

Aplicação de um algoritmo evolutivo na solução de problemas de sequenciamento da produção

Lisandro Luiz Schneider e Leandro Luís Corso

Resumo

A utilização de técnicas de sequenciamento da produção proporcionam uma melhor eficiência das linhas, assim como também verifica-se uma distribuição equilibrada dos roteiros produtivos. Desta forma, a busca pela otimização dos recursos existentes e pelas melhores práticas quanto ao tratamento das restrições, faz com que as organizações de um modo em geral pensem em diferentes estratégias. Neste contexto, buscam-se soluções aplicadas ao sequenciamento de linha de produção, como, por exemplo, o *Flexible Job Shop Schedule Problem* (FJSP), objetivando-se encontrar o menor tempo de ocupação das linhas, dentre outras funcionalidades. Desta maneira, as metodologias heurísticas de sequenciamento podem ser consideradas como importantes ferramentas para o gerenciamento da produção. Entretanto, este estudo propõe a aplicação de um algoritmo evolutivo para a solução de problemas de FJSP, baseado na metodologia *Hybrid Taguchi Genetic Algorithm* (HTGA). Sendo assim, primeiramente elaborou-se o modelo matemático a ser utilizado, para que o mesmo seja executado realizou-se a junção dos métodos de algoritmo genético com o método Taguchi e por fim realizou-se a validação da proposta por meio da criação de duas variações do algoritmo evolutivo com taxas de mutação diferentes. Compara-se estes dois algoritmos elaborados com um método existente, a partir de problemas de teste, também presentes na literatura. Tornando-se possível então, verificar que o algoritmo apresentado foi capaz de resolver os problemas e de encontrar o mesmo valor da função objetivo. Desta forma, observa-se neste artigo a possibilidade de utilização do algoritmo híbrido proposto para a resolução de problemas de sequenciamento, o qual é composto por algoritmos reconhecidos por serem relativamente de fácil aplicação. Conclui-se neste estudo, a validação do modelo matemático evolutivo na resolução de problemas de FJSPs, tornando-se uma nova alternativa de solução para os problemas de sequenciamento.

Palavras-chave

Algoritmo evolutivo, Indústria 4.0, Sequenciamento da produção, Algoritmo genético, Método Taguchi, Modelo matemático.

Application of an evolutionary algorithm to solve production scheduling problems

Abstract

The use of production sequencing techniques provides better line efficiency, as well as a balanced distribution of production routes. In this way, the search for the optimization of existing resources and the best practices regarding the treatment of restrictions, makes organizations in general think about different strategies. In this context, solutions applied to the sequencing of the production line are sought, such as, for example, the Flexible Job Shop Schedule Problem (FJSP), aiming to find the shortest time to occupy the lines, among other features. In this way, heuristic sequencing methodologies can be considered as important tools for production management. However, this study proposes the application of an evolutionary algorithm for the solution of FJSP problems, based on the Hybrid Taguchi Genetic Algorithm (HTGA) methodology. Therefore, first the mathematical model to be used was elaborated, so that it can be executed, the genetic algorithm methods were joined with the Taguchi method and, finally, the proposal was validated by creating two variations of the evolutionary algorithm with different mutation rates. These two algorithms are compared with an existing method, based on test problems, also present in the literature. Making it possible then, to verify that the presented algorithm was able to solve the problems and to find the same value of the objective function. Thus, it is observed in this article the possibility of using the hybrid algorithm proposed to solve scheduling problems, which is composed of algorithms recognized for being relatively easy to apply. It concludes in this study, the validation of the evolutionary mathematical model in the resolution of problems of FJSPs, becoming a new alternative of solution for the scheduling problems.

Keywords

Evolutionary algorithm, Industry 4.0, Production scheduling, Genetic algorithm, Taguchi method, Mathematical model.

I. INTRODUÇÃO

A utilização de metodologias ligadas à Indústria 4.0 traz consigo desafios, os quais ao serem superados, geram

oportunidades de crescimento e de consolidação. Considerando-se que qualidade deve estar intrínseca ao produto fabricado, as organizações tem direcionado seus esforços para a melhoria de seus processos, no

Pós graduação em Engenharia 4.0 – Universidade de Caxias do Sul (UCS)

E-mails: llschne1@ucs.br; llcorso@ucs.br

Data de envio: 03/09/2020

Data de aceite: 27/10/2020

<http://dx.doi.org/10.18226/23185279.v8iss2p175>

desenvolvimento de novas tecnologias, na contratação de mão de obra qualificada e na otimização dos recursos existentes. Torna-se necessário introduzir novos métodos para atender uma manufatura flexível, a qual faz parte de um contexto com problemas de demanda, com restrição de recursos e com frequentes oscilações de mercado. A utilização dos conceitos da quarta revolução industrial melhoram os índices de eficiência das operações, proporcionando maior produtividade e economia dos recursos produtivos [1]. De acordo com [2], sob o ponto de vista da Indústria 4.0 aplicam-se cada vez mais sistemas inteligentes de manufatura, tais como, Internet Industrial das Coisas (IIoT, do inglês *Internet Industrial of Things*), *Big Data*, Inteligência Artificial (AI, do inglês *Artificial Intelligence*), *Smart Manufacturing*, *Digital Twin*, entre outros. Desta maneira, torna-se importante a sincronização do planejamento da produção com estas tecnologias dentro do ambiente fabril, focando-se os esforços para utilizar inteligentes modelos de sequenciamento e de otimização. Algumas das técnicas utilizadas para minimizar o *makespan*, assim como visões de futuras soluções dos problemas de otimização, podem ser verificados em [3]. O qual também descreve a necessidade de se encontrar um sequenciamento fabril que otimiza a função objetivo.

Nos últimos 20 anos, realizaram-se estudos em diversos setores com o objetivo de minimizar os tempos de paradas de linha e dos gargalos de produção, buscando-se as melhores alternativas para sequenciar linhas de produção. A técnica de Problema de Sequenciamento do Trabalho (JSP, do inglês *Job Shop Schedule Problem*) pode ser aplicada para os casos de sequência linear de produção [4]. Esta metodologia pode ser empregada para resolver problemas em operações complexas, conforme em [5]. Também verifica-se a necessidade de se utilizar modelos de planejamento mais robustos e que contemplem as incertezas de processo. Sendo assim, ter um modelo que permita criar e desenvolver cenários de forma prática, podendo inclusive auxiliar nesta verificação [4]. De acordo com [6], aplica-se a metodologia Problema de Sequenciamento de Trabalho Flexível (FJSP, do termo inglês *Flexible Job Shop Scheduling Problem*) para flexibilizar o roteamento das atividades e também para minimizar o *makespan* (C_{max}), que equivale ao tempo de conclusão de todas as atividades. Todavia, para que a competitividade da organização não seja afetada, torna-se necessário avaliar também outros critérios como o custo das operações e seus prazos de entrega.

Observa-se que a resolução dos problemas de FJSP pode ser obtida por meio da utilização de algoritmos heurísticos, como por exemplo, *Tabu Search* (TS), *Simulated Annealing* (SA), *Ant Colony Optimization*, Algoritmo Genético (GA, do termo inglês, *Genetic Algorithm*), metodologias híbridas, entre outros. Dentre as alternativas, o GA vem sendo aplicado em grande escala e em diferentes áreas, podendo ser também modificado para atender as expectativas [7]. Os modelos matemáticos também vem contribuindo para a solução de problemas produtivos. Também citam-se trabalhos envolvendo a programação linear inteira-mista (MILP do termo inglês, *Mixed-Integer Linear Programming*) para problemas de JSP. Torna-se importante salientar a utilização de metodologias híbridas para a solução de problemas de FJSP,

as quais estão sendo utilizadas em diversos segmentos. Como exemplo, verifica-se em [8], o emprego do método híbrido que utiliza um modelo GA juntamente a um método determinístico em operações consideradas complexas.

O desenvolvimento de metodologias metaheurísticas, assim como a utilização de algoritmos híbridos, em grande parte das vezes, torna mais eficiente a resolução de problemas fabris [9]. Como, exemplos de aplicações de metodologias híbridas e algoritmos evolutivos podem ser verificados em [10], os quais visam a resolução de tarefas dinâmicas e problemas de sequenciamento. Todavia, podem ser verificadas lacunas quanto a aplicação de metodologias voltadas ao sequenciamento de tarefas. Desta maneira, evidenciam-se oportunidades de empregar modelos heurísticos nas organizações, visando por exemplo, a redução das perdas de produção e dos atrasos de entrega, melhoria do fluxo produtivo e da programação. As técnicas de sequenciamento devem levar em consideração a quantidade de equipamentos, as particularidades do sistema de produção, os indicadores de performance e das restrições dos recursos [2].

Os planos de produção sem otimização apresentam perdas significativas para as organizações em geral. Em alguns casos as tratativas para solução dos mesmos são negligenciadas pela falta de conhecimento de metodologias ou ainda não são utilizadas pelo fato do elevado tempo de pesquisa necessário para a sua elaboração. Contudo, o presente artigo propõe um algoritmo evolutivo para solução de problemas de FJSP. Ao mesmo tempo, justifica-se a aplicação de um algoritmo como este pela facilidade de implementação, enquanto, a solução por se tratar de um problema complexo como este pode exigir elevado custo computacional para implementação e solução [8]. Os demais tópicos deste trabalho estão dispostos da seguinte maneira: referencial teórico, metodologia utilizada para a resolução do problema, apresentação dos resultados obtidos, listagem das conclusões obtidas e por fim elencam-se as referências bibliográficas.

II. REFERENCIAL TEÓRICO

A. Indústria 4.0

Quando examina-se a evolução e o desenvolvimento das técnicas produtivas ao longo do tempo, constata-se uma alteração dos cenários até então existentes. A quarta revolução industrial ou Indústria 4.0 traz consigo a oportunidade de uma significativa transformação digital em nossos meios de fabricação. Neste cenário se faz necessária a utilização de tecnologias como: sensores inteligentes, computação em nuvem, impressão 3D, realidade aumentada, interfaces humano-máquinas, plataformas para a internet das coisas (IoT, do inglês *Internet Of Things*), dispositivos móveis, *Big Data* e algoritmos avançados [1]. Apresenta-se a possibilidade de objetos realizarem integrações inteligentes a qualquer momento e em diferentes lugares a partir da conectividade com a *internet*. Possibilita-se a obtenção de informações do produto e do local de trabalho simultaneamente à sua fabricação [11]. Propõe-se a utilização dos conceitos da Indústria 4.0 para que as organizações tenham flexibilização e conectividade dos processos, proporcionando um planejamento operacional mais adequado à realidade, visto

também que as informações estão dispostas em tempo real [12]. Em [13], projeta-se como um aspecto importante neste novo contexto industrial, a obtenção das integrações horizontal, vertical e a digital em toda a planta.

Esta nova era fora estudada sob muitos aspectos, diferentes autores e em vários países, com o objetivo de encontrar as melhores práticas para cada contexto. Avalia-se esta metodologia em quatro grupos: fábrica, negócio, produtos e cliente. As atividades industriais foram classificadas com o objetivo de estruturar a respectiva transição por meio de um roteiro de atividades, sendo que cada fator analisado influencia na rotina fabril de uma organização [1]. Define-se a quarta revolução industrial de acordo com os seus princípios de *design* e tendências tecnológicas em [14]. Onde os princípios fornecem os recursos necessários para a transição e as tendências são responsáveis pelas inovações tecnológicas digitais. Citam-se também desafios como segurança da informação, confiabilidade e integridade para suportar os novos e complexos requisitos, além de tecnologias existentes nestes novos cenários produtivos [15].

Analisa-se esta transformação da manufatura como uma metodologia integradora, com foco no uso de robótica, automação avançada, conectividade, sensores inteligentes, inteligência artificial e análise dos dados, para melhorar o desempenho organizacional. A utilização da digitalização também contribuiu para a melhoria da segurança dos processos. Do mesmo modo, verifica-se um papel importante quanto a transformação da mentalidade corporativa [16]. Verifica-se em [1] que a instabilidade e a volatilidade dos mercados exigem das organizações um constante exercício de adaptabilidade, onde frequentemente é necessário inovar e tratar os problemas a partir de novas metodologias. Já em [13] confere-se que a transição para o contexto da Indústria 4.0 exige a superação de obstáculos e de persistência para manter o foco na realização das ações pertinentes. Fatores como descentralização, adaptabilidade, flexibilidade e integração das informações devem estar presentes nessa trajetória. As estratégias de sequenciamento, dos problemas de programação e as conexões com a rede de fornecedores também fazem parte deste roteiro e podem ser analisadas de acordo com metodologias específicas. Sob o ponto de vista da quarta revolução industrial [2], o sequenciamento de produção deve interagir com um sistema de fabricação inteligente e estar suportado por novas tecnologias.

B. Metodologias para os problemas de sequenciamento de trabalho flexível

Cada vez mais, torna-se necessário alterar o foco do sequenciamento produtivo para distribuições inteligentes. O estudo dos problemas de sequenciamento fabril tem sido abordado desde os anos 90 por diversos pesquisadores, tendo como principais metas minimizar as perdas produtivas, aumentar a eficiência de utilização e contornar as restrições. O *Flexible Job-shop Scheduling Problem* (FJSP) é uma extensão do *Job Shop Scheduling Problem* (JSP), incrementando-se a flexibilidade de roteamento. De forma geral, o FJSP é referido como problema industrial onde as operações podem ter mais de uma origem, assim como mais de uma possibilidade em relação às atividades posteriores

[17]. Presenciam-se também no FJSP as restrições de disponibilidade, em [18] são verificadas pesquisas relacionadas a problemas de sequenciamento com restrições de disponibilidade do equipamento. O FJSP possui ampla aplicação em casos reais, pois compreende as principais características dos atuais sistemas de manufatura [6].

Verificam-se ambientes dinâmicos nas organizações onde torna-se necessário lidar com situações de falta de funcionários, espera por transporte, estoques intermediários, ociosidade, ineficiência produtiva, gargalos, entre outros. Neste sentido, desenvolveram-se metodologias heurísticas para minimizar tais contratempos. Em [10] verificam-se técnicas de despacho para o problema de programação dinâmica, as quais incluem novas demandas de trabalho, incertezas como erros na estimativa de tempos de processo, tempos de espera e manutenção dos equipamentos. Tornando-se necessário maior controle para obter um melhor resultado do sequenciamento fabril. O tempo de *set up* também influencia nas decisões de sequenciamento, principalmente em operações complexas, sendo assim, também aborda-se o FJSP com o objetivo de minimizar o atraso considerando o tempo de preparação [7]. Analisou-se em [19] o problema de múltiplos produtos, horizonte finito e estático de programação de demanda de uma organização, incluindo três etapas principais: sequenciamento, lotes e programação. Outro problema considerado por estes autores considerava: lotes com quantidade definida pelo cliente, remessa com data de vencimento e possibilidade de processamento dos itens por mais de um equipamento. Lembra-se também dos fornecedores automotivos que possuem contratos, onde os atrasos são penalizados e os adiantamentos não são permitidos.

De acordo com [20], apresentam-se no FJSP duas dificuldades: atribuição de cada operação a uma máquina e o agendamento dessas atividades para tornar o objetivo pré-definido mínimo. Observam-se técnicas de resolução do FJSP com um ou múltiplos objetivos, denominada *Multiple Objective Flexible Job Shop Schedule Problem* (MOFJSP) utilizando-se muitas vezes de algoritmos metaheurísticos, como, por exemplo, o *Tabu Search* que emprega regras de pesquisa de vizinhança para atribuição de máquinas e programação de operações [6]. Também sugere-se o algoritmo de enxame de partículas para a minimização simultânea do tempo de fabricação com o período de atraso médio dentro de uma perspectiva de MOFJSP [6]. Apresenta-se um modelo de Programação Linear Inteira Mista (MILP) em [7] para um FJSP com configurações dependentes de sequência, em conjunto com um algoritmo de busca local que utiliza um princípio de envio para obter um valor mínimo do makespan. As atividades que possuem roteiros alternativos, também chamados FJSP com Flexibilidade de Plano de Processo (FJSP-PPF, do termo inglês *FJSP-Process Plan Flexibility*) utilizam, por exemplo, métodos heurísticos como algoritmos evolutivos, *Tabu Search*, *Annealing Simulation*, *Ant Colony Optimization* e abordagens híbridas [20].

Como mencionado anteriormente neste artigo, a utilização de algoritmos genéticos vem crescendo devido a sua flexibilidade e adaptabilidade. De acordo com [2], o GA é um método metaheurístico o qual baseia-se nos mecanismos da

evolução genética. Como características, agem diretamente na estrutura do problema e tem a capacidade de se ajustar à pesquisa. Algumas adaptações vão sendo feitas aos GAs para a resolução de diferentes FJSPs ao longo do tempo, inclusive com a integração de outros métodos heurísticos. Aplica-se esta metodologia em problemas de múltiplos critérios em [17]. Para um problema de programação multiobjetivo flexível do plano de processo, aplica-se em [20] uma abordagem hierárquica, fundamentando-se na decomposição da situação em subproblemas de carregamento e programação do equipamento. Empregando-se duas heurísticas, uma baseada em substitutos da teoria da dualidade e outra em um algoritmo genético. Além disso, desenvolveu-se também em [20] um algoritmo metaheurístico baseando-se em um GA de sequenciamento versátil em um sistema fabril flexível levando-se em conta a eficiência e a estabilidade dos roteiros de fabricação. Visualiza-se a utilização de GA em [7], onde para um problema de FJSP-PPF existe um modelo matemático no qual cada pedido do cliente tem uma data de vencimento e a terceirização está disponível. Este diagnóstico leva em consideração a avaliação das máquinas alternativas e dos processos disponíveis para cada operação. Evidenciam-se soluções eficazes para o atendimento da diversidade encontrada em uma programação fabril com a utilização de um modelo de GA com regras de envio em [10].

C. Aplicação de metodologias híbridas

Elencam-se também os modelos matemáticos e os algoritmos híbridos como alternativas para a resolução dos problemas de agendamento nas organizações. Entretanto, quanto aos métodos matemáticos, torna-se necessário incrementar técnicas heurísticas para que os problemas sejam resolvidos. Verifica-se em [7] o desenvolvimento de um modelo matemático para um caso de FJSP-PPF, o qual foi resolvido por meio de um GA. Propõe-se um outro exemplo de aplicação para integrar o planejamento e o sequenciamento de processos, sendo sua resolução realizada por um algoritmo de otimização de enxame de partículas [17]. Adota-se em [19] a metodologia de GA para resolução do modelo matemático em um problema cíclico de produção. Aplicam-se em [20] duas metodologias para a resolução de um modelo matemático, resultando em dois algoritmos: abordagem integrada com heurística de *Simulated Annealing* (ISA, do inglês, *Integrated approach with Simulated Annealing heuristic*) e a abordagem integrada com heurística de *Tabu Search* (ITS, do inglês, *Integrated approach with Tabu Search heuristic*).

A flexibilidade dos planos de processo acrescentam uma terceira dimensão ao FJSP. Nesta direção, diferentes autores desenvolveram métodos híbridos ou evolutivos combinando dois ou mais algoritmos com o objetivo de aprimorar a convergência e a diversidade de soluções, assim como resolver problemas de sequenciamento em atividades dinâmicas. Identifica-se em [8] o Algoritmo Genético Lagrangeano Aumentado para tratar restrições por meio de funções de penalização e baseado no uso de operadores genéticos como mutação, cruzamento e elitismo. Um algoritmo genético incremental híbrido (HIGA, do inglês, *Hybrid Incremental Genetic Algorithm*) é proposto em [9]

para resolver problemas de grande porte, onde a metodologia combina uma estratégia de desenvolvimento incremental, um refinamento local e o monitoramento da diversidade populacional. Sugere-se em [7] um algoritmo genético híbrido para resolver um problema de FJSP com restrições de disponibilidade de equipamento, recursos não fixados e ocorrência de manutenções preventivas.

Citam-se exemplos de hibridização em [6], como a combinação de um algoritmo genético com um *Simulated Annealing*, a hibridização da otimização de enxame de partículas e *Tabu Search*, a união de algoritmo híbrido discreto de otimização de enxame de partículas e *Simulated Annealing*, a junção dos algoritmos genéticos com o *Shifting Bottleneck* e a conexão da otimização genética com um método de busca local para tratar dos tempos de processamento e datas de vencimento imprecisos. A elaboração de uma hibridização que otimiza os critérios de algoritmo genético e de *Tabu Search* para minimizar o *makespan* e também as perdas por atrasos, podem ser exemplificadas em [17]. Já em [10], mencionam-se abordagens híbridas envolvendo: redes neurais e GA para resolver problemas de programação dinâmica; pesquisa em vizinhança variável (VNS) com a Rede Neural Artificial (RNA) considerando-se chegadas aleatórias de trabalhos e manutenções; hibridização de um algoritmo genético com *Ant Colony Optimization* levando-se em conta o tempo de conclusão das atividades e o custo de fabricação.

De acordo com o exposto, verificam-se variadas alternativas para hibridização de técnicas heurísticas ou ainda de um modelo matemático com um algoritmo. Tais combinações oferecem robustez à solução de problemas de sequenciamento, além de oferecer um maior número de possibilidades aos constantes desafios presentes em nosso cotidiano.

III. MÉTODO

Como pode ser verificado, geralmente as análises para otimização do sequenciamento produtivo demanda a utilização de metodologias complexas. Neste artigo, propõe-se utilizar um modelo matemático para maximizar os resultados produtivos a partir de um planejamento de produção voltado a minimizar o *makespan*.

Do mesmo modo, recomenda-se para a sua resolução a aplicação de um algoritmo híbrido. Para isso, as etapas a seguir ilustram as fases necessárias para atingir este objetivo.

A. Apresentação do modelo matemático utilizado

Nesta etapa do estudo apresenta-se o modelo matemático linear adaptado de [7], o qual tem por objetivo possibilitar o cálculo de otimização do processo de sequenciamento. Para tal, salientam-se algumas definições para os FJSPs, como por exemplo: as atividades são independentes, os tempos são fixos, as máquinas estão disponíveis no momento inicial e é elaborado um plano de processo viável para cada atividade. Na Equação 1 se apresenta a função objetivo e nas Equações 2 a 8 estão mostradas as restrições do modelo, onde o detalhamento dos termos está apresentado a seguir.

Função Objetivo: Minimizar C_{\max}

Restrições:

$$\sum_{k \in M_j} X_{ijk} = 1 \quad \forall i \in J, \forall j \in O_i, \quad (1)$$

$$S_{ijk} + C_{ijk} \leq (X_{ijk}) * L \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j, \quad (2)$$

$$C_{ijk} \geq S_{ijk} + t_{ijk} - (1 - X_{ijk}) * L \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j, \quad (3)$$

$$S_{ijk} \geq C_{i'j'k} - (Y_{ij'j'k}) * L \quad \forall i < i', \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'}, \quad (4)$$

$$S_{i'j'k} \geq C_{ijk} - (1 - Y_{ij'j'k}) * L \quad \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'}, \quad (5)$$

$$\sum_{k \in M_j} S_{ijk} \geq \sum_{k \in M_j} C_{i,j-1,k} \quad \forall i \in J, \forall j \in O_i - \{O_{if(i)}\}, \quad (6)$$

$$C_i \geq \sum_{k \in M_j} C_{i,O_{il(i)},k} \quad \forall i \in J, \quad (7)$$

$$C_{\max} \geq C_i \quad \forall i \in J, \quad (8)$$

e

$$X_{ijk} \in \{0,1\} \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j,$$

$$S_{ijk} \geq 0 \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j,$$

$$C_{ijk} \geq 0 \quad \forall i \in J, \forall j \in O_i, \forall k \in M_j,$$

$$Y_{ij'j'k} \in \{0,1\} \quad \forall i < i', \forall j \in O_i, \forall j' \in O_{i'}, \forall k \in M_j \cap M_{j'},$$

$$C_i \geq 0 \quad \forall i \in J.$$

Onde:

C_{\max} = tempo máximo de conclusão de todas as tarefas (*makespan*)

C_i = tempo completo da tarefa i

t_{ijk} = tempo processamento da operação O_{ij} na máquina k

L = maior número

X_{ijk} igual a 1 se a máquina k é selecionada para a operação O_{ij} , senão é igual a zero

S_{ijk} é o tempo de início da operação O_{ij} na máquina k

C_{ijk} é o tempo total de operação O_{ij} na máquina k

$Y_{ij'j'k}$ = é igual a 1 se a operação O_{ij} precede a operação $O_{i'j'}$ na máquina k , senão o tempo completo da tarefa i

i = tarefas ($i, i' \in J$)

j = atividades ($j, j' \in O$)

k = máquinas ($k \in M$)

J = conjunto de tarefas

M = conjunto de máquinas

O = conjunto de atividades

O_i = conjunto ordenado de operações da tarefa i ($O_i \subseteq O$), onde $O_{if(i)}$ é o primeiro e $O_{il(i)}$ é o último elemento de O_i

M_j = conjunto de máquinas alternativas nas quais a operação j pode ser processada, ($M_j \subseteq M$)

$M_j \cap M_{j'}$ = conjunto de máquinas nas quais as operações j e j' podem ser processadas

Observa-se em [7], no método MILP-1, que a restrição (1) garantem que a operação O_{ij} esteja atribuída a apenas um equipamento. A restrição (2) define os tempos de início e conclusão na máquina k iguais a zero, caso não ocorra atribuição das atividades O_{ij} à máquina k . Garante-se na terceira restrição (3) que a diferença entre os tempos de início e de conclusão seja ao tempo de processamento na máquina k . As duas próximas restrições (4) e (5) fazem com que as operações O_{ij} e $O_{i'j'}$ não sejam processadas simultaneamente em qualquer um dos equipamentos do grupo $M_j \cap M_{j'}$. A próxima restrição (6) impede que a atividade O_{ij} seja iniciada antes da operação O_{ij-1} ser encerrada. Os tempos de

conclusão das tarefas são determinadas na restrição (7). Determina-se o *makespan* nas restrições (8).

Verifica-se que o conjunto de equações representa um problema de programação linear. Entretanto, a modelagem matemática de problemas de sequenciamento de linha, normalmente é bastante complexa e pode exigir um elevado custo computacional de solução com a utilização de algoritmos convencionais. Fazendo com que muitas vezes as empresas busquem a compra de softwares ou mesmo solver matemáticos comerciais. Ao mesmo tempo, a resolução do modelo de otimização como este pode ser feita por implementação de algoritmos evolutivos, como é proposto neste trabalho. A seguir, o modelo proposto é apresentado.

B. Implementação do algoritmo evolutivo proposto

Buscando-se alternativas para a resolução do modelo matemático apresentado, este artigo sugere a utilização de um algoritmo híbrido para facilitar a obtenção de resultados e manter a qualidade do valor da função objetivo. Procura-se com este algoritmo evolutivo, viabilizar um planejamento da produção equilibrado e uma otimização das atividades. Contribuindo também com o tratamento das restrições apresentadas, sejam elas referentes ao equipamento ou às atividades realizadas. Neste estudo, dentre as alternativas avaliadas, definiu-se em utilizar o método denominado *Hybrid Taguchi Genetic Algorithm* (HTGA).

Primeiramente definiu-se pela utilização do GA por apresentar ótimos resultados em vários campos de aplicação, tornando-se um dos métodos evolutivos mais importantes e utilizados atualmente [5] e [21]. Mesmo com o alto índice de desempenho de resolução do GA, verifica-se que a hibridização pode tornar os resultados mais eficientes nos casos de FJSP [5]. Dentre as possibilidades de combinação, as pesquisas indicaram excelentes resultados quando aplicavam-se as metodologias de GA com o método Taguchi. Este por sua vez, vem sendo empregado em projetos robustos desde sua primeira utilização por Genichi Taguchi, pois identifica a combinação cromossômica ideal após o cruzamento, acelerando a convergência e aumentando a eficácia do GA [22].

O resultado da combinação do método híbrido Taguchi com GA resultou no método denominado HTGA, proposto por Jyh-Horng Chou em [21], [23] e [24]. A abordagem combina a importante capacidade de exploração geral do algoritmo genético com o método Taguchi, que pode explorar a prole preterida [23]. Portanto, é mais provável que o algoritmo proposto identifique uma solução quase ótima e evite soluções ótimas locais [22]. No HTGA, a metodologia Taguchi substitui a operação de cruzamento, sendo assim o HTGA pode convergir rapidamente e ser mais robusto e estatisticamente correto [21]. Verifica-se em [22] e [23], a inserção do modelo Taguchi entre as operações de crossover e de mutação de um GA, identificado como o fluxo A-B, elegendo ordenadamente os melhores genes para alcançar crossover e, conseqüentemente, aprimorando o algoritmo genético. A Fig. 1 descreve as etapas de processamento do algoritmo HTGA, proposto em [23] para a resolução do modelo matemático.

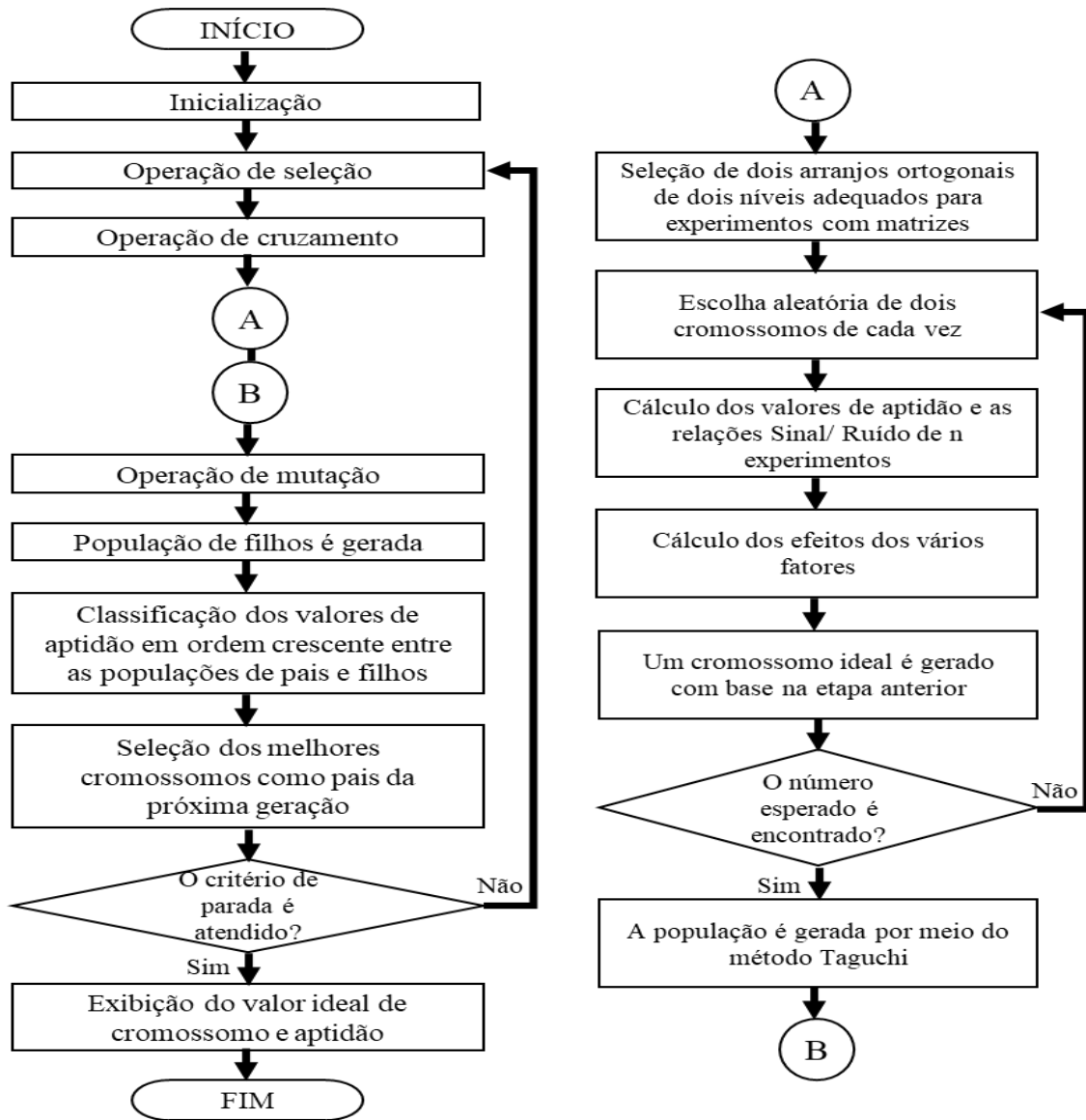


Fig. 1: HTGA para JSP adaptado de [23].

De forma geral, as etapas do HTGA, conforme em [23], podem ser descritas como:

- Fase 0: Definição dos parâmetros;
- Fase 1: Inicialização da população, onde a população inicial é gerada por meio de um algoritmo;
- Fase 2: Realização da operação de seleção das informações;
- Fase 3: Execução da atividade de cruzamento;
- Fase 4: Início da utilização do método Taguchi (A). Seleção de uma matriz ortogonal de dois níveis adequada para a matriz;
- Fase 5: Escolha aleatória de dois cromossomos para realizar experimentos matriciais;
- Fase 6: Cálculo dos valores de adequação e as relações de Sinal/ Ruído (S/ R) de n ensaios no arranjo ortogonal;
- Fase 7: Cálculo dos efeitos de vários fatores;
- Fase 8: Geração de um cromossomo ideal baseada na etapa 7;

- Fase 9: Repetição das fases 5 a 8 até a obtenção do valor pretendido;
- Fase 10: Obtenção da população via modelo Taguchi (B);
- Fase 11: Execução da operação de mutação;
- Fase 12: Geração de uma população de filhos;
- Fase 13: Classificação dos valores encontrados em ordem crescente entre os pais e populações descendentes;
- Fase 14: Seleção dos melhores cromossomos como pais da próxima geração;
- Fase 15: Verificação do atendimento do critério. Em caso positivo, dirige-se para a próxima etapa. Caso contrário, retorna-se para a fase 2;
- Fase 16: Exibição do cromossomo ideal e do valor preterido.

O detalhamento das etapas do HTGA pode ser visualizado em [23], pode-se destacar a forma de codificação da solução que baseia-se em uma sequência de tarefas e operações, onde cada gene significa cada operação de um trabalho, e um cromossomo pode ser decodificado em uma programação.

Outro aspecto importante verificado em [23] refere-se a geração da população inicial, onde uma nova série de cromossomos é gerada, sendo o esquema de codificação apresentado, uma ferramenta para reproduzir a população inicial alternando os genes aleatoriamente.

Além disso, observa-se que os arranjos ortogonais do método Taguchi são usados para estudar um grande número de variáveis de decisão com um pequeno número de experimentos, resultando nas melhores combinações de variáveis de decisão por meio da interação entre as matrizes ortogonais e as relações sinal-ruído. Também pode ser visualizado em [23], a geração de descendência por operação cruzada, a qual é utilizada para originar dois novos cromossomos de alta diversidade empregando um cromossomo e o método *swap-change*, alia-se a isto o fato de cada cromossomo ser dividido em seções para conduzir a matriz ortogonal do método Taguchi. O procedimento *swap-change* também é utilizado na operação de mutação, conforme em [23]. Seguindo em [23], salienta-se que o método Taguchi é inserido entre as operações de crossover e mutação, selecionando assim os melhores genes para atingir o crossover e desta maneira aperfeiçoar o algoritmo genético.

O sequenciamento do algoritmo híbrido HTGA possui a base algorítmica descrita como função *FINDSCHEDULE* [25]. Durante a programação, a sequência de tarefas é planejada de forma individual para que todas as restrições sejam atendidas e o tempo de produção de cada atividade seja minimizado. O número total de tarefas é representado por N_j . O algoritmo de cronograma determina os tempos de partida $t_{j,n}$ e as durações $d_{j,n}, \forall j, n = 1, \dots, N_{T,j}$ para todas as atividades e todas as tarefas. As durações $d_{j,n}$ das operações são delimitadas pelos limites inferiores $\underline{d}_{j,n}$ e superiores $\bar{d}_{j,n}$ sendo estes especificados pelo plano produtivo.

O algoritmo também atribui tarefas às máquinas. Para evitar uma atribuição inicial de tarefas a um dos equipamentos, o que poderia levar à soluções não ótimas, os chamados tipos de máquinas m são definidos. Registra-se para cada máquina, a utilização da capacidade $c_m(t)$ ao longo do tempo. Desta forma, a função *FINDSCHEDULE* () é resumida no algoritmo descrito na Fig. 2, onde a mesma procura agendar as tarefas de acordo com a estratégia mais cedo e mais curto possível, tão tarde quanto mais necessário.

função $[t_{j,n}, d_{j,n}, err] = \text{FINDSCHEDULE}(j, n, t_{rl}, t_{lst})$

$m = \text{tipo de máquina}(j, n);$
 $[\underline{d}, \bar{d}] = \text{duração da atividade}(j, n);$
 $N_{T,j} = \text{número de atividades}(j);$
 $[t_1, t_2] = \text{FINDSLOT}(m, t_{rl}, \underline{d});$
 $d_{j,n} = \underline{d};$
 $err = 0;$

enquanto 1 faz
se $t_{lst} \geq t_1$ **então**
se $n = N_{T,j}$ **então**
 $t_{j,n} = t_1;$
retorno
outro

$[t_{j,n+1}, d_{j,n+1}, err_{+1}] =$
 $\text{FINDSCHEDULE}(j, n + 1, t_1 + \underline{d}, t_2);$
se $err_{+1} = 0$ **então**
se $t_{lst} \geq (t_{j,n+1} - \underline{d})$ **então**
 $t_{j,n} = t_{j,n+1} - \underline{d};$
outro se $t_{lst} \geq (t_{j,n+1} - \bar{d})$ **então**
 $t_{j,n} = t_{lst};$
 $d_{j,n} = t_{j,n+1} - t_{lst};$
outro
 $t_{j,n} = t_{j,n+1} - \bar{d};$
 $err = 1;$
fim se
retorno
outro
 $[t_1, t_2] = \text{FINDSLOT}(m, \text{MAX}(t_{j,n+1} - \bar{d}, t_2), \underline{d});$
enquanto $t_2 < t_{j,n+1}$ **faz**
 $[t_1, t_2] = \text{FINDSLOT}(m, t_2, \underline{d});$
fim enquanto
fim se
fim se
outro
 $t_{j,n} = t_1;$
 $err = 1;$
retorno
fim se
fim enquanto
fim da função

Fig. 2: Função *FINDSCHEDULE* () adaptado de [25].

Conforme em [25], na função *FINDSCHEDULE* () geralmente não são nomeados índices para as variáveis de tarefas locais e os tempos de partida $t_{j,n}$ e as durações $d_{j,n}$ possuem índices j e n . Inicia-se com o número de tarefa j , o número da tarefa n , a primeira hora de início possível ou hora de lançamento t_{rl} e a última hora de início admissível t_{lst} . Os valores retornados da função são a hora de início $t_{j,n}$ e a duração $d_{j,n}$ da atividade n da tarefa j , bem como um sinalizador de erro err .

Para garantir que o novo intervalo de tempo livre $[t_1, t_2)$ alcance ao menos até o $t_{j,n+1}$ da atividade subsequente $n + 1$, ou seja, $t_2 \geq t_{j,n+1}$, um mecanismo de looping é instalado. Quando este possui a condição $t_2 < t_{j,n+1}$ solicita-se a função *FINDSLOT* () com o tempo de liberação t_2 até encontrar um novo intervalo de tempo livre $[t_1, t_2)$, satisfazendo a condição $t_2 \geq t_{j,n+1}$. Desta maneira, o novo $[t_1, t_2)$, repete o looping externo. Essa é a única situação em que o mecanismo externo e as funções não são concluídas. Determina-se uma programação viável com tempo de término mínimo para a tarefa j por intermédio da função *FINDSCHEDULE* (). O período entre a hora de início da primeira tarefa e a hora de término da última atividade define o *makespan* (T_m) de todo o lote produtivo.

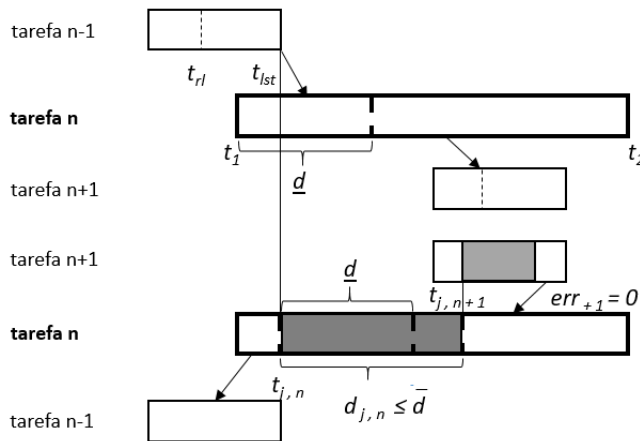


Fig. 3: Exemplo de gráfico de sequenciamento de uma tarefa n pela função *FINDSCHEDULE* () adaptado de [25].

C. Validação do modelo apresentado

A partir do momento da apresentação da metodologia a ser utilizada, torna-se necessário validar a mesma para que os resultados alcancem confiabilidade e integridade das informações. Com base em modelos da literatura, o algoritmo é testado para verificação dos valores da função objetivo. Esta etapa é fundamental, uma vez que visa encontrar os valores que artigos da literatura obtiveram, de modo a concluir a validade do algoritmo proposto. O desempenho do presente algoritmo, com relação ao modelo matemático apresentado é comparado com as soluções obtidas pelo modelo matemático proposto por [7] chamado de MILP-1. Os dados utilizados no teste também são os apresentados pelos autores.

As soluções ideais do FJSP podem consumir um elevado tempo e resultar na impossibilidade de execução pelos softwares, principalmente nos problemas de médio e grande portes. Estudos concentram esforços no desenvolvimento de metodologias heurísticas para solucionar estas situações. Os modelos testados foram: FJSPs de tamanho pequeno (SFJS1-SFJS5) e FJSPs de tamanho médio e grande porte (MFJS1-MFJS5). Os problemas de teste são definidos por suas dimensões (i, j, k) , no qual os índices indicam o número de tarefas, o número de operações para todas as tarefas e o número de máquinas, respectivamente.

Neste artigo propõe-se a utilização do modelo matemático proposto em [7], sendo sua resolução por meio do algoritmo híbrido HTGA identificado em [23], para a busca de soluções otimizadas. Aqui são comparados apenas os valores da função objetivo. Para o modelo criado de GA são relevantes avaliar o tempo computacional e o critério de parada do algoritmo. Sendo assim, formularam-se os modelos chamados de GAJS – config. 1 e GAJS – config. 2 para possibilitar as simulações de resolução dos problemas testes. Sendo que o primeiro método utiliza uma mutação fixa em 2% e o segundo uma configuração com taxa de mutação adaptativa. Desta maneira, segue-se a proposta apresentada em [8]. Esta função faz com que os indivíduos com menor aptidão tenham maior

probabilidade de sofrer mutação, ou seja, visa-se melhorar a busca por pontos ótimos, uma vez que os indivíduos de baixa aptidão podem melhorar o seu papel na busca. Sabe-se que a alteração de parâmetros genéticos podem alterar consideravelmente o resultado [26]. Para isso, é inserida na Fase 11 do HTGA.

IV. RESULTADOS

Como apresentado na metodologia, o modelo matemático foi criado de forma a obter a melhor solução na minimização de tempo de ocupação da linha. Nesta seção apresentam-se os resultados obtidos a partir de testes computacionais, principalmente na comparação dos resultados obtidos com o presente trabalho e de acordo com [20]. Esta é uma técnica de otimização que visa obter as melhores alternativas possíveis.

Salienta-se que a utilização de algoritmos metaheurísticos de rápida convergência pode fazer o método parar facilmente em um ponto de mínimo local, principalmente em problemas complexos, seja por utilizar variáveis discretas e/ou pelas discontinuidades da função objetivo. Por esta razão, tratando-se de um modelo com o GA, sugere-se a utilização de mutação adaptativa, fazendo com que o algoritmo busque outras regiões do espaço de soluções em identificação de convergência prematura. Para a construção do primeiro modelo apresentado, utiliza-se uma taxa de mutação de 2%. Já no segundo, aplica-se o conceito de mutação adaptativa.

A validação dos mesmos foi realizada por meio de comparações com o método ideal denominado MILP-1 [7]. Executaram-se cinco rodadas para cada problema e algoritmo, sendo que os métodos utilizados podem gerar diferentes soluções em cada ciclo. Nota-se que os algoritmos propostos alcançam a solução ideal nos problemas de tamanho pequeno (SFJS1:5) e o aumento do tamanho de problemas ocasiona a um maior desvio padrão dos algoritmos. Para a solução de problemas de maior porte, provavelmente podem ser necessários maiores números de gerações e indivíduos, onde constatou-se neste trabalho que para mutação no formato convencional, nem sempre se atingiu o ótimo global. Os resultados obtidos revelam que as soluções finais podem ser atingidas em um curto período.

Na Tab. 1 estão apresentados, para cada um dos dez problemas de sequenciamento, os valores da função objetivo dos três modelos, o número de gerações e dos indivíduos utilizados. Como mencionado anteriormente, o modelo GAJS – config. 1 é o algoritmo desenvolvido com taxa de mutação fixa de 2% e o GAJS – config. 2, possui taxa de mutação adaptativa. Apresenta-se na Tab. 1 os valores elencados por [7] para o modelo MILP-1. Somando-se a estas informações, tem-se o tamanho dos modelos, as variáveis inteiras e o número de restrições para cada problema testado. Definiu-se no presente estudo o número de gerações e de indivíduos para a realização dos testes.

Problema	Tamanho dos Modelos	Variáveis Inteiras	Número de Restrições	Função Objetivo (Cmax)			Gerações	Indivíduos
				Modelo MILP-1	GAJS - Config 1	GAJS - Config 2		
SFJS1	2,2,2	40	134	66	66	66	50	100
SFJS2	2,2,2	32	108	107	107	107	50	100
SFJS3	3,2,2	72	234	221	267	221	50	100
SFJS4	3,2,2	84	236	355	374	355	50	100
SFJS5	3,2,2	84	272	119	119	119	50	100
MFJS1	5,3,6	720	1829	468	468	468	50	100
MFJS2	5,3,7	840	1986	446	446	446	50	100
MFJS3	6,3,7	1260	2819	466	512	466	50	100
MFJS4	7,3,7	1617	3789	564	580	564	50	100
MFJS5	7,3,7	1617	3726	514	520	514	50	100

Tab. 1: Valores da funções objetivo GAJS – Config 1 e GAJS – Config. 2 e valores obtidos de [7] e [20].

Como pode ser verificado na Tab. 1, o desempenho do algoritmo GAJS - config. 2 apresentou resultados idênticos aos valores do modelo literário. Já o modelo GAJS – config. 1 não obteve os mesmos resultados nos problemas SFJS3, MFJS3, MFJS4 e MFJS5, os quais apresentaram valores acima dos demais métodos, não indicando assim equivalência ou melhoria em relação ao padrão.

Conforme mencionado anteriormente, o modelo GAJS – config. 1 utiliza uma taxa de mutação fixa de 2%, significando

uma multiplicação de forma aleatória do conjunto de dados ou parâmetros. Com a mutação adaptativa, aplicada no segundo método, ocorre uma verificação da variância dos dados e alguns indivíduos são multiplicados por fatores 1,2 ou 1,3 por exemplo, e não a um fator fixo de 1,02 como ocorre no GAJS – config. 1. Uma ilustração dos resultados obtidos nos modelos em relação aos problemas pode ser verificada na Fig. 4. Além de ser utilizado o método literário MILP-1.

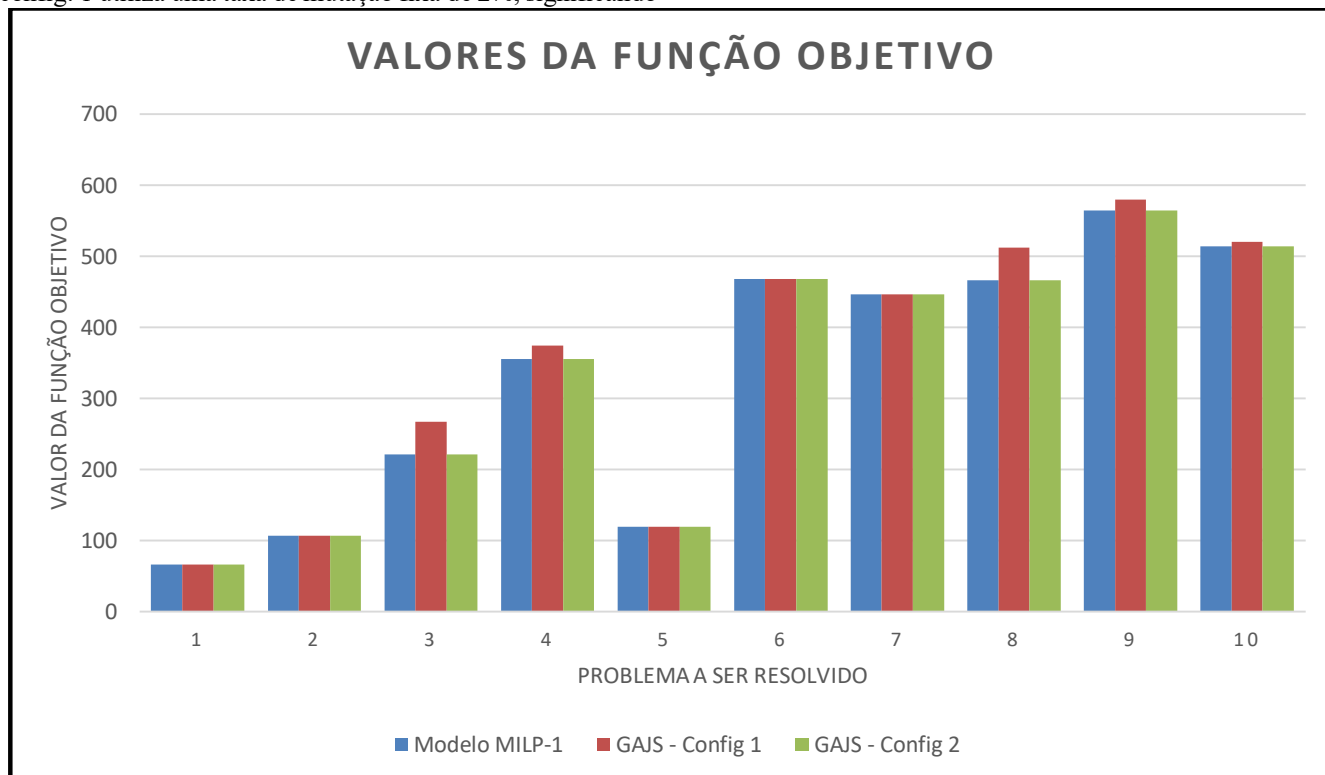


Fig. 4: Valores obtidos para a função objetivo dos modelos matemáticos.

A performance geral dos modelos propostos indicam resultados positivos em relação ao MILP-1. Observa-se que o modelo GAJS – config. 1 apresentou valores piores nos casos

3, 4, 8, 9 e 10. Já o modelo proposto GAJS – config. 2 atingiu os mesmos valores do modelo MILP-1.

Visando comparar o efeito da mutação adaptativa, para os algoritmos de GAJS – config. 1 e GAJS – config. 2, geraram-se os gráficos de convergência dos valores da função objetivo para cada um dos casos de otimização. Para os respectivos modelos, tem-se os valores da função objetivo ao longo das iterações ilustradas nas Figs. 5 e 6. Na Fig. 5 pode-se verificar que existe uma convergência dos valores da função objetivo,

ou seja, o valor estabiliza após um certo número de iterações. Esta figura representa os problemas-teste MFJS1 a MFJS5 para as configurações 1 e 2. Na Fig. 6, apresentam-se os problemas-teste SFJS1 a SFJS5, revelando-se que os valores apresentam convergência mais rápida, mostrando que explorar diferentes regiões do espaço de soluções pode acelerar a convergência para este tipo de problema.

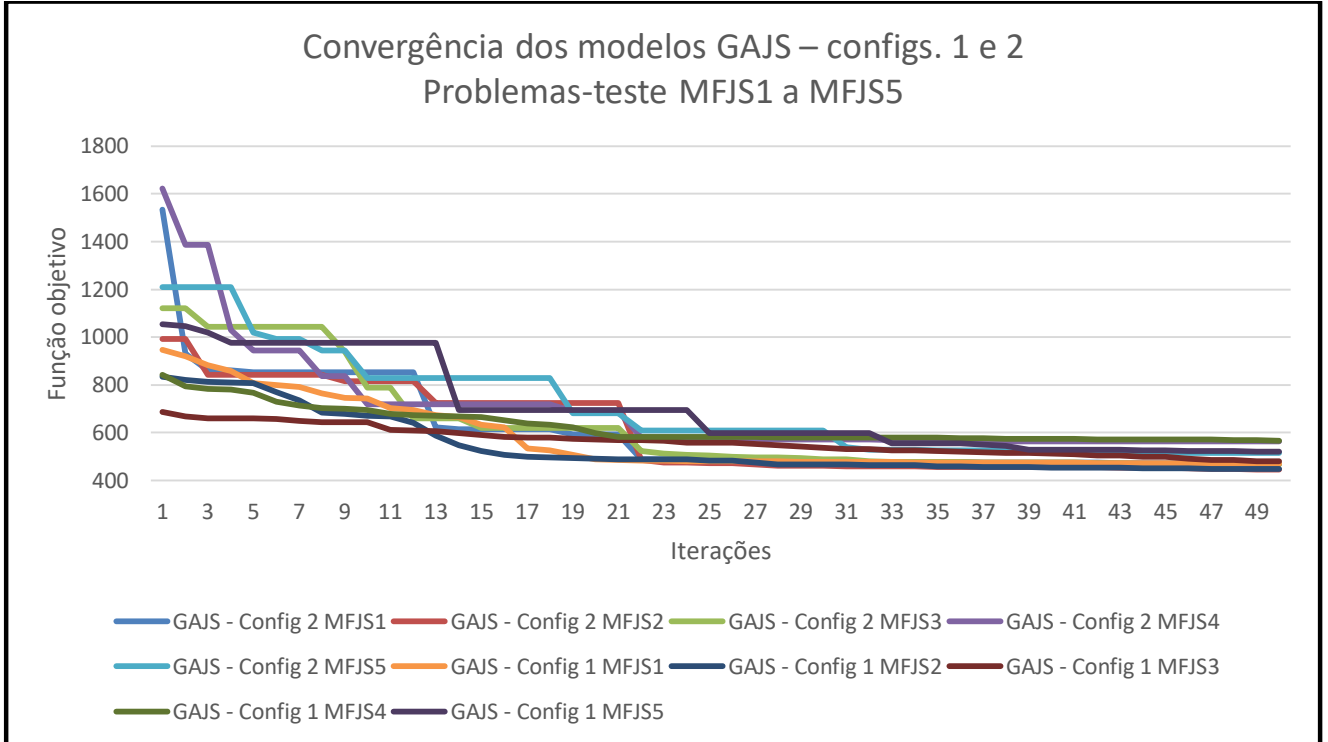


Fig. 5: Convergência dos modelos GAJS – configs. 1 e 2 para os problemas-teste MFJS1 a MFJS5.

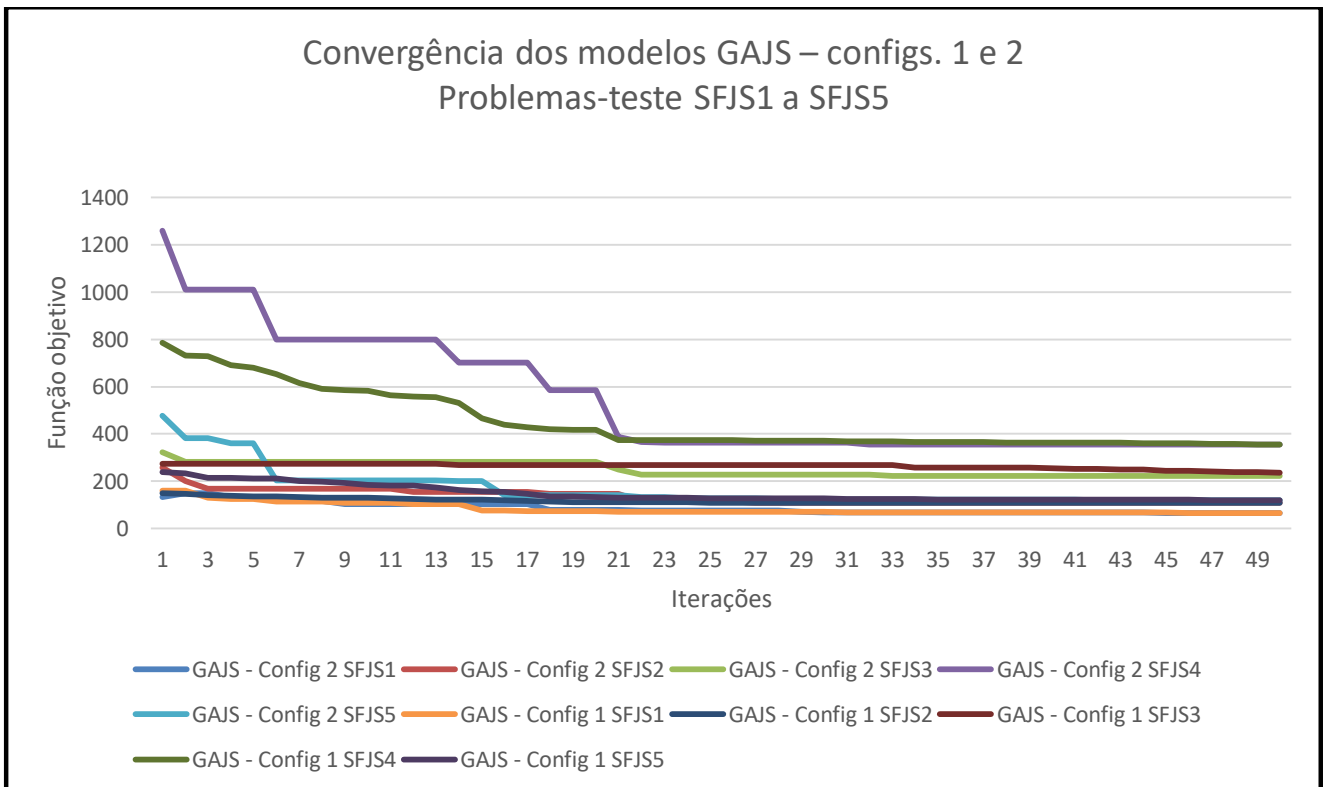


Fig. 6: Convergência dos modelos GAJS – configs. 1 e 2 para os problemas-teste SFJS1 a SFJS5.

Constatam-se nas Figs. 5 e 6 as comparações dos dois modelos sugeridos para os problemas MFJS1 a MFJS5 e SFJS1 a SFJS5, respectivamente. Na Fig. 5 as curvas apresentadas possuem comportamento similar, com exceção dos problemas MFJS2 para a GAJS – config. 1 e MFJS5 para a GAJS – config. 2. Na figura seguinte, observam-se basicamente dois grupos de resultados, sendo o primeiro formado por GAJS – config. 1 SFJS4 e GAJS – config. 1 SFJS 5. O segundo agrupamento contempla os demais problemas SFJS para os dois modelos propostos.

Desta forma observa-se que o modelo GAJS – config.2 apresenta resultados idênticos ao do modelo de literatura, validando-se a aplicação do mesmo para resolução de problemas de sequenciamento produtivo. Baseado nos resultados obtidos, torna-se possível validar a utilização do modelo matemático observado em [7] e de sua execução pela metodologia HTGA, na resolução de problemas de FJSPs, alcançando-se assim o objetivo deste artigo.

Podem ser verificados ganhos quanto a utilização destes métodos na realização de roteiros mais inteligentes, em uma otimização dos recursos produtivos, no aumento da disponibilidade fabril e em uma maior flexibilidade organizacional. Proporcionando-se também o emprego de conceitos de manufatura inteligente e de integração digital entre os programas de sequenciamento e os equipamentos.

V. CONCLUSÕES

A quarta revolução industrial nos traz reflexões quanto as inúmeras alterações digitais que ocorreram e as muitas outras que virão com o objetivo de maximizar os resultados e de integrar as tecnologias e os sistemas de transformação. Neste sentido, alguns desafios precisam ser resolvidos por meio de ferramentas que possibilitem a otimização dos objetivos almejados. Com isso, o presente estudo expõe a possibilidade de se utilizar um método híbrido para solução de problemas de sequenciamento produtivo.

Os resultados obtidos demonstram que a resolução de problemas de sequenciamento produtivo por meio do algoritmo evolutivo proposto é factível e também apresenta a confiabilidade esperada quando aplica-se este método. O desempenho pode ser verificado na utilização dos modelos aos problemas de teste com pequenos e grandes tamanhos. A validação do algoritmo proposto ocorreu a partir da comparação com o modelo denominado MILP-1 verificado em [7]. O modelo da literatura utiliza o método Simplex, comprovando a obtenção dos pontos de ótimo global.

Ao se comparar aos valores da literatura, obteve-se a mesma performance nos problemas SFJS1, SFJS2, SFJS5, MFJS1 e MFJS2 para o método GAJS – config. 1. Já na configuração GAJS – config. 2, verificam-se os mesmos resultados nos 10 casos de aplicação. Desta maneira, o algoritmo proposto GAJS – config. 2 que utiliza o conceito de mutação adaptativa, mostrou que é possível atingir a solução ideal ao apresentar os mesmos resultados do que o modelo literário MILP-1, validando-se desta forma a sua implementação, assim como a sua aplicação em casos similares.

Conforme já mencionado, este artigo utilizou como base de pesquisa outros estudos que tratavam de problemas de sequenciamento produtivo. Destacam-se os trabalhos [7] e

[23] dentre os materiais analisados. Comparando-se os resultados encontrados nestas duas obras em relação a este paper, observam-se similaridades nas soluções obtidas. Ressalta-se o desenvolvimento e validação do modelo MILP-1 em [7] e a hibridização do GA com o método Taguchi em [23], as quais visam obter alternativas para a solução de FJSPs. Da mesma maneira que a validação do modelo exposto neste estudo resultou em uma nova opção para a otimização de rotas e dos recursos disponíveis.

A pesquisa realizada teve como premissa a realização da validação de uma metodologia de otimização da manufatura, podendo esta ser utilizada em variados modelos de negócios. Desta forma, os resultados obtidos aqui, contribuirão para a resolução de problemas de planejamento das atividades em diferentes contextos organizacionais, assim como atenuarão os impactos provocados pela presença das restrições, sejam elas ligadas ao número de equipamentos, dos colaboradores ou de atividades concorrentes.

O presente artigo contribuiu com alternativas para a resolução de *Flexible Job Shop Schedule Problems* (FJSP). Todavia, faz parte de um ciclo de melhoria contínua onde podem ser observadas oportunidades de continuação do trabalho. Verifica-se, por exemplo, a possibilidade de serem realizados estudos para identificar o grau de integração deste modelo com os softwares de ERP. Também sugere-se a aplicação deste modelo em problemas reais de FJSP, inclusive sendo aplicável nos mais diversos segmentos.

VI. BIBLIOGRAFIA

- [1] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, no. 1, pp. 127-182, 2020.
- [2] J. Zhang et al., "Review of job shop scheduling research and its new perspectives under Industry 4.0," *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1809-1830, 2019.
- [3] A. S. Jain and S. Meeran, "Deterministic job-shop scheduling: Past, present and future," *European journal of operational research*, vol. 113, no. 2, pp. 390-434, 1999.
- [4] Z. Cao, C. Lin, M. Zhou, "A Knowledge-Based Cuckoo Search Algorithm to Schedule a Flexible Job Shop With Sequencing Flexibility," *IEEE Transactions on Automation Science and Engineering*, 2019.
- [5] N. Xie and N. Chen, "Flexible job shop scheduling problem with interval grey processing time," *Applied Soft Computing*, vol. 70, pp. 513-524, 2018.
- [6] A. A. Garcia-Leon, S. Dauzère-Pères, Y. Mati, "An efficient Pareto approach for solving the multi-objective flexible job-shop scheduling problem with regular criteria," *Computers & Operations Research*, vol. 108, pp. 187-200, 2019.
- [7] C. Özgüven, L. Özbakir, Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility," *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1539-1548, 2010.
- [8] H. M. Gomes and L. L. Corso, "A Hybrid Method for Truss Mass Minimization considering Uncertainties," *Mathematical Problems in Engineering*, vol. 2017, 2017.
- [9] W. Cui et al., "A hybrid genetic algorithm for non-permutation flow shop scheduling problems with unavailability constraints," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 9, pp. 944-961, 2016.
- [10] N. Kundakci and O. Kulak, "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem," *Computers & Industrial Engineering*, vol. 96, pp. 31-51, 2016.
- [11] A. Dolgui et al., "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications," *International Journal of Production Research*, vol. 57, no. 2, pp. 411-432, 2019.
- [12] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, "The expected contribution of Industry 4.0 technologies for industrial performance," *International Journal of Production Economics*, vol. 204,

- pp. 383–394, 2018.
- [13] I. R. Uhlmann and E. M. Frazzon, “Production rescheduling review: Opportunities for industrial integration and practical applications,” *Journal of manufacturing systems*, vol. 49, pp. 186-193, 2018.
- [14] M. Ghobakhloo, “The future of manufacturing industry: a strategic roadmap toward Industry 4.0,” *Journal of Manufacturing Technology Management*, 2018.
- [15] P. K. Muhuri, A. K. Shukla, A. Abraham, “Industry 4.0: A bibliometric analysis and detailed overview,” *Engineering applications of artificial intelligence*, vol. 78, pp. 218-235, 2019.
- [16] J. Lee, I. Cameron, M. Hassall, “Improving process safety: What roles for Digitalization and Industry 4.0,” *Process Safety and Environmental Protection*, vol. 132, pp. 325-339, 2019.
- [17] L. Shen, S. Dauzère-Pèrès, J. S. Neufeld, “Solving the flexible job shop scheduling problem with sequence-dependent setup times,” *European Journal of Operational Research*, vol. 265, no. 2, pp. 503-516, 2018.
- [18] K. Tamssaouet, S. Dauzère-Pèrès, C. Yugma, “Metaheuristics for the job-shop scheduling problem with machine availability constraints,” *Computers & Industrial Engineering*, vol. 125, pp. 1-8, 2018.
- [19] A. Jalilvand-Nejad and P. Fattahi, “A mathematical model and genetic algorithm to cyclic flexible job shop scheduling problem,” *Journal of Intelligent Manufacturing*, vol. 26, no. 6, pp. 1085-1098, 2015.
- [20] P. Fattahi, M. S. Mehrabad, F. Jolai, “Mathematical modeling and heuristic approaches to flexible job shop scheduling problems,” *Journal of intelligent manufacturing*, vol. 18, no. 3, pp. 331-342, 2007.
- [21] C. Chen et al., “The hybrid Taguchi-Genetic algorithm for mobile location,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, pp. 489-563, 2014.
- [22] H. Chang et al., “Solving the flexible job shop scheduling problem with makespan optimization by using a hybrid Taguchi-genetic algorithm,” *IEEE access*, vol. 3, pp. 1740-1754, 2015.
- [23] P. Nikdel et al., “Improved Takagi–Sugeno fuzzy model-based control of flexible joint robot via Hybrid-Taguchi genetic algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 12-20, 2014.
- [24] T. Liu, J. Tsai, J. Chou, “Improved genetic algorithm for the job-shop scheduling problem,” *The International Journal of Advanced Manufacturing Technology*, vol. 27, no. 9-10, pp. 1021-1029, 2006.
- [25] A. Aschauer et al., “An efficient algorithm for scheduling a flexible job shop with blocking and no-wait constraints,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12490-12495, 2017.
- [26] L. L. Corso, A. L. Gasparin, H. M. Gomes, “A Reliability based design optimization using a genetic algorithm : application to bonded thin films areas of copper/polypropylene,” *Lume - Repositório Digital UFRGS Online*, vol. 24, no. 3, pp. 510-519, 2016. Available: <https://www.lume.ufrgs.br/handle/10183/143980>. Accessed: 22-Oct-2020.