

Otimização de processos produtivos em sistemas de manufatura flexível

Leonardo Cabral da Rocha Soares

Instituto Federal do Sudeste MG
BR-116, 589 - Realeza, Manhuaçu - MG, 36905-000
leonardo.soares@ifesudestemg.edu.br

Marco Antonio Moreira de Carvalho

Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, s/n - Ouro Preto - MG, 35400-000
mamc@ufop.edu.br

RESUMO

Diversos problemas computacionais oriundos de sistemas de manufatura flexíveis têm sido estudados na literatura desde a década de 1980. De fato, estudos recentes demonstram que o número de publicações em temas correlatos cresce desde o ano de 1988. Diante disso, neste trabalho de pesquisa são abordados quatro problemas derivados de tais sistemas, a saber, o problema de minimização de blocos de uns consecutivos, o problema de minimização de trocas de ferramentas, o problema de sequenciamento de tarefas em máquinas paralelas com limitação de recursos e o problema de sequenciamento de tarefas em máquinas paralelas não-idênticas com restrições de ferramentas. Duas meta-heurísticas são apresentadas para abordagem aos problemas. Os experimentos computacionais realizados consideraram os *benchmarks* de instâncias disponíveis na literatura e os resultados que constituem o atual estado da arte para os problemas. Para três dos quatro problemas abordados, os resultados apresentados superam o estado da arte.

PALAVRAS CHAVE. Escalonamento de tarefas, Manufatura Flexível, Otimização Combinatória.

POI - PO na Indústria, OC - Otimização Combinatória.

ABSTRACT

Several computational problems arising from flexible manufacturing systems have been studied in the literature since the 1980s. Recent studies show that the number of publications on related topics has grown since 1988. Given this, in this work four problems derived from such systems are addressed: (i) the consecutive block minimization; (ii) the job sequencing and tool switching problem; (iii) the resource-constrained parallel machine scheduling; and, (iv) the job sequencing and tool switching problem with non-identical parallel machines. Two metaheuristics are presented to approach the problems. The computational experiments carried out considered the benchmark instances available in the literature and the results that constitute the current state-of-the-art for the problems. Except for one, the reported results outperforms the current state-of-the-art for the addressed problems.

KEYWORDS. Job Scheduling. Flexible Manufacturing. Combinatorial Optimization.

POI - OR in Industry, OC - Combinatorial Optimization.

1. Resumo

Em termos gerais, pode-se afirmar que *sistemas de manufatura flexíveis* (SMFs) são constituídos por um conjunto de máquinas flexíveis interligadas por um sistema automático de manuseio e armazenamento de materiais, controlados por um sistema distribuído de computação. Uma *máquina flexível* é capaz de executar um grande número de tarefas diferentes, incluindo ações como fresar, furar, lixar, entre outras. O processamento de cada tarefa por uma máquina flexível requer um conjunto específico de ferramentas. As ferramentas disponíveis para utilização pela máquina flexível durante o processamento de uma determinada tarefa ficam armazenadas em um compartimento específico de capacidade limitada denominado *magazine*.

Dada a ampla utilização de SMFs em diversos cenários industriais [Shirazi e Frizelle, 2001; Meeran e Morshed, 2012; González et al., 2015], numerosos problemas advindos deste cenário têm sido estudados na literatura. Em especial, os problemas derivados do *sequenciamento de tarefas* possuem grande importância teórica, pois pertencem à classe de complexidade \mathcal{NP} -Difícil [Pinedo, 2008] e notória relevância prática, evidenciada por suas aplicações em diversos segmentos industriais [Van Hop e Nagarur, 2004; Meeran e Morshed, 2012; Burger et al., 2015].

Diante do exposto, o presente trabalho propõe abordagens computacionais competitivas para quatro relevantes problemas relacionados ao sequenciamento de tarefas, todos com aplicações práticas em SMFs. Para cada problema abordado, são feitas contribuições significativas para a progressão do atual estado da arte destes. Nas seções que se seguem, são descritos os métodos computacionais propostos, os problemas considerados e as principais contribuições apresentadas.

1.1. Métodos

Com o intuito de abordar os problemas considerados nesta pesquisa, são apresentadas implementações específicas das meta-heurísticas busca local iterada (ILS) e do algoritmo genético de chave aleatórias viciadas (BRKGA) hibridizado com buscas locais organizadas em um método de descida em vizinhança variável (VND). A seguir, apresenta-se o conceito geral dos métodos utilizados.

1.1.1. Busca local iterada

A meta-heurística ILS [Lourengo et al., 2003] realiza uma exploração enviesada no espaço de soluções. A partir de uma dada solução inicial s para um problema de otimização específico, a ILS aplica alternadamente mecanismos de intensificação e diversificação, gerando uma nova solução s' que é utilizada como solução inicial para a próxima iteração do método. Apesar de conceitualmente simples, a ILS tem sido aplicada com sucesso a diversos problemas computacionalmente difíceis [Lourengo et al., 2019].

As etapas desta meta-heurística são apresentadas no Algoritmo 1. Uma solução viável para o problema é obtida, aprimorada pela utilização de métodos de buscas locais e utilizada como solução corrente, s , e melhor solução encontrada, s^* (linhas 1 e 2). O laço principal (linhas 3 a 6) garante a execução do método até que algum critério de parada seja atendido. A solução corrente é perturbada e aprimorada pelos métodos de busca local, resultando em uma solução vizinha s'' (linhas 4 e 5). Caso s'' atenda aos critérios de aceitação, a solução corrente é atualizada (linha 6). Sempre que uma nova solução com melhor valor de avaliação for encontrada, atualiza-se s^* (linha 7). Após alcançado o critério de parada, o método se encerra e a melhor solução encontrada é retornada (linha 8).

1.1.2. Algoritmo genético de chaves aleatórias viciadas hibridizado

Introduzido por Gonçalves e Resende [2011], o BRKGA é um algoritmo genético que utiliza como codificação chaves geradas aleatoriamente no intervalo contínuo $[0, 1]$. Além de sua codificação, o BRKGA difere dos algoritmos genéticos tradicionais na forma como seus operadores

Algoritmo 1: Busca local iterada

Saída: solução s ;

```

1  $s \leftarrow \text{soluçãoInicial}()$ ;
2  $s, s^* \leftarrow \text{intensificação}(s)$ ;
3 enquanto critério de parada não for encontrado faça
4    $s' \leftarrow \text{perturbação}(s)$ ;
5    $s'' \leftarrow \text{intensificação}(s')$ ;
6    $s \leftarrow \text{criterioAceitacao}(s, s'')$ 
7   se  $s$  é melhor que  $s^*$  então  $s^* \leftarrow s$ ;
8 retorne  $s^*$ ;
```

de reprodução e mutação são implementados. O Algoritmo 2 apresenta as etapas deste método já considerando um modelo de hibridização utilizado na abordagem aos problemas.

Como parâmetros de entrada, são fornecidos o número x de chaves aleatórias de cada indivíduo, o tamanho p de cada população, o tamanho p_e do conjunto elite, o número p_m de mutantes e a probabilidade ρ de se herdar os genes de cada um dos pais. A população inicial é gerada aleatoriamente (linha 1). O laço principal (linhas 2 a 21) garante que o processo evolutivo continue até que um critério de parada seja alcançado. Cada indivíduo da população é decodificado em uma solução do problema tratado (linha 3) e o valor de solução para o problema é utilizado como *fitness* do indivíduo (linha 4). O componente de hibridização é aplicado nas melhores p_e soluções de cada geração (linha 5). Após a aplicação do VND, as soluções modificadas são codificadas novamente em indivíduos da população e seu valor de *fitness* é atualizado (linha 6). Os indivíduos são ordenados pelo *fitness* e os primeiros p_e indivíduos são considerados como conjunto elite (linhas 7 e 8). O conjunto elite é integralmente copiado para a próxima geração (linha 9). Em seguida, são introduzidos na nova geração p_m indivíduos mutantes, gerados aleatoriamente (linha 10). O restante da população é gerado pela reprodução utilizando-se a recombinação uniforme parametrizada de Spears e De Jong [1995] (linhas 11 a 20). Os indivíduos são selecionados aleatoriamente, um do conjunto elite e o outro do conjunto não-elite (linhas 12 e 13). O indivíduo descendente, pode herdar de cada um dos pais com uma probabilidade ρ de herdar do pai elite (linhas 15 a 19). Ao final do processo evolutivo, a solução do problema será o indivíduo com o melhor valor de *fitness*.

1.1.3. Descida em vizinhança variável

Introduzido por Mladenović e Hansen [1997], este método baseia-se na mudança regular da estrutura de vizinhança durante a exploração do espaço de soluções. Dada uma solução inicial, as vizinhanças são exploradas de forma sequencial. A cada solução vizinha com melhor valor de avaliação, o método reinicia a exploração a partir da primeira vizinhança. Não havendo solução vizinha com melhor valor de solução, o método avança para a próxima vizinhança. O método encerra-se quando a exploração de todas as vizinhanças resultar em um ótimo local comum a todas.

As etapas do método VND são descritas no algoritmo 3. Como parâmetro de entrada são fornecidos uma solução viável s e o número r de vizinhanças consideradas. A ordem de exploração das buscas locais é definida por uma variável de controle (linha 2). O laço principal (linhas 3 a 9) garante a exploração das N vizinhanças até que um ótimo local, comum a todas, seja alcançado. As vizinhanças são exploradas linearmente (linha 4). Sempre que uma solução vizinha com melhor valor de solução é encontrada, a solução s é atualizada e a exploração se reinicia pela primeira vizinhança (linhas 5 a 7). Caso não haja melhoria da solução na vizinhança atual, move-se para a

Algoritmo 2: Algoritmo genético de chaves aleatórias viciadas hibridizado.

Entrada: x, p, p_e, p_m, ρ ;

- 1 Gere a população P com p vetores de x chaves aleatórias;
- 2 **enquanto** critério de parada não for satisfeito **faça**
- 3 Decodifique cada indivíduo;
- 4 Calcule o *fitness* de cada indivíduo em P ;
- 5 Aplique o VND nas primeiras p_e soluções;
- 6 Codifique cada solução como um indivíduo e atualize seu *fitness*;
- 7 Ordene os indivíduos em ordem não decrescente pelo valor de *fitness*;
- 8 Agrupe os primeiros p_e indivíduos em P_e ;
- 9 Inicialize a população da próxima geração $P^+ \leftarrow P_e$;
- 10 Gere um conjunto p_m de mutantes e os adicione em P^+ ;
- 11 **para** $i \leftarrow 1$ até $p - p_e - p_m$ **faça**
- 12 Selecione aleatoriamente um indivíduo a do conjunto elite;
- 13 Selecione aleatoriamente um indivíduo b do conjunto não-elite;
- 14 **para** $j \leftarrow 1$ até n **faça**
- 15 $num \leftarrow$ número aleatório entre 0 e 1;
- 16 **se** $num \geq \rho$ **então**
- 17 $c[j] \leftarrow a[j]$;
- 18 **senão**
- 19 $c[j] \leftarrow b[j]$;
- 20 $P^+ \leftarrow P^+ \cup \{c\}$;
- 21 $P \leftarrow P^+$;

próxima vizinhança (linhas 8 e 9). Quando a exploração de todas as vizinhanças não resultar em melhoria da solução atual, o método se encerra e a solução atual é retornada.

1.2. Problema de minimização de blocos de uns consecutivos

Diversos problemas estudados em otimização discreta podem ser modelados utilizando-se matrizes binárias. Entre estes, diversos são intrinsecamente permutacionais, ou seja, sua solução é determinada a partir do ordenamento das colunas (ou linhas) em uma matriz binária. Embora a avaliação da solução seja dependente do problema, em geral, ela considera a minimização do número de blocos de uns consecutivos, isto é, uma sequência contígua de elementos não-nulos em uma mesma linha de uma matriz binária.

A minimização de blocos de uns consecutivos (CBM) é um problema \mathcal{NP} -Difícil [Kou, 1977] definido como o problema de encontrar uma permutação de colunas em uma matriz binária que minimize o número total de blocos de uns. Além de sua relevância teórica, conforme mencionado, este problema pode ser aplicado em diversos contextos industriais [Soares et al., 2020].

Abordando o CBM, apresentou-se uma nova representação em grafos para o problema, uma heurística construtiva e uma implementação da meta-heurística ILS. A qualidade dos métodos propostos foi avaliada considerando-se o único conjunto *benchmark* de instâncias disponível na literatura e o método estado da arte para o CBM. Os experimentos computacionais realizados demonstraram que a nova representação e a heurística construtiva são eficazes na geração de boas soluções iniciais. O método ILS proposto mostrou-se eficaz na produção de soluções de alta qualidade. Os resultados apresentados superam os existentes na literatura anterior para todos os nove grupos de instâncias artificiais e para quatro dos cinco problemas pertencentes ao conjunto de instâncias reais.

Algoritmo 3: Descida em vizinhança variável.

Entrada: Solução s , número de vizinhanças r
Saída: Solução s

```

1  início
2       $k \leftarrow 1$ ;
3      enquanto  $k \leq r$  faça
4           $s' \leftarrow$  melhor vizinho  $\in N^{(k)}(s)$ ;
5          se valor de avaliação de  $s' <$  valor de avaliação de  $s$  então
6               $s \leftarrow s'$ ;
7               $k \leftarrow 1$ ;
8          senão
9               $k \leftarrow k + 1$ ;
10     retorna  $s$ ;
    
```

1.3. Problema de minimização de trocas de ferramentas

Conforme mencionado, para ser processada por uma máquina flexível, cada tarefa requer um conjunto específico de ferramentas. As ferramentas disponíveis para utilização durante o processamento de uma tarefa ficam alocadas no *magazine* máquina. Em geral, a capacidade do *magazine* é o suficiente para armazenar todas as ferramentas necessárias para o processamento de uma tarefa isolada. Entretanto, o conjunto total de ferramentas pode ultrapassar esta capacidade. Assim, dado um conjunto de diferentes tarefas a serem processadas sequencialmente, pode ser necessário efetuar trocas de ferramentas no *magazine* antes que uma tarefa diferente possa ser processada.

Dado o alto custo e consumo de tempo das operações advindas das trocas de ferramentas em um SMF [Agapiou, 1991; Zhang e Hinduja, 1995; Van Hop e Nagarur, 2004], torna-se essencial planejar a produção visando minimizá-las. Quando tal propósito é abordado em um SMF constituído por apenas uma máquina flexível, temos o *problema de minimização de trocas de ferramentas* (ou *job sequencing and tool switching problem*, SSP), cujo objetivo é determinar uma sequência para o processamento de n tarefas que minimize o número de trocas de ferramentas necessárias. Este problema é constituído pela combinação de dois outros problemas, o sequenciamento do processamento das tarefas na máquina e a determinação do plano de trocas de ferramentas. O primeiro problema é \mathcal{NP} -Difícil [Crama et al., 1994], sendo responsável pela complexidade do SSP. Após a definição da sequência de processamento das tarefas, o plano de trocas de ferramentas pode ser definido em tempo determinístico polinomial utilizando-se a política de manter no *magazine* as ferramentas que serão utilizadas mais cedo, denominada *Keep Tool Needed Soonest* (KTNS) e introduzida por Tang e Denardo [1988].

Para abordagem ao SSP, apresentou-se uma implementação da meta-heurística BRKGA hibridizada com três buscas locais organizadas em um VND e, uma implementação da meta-heurística ILS. Os experimentos computacionais realizados consideraram todas as instâncias disponíveis na literatura. Os resultados obtidos foram comparados aos reportados pelo atual estado da arte para o problema [Mecler et al., 2021]. Considerando-se todas as instâncias disponíveis, o BRKGA reportou um *gap* médio de 2,51% e a ILS de apenas 0,05%. A análise estatística realizada demonstrou que não há diferença estatisticamente significativa entre os resultados reportados pelos métodos propostos e pelo método detentor do atual estado da arte.

Embora competitivos em termos de qualidade da solução reportada, os métodos propostos apresentam vantagem significativa em relação ao tempo de execução. Considerando-se apenas o

conjunto contendo as maiores instâncias, o tempo de execução médio do BRKGA e do ILS são, respectivamente, 35,00% e 41,81% menores do que o tempo requerido pelo método detentor do atual estado da arte. Dada a característica prática do problema abordado, a geração de soluções com qualidade equiparável em menor tempo computacional constitui uma vantagem estratégica na geração do planejamento da produção e em eventuais replanejamentos que se façam necessários.

1.4. Problema de sequenciamento de tarefas em máquinas paralelas com limitação de recursos

O processamento de tarefas em SMFs demanda a disponibilidade de uma série de recursos, tais como máquinas flexíveis, ferramentas, paletes e manipuladores automáticos de matéria-prima, entre outros. Quando tais recursos geram limitações ao processamento de tarefas em um ambiente composto por máquinas flexíveis em paralelo, tem-se o *problema de sequenciamento em máquinas paralelas com limitações de recursos* (*resource constrained parallel machine scheduling*, RCPMS). Primeiramente estudado por Garey e Graham [1975], o RCPMS pertence à classe \mathcal{NP} -Difícil, visto que, se considerarmos cada máquina do RCPMS isoladamente, tem-se uma instância do problema de sequenciamento de tarefas em uma máquina flexível, um problema provado pertencer à classe \mathcal{NP} -Difícil por Crama et al. [1994]. Entre as diversas aplicações do RCPMS, optou-se por abordá-lo no contexto prático que ocorre nas indústrias de fabricação de componentes microeletrônicos [Chung et al., 2019], em especial na fabricação de microchips, onde o *molde* utilizado durante o processo de impressão dos circuitos é o recurso limitado compartilhado.

Para abordagem ao problema, apresentou-se uma implementação do método BRKGA híbrido com quatro buscas locais organizadas em um VND. O único *benchmark* de instâncias disponíveis na literatura foi analisado e provou-se trivial. Resultados ótimos foram reportados para todas as instâncias. Um novo conjunto de instâncias, contendo 270 problemas mais desafiadores foi apresentado. Uma extensiva campanha experimental foi realizada para geração de soluções ótimas e limitantes inferiores e superiores para o novo conjunto. Foram considerados quatro modelos matemáticos, duas heurísticas baseadas em processamento de listas e uma implementação da meta-heurística *general variable neighborhood search*. Os resultados reportados pelo método proposto para o novo conjunto de instâncias foram avaliados utilizando-se os melhores limitantes superiores e inferiores obtidos. Considerando-se todas as instâncias, o método proposto reportou um *gap* médio de 22,44% e -7,62%, em comparação aos limites inferiores e superiores, respectivamente.

1.5. Problema de sequenciamento de tarefas em máquinas paralelas não-idênticas com restrições de ferramentas

Apesar de originalmente abordado em um ambiente composto por apenas uma máquina flexível, o SSP está presente em diversos cenários industriais. Quando estudado em um ambiente composto por máquinas paralelas não-idênticas, o problema é conhecido na literatura como *problema do escalonamento de tarefas em máquinas paralelas não-idênticas com restrições de ferramentas* (ou *job sequencing and tool switching problem with non-identical parallel machines*, SSP-NPM), consistindo em determinar a alocação e o sequenciamento das tarefas nas máquinas, bem como o plano de trocas de ferramentas de acordo com uma função objetivo.

Especificamente, as máquinas do ambiente abordado diferem no tempo necessário para processar cada tarefa, no tempo requerido para troca de cada ferramentas e na capacidade do *magazine*, adicionando ao problema uma restrição referente à elegibilidade de máquinas. Como objetivos, foram considerados, isoladamente, a minimização do *makespan* e a minimização do *flow time* total. Embora a função objetivo original para o SSP seja considerada em alguns trabalhos sobre o tema, esta não foi incluída por ser uma métrica com menor aplicação prática e relevância acadêmica, para o contexto com múltiplas máquinas não-idênticas [Calmels, 2022].

Para cada objetivo apresentou-se uma implementação específica do método BRKGA híbrido com buscas locais organizadas em um VND. Os experimentos computacionais realizados utilizaram o único *benchmark* de instâncias disponível na literatura. Os melhores resultados conhecidos para o conjunto de instâncias foram considerados na aferição da qualidade das soluções reportadas pelos métodos propostos. Todos os resultados ótimos conhecidos foram alcançados. Resultados médios melhores ou iguais foram reportados para todos os subconjuntos de instâncias, considerando-se os dois objetivos. Para o conjunto contendo as maiores instâncias, reporta-se um *gap* médio de -16,07% para a minimização do *makespan* e -14,54% para a minimização do *flow time* total.

1.6. Conclusão

Este resumo apresenta as principais contribuições da tese intitulada “Otimização de processos produtivos em sistemas de manufatura flexível”. Relevantes problemas computacionais oriundos de sistemas de manufatura flexíveis foram abordados. Para solução dos problemas, foram apresentadas implementações especializadas das meta-heurísticas busca local iterada e algoritmo genético de chaves aleatórias viciadas. Os resultados obtidos foram comparados ao atual estado da arte para os problemas. Todas as abordagens resultaram em relevantes contribuições. Para três dos quatro problemas abordados, os novos resultados publicados superaram o anterior estado da arte.

2. Publicações

Durante este trabalho de pesquisa três artigos foram elaborados. Destes, dois já foram publicados e um encontram-se em processo de revisão por pares.

2.1. Artigos publicados

- Soares, L. C. R., Reinsma, J. A., Nascimento, L. H. L., e Carvalho, M. A. M. (2020) Heuristic methods to consecutive block minimization. *Computers Operations Research*, 120, 104948. <https://doi.org/10.1016/j.cor.2020.104948>.
- Soaers, L. C. R. e Carvalho, M. A. M. (2022). Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times. *Computers Operations Research*, 139, 105637. <https://doi.org/10.1016/j.cor.2021.105637>.

2.2. Artigo em processo de revisão por pares

- *Biased random-key genetic algorithm for job sequencing and tool switching problem with non-identical parallel machines*. Submetido ao periódico *Computers and Operations Research*

Referências

- Agapiou, J. S. (1991). Sequence of operations optimization in single-stage multifunctional systems. *Journal of Manufacturing Systems*, 10(3):194–208.
- Burger, A. P., Jacobs, C., van Vuuren, J. H., e Visagie, S. E. (2015). Scheduling multi-colour print jobs with sequence-dependent setup times. *Journal of Scheduling*, 18(2):131–145.
- Calmels, D. (2022). An iterated local search procedure for the job sequencing and tool switching problem with non-identical parallel machines. *European Journal of Operational Research*, 297(1):66–85.
- Chung, T., Gupta, J. N., Zhao, H., e Werner, F. (2019). Minimizing the makespan on two identical parallel machines with mold constraints. *Computers & Operations Research*, 105:141–155.

- Crama, Y., Kolen, A. W. J., Oerlemans, A. G., e Spijksma, F. C. R. (1994). Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6(1):33–54. ISSN 0920-6299, 1572-9370.
- Garey, M. R. e Graham, R. L. (1975). Bounds for multiprocessor scheduling with resource constraints. *SIAM Journal on Computing*, 4(2):187–200.
- Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- González, M. A., Oddi, A., Rasconi, R., e Varela, R. (2015). Scatter search with path relinking for the job shop with time lags and setup times. *Computers & Operations Research*, 60:37–54.
- Kou, L. T. (1977). Polynomial complete consecutive information retrieval problems. *SIAM Journal on Computing*, 6(1):67–75.
- Lourenço, H. R., Martin, O. C., e Stützle, T. (2003). Iterated local search. In *Handbook of metaheuristics*, p. 320–353. Springer, New York, United States.
- Lourenço, H. R., Martin, O. C., e Stützle, T. (2019). Iterated local search: Framework and applications. *Handbook of metaheuristics*, p. 129–168.
- Mecler, J., Subramanian, A., e Vidal, T. (2021). A simple and effective hybrid genetic search for the job sequencing and tool switching problem. *Computers & Operations Research*, 127:105153.
- Meeran, S. e Morshed, M. (2012). A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study. *Journal of intelligent manufacturing*, 23(4):1063–1078.
- Mladenović, N. e Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100.
- Pinedo, M. (2008). Scheduling: theory, and systems.
- Shirazi, R. e Frizelle, G. D. M. (2001). Minimizing the number of tool switches on a flexible machine: an empirical study. *International Journal of Production Research*, 39(15):3547–3560. ISSN 0020-7543.
- Soares, L. C., Reinsma, J. A., Nascimento, L. H., e Carvalho, M. A. (2020). Heuristic methods to consecutive block minimization. *Computers & Operations Research*, 120:104948.
- Spears, W. M. e De Jong, K. D. (1995). On the virtues of parameterized uniform crossover. Technical report, Naval Research Lab, Washington D.C., Washington D.C., United States.
- Tang, C. S. e Denardo, E. V. (1988). Models arising from a flexible manufacturing machine, part I: Minimization of the number of tool switches. *Operations Research*, 36(5):767–777. ISSN 0030-364X.
- Van Hop, N. e Nagarur, N. N. (2004). The scheduling problem of pcbs for multiple non-identical parallel machines. *European Journal of Operational Research*, 158(3):577–594.
- Zhang, J. e Hinduja, S. (1995). Determination of the optimum tool set for a given batch of turned components. *CIRP annals*, 44(1):445–450.