LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

Comparação de Algoritmos Genéticos Multiobjetivo NSGA-II e RNSGA-II para Otimização de Sequenciamento em Máquinas Paralelas Não-Relacionadas

Guilherme Lopes Weis
Universidade Federal de Santa Maria - Centro de Tecnologia - UFSM/CT
guilherme.weis@ebserh.gov.br

Angelita Pezzi Pasqualon Bridi
Universidade Federal de Santa Maria - Centro de Tecnologia - UFSM/CT
angelitabridi@gmail.com

Suzianne Miranda Tormann
Universidade Federal de Santa Maria - Centro de Tecnologia - UFSM/CT
suzianne.tormann@redes.ufsm.br

Felipe Martins Muller
PPGEP - UFSM e CTI Renato Archer
felipe@ufsm.br e felipe.muller@cti.gov.br

RESUMO

Este estudo apresenta uma abordagem para validar e comparar os algoritmos genéticos (AG) multiobjetivo NSGA-II e RNSGA-II aplicados ao sequenciamento em máquinas paralelas não relacionadas. O objetivo é otimizar simultaneamente o *makespan* e o consumo total de energia (TEC), gerando a fronteira de Pareto e comparando as soluções obtidas entre os AGs e com as de objetivo único do GNU *Linear Programming Kit* (GLPK). Os resultados demonstram que o NSGA-II supera consistentemente o RNSGA-II em diversidade de soluções e eficiência na minimização dos objetivos. O NSGA-II apresentou 88,33% das soluções não dominadas. Além disso, o NSGA-II teve melhores valores de hipervolume em 70% das instâncias analisadas. Em termos de *makespan*, o NSGA-II obteve melhores resultados em 92% das instâncias e foi superior em todas as avaliações de TEC. O tempo de execução do NSGA-II também foi consistentemente menor em comparação ao RNSGA-II.

PALAVRAS CHAVE. Multiobjetivo; Algoritmo Genético; Máquinas Não-Relacionadas.

Tópicos: PO na Indústria.

ABSTRACT

This study presents an approach to validate and compare the multi-objective genetic algorithms (GA) NSGA-II and RNSGA-II applied to sequencing in unrelated parallel machines. The objective is to simultaneously optimize the makespan and the total energy consumption (TEC), generating the Pareto frontier and comparing the solutions obtained between the GAs and with the single-objective solutions from the GNU Linear Programming Kit (GLPK). The results demonstrate that NSGA-II consistently outperforms RNSGA-II in solution diversity and efficiency in minimizing the objectives. NSGA-II presented 88.33% of the non-dominated solutions. Additionally, NSGA-II had better hypervolume values in 70% of the analyzed instances. In terms of makespan, NSGA-II obtained better results in 92% of the instances and was superior in all TEC evaluations. The execution time of NSGA-II was also consistently lower compared to RNSGA-II.

KEYWORDS. Multiobjective, Genetic Algorithm, Unrelated Machines.

Paper topics: OR in Industry.

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

1. Introdução

A crescente priorização da busca por uma maior eficiência energética, é impulsionada pela escassez de recursos e pelas preocupações ambientais. Desta forma, está se tornando cada vez mais relevante em uma economia global, onde a sustentabilidade ambiental é uma prioridade, conforme destacado por [Giret et al., 2015].

Como indicado por [Meloni, 2020], indútrias consomem 36% de energia de forma global, já no território brasileiro, a indústria utiliza quase 40% da eletricidade do país, segundo informações da [EPE, 2020]. Considerando esse cenário, a diminuição das despesas com energia elétrica se torna fundamental para enfrentar o aumento dos custos de produção na indústria do Brasil, conforme apontado por [Ecom, 2022].

Este imperativo de eficiência energética não apenas ressalta a importância da alocação eficiente de recursos limitados, mas também destaca a necessidade de resolver problemas de sequenciamento de produção. Estes últimos são cruciais para o funcionamento e a sobrevivência do setor de manufatura, representando uma ferramenta fundamental para otimizar o uso dos recursos disponíveis. Além disso, tais técnicas podem ser aplicadas em diversos setores, incluindo a produção de objetos de plástico, usinagem, entre outros [Lacerda et al., 2016].

A importância desse desafio na indústria reside no seu impacto na distribuição eficiente de recursos [Pinedo, 2012]. O artigo busca identificar o problema e oferecer soluções inovadoras para otimização nesse contexto. Por outro lado, na Pesquisa Operacional, a eficiência energética é crucial para maximizar os recursos disponíveis. Problemas de programação multiobjetivo fornecem alternativas que auxiliam na alocação eficiente de recursos, com ferramentas para avaliar fatores relevantes [de Souza Brandão and Cortez, 2008].

Sendo assim, a pesquisa se concentra na otimização, minimizando simultaneamente o *makespan* e o consumo total de energia (TEC) em um sistema de produção com n tarefas independentes alocadas em m máquinas paralelas não relacionadas, conforme Muller et al. [2023], com a adição de fatores de potência para cada máquina, que variam em três modos de trabalho. Esses modos alteram a velocidade de realização das tarefas e, consequentemente, o consumo de energia, tornando o problema ainda mais complexo e justificando o uso de métodos computacionais. Para atingir esse objetivo, utiliza-se o framework de código aberto Pymoo [Blank and Deb, 2020] para obter as soluções dos Algoritmos Genéticos (AG) e o *GNU Linear Programming Kit* (GLPK) para fornecer uma análise referenciada em comparação com as soluções obtidas pelos algoritmos NSGA-II e RNSGA-II.

Este artigo está organizado da seguinte forma. Na Seção 2, apresenta-se a Metaheurística Proposta, os algoritmos NSGA-II e R-NSGA-II, e a Métrica de Desempenho Hipervolume. Na Seção 3, encontram-se a Metodologia, Definição do Problema, Pseudocódigo e o Modelo Matemático. Na Seção 4, encontram-se os Resultados Computacionais obtidos, a Geração de Instâncias e um Exemplo de Indivíduo. A Seção 5 trata do modelo proposto para o NSGA-II e RNSGA-II. E por fim, na Seção 6, estão as Considerações Finais e Trabalhos Futuros.

2. Metaheurística Proposta

Nesta seção, serão apresentados os algoritmos genéticos NSGA-II e RNSGA-II utilizados para resolver o problema multiobjeivo de minimização do *makespan* e do consumo total de energia (TEC).

2.1. NSGA-II e RNSGA-II, Métrica de Desempenho Hipervolume

Em problemas de otimização multiobjetivo, que envolvem duas ou mais funções objetivo, a escolha entre métodos exatos e heurísticos depende da natureza do problema. Métodos heurísticos são frequentemente utilizados, pois buscam soluções próximas do ótimo que compõem a Fronteira de Pareto. No contexto dos métodos heurísticos, uma aproximação da fronteira é obtida.

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

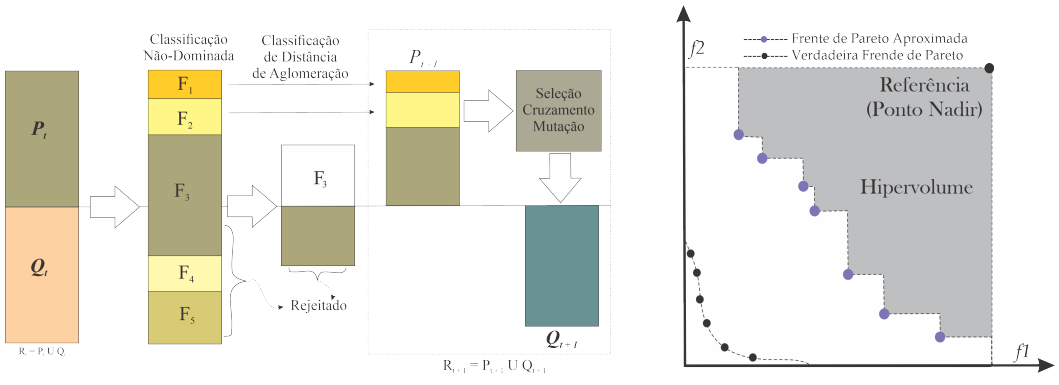
Em vista disso, há um avanço de algoritmos na literatura, como o *MultiObjective Genetic Algorithm* (MOGA), o *Niched-Pareto Genetic Algorithm* (NPGA), o *Non-dominated Sorting Genetic Algorithm* (NSGA-II), entre outros [Miloca et al., 2012].

Conforme os pesquisadores [Verma et al., 2021] o processo do NSGA-II além de ser computacionalmente rápido, utiliza o operador de distância de aglomeração para o mecanismo de preservação da diversidade, como mostra a Figura 1(a). Seu método é embasado em quatro princípios fundamentais: Seleção Não-Dominada, Operador de Elite, Distância de Aglomeração e Operador de Seleção.

A Figura 1(a), demonstra o funcionamento do algoritmo NSGA-II, que inicia com uma população P_t de tamanho N. A partir dessa população, é gerada uma nova população descendente, Q_t, de mesmo tamanho, por meio de processos de seleção, cruzamento e mutação. As soluções não dominadas (ND) são classificadas em frentes de Pareto, $F1$, $F2$, $F3$, etc. Na fase subsequente, essas frentes de Pareto são avaliadas, e o ponto de corte é feito na frente de Pareto cuja população excede o tamanho da população inicial. A seleção das soluções dentro dessa frente de Pareto é realizada utilizando o método de distância de aglomeração (*crowding distance*), formando a próxima população P_{t+1}, que, por sua vez, passa pelos processos de seleção, cruzamento e mutação, resultando na população descendente Q_{t+1}.

De modo semelhante, o algoritmo RNSGA-II utiliza uma estrutura geral similar ao NSGA-II, com uma modificação na seleção de sobreviventes. O objetivo principal do RNSGA-II é gerar pontos concentrados na região frontal da Fronteira de Pareto, levando em consideração as preferências do tomador de decisão (*Decision Maker*). Nessa situação, o algoritmo RNSGA-II encontra desafios em obter uma distribuição satisfatória de pontos que se aproximem de todas as partes da Fronteira de Pareto [Filatovas et al., 2017].

Um dos principais desafios associados aos resultados derivados de técnicas multiobjetivos diz respeito às avaliações das fronteiras de pareto. Em contraste com questões mono-objetivas, em que as comparações entre as soluções podem ser feitas diretamente com base nos valores da função objetivo, esse método não é aplicável a cenários multiobjetivos. Isso se deve ao fato de que o resultado não é um valor único, mas sim um conjunto de soluções [Costa et al., 2015].



(a) *Processo do NSGA-II.* (b) *Exemplo de Hipervolume.*

Figura 1: Processo do NSGA-II e Exemplo de Hipervolume.

Em [Zitzler, 1999], é demonstrada a métrica de performance chamada Hipervolume, que mede o volume (ou área, no caso de problemas com dois objetivos) aproximadamente debaixo da curva formada pelas soluções não dominadas em relação a um ponto de referência (ou seja, um ponto dominado por todos os pontos da fronteira). A Figura 1(b), apresenta o hipervolume para a

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

minimização de um problema multiobjetivo.

3. Metodologia

3.1. Definição do Problema

Para formalizar o problema abordado, assume-se um conjunto de *n* tarefas independentes a *m* máquinas não-relacionadas (*unrelated*), além disto, considera-se que todas as tarefas estão disponíveis para iniciar suas execuções ao mesmo tempo, e uma máquina com mais de uma tarefa alocada a ela, deve executá-las uma após a outra, em alguma sequência, sem interrupção, ou seja, a tarefa que iniciar sua execução em uma máquina deve permanecer nela até o seu final.

Considerando que cada tarefa pode ser realizada por, no máximo, uma máquina, e que cada máquina é capaz de processar apenas uma tarefa por sua vez, sem interrupção, são estabelecidos os parâmetros e as variáveis necessárias na descrição e modelagem do problema.

Parâmetros:

M: Conjunto de máquinas, $j = 0, 1, ..., m$.

N: Conjunto de tarefas, $i = 1, 2, ..., n$.

L: Conjunto de tarefas, $l = 1, 2 \ \ e \ \ 3$.

t_{ij}: Tempo necessário para a máquina i completar a tarefa j.

π_i: Potência da máquina i em modo normal de operação (kW).

v_l : Fator de velocidade em operação normal $l \in L$.

λ_l: Fator de potência em velocidade normal $l \in L$.

Variáveis de Decisão:

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } \ j \ \text{ é atribuída à máquina } \ i, \\ 0 & \text{caso contrário.} \end{cases}$$

C_j: Tempo de conclusão da tarefa j na máquina i.

O_i: Tempo total na máquina i.

C_{max}: Makespan do conjunto de máquinas.

TEC: Energia total consumida por todas as máquinas (kWh).

Relação entre os fatores:

$$v_l \text{ e } \lambda_l = \begin{cases} v_1 = 1 \text{ e } \lambda_1 = 1, & \text{velocidade normal de operação,} \\ v_2 = 0,8 \text{ e } \lambda_2 = 0,6, & \text{velocidade menor, menos consumo de energia,} \\ v_3 = 1,2 \text{ e } \lambda_3 = 1,5, & \text{velocidade maior, mais consumo de energia.} \end{cases}$$

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

3.2. Pseudocódigo

Algoritmo 1: PSEUDOCÓDIGO PARA NSGA-II E RNSGA-II

Entrada: Inicilaiza *População*
Gere *N* soluções aleatórias e insira em *População*
para i ← 1 até número máximo de gerações **faça**
início
    Gere *PopulaçãoProle* de tamanho *N*
    Selecione os *Pais de População*
    Crie a *Prole de Pais*
    Aplique mutação em *Prole*
    Combine *População* e *PopulaçãoProle* em *População Atual*
    para *todos indivíduos em População Atual* **faça**
        Atribua *rank* baseado no Pareto do *Fast Non Dominated Stort*
    fim
    Gere conjunto das soluções não-dominadas no Pareeto
    Calcule o *Crowding Distance* (NSGA-II)
    Calcule o Reference point Based (RNSGA-II)
    Selecione *N* melhores soluções começando no melhor *front*
fim
retorna *os resultados ([makespan, consumo total de energia])*

3.3. Modelo Matemático

Esse modelo foi baseado no problema resolvido por [Cota et al., 2021]. Aqui, apresentam-se somente os parâmetros mais relevantes e que se diferem do original. Problema de alocação de tarefas em máquinas paralelas não-relacionadas com o objetivo de minimizar o *Makespan* e o Consumo Total de Energia - TEC.

Objetivos:

$$\text{Min } f1 = C_{\max}$$
$$\text{Min } f2 = TEC$$

Restrições:

$$\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{q}(\frac{t_{ij}}{v_l})x_{ijl} = O_i \quad \forall i \in M \tag{1}$$

$$C_{max} \geq O_i \qquad \forall i \epsilon M \tag{2}$$

$$TEC \geq \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q}(\lambda_l \times \frac{\pi_i}{60} \times \frac{t_ij}{v_l})x_{ijl} \tag{3}$$

$$x_{ijl} \in \{0,1\} \quad \forall j \in N, \forall i \in M, \forall l \in L. \tag{4}$$

A restrição (1) calcula o tempo total de processamento em cada máquina, considerando todas as tarefas nela alocadas e ajustando o tempo de processamento pela velocidade da máquina. A restrição (2) define o valor de $C_{\max}$, que é o maior tempo total de processamento entre todas as máquinas. O cálculo do TEC é realizado pela restrição (3). Para realizar este cálculo, é necessário considerar os tempos de processamento $\left(\frac{p_{ik}}{v_l}\right)$, a potência de cada máquina em velocidade normal de operação (dada por π_i), e os fatores multiplicadores λ_l e ν_l, ajustado pelo fator 60 min/h para fornecer o TEC em quilowatt-hora. Por fim, a restrição (4) define a variável binária.

3.4. Frameworks e parâmetros de ajustes dos algoritmos NSGA-II e RNSGA-II

Foi utilizada um framework de código aberto, escrito em Python chamada Pymoo, que fornece uma base de algoritmos de otimização para problemas mono e multiobjetivo, bem como outros recursos relacionados à otimização multiobjetivo, como tomada de decisão e visualização [Blank and Deb, 2020]. Os parâmetros escolhidos utiizados nesse trabalho foram:

Seleção: ranking de dominância e distância de aglomeração

Cruzamento: Two point crossover

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

Mutação: Polynomial Mutation, (prob. global para gene mutar de 50%)
Reparo: todas as soluções geradas estão dentro do domínio permitido.
Término: 50 gerações

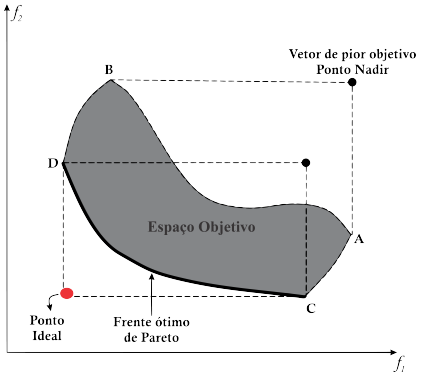
Ainda, foi aplicada uma ferramenta para solução exata GLPK (*GNU Linear Programming Kit*) que é utilizado para problemas de programação linear e MILP (*Mixed Integer Linear Program*) [Makhorin, 2014], empregando o método Simplex, GLPK *Simplex Optimizer*, v4.65. Esses algoritmos foram utilizados e adaptados pelos autores para atender as necessidades específicas do problema.

3.5. Fronteira de Pareto, Ponto *Ideal* e Ponto *Nadir*

Criada em 1896 pelo economista italiano Vilfredo Pareto, a Fronteira de Pareto pode ajudar a melhorar as decisões ao resolver problemas [Kádár, 2022]. A Fronteira de Pareto analisa o conjunto de soluções em relação à dominância entre elas. Podendo as soluções serem classificadas como dominadas, que é quando uma solução é pior que outra em todos os objetivos simultaneamente; e não dominadas, que é quando uma solução não pode ser considerada pior que outra em todos os objetivos simultaneamente [Cid et al., 2023].

O ponto *ideal* é obtido pela minimização de cada objetivo individualmente. Segundo [Ehrgott, 2005] e [Coello Coello et al., 2010], este ponto não é factível, pois, se fosse, significaria que não haveria conflito entre os objetivos, tornando desnecessário o conceito de Otimalidade de Pareto. Portanto, minimizar um objetivo isoladamente pode não ser a melhor abordagem. No entanto, o ponto ideal serve como um limite inferior para o conjunto de pontos não dominados.

O ponto *Nadir* é o vetor que indica os valores mais baixos de cada objetivo ao otimizar os demais, estabelecendo um limite superior para o conjunto de pontos não dominados, a combinação no espaço objetivo do problema mostra as soluções ótimas de Pareto. A Figura 2, apresenta os a fronteira de Pareto, o ponto ideal e o ponto Nadir para uma melhor compreensão do processo multiobjetivo.



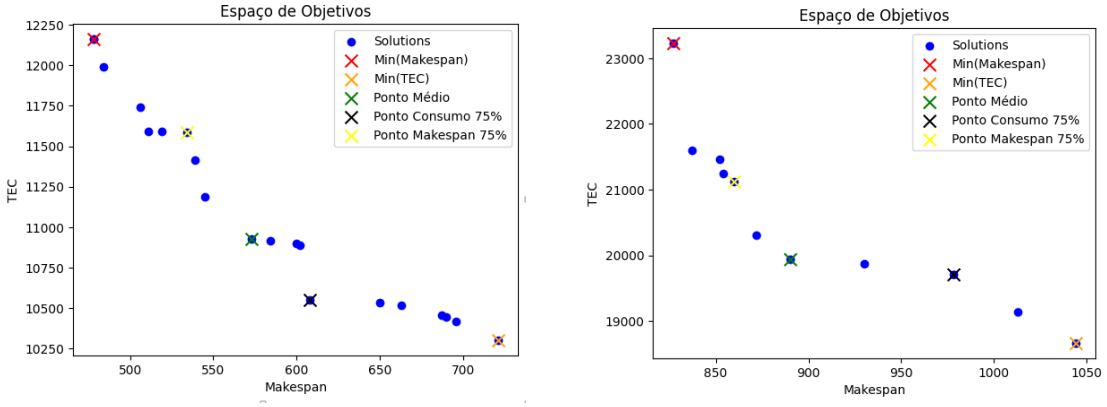
Fonte: Adaptado de [Amouzgar et al., 2020].

Figura 2: Fronteira de Pareto, Ponto *Ideal* e Ponto *Nadir*

3.6. Distribuição de Soluções Não Dominadas (ND) na Fronteira de Pareto

Neste estudo, utilizamos a representação visual para uma melhor compreensão do espaço de objetivos para analisar o desempenho dos algoritmos NSGA-II e RNSGA-II, das soluções que foram encontradas. A Figura 3, apresenta dois gráficos de dispersão que representam o comportamento das soluções não dominadas (ND) em termos de dois objetivos conflitantes: *Makespan* e o Consumo Total de Energia (TEC), formando uma Fronteira de Pareto.

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br



(a) *NSGA-II* (b) *RNSGA-II*

Figura 3: Distribuição das soluções não dominadas no espaço de objetivos.

Foram definidos alguns pesos para orientar o tomador de decisão, descritos da seguinte forma: cada ponto azul representa uma solução potencial no espaço objetivo; o ponto vermelho indica a solução com o menor *makespan* entre todas as soluções não dominadas; o ponto amarelo indica a solução com o menor TEC; o ponto verde representa uma solução que pondera igualmente (50%) os objetivos; o ponto preto corresponde à solução no percentil 75% do TEC; e o ponto amarelo citrus é também a solução no percentil 75% do TEC.

4. Resultados Computacionais

Os experimentos computacionais foram escritos e executados em Linguagem Python, utilizando Jupyter, em um computador com Sistema Operacional Windowns 10 Pro, Processador AMD Ryzen 5 3400G Radeon, 3700Mhz, 4 núcleos e 8 processadores lógicos.

4.1. Geração das Instâncias

As instâncias, conforme tabela 1, foram geradas de forma aleatória, com distribuição uniforme, baseados nos intervalos estabelecidos em [Cota et al., 2021], para os tempos das tarefas de [1,99] e para as potências de cada máquina de [40,200]. O tamanho da população aumenta conforme aumenta a complexidade do problema, sendo 100, 200, 300 e 400 indivíduos.

Tabela 1: Distribuição de tarefas nas máquinas.

Número de Máquinas (M)	Número de Tarefas (N)
5	20, 50, 75 e 100
10	50, 100, 150 e 200
15	75, 150, 225 e 300
20	100, 200, 300 e 400

4.2. Exemplo de indivíduo gerado para esse algoritmo genético

Apresentamos na Tabela 2, um exemplo de indivíduo gerado por este algoritmo genético, ilustrando um cenário com 2 máquinas e 10 tarefas.

O modo (λ) representa um fator de potência que afeta o consumo de energia elétrica, enquanto as velocidades (v) influenciam o tempo de execução das tarefas. Em velocidades mais altas, como 1,2; o *Makespan* pode ser menor, indicando que as tarefas são concluídas mais rapidamente. No entanto, isso geralmente resulta em um maior consumo de energia. Em velocidades mais baixas, como 0.8; é comum que o consumo total de energia (TEC) seja reduzido, pois está associado

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

a um fator de potência de 0.6; indicando uma operação mais eficiente em termos de energia. Por outro lado, o *Makespan* tende a ser maior, significando que as tarefas demoram mais para serem concluídas.

Tabela 2: Exemplo de geração de indivíduo

Modo	**Velocidade**	**Máquinas**	**M**	**N**	**Tempo das Máquinas (s)**	**Potência π**
1	1	M0	2	10	44,76,85,80,16,40,73,05,58,70	194
1	1	M1	2	10	79,52,29,92,58,61,22,82,63,54	198

A Tabela 3 apresenta os tempos de execução das tarefas para duas máquinas (Máquina 0 e Máquina 1) e os consumos de energia associados a cada tarefa, valores que foram gerados aleatoriamente para ilustrar o problema. Cada tarefa é atribuída aleatoriamente a uma das duas máquinas. Por exemplo, a tarefa 1 foi alocada na Máquina 0, assim como as tarefas 3, 5, 7, 9 e 10. As demais tarefas (2, 4, 6, 8) foram alocadas na Máquina 1.

Além disso, são fornecidos *makespan*, os consumos de energia individuais para cada tarefa, o somatório do consumo de energia para a Máquina 0 e para a Máquina 1, bem como o consumo total de energia, que é a soma desses dois valores.

- Makespan: 346 na Máquina 0
- TEC (Total Energy Consumption): 2065,83 é o consumo total de energia.

Tabela 3: Processamento das tarefas nas máquinas.

M0	**M1**	**t M0**	**t M1**	**t M0/v**	**t M1/v**	$\lambda * (\pi\ /60) * (t\ M0/v)$	$\lambda * (\pi\ /60) * (t\ M1/v)$
1	0	44	0	44	0	142,27	0
0	1	0	52	0	52	0,00	171,6
1	0	85	0	85	0	274,83	0
0	1	0	92	0	92	0,00	303,6
1	0	16	0	16	0	51,73	0
0	1	0	61	0	61	0,00	201,3
1	0	73	0	73	0	236,03	0
0	1	0	82	0	82	0,00	270,6
1	0	58	0	58	0	187,53	0
1	0	70	0	70	0	226,33	0
	Makespan	**346**	287			1118,73	947,1
						TEC	**2065,83**

5. Resultados do modelo proposto para o NSGA-II e RNSGA-II

Compararam-se dois algoritmos genéticos utilizando as métricas de soluções Não Dominadas, Hipervolume e Ponto Médio, sendo o último analisado pela mediana dos valores. Além disso, comparou-se o desempenho desses algoritmos ao método exato GLPK em termos de tempo de execução e pontos lexicográficos. Na Figura 4, o gráfico das soluções Não Dominadas (ND) destaca o desempenho dos algoritmos NSGA-II e RNSGA-II. O NSGA-II apresentou uma representatividade de 88,33%, enquanto o RNSGA-II obteve 5,00%, com 6,67% das soluções ND empatadas.

Ao analisar a Figura 5, observa-se o comportamento dos algoritmos NSGA-II e RNSGA-II em relação ao hipervolume, uma métrica utilizada para avaliar a qualidade das soluções não dominadas. O NSGA-II apresentou valores de hipervolume maiores ou iguais ao RNSGA-II em 42 (70%) instâncias analisadas. Esses resultados indicam que, na maioria significativa dos casos,

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

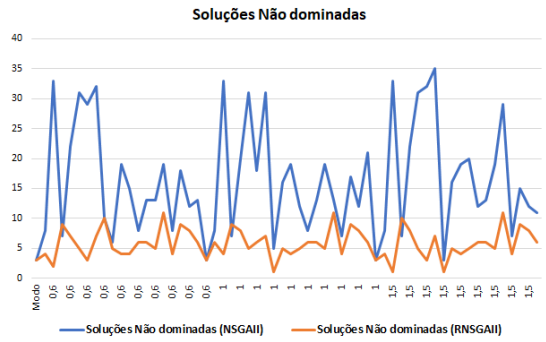


Figura 4: Soluções Não Dominadas(ND).

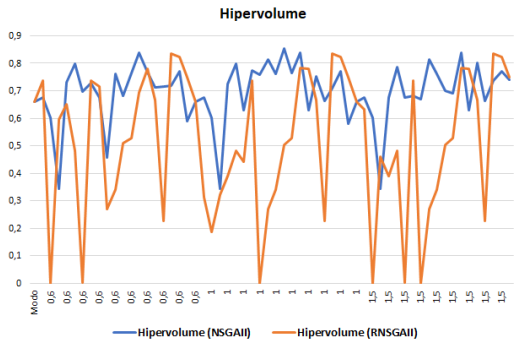
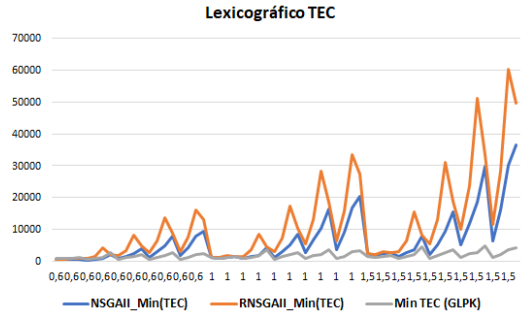


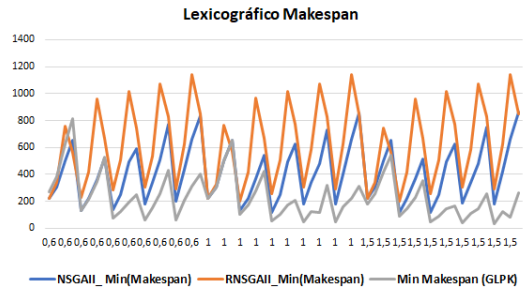
Figura 5: Hipervolume do NSGA-II e RNSGA-II.

o NSGA-II produziu soluções de maior qualidade em termos de hipervolume em comparação ao RNSGA-II. Isso sugere que o NSGA-II é mais eficaz na exploração e otimização do espaço das soluções não dominadas para os critérios avaliados.

A Figura 6 mostra o desempenho dos algoritmos NSGA-II e RNSGA-II em relação ao *makespan* e ao TEC, comparados ao método exato. A análise revela que os algoritmos genéticos conseguem obter resultados muito próximos aos valores exatos para instâncias menores. No entanto, à medida que a complexidade aumenta, ambos os algoritmos genéticos apresentam piora nos resultados. Notoriamente, o NSGA-II apresenta soluções melhores do que o RNSGA-II. Os dados mostram que 55 instâncias (92%) tiveram resultados melhores (menores ou iguais) aos obtidos pelo RNSGA-II no critério *lexicográfico de makespan*. Para o critério *lexicográfico de TEC*, o NSGA-II apresentou valores melhores (menores) em 100% das situações avaliadas, indicando sua superioridade na otimização desses critérios.



(a) *TEC*



(b) *Makespan*

Figura 6: Desempenho nos lexicográficos TEC e *Makespan*.

O tempo de execução do método exato (GLPK) é o tempo total necessário para obter os dois objetivos. Assim, comparou-se o tempo de execução entre as soluções obtidas a partir dos algoritmos genéticos e, ainda, em relação ao método exato. Os tempos de execução utilizando o NSGA-II, em comparação ao RNSGA-II, foram menores em todas as instâncias do problema, conforme a Figura 7 apresenta.

Para avaliar as soluções obtidas nas frentes de Pareto, foram selecionados os pontos de peso 50% (no espaço objetivo normalizado foi selecionado o ponto de equilíbrio entre os dois objetivos , 50%). Para representar os resultados de forma global, foram calculadas as medianas

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

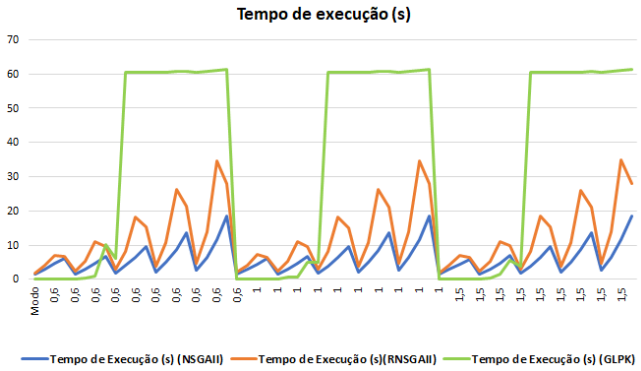


Figura 7: Tempo de Execução - NSGA-II e RNSGA-II.

para cada grupo de valores de (*makespan* e TEC). A Tabela 4 apresenta os dados, demonstrando que o NSGA-II obteve valores inferiores aos do RNSGA-II, indicando um desempenho superior.

Tabela 4: Pontos Médios (50%) dos algoritmos NSGA-II e RNSGA-II.

Modo	PM - NSGAII - *Makespan*	PM - NSGAII - TEC	PM - RNSGAII - Makespan	PM - RNSGAII - TEC
0,6	515,5	1674,46	697,5	3626,75
1	481,0	3450,78	699,5	7555,71
1,5	495,0	6211,41	703,0	13600,29

Nesta pesquisa, os algoritmos NSGA-II e RNSGA-II foram comparados em um problema com dois objetivos. O NSGA-II superou o RNSGA-II em todas as instâncias, tanto para o *Makespan* quanto para o *TEC*. Além disso, o NSGA-II resolveu todos os casos em menos tempo. Os dados obtidos com o uso do GLPK foram importantes como referência para as soluções dos algoritmos genéticos. Assim, pode-se constatar que o NSGA-II alcançou resultados superiores.

Na Figura 8 representamos duas soluções geradas pelos dois algoritmos, que simulam cenários práticos de otimização com múltiplos objetivos, onde é necessário encontrar um equilíbrio entre diversas metas, onde a eficácia de um algoritmo pode resultar em economias significativas de tempo e recursos.

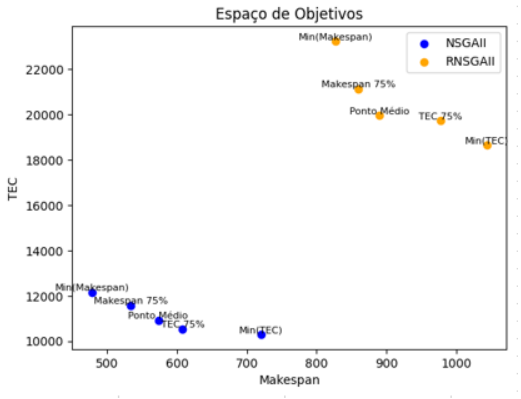


Figura 8: Escolha para Tomada de Decisão.

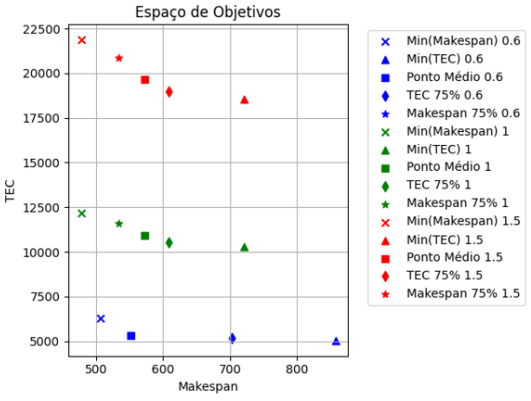


Figura 9: Impacto das Velocidades no *Makespan* e TEC.

Para empresas que necessitam solucionar problemas complicados de forma rápida, o

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

NSGA-II proporciona menor tempo de processamento, diminuição de despesas operacionais e tempos de execução das tarefas, conferindo uma vantagem competitiva ao lidar com desafios de forma ágil e precisa.

Após a análise, onde se observou que o NSGA-II apresentou melhores resultados, a Figura 9 demonstra as soluções de *makespan* e TEC para diferentes modos e velocidades de operação. Esta figura auxilia na identificação das melhores opções, de acordo com os objetivos do tomador de decisão. Dessa forma, foi possível apresentar ao tomador de decisão a melhor alternativa em termos de comparação de algoritmos e, em termos operacionais, as soluções possíveis em relação aos arranjos de modo e velocidade operacional.

6. Considerações finais e Trabalhos Futuros

No contexto da pesquisa operacional, o desempenho é um fator crucial. A avaliação realizada demonstrou que o NSGA-II não só superou as previsões, como também definiu um nível de desempenho e velocidade. O seu alto desempenho em todos os testes realizados, tanto para *makespan* quanto para TEC, demonstra a sua robustez e confiabilidade. A interação com o GLPK possibilitou uma métrica de comparação que ajuda a avaliar sua eficácia em um cenário mais abrangente. Reconhecer a importância de continuar as pesquisas nesse campo é fundamental, principalmente diante da crescente necessidade de soluções mais ágeis e precisas em ambientes industriais e empresariais.

A atenção dada a balancear o tempo de execução e o uso de energia cria novas oportunidades de pesquisa, com um foco especial nas diferentes formas de tarifação de energia. Em um cenário cada vez mais dominado por informações e cálculos, o NSGA-II se destaca como uma ferramenta essencial, permitindo que empresas se ajustem, criem novidades e tenham sucesso em ambientes difíceis. Pesquisas futuras podem investigar a relação entre o tempo de conclusão das tarefas e o gasto de energia. A conexão entre o consumo de energia e as tarifas pode ser avaliada para melhorar a utilidade do *framework* em situações reais, garantindo tempos de execução adequados e resultados de excelência.

Referências

Amouzgar, K., Bandaru, S., Andersson, T., and Ng, A. H. (2020). Metamodel-based multi-objective optimization of a turning process by using finite element simulation. *Engineering Optimization*, 52(7):1261–1278.

Blank, J. and Deb, K. (2020). A running performance metric and termination criterion for evaluating evolutionary multi-and many-objective optimization algorithms. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, p. 1–8. IEEE.

Cid, L. M., San Felice, M. C., and Hokama, P. H. D. B. (2023). Uma abordagem multiobjetivo para o problema do escalonamento de médicos. In *Anais do XXIII Simpósio Brasileiro de Computação Aplicada à Saúde*, p. 372–383. SBC.

Coello Coello, C. A., Dhaenens, C., and Jourdan, L. (2010). Multi-objective combinatorial optimization: Problematic and context. In *Advances in multi-objective nature inspired computing*, p. 1–21. Springer.

Costa, L. C. A. d. et al. (2015). Uma huerística baseada em busca local de pareto para o pollution-routing problem bi-objetivo.

Cota, L. P., Coelho, V. N., Guimarães, F. G., and Souza, M. J. (2021). Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times. *International Transactions in Operational Research*, 28(2):996–1017.

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br

de Souza Brandão, R. and Cortez, J. M. R. (2008). A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*.

Ecom (2022). Indústria e energia: qual é a relação entre os setores? urlhttps://ecomenergia.com.br/blog/industria-e-energia-impulsionam-a-economia/.

Ehrgott, M. (2005). *Multicriteria optimization*, volume 491. Springer Science & Business Media.

EPE (2020). Propostas de medidas no setor industrial brasileiro. urlhttps://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/caderno-sobre-acoes-de-eficiencia-energetica-em-industrias brasileiras.

Filatovas, E., Lančinskas, A., Kurasova, O., and Žilinskas, J. (2017). A preference-based multi-objective evolutionary algorithm r-nsga-ii with stochastic local search. *Central European Journal of Operations Research*, 25:859–878.

Giret, A., Trentesaux, D., and Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. *Journal of Manufacturing Systems*, 37:126–140.

Kádár, P. (2022). Multi objective energy mix optimalization. In *2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES)*, p. 000187–000192. IEEE.

Lacerda, H., Feitosa, A., Filho, A., Dos Santos, W., and Cordeiro, F. (2016). Redução de energia em casas inteligentes utilizando uma abordagem multiobjetivo baseada nos perfis de uso de equipamentos.

Makhorin, A. (2014). Glpk (gnu linear programming kit). 2008. *URL http://www. gnu. org/software/glpk/.(11)*.

Meloni, P. S. R. (2020). Proposta para melhoria da eficiência energética em uma indústria metalúrgica.

Miloca, S. A., Faria, T. M. B., and Volpi, N. M. P. (2012). Aplicaçao de algoritmo genético multiobjetivo em otimizaçao de portfólios.

Muller, F. M., Bridi, A. P. P., and Weis, G. (2023). Algoritmo genético multiobjetivo para o problema de sequenciamento em máquinas paralelas não-relacionadas. In *Anais do Simpósio Brasileiro de Pesquisa Operacional*, São José dos Campos. Galoá.

Pinedo, M. L. (2012). *Scheduling*, volume 29. Springer.

Verma, S., Pant, M., and Snasel, V. (2021). A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*, 9:57757–57791.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Shaker Ithaca.

DOI: 10.59254/sbpo-2024-193747
https://proceedings.science/p/193747?lang=pt-br