvolvido com base no software R (R CORE TEAM, 2021), o script proporciona uma interação simples e intuitiva, providenciando uma análise prática de um conjunto de dados relativos ao problema de transporte de mercadorias de m pontos de origem a n pontos de destino.

A entrada de dados ocorre mediante arquivos .csv, comumente utilizados para o compartilhamento de dados entre softwares. Com base no conjunto de dados, o modelo confere a factibilidade do problema, modela como PL e calcula a solução ótima através dos pacotes LP Solve (BERKELAAR, 2020) e LpSolveAPI (KONIS; SCHWENDINGER, 2020), provenientes do próprio software R. A figura 2 apresenta a estrutura de entrada e leitura de dados do script e a conversão desses dados para o formato de listas.

```

library(lpSolve)
library(lpSolveAPI)

dados_nomes <-
  read.csv2(
    "entrada_transporte.csv",
    header = FALSE,
    sep = ";",
    quote = "\"",
    dec = ".",
    fill = TRUE,
    stringsAsFactors = FALSE
  )

dados <-
  read.csv2(
    "entrada_transporte.csv",
    header = TRUE,
    sep = ";",
    quote = "\"",
    dec = ".",
    fill = TRUE,
    stringsAsFactors = FALSE
  )
dados

#Lista de nomes das fontes
lista_fontes <- as.vector(unlist(dados[1]))
lista_fontes <- lista_fontes[1:(length(lista_fontes) - 1)]

#Lista de nomes dos mercados consumidores
lista_mc <- c()
for (i in 2:(length(dados) - 1)) {
  lista_mc <- append(lista_mc, dados_nomes[1, i])
}

#lista de custos
custos <- c()
for (j in 1:length(lista_fontes)) {
  custos <-
    append(custos, as.vector(unlist(dados[j, 2:(length(lista_mc) + 1)])))
}

#lista de valores de demandas
lista_demanda <-
  as.vector(unlist(dados[length(dados)[1]], 2:(length(dados) - 1)))
soma_demanda <- sum(lista_demanda)
n_demanda <- length(lista_demanda)

#lista de valores das fontes
lista_valor_fontes <- as.vector(unlist(dados[length(dados)]))
lista_valor_fontes <-
  lista_valor_fontes[1:(length(lista_valor_fontes) - 1)]
soma_fontes <- sum(lista_valor_fontes)
n_fontes <- length(lista_fontes)

```

FIGURA 2 – Entrada, leitura e conversão de dados para formato de listas. Fonte: Autores (2021).

Com os dados convertidos para o formato de lista, o script executará os próximos passos até chegar na solução ótima. Primeiro ele confere a factibilidade (equilíbrio) do problema a partir da massa de dados inserida. Se o problema for factível, ele será modelado como PL onde as restrições de demanda de cada destino j e de oferta de cada fonte i serão implementadas. As etapas descritas nesse parágrafo podem ser visualizadas na figura 3 a seguir.

```

#conferindo se o problema está equilibrado
if (soma_fontes >= soma_demanda) {
  print("O Problema está equilibrado!")
} else {print('O problema não está equilibrado! Por favor, revise os dados.')}

# Modelo de Otimização
n_variaveis <- length(custos)
lp_model <- make.lp(0, n_variaveis)
set.objfn(lp_model, custos)

# Restrição de demanda
v_aux <- rep(1, n_fontes)
for (j in 1:n_demanda) {
  v_ind <- c()
  for (i in 1:n_fontes) {
    ind <- ((i * n_demanda) - n_demanda) + j
    v_ind <- append(v_ind, ind)
  }
  add.constraint(lp_model, v_aux, '=', indices = v_ind, lista_demanda[j])
}

#Restrição das fontes
v_aux2 <- rep(1, n_demanda)
for (j in 1:n_fontes) {
  v_ind2 <- c()
  for (i in 1:n_demanda) {
    ind2 <- ((j * n_fontes) - n_fontes) + i
    v_ind2 <- append(v_ind2, ind2)
  }
  add.constraint(lp_model, v_aux2, '<=', indices = v_ind2, lista_valor_fontes[j])
}

lp.control(lp_model, sense = "min")
set.type(lp_model, 1:n_variaveis, type = "integer")
write.lp(lp_model, "lp_modeling.txt")
solve(lp_model)
dec <- get.variables(lp_model)
dec

```

FIGURA 3 – Verificação de factibilidade e implementação do modelo. Fonte: Autores (2021).

Após implementação do modelo de otimização, os últimos comandos executados pelo script serão responsáveis por mostrar ao usuário a solução ótima obtida através do Método Simplex. Tal solução está dividida em duas etapas: a primeira mostra o resultado de quanto se deve transportar em unidades de mercadoria de cada origem i para cada destino j . Já a segunda que é chamada de resultado ponderado informa o valor em reais do custo de transporte das mercadorias de cada origem i para cada destino j . O código que executa esses comandos se encontra na figura 4.

```
#Resultado
tabela_resultado <- dados[1:n_fontes, 1:(n_demanda + 1)]
x <- 1
for (j in 1:n_fontes) {
  for (i in 2:(n_demanda + 1)) {
    tabela_resultado[j, i] <- dec[x]
    x <- x + 1
  }
}
tabela_resultado

#Resultado Ponderado
ponderado <- dec * custos
tabela_pond <- dados[1:n_fontes, 1:(n_demanda + 1)]
x <- 1
for (j in 1:n_fontes) {
  for (i in 2:(n_demanda + 1)) {
    tabela_pond[j, i] <- ponderado[x]
    x <- x + 1
  }
}
tabela_pond

print(paste(c("O custo total de transporte é: ", sum(ponderado))))
```

FIGURA 4 – Resultado do modelo. Fonte: Autores (2021).

3. Exemplo numérico

Para demonstrar o funcionamento do script abordado na seção anterior, o exemplo de Problema de Transporte utilizado no presente trabalho será o da distribuição de respiradores usados no combate a pandemia, desde as fábricas até os hospitais. O objetivo é descobrir quantos respiradores cada fábrica terá de enviar para cada hospital, nenhum hospital e nenhuma fábrica opere além de sua capacidade. Serão quatro fábricas (ponto de origem) i , onde $i = 1, 2, 3, 4$; e quatro hospitais (ponto de destino) $j = 1, 2, 3, 4$. A representação em redes do problema contendo os dados de capacidade de oferta das fábricas o_i , a demanda por respiradores d_j e os custos de transporte por unidade de respirador c_{ij} , se encontram na figura 5.

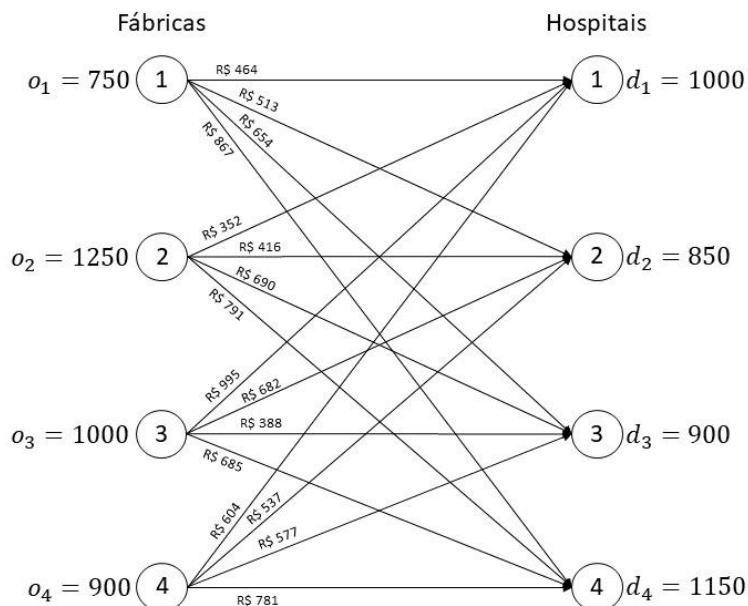


FIGURA 5 – Representação em rede do exemplo numérico. Fonte: Autores (2021).

Com os dados do problema, sua modelagem em PL com a função objetivo e suas restrições ficam da seguinte forma:

$$\begin{aligned} \text{Min } Z = & 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} + 352x_{21} + 416x_{22} + 690x_{23} + 791x_{24} \\ & + 995x_{31} + 682x_{32} + 398x_{33} + 685x_{34} + 604x_{41} + 537x_{42} + 577x_{43} \\ & + 781x_{44} \end{aligned}$$

Restrições de oferta:

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} & \leq 750 \\ x_{21} + x_{22} + x_{23} + x_{24} & \leq 1250 \\ x_{31} + x_{32} + x_{33} + x_{34} & \leq 1000 \\ x_{41} + x_{42} + x_{43} + x_{44} & \leq 900 \end{aligned}$$

Restrições de demanda:

$$\begin{aligned} x_{11} + x_{21} + x_{31} + x_{41} & \geq 1000 \\ x_{12} + x_{22} + x_{32} + x_{42} & \geq 850 \\ x_{13} + x_{23} + x_{33} + x_{43} & \geq 900 \\ x_{14} + x_{24} + x_{34} + x_{44} & \geq 1150 \end{aligned}$$

Para dar início ao script e obter a solução ótima deste PL, é necessário organizar os dados do modelo em uma tabela de arquivo formato .csv para que seja feita a entrada e leitura desses dados. A tabela de entrada com os dados do problema de exemplo protótipo encontra-se na tabela 1.

TABELA 1 – Entrada de dados no script

	Hospital 1	Hospital 2	Hospital 3	Hospital 4	Oferta de respiradores
Fábrica 1	R\$ 464	R\$ 513	R\$ 654	R\$ 867	750
Fábrica 2	R\$ 352	R\$ 416	R\$ 690	R\$ 791	1250
Fábrica 3	R\$ 995	R\$ 682	R\$ 388	R\$ 685	1000
Fábrica 4	R\$ 604	R\$ 537	R\$ 577	R\$ 781	900
Demanda de respiradores	1000	850	900	1150	

Fonte: Autores (2021).

Uma vez inseridos os dados e colocando o script para rodar, ele irá resolver o problema através do Método Simplex retornará a solução ótima informando através de uma tabela de resultados o número de respiradores a se transportar de cada origem até cada destino. Ademais, o script também retornará a solução ponderada, que consiste em uma tabela com o custo do transporte minimizado. Ambas as podem ser visualizadas abaixo nas tabelas 2 e 3.

TABELA 2 – Número de respiradores transportados de i a j

	Hospital 1	Hospital 2	Hospital 3	Hospital 4
Fábrica 1	0	600	0	150
Fábrica 2	1000	250	0	0
Fábrica 3	0	0	900	100
Fábrica 4	0	0	0	900

Fonte: Autores (2021).

TABELA 3 – Tabela ponderada do custo de transporte

	Hospital 1	Hospital 2	Hospital 3	Hospital 4
Fábrica 1	0	R\$ 307.800	0	R\$ 130.050
Fábrica 2	R\$ 352.000	R\$ 104.000	0	0
Fábrica 3	0	0	R\$ 349.200	R\$ 68.500
Fábrica 4	0	0	0	R\$ 702.900

Fonte: Autores (2021).

4. Considerações finais

Neste trabalho foi discutida a larga aplicabilidade da PL e dos Problemas de Transporte, bem como sua importância nas mais diversas áreas da Pesquisa Operacional. A ferramenta nesse trabalho facilita a resolução do Problema de Transporte com o método Simplex, e contribui para a aprendizagem em PL, visto que modela o problema em sua forma clássica através de um script intuitivo em Linguagem R. Ademais, com o auxílio de um exemplo, chegou-se na solução ótima para o problema de transporte, ilustrando o funcionamento do script, que também é capaz de resolver problemas maiores da mesma maneira. Como forma de trabalhos futuros, pretende-se possibilitar a implementação do script na web com acesso online através do pacote de desenvolvimento *Shiny*.

Referências

- ABOELMAGD, Yasser M.r.. Linear programming applications in construction sites. **Alexandria Engineering Journal**, v. 57, n. 4, p. 4177-4187, dez. 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1110016818302126>. Acesso em: 23 set 2021.
- ARENALES, M. N. et al. Pesquisa operacional: para cursos de engenharia. 2. ed. Rio de Janeiro: Elsevier, 2015.
- BELFIORE, Patrícia; FÁVERO, Luiz Paulo. Pesquisa operacional para cursos de Engenharia. Rio de Janeiro: Elsevier, 2013.
- BERKELAAR, M. LpsolveR package version 5.6.15, 2020.
- BRESSAN, G.M; RAMOS, T, A. Linear Programming and Transport Problem Application to optimize agricultural production. **Revista Eletrônica Paulista de Matemática**, São Paulo, v. 19, 2020.
- HILLIER, F.S.; LIEBERMAN, G.J. Introdução à pesquisa operacional. 9. ed. Porto Alegre: AMGH, 2013.

HIRATA, A.C.L; CUNHA, A.M.S; CARMO, L.A.G; SANTOS, W.E.L; GEORGES, M.R.R. Springfield School Board: Aplicação de um caso de ensino na disciplina de Pesquisa Operacional. **Anais do LII Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. João Pessoa/PB, 2020.

KONIS, K.; SCHWENDINGER, F. LpsolveAPIR package version 5.5.2., 2020.
MACIEL, R.P.; PEREIRA, J.G.R.; GALVÃO, L.C.; SOUZA, W.R.; SANTOS, Y.B.I. Otimização de custos na produção de euterpe oleracea (açai): um estudo de caso. **Anais do XXVII Simpósio de Engenharia de Produção (SIMPEP)**. Bauru/SP, 2020.

MOREIRA, M. Â. L.; GOMES, C. F. S.; DOS SANTOS, M.; DO CARMO SILVA, M.; ARAUJO, J. V. G. A. PROMETHEE-SAPEVO-M1 a Hybrid Modeling Proposal: Multicriteria Evaluation of Drones for Use in Naval Warfare. **Springer Proceedings in Mathematics & Statistics**. 1st ed. Cham: Springer, 2020. p. 381–393. DOI 10.1007/978-3-030-56920-4_31.

R CORE TEAM. R: A Language and Environment for Statistical Computing. (2021).

SANTOS, M.; LIMA, I.C.; CARVALHO, F.B.; REIS, M.F.; JUNIOR, P.R.S. O uso da Programação Linear Inteira (PLI) no apoio a decisão e otimização do mix de produção. **Anais do XXXVII Encontro Nacional de Engenharia de Produção (ENEGEP)**. Joinville/SC, 2017.

SANTOS, M.; QUINTAL, R.S; PAIXÃO, A. C.; GOMES, C. F. S. Simulation of Operation of An Integrated Information for Emergency Pre-Hospital Care In Rio de Janeiro Municipality. **Procedia Computer Science**. Rio de Janeiro, v. 55, p. 931-938, 2015.

SOBRAPO. “O que é pesquisa operacional?”. Sociedade Brasileira de Pesquisa Operacional. 2021. Disponível em: www.sobrapo.org.br/o-que-e-pesquisa-operacional.