

UM MODELO DE PROGRAMAÇÃO LINEAR INTEIRA PARA A ALOCAÇÃO DE PACIENTES A MÉDICOS

Vítor Hugo Cândido Camargo, Unidade de Matemática e Tecnologia, Universidade Federal de Catalão, vhc.camargo@gmail.com

Alexandre Magno Bernardes Fonseca, Unidade de Matemática e Tecnologia, Universidade Federal de Catalão, alexandreufcat@gmail.com

Thiago Alves de Queiroz, Unidade de Matemática e Tecnologia, Universidade Federal de Catalão, taq@ufcat.edu.br

***Resumo:** Este trabalho apresenta um modelo de programação linear inteira para o problema de alocação de pacientes a médicos em unidades de pronto-socorro. O pronto-socorro recebe os pacientes no decorrer do seu horizonte de funcionamento e precisa determinar quando cada paciente deverá ser alocado a um dos médicos disponíveis. Cada paciente só pode ser atendido após passar pela triagem, momento em que fica à espera de atendimento. Além disso, cada paciente recebe um grau de urgência, que determina o tempo máximo dentro do qual ele deveria ser atendido, caso contrário, ocorrerá um atraso. O objetivo do problema é alocar os pacientes de forma que o atraso total ponderado seja o mínimo possível. O modelo desenvolvido assume que todas as informações são conhecidas a priori e, assim, utiliza variáveis de decisão de três e dois índices para determinar por qual médico cada paciente será atendido, o momento de início do atendimento e se, ocorreu atraso, de quanto foi esse atraso. A validação do modelo matemático ocorre sobre instâncias da literatura contendo 20 pacientes e com o número de médicos variando de 2 a 10. Os resultados apontam que o modelo consegue obter a solução ótima mais rapidamente para os casos onde há maior disponibilidade de médicos.*

***Palavras-chave:** alocação de pacientes, pronto-socorro, programação linear inteira, atraso total ponderado.*

1. INTRODUÇÃO

Hospitais e clínicas precisam cada vez mais estar em busca de otimizar seus processos e minimizar qualquer desperdício de tempo. Isso pode ser conseguido através de uma boa gestão da qualidade presente no ambiente organizacional, que ocorre através da tomada de boas decisões. Uma das formas de se conseguir isso é resolvendo o Problema de Escalonamento de Pacientes (PEP) em unidades de pronto-socorro, visto que geralmente os prontos-socorros enfrentam problemas de superlotação (DI SOMMA et al., 2015). Trata-se de um problema de otimização combinatória, que lida majoritariamente com objetivos qualitativos e busca por minimizar o tempo total de atraso dado o atendimento a um conjunto de pacientes.

O PEP surge na literatura de problemas de programação da produção, onde as máquinas e tarefas em uma organização podem assumir formas diferentes. As máquinas podem representar pistas em um aeroporto, equipes em um canteiro de obras, unidades de processamento em um ambiente de computação e, no presente caso, médicos de uma unidade hospitalar. As tarefas também podem ser decolagens e pousos em um aeroporto, estágios em um projeto de construção, a execução de programas de computador e, no presente caso, pacientes requerendo algum tipo de tratamento (PINEDO, 2008). Dessa forma, os termos paciente e tarefa podem ser usados como sinônimos, assim como médico e máquina.

O problema em estudo advém de situações reais em unidades de pronto-socorro, mas também pode ser aplicado em outros contextos da programação da produção. O processo de atendimento

nessas unidades consiste em algumas etapas: a chegada do paciente, que passa pela triagem, recebendo um grau de urgência conforme os sintomas que descreve, um tempo esperado em fila para receber o atendimento pelo médico e qual seria a duração do atendimento; a espera do paciente para receber o atendimento; e o recebimento do atendimento por parte do médico. Neste trabalho, assume-se que os médicos possuem eficiência idêntica e, assim, um paciente pode receber atendimento de qualquer médico sem interferência na duração prevista do seu serviço (QUEIROZ et al., 2021).

Ao considerar que os médicos têm a mesma eficiência e são tratados de forma idêntica, o PEP está relacionado à um processo de tomada de decisão que surge com frequência nos ramos de manufatura e serviços, lidando com o sequenciamento de tarefas para máquinas paralelas idênticas (KRAMER; DELL'AMICO; IORI, 2019).

No PEP, os pacientes possuem um peso, ou urgência, que impacta na sua importância e prioridade para receber o atendimento. Um paciente só pode ser atendido após a liberação pela triagem (*release date*), além de haver um tempo de espera dentro do qual deveria receber o atendimento (*due date*). Quando o atendimento iniciar após o tempo de espera, ocorrerá um atraso, que terá seu valor ponderado pela a urgência do paciente. O objetivo final é que a soma dos atrasos ponderados seja a mínima possível.

Desenvolve-se um modelo de programação linear inteira para o PEP, o qual é resolvido com o algoritmo *branch-and-cut* do resolvidor *Gurobi Optimizer*. Para testar a eficácia do modelo, consideramos instâncias propostas por Queiroz et al. (2021) para diferentes configurações de médicos dentro do pronto-socorro. Em geral, os resultados mostram que o modelo é eficaz quando se tem mais médicos a disposição. Na próxima seção há uma breve revisão da literatura, enquanto a definição formal do problema e o modelo de programação linear inteira são discutidos na Seção 3. Os testes computacionais são apresentados na Seção 4, enquanto as conclusões e direções para trabalhos futuros são dadas na Seção 5.

2. REFERENCIAL TEÓRICO

Problemas de sequenciamento e programação da produção, referenciados na literatura como *scheduling problems* (problemas de escalonamento), estão relacionados ao processo de tomada de decisão quanto ao sequenciamento de tarefas em máquinas, sendo muito comum na Pesquisa Operacional. Em geral, esses problemas lidam com a alocação de tarefas em máquinas, com diferentes objetivos, por exemplo, minimizar o tempo total ou o atraso total gasto com o processamento de todas as tarefas (PINEDO, 2008).

A literatura em problemas de escalonamento é abrangente. Uma revisão sistemática foi realizada por Potts e Strusevich (2009), que sumarizaram os principais problemas e variantes, bem como métodos de solução propostos nos últimos 50 anos. Por outro lado, Allahverdi (2016) apresentou uma revisão de cerca de 300 trabalhos que lidaram com o problema de escalonamento contendo a restrição de não haver espera entre o término de uma tarefa e o início da próxima em cada máquina.

Por outro lado, Gupta e Ho (2000) consideraram um problema de duas máquinas paralelas idênticas para minimizar o tempo de fluxo total sujeito a um dado *makespan*. O algoritmo de otimização proposto pelos autores foi bastante eficiente, mesmo para a solução de problemas de grande porte. Os autores Salazar-Hornig e Medina (2013) utilizaram um algoritmo genético para resolver o problema de escalonamento em máquinas paralelas idênticas com tempos de configuração dependentes da sequência, para o qual a heurística do melhor vizinho foi introduzida como um algoritmo de melhoria que otimiza a alocação da tarefa resolvendo um problema de minimização de *makespan* independente em cada máquina. Ao introduzir o algoritmo de melhoramento, o aumento no desempenho do algoritmo genético foi significativo, melhorando em média as soluções obtidas com o algoritmo genético padrão em cerca de 8%.

Díaz-Cazañas, Paz-Martínez e Delgado-Sobrino (2019) implementaram um algoritmo para a programação integrada de produção e manutenção preventiva em máquinas paralelas idênticas. O algoritmo foi aplicado em uma oficina da indústria de plásticos, composta por 7 injetoras dispostas em paralelo. Para cada uma das máquinas, foram retirados os dados das variáveis tempo entre falhas e tempo de inatividade por falhas, expressos em horas (mais de 30 dados por máquina). Inicialmente, apresentaram a solução obtida pela heurística construtiva que constituiu a fase 1 do algoritmo. Logo após, apresentaram os resultados alcançados com a aplicação da fase 2, combinando uma heurística construtiva inspirada na regra *longest processing time* (LPT) e uma abordagem de recozimento simulado. O algoritmo proposto superou outras abordagens encontradas na literatura para resolver este tipo de problema, considerando possíveis diferenças nas datas de lançamento de vários produtos.

No que diz respeito aos problemas de escalonamento na área médico-hospitalar, a superlotação em unidades de pronto-socorro vem sendo apontada em várias partes do mundo, sendo um problema de interesse mundial para os gestores desses ambientes (RICHARDSON, 2006). Uma das formas mais eficazes para diminuir a superlotação tem sido aumentar a capacidade local e dos recursos, embora essa prática seja bastante lenta e dependente de políticas de governo em hospitais e unidades públicas. Por outro lado, pode-se melhorar a performance das unidades atuais a partir do gerenciamento otimizado das atividades, com a melhora do fluxo de pacientes, assim reduzindo as longas filas e possíveis atrasos nos atendimentos (SAGHAFIAN; AUSTIN; TRAUB, 2015).

Em Hoot et al. (2008) há um modelo de simulação discreta, que faz uso dos dados dos pacientes, para obter previsões de indicadores de performance da unidade de pronto-socorro, como o tempo médio de espera e quantidade esperada de pacientes em determinado período. Por outro lado, Kuo et al. (2016) desenvolveram um modelo de simulação para investigar o impacto que diferentes estratégias podem ocasionar dentro do pronto-socorro, como o aumento do local, a capacitação dos trabalhadores, uso de sistemas digitais e outros. Esses autores mediram o ganho que essas intervenções poderiam trazer sobre os indicadores de performance da unidade de pronto-socorro.

Queiroz et al. (2021) lidaram com a versão dinâmica do PEP em que os pacientes chegam no decorrer do dia e precisam ser alocados aos médicos para receberem tratamento. Esses autores propuseram heurísticas gulosas, uma baseada em filas de prioridade e a outra que usa uma busca em vizinhança variável, para determinar o escalonamento dos pacientes que estão em espera de atendimento. Diferente desses autores, neste trabalho busca-se propor um modelo de Programação Linear Inteira (PLI) para o PEP na sua versão estática, ou seja, onde todas as informações são conhecidas a priori. Um modelo de PLI retorna à solução ótima do problema, embora possa requerer tempo computacional exponencial no tamanho da instância.

Além disso, a solução de problemas de máquinas paralelas é de muita importância e tem grande aplicação também em outros tipos de empresas e indústrias. Devido às máquinas e recursos neste problema serem pessoas e estar ligado com a área de saúde, o atraso no atendimento pode trazer mais riscos e deveria ser evitado a todo custo. Os hospitais e unidades de pronto-socorro buscam, assim, a otimização do atendimento prestado aos pacientes para diminuir o tempo de espera (PUENTE et al., 2009).

3. DESCRIÇÃO DO PROBLEMA E DO MODELO

O problema de escalonamento de pacientes em unidades de pronto socorro considera um conjunto J de n pacientes, em que cada paciente j de J possui um tempo de liberação (*release date*) r_j , um nível de urgência w_j , um tempo esperado para a duração do seu tratamento p_j e um tempo d_j dentro do qual deveria iniciar o seu atendimento. O tempo que o paciente deveria esperar está relacionado com o tipo de urgência, sendo que pacientes com maior grau de urgência devem esperar menos. Quando um paciente inicia o recebimento de atendimento após o seu tempo de espera d_j , ocorre um atraso t_j , que

é dado pela diferença $t_j = d_j - s_j$, onde s_j é o momento que o atendimento do paciente j inicia. Os pacientes são atendidos por um conjunto de m médicos, todos com eficiência idêntica e sendo capazes de atender qualquer paciente. Um paciente recebe tratamento de somente um médico e somente uma vez, sendo a duração do atendimento dada por p_j . O objetivo é minimizar o atraso total ponderado, que é dado pela soma do produto $w_j T_j$ para todos os pacientes j em J .

O modelo de PLI para o PEP é descrito na função objetivo (1) e contém as restrições (2)-(7). O modelo considera as seguintes variáveis de decisão: T_{jk} , que mede o atraso do paciente j no médico k ; x_{ijk} , que indica se o paciente j recebe atendimento imediatamente após o paciente i no médico k ; e S_{ik} , que marca o início do atendimento do paciente j pelo médico k , para $i, j = 0, 1, \dots, n$ e $k=1, 2, \dots, m$. Considera-se que há um paciente 0, fictício, com $p_0 = 0$ e $r_0 = 0$, sendo o primeiro paciente associado a cada médico, facilitando na hora de escrever o modelo de PLI (ARENALES et al., 2007).

$$\text{Minimizar } \sum_{k=1}^m \sum_{j=0}^n w_j T_{jk} \quad (1)$$

Sujeito a:

$$\sum_{k=1}^m \sum_{j=0}^n x_{ijk} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{0jk} \leq 1, \quad k=1, \dots, m \quad (3)$$

$$\sum_{i=0, i \neq h}^n x_{ihk} = \sum_{j=0, j \neq h}^n x_{hjk}, \quad h=1, \dots, n; k=1, \dots, m \quad (4)$$

$$S_{jk} \geq S_{ik} - M + (p_i + M)x_{ijk}, \quad i=0, \dots, n; j=1, \dots, n; k=1, \dots, m \quad (5)$$

$$T_{ik} \geq S_{ik} - (r_i + d_i), \quad i=0, \dots, n; k=1, \dots, m \quad (6)$$

$$S_{ik} \geq r_i, x_{ijk} \in \{0, 1\}, T_{ik} \geq 0, \quad i=0, \dots, n; j=1, \dots, n; k=1, \dots, m \quad (7)$$

A função objetivo (1) representa a minimização do atraso total ponderado, que advém da soma dos atrasos para todos os pacientes multiplicados pelas suas respectivas urgências. As restrições (2) impõem que cada paciente j tenha um único paciente predecessor i em um único médico k . As restrições (3) garantem que cada médico k tenha um único paciente inicial j , que é atendido imediatamente após o paciente fictício 0.

As restrições (4) asseguram que cada paciente h , exceto o paciente 0, tenha um único paciente predecessor e sucessor imediato, sendo que o paciente fictício estabelece o início e o final da sequência de atendimento em um médico k . As restrições (5) asseguram que o início do atendimento do paciente j no médico k depende do início do paciente predecessor i no mesmo médico somado ao tempo de atendimento do paciente i , no caso do paciente i ser o predecessor imediato do paciente j no médico k ; caso contrário, a restrição é desativa pelo uso de um número M grande (ARENALES et al., 2007).

As restrições (6) impõem que o atraso do paciente i no médico k depende do início do atendimento do paciente subtraído da soma do seu horário de liberação mais o tempo que o paciente deveria esperar para receber o atendimento. As restrições (7) indicam o domínio das variáveis do modelo, sendo que S_{ik} e T_{ik} são variáveis contínuas não-negativas, enquanto x_{ijk} é uma variável binária.

O modelo de PLI deverá ser resolvido pelas bibliotecas do *Gurobi*, um resolvidor que implementa o algoritmo *branch-and-cut*. Esse algoritmo é usado para encontrar soluções ótimas de modelos de PLI e combina os métodos *branch-and-bound* e planos de corte. O modelo proposto deverá ser

analisado quanto ao número de instâncias que consegue resolver dentro de um dado tempo limite, observando ainda o *gap* e o número de instâncias resolvidas na otimalidade.

4. RESULTADOS COMPUTACIONAIS

Para testar a eficácia do modelo de PLI para o PEP, realizaram-se testes numéricos com instâncias propostas por Queiroz et al. (2021). Utilizaram-se as bibliotecas presentes no *Gurobi Optimizer* na sua versão 9.1 para resolver as instâncias do modelo. Todos os algoritmos foram implementados utilizando a linguagem de programação C++ e os testes foram realizados em um computador com a seguinte configuração: processador *Intel Xeon E3-1245 v5* de 3,5 GHz, 32 GB de RAM e sistema operacional *Linux Ubuntu 16.04.7 LTS*. O resolvidor foi usado com seus parâmetros padrões, limitando-se apenas o tempo de resolução de cada instância em 3600 segundos.

4.1. Instâncias

As instâncias utilizadas neste trabalho foram obtidas de Queiroz et al. (2020) e consideram o número de pacientes igual a 20, enquanto o número de médicos está no conjunto {2, 3, 5, 10}. Os dados de cada paciente foram gerados de forma aleatória usando uma distribuição uniforme.

O momento de liberação r_j dos pacientes foi obtido do intervalo $[0; 50,5 \frac{n}{m} \alpha]$, para α variando de 0,2 a 3. A urgência w_j do paciente corresponde a um valor inteiro no intervalo de 1 (menos urgente) a 5 (mais urgente), enquanto o tempo de atendimento p_j requerido pelo paciente foi obtido do intervalo $[30, 180]$. O tempo de espera d_j do paciente foi gerado observando a sua urgência e o momento de liberação, de forma que $d_j = r_j + 60(w_j - 1)$.

4.2. Resultados

Os resultados obtidos com a aplicação do *Gurobi* sobre o modelo de PLI estão na Tabela 1. Cada linha da tabela contém informações quanto ao nome da instância, onde *N20* significa que se tem $n = 20$ pacientes, *Mx* significa que se tem $m = x$ médicos, para $x = 2, 3, 5$ e *10*, e *Ry* aponta que o y -ésimo valor de α foi usado para definir o tempo de liberação dos pacientes, para 10 valores de α . Há também informações com relação ao número de pacientes, o número de médicos, o número de variáveis e restrições que o modelo de PLI apresenta para a referida instância, o valor da função objetivo, que é o atraso total ponderado, o *gap* em porcentagem, que é a distância entre a solução retornada pelo resolvidor e uma solução ótima da instância, de forma que o valor do *gap* igual a zero significa que foi obtida uma solução ótima, e o tempo computacional em segundos para resolver a instância.

Tabela 1: Resultados para as instâncias com 20 pacientes e 2, 3, 5 e 10 médicos (Fonte: Autor).

Instância	Quantidade de Pacientes	Quantidade de Médicos	Número de Variáveis	Número de Restrições	Função Objetivo	GAP (%)	Tempo (s)
N20_M2_R0	20	2	966	944	15373	90,0	3600,0
N20_M2_R1	20	2	966	944	8217	90,0	3600,0
N20_M2_R2	20	2	966	944	7488	90,0	3600,0
N20_M2_R3	20	2	966	944	5560	90,0	3600,0
N20_M2_R4	20	2	966	944	11996	90,0	3600,0
N20_M2_R5	20	2	966	944	3780	80,0	3600,0
N20_M2_R6	20	2	966	944	697	60,0	3600,0

N20_M2_R7	20	2	966	944	55123	80,0	3600,0
N20_M2_R8	20	2	966	944	811	0,0	53,1
N20_M2_R9	20	2	966	944	0	0,0	1,3
MÉDIA_M2	-	-	-	-	-	67,0	2885,4
N20_M3_R0	20	3	1449	1406	7668	80,0	3600,0
N20_M3_R1	20	3	1449	1406	5236	90,0	3600,0
N20_M3_R2	20	3	1449	1406	4545	90,0	3600,0
N20_M3_R3	20	3	1449	1406	2928	90,0	3600,0
N20_M3_R4	20	3	1449	1406	0	0,0	42,5
N20_M3_R5	20	3	1449	1406	271	100,0	3600,0
N20_M3_R6	20	3	1449	1406	4126	70,0	3600,0
N20_M3_R7	20	3	1449	1406	778	10,0	3600,0
N20_M3_R8	20	3	1449	1406	38	0,0	114,6
N20_M3_R9	20	3	1449	1406	0	0,0	0,6
MÉDIA_M3	-	-	-	-	-	53,0	2535,8
N20_M5_R0	20	5	2415	2330	769	100,0	3600,0
N20_M5_R1	20	5	2415	2330	1095	100,0	3600,0
N20_M5_R2	20	5	2415	2330	192	50,0	3600,0
N20_M5_R3	20	5	2415	2330	116	100,0	3600,0
N20_M5_R4	20	5	2415	2330	0	0,0	6,8
N20_M5_R5	20	5	2415	2330	0	0,0	1,8
N20_M5_R6	20	5	2415	2330	0	0,0	2,7
N20_M5_R7	20	5	2415	2330	0	0,0	4,9
N20_M5_R8	20	5	2415	2330	0	0,0	1,3
N20_M5_R9	20	5	2415	2330	0	0,0	1,7
MÉDIA_M5	-	-	-	-	-	35,0	1441,9
N20_M10_R0	20	10	4830	4640	0	0,0	0,5
N20_M10_R1	20	10	4830	4640	0	0,0	0,5
N20_M10_R2	20	10	4830	4640	0	0,0	0,4
N20_M10_R3	20	10	4830	4640	0	0,0	0,3
N20_M10_R4	20	10	4830	4640	0	0,0	0,4
N20_M10_R5	20	10	4830	4640	0	0,0	0,1
N20_M10_R6	20	10	4830	4640	0	0,0	0,1
N20_M10_R7	20	10	4830	4640	0	0,0	0,2
N20_M10_R8	20	10	4830	4640	0	0,0	0,2
N20_M10_R9	20	10	4830	4640	0	0,0	0,1
MÉDIA_M10	-	-	-	-	-	0,0	0,3

Nos resultados da Tabela 1, observa-se que para as instâncias com a quantidade de 2 médicos, o resolvidor conseguiu soluções cujo gap médio foi de 67% e o tempo médio de 2885,4 segundos. O resolvidor atingiu o tempo máximo de 3600 segundos para 8 dessas 10 instâncias. A quantidade

presente de variáveis foi de 966 e a de restrições de 944. Por outro lado, para duas instâncias foi possível obter a solução ótima, com gap igual a zero, e tempo máximo de 53,1 segundos.

Para as instâncias com a disponibilidade de 3 médicos, as soluções apresentam um valor de função objetivo relativamente menor, inclusive o resolvidor apontou um gap de 53%, uma diferença de 14% se comparado à média anterior, e 2535,8 segundos no tempo médio gasto com a resolução de cada instância. A quantidade presente de variáveis foi de 1449 e a de restrições de 1406. O resolvidor conseguiu retornar uma solução ótima para 3 instâncias, com gap igual a zero, e tempo máximo de 114,6 segundos.

Já para as instâncias com a quantidade de 5 médicos, o resolvidor conseguiu soluções ainda melhores em termos de atraso total ponderado, apresentando um gap médio de 35%, que é uma diferença de 19% se comparado à média anterior com 3 médicos. O tempo computacional médio foi de 1441,9 segundos, enquanto o modelo teve um número de variáveis igual a 2415 e o de restrições igual a 2330, por instância. Dessa vez, o resolvidor conseguiu retornar uma solução ótima para 6 instâncias, com o tempo computacional variando entre 1,3 e 6,8 segundos.

Finalmente, para as instâncias com 10 médicos, o resolvidor conseguiu obter a solução ótima para todas elas, com gap médio de 0,0% e uma diferença de 35% se comparado à média do gap das instâncias com 5 médicos. O tempo gasto para a otimização mínimo foi de 0,1 segundos e o máximo de 0,5 segundos. A quantidade presente de variáveis foi de 4830 e a de restrições de 4640.

De forma geral, os resultados mostram que o modelo permite encontrar soluções ótimas para as instâncias, mesmo naquelas com quantidade mínima de 2 médicos, que são mais difíceis de resolver, aumentando essa quantidade conforme há a disponibilidade de mais médicos. Assim, quanto maior a quantidade médicos, melhores os resultados obtidos, o que é esperado, considerando que quanto mais médicos maior a capacidade de atender pacientes. No demais casos, o gap e o tempo computacional tendem a crescer conforme a quantidade de médicos diminui.

5. CONCLUSÕES

O estudo de modelos para a solução de problemas da vida real é cada dia mais importante, principalmente com a diminuição de recursos devido a diversos fatores, principalmente o financeiro. Este trabalho procura encontrar soluções para o problema de escalonamento de pacientes em unidades de pronto-socorro, com a intenção de minimizar o tempo total dos atrasos ponderados pela urgência dos pacientes e evitar que pacientes passem mais tempo no hospital do que o necessário e consequentemente há um risco maior de se exporem a doenças. A redução do atraso também ajuda a melhorar o fluxo de pacientes através do pronto-socorro, melhorando os índices do serviço e trazendo maior satisfação aos pacientes.

Com o modelo em estudo foi possível obter soluções ótimas para instâncias com diferentes quantidades de médicos, dentro do tempo de processamento estipulado. O modelo permitiu que as instâncias em que estavam disponíveis 10 médicos tivessem os melhores resultados, justamente por possuírem uma maior flexibilidade para atender os 20 pacientes pelos médicos. Em geral, quanto menor a disponibilidade de médicos, mais difícil se torna para o resolvidor encontrar uma solução ótima, apresentando resultados com gaps elevados, assim como precisando de maior esforço computacional.

Como forma de dar continuidade a este trabalho em pesquisas futuras, considera-se estudar a capacidade do modelo em resolver de forma ótima instâncias com mais pacientes, por exemplo. Outra linha de investigação contará com o desenvolvimento de métodos heurísticos que, embora não garantam uma solução ótima, podem retornar boas soluções em um rápido tempo computacional, que servirão de base para a comparação com o modelo de PLI.

6. REFERÊNCIAS

- ALLAHVERDI, A. A survey of scheduling problems with no-wait in process. *European Journal of Operational Research*, v. 255, n. 3, p. 665-686, 2016.
- ARENALES M.; ARMENTANO V.; MORABITO R.; YANASSE H. *Pesquisa Operacional*. Rio de Janeiro: Campus, 2007.
- DÍAZ-CAZAÑAS, R.; PAZ-MARTÍNEZ, E. M.; DELGADO-SOBRINO, D. R. Algoritmo para la programación integrada producción - mantenimiento preventivo en máquinas paralelas idénticas. *Ingeniería Industrial*, v. 40, n. 3, p. 260-271, 2019.
- DI SOMMA, S.; PALADINO, L.; VAUGHAN, L.; LALLE, I.; MAGRINI, L.; MAGNANTI, M. Overcrowding in emergency department: an international issue. *Internal and Emergency Medicine*, v. 10, n. 2, p. 171-175, 2015.
- GUPTA, J. N. D.; HO, J. C. Minimizing Flowtime Subject to Optimal Makespan on Two Identical Parallel Machines. *Pesquisa Operacional*, v. 20, n. 1, p. 5-17, 2000.
- HOOT, N. R.; LEBLANC, L. J.; JONES, I.; LEVIN, S. R.; ZHOU, C.; GADD, C. S.; ARONSKY, D. Forecasting emergency department crowding: a discrete event simulation. *Annals of Emergency Medicine*, v. 52, n. 2, p. 116-125, 2008.
- KRAMER, A.; DELL'AMICO, M.; IORI, M. Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *European Journal of Operational Research*, v. 275, p. 67-79, 2019.
- KUO, Y.-H.; RADO, O.; LUPIA, B.; LEUNG, J. M.; GRAHAM, C. A. Improving the efficiency of a hospital emergency department: a simulation study with indirectly imputed service-time distributions. *Flexible Services and Manufacturing Journal*, v. 28, n. 1-2, p. 120-147, 2016.
- PINEDO, M. L. *Scheduling: Theory, Algorithms, and Systems*. Springer, 2008.
- POTTS, C. N.; STRUSEVICH, V. A. Fifty Years of Scheduling: A Survey of Milestones. *Journal of the Operational Research Society*, v. 60, pp. s41-s68, 2009.
- PUENTE, J.; GÓMEZ, A.; FERNÁNDEZ, I.; PRIORE, P. Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms. *Computers & Industrial Engineering*, v. 56, n. 4, p. 1232-1242, 2009.
- QUEIROZ, T. A.; IORI, M.; KRAMER, A.; KUO, Y.-H. Scheduling of patients in emergency departments with a variable neighborhood search. *8th International Conference on Variable Neighborhood Search*, Abu Dhabi, United Arab Emirates, p. 138-151, 2021.
- RICHARDSON, D. B. Increase in patient mortality at 10 days associated with emergency department overcrowding. *Medical Journal of Australia*, v. 184, n. 5, p. 213-216, 2006.
- SAGHAFIAN, S.; AUSTIN, G.; TRAUB, S. J. Operations research/management contributions to emergency department patient flow optimization: Review and research prospects. *IIE Transactions on Healthcare Systems Engineering*, v. 5, n. 2, p. 101-123, 2015.
- SALAZAR-HORNIG, E.; MEDINA, S. J. C. Minimización del makespan en máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia utilizando un algoritmo genético. *Ingeniería, Investigación y Tecnología*, v. 14, p. 43-51, 2013.

7. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.