

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MÉTODO DE ESCOLHA DE SCHEDULING PARA
PROBLEMAS DE JOB SHOP FLEXÍVEL UTILIZANDO
UM TOMADOR DE DECISÃO FUZZY**

DIANA MARIMOTO PRAUSE DA SILVA

ORIENTADOR: PROF. DR. ROBERTO SANTOS INOUE
COORIENTADOR: PROF. DR. EDILSON REIS RODRIGUES KATO

São Carlos - SP
Novembro/2022

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MÉTODO DE ESCOLHA DE SCHEDULING PARA
PROBLEMAS DE JOB SHOP FLEXÍVEL UTILIZANDO
UM TOMADOR DE DECISÃO FUZZY**

DIANA MARIMOTO PRAUSE DA SILVA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Roberto Santos Inoue

Coorientador: Prof. Dr. Edilson Reis Rodrigues Kato

São Carlos - SP
NOVEMBRO/2022



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado da candidata Diana Marimoto Prause da Silva, realizada em 11/11/2022.

Comissão Julgadora:

Prof. Dr. Roberto Santos Inoue (UFSCar)

Profa. Dra. Juliana Keiko Sagawa (UFSCar)

Prof. Dr. Glauco Augusto de Paula Caurin (USP)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

AGRADECIMENTO

Agradeço primeiramente aos Professores Edilson Kato e Roberto Inoue, por toda a orientação e paciência, muito obrigada.

Agradeço também a minha família por estarem sempre ao meu lado me apoiando. E também aos meus amigos e colegas de laboratório que me acompanharam e me acolheram nesse período que passei em São Carlos.

RESUMO

Dentre os problemas de programação encontrados na produção, há o *job shop* flexível (FJSP), que é uma extensão do clássico *Job Shop*. O FJSP é classificado com um problema NP-Difícil, e pode ser descrito como um conjunto de tarefas ou *jobs*, formados por um determinado número de operações que podem ser alocadas em um conjunto predeterminado de máquinas, com seus respectivos tempos de execução. A maneira em que as operações são distribuídas nas máquinas influencia diretamente na eficiência da programação. Eficiência essa que pode ser mensurada e otimizada através de vários critérios de desempenho, ou seja, vários objetivos ao mesmo tempo. A programação da produção gerada por um algoritmo de otimização multiobjetivo (MOFJSP), os quais consideram principalmente três critérios de desempenho, sendo eles: tempo de conclusão de todas as operações (Makespan), carga atribuída à máquina mais carregada e a soma da carga de todas as máquinas, não apresenta uma única solução ótima global, e sim um conjunto de soluções não dominadas e dominadas, chamado de conjunto de Pareto. As soluções deste conjunto de Pareto são soluções ótimas ou perto da ótima, sendo consideradas boas soluções pois conseguem gerar uma diversidade de representações da programação da produção, por exemplo, na forma de gráficos de Gantt. Para que os tomadores de decisão escolham qual a melhor solução da programação da produção dentre as possíveis encontradas pelo MOFJSP outras variáveis podem ser levadas em consideração, tais como a maximização ou minimização da ociosidade das máquinas, da carga das operações em uma máquina etc. as quais podem fornecer uma maior aderência da decisão em vista as necessidades do sistema produtivo. Para a inclusão destas variáveis e a seleção da melhor programação de produção entre as fornecidas pelo conjunto de Pareto é proposto a utilização de um algoritmo tomador de decisão baseado na Técnica para Ordem de Preferência por Semelhança com a Solução Ideal em um ambiente Fuzzy, chamado Fuzzy-TOPSIS. Esse tomador de decisão pode ponderar variáveis que não foram contempladas no algoritmo MOFJSP e auxiliar na tomada de decisão da melhor programação de produção dentre as ótimas ou perto das ótimas obtidas. Nos resultados, foi possível notar que os valores obtidos com a aplicação do algoritmo proposto para esse problema obtiveram resultados próximos aos esperados, de acordo com as variáveis analisadas, podendo ser uma ferramenta importante no auxílio à tomada de decisão em um sistema produtivo.

Palavras-chave: FJSP, Colônia de Abelhas Artificial, ABC, otimização multiobjetivo, Fronteira de Pareto, Tomador de Decisão, Fuzzy-TOPSIS.

ABSTRACT

Among the scheduling problems encountered in production, there is the flexible job shop (FJSP), which is an extension of the classic Job Shop. The FJSP is classified as an NP-Hard problem and can be described as a set of jobs, formed by a certain number of operations that can be allocated on a predetermined set of machines, with their respective execution times. The way operations are distributed across machines directly influences programming efficiency. Efficiency that can be measured and optimized through several performance criteria, that is, several objectives at the same time. The production schedule generated by a multi-objective optimization algorithm (MOFJSP), which mainly considers three performance criteria, namely: completion time of all operations (Makespan), load assigned to the most loaded machine and the sum of the load of all machines, does not present a single global optimal solution, but a set of non-dominated and dominated solutions, called the Pareto set. The solutions of this Pareto set are optimal or close to optimal solutions, being considered good solutions because they can generate a diversity of representations of the production schedule, for example, in the form of Gantt charts. For decision makers to choose the best production scheduling solution among the possible ones found by MOFJSP, other variables can be taken into account, such as maximizing or minimizing machine idleness, the load of operations on a machine, etc. which can provide greater adherence to the decision in view of the needs of the production system. For the inclusion of these variables and the selection of the best production schedule among those provided by the Pareto set, it is proposed to use a decision-making algorithm based on the Technique for Order of Preference by Similarity with the Ideal Solution in a Fuzzy environment, called Fuzzy - TOPSIS. This decision maker can weight variables that are not contemplated in the MOFJSP algorithm and assist in the decision making of the best production schedule among the optimal ones or close to the optimal ones obtained. In the results, it was possible to notice that the values obtained with the application of the algorithm proposed for this problem obtained results close to the expected ones, according to the variables analyzed, and can be an important tool in aiding decision-making in a production system.

Keywords: FJSP, Artificial Bee Colony, multiobjective optimization, Pareto Frontier, Decision Maker, Fuzzy-TOPSIS.

SUMÁRIO

CAPÍTULO 1	10
INTRODUÇÃO.....	10
1.1 Objetivos	12
1.2 Organização do Trabalho	13
CAPÍTULO 2	15
CONCEITOS BÁSICOS	15
2.1 Planejamento e controle de produção	15
2.1.1 Job Shop Flexível.....	17
2.2 Problemas de Otimização Multiobjetivos	19
2.2.1 Ótimo de Pareto	20
2.2.2 Métodos de otimização para problemas multiobjetivos.....	21
2.3 Algoritmo Colônia de Abelhas Artificial.....	22
2.4 Tomador de Decisão	24
2.5 Fuzzy TOPSIS	26
CAPÍTULO 3	32
REVISÃO BIBLIOGRÁFICA	32
3.1 Métodos de resolução multiobjetivo do Job Shop Flexível	32
3.1.1 Soluções com o Algoritmo Colônia Artificial de Abelhas	33
3.1.2 Implementações focados na diversidade dos resultados	35
3.2 Tomadores de decisão para problemas multiobjetivos	37
3.2.1 Tomadores de Decisão	38
3.2.2 Fuzzy-TOPSIS para problemas multicritérios	39
CAPÍTULO 4	41
PROPOSTA DO TRABALHO	41

4.1 - O algoritmo de resolução da programação da produção FJSP Multiobjetivo e o algoritmo DM Fuzzy-TOPSIS.....	42
4.1.1 Algoritmo Colônia Artificial de Abelhas.....	42
4.1.2 Configurações iniciais.....	43
4.1.3 Geração das soluções iniciais.....	44
4.1.4 Abelhas empregadas e expectadoras.....	46
4.1.5 Arquivo de Pareto	47
4.1.6 Abelha exploradora.....	47
4.1.7 Implementação do Tomador de decisão Fuzzy-TOPSIS	48
4.1.8 Variáveis Linguísticas e Critérios de Decisão	49
4.1.9 Normalização das Soluções	51
4.1.10 Criação da Matriz de Decisão Fuzzy	52
4.1.11 Definir FPIS e FNIS.....	54
4.1.12 Cálculo da Distância Euclidiana e do Coeficiente de Proximidade.....	54
4.2 Validação dos resultados.....	55
4.3 Resultados esperados	56
CAPÍTULO 5	57
RESULTADOS	57
5.1 Cenário 1: Prioridade para menor ociosidade entre operações.....	62
5.1.1 Kacem 4x5 caso 1	64
5.1.2 Kacem 4x5 caso 2	67
5.1.3 Kacem 8x8 caso 2	70
5.1.4 Kacem 8x8 caso 5	73
5.1.5 Kacem 10x10 caso 2	76
5.1.6 Kacem 10x10 caso 4	79
5.1.7 Bradimarte MK2 caso 1	82
5.1.8 Bradimarte MK2 caso 2	85
5.1.9 Bradimarte MK4 caso 3	88
5.1.10 Bradimarte MK4 caso 4	91
5.2 Cenário 2: Prioridade para maior ociosidade entre operações.....	94
5.2.1 Bradimarte MK2 caso 1	95
5.2.2 Bradimarte MK2 caso 2	98
5.2.3 Bradimarte MK4 caso 3	101
5.2.4 Bradimarte MK4 caso 4	104

5.2.5 Bradimarte MK8 caso 5	107
5.2.6 Bradimarte MK8 caso 6	112
5.3 Discussão Final	117
CONCLUSÕES	119
6.1 Trabalhos Futuros	120
REFERÊNCIAS BIBLIOGRÁFICAS.....	121

CAPÍTULO 1

INTRODUÇÃO

A programação da produção ou *Scheduling*, é um importante processo na maioria dos sistemas de manufatura e produção, pois trata da alocação dos recursos, geralmente limitados, e atividades a serem realizadas (PREVIERO, 2016). Dentre esses problemas temos o Problema do *Job Shop* Flexível (FJSP – do inglês *Flexible Job Shop Problem*), que é de uma extensão do clássico *Job Shop* (BRANDIMARTE, 1993).

Segundo Chang et al. (1989) o FJSP é caracterizado por possuir um conjunto tarefas, também chamadas de *jobs* que devem ser atribuídos a um conjunto de máquinas. Cada *job* possui um determinado número de operações e cada operação pode ser realizada em um subconjunto previamente especificado de máquinas, podendo assim cada operação ter um tempo de execução diferente em cada máquina desse subconjunto.

Um fator importante que deve ser observado é como cada operação será alocada nas máquinas e como qual será a ordem de execução das tarefas que compõem o FJSP, pois a distribuição desses recursos está diretamente ligada a eficiência da programação executada.

O problema de *job shop* flexível, quando tratado como um problema de otimização multiobjetivo, comumente tem como objetivos de otimização e parâmetros de avaliação da sua eficiência os seguintes critérios de desempenho: tempo de conclusão de todas as operações (*Makespan*), carga atribuída a máquina mais carregada (*Maximum Workload*) e a soma da carga de todas as máquinas (*Total Workload*).

Além disso, o FJSP é considerado como um problema NP-difícil (GAREY et al., 1976), que faz com que uma abordagem heurística ou meta-heurística possa ser a mais adequada para resolvê-lo devido aos altos custos computacionais desse tipo de problema.

Na literatura encontramos diversos trabalhos de autores que para resolver problemas multiobjetivos de FJSP utilizam abordagens heurísticas e meta heurísticas como: Algoritmos Genético (GA, do inglês *Genetic algorithm*) (HUANG et al. 2018), Otimização por Enxame de Partículas (PSO, do inglês *Particle swarm optimization*) (HAUNG et al. 2016, KATO et al.

2018), *Tabu Search* (TS) (JIA E HU, 2014, KEFALAS et al. 2019), entre outros métodos, incluindo o Algoritmo Colônia de Abelhas Artificial (ABC, do inglês *Algorithm bee colony*) (LI et al., 2011, LI et al., 2014, MENG et al., 2020, CAO e SHI, 2019). Estas técnicas, vêm apresentando bons resultados considerando a aplicação ao FJSP e fornecem números absolutos, de acordo com os critérios de otimização utilizados, os valores ótimos ou perto do ótimo encontrados até o momento.

Esses valores são validados utilizando o gráfico de Gantt para a alocação no tempo das operações dos Jobs nas máquinas. Dessa forma, o mesmo valor absoluto “ótimo, ou perto do ótimo” pode representar gráficos de Gantt diferentes, sugerindo as alternativas viáveis de se implementar a programação da produção no chão de fábrica.

Essas alternativas são úteis frente as condições atuais do sistema de produção, onde algumas variáveis, tais como: máquinas ociosas, parada de alguma máquina, falta de operador para uma determinada máquina, dentre outras, não foram contempladas anteriormente no modelo de busca da melhor solução. No entanto, alternativas do gráfico de Gantt (programação da produção) dos melhores objetivos absolutos encontrados, podem contemplar essas ocorrências e fornecer uma solução alternativa que mantenha os mesmos valores absolutos encontrados anteriormente, diminuindo ou eliminando qualquer impacto dessas variáveis não contempladas.

Tratando de problemas de otimização multiobjetivo a aplicação da meta heurística não resultará em apenas uma solução ótima global, mas sim em um conjunto de soluções ótimas, conhecido como Fronteira de Pareto, que representa as melhores respostas para o problema (DEB,2001). No entanto, Deb (2001) ressalta que as soluções estando ou não na Fronteira de Pareto são importantes, pois quando se trata da otimização de problemas multiobjetivo é necessário buscar as soluções próximas à Fronteira de Pareto para que haja uma diversidade opções possíveis e de qualidade da programação da produção a ser implementada.

Por gerar um conjunto de soluções absolutas viáveis, tanto as ótimas quanto as próximas do ótimo, um conjunto de Pareto irá contemplar essas soluções, permitindo que uma quantidade maior ainda de gráficos de Gantt de programações possíveis seja gerada. Uma solução no conjunto de Pareto pode se remeter à vários gráficos de Gantt com uma solução de programação da produção viável, ou seja, há um aumento não polinomial das soluções possíveis de serem empregadas pelo tomador de decisão, o que pode ser um algo que demandará um certo esforço.

A decisão de qual programação da produção será a escolhida para o sistema produtivo leva em consideração os valores absolutos encontrados e sua melhor disposição em um gráfico de Gantt, que é a ferramenta utilizada no chão de fábrica para a programação da produção, para o momento atual do sistema produtivo. Essa escolha é, muitas vezes, influenciada por variáveis locais, inerentes ao chão de fábrica que fazem com que determinados gráficos de Gantt da

programação da produção sejam melhores que outro para o momento. Escolher a melhor programação de produção de acordo com as necessidades reais e atuais do sistema produtivo sem alterar os melhores valores (objetivos) de programação da produção encontrados na forma absoluta, permite que metas do sistema produtivo, como qualidade, data de entrega, custos etc, não sejam alterados, garantido assim o melhor desempenho do sistema produtivo.

Neste contexto, para auxiliar nessa escolha pode-se usar como recurso um tomador de decisão (DM, do inglês *Decision Maker*). O DM realiza o processo de selecionar uma possível solução a partir de todas as alternativas disponíveis. Na maioria dos problemas, a multiplicidade de critérios para julgar as alternativas é generalizada, ou seja, para muitos desses problemas, o tomador de decisão deseja satisfazer mais de um objetivo ao selecionar a solução enquanto atende as restrições ditadas pelo ambiente, processos e recursos (HWANG e MASUD, 1979).

As dificuldades decorrentes do processo de tomada de decisão em um ambiente multiobjetivo tornam necessária a definição de uma metodologia capaz de atender as preferências do DM (FERREIRA et al. 2007), e na literatura é possível encontrar métodos de tomada de decisão que satisfazem essas condições de preferência, tais como: Processo de Hierarquia Analítica (AHP, do inglês *Analytic hierarchy process*)(SAATY, 2014), a Técnica para ordem de preferência por semelhança com a solução ideal (TOPSIS, do inglês *Technique for similarity preference order*) proposta por Hwang e Yoon (1981), a Lógica Fuzzy (CHEN E HWANG, 1992), além da combinação desses métodos, gerando o Fuzzy-TOPSIS (CHEN, 2000, LUUKKA, 2011, KUMAR et al. 2018), o Fuzzy-AHP (van LAARHOVEN e PEDRYCZ, 1983), o TOPSIS –AHP (CHAKLADAR e CHAKRABORTY, 2008), dentre outros.

Diante dos problemas apresentados e das metodologias que podem ser utilizadas para resolver um FJSP multiobjetivo, um tomador de decisão para o auxílio da escolha da melhor programação da produção, na forma de um gráfico de Gantt, pode contemplar não só os melhores valores absolutos dos objetivos (especificados por um sistema de programação da produção), mas também as condições reais e atuais do sistema produtivo no momento de sua implementação.

1.1 Objetivos

O principal objetivo desse trabalho é implementar um método de escolha da melhor programação da produção de um FJSP multiobjetivo (MOFJSP). O método utiliza o conjunto de Pareto resultante de um programador da produção MOFJSP e um algoritmo tomador de decisão que leva em consideração as soluções disponíveis na fronteira ou perto da fronteira de Pareto,

encontradas pelo programador da produção, e das condições atuais do chão de fábrica e/ou necessidades dos cenários de funcionamento de um sistema produtivo.

O conjunto de Pareto é fornecido pelo FJSP – ABC, isto é, um programador da produção multiobjetivo utilizando uma meta-heurística com estratégias de busca diversas, baseada na Colônia de Abelhas Artificiais (ABC), objetivando gerar uma alta diversidade de resultados na fronteira ou perto da fronteira de Pareto (FLORÊNCIO,2019). Os objetivos foram o de minimização de tempo de conclusão de todas as operações (Makespan), minimização da carga atribuída a máquina mais carregada e minimização da carga total de todas as máquinas.

O algoritmo de um Tomador de Decisão (DM) baseado em Fuzzy-TOPSIS, deve escolher, a partir das programações da produção geradas no algoritmo programador da produção FJSP – ABC, a melhor programação de acordo com o cenário atual do sistema produtivo. O DM utiliza variáveis tratadas no FJSP – ABC e outras variáveis incorporadas de acordo com o cenário da produção (relacionados às máquinas), permitindo que a tomada de decisão seja mais aderente com as condições atuais, do momento, do sistema produtivo. Nesse trabalho será tratada a variável incorporada de ociosidade de cada máquina, ou seja, a ideia é minimizar ou maximizar o tempo ocioso entre as operações de cada máquina.

1.2 Organização do Trabalho

Visando atingir o objetivo proposto, foi realizada uma pesquisa bibliográfica abordando os principais temas envolvidos, como os Problemas de Planejamento, Controle de Produção, o Problema do Job Shop Flexível, problemas multiobjetivos, Algoritmo Colônia Artificial de Abelhas, tomadores de decisão e o método de escolha Fuzzy-TOPSIS. A estrutura organizacional do trabalho é apresentada na sequência.

No Capítulo 2 são apresentados os conceitos sobre Planejamento, Controle de Produção, com foco no problema do *Job Shop* Flexível, apresentando sua definição, restrições de implementação e seus critérios de desempenho. A seguir é abordado problemas multiobjetivos e alguns detalhes da meta-heurística ABC proposta para resolvê-lo. Neste Capítulo também é apresentado as definições de tomadores de decisão e os principais conceitos envolvendo o tomador de decisão Fuzzy-TOPSIS. No Capítulo 3 está o levantamento e análise de trabalhos encontrados na literatura com temas e métodos de resolução, iguais e semelhantes aos abordados nesse trabalho. O Capítulo 4 contém a proposta desse trabalho, apresentando quais métodos de validação serão utilizados e quais os resultados esperados. O Capítulo 5 apresenta os resultados

encontrados das implementações propostas nos Capítulo 4. Por fim, o Capítulo 6 apresenta as conclusões desse trabalho.

CAPÍTULO 2

CONCEITOS BÁSICOS

Como o FJSP abordado nesse trabalho se trata tanto de um problema de otimização multiobjetivo como um problema NP-difícil, neste capítulo será apresentado os conceitos de otimização de problemas multiobjetivos, também será explorado a utilização da meta heurística da colônia artificial de abelhas como opção para resolução do problema. Além disso, será apresentado os conceitos do tomador de decisão e da utilização da técnica Fuzzy-TOPSIS como solução para de tomadores de decisão multiobjetivo, o qual é objetivo desse trabalho.

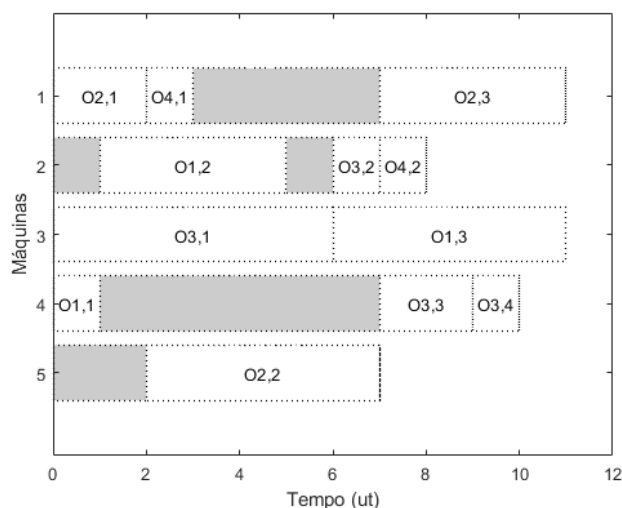
2.1 Planejamento e controle de produção

O Planejamento e controle da produção (PCP) é a atividade responsável pela organização dos dados e pelas tomadas de decisões considerando todos os fatores do processo produtivo. O PCP tende a definir qual a melhor maneira de coordenar e aplicar os recursos da empresa de modo a atender os planos estabelecidos nos níveis estratégico, tático e operacional (TUBINO, 2007).

Planejamento e controle, segundo Slack, Chambers e Johnston (2006) requer a conciliação do fornecimento e da demanda em termos de volume e tempo e para isso são desempenhadas três atividades distintas embora integradas, que são: carregamento, sequenciamento e programação.

O carregamento é a quantidade de trabalho alocado em um centro de trabalho, levando em consideração e tempo útil disponível de operação da máquina. Após a etapa do carregamento deve-se definir a ordem em que as tarefas serão executadas. Essa atividade é denominada sequenciamento. Para determinar as prioridades dadas na execução do trabalho, são definidas algumas regras, sendo algumas delas: a prioridade ao consumidor, o prazo de entrega, o LIFO (último a entrar, primeiro a sair, do inglês *last in, first out*) e o FIFO (primeiro a entrar, primeiro a sair, do inglês *first in, first out*). Tendo determinado a sequência em que o trabalho será desenvolvido, algumas operações requerem um cronograma detalhado, mostrando em que momento os trabalhos devem começar e quando eles devem terminar. Isto é uma programação da produção (SLACK et al., 2006).

Figura 2.1 - Representação do gráfico de Gantt



Fonte: Elaborado pelo autor

Um dos métodos mais utilizados para representar a programação é o gráfico de Gantt, nele é possível observar quando cada trabalho/produto (*job*) está programado para começar e terminar, assim como seu grau de acabamento, além do tempo ocioso de cada máquina. (SLACK et al., 2006). No gráfico de Gantt a operação de um determinado *job* é representada por O_{ij} , sendo $j = \{1, 2, 3 \dots k\}$ a operação do *job* i e $i = \{1, 2, 3 \dots n\}$, n o número máximo de *jobs* e k o número máximo de operações de cada *job* que compõem a programação, as operações O_{ij} são alocadas nas máquinas dispostas no eixo vertical e possuem um tempo de execução indicadas pelo eixo horizontal em unidades de tempo (ut). Por exemplo, a Figura 2.1, ilustra um gráfico de Gantt de uma possível programação composta por quatro *jobs*, sendo o primeiro *job* formado por três operações, o segundo por três operações, o terceiro por quatro operações e o quarto *job* por duas

operações, somando um total de doze operações que são distribuídas entre as cinco máquinas disponíveis.

Ainda no contexto da programação, pode haver restrições quando ao layout da indústria, as tecnologias disponíveis nos trabalhos e os objetivos de *scheduling* especificados (MACCARTHY e LIU, 1993). Essas restrições são determinadas principalmente pelo padrão de fluxo dos trabalhos nas máquinas, com base nisso Maccarthy e Liu (1993) apresentaram a seguinte classificação para os problemas clássicos de programação:

- *Job Shop*: cada *job* tem seu próprio padrão de fluxo individual ou rota específica através das máquinas às quais devem ser respeitadas;
- *Flow Shop*: cada *job* tem um padrão de fluxo idêntico;
- *Open Shop*: não há padrão de fluxo especificado para nenhum *job*;
- *Single Machine Shop*: apenas uma máquina está disponível;
- *Parallel Machines*: x máquinas idênticas em uma única etapa de processamento. Cada *job* utiliza uma e apenas uma dessas máquinas.

Além dessa classificação, existem alguns problemas de programação que são extensões desses clássicos, como é o caso do *Job Shop* Flexível, que é uma extensão do *Job Shop*, em que é permitido que uma operação seja processada por qualquer máquina um determinado conjunto de recursos (TAY e WIBOWO, 2004). Por ser objeto desse trabalho, será detalhado o *Job Shop* Flexível.

2.1.1 Job Shop Flexível

Segundo Chen et al. (1999), o *Job Shop* Flexível pode ser descrito como, sejam $T = \{T_1, T_2, T_3, \dots, T_t\}$ o conjunto de t jobs independentes entre si, e $M = \{M_1, M_2, M_3, \dots, M_m\}$ o conjunto das m máquinas. Para cada *job* $i \in T$, seja O_i o conjunto de operações j a serem processadas em sequência:

$$O_{i1} \rightarrow O_{i2} \rightarrow \dots \rightarrow O_{ij}, \quad (1)$$

sendo $j = \{1, 2, 3 \dots k\}$ e k o número máximo de operação pertencentes a um *job* i .

Cada operação O_{ij} pode ser processada por um conjunto de máquinas $M_{ij} \subseteq M$, respeitando a condição de que cada máquina pode executar uma e somente uma operação por vez.

Além disso, Chen, Thloew e Lehmann (1999) ressaltam que as restrições de precedência das operações de um *job* devem ser respeitadas e que uma vez iniciado o processamento de uma operação ele não pode ser interrompido.

Por se tratar de um problema complexo, o FJSP geralmente é dividido em dois subproblemas, o de roteamento e o de sequenciamento. O subproblema de roteamento atribui para cada operação uma máquina entre um conjunto de máquinas especificadas, enquanto o subproblema de sequenciamento seleciona a ordem em que serão executadas as operações (BRANDIMARTE, 1993).

O FJSP pode ser classificado quanto a sua flexibilidade, sendo ele totalmente ou parcialmente flexível, conforme a quantidade de máquinas habilitadas a processar uma operação. Para a flexibilidade total, cada operação pode ser processada por todas as máquinas e a flexibilidade parcial indica que cada operação pode ser processada apenas por um subconjunto do total de máquinas (KACEM et al., 2002).

Para exemplificar, considere uma instância do FJSP composta por quatro máquinas, três *jobs* e cada *job* constituída por três operações. O tempo de processamento e a máquina designada para processar cada operação estão descritos na Tabela 2.1. Nesse exemplo temos um problema parcialmente flexível, as operações O_{12} , O_{21} , O_{31} , O_{33} podem ser processadas apenas por duas máquinas distintas, já as operações O_{13} , O_{22} e O_{32} possuem três máquinas disponíveis para processá-las.

Tabela 2.1 - Exemplo de uma instância do FJSP.

<i>Job</i>	Operação	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>
1	O_{11}	2	1	3	2
	O_{12}	-	2	-	2
	O_{13}	2	-	3	3
2	O_{21}	-	1	3	-
	O_{22}	3	3	-	3
	O_{23}	2	1	3	-
3	O_{31}	-	3	-	7
	O_{32}	3	-	1	5
	O_{33}	-	2	-	3

Fonte – Próprio autor.

Uma vez definido o roteamento e o sequenciamento, há alguns critérios de desempenho que podem ser utilizados para verificar a eficiência da programação proposta. Os critérios de desempenho escolhidos para avaliar os resultados desse trabalho foram:

- *Makespan*; está relacionado com o tempo total utilizado para a conclusão de todas as operações;
- *Maximum Workload*; representa a quantidade de tempo que a máquina com mais quantidade de carga trabalhou;

- *Total Workload*; representa o somatório da quantidade de tempo que todas as máquinas trabalharam.

2.2 Problemas de Otimização Multiobjetivos

Problemas de otimização buscam minimizar ou maximizar uma função por meio de escolhas sistemáticas dos valores de variáveis dentro de um conjunto viável (FERREIRA, 2020). De forma conceitual, apresenta as seguintes características (LOBATO, 2008):

- *Função objetivo*: define qual característica do problema deseja ser maximizada ou minimizada;
- *Variáveis de decisão*: conjunto de parâmetros que podem influenciar os valores da função objetivo;
- *Restrições*: características que dependem matematicamente das variáveis de decisão e limitam os valores da função de objetivo a certas regiões do espaço de projeto. As restrições podem ser de desigualdade, que estabelecem uma região do espaço do projeto dentro da qual o seu valor deve ser maior ou igual/menor ou igual, a um valor pré-estabelecido, de igualdade, em que é definido uma região na qual as variáveis de decisão conferem a restrição um valor pré-determinado, e laterais, que delimitam uma faixa de variação para cada variável de decisão.

Quando o objetivo é a minimização ou a maximização simultânea de uma série de funções objetivo, temos um problema de otimização multiobjetivo (MOOP, do inglês *Multiobjective Optimization Problem*), que pode ser definido matematicamente como (DEB, 2001):

$$\begin{array}{ll}
 \text{Minimizar/maximizar} & H_b(x), \quad b = 1, 2, 3, \dots, B; \\
 \text{sujeito a} & g_c(x) \geq 0 \quad c = 1, 2, 3, \dots, C; \\
 & h_k(x) = 0, \quad k = 1, 2, 3, \dots, K; \\
 & x_d^L \leq x_d \leq x_d^U, \quad d = 1, 2, 3, \dots, q.
 \end{array}$$

sendo x o vetor de q variáveis de decisão $x = (x_1, x_2, x_3, \dots, x_q)$. Os valores de x_d^L e x_d^U representam os valores máximos e mínimos para a variável x_d , respectivamente e constituem as restrições laterais. Os termos g_c e h_k representam as C restrições de desigualdades e as K restrições de igualdades, respectivamente. Portanto, para uma solução x que satisfaz todas as restrições e limites de variáveis, temos uma solução viável. O conjunto de todas as soluções viáveis é chamado de região viável, ou espaço de busca.

Cada B função objetiva $H_b(x) = (H_1(x), H_2(x), H_3(x), \dots, H_b(x))$ pode ser otimizada, no entanto Konak et al. (2006) ressalta que em muitos problemas, os objetivos podem conflitar entre si. Por exemplo, se $H_1(x)$ for otimizada como mono objetivo, não necessariamente obteríamos os melhores resultados esperados para os demais objetivos. Portanto, uma perfeita solução multiobjetivo que simultaneamente otimize cada objetivo se torna algo inalcançável. Desta maneira, a solução para um problema multiobjetivo é dada por um conjunto de soluções, em que cada uma satisfaz os objetivos em um nível aceitável, sem ser dominada por qualquer outra solução. Esse conjunto é chamado de Ótimo de Pareto.

2.2.1 Ótimo de Pareto

A definição de ótimo de Pareto é baseada na convicção intuitiva que uma solução z é tomada como ótimo se “*nenhum critério utilizado pode melhorar a solução, sem piorar pelo menos um outro critério*”. (LOBATO, 2008).

Com isso, temos que se existem L funções objetivo f_l , com $l = \{1, 2, 3, \dots, L\}$, uma solução z_1 é dita dominante de outra solução z_2 , se as seguintes condições forem satisfeitas:

- A solução z_1 não é pior que z_2 para todos os objetivos, ou seja, para todo $l = \{1, 2, 3, \dots, L\}$, $f_l(z_1)$ não é pior que $f_l(z_2)$;
- A solução z_1 é estritamente melhor que z_2 em pelo menos um objetivo.

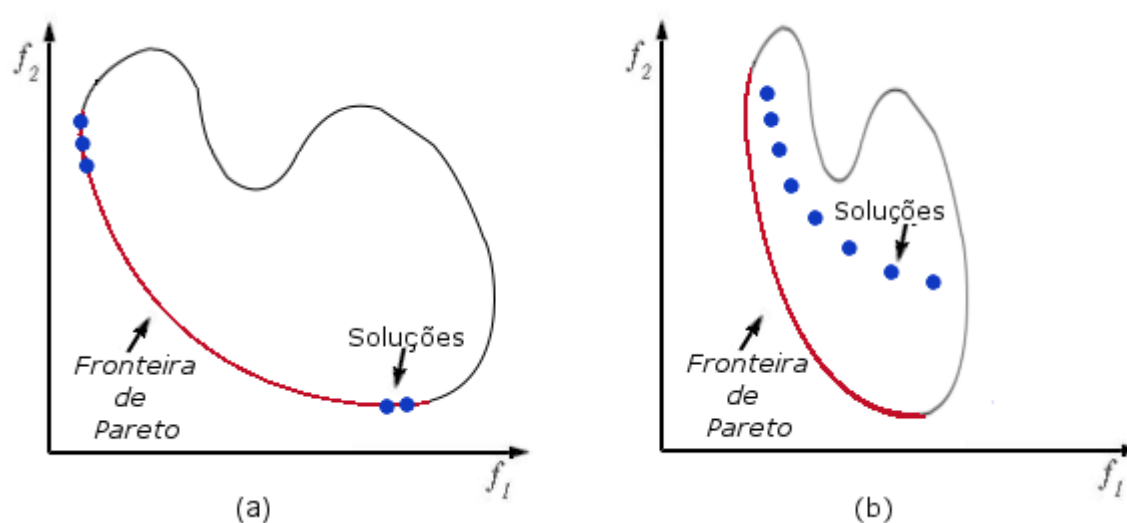
Se ambas as condições são satisfeitas, pode-se dizer que z_2 é dominado por z_1 e que z_1 é não-dominada por z_2 .

O ótimo de Pareto, tratando-se de problemas multiobjetivos, geralmente fornece um conjunto de soluções não dominadas e não apenas uma única solução como é esperado em problemas mono objetivos.

Algoritmos que tratam soluções multiobjetivo de Pareto (MOOP) tem como objetivo identificar soluções no conjunto de Pareto. No entanto, devida a complexidade do problema, acaba se tornando computacionalmente inviável a busca por tais conjuntos (KONAK et al., 2006). Por isso Deb (2001) propôs duas metas para possíveis soluções de MOOP:

- (1) Convergência: Encontrar um conjunto de soluções o mais próximo possível da Fronteira de Pareto;
- (2) Diversidade: Encontrar um conjunto de soluções com a maior diversidade possível.

Figura 2.1 Relação de Convergência (a) e diversidade (b) na solução do conjunto de Pareto



Fonte: Adaptado de Deb, 2001

A Figura 2.2 (a) ilustra um algoritmo que atende a primeira meta de convergência, pois apresenta uma boa convergência (as soluções estão perto da Fronteira de Pareto) em sua solução, no entanto ele não possui boa diversidade. Já a Figura 2.2 (b) o algoritmo possui em suas soluções uma boa diversidade (maior quantidade de soluções) e uma má convergência.

2.2.2 Métodos de otimização para problemas multiobjetivos

Para diferentes tipos de problemas de otimização, existem diferentes métodos de otimização, pois certas técnicas são mais adequadas para certos tipos de otimização que outras (YANG, 2010).

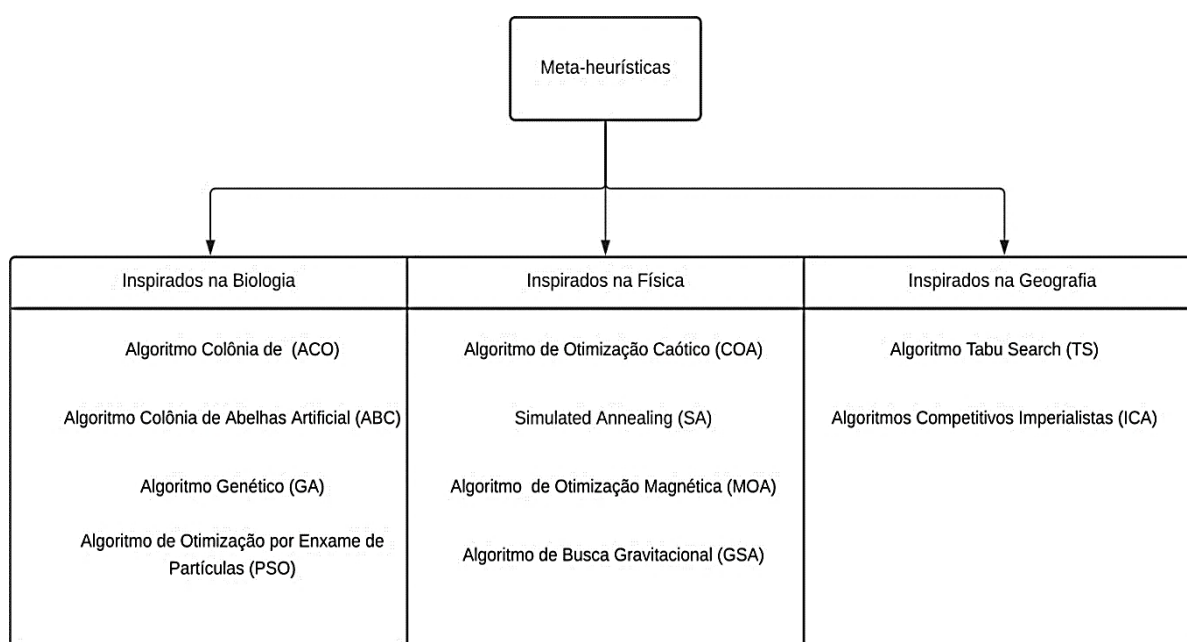
Cui et al. (2017) destaca dois métodos para resolver problemas de otimização: o método analítico que envolve estritamente provas matemáticas e derivações, e pode chegar a uma solução exata, no entanto esse método é limitado as características do problema; e o método numérico que é projetado com fórmulas de iteração apropriadas e aplicado com uma série de iterações para obter a solução aproximada.

Diante das limitações da resposta dos métodos analíticos, o método numérico oferece como alternativa para solução de problemas NP-difícil e problemas com objetivos conflitantes os chamados algoritmos heurísticos e meta-heurísticos.

Uma heurística é uma técnica computacional aproximativa que visa alcançar uma solução avaliada como aceitável para um dado problema, que em determinadas condições pode ser capaz de garantir a viabilidade ou a otimalidade da solução encontrada. A meta-heurística trata-se de uma arquitetura geral de regras que combinam métodos heurísticos com o objetivo de promover a eficiência e eficácia na exploração do espaço de busca (GOLDBARG et al., 2016).

Figura 2.2 - Exemplos de metas – heurísticas baseadas em analogias

Fonte: Adaptado de GOLDBARG et al. (2016).



Segundo Goldberg et al. (2016), as regras que compõem o contexto meta-heurístico podem ser desenvolvidas arbitrariamente e buscar base em analogia como fenômenos físicos, químicos, biológicos, sociais, dentre outros. Alguns exemplos de algoritmos inspirados nessas bases de analogia podem ser observados na Figura 2.3.

Neste trabalho utilizaremos a meta heurística bioinspirada Colônia de Abelhas Artificial.

2.3 Algoritmo Colônia de Abelhas Artificial

O modelo elaborado por Karaboga (2005), é baseado em uma colônia de abelhas que é dividida em três grupos de abelhas: empregadas, espectadoras e exploradoras. As abelhas empregadas têm a função de buscar o alimento em conjunto pré-especificado de fontes de alimento e transmitir informação acerca da localização e quantidade do néctar. Essa informação é passada para as outras abelhas em um local chamado área de dança.

Na área de dança, as abelhas espectadoras assistem à dança das abelhas empregadas e tomam uma decisão sobre qual fonte desejam visitar naquele instante, uma vez realizada essa escolha, as abelhas visitam essa fonte e retornam à colmeia com néctar, onde aguardam a volta de outras abelhas empregadas para repetirem o processo de escolha.

As abelhas exploradoras são abelhas empregadas que tiveram sua fonte de alimento esgotada e iniciaram uma nova busca aleatória nos arredores da colmeia para encontrarem novas fontes de alimento.

Para elaboração do algoritmo Karaboga (2005), fez uma analogia entre o espaço ao redor da colmeia, as fontes de alimento, a quantidade de néctar e o espaço de busca do problema, as possíveis soluções e a qualidade da solução (*fitness*), respectivamente. Para executar os operadores de busca cada fonte de alimento (solução) é atribuída a uma abelha diferente. O pseudocódigo algoritmo ficou estruturado da seguinte forma:

```

Gera o conjunto de fontes de alimento  $p_i$ ;
Relaciona cada fonte de alimento com uma abelha  $P$ ;
Calcula o fitness de cada solução;
iteração  $\leftarrow 1$ 
Repita
  /*Abelhas empregadas*/
  Para  $u \leftarrow 1$  até  $P$ 
    Realizar busca local em cada fontes de alimento  $p_u$  gerando uma nova solução  $a_u$ ;
    Calcular fitness de  $a_u$ ;
    Se fitness de  $a_i$  é melhor que o fitness de  $p_u$ 
       $p_u \leftarrow a_u$ 
      Zerar contador de não melhora de  $p_u$ ;
    Senão
      Incrementar contador de não melhora de  $p_u$ ;
    Fim se
  Fim para
  /*Abelhas expectadoras*/
  Para  $u \leftarrow 1$  até  $N$ 
    Realizar busca global atribuindo uma solução  $p_u$  para cada abelha  $b_u$ 
    Executar busca local em cada fontes de alimento  $b_u$  gerando uma nova solução
     $nova\_b_i$ ;
    Calcular fitness de  $nova\_b_i$ ;
    Se fitness de  $nova\_b_u$  é melhor que o fitness de  $b_u$ 
       $p_u \leftarrow nova\_b_u$ 

```

```

Zerar contador de não melhora de  $p_u$ ;
Senão
Incrementar contador de não melhora de  $p_u$ ;
Fim se
Fim para
/*Abelhas exploradoras*
Para  $u \leftarrow 1$  até  $X$ 
Se contador de não melhora de  $p_u$  atingiu o valor limite
Gerar nova solução  $nova_{c_u}$ ;
 $p_i \leftarrow nova_{c_u}$ ;
Fim se
Fim para
Incrementar iteração;
Até que iteração > número máximo de gerações

```

2.4 Tomador de Decisão

A tomada de decisão é um processo de selecionar um possível curso de ação a partir de todas as alternativas disponíveis para um determinado problema (HWANG e MASUD, 1979). Em muitos casos esses problemas são conhecidos como problema de tomador de decisão multicritérios (MCDM, do inglês *Multiple criteria decision making*), pois possuem uma multiplicidade critérios que devem ser melhorados e levados em consideração pelo tomador de decisão para apontar uma possível solução. Um problema de MCDM pode ser representado em forma de matriz como (CHEN, 2000):

$$D = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ A_1 & \left[\begin{array}{cccc} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{array} \right. & & & \end{matrix} \quad (1)$$

e

$$W = [w_1 \quad w_2 \quad \dots \quad w_n], \quad (2)$$

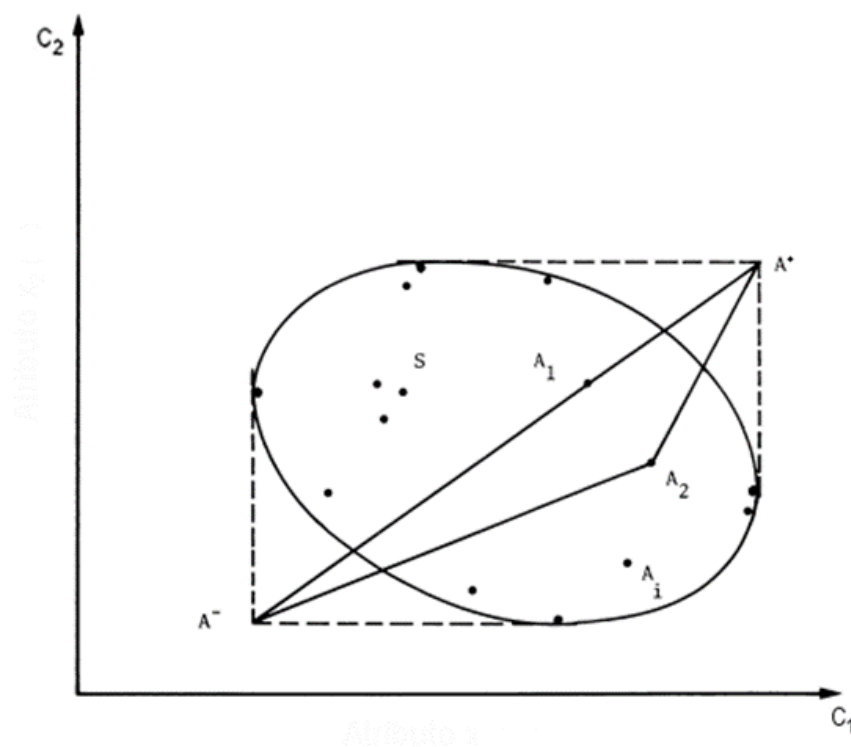
sendo A_1, A_2, \dots, A_m as possíveis soluções disponíveis que o DM deve escolher, C_1, C_2, \dots, C_n os critérios com os quais a alternativa é medida, x_{mn} o valor de cada critério C_n que compõem uma

determinada alternativa A_m . E cada elemento W_n do conjunto W representa o peso que cada critério C_n possui para uma determinada alternativa A_m .

Na literatura é possível encontrar várias técnicas de resolver problemas de MCDM. Dentre os métodos de decisão, podemos destacar a técnica para ordem de preferência por semelhança com a solução ideal (TOPSIS) proposta por Hwang e Yoon (1981), que dentre as abordagens matemáticas é uma das mais populares (SALIH et al. 2019).

O TOPSIS é baseado no conceito de que a melhor alternativa é aquela que possui a menor distância euclidiana da solução ideal positiva A^+ (PIS, do inglês *Positive ideal solution*) e a maior distância euclidiana da solução ideal negativa A^- (NIS, do inglês *Negative ideal solution*). Essa relação entre PIS e NIS pode ser ilustrada na Figura 2.4, em que S representa o espaço de busca e $A_1, A_2 \dots A_i$ são as soluções pertencentes a ele.

Figura 2.3 Distância euclidiana entre as alternativas A_m e o PIS (A^*) e NIS (A^-)



Fonte: adaptado de Hwang e Yoon, 1981.

No entanto, Chen (2000) ressalta que no processo de TOPSIS, as classificações de desempenho e os pesos dos critérios são dados através da relação entre conjuntos convencionais (*crisp*) que apresentam uma associação bem definida entre dois ou mais conjuntos. Sob muitas condições, os *crisp* são inadequados para modelar situações na vida real, uma vez que muitos problemas necessitam de julgamentos subjetivos, pois lidam com questões que podem ser imprecisas, indefinidas e incertas (SALIH et al., 2019). Para tratar essas questões subjetivas Chen (2000) propôs uma extensão do TOPSIS para o ambiente Fuzzy, chamada Fuzzy TOPSIS.

2.5 Fuzzy TOPSIS

Embora o TOPSIS e o Fuzzy-TOPSIS sejam desenvolvidos com base no mesmo princípio de proximidade com as soluções ideal positiva e negativa, estes se distinguem em relação à lógica que embasa seus procedimentos matemáticos (LIMA JUNIOR, et al., 2015)

Uma vez que o TOPSIS utiliza valores numéricos absolutos em formato *crisp* que são manipulados por meio da lógica clássica (ou lógica aristotélica), o método Fuzzy-TOPSIS faz uso da Lógica Fuzzy para realizar operações algébricas com valores numéricos intervalares (CHEN, 2000). Por conta disso, os métodos podem se distinguir quanto à complexidade da modelagem, ao processo de coleta de dados, ao esforço computacional requerido e até mesmo em relação às decisões finais fornecidas para um mesmo problema (LIMA et al., 2015).

Chen (2000) propôs a desenvolvimento do Fuzzy-TOPSIS seguindo as seguintes etapas:

Passo 1: Estabelecer o peso (importância) de cada critério C para o tomador de decisão, para isso utiliza-se as variáveis linguísticas descritas na Tabela 2.2:

Tabela 2.2 - Variáveis linguísticas para importância de cada critério (peso \tilde{w}_n)

Variável Linguística (\tilde{w}_n)	[w_{n1}, w_{n2}, w_{n3}]
Baixíssimo (BB)	[0, 0, 0.1]
Baixo (B)	[0, 0.1, 0.3]
Médio Baixo (MB)	[0.1, 0.3, 0.5]
Médio (M)	[0.3, 0.5, 0.7]
Médio Alto (MA)	[0.5, 0.7, 0.9]
Alto (A)	[0.7, 0.9, 1.0]
Altíssimo (AA)	[0.9, 1.0, 1.0]

Fonte: Adaptado do Chen, 2000.

Chen (2000) sugere a utilização da seguinte distribuição das variáveis linguísticas apresentada na Tabela 2.2, em que ele classificou com sete variáveis e atribuiu de forma uniforme pesos de 0 a 1 em forma de números que compõem o triângulo Fuzzy positivo.

Com a variável linguística de importância de cada critério é definido o grau de importância de cada critério para cada decisor (DM). A Tabela 2.3 ilustra o grau de importância atribuído a cada critério.

Tabela 2.3 – Grau de importância de cada critério

DM_1	DM_2	DM_3
--------	--------	--------

C_1	A	MB	MA
C_2	AA	AA	A
C_3	AA	BB	AA
C_4	M	MA	M
...
C_n	MA	A	BB

Para definir o peso de cada critério devem-se levar em consideração quais os objetivos que se deseja dar prioridade na solução final, na Tabela 2.3 são simulados três tomadores de decisão DM (DM_1, DM_2 e DM_3) com n critérios de decisão C cada.

Passo2: Estabelecer as variáveis linguísticas utilizadas pelos decisores DM para a avaliação das alternativas a serem escolhidas com respeito a cada critério.

A Tabela 2.4 ilustra estas variáveis linguísticas utilizadas para a avaliação de cada alternativa segundo cada critério:

Tabela 2.4 - Variáveis linguísticas para a importância de cada alternativa

Variável Linguística (\tilde{x}_n)	$[x_{n1}, x_{n2}, x_{n3}]$
Muito Pobre (MP)	[0, 0, 1]
Pobre (P)	[0, 1, 3]
Médio Pobre (MP)	[1, 3, 5]
Razoável (R)	[3, 5, 7]
Médio Bom (MR)	[5, 7, 9]
Bom (B)	[7, 9, 10]
Muito Bom (MB)	[9, 10, 10]

Fonte: Adaptado do Chen, 2000.

Chen (2000) sugere a utilização desta distribuição das variáveis linguísticas apresentada na Tabela 2.4, em que ele classificou com sete variáveis e atribuiu de forma uniforme pesos de 0 a 10 em forma de números que compõem o triângulo Fuzzy positivo.

Com a variável linguística de importância de cada alternativa é realizada a avaliação de cada alternativa pelo decisor DM em relação a cada critério. A Tabela 2.5 ilustra a avaliação.

Tabela 2.5 – Grau de importância de cada critério segundo cada decisor DM

		DM_1	DM_2	DM_3
C_1	A1	B	MP	MB
	A2	MB	MB	B
	A3	MB	MP	MB

C_2	A1	MB	MB	B
	A2	MB	MP	MB
	A3	R	MR	R

C_3	A1	MB	MB	B
	A2	MB	MP	MB
	A3	R	MR	R

C_4	A1	B	MP	MB
	A2	MB	MB	B
	A3	MB	MP	MB

C_n	A1	MB	MB	B
	A2	MB	MP	MB
	A3	R	MR	R

Tabela 2.5 são simulados três tomadores de decisão DM (DM_1, DM_2 e DM_3) com n critérios de decisão C cada.

Passo 3: Montar a matriz de decisão Fuzzy \tilde{D} estabelecendo o valor de cada critério avaliado C_n pelos decisores de acordo com a alternativa A_m .

$$\tilde{D} = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} & \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{m1} & \tilde{x}_{m2} & \dots & \tilde{x}_{mn} \end{bmatrix} \end{matrix} \quad (3)$$

A equação 3 ilustra a matriz de decisão, sendo A_1, A_2, \dots, A_m as possíveis soluções disponíveis que o DM deve escolher, C_1, C_2, \dots, C_n os critérios com os quais a alternativa é medida, \tilde{x}_{mn} o valor de cada critério C_n que compõem uma determinada alternativa A_m .

Cada elemento \tilde{x}_{yz} pode ser determinado de acordo com a quantidade de decisores, utilizando-se a equação 4 ilustrada abaixo:

$$\tilde{x}_{yz} = \frac{1}{k} [\tilde{x}_{yz}(DM_1) + \tilde{x}_{yz}(DM_2) + \tilde{x}_{yz}(DM_3) + \dots \sim \tilde{x}_{yz}(DM_k)] \quad (4)$$

Onde $\tilde{x}_{yz}(DM_1)$ equivale ao número triangular fuzzy da atribuição do decisor DM_1 para uma alternativa A_m . Ou seja, \tilde{x}_{11} é a média dos valores das atribuições dos decisores DM para cada alternativa (Tabela 2.5), por exemplo, para a alternativa A1 com o critério C1, os decisores

atribuíram B, MP e MB, ou seja (7 9 10) (0 0 1) (9 10 10) respectivamente, então \tilde{x}_{11} equivale a (5,3 6,3 6,7). Feito isso para cada elemento \tilde{x}_{yz} , a matriz de decisão é constituída.

Passo 4: Normalizar a matriz de decisão: A normalização é realizada utilizando as equações (5) e (6), sendo r_{mn} os valores de x_{mn} normalizados (KLIR e YUAN, 1995).

$$\tilde{r}_{mn} = \frac{\tilde{x}_{mn} - \min[\tilde{x}_{mn}]}{[\max[\tilde{x}_{mn}] - \min[\tilde{x}_{mn}]]} \quad (5)$$

ou

$$\tilde{r}_{mn} = \frac{\max[\tilde{x}_{mn}] - \tilde{x}_{mn}}{[\max[\tilde{x}_{mn}] - \min[\tilde{x}_{mn}]]} \quad (6)$$

As equações (5) e (6) são utilizadas para problemas de maximização e minimização respectivamente. Após a normalização tem-se a matriz Fuzzy normalizada \tilde{R} (Equação 7):

$$\tilde{R} = \begin{bmatrix} \tilde{r}_{11} & \tilde{r}_{12} & \dots & \tilde{r}_{1n} \\ \tilde{r}_{21} & \tilde{r}_{22} & & \tilde{r}_{2n} \\ & \vdots & \ddots & \vdots \\ \tilde{r}_{m1} & \tilde{r}_{m2} & \dots & \tilde{r}_{mn} \end{bmatrix} \quad (7)$$

Passo 5: Construir a matriz de decisão Fuzzy ponderada, multiplicado \tilde{W} à matriz \tilde{R} .

O vetor de peso \tilde{W} pode ser montado na forma Fuzzy, de acordo com a equação 8, sendo $\tilde{w}_J = (\tilde{w}_{J1}, \tilde{w}_{J3}, \tilde{w}_{J3})$ com $J= 1, 2, 3\dots n$. onde \tilde{w}_J é a variável linguística triangular Fuzzy de importância do critérios (Tabela 2.3).

$$\tilde{W} = \{\tilde{w}_1, \tilde{w}_2, \tilde{w}_3 \dots \tilde{w}_n\} \quad (8)$$

Cada elemento Fuzzy \tilde{w}_J pode ser determinado utilizando-se a equação 9 ilustrada abaixo:

$$\tilde{w}_J = \frac{1}{k} [\tilde{w}_J(DM_1) + \tilde{w}_J(DM_2) + \tilde{w}_J(DM_3) + \dots \tilde{w}_J(DM_k)] \quad (9)$$

Onde k é o número de decisores D, e cada elemento \tilde{w}_J é calculado sobre a média dos pesos estabelecidos para cada decisor a cada critério (Tabela 2.3). Por exemplo, \tilde{w}_1 é a média dos valores das atribuição para o critério C1, isto é, os decisores atribuíram (A MB MA), que equivale aos números triangulares Fuzzy (0,7, 0,9, 1,0) (0,1, 0,3, 0,5) (0,5, 0,7, 0,9), respectivamente, ou seja, \tilde{w}_J equivale a (0,43 0,63 0,8).

A construção da matriz de decisão Fuzzy ponderada é dada pela Equação 10 por:

$$\tilde{v}_{nm} = [\tilde{w}_{n1} \times r_{nm}, \tilde{w}_{n2} \times r_{nm}, \tilde{w}_{n3} \times r_{nm}]. \quad (10)$$

Resultado em uma matriz \tilde{V} (Equação 11):

$$\tilde{V} = \begin{bmatrix} \tilde{v}_{11} & \tilde{v}_{12} & \cdots & \tilde{v}_{1n} \\ \tilde{v}_{21} & \tilde{v}_{22} & \cdots & \tilde{v}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{v}_{m1} & \tilde{v}_{m2} & \cdots & \tilde{v}_{mn} \end{bmatrix} \quad (11)$$

Passo 6: Cálculo da Solução Ideal Positiva Fuzzy A^+ (FPIS, do inglês *Fuzzy Positive Ideal Solution*) e da Solução Ideal Positiva Fuzzy A^- (FNIS, do inglês *Fuzzy Negative Ideal Solution*), que representa os melhores e piores valores dos critérios, podem ser definidas respectivamente como

$$A_n^+ = \begin{cases} \text{Max} [\tilde{v}_o], & \text{para maximização do critério} \\ \text{Min} [\tilde{v}_o], & \text{para minimização do critério} \end{cases} \quad (12)$$

e

$$A_n^- = \begin{cases} \text{Min} [\tilde{v}_o], & \text{para maximização do critério} \\ \text{Max} [\tilde{v}_o], & \text{para minimização do critério} \end{cases} \quad (13)$$

onde $o = \{1, 2, 3, \dots, n\}$ e $\text{Max} [\tilde{v}_o] = (1,1,1)$ e $\text{Min} [\tilde{v}_o] = (0,0,0)$. Então, por exemplo, para a maximização do critério:

$$A_n^+ = [(1 \ 1 \ 1), (1 \ 1 \ 1), (1 \ 1 \ 1), (1 \ 1 \ 1) \dots (1 \ 1 \ 1)]$$

e

$$A_n^- = [(0 \ 0 \ 0), (0 \ 0 \ 0), (0 \ 0 \ 0), (0 \ 0 \ 0) \dots (0 \ 0 \ 0)]$$

Passo 7: Cálculo da distância S^+ entre FPIS e a distância S^- e FNIS:

Para isso é necessário determinar a distância euclidiana entre os valores Fuzzy de cada atributo e os valores de FPIS e FNIS através da equação 14:

$$d(\tilde{A}, \tilde{V}) = \sqrt{\frac{(a_1 - v_1)^2 + (a_2 - v_2)^2 + (a_3 - v_3)^2}{3}}, \quad (14)$$

sendo (a_1, a_2, a_3) as coordenadas do conjunto fuzzy triangular \tilde{A} e (v_1, v_2, v_3) as coordenadas do conjunto \tilde{v}_{mn} .

Uma vez definida as distancias euclidianas, as distâncias S^+ e S^- podem ser calculadas por:

$$S_r^+ = \sum_{s=1}^n d(A_s^+, v_{rs}) \quad (15)$$

e

$$S_r^- = \sum_{s=1}^n d(A_s^-, v_{rs}) , \quad (16)$$

sendo $r = \{1, 2, 3, \dots, n\}$, $s = \{1, 2, 3, \dots, m\}$ e as equações (15) e (16) as distâncias de cada solução do FPIS e do FNIS.

Passo 8: Calcular do coeficiente de proximidade CC_s que é dado pela equação 17 tem-se:

$$CC_s = \frac{S_s^-}{S_s^+ + S_s^-} , \quad (17)$$

sendo $s = \{1, 2, 3, \dots, m\}$.

A Equação 17 irá resultar em um valor entre 0 e 1, para cada solução. Assim, o *ranking* de preferência das alternativas apresentará como melhor solução a alternativa que tiver o maior valor do coeficiente de proximidade, seguida pelas demais em ordem decrescente.

CAPÍTULO 3

REVISÃO BIBLIOGRÁFICA

Esse capítulo apresenta a análise de alguns estudos encontrados na literatura que utilizam os mesmo métodos ou semelhantes dos propostos para resolver os problemas abordados nesse trabalho.

Na revisão os problemas de otimização multiobjetivo e do tomador de decisão multicritério são tratados de formas individuais, assim as pesquisas realizadas para essa revisão foram feitas em duas etapas focando nas necessidades de cada problema.

Para o problema de otimização foram analisados trabalhos que com o objetivo de minimizar o tempo total de execução de todas as operações (C_m - *Makespan*), a carga máxima de todas as máquinas (W_t - *Total Workload*) e a carga da máquina mais carregada (W_m - *Maximum Workload*). Como a abordagem meta-heurística escolhida foi a Colônia de Abelhas Artificial (ABC), foi buscado artigos que utilizavam esse método e que apresentassem diferentes estratégias de busca. Além disso também foram analisados trabalhos que focassem na busca pela diversidade dos resultados gerados.

Para o problema do tomador de decisão multicritério, foram analisados trabalhos cujo objetivo é a utilização de DM. tais como o Fuzzy-TOPSIS, capaz de lidar com problemas multicritério. Nessa análise foram encontrados diferente métodos de distribuição dos pesos entre as variáveis linguísticas, e a como a distribuição do peso nos critérios de decisão influenciam na resposta final.

3.1 Métodos de resolução multiobjetivo do Job Shop Flexível

Os trabalhos que serão apresentados a seguir foram realizados com o objetivo de otimizar o FJSP multiobjetivo utilizando abordagens baseadas na meta – heurística Colônia Artificial de Abelhas e abordagens que visam o aumento da diversidade dos resultados.

3.1.1 Soluções com o Algoritmo Colônia Artificial de Abelhas

Li, Pan e Gao (2011) foram pioneiros em aplicar a meta-heurística Colônia Artificial de Abelhas para o problema no *Job Shop* Flexível multiobjetivo. Em seu trabalho os autores utilizaram um algoritmo híbrido combinando o ABC com um arquivo do conjunto de Pareto para minimizar o *Makespan* (C_m), a carga máxima de todas as máquinas (W_t) e a carga da máquina mais carregada (W_m). No qual Pareto atua reunindo as soluções de Pareto não dominadas em um conjunto de soluções. Este conjunto é utilizado como referência para as abelhas exploradoras tomarem como ponto de partida e realizarem a busca local nas proximidades de cada solução não dominada.

Além disso Li, Pan e Gao (2011) consideram a solução como sendo dividida em dois vetores, um de operação, que indica a sequência em que as operações iram ser executadas, e um vetor de roteamento, que indica em quais máquina serão executadas cada operação, com isso propuseram um componente de busca local para o vetor de roteamento utilizando os seguintes métodos:

- Operador aleatório: Este operador de pesquisa local é gerado pelas seguintes etapas:
 - (1) Seleciona aleatoriamente uma operação com mais de duas máquinas disponíveis;
 - (2) Seleciona aleatoriamente, dentro das máquinas disponíveis para aquela operação, uma máquina diferente da máquina atual;
 - (3) Substitui a máquina atual pela nova selecionada.

- Top-k:
 - (1) Seleciona as k ($1 \leq k \leq m$) principais máquinas com carga de trabalho relativamente pesada;
 - (2) Seleciona aleatoriamente q ($1 \leq q \leq k$) operações que utilizam as k máquinas selecionadas;
 - (3) Para cada operação q , atribui uma máquina com relativamente menos carga de trabalho que a anterior.

Outra proposta encontrada na literatura para o FSJP é a elaborada por Li, Pan e Tasgetiren (2014) que junto com o ABC utiliza o método *Tabu Search* (TS) e Pareto. No qual o TS é utilizado com ferramenta de busca local tanto na inicialização das fontes de alimento, quanto na fase das abelhas. Já o Pareto é utilizado de forma semelhante ao proposto por Li, Pan e Gao

(2011) reunindo as soluções não dominadas em um conjunto que é utilizado como referência para as abelhas realizarem suas buscas.

Li, Pan e Tasgetiren (2014) também propõem em seu trabalho um método de construção do vetor de máquina nomeado de regra do tempo mínimo de operação, em no conjunto das máquinas disponíveis para determinada operação provavelmente a máquina com o menor tempo de execução será escolhida.

Ainda utilizando um algoritmo híbrido entre ABC e TS para como método de busca, temos o trabalho feito por Meng, Chen, Yang e Zhang (2020) que junto TS conta com mecanismos de busca múltiplos, que utiliza *crossover* de operação de precedência aprimorada (IPOX, do inglês *Improved precedence operation crossover*) e *crossover* de preservação multiponto (MPX, do inglês *Multipoint preservative crossover*) na fase de abelha empregada e realiza a operação de troca e operação de mutação na fase de abelha expectadora. Além disso, para avaliar melhor a solução, é proposta uma função de *fitness* baseada na classificação de Pareto e na distância de agrupamento.

Por fim, temos também o método proposto por Cao e Shi (2019) que também utiliza o ABC, no entanto ele baseada em um esquema de multi população, em que toda a população é dividida em várias subpopulações em cada geração e o tamanho de cada subpopulação é ajustado de forma adaptativa com base nas informações de seus resultados. Além disso, são implementados os operadores de busca IPOX e MPX para facilitar a troca de informações em cada subpopulação e entre diferentes subpopulações.

Para validar seus trabalhos, Li, Pan e Gao (2011), Li, Pan e Tasgetiren (2014), Meng, Chen, Yang e Zhang (2020) e Cao e Shi (2019) testaram seus métodos em diversas instâncias conhecidas, como a de Kacem, Hammadi e Borne (2002) como mostram a Tabela 3.1

Como podemos notar todos os métodos aplicados obtiveram resultados bem próximos ou iguais, que são os melhores resultados conhecidos, para as instâncias testadas.

Tabela 3.1 Resultados para duas instâncias de Kacem, Hammadi e Borne (2002)

Autores	Instâncias											
	4x5			8x8			10x7			10x10		
	C_m	W_t	W_m	C_m	W_t	W_m	C_m	W_t	W_m	C_m	W_t	W_m
Li et al., 2014	11	32	10	14	77	12	11	61	10	8	41	7
	12	32	8	15	75	12	11	62	11	7	42	5
	13	33	7	16	73	13	11	63	12	8	42	5
Li et al., 2014	11	32	10	14	77	12	11	61	11	7	42	6
	12	32	8	15	75	12	11	62	10	8	42	5
	11	34	9	16	73	13	12	60	12	7	43	5
	13	33	7	16	77	11	-	-	-	8	41	7
Meng et al., 2020	11	32	10	14	77	12	11	61	10	7	41	6
	12	32	8	15	75	12	11	62	11	7	41	7
	11	34	9	15	75	15	11	63	12	7	6	42
Cao e Shi, 2019	11	32	10	14	77	12	12	61	11	8	41	7
	12	32	8	15	75	12	11	63	11	8	42	5
	13	33	7	16	73	13	12	60	12	8	43	5
	-	-	-	17	77	11	-	-	-	7	42	6

Fonte: Adaptado de Li et al., 2011, Li et al. 2014, Meng et al., 2020 e Cao e Shi, 2019.

3.1.2 Implementações focados na diversidade dos resultados

Chao, Lin e Lu (2017) para resolver o problema de otimização multiobjetivo do FJSP, apresentam uma combinação do algoritmo ABC com um Busca de Índice de Diversidade (DIS, do inglês *Diversity index search*), além disso, é utilizada uma técnica de atribuição sequencial de operações por máquina (SOMA, do inglês *Sequential operations by machine attribution*) além de garantir a viabilidade das soluções e identificar a diversidade entre as soluções não dominadas. A tabela 3.2 ilustra os resultados para duas instâncias de Kacem, Hammadi e Borne (2002), destacando a quantidade de soluções encontradas para a programação da produção, na forma de gráficos de Gantt, para cada solução encontrada, por exemplo, para a solução da instância 10x7 onde os valores absolutos são $C_m = 16$, $W_t = 60$ e $W_m = 12$, foram encontradas 22 representações diferentes de gráficos de Gantt a partir desses valores absolutos de C_m , W_t e W_m .

Tabela 3.2 Resultados para duas instâncias de Kacem, Hammadi e Borne (2002).

Instância		C_t	W_t	W_m	Quantidade de soluções
		10x7	16	60	12

	15	61	11	30
	15	62	10	16
	8	41	7	66
10x10	8	42	5	42
	7	43	5	58
	7	42	6	34

Fonte: Adaptado de Chao, Lin e Lu (2017)

Além de buscar resultados na fronteira de Pareto para os critérios de desempenho W_t , W_m e C_m , Chao, Lin e Lu (2017) focaram na busca pela diversidade de gráficos de Gantt que representassem os mesmos valores para os três objetivos citado anteriormente. Essa quantidade de soluções foi utilizada para validando seu método nas instâncias de Kacem, Hammadi e Borne (2002), sendo utilizados como medida de desempenho do algoritmo proposto, pois os autores obtiveram os melhores resultados conhecidos para as instâncias para os critérios de desempenho utilizados, além de uma diversidade considerável para essas soluções, como mostra a Tabela 3.2.

Florêncio (2019) implementou em sua dissertação de mestrado um algoritmo de programação MOFJSP utilizando uma abordagem de otimização de Colônia de Abelhas, o FJSP – ABC, que proporcionou resultados animadores em relação a diversidade de soluções e variedade de gráficos de Gantt para a mesma solução. A Tabela 3.3 e 3.4 ilustram os resultados para as instâncias de Kacem, Hammadi e Bourne (2002) e para as instâncias de Bradimarte (1993) estudadas.

Wang, Zhou, Xu, and Liu (2013), propõem também a utilização de uma abordagem híbrida, porém com o algoritmo ABC para resolver o FJSP. Os autores apresentaram um interessante método de busca local intensificada baseado no método de pesquisa de vizinhança variável (VNS, do inglês *Variable neighborhood search*) que proporcionou além de uma melhor nos resultados, ajudou a evitar a convergência prematura o que garantiu uma maior diversidade dos resultados.

Tabela 3.3 Resultados para duas instâncias de Kacem, Hammadi e Borne (2002).

	Instâncias											
	8x8				10x10				15x10			
	C_m	W_t	W_m	sol	C_m	W_t	W_m	sol	C_m	W_t	W_m	sol
FSJP – ABC	14	77	12	4	7	42	6	85	12	91	12	1

15	75	12	24	8	41	7	9600	13	91	12	897
16	73	13	53	8	42	5	187	14	91	12	1968
16	77	11	2	9	41	7	9926	12	92	12	18
16	78	11	14	7	43	6	3	12	93	11	28
16	77	12	1					13	92	11	149
								13	91	13	10996

Fonte: Adaptado de Florêncio (2019)

Tabela 3.4 Resultados para duas instâncias de Bradimarte (1993).

		Instâncias											
		MK2				MK4				MK8			
		C_t	W_t	W_m	Sol	C_t	W_t	W_m	sol	C_t	W_t	W_m	sol
FSJP – ABC		27	151	27	35	75	356	75	1	564	2513	542	1
		28	146	27	1	76	349	75	1	565	2514	533	1
		30	146	27	1796	77	349	75	32	568	2515	542	1
		29	144	29	3	150	324	146	1	594	2484	587	1
		37	140	33	1	138	325	138	1	596	2486	587	1
		34	141	34	1	130	326	130	2	597	2488	587	1
		32	142	31	2	93	357	61	1	594	2524	523	1
		30	151	26	1	84	353	62	7	573	2519	524	2
		30	145	27	1	86	348	63	5	631	2535	527	1
		31	144	28	2								

Fonte: Adaptado de Florêncio (2019)

3.2 Tomadores de decisão para problemas multiobjetivos

Na revisão da literatura realizada foram encontrados vários tomadores de decisão aplicados a diferentes problemas. No entanto, tomadores de decisão baseado aplicados para escolha de soluções do problema no Job Shop Flexível (FJSP) especificamente, não foram encontrados. Os diversos autores propuseram formas de seleção de resultados que podem ser adaptados para o

FJSP, uma vez que levanto em conta as características do problema, foram encontrados diversos trabalhos que abordam a solução de problemas multicritérios.

3.2.1 Tomadores de Decisão

Nayak et al. (2019), fizeram um estudo comparando as técnicas de decisão Processo de Hierarquia analítica (do inglês *Analytical Hierarchy Process* - AHP), que é uma técnica em que a relevância de cada critério utilizado com para tomada de decisão é determinada com base na experiência empírica de especialistas sobre o caso analisado, Solução de Compromisso e Otimização Multicritério (do bósnio *VIseKriterijumska Optimizacija I Kompromisno Resenje* - VIKOR), que estabelece um ranking segundo as distâncias com relação a um cenário ideal, e o TOPSIS, que classifica os resultados baseado nas distâncias com relação as soluções ideias do conjunto avaliado, para problemas de Scheduling afim de evitar conflitos em tarefas semelhantes para otimizar a alocação de slots de uma máquina virtual e sua utilização para tarefas baseadas em prazos em um ambiente de nuvem. A Tabela 3.5 ressalta os aspectos dessa comparação:

Tabela 3.5 - Resultados para duas instâncias de Kacem, Hammadi e Borne (2002).

Método	Taxa média de aceitação da tarefa
AHP	80.252
VIKOR	85.143
TOPSIS	88.016

Fonte – Adaptado de Nayak(2019).

Conforme a Tabela 3.5 notamos que a técnica TOPSIS obteve maior taxa de aceitação que os demais métodos utilizados, o que significa que seu desempenho na alocação das tarefas foi mais eficiente AHP e VIKOR.

Syedmohammadie et al. (2018), fizeram um estudo comparativo utilizando a Ponderação Aditiva Simple (do *ingles Simple Additive Weighting* – SAW), em que a classificação dos resultados é obtido através da relação entre um peso designado para cada critério e o valor de do critério na alternativa, TOPSIS e Fuzzy TOPSIS, para escolha prioridade em plantio de grãos. A diferença menos significava dos resultados estão apresentadas na tabela 3.6.

Tabela 3.6 - Resultados para duas instâncias de Kacem, Hammadi e Borne (2002).

Método	Milho	Canola	Soja
Fuzzy-TOPSIS	0.514	0.355	0.132
TOPSIS	0.474	0.351	0.175
SAW	0.363	0.338	0.299

Fonte – Adaptado de Seyedmohammadie (2018).

Com base nos resultados apresentados na tabela 3.3 notamos que para dois dos cenários avaliados o Fuzzy Topsis apresentou melhores resultados que os demais métodos utilizados.

3.2.2 Fuzzy-TOPSIS para problemas multicritérios

Kumar, Kumar e Barman (2018) exploram em seu trabalho a possibilidade de distinguir entre os critérios de custo (minimização do critério) e benefício (maximização do critério) em uma mesma solução analisada, oferecida pelo tomador de decisão Fuzzy-TOPSIS, aplicando a técnica para seleção de fornecedores. Os autores realizaram análise de sensibilidade para investigar o efeito dos pesos dos critérios na seleção de fornecedores, para isso foram testadas diversas combinações de peso para cada critério.

Analisando os resultados desses testes obtidos por Kumar, Kumar e Barman (2018) apresentados na Tabela 3.7 pode-se observar como a mudança de pesos dos critérios influencia na decisão final do DM.

Tabela 3.7 - Mudanças de peso dos critérios para análise de sensibilidade

Caso	Peso do critério modificados	Coeficiente de proximidade CC_j				Ranking
		A_1	A_2	A_3	A_4	
Caso 1	$w_{c1-c5} = [0.1 \ 0.1 \ 0.3]$	0.3520	0.3320	0.4231	0.4268	$A_2 < A_1 < A_3 < A_4$
Caso 2	$w_{c1-c5} = [0.1 \ 0.3 \ 0.5]$	0.3818	0.3640	0.4623	0.4686	$A_2 < A_1 < A_3 < A_4$
Caso 3	$w_{c1-c5} = [0.3 \ 0.5 \ 0.7]$	0.4000	0.3650	0.4875	0.5015	$A_2 < A_1 < A_3 < A_4$
Caso 4	$w_{c1-c5} = [0.5 \ 0.7 \ 0.9]$	0.4086	0.3658	0.5043	0.5238	$A_2 < A_1 < A_3 < A_4$
Caso 5	$w_{c1-c5} = [0.7 \ 0.9 \ 0.9]$	0.4453	0.3842	0.5626	0.5923	$A_2 < A_1 < A_3 < A_4$
Caso 6	$w_{c1} = [0.7 \ 0.9 \ 0.9], w_{c2-c5} = [0.1 \ 0.1 \ 0.3]$	0.3128	0.3131	0.4356	0.5448	$A_1 < A_2 < A_3 < A_4$
Caso 7	$w_{c2} = [0.7 \ 0.9 \ 0.9], w_{c1,c2-c5} = [0.1 \ 0.1 \ 0.3]$	0.3128	0.3163	0.4650	0.5448	$A_1 < A_2 < A_3 < A_4$
Caso 8	$w_{c3} = [0.7 \ 0.9 \ 0.9], w_{c1,c2,c4,c5} = [0.1 \ 0.1 \ 0.3]$	0.4709	0.3857	0.5225	0.4188	$A_2 < A_4 < A_1 < A_3$

Caso 9	$w_{c4} = [0.7 \ 0.9 \ 0.9]$, $w_{c1-c3,c5} = [0.1 \ 0.1 \ 0.3]$	0.3792	0.3627	0.4742	0.5209	$A_2 < A_1 < A_3 < A_4$
Caso 10	$w_{c5} = [0.7 \ 0.9 \ 0.9]$, $w_{c1-c5} = [0.1 \ 0.1 \ 0.3]$	0.4709	0.3857	0.4749	0.4188	$A_2 < A_4 < A_1 < A_3$

Fonte: Adaptado de Kumar, Kumar e Barman, 2018.

Analisando os resultados desses testes obtidos por Kumar, Kumar e Barman (2018) apresentados na Tabela 3.7 pode-se observar como a mudança de pesos dos critérios influencia na decisão final do DM.

Perera, Attalage, Perera e Dassanayake (2013) propuseram uma ferramenta híbrida que combina métodos de otimização multiobjetivo e tomador de decisão multicritério para o conjunto de Pareto. Como técnica de otimização foi utilizada um algoritmo evolutivo de estado estacionário baseado no método de ϵ -dominância, com o objetivo de otimizar quatro critérios de desempenho e por fim formar um conjunto de Pareto. Para o tomador de decisão os autores aplicaram o Fuzzy-TOPSIS utilizando Diagrama de Nível para elaboração da matriz de pesos. O Diagrama de Nível é uma técnica de apoio de decisão baseada na classificação de aproximação da fronteira de Pareto de acordo com a proximidade do ponto ideal. Essa ferramenta foi utilizada no estudo de caso de projetos de sistemas de energia híbridos autônomos.

Sharma e Singhal (2017) utilizam o Fuzzy-TOPSIS como método de seleção de uma abordagem procedimental para o tratamento do problema de layout de instalação (FLP, do inglês *Facility layout problem*). Os autores testaram duas técnicas de definição dos pesos das variáveis linguísticas, a primeira foi baseada em uma lógica digital modificada (MDL, do inglês *Modified digital logic*) e a segunda no processo de hierarquia analítica (AHP). Na análise dos resultados feitas Sharma e Singhal (2017) foi concluído que ambos os métodos foram eficazes, pois obtiveram a mesma resposta no *rank* 1.

Utilizando uma combinação do Fuzzy-TOPSIS com o Fuzzy-AHP, Abd, Abhary e Marian (2017) propuseram um método de seleção de *schedulings* para problemas multiobjetivo em células de montagem robótica flexível. O método Fuzzy-AHP foi usado para determinar os pesos das variáveis linguísticas e na construção da matriz dos pesos dos critérios de decisão, enquanto o Fuzzy-TOPSIS foi utilizado para avaliar as soluções viáveis e indicar uma solução final. Os autores ressaltaram a carência de métodos de seleção de *schedulings*, uma vez que a maioria dos problemas de otimização terminam na geração da fronteira de Pareto.

CAPÍTULO 4

PROPOSTA DO TRABALHO

A proposta deste trabalho é implementar método de programação da produção utilizando um programador baseado na meta-heurística de Colônia de Abelhas Artificiais (ABC) e um tomador de decisão (DM) baseado no Fuzzy-TOPSIS capaz de selecionar e ranquear as melhores soluções de programação da produção para o sistema produtivo do tipo Job Shop Flexível (FJS) dentre os melhores resultados disponíveis no conjunto de Pareto.

O algoritmo de programação da produção baseado na metaheurística ABC utilizado foi desenvolvido por Florêncio (2019), o FJSP-ABC, o qual disponibiliza as soluções encontradas para os problemas das instâncias de Kacem, Hammadi e Bourne (2002) e e Bradimarte (1993). Esse algoritmo fornece uma arquivo com as soluções de Pareto para os objetivos C_m , W_t e W_m . No arquivo de Pareto fornecido também são fornecidos os dados para a construção dos gráficos de Gantt de cada solução. Uma das características dessa implementação é a de fornecer uma grande variedade de gráficos de Gantt para a mesma solução C_m , W_t e W_m , conforme descrito na secção 3.1.2.

O algoritmo do DM implementado analisa os diversos cenários propostos pelos gráficos de Gantt, para as soluções encontradas pelo algoritmo de programação da produção baseado na meta-heurística ABC (C_m , W_t e W_m) e para variáveis relacionadas à operação das máquinas no tempo do sistema produtivo para a escolha da melhor programação da produção. A saída do algoritmo é o ranqueamento das soluções que melhor atende as necessidades do sistema produtivo. Para tratar o desta implementação é proposta a utilização um tomador de decisão baseado na lógica Fuzzy TOPSIS proposto por Chen (2000) e Kumar, Kumar e Barman (2018). A validação do algoritmo do tomador de decisão (DM) será realizada utilizando os cenários estabelecidos pelas instância de benchmark citadas e as soluções do conjunto de Pareto encontradas pelo algoritmo FJSP - ABC multiobjetivo.

4.1 - O algoritmo de resolução da programação da produção FJSP Multiobjetivo e o algoritmo DM Fuzzy-TOPSIS.

Com base nas técnicas apresentadas nos capítulos 2 e 3, serão detalhadas a forma de implementação dos algoritmos ABC do FJSP multiobjetivo e do Fuzzy-TOPSIS.

4.1.1 Algoritmo Colônia Artificial de Abelhas

No trabalho de Flôrencio (2019), assim como nos trabalhos de Li, Pan e Gao (2011), por Li, Pan e Tasgetiren (2014), Meng, Chen, Yang e Zhang (2020), Cao e Shi (2019) e Chao, Lin E Lu (2017) são implementados dois vetores, um de máquinas e outro de operações. Para a implementação do vetor de máquinas e de operação, estes serão tratados como dois problemas distintos, por isso cada um será processado por diferentes funções tanto de criação quanto de busca.

Após o término de todas as interações, os melhores resultados obtidos em todas as gerações são salvos em um arquivo de Pareto, que contém os vetores das operações e das máquinas e seus respectivos valores do *makespan*, *total workload* e *maximum workload*. O algoritmo da implementação do problema FJSP utilizando a meta-heurística ABC (FJSP-ABC) pode ser ilustrado no Algoritmo 4.1 como:

Algoritmo 4.1 - Algoritmo Colônia Artificial de Abelhas (FJSP – ABC)

```
1. Carregar instancia selecionada
2. Inicialização
3. Repita
4.     Abelhas empregadas
5.     Abelhas expectadoras
6.     Salvar melhores soluções //Arquivo de Pareto
7.     Abelhas exploradoras
8. Até que iteração > MaxGen
9. Salvar arquivo de Pareto
```

4.1.2 Configurações iniciais

Inicialmente são configurados os parâmetros necessários, que são apresentados na Tabela 4.1, para o funcionamento do algoritmo, esses valores devem ser ajustados de acordo com o *benchmark* selecionado.

Tabela 4.1 Relação das variáveis do programa

Nome da variável	Descrição
<i>PROBLEMA</i>	Instancia do benchmark
<i>Num_sol</i>	Quantidade de soluções iniciais geradas
<i>Num_abelhas</i>	Quantidade de abelhas associadas a uma solução
<i>Mudancas_Aem</i>	Número de buscas realizada por cada abelha empregada
<i>Mudancas_Aes</i>	Número de buscas realizada por cada abelha espectadora
<i>MaxGen</i>	Quantidade de iterações
<i>Limite</i>	Quantidade limite de iterações que não apresentaram melhora
<i>Num_op_Troca_maq</i>	Número de máquinas que serão substituídas na função de busca no vetor de máquinas

A seleção do *benchmark* é feita através da variável *PROBLEMA*, em que é possível escolher uma entre as quatro opções de instâncias disponíveis, em que cada instância é relacionada com um número inteiro entre um e quatro.

Uma vez definida qual problema será tratado, é necessário escolher a quantidade de soluções que serão geradas, as abelhas empregadas e espectadoras, o número de buscas realizadas pelas abelhas, o limite interações que máximo que uma solução sofre antes de ser substituída por uma nova solução e a prioridade que cada objetivo tem. A Tabela 4.2 ilustra a instância 4x5 de Kacem, Hammadi e Borne (2002).

Definidos os parâmetros iniciais, a instância do *benchmark* é carregada em uma matriz relacionando o *job* e suas operações com as máquinas disponíveis e seus respectivos tempos de execução, como pode ser observado no exemplo na Tabela 4.2.

Tabela 4.2 Instancia 4x5 de Kacem, Hammadi e Borne (2002)

<i>Job</i>	Operação	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>
1	1	2	5	4	1	2
	2	5	4	5	7	5
	3	4	5	5	4	5
2	1	2	5	4	7	8
	2	5	6	9	8	5
	3	4	5	4	54	5
3	1	9	8	6	5	9
	2	6	1	2	7	4
	3	2	5	4	2	4
	4	4	5	2	1	5
4	1	1	5	2	4	12
	2	5	1	2	1	2

Fonte: Adaptado de Kacem, Hammadi e Borne (2002)

4.1.3 Geração das soluções iniciais

A inicialização do algoritmo necessita uma série de procedimentos, tais como gerar os vetores de operações, os vetores de máquinas e em primeiro momento encontrar um ranqueamento inicial das soluções de acordo com um critério de desempenho. A partir dessa seleção, é possível se ter uma relação com as soluções que iniciarão o algoritmo FJSP-ABC. O algoritmo da inicialização pode ser ilustrado abaixo no Algoritmo 4.2:

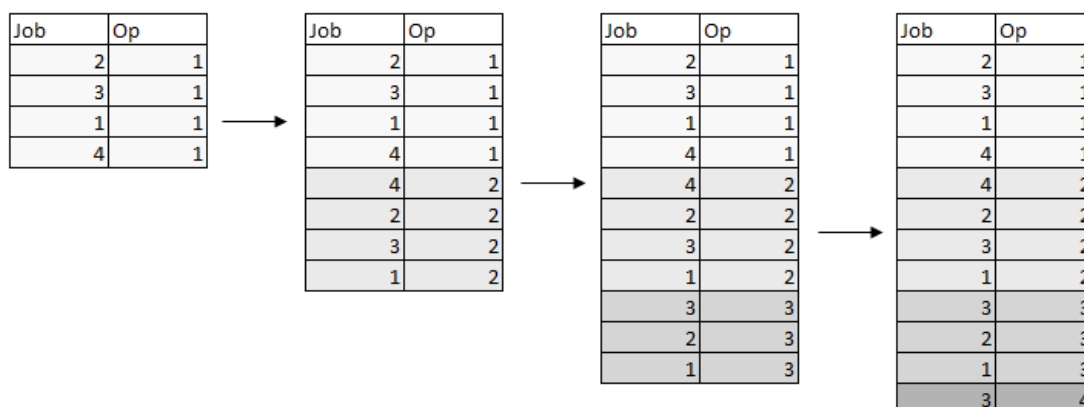
Algoritmo 4.2 - Algoritmo de inicialização do FJSP – ABC.

1.	Repita
2.	Gerar vetor das operações
3.	Gerar vetor das máquinas
4.	Calcular critérios de desempenho
5.	Até que interação > <i>Num_sol</i>
6.	Selecionar <i>Num_abelhas</i> melhores resultados

Fonte – Próprio autor.

O primeiro passo para gerar as soluções é a alocação do *jobs* e das operações. Como ilustrado na Figura 4.1 os *jobs* são distribuídos pelas suas operações, ou seja, o algoritmo aloca as operações 1 e de forma aleatória de distribui todos os *jobs* que possuem essa operação, depois as operações 2 e assim até que todas os *jobs* e operações do processo sejam alocados.

Figura 4.1 Função de alocação dos *jobs* e das operações



Fonte: Próprio autor.

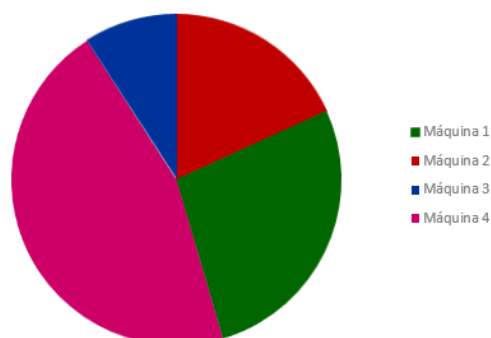
Para distribuição das máquinas foi utilizada uma técnica baseada na regra do tempo mínimo de operação proposta por Li, Pan e Tasgetiren (2014). Levado em consideração que cada operação possui um conjunto de máquina em que ela pode ser executada com seus respectivos tempos de execução, como pode ser visto no exemplo na Tabela 4.3

Tabela 4.3 Exemplo de conjunto de máquinas disponíveis para um *job/operação*

<i>Job</i>	Operação	Máquina	Tempo
1	1	1	2
		2	3
		3	5
		4	1

Os valores dos tempos são distribuídos em um da roleta em que as operações que possuem menor tempo de execução recebem maior fatia, aumentando assim a probabilidade de estes serem escolhidos. A Figura 4.2 ilustra a distribuição dos tempos na roleta do exemplo apresentado na Tabela 4.3, na qual podemos notar que as maiores fatias representam as máquinas que possuem os menores tempos.

Figura 4.2 Roleta da distribuição dos tempos



Fonte: Próprio autor.

Depois de geradas e calculados seus respectivos critérios de desempenho, utilizando como parâmetro o menor *Makespan*, as soluções passam por um processo de elitismo, no qual apenas as com os menores valores são atribuídas a abelhas.

4.1.4 Abelhas empregadas e expectadoras

Nas fases das abelhas empregadas e expectadoras a solução é submetida a operações de buscas, a fim de encontrar uma nova solução no espaço de busca que possua melhores atributos. O algoritmo para as abelhas empregadas e expectadoras pode ser ilustrado abaixo no Algoritmo 4.3:

Algoritmo 4.3 - Abelhas Empregadas e Expectadoras

```

1. Para i=1 até Mudancas_Aem // Mudancas_Aes
2.     Modificador do vetor de operações
3.     Modificador do vetor de máquina (Num_op_Troca_maq)
4. Fim_Para
5. Se houve melhora nos critério de desempenho
6.     Salva resposta nova
7.     Zera contador de não melhora
8. Se não
9.     Mantem resposta antiga
10.    incrementa contador de não melhora
11. Fim_Se

```

Fonte – Próprio autor.

No vetor de operações é utilizado um modificador de sequências baseado no *random swap* (troca aleatória), sendo O_{ij} uma operação j pertencente a um *job* i . As operações são alocadas como mostra a Figura 4.3.

Figura 4.3 Alocação das operações

O_{21}	O_{11}	O_{31}	O_{12}	O_{22}	O_{32}	O_{33}	O_{23}	O_{13}	O_{24}
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Fonte: Próprio autor.

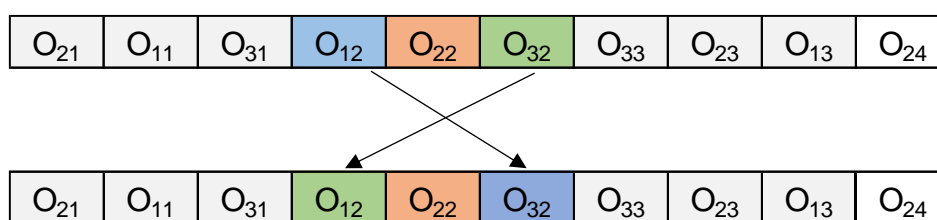
É escolhida, de forma aleatória, uma operação j , a qual pertencente a pelo menos dois *jobs* i diferentes, como mostra no exemplo da Figura 4.4.

Figura 4.4 - Exemplo de operações escolhidas

O_{21}	O_{11}	O_{31}	O_{12}	O_{22}	O_{32}	O_{33}	O_{23}	O_{13}	O_{24}
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Fonte: Próprio autor.

Figura 4.5 - Exemplo da seleção e troca de dois *jobs*



Fonte: Próprio autor.

Então no conjunto das operações i selecionada, dois *jobs* j são escolhidos aleatoriamente e tem suas posições invertidas, como ilustra a Figura 4.5.

Para o vetor de alocação das máquinas, é aplicado um modificador baseado no operador *top-k* proposto por Li, Pan e Gao (2011). Nessa função as k máquinas alocadas que possuem os maiores tempos de execução são substituídas, sendo que k é representado pela variável *Num_op_Troca_maq*. A escolha da nova máquina é feita através de um sorteio em que as máquinas mais ociosas têm a maior probabilidade de serem escolhidas.

Após passar pelas funções de busca a nova solução gerada é comparada com a anterior, se houve melhora de algum dos critérios de desempenho a antiga solução é substituída pela nova, caso não haja melhora a abelha (solução) é mantida e seu contador de não melhora é incrementado.

O que diferencia a fase das abelhas empregadas das expectadoras é que, na primeira todas as soluções passam pelas funções de busca, enquanto nas expectadoras as funções de busca são aplicadas apenas em algumas abelhas selecionadas de forma que as que possuem os melhores valores para os critérios de desempenho tem a maior probabilidade de serem escolhidas.

4.1.5 Arquivo de Pareto

Após a execução da fase das abelhas empregadas e expectadoras, os resultados que obtiveram os melhores valores para cada critério de desempenho são salvos em um arquivo externo, que compõem o conjunto das melhores soluções encontradas durante a execução do algoritmo. Esse arquivo é chamado de Arquivo de Pareto.

4.1.6 Abelha exploradora

Na fase das abelhas exploradoras as soluções (abelhas empregadas) que atingirem o valor máximo do contador de não melhora, pré-estabelecido pela variável *Limite*, são substituídas. O Algoritmo 4.4 ilustra essa operação.

Algoritmo 4.4 - Abelhas Exploradoras

```

1.   Para  $i = 1$  até Num_abelhas
2.       Se contador de não melhora  $i > LIMITE$ 
3.           Selecionar solução no arquivo de Pareto
4.           Modificador do vetor de operações //Random Swap
5.           Modificador do vetor de máquina //Top-k
6.       Fim_Se
7.   Fim_Para

```

Fonte – Próprio autor.

Essa troca pode ser feita de duas maneiras, a primeira é gerando uma solução nova e a segunda é baseada em no trabalho Li, Pan e Gao (2011) em que uma solução nova é escolhida aleatoriamente no arquivo de Pareto e essa nova solução passar por uma série de perturbações pelos operadores de busca. A maneira em que a troca é feita é escolhida de forma aleatória.

O formato do arquivo de Pareto pode ser ilustrado na Figura 4.5.

Figura 4.5 - Exemplo de formato do arquivo de Pareto

Geração	C_m	W_t	W_m	Vetor das Operações												Vetor das máquinas										Vetor com os tempos													
65	11	33	10	2	5	10	1	6	11	4	7	9	12	3	8	5	2	4	1	5	3	3	2	1	4	1	2	2	4	4	2	5	4	6	1	2	1	1	1
86	11	33	10	2	5	9	3	6	10	1	8	11	12	4	7	5	2	4	1	5	3	3	2	1	4	1	2	2	4	4	2	5	4	6	1	2	1	1	1
5	12	33	10	3	5	9	1	8	11	4	7	10	12	2	6	5	2	4	1	1	3	3	2	1	4	1	2	2	4	4	2	5	4	6	1	2	1	1	1
⋮	⋮	⋮	⋮	⋮												⋮										⋮													

Fonte: Próprio autor.

Na Figura 4.5, em azul são os valores dos objetivos as serem minimizados ordenados pelos mais perto da fronteira de Pareto. Na primeira coluna encontra-se a geração em que ele foi encontrado no algoritmo. Na cor rosa tem o valor do vetor de operações para aquela determinada geração, seguido pelo valor do vetor de máquinas, na cor amarela, e dos tempos de processamento das máquinas na cor laranja.

4.1.7 Implementação do Tomador de decisão Fuzzy-TOPSIS

A implementação do algoritmo do Fuzzy – TOPSIS foi feita com base nos métodos propostos por Chen (2000) e por Kumar, Kumar e Barman (2018) e Klir e Yuan (1995). O algoritmo da implementação pode ser ilustrado no Algoritmo 4.5.

Algoritmo 4.5 - Tomador de decisão FUZZY-TOPSIS

1. Carregar arquivo de Pareto
2. Criar matriz com os critérios de desempenho contidos no arquivo de Pareto
3. Carregar Variáveis Linguísticas
4. Carregar peso dos critérios de decisão
5. Normalizar matriz das soluções
6. Criar matriz Fuzzy
7. Definir FPIS e FNIS
8. Calcular distância euclidiana
9. Calcular Grau de proximidade
10. Ordenar soluções
11. Salvar soluções ordenadas

Fonte – Próprio autor.

4.1.8 Variáveis Linguísticas e Critérios de Decisão

O primeiro passo para a construção do algoritmo é a definição das variáveis linguísticas relativas aos pesos dos critérios utilizados, nessa implementação optou-se por utilizar os valores sugeridos por Chen (2000), apresentados na Tabela 4.4 com seus respectivos pesos. Onde w corresponde aos pesos dos critérios C_n , conforme explicado na Seção 2.5.

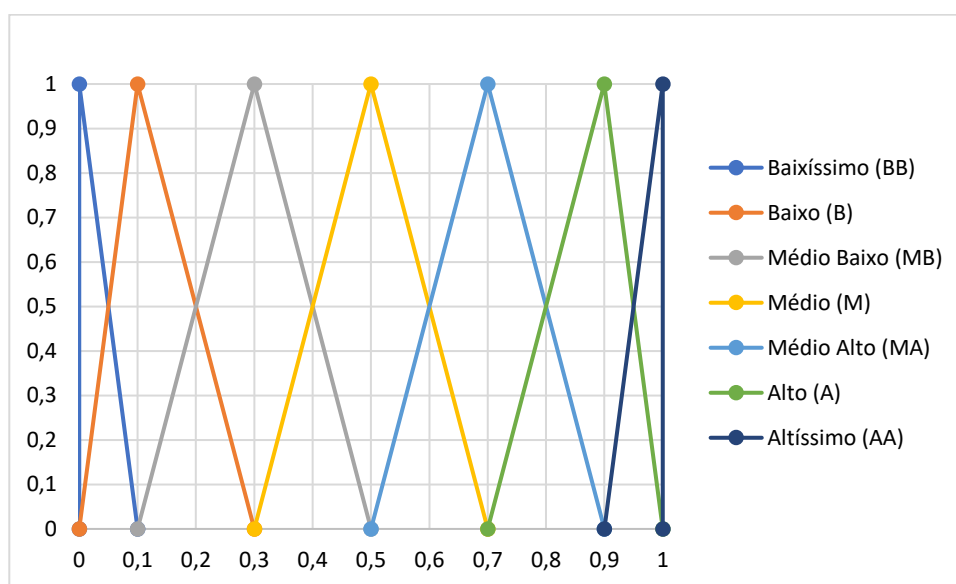
Tabela 4.4 - Variáveis Linguísticas

Variável Linguística (\tilde{w}_n)	Peso [w_{n1} w_{n2} w_{n3}]
Baixíssimo (BB)	[0 0 0.1]
Baixo (B)	[0 0.1 0.3]
Médio Baixo (MB)	[0.1 0.3 0.5]
Médio (M)	[0.3 0.5 0.7]
Médio Alto (MA)	[0.5 0.7 0.9]
Alto (A)	[0.7 0.9 1.0]
Altíssimo (AA)	[0.9 1.0 1.0]

Fonte – Adaptado de Chen (2000).

O gráfico da Figura 4.6 representa a distribuição dos pontos das variáveis linguísticas.

Figura 4.6 - Distribuição das variáveis linguística no gráfico Fuzzy



Fonte – Adaptado de Chen (2000).

Uma vez que as variáveis foram definidas, os critérios de decisão devem ser escolhidos. Os critérios de desempenho (*Makespan* (C_m), *Total Workload* (W_t) e *Maximum Workload* (W_m)) são selecionados diretamente por serem os objetivos da otimização e os critérios relacionados ao ambiente do chão de fábrica são selecionados de acordo com as necessidades desejadas, no momento, para o sistema produtivo, ou seja, critérios tais como o menor tempo de máquina ociosa ou a não utilização de uma determinada máquina na produção podem permitir a escolha do melhor cenário para essa ocasião, sem perdas ou com as menores perdas possíveis. Esses cenários podem ser visualizados nos gráficos de Gantt das soluções do FJSP-ABC, pois fazem parte dos dados agregados nas soluções de Pareto alocadas no arquivo de Pareto, encontrado anteriormente.

Na implementação realizada o DM utilizou os critérios C_m , W_t , W_m os quais foram otimizados pelo algoritmo FJSP – ABC e o critério de ociosidade de cada máquina. O critério de ociosidade de cada máquina é atribuído individualmente a cada máquina do sistema produtivo. Portanto, de acordo com cada cenário estudado a quantidade de variáveis de decisão vai então depender da quantidade de máquinas do cenário.

Conforme a Seção 2.5, uma matriz de grau de importância de cada critério pode ser estabelecida para auxiliar na execução do algoritmo. Dessa forma baseado na Tabela 4.4 pode-se construir a matriz de grau de importância de cada critério, ilustrada na Tabela 4.5:

Tabela 4.5 – Grau de importância de cada critério

Cm	Wt	Wm	M1	M2	...	Mn
AA	AA	AA	BB	BB	...	BB

Fonte – Próprio autor.

A Tabela 4.5 ilustra o tomador de decisão D com n critérios de decisão C , onde C_m é o Makespan, W_t é a Carga total das máquinas (Total Workload), W_m é maior carga de uma máquina e M_n são as cargas ociosas das máquinas do cenário.

O grau de importância de cada critério é selecionado para cada critério de tomada de decisão de acordo com as necessidades do sistema produtivo, como destacado na Tabela 4.4.

Utilizando-se o arquivo de Pareto, se é estabelecida a matriz de Avaliação de cada alternativa de cenário (utiliza-se para isto os gráficos de Gantt). Esta matriz estabelece os valores extraídos dos gráficos de Gantt para cada critério estabelecido em relação aos distintos cenários apresentados pelo arquivo de Pareto. A Tabela 4.6 ilustra a matriz de avaliação de cada alternativa para um cenário com 6 máquinas.

Tabela 4.6 – Matriz Fuzzy dos critérios de desempenho e Alternativas

Cm	Wt	Wm	M1	M2	M3	M4	M5	M6
28	146	27	1	0	1	12	1	1
29	146	27	1	0	0	12	1	2
29	146	27	1	0	1	12	1	2
30	145	27	2	1	0	13	1	3
29	146	27	1	0	1	16	1	2
29	146	27	1	1	2	12	1	2
29	146	27	1	1	2	12	1	2
29	146	27	1	0	1	15	1	3
29	146	27	1	1	0	13	2	2
29	146	27	1	0	1	16	1	2

Os valores da matriz da Tabela 4.6 são alocados em conjuntos triangulares fuzzy de forma a estabelecerem a matriz de decisão fuzzy. É necessária uma normalização destes valores para que se possa processar a inferência fuzzy.

4.1.9 Normalização das Soluções

A partir da definição do das variáveis linguísticas e do critério de decisão, é necessário a normalização das soluções, conforme descrito na Seção 2.5. O algoritmo de normalização é ilustrado no Algoritmo 4.6 abaixo:

Algoritmo 4.6 – Normalização das Soluções

```

1. Para m =1 até Quantidade de soluções
2. Para n =1 até Quantidade de critérios
3.  $r_{mn} = \frac{\max[x_{mn}] - x_{mn}}{[\max[x_{mn}] - \min[x_{mn}]]}$ 
4. Fim_Para
5. Fim_Para

```

Fonte – Próprio autor.

Foi utilizado, como ilustrado no Algoritmo 4.56 r_{mn} , gerando a matriz de soluções normalizadas R , como ilustrada na Figura 4.6.

Figura 4.6 – Matriz de Critérios de desempenho Normalizada

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix}$$

Fonte – Próprio autor.

Aplicando a equação de normalização na tabela 4.6, obtemos a tabela 4.7 com os valores dos critérios de desempenho normalizados.

Tabela 4.7 – Matriz Fuzzy Normalizada dos critérios de desempenho e Alternativas

Cm	Wt	Wm	M1	M2	M3	M4	M5	M6
1	0	0	1	1	0,5	1	1	1
0,5	0	0	1	1	1	1	1	0,5
0,5	0	0	1	1	0,5	1	1	0,5
0	1	0	0	0	1	0,75	1	0
0,5	0	0	1	1	0,5	0	1	0,5
0,5	0	0	1	0	0	1	1	0,5
0,5	0	0	1	0	0	1	1	0,5
0,5	0	0	1	1	0,5	0,25	1	0
0,5	0	0	1	0	1	0,75	0	0,5
0,5	0	0	1	1	0,5	0	1	0,5

4.1.10 Criação da Matriz de Decisão Fuzzy

Na sequência, é realizada a criação da Matriz Fuzzy, onde o Algoritmo 4.7 ilustra a implementação da criação da matriz Fuzzy que é o produto escalar de cada elemento pelo peso do critério utilizado.

Algoritmo 4.7 – Matriz de Decisão Fuzzy

```

1. Para m =1 até Quantidade de soluções
2.     Para n =1 até Quantidade de critérios
3.         Para l =1 até Quantidade de critérios
4.             V(m,n) = R(m,n) * D1(l)
5.         Fim_Para
6.     Fim_Para
7. Fim_Para

```

Fonte – Próprio autor.

A matriz Fuzzy V é montada a partir da matriz R de normalização e do valor dos pesos de cada critério de desempenho estabelecido para cada variável linguística \tilde{w}_n . A Figura 4.7 ilustra a matriz V .

O cálculo do vetor de importância dos critérios (peso W) utiliza os valores da tabela 4.4, aplicada às escolhas da tabela 4.5. Este vetor pode ser ilustrado para o cenário de 6 máquinas, por exemplo.

$$\tilde{W} = [AA, AA, AA, BB, BB, BB, BB, BB, BB]$$

$$\tilde{W} = [(0.9 \ 1 \ 1) (0.9 \ 1 \ 1) (0.9 \ 1 \ 1) (0 \ 0 \ 0.1) (0 \ 0 \ 0.1) (0 \ 0 \ 0.1) (0 \ 0 \ 0.1) (0 \ 0 \ 0.1) (0 \ 0 \ 0.1)]$$

Figura 4.7 – Criação da Matriz Fuzzy

$$\tilde{W} = [\tilde{w}_1, \tilde{w}_2, \tilde{w}_3 \dots \tilde{w}_n],$$

$$\tilde{w}_n = [w_{n1}, w_{n2}, w_{n3}].$$

$$\tilde{v}_{nm} = [\tilde{w}_{n1} \times r_{nm}, \tilde{w}_{n2} \times r_{nm}, \tilde{w}_{n3} \times r_{nm}, \dots]$$

$$V = \begin{bmatrix} \tilde{v}_{11} & \tilde{v}_{12} & \dots & \tilde{v}_{1n} \\ \tilde{v}_{21} & \tilde{v}_{22} & \dots & \tilde{v}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{v}_{m1} & \tilde{v}_{m2} & \dots & \tilde{v}_{mn} \end{bmatrix}$$

Fonte – Próprio autor.

A matriz Fuzzy resultante do exemplo apresentado na Tabela 4.7 pode ser ilustrada na Tabela 4.8.

Tabela 4.8 - Exemplo numérico da Matriz Fuzzy para os 9 critérios multiobjetivos

C ₁			C ₂			C ₃			C ₄			C ₅			C ₆			C ₇			C ₈			C ₉		
Cm			Wt			Wm			M1			M2			M3			M4			M5			M6		
0,9	1	1	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,05	0	0	0,1	0	0	0,1	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,1	0	0	0,1	0	0	0,1	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,05	0	0	0,1	0	0	0,1	0	0	0,1
0	0	0	0,9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0	0	0,075	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,05	0	0	0	0	0	0	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0	0	0	0,1	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0	0	0	0	0,1	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,05	0	0	0,025	0	0	0,1	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0	0	0	0,1	0	0	0,075	0	0	0	0	0	0,1
0,45	0,5	0,5	0	0	0	0	0	0	0	0	0,1	0	0	0,1	0	0	0,05	0	0	0	0	0	0	0	0	0,1

Fonte – Próprio autor.

4.1.11 Definir FPIS e FNIS

Como o objetivo da otimização foram as minimizações dos critérios de desempenho, é necessário utilizar equações específicas no momento da normalização, e para os FPIS e FNIS. Para normalização é utilizado a equação (4) da Seção 2.5, para os FPIS e FNIS deve ser considerado o valor mínimo e máximo, respectivamente, da matriz de decisão Fuzzy de cada critério. O Algoritmo 4.8 ilustra essa implementação.

Algoritmo 4.8 – FPIS e FNIS

1.	Para $i = 1$ até quantidade de critérios de desempenho
2.	$A^+(i) = \text{Min} [\tilde{v}(i)]$
3.	$A^-(i) = \text{Max} [\tilde{v}(i)]$
4.	Fim_para

Fonte – Próprio autor.

4.1.12 Cálculo da Distância Euclidiana e do Coeficiente de Proximidade

A próxima etapa é o cálculo da distância S^+ entre FPIS e a distância S^- e FNIS e para isso é necessário determinar a distância euclidiana entre os valores Fuzzy de cada atributo e os valores de FPIS e FNIS. Como $\tilde{v}_{nm} = [v_1 \ v_2 \ v_3]$ e $A_n = [a_1 \ a_2 \ a_3]$, pode-se calcular a distância $d(\tilde{A}, \tilde{V})$.

$$d(\tilde{A}, \tilde{V}) = \sqrt{\frac{(a_1 - v_1)^2 + (a_2 - v_2)^2 + (a_3 - v_3)^2}{3}}, \quad (16)$$

sendo (a_1, a_2, a_3) as coordenadas do conjunto fuzzy triangular \tilde{A} e (v_1, v_2, v_3) as coordenadas do conjunto \tilde{v}_{mn} .

Uma vez definida as distancias euclidianas, as distâncias S^+ e S^- podem ser calculadas por:

$$S_i^+ = \sum_{j=1}^n d(A_j^+, v_{ij}) \quad (17)$$

e

$$S_i^- = \sum_{j=1}^n d(A_j^-, v_{ij}), \quad (18)$$

sendo $i = \{1, 2, 3, \dots, n\}$, $j = \{1, 2, 3, \dots, m\}$ e as equações (17) e (18) as distâncias de cada solução do FPIS e do FNIS. Com os valores das distâncias de cada solução pode-se, então, calcular o coeficiente de proximidade CC_j que é dado pela equação 19 tem-se:

$$CC_j = \frac{S_j^-}{S_j^+ + S_j^-}, \quad (19)$$

sendo $j = \{1, 2, 3, \dots, m\}$.

A Equação 19 irá resultar em um valor entre 0 e 1, para cada solução. Assim, o *ranking* de preferência das alternativas apresentará como melhor solução a alternativa que tiver o maior valor do coeficiente de proximidade, seguida pelas demais em ordem decrescente.

4.2 Validação dos resultados

Para validar o funcionamento dos algoritmos propostos, serão realizados testes utilizando diversas instâncias dos *benchmarks* de Kacem, Hammadi e Borne (2002) e Brandimarte (1993).

A verificação do resultado obtidos nos algoritmos do FJSP multiobjetivo utilizado a meta-heurística ABC e o DM de tomada de decisão utilizando o Fuzzy-TOPSIS serão feitas das seguintes formas:

- Gráfico de Gantt: através da análise do gráfico de Gantt é possível verificar que as soluções geradas respeitam as todas as restrições do FJSP e atendem os pesos dos critérios estabelecidos para cada variável;
- Gráfico da fronteira de Pareto: através do gráfico da fronteira de Pareto é possível verificar se soluções apontadas pelo tomador de decisão são as soluções não dominadas do conjunto testado ou estão em aderência com os melhores resultados encontrados na Fronteira de Pareto, considerando os critérios de decisão e os de desempenho estabelecidos para o sistema produtivo.

4.3 Resultados esperados

Ao final da execução espera-se que algoritmo realize o ranqueamento das melhores programações da produção (scheduling) de acordo com os parâmetros de decisão de C_m , W_t , W_m e de ociosidade de cada máquina, presente um conjunto de soluções encontradas. É desejado que estas soluções estejam perto ou na Fronteira de Pareto de acordo o desempenho desejado para o sistema produtivo.

CAPÍTULO 5

RESULTADOS

Este capítulo apresenta os resultados obtidos através dos experimentos realizados com a aplicação da metodologia de programação da produção proposta no capítulo anterior. Na metodologia, há a execução da programação da produção utilizando o algoritmo FJSP-ABC, o qual fornece o conjunto de Pareto que será utilizado para geração das soluções na forma de gráficos de Gantt, os quais serão analisados pelo DM para se ranquear as melhores soluções da programação da produção, de acordo com as variáveis multiobjetivos utilizadas e as variáveis individuais de ociosidade de cada máquina.

O DM baseado no FUZZY TOPSIS deve realizar a escolha da solução mais adequada para FJSP, selecionando os melhores cenários a partir dos gráficos de Gantt, ranqueando os melhores resultados. Dessa forma, para a tomada de decisão, o DM pode ser formulado para as variáveis de decisão de acordo com as necessidades do processo produtivo.

Para a implementação do DM utilizou-se dois cenários diferentes, o primeiro cenário trata de estabelecer a maior carga de trabalho consecutivo para a máquina escolhida e o segundo cenário o de estabelecer a soma do maior tempo ocioso entre as operações realizadas na máquina escolhida.

Os critérios C_m , W_i e W_m , os quais foram otimizados pelo algoritmo FJSP-ABC, e os critérios de máquina com mais ou menos ociosidade entre operações foram utilizados para compor os cenários de execução do DM. O critério de máquina menos ociosidade entre operações prioriza que as operações sejam alocadas sequencialmente sem intervalos de tempo na máquina escolhida, enquanto o critério de maior ociosidade entre operações prioriza a escolha do cenário em que a soma dos intervalos de tempo entre operações na máquina seja alto.

Como a atribuição é realizada individualmente a cada máquina do sistema produtivo, a quantidade de variáveis de decisão vai depender da quantidade de máquinas de cada cenário estudado.

Para se avaliar o DM, os conjuntos de soluções foram submetidos a diferentes layouts de produção, onde foram utilizados os benchmarks de Kacem, Hammadi e Borne (2002) e os de

Brandimarte (1993). O algoritmo proposto foi desenvolvido em MatLab e testado em um PC com 2,6 GHz e 16 GB de RAM.

As instâncias de Kacem, Hammadi e Borne (2002) utilizadas foram a 4x5, a 8x8 e a 10x10. A instância 4x5, que é considerada uma instância pequena, é composta por cinco máquinas e quatro *jobs*, que somam doze operações, ela tem a característica de ser totalmente flexível, isto é, todas as máquinas podem executar qualquer operação. A 8x8 que é a única instância parcialmente flexível, isto é, onde algumas operações não são realizadas por algumas máquinas, é formada por oito máquina, vinte e sete operações divididos entre oito *jobs*. E a instância 10x10 possui dez máquinas, dez *jobs* e trinta operações, ela assim como a 4x5 é totalmente flexível.

Os cenários das instâncias de Brandimarte (1993) (BRData) foram as instâncias MK2, a MK4 e a MK8. Os cenários diferem das instâncias de Kacem, Hammadi e Borne (2002) por serem parcialmente flexível, onde as operações possuem poucas máquinas para serem executadas, com uma quantidade maior de Jobs e operações, o que faz com que o problemas seja mais complexo, porém mais próximo da realidade. A instância MK2 é composta por 6 máquinas, 10 jobs e 58 operações, a instância MK4 é composta por 8 máquinas, 15 jobs e 90 operações e instância MK8 é composta por 10 máquinas, 20 jobs e 225 operações.

Para a execução do algoritmo FJSP-ABC foram usados os parâmetros apresentados da Tabela 5.1.

Tabela 5.1 - Parâmetros utilizados nos testes do algoritmo FJSP multiobjetivo ABC

Nome da variável	Instâncias		
	4x5	8x8	10x10
<i>PROBLEMA</i>	1	2	3
<i>Num_sol</i>	1000	1000	1000
<i>Num_abelhas</i>	100	100	100
<i>Mudancas_Aem</i>	25	25	25
<i>Mudancas_Aes</i>	25	25	25
<i>MaxGen</i>	500	2000	3000
<i>Limite</i>	20	20	20
<i>Num_op_Troca_maq</i>	2	4	4

Fonte – Próprio autor.

O algoritmo FJSP-ABC fornece para cada instância (4x5, 8x8, 10x10, MK2, MK4 e MK8) um conjunto de soluções, as quais são armazenadas em um arquivo, como o objetivo de ser utilizado pelo DM. O arquivo gerado pelo algoritmo FJSP-ABC é composto pelas soluções ótimas e soluções perto das ótimas, ou seja, na Fronteira de Pareto e perto da Fronteira de Pareto, além dos seus respectivos gráficos de Gantt.

O arquivo com as soluções parciais de programação da produção, pode ser chamado de conjunto de Pareto possui os valores de C_m , W_m e W_t encontrados na forma otimizada pelo algoritmo FJSP-ABC. Também possui os gráficos de Gantt gerada para cada solução, onde cada conjunto de solução C_m , W_m e W_t encontradas podem gerar mais de um gráfico de Gantt, ou seja, diferentes gráficos de Gantt podem ter os mesmos valores de C_m , W_m e W_t .

O algoritmo tomador de decisão (DM) irá realizar a escolha do melhor cenário, baseado na programação da produção ilustrada pelo gráfico de Gantt, de acordo com as variáveis Fuzzy linguísticas estabelecidas pelo usuário a cada critério de decisão escolhido.

Para os experimento realizado, os critérios de decisão foram C_m , W_m , W_t e os critérios de ociosidade entre operações de uma máquina, e as variáveis linguísticas utilizadas foram as descritas conforme a Tabela 4.4.

Os resultados do DM estabelecem as 10 melhores soluções e os seus respectivos gráficos de Gantt. Para os testes foram utilizados seis (6) casos diferentes de escolha dos pesos de importância de cada critério, conforme a Tabela 4.4.

Os gráficos de Gantt para cada solução ranqueada são fornecidos, permitindo a validação do resultado em função dos critérios estabelecidos e um gráfico de comparação com a Fronteira de Pareto estabelece visualmente a relação dos resultados obtidos com os valores não dominados.

O gráfico que ilustra os resultados encontrados pelo DM e a Fronteira de Pareto é apresentado na forma 2D, sendo analisados em função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$. Também é realizada uma medida da distância média entre os resultados encontrados pelo DM e os não dominados (da fronteira de Pareto), sendo essa medida ilustrada no próprio gráfico.

A distância média estabelecida é a média das distâncias encontradas pela menor distância da solução do DM à Fronteira de Pareto, isto é, a cada ponto do gráfico de uma solução do DM é calculada a distância a todos os pontos não dominados e a menor distância desse ponto irá compor a média.

A distância mínima das soluções estabelecidas pelo DM em relação à fronteira de Pareto é calculada ponto a ponto em cada gráfico como ilustra as Equações 20 e 21, onde $n = 1, 2, 3, \dots, k$, são os pontos não dominados da Fronteira de Pareto $fn(x,y)$ e $dm(x,y)$ são os pontos encontrados pelo DM no gráfico em relação aos eixos x e y .

$$dm_{min} = \min \sum_{i=1}^n d(f_n, dm) \quad (20)$$

$$d(f_n, dm) = \sqrt{(fn_{x2} - dm_{x1})^2 + (fn_{y2} - dm_{y1})^2} \quad (21)$$

Assim, a Equação 22 ilustra o cálculo da média das distâncias encontradas pelo DM para cada caso estudado, onde m é o total de soluções diferentes encontradas pelo DM.

$$d_{media} = \frac{1}{m} \sum_{j=1}^m dm_{min} \quad (22)$$

A Tabela 5.2 ilustra a compilação dos resultados não dominados encontrados na literatura (Kato et. al. 2018) (Wang et. al. 2010) (Xia & Wu, 2005) (Li et. al., 2010) (Li et. al., 2011). Estes resultados compõem a Fronteira de Pareto para as soluções encontradas para essas instâncias.

Tabela 5.2 – Resultados na Fronteira de Pareto para as instancias de Kacem et. al. (2002)

Instâncias								
4x5			8x8			10x10		
C_m	W_m	W_t	C_m	W_m	W_t	C_m	W_m	W_t
11	10	32	14	12	77	7	5	43
12	8	32	15	12	75	7	6	42
13	7	33	16	11	77	8	5	42
11	9	34	16	13	73	8	7	41
			15	11	81			

Fonte – Próprio autor.

A Tabela 5.3 ilustra a compilação dos resultados não dominados encontrados na literatura (Wang et. al. 2010) (Shao et al. ,2013) (Deng et al., 2017) (Chiang & Lin, 2012). Estes resultados compõem a Fronteira de Pareto para as soluções encontradas para essas instâncias.

Tabela 5.3 – Resultados na Fronteira de Pareto para as instancias de Brandimarte (1993)

Instâncias		
MK2	MK4	MK8

C_m	W_m	W_t	C_m	W_m	W_t	C_m	W_m	W_t
26	26	151	60	60	372	523	515	2524
27	27	145	60	59	390	523	497	2534
28	28	144	61	60	366	524	524	2519
29	29	143	62	60	363	533	533	2514
30	30	142	62	61	360	542	542	2509
31	31	141	63	60	360	551	551	2504
33	33	140	64	62	353	560	560	2499
			66	66	345	569	569	2494
			68	67	372	578	578	2489
			69	67	343	587	587	2484
			72	72	340			
			84	84	334			
			90	90	331			
			114	114	328			
			122	122	327			
			130	130	326			
			138	138	325			
			146	146	324			

Fonte – Próprio autor.

Utilizando os valores das soluções não dominadas (Tabela 5.2 e 5.3) e as soluções ranqueadas pelo DM pode se estabelecer os gráficos de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ na forma de uma nuvem de pontos para cada caso estudado por instâncias utilizadas.

Os próximos itens ilustram os cenários estudados onde o DM é configurado para priorizar as soluções da programação da produção que possuam máquinas com menor tempo ocioso possível entre as suas operações (Cenário 1) e o cenário em que o DM é configurado para priorizar as soluções de programação da produção que possuam máquinas com maior tempo ocioso possível entre suas operações (Cenário 2). Dentre esses cenários são estudados vários casos de configuração dos pesos dos critérios de forma diferente, onde alguns serão apresentados para se realizar a verificação e a validação do funcionamento do algoritmo implementado pelo DM.

5.1 Cenário 1: Prioridade para menor ociosidade entre operações

Para o cenário com prioridade para a máquina com menor ociosidade entre operações, o usuário pode estabelecer os critérios de decisão para buscar as programações da produção onde determinada máquina pode ter os intervalos de tempo entre suas operações minimizados. Este cenário pode ser interessante devido às necessidades de parada de máquina programada ou não.

O algoritmo do DM foi executado para seis (6) casos diferentes, onde os pesos de cada critério foram alterados de forma a ilustrar uma situação real possível no sistema de fabricação. Os resultados utilizam as instâncias de Kacem, Hammadi e Borne (2002) e Bradimarte (1993) nos seis casos estudados, conforme ilustrado nas tabelas 5.4, 5.5 e 5.6 para Kacem e tabelas 5.7, 5.8 e 5.9 para Bradimarte.

Tabela 5.4 – Pesos dos critérios para Kacem 4x5 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5
caso 1	AA	AA	AA	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	AA	BB	BB	BB	BB
caso 3	AA	AA	AA	A	BB	BB	BB	BB
caso 4	AA	AA	AA	M	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	AA	BB
caso 6	AA	AA	AA	BB	BB	BB	A	BB

Fonte – Próprio autor.

Tabela 5.5 – Pesos dos critérios para Kacem 8x8 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB
caso 3	AA	AA	AA	A	BB	BB	BB	BB	BB	BB	BB
caso 4	AA	AA	AA	M	BB	BB	BB	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	BB	BB	BB	AA	BB
caso 6	AA	AA	AA	BB	BB	BB	BB	BB	BB	A	BB

Fonte – Próprio autor.

Tabela 5.6 – Pesos dos critérios para Kacem 10x10 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	BB	BB	AA	BB	BB	BB	BB	BB	BB	BB
caso 3	AA	AA	AA	BB	BB	A	BB	BB	BB	BB	BB	BB	BB
caso 4	AA	AA	AA	BB	BB	M	BB	BB	BB	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	AA	BB
caso 6	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	A	BB

Fonte – Próprio autor.

Tabela 5.7 – Pesos dos critérios para Bradimarte MK2 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB
caso 2	BB	BB	BB	AA	AA	AA	AA	AA	AA
caso 3	A	AA	BB	A	M	AA	M	B	MB
caso 4	MA	M	MA	BB	AA	M	A	M	AA
caso 5	BB	MA	MA	MB	M	MB	B	MA	B
caso 6	AA	B	MA	B	MB	BB	MA	AA	M

Fonte – Próprio autor.

Tabela 5.8 – Pesos dos critérios para Bradimarte MK4 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	BB	BB	BB	AA	AA	AA	AA	AA	AA	AA	AA
caso 3	A	AA	BB	A	M	AA	M	B	MB	AA	BB
caso 4	MA	M	MA	BB	AA	M	A	M	AA	MA	AA
caso 5	BB	MA	MA	MB	M	MB	B	MA	B	AA	AA
caso 6	AA	B	MA	B	MB	BB	MA	AA	M	MA	AA

Fonte – Próprio autor.

Tabela 5.9 – Pesos dos critérios para Bradimarte MK8 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	BB	BB	BB	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA
caso 3	A	AA	BB	A	M	AA	M	B	MB	AA	BB	MB	AA
caso 4	MA	M	MA	BB	AA	M	A	M	AA	MA	AA	MA	BB
caso 5	MA	MB	BB	MA	MA	MB	M	MB	B	MA	B	AA	AA
caso 6	AA	B	MA	B	MB	BB	MA	AA	M	AA	M	MA	AA

Fonte – Próprio autor.

As análises foram realizadas para todos os cenários e casos descritos nas tabelas 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9, no entanto, nas seções seguintes serão apresentados somente alguns casos mais relevantes. Os casos apresentados a seguir estão em destaque nas tabelas acima.

5.1.1 Kacem 4x5 caso 1

Para o cenário Kacem 4x5 caso 1 a solução do DM para o cenário utilizado é apresentada na Tabela 5.10 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.10 – Resultado do DM para Kacem 4x5 caso 1 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB			
1	12	32	9	4	2	0	3	2	2,982	0,258	0,920
2	13	32	8	5	1	0	5	3	2,966	0,273	0,916
3	12	32	10	0	2	2	4	0	2,926	0,313	0,903
4	12	32	10	0	1	1	4	2	2,926	0,314	0,903
5	12	32	10	0	1	2	4	3	2,919	0,321	0,901
6	12	32	10	2	1	2	6	3	2,905	0,335	0,897
7	12	32	10	3	1	2	6	3	2,900	0,339	0,895
8	14	32	8	4	1	0	6	2	2,896	0,344	0,894
9	12	32	10	5	3	1	8	2	2,885	0,355	0,891
10	14	33	7	5	1	0	7	1	2,864	0,375	0,884

Fonte – Próprio autor.

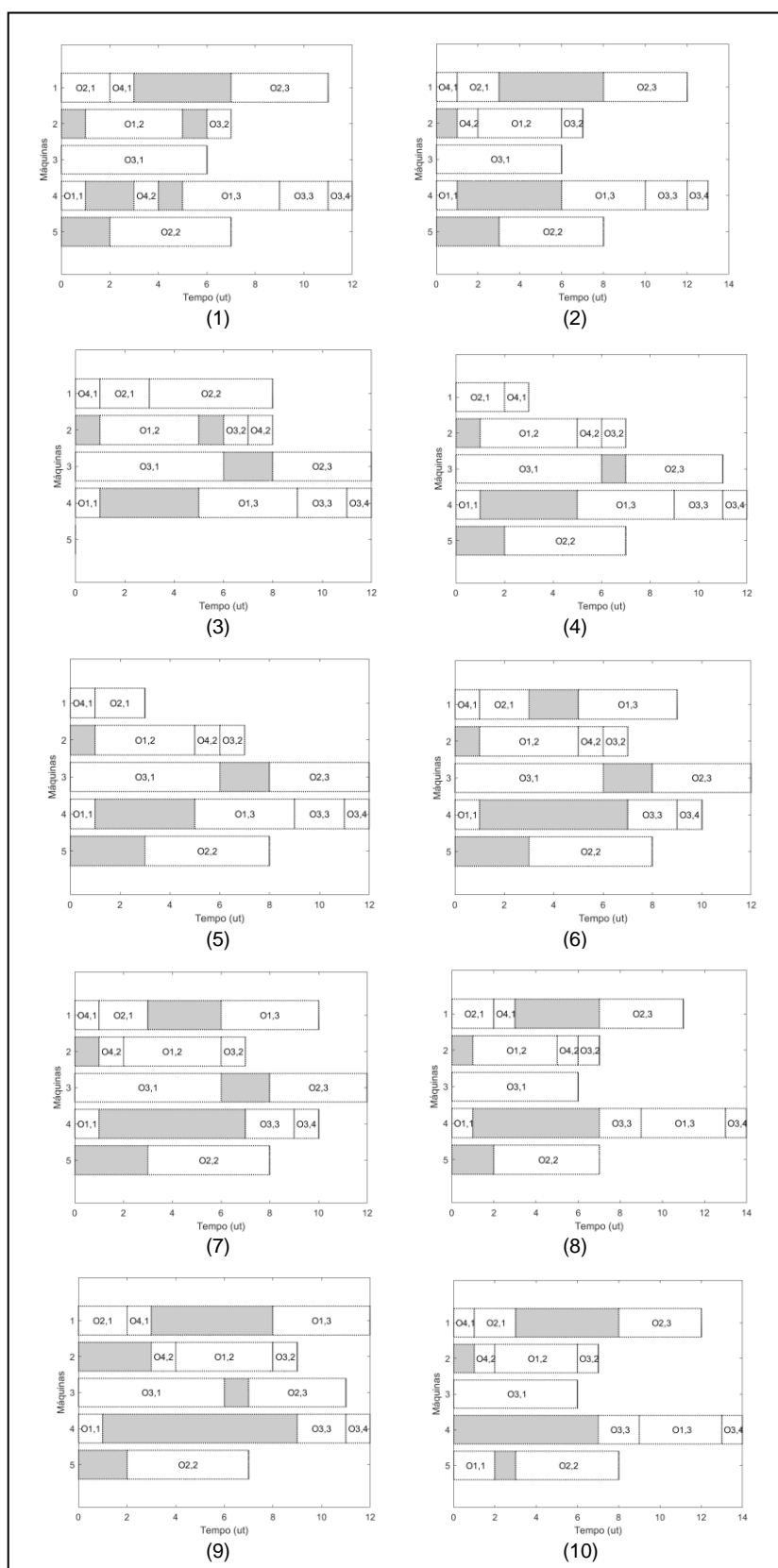
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.1.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 4x5 caso 1 são mostrados na Figura 5.2.

Nos experimentos relacionados à instância 4x5, forma colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e baixíssima prioridade (BB) para as máquinas, ou seja, é esperado que no ranqueamento dos resultados os que estão mais próximos da fronteira de Pareto fiquem nas primeiras posições.

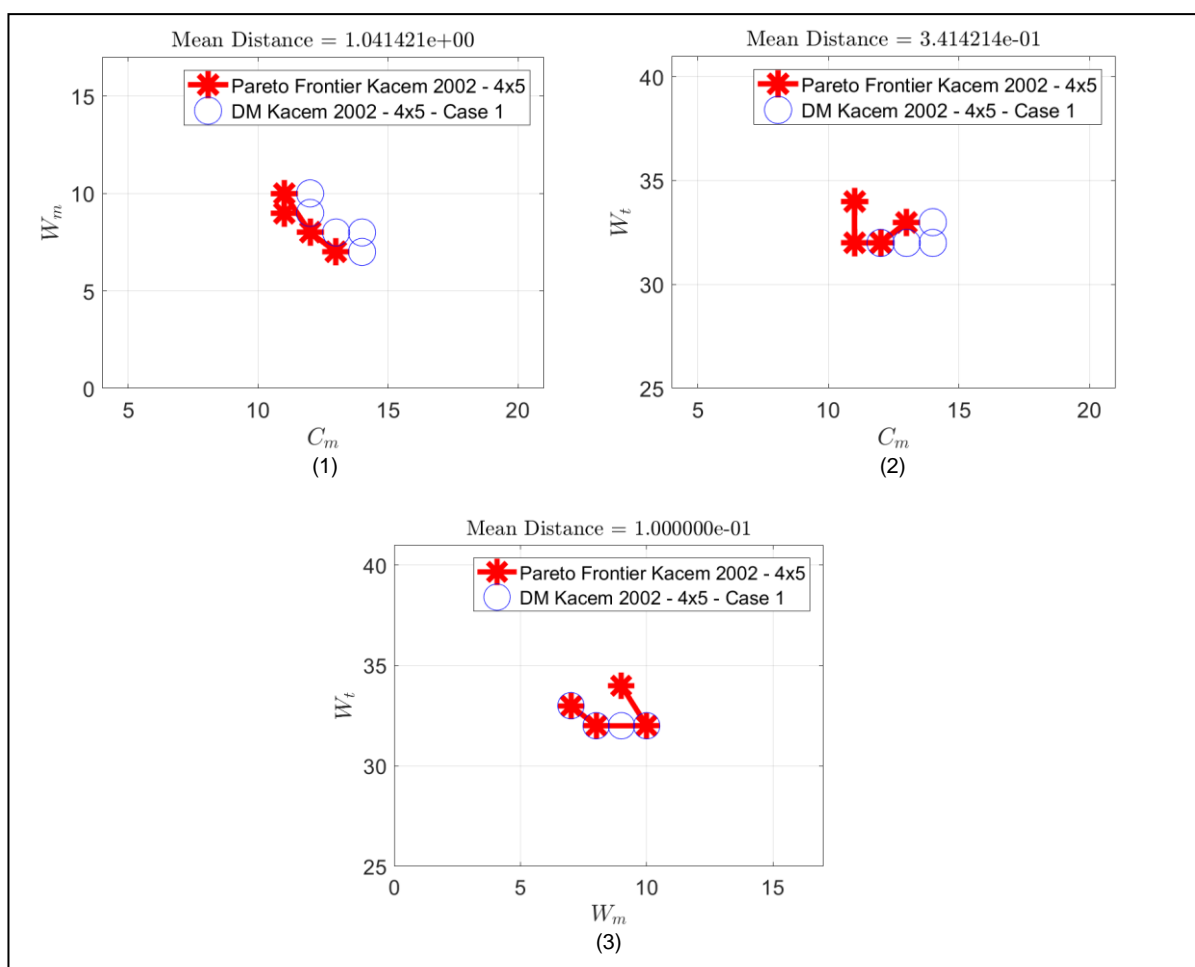
Pode-se observar que, para cada uma das soluções encontradas de C_m , W_t e W_m fornecidas pelo algoritmo FJSP-ABC, pode haver gráficos de Gantt diferentes, é o caso, por exemplo, da solução $C_m = 12$, $W_t = 32$ e $W_m = 10$ (em negrito na Tabela 5.7) que possui 5 gráficos de Gantt diferentes veja na Figura 5.3 os gráficos (3), (4), (5), (6) e (9).

Figura 5.1- Gráficos de Gantt da instância 4x5 caso 1 (cenário 1).



Fonte – Próprio autor.

Figura 5.2- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância 4x5 caso 1 (cenário 1).



Fonte – Próprio autor.

Na Figura 5.2 observa-se que as soluções encontradas pelo DM estão muito próximas ou sobrepondo-se à Fronteira de Pareto (valores não dominados) e possuem uma distância média muito baixa, confirmando-se o que era esperado para esta maneira de escolha dos pesos dos critérios de seleção dos cenários.

5.1.2 Kacem 4x5 caso 2

Para a instância Kacem 4x5 caso 2 a solução do DM é apresentada na Tabela 5.11 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.11 – Resultado do DM para Kacem 4x5 caso 2 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	SIN	SIP	CCI
	AA	AA	AA	AA	BB	BB	BB	BB			
1	12	32	10	0	2	2	4	0	3,852	0,313	0,925
2	12	32	10	0	1	1	4	2	3,852	0,314	0,925
3	12	32	10	0	1	2	4	3	3,845	0,321	0,923
4	12	34	8	0	3	0	4	7	3,771	0,394	0,905
5	12	33	10	0	6	2	4	1	3,733	0,432	0,896
6	12	33	10	0	1	2	6	6	3,729	0,437	0,895
7	12	32	12	0	1	0	4	0	3,722	0,443	0,894
8	14	33	8	0	2	0	6	8	3,713	0,453	0,891
9	12	32	12	0	1	2	6	0	3,710	0,456	0,891
10	12	32	12	0	2	2	6	0	3,706	0,460	0,890

Fonte – Próprio autor.

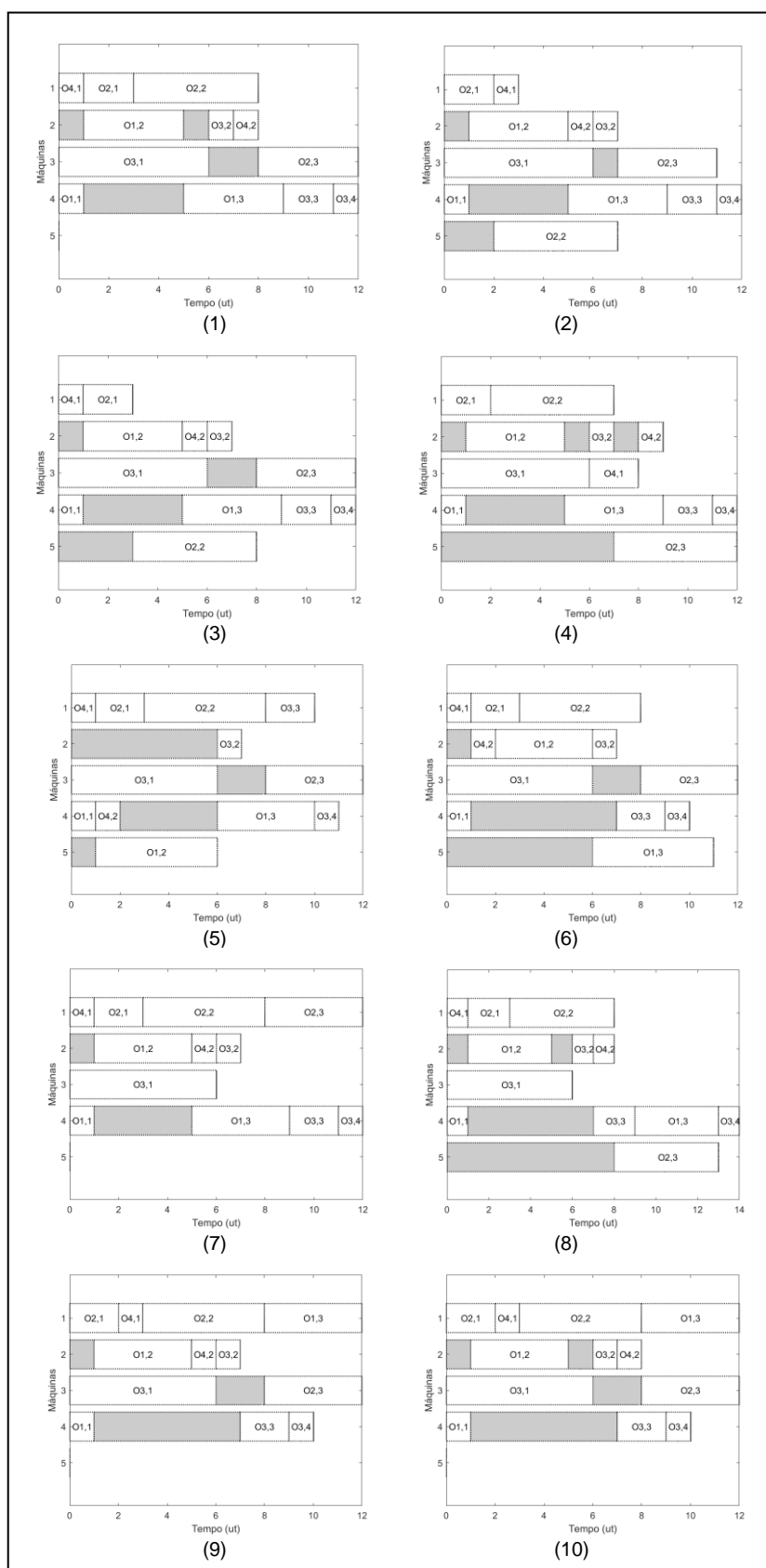
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.3.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 4x5 caso 2 são mostrados na Figura 5.4.

Nos experimentos relacionados à instância 4x5 caso 2, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e também para M1, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

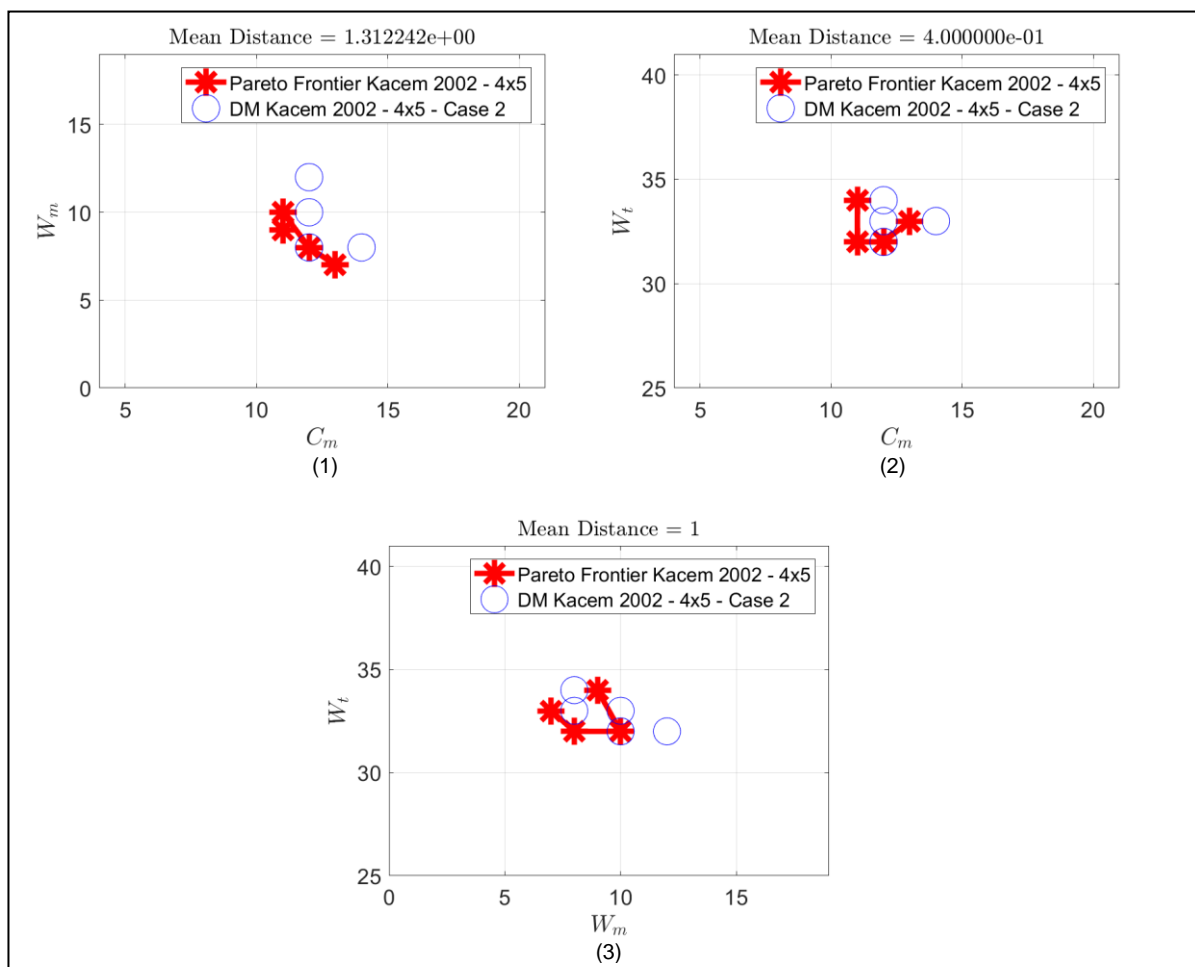
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção.

Figura 5.3- Gráficos de Gantt da instância 4x5 caso 2 (cenário 1).



Fonte – Próprio autor.

Figura 5.4- Gráficos $C_m W_m$, $C_m W_t$ e $W_m W_t$ da instância 4x5 caso 2 (cenário 1).



Fonte – Próprio autor.

É esperado que no ranqueamento do DM sejam estabelecidos cenários onde a máquina M1 possua pouco tempo ocioso entre as operações, ou seja, as operações estejam alocadas uma em seguida da outra. Isto pode ser observado na Figura 5.3, onde em todos os gráficos de Gantt escolhidos pelo DM, não há intervalos de tempo ocioso entre as operações alocadas à máquina.

Pode-se observar que, para cada uma das soluções encontradas de C_m , W_t e W_m fornecidas pelo algoritmo FJSP-ABC, pode haver gráficos de Gantt diferentes, é o caso, por exemplo, da solução $C_m = 12$, $W_t = 32$ e $W_m = 10$ (em negrito na Tabela 5.8) que possui 4 gráficos de Gantt diferentes veja na Figura 5.3 os gráficos (1), (2), (3) e (4).

Na Figura 5.4 observa-se novamente que as soluções encontradas pelo DM estão muito próximas ou sobrepondo-se à Fronteira de Pareto (valores não dominados) e possuem uma distância média muito baixa, confirmando-se o que era esperado para esta maneira de escolha dos pesos dos critérios de seleção dos cenários.

5.1.3 Kacem 8x8 caso 2

Para a instância Kacem 8x8 caso 2 a solução do DM é apresentada na Tabela 5.12 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.12 – Resultado do DM para Kacem 8x8 caso 2 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP	CCI
	AA	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB			
1	18	79	13	1	0	0	2	7	3	5	4	4,143	0,196	0,955
2	18	79	13	1	1	0	2	6	2	6	5	4,140	0,199	0,954
3	18	79	13	1	1	0	2	7	3	5	5	4,137	0,202	0,954
4	18	82	13	0	1	0	2	7	3	5	8	4,131	0,208	0,952
5	18	82	13	0	3	0	2	7	3	5	8	4,125	0,214	0,951
6	18	82	13	0	6	0	2	5	2	6	8	4,121	0,218	0,950
7	18	82	13	0	6	0	2	7	2	6	8	4,116	0,222	0,949
8	18	84	13	1	1	0	2	7	2	4	1	4,091	0,248	0,943
9	18	84	13	1	1	0	2	7	2	4	1	4,091	0,248	0,943
10	18	84	13	1	1	0	2	7	2	4	1	4,091	0,248	0,943

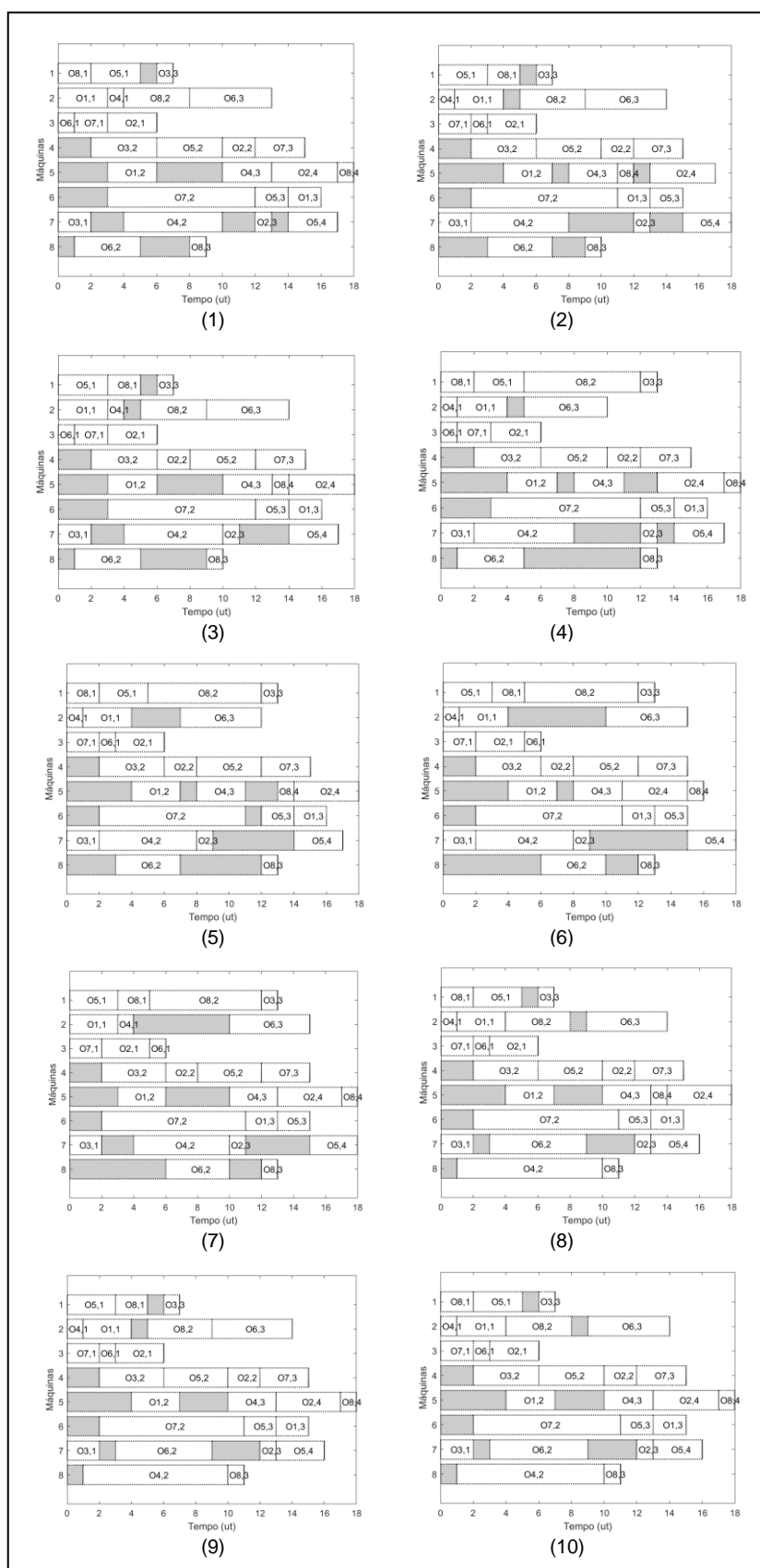
Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.5.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 8x8 caso 2 são mostrados na Figura 5.6.

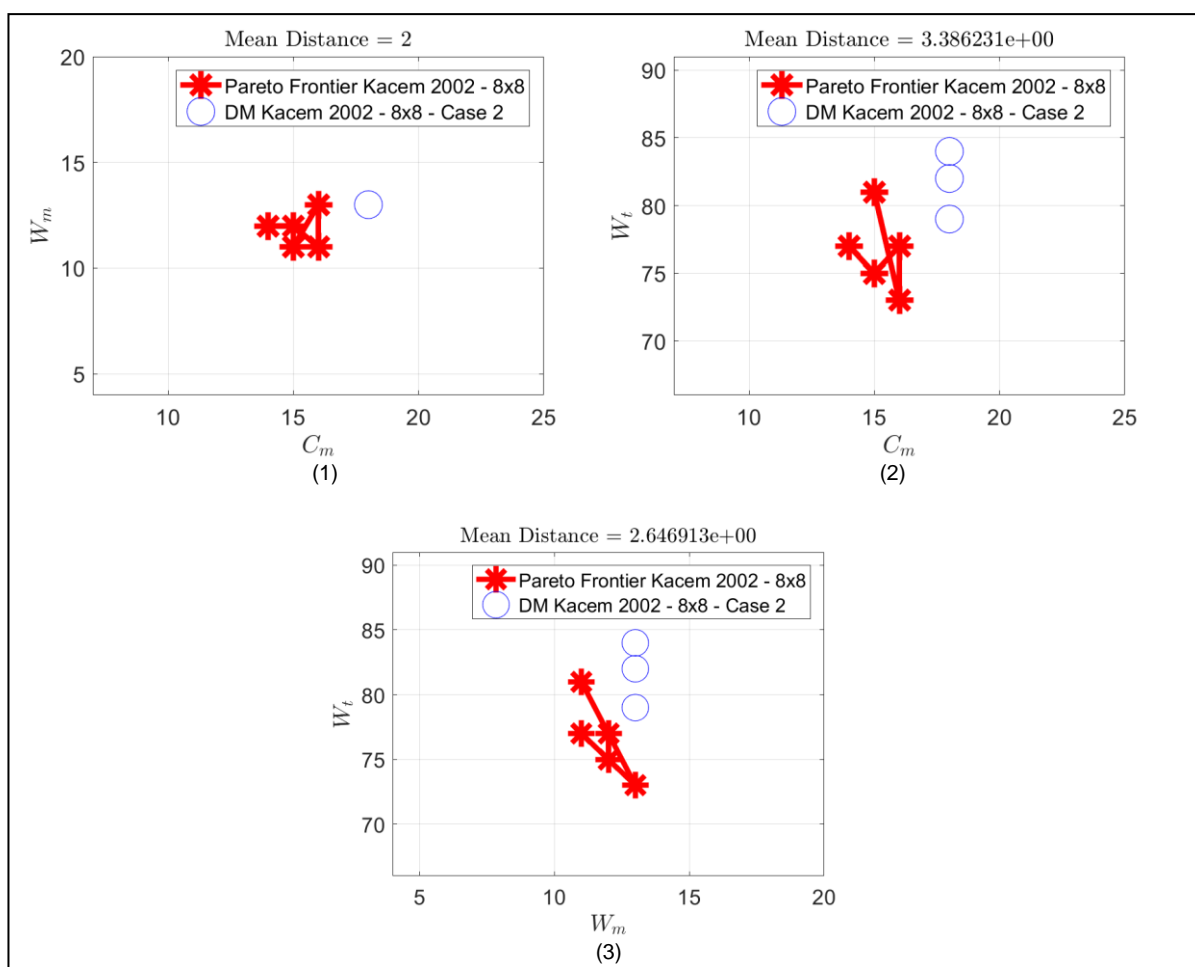
Nos experimentos relacionados à instância 8x8 caso 2, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e também para M1, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

Figura 5.5- Gráficos de Gantt da instância 8x8 caso 2 (cenário 1).



Fonte – Próprio autor.

Figura 5.6- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância 8x8 caso 2 (cenário 1).



Fonte – Próprio autor.

A ideia de manter os critérios C_m , W_t , W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção. Isto pode ser verificado na Figura 5.6, onde são notadas pequena distância média entre as soluções encontradas e as soluções na fronteira de Pareto. Na Figura 5.6 nenhuma solução selecionada pelo DM se encontra na Fronteira de Pareto, isso se deve ao peso dos critérios estabelecidos pelo usuário para se encontrar a melhor programação da produção.

É esperado que no ranqueamento do DM sejam estabelecidos cenários onde a máquina M1 possua pouco tempo ocioso entre as operações, ou seja, as operações estejam alocadas uma em seguida da outra. Analisando-se os gráficos de Gantt da Figura 5.5, observa-se que o DM busca ponderar as prioridades altas dada aos critérios C_m , W_t , W_m com a minimização do tempo ocioso interoperações.

Observa-se também que as soluções no topo do ranking possuem o mesmos valores de C_m , W_t , W_m diferindo do gráfico de Gantt da programação da produção.

5.1.4 Kacem 8x8 caso 5

Para a instância Kacem 8x8 caso 5 a solução do DM é apresentada na Tabela 5.13 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.13 – Resultado do DM para Kacem 8x8 caso 5 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB	BB	AA	BB			
1	16	82	13	8	2	4	3	4	2	2	6	4,139	0,200	0,954
2	16	81	13	7	2	0	5	5	2	4	6	4,090	0,248	0,943
3	18	82	13	8	1	4	5	4	6	2	5	4,039	0,300	0,931
4	18	79	13	1	0	0	2	7	3	5	4	4,022	0,317	0,927
5	17	81	13	7	0	0	6	5	3	5	4	4,018	0,321	0,926
6	18	79	13	1	1	0	2	7	3	5	5	4,016	0,323	0,926
7	18	84	13	1	1	0	2	7	2	4	1	4,001	0,337	0,922
8	18	84	13	1	1	0	2	7	2	4	1	4,001	0,337	0,922
9	18	84	13	1	1	0	2	7	2	4	1	4,001	0,337	0,922
10	18	87	13	8	2	0	5	4	6	2	1	3,998	0,340	0,922

Fonte – Próprio autor.

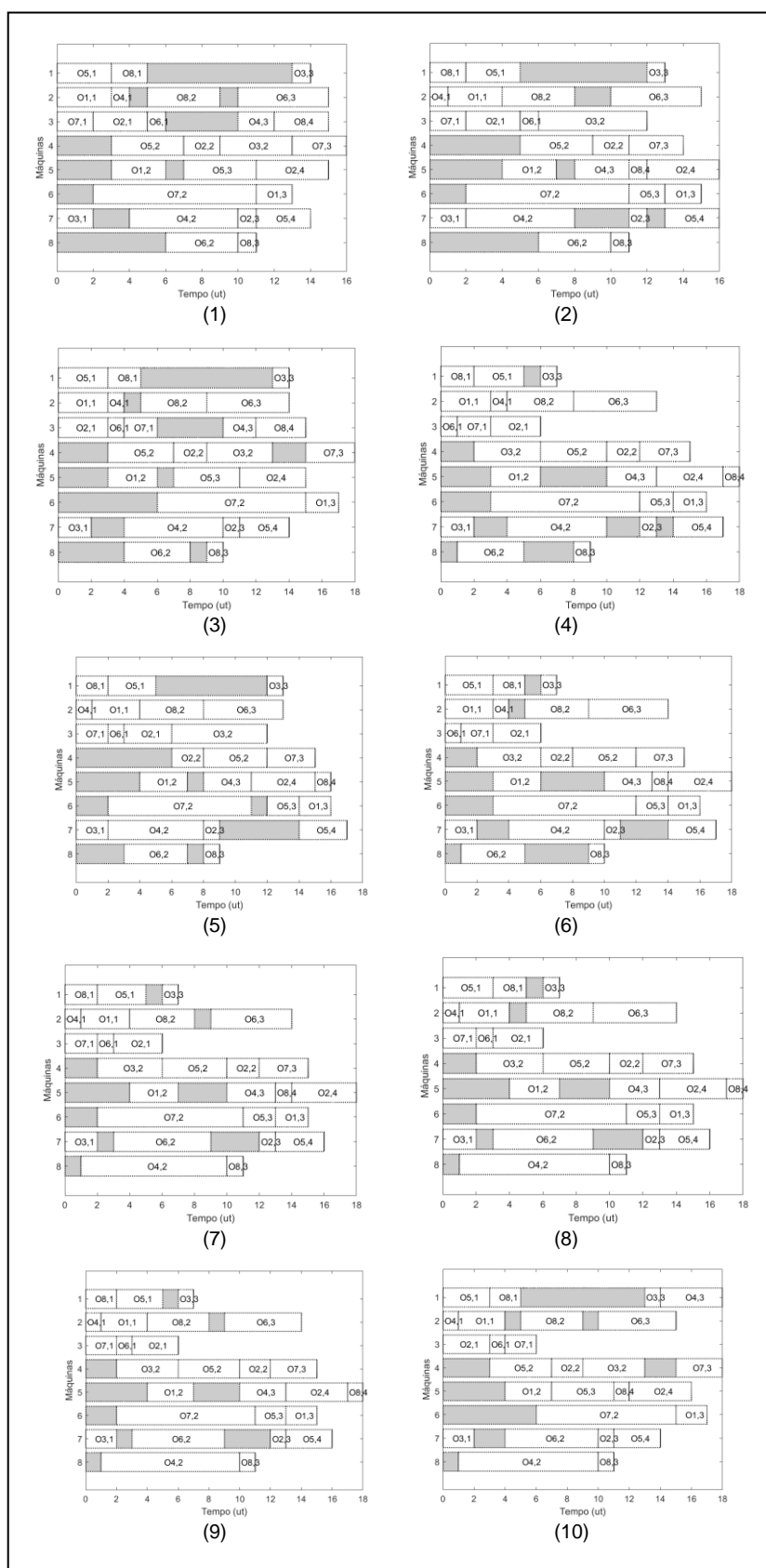
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.7.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 8x8 caso 5 são mostrados na Figura 5.8.

Nos experimentos relacionados à instância 8x8 caso 5, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e também para M7, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

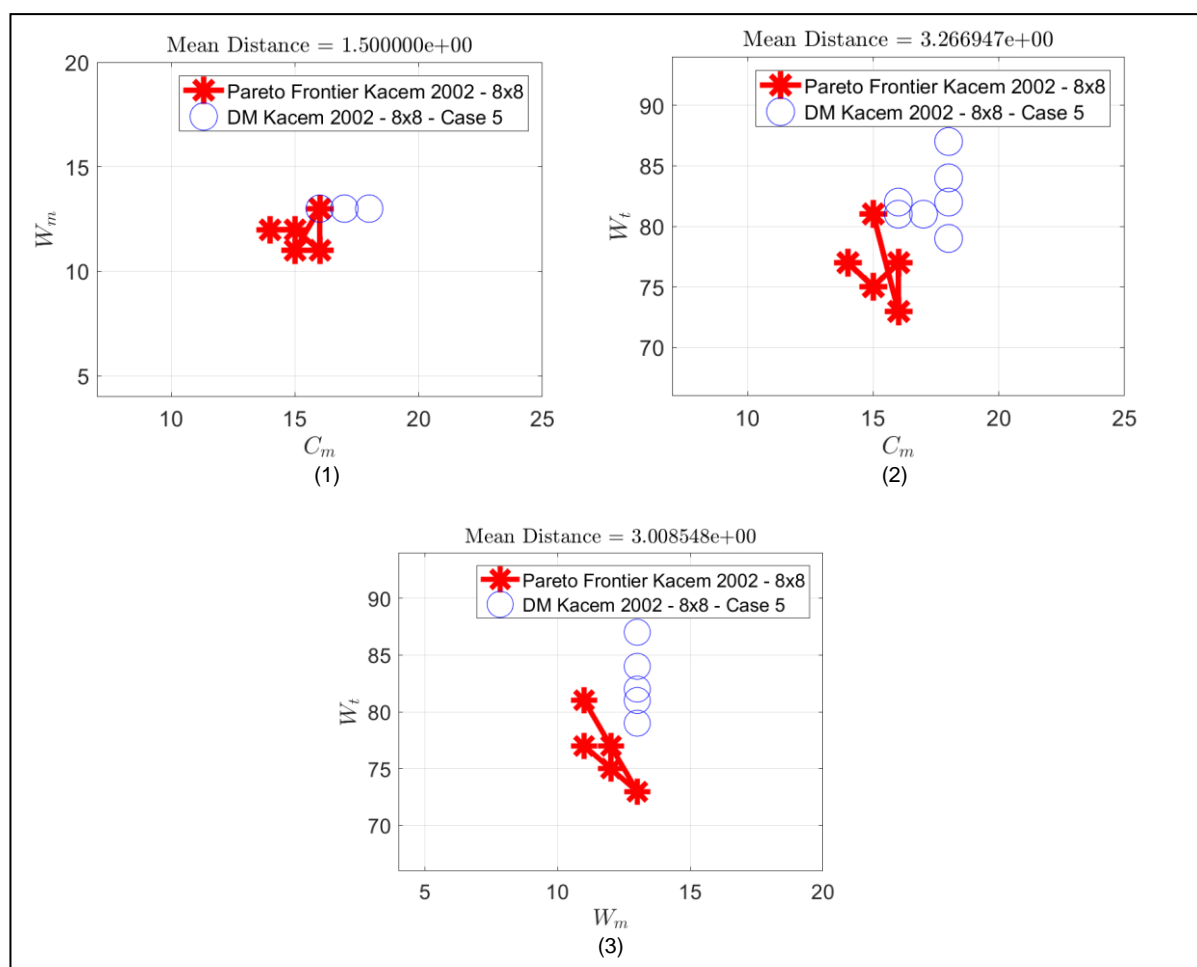
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção.

Figura 5.7- Gráficos de Gantt da instância 8x8 caso 5 (cenário 1).



Fonte – Próprio autor.

Figura 5.8- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância 8x8 caso 5 (cenário 1).



Fonte – Próprio autor.

É esperado que no ranqueamento do DM sejam estabelecidos cenários onde a máquina M7 possua pouco tempo ocioso entre as operações, ou seja, as operações estejam alocadas uma em seguida da outra. Pode ser observado na Figura 5.7, que o DM não encontrou uma programação da produção sem nenhum tempo ocioso interoperações, no entanto a ideia foi minimizar esses tempos. Isto pode ser constatado, observe a Tabela 5.13, pois em primeiro lugar no ranqueamento há uma solução dominada pela solução em segundo lugar. Os coeficientes de proximidades calculados, de acordo com as escolhas dos pesos dos critérios atestam esse fato.

Pode-se observar que, para as soluções encontradas existe pouca variedade de gráficos de Gantt para as primeiras posições do ranqueamento e a variação dos tempos ociosos interoperações da máquina M7 oscilam em demasia devido as altíssimas prioridades atribuídas à C_M , W_T e W_M , buscando-se não se afastar da Fronteira de Pareto, o que pode ser observado pelas baixas distâncias médias encontradas na Figura 5.8.

5.1.5 Kacem 10x10 caso 2

Para a instância Kacem 10x10 caso 2 a solução do DM é apresentada na Tabela 5.14 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.14 – Resultado do DM para Kacem 10x10 caso 2 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	SIN	SIP	CCI
	AA	AA	AA	BB	BB	AA	BB	BB	BB	BB	BB	BB	BB			
1	8	43	7	0	2	0	3	0	0	1	1	0	1	4,298	0,099	0,978
2	9	43	7	0	2	0	4	0	0	3	1	0	1	4,232	0,164	0,963
3	8	48	7	0	1	0	4	0	1	3	1	0	1	4,225	0,171	0,961
4	9	45	7	0	2	0	4	0	0	3	1	0	1	4,207	0,189	0,957
5	9	46	7	0	3	0	2	0	0	0	0	1	3	4,195	0,201	0,954
6	10	43	7	0	2	0	3	0	0	3	0	0	1	4,187	0,209	0,952
7	9	46	7	0	3	0	2	2	0	1	1	0	3	4,185	0,211	0,952
8	9	46	7	0	3	0	4	0	0	0	1	1	3	4,181	0,215	0,951
9	9	47	7	0	1	0	0	0	3	2	1	1	3	4,176	0,220	0,950
10	9	46	7	0	3	0	2	0	1	3	1	2	2	4,174	0,222	0,949

Fonte – Próprio autor.

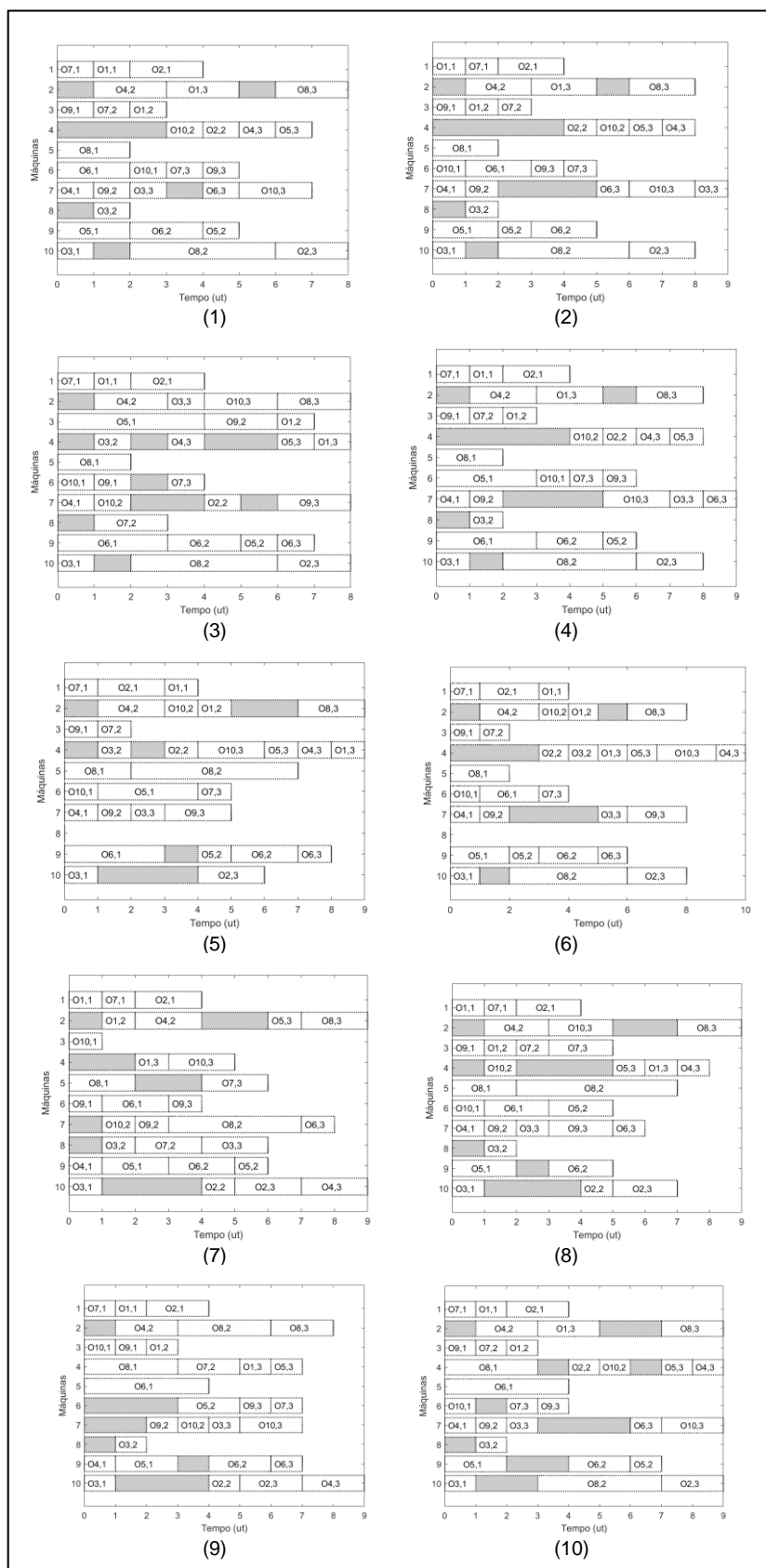
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.9.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 10x10 caso 2 são mostrados na Figura 5.10.

Nos experimentos relacionados à instância 10x10 caso 2, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e também para M3, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

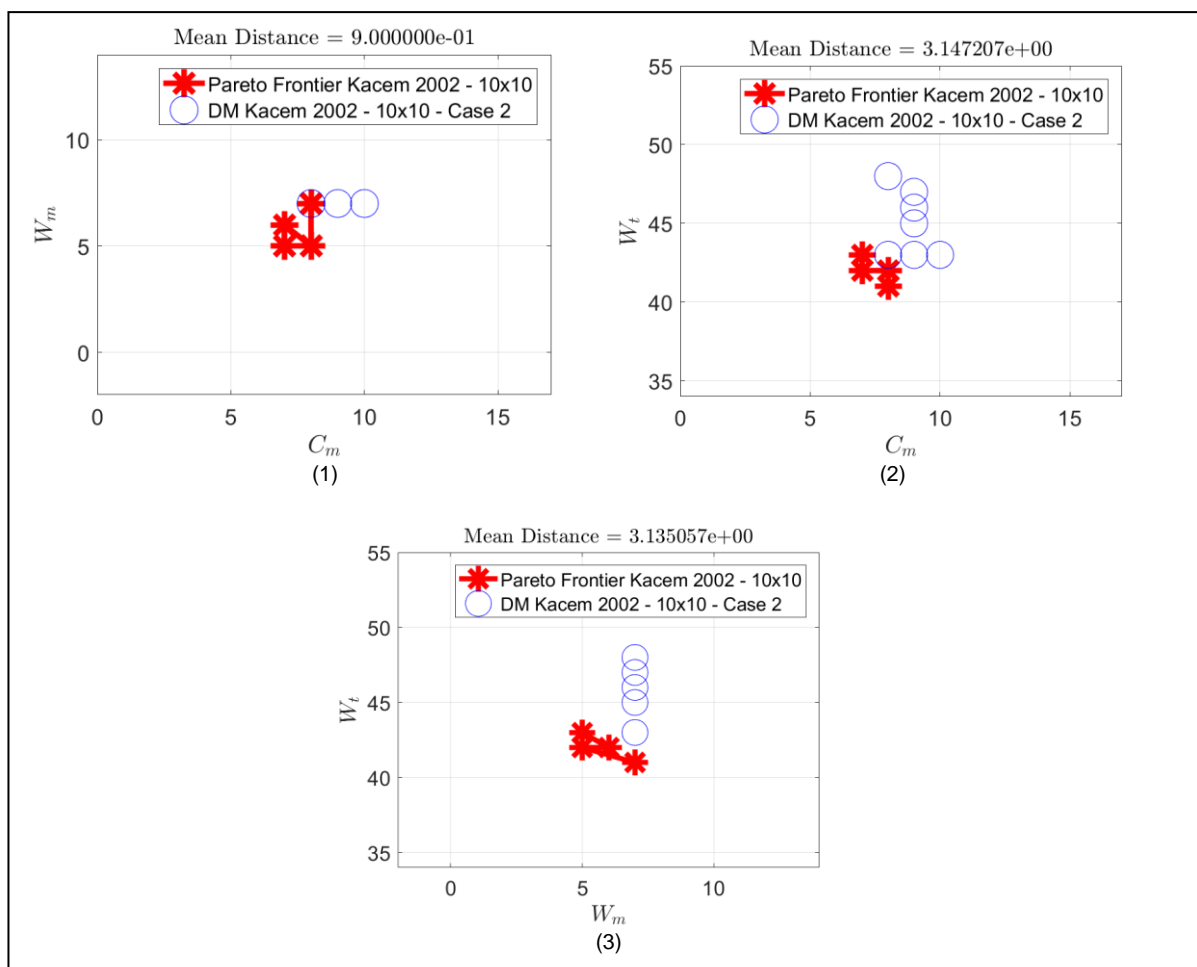
Utilizando os critérios C_m , W_t e W_m com prioridade AA espera-se que o sistema produtivo não sofra mudanças significativas no desempenho da produção. Também se é esperado que o DM selecione cenários onde a máquina M3 possua pouco tempo ocioso entre as operações, ou seja, as operações estejam alocadas uma em seguida da outra. Isto pode ser observado na Figura 5.9, onde em todos os gráficos de Gantt escolhidos pelo DM, não há intervalos de tempo ocioso entre as operações alocadas à máquina.

Figura 5.9- Gráficos de Gantt da instância 10x10 caso 2 (cenário 1).



Fonte – Próprio autor.

Figura 5.10- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância 10x10 caso 2 (cenário 1).



Fonte – Próprio autor.

Pode-se observar na Figura 5.14 o ranqueamento de soluções como os mesmos C_m , W_t e W_m nas últimas posições. Mostrando que uma diversidade maior programações da produção (com os mesmos C_m , W_t e W_m , verificada através dos gráficos de Gantt) são encontradas mais longe da fronteira de Pareto, como era de se esperar.

Na Figura 5.10 observa-se que algumas soluções encontradas pelo DM estão muito próximas ou sobrepondo-se à Fronteira de Pareto (valores não dominados) e possuem uma distância média relativamente baixa, confirmando-se o que era esperado para esta maneira de escolha dos pesos dos critérios de seleção dos cenários.

5.1.6 Kacem 10x10 caso 4

Para a instância Kacem 10x10 caso 4 a solução do DM é apresentada na Tabela 5.15 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.15 – Resultado do DM para Kacem 10x10 caso 4 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	SIN	SIP	CCI
	AA	AA	AA	BB	BB	M	BB	BB	BB	BB	BB	BB	BB			
1	8	43	7	0	2	0	3	0	0	1	1	0	1	3,840	0,099	0,975
2	8	42	7	0	2	2	3	0	0	1	0	0	1	3,777	0,161	0,959
3	9	43	7	0	2	0	4	0	0	3	1	0	1	3,775	0,164	0,958
4	8	48	7	0	1	0	4	0	1	3	1	0	1	3,768	0,171	0,957
5	8	43	7	0	1	2	3	0	0	1	1	0	3	3,752	0,186	0,953
6	9	45	7	0	2	0	4	0	0	3	1	0	1	3,750	0,189	0,952
7	9	46	7	0	3	0	2	0	0	0	0	1	3	3,738	0,201	0,949
8	10	43	7	0	2	0	3	0	0	3	0	0	1	3,730	0,209	0,947
9	9	46	7	0	3	0	2	2	0	1	1	0	3	3,728	0,211	0,946
10	9	46	7	0	3	0	4	0	0	0	1	1	3	3,724	0,215	0,945

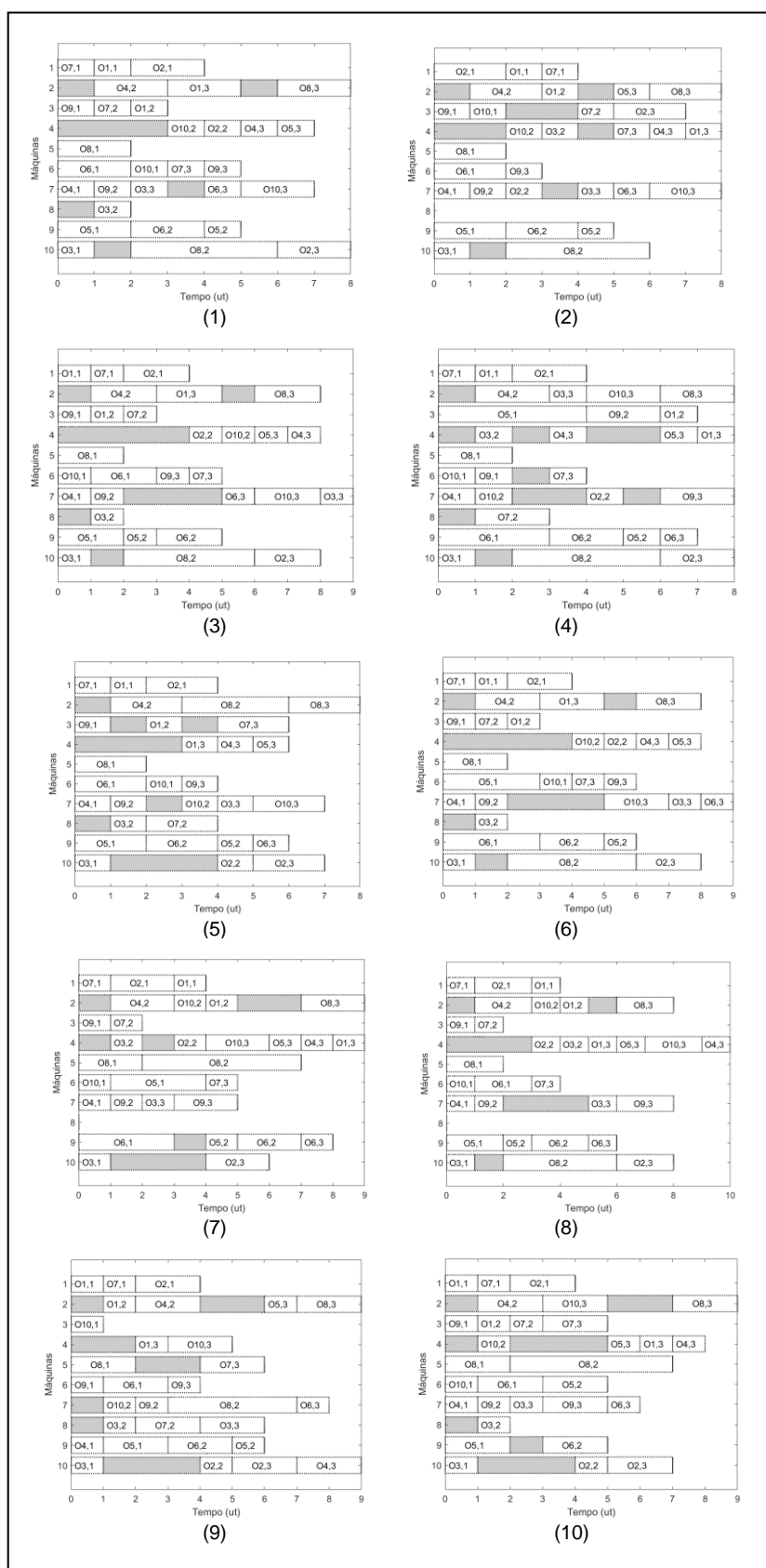
Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.11.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Kacem 10x10 caso 4 são mostrados na Figura 5.12.

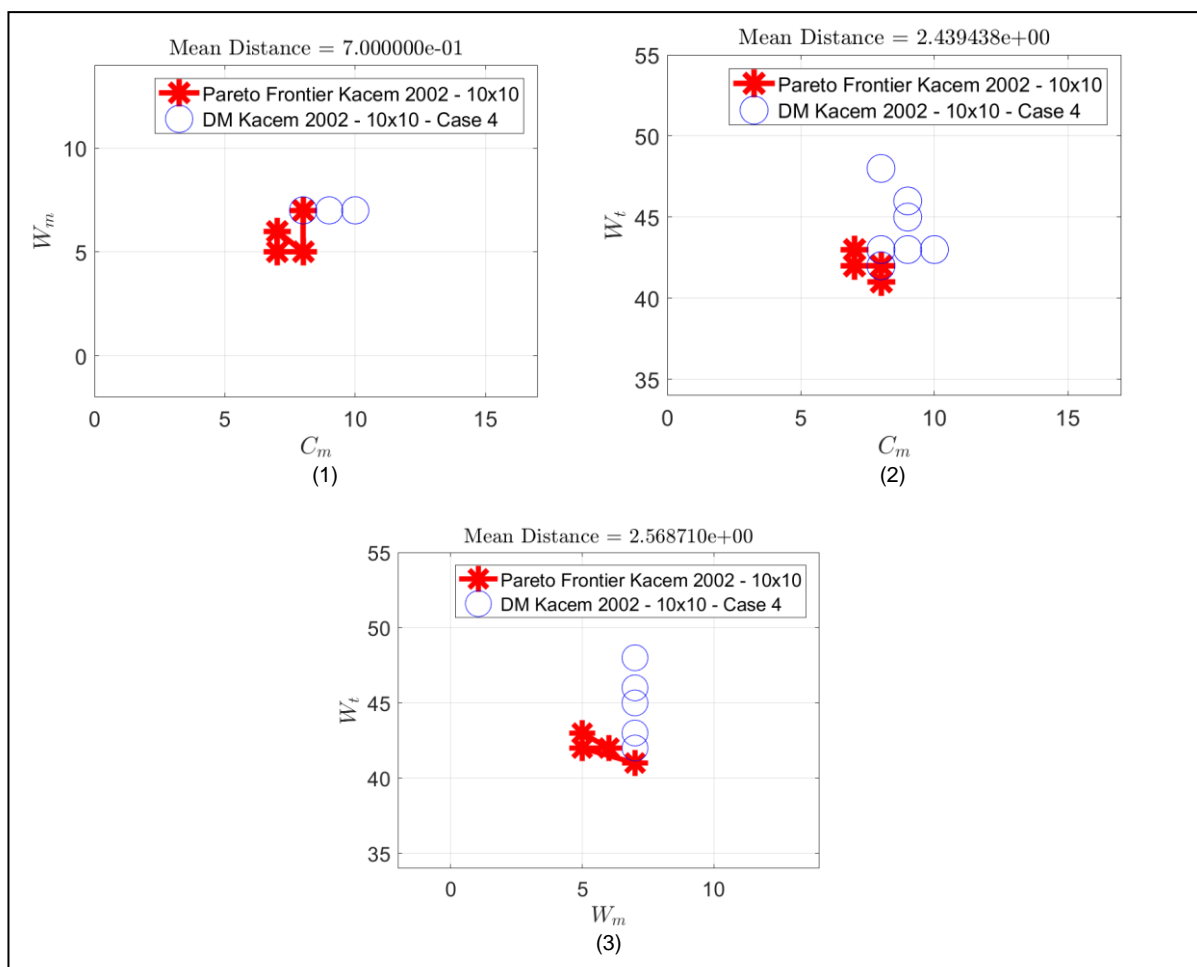
Nos experimentos relacionados à instância 10x10 caso 4, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e para a máquina M3 foi aplicada a prioridade média (M), sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

Figura 5.11- Gráficos de Gantt da instância 10x10 caso 4 (cenário 1).



Fonte – Próprio autor.

Figura 5.12- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância 10x10 caso 4 (cenário 1).



Fonte – Próprio autor.

A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção e verificar a influência da prioridade média (M) nesta instância.

Na Tabela 5.15 o destaque é para os tempos ociosos da máquina M3 para os cenários ranqueados em 2º e 5º lugares, ou seja, apesar do peso médio do critério (M) foram escolhidos cenários com tempos ociosos para essa condição, devido a necessidade das soluções não se distanciarem da Fronteira de Pareto.

Como pode-se observar, na Figura 5.11 a distância média das soluções encontradas é menor em comparação com as distâncias médias das Figura 5.10, confirmando-se o que era esperado para esta maneira de escolha dos pesos dos critérios de seleção dos cenários.

5.1.7 Bradimarte MK2 caso 1

Para a instância Bradimarte MK2 caso 1 a solução do DM é apresentada na Tabela 5.16 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.16 – Resultado do DM para Bradimarte MK2 caso 1 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB	BB			
1	28	146	27	1	0	1	12	1	1	2,731	0,566	0,828
2	29	146	27	1	0	0	12	1	2	2,674	0,623	0,811
3	29	146	27	1	0	1	12	1	2	2,670	0,627	0,810
4	30	145	27	2	1	0	13	1	3	2,667	0,630	0,809
5	29	146	27	1	0	1	16	1	2	2,661	0,636	0,807
6	29	146	27	1	1	2	12	1	2	2,661	0,636	0,807
7	29	146	27	1	1	2	12	1	2	2,661	0,636	0,807
8	29	146	27	1	0	1	15	1	3	2,660	0,637	0,807
9	29	146	27	2	1	0	13	2	2	2,658	0,639	0,806
10	29	146	27	2	0	1	16	1	2	2,658	0,639	0,806

Fonte – Próprio autor.

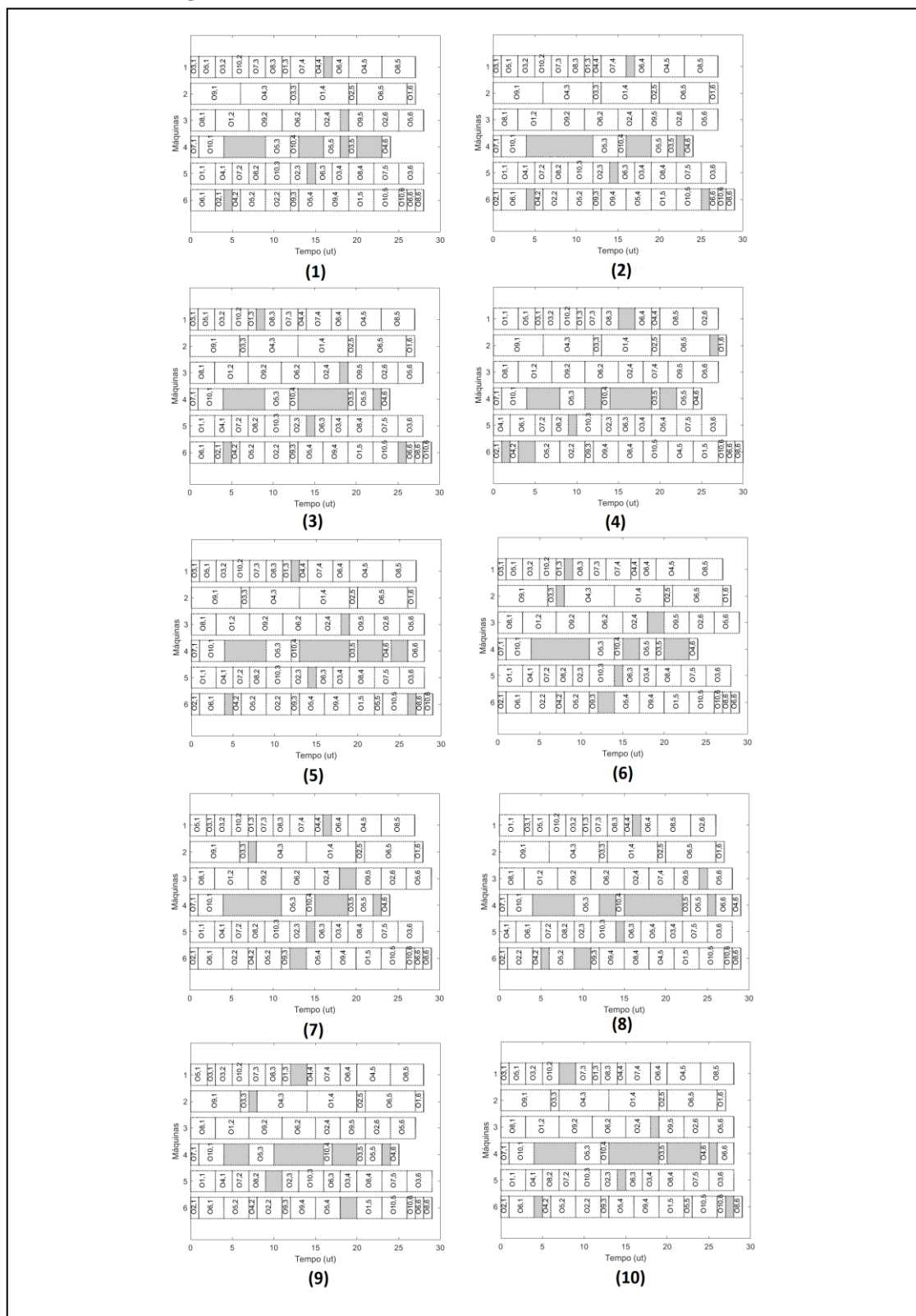
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.13.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK2 caso 1 são mostrados na Figura 5.14.

Nos experimentos relacionados à instância MK2 caso 1, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t e W_m , sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

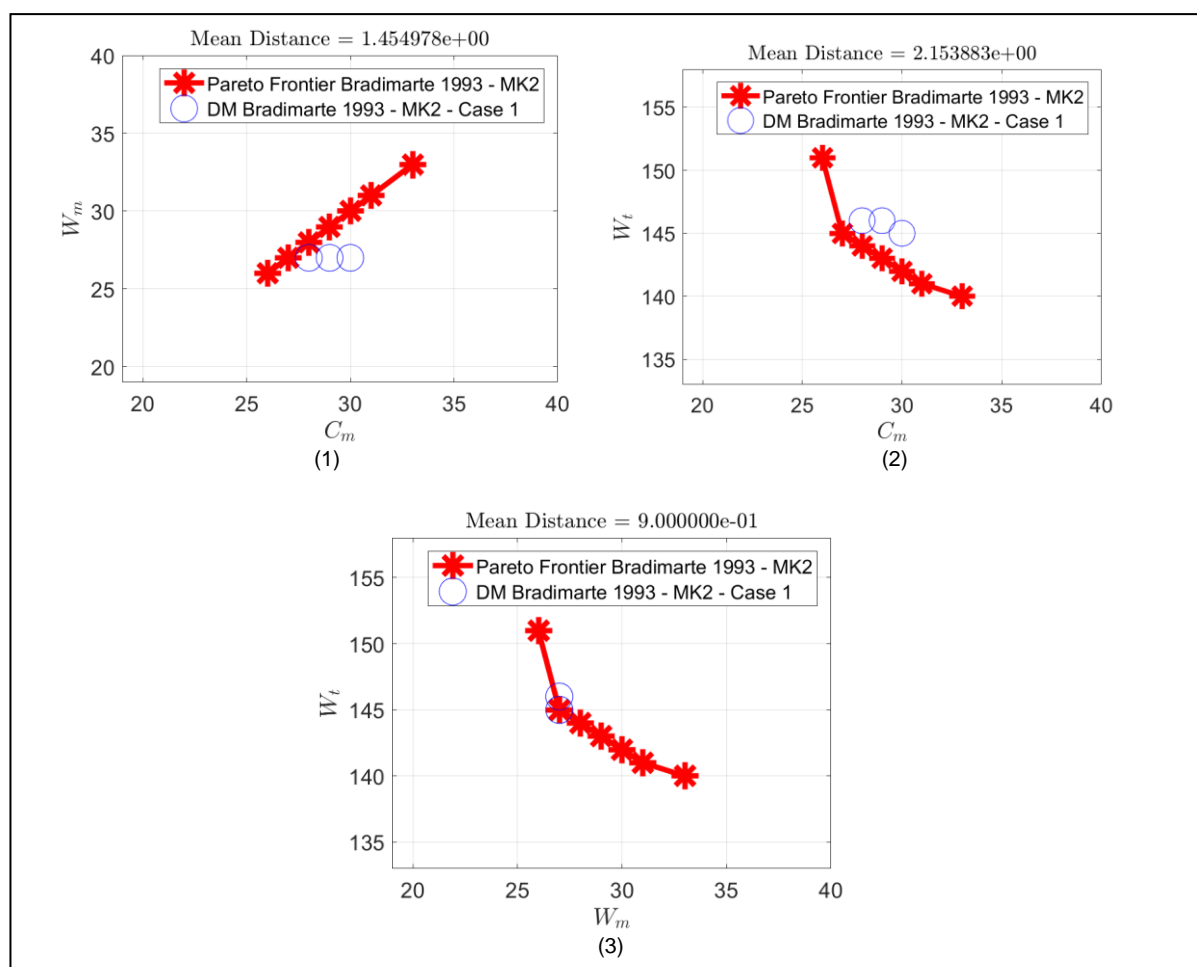
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção. E para as máquinas atribuídas com prioridade BB a ideia é de que os resultados fornecidos pelo DM se aproximem o máximo possível da Fronteira de Pareto.

Figura 5.13- Gráficos de Gantt da instância MK2 caso 1 (cenário 1).



Fonte – Próprio autor.

Figura 5.14- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK2 caso 1 (cenário 1).



Fonte – Próprio autor.

O ranqueamento das soluções do DM deve estabelecer resultados semelhantes aos resultados otimizados pelo algoritmo FJSP-ABC, isto é, de acordo com a eficiência do algoritmo FJSP-ABC conseguir otimizar os parâmetros C_m , W_t e W_m , menor será a distância média das soluções do DM para a Fronteira de Pareto (Figura 5.14).

Soluções eficientes do FJSP-ABC são aquelas que conseguem atingir os resultados não dominados da Fronteira de Pareto com uma diversidade grande de representações da programação pelos gráficos de Gantt. Na Tabela 5.16 foram ranqueadas 3 soluções diferentes de C_m , W_t e W_m , o que se é buscado otimizar no algoritmo FJSP-ABC, sendo que uma delas (29,146,27) possui oito representações da programação da produção por gráficos de Gantt diferentes, como pode ser visto em destaque na Tabela 5.16 e na Figura 5.13.

5.1.8 Bradimarte MK2 caso 2

Para a instância Bradimarte MK2 caso 2 a solução do DM é apresentada na Tabela 5.17 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.17 – Resultado do DM para Bradimarte MK2 caso 2 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	SIN	SIP	CCI
	BB	BB	BB	AA	AA	AA	AA	AA	AA			
1	28	149	28	0	0	2	6	0	0	5,823	0,252	0,959
2	28	149	28	1	0	0	7	0	1	5,799	0,276	0,955
3	27	151	27	0	0	0	6	2	0	5,791	0,284	0,953
4	28	149	28	1	0	0	8	0	1	5,762	0,313	0,949
5	28	149	28	1	0	2	6	0	0	5,762	0,313	0,948
6	28	149	28	1	0	2	6	0	0	5,762	0,313	0,948
7	28	149	28	0	0	0	10	0	1	5,751	0,324	0,947
8	28	150	28	0	0	0	10	0	1	5,747	0,328	0,946
9	28	150	28	0	0	0	10	0	1	5,747	0,328	0,946
10	28	150	28	0	0	0	10	0	1	5,747	0,328	0,946

Fonte – Próprio autor.

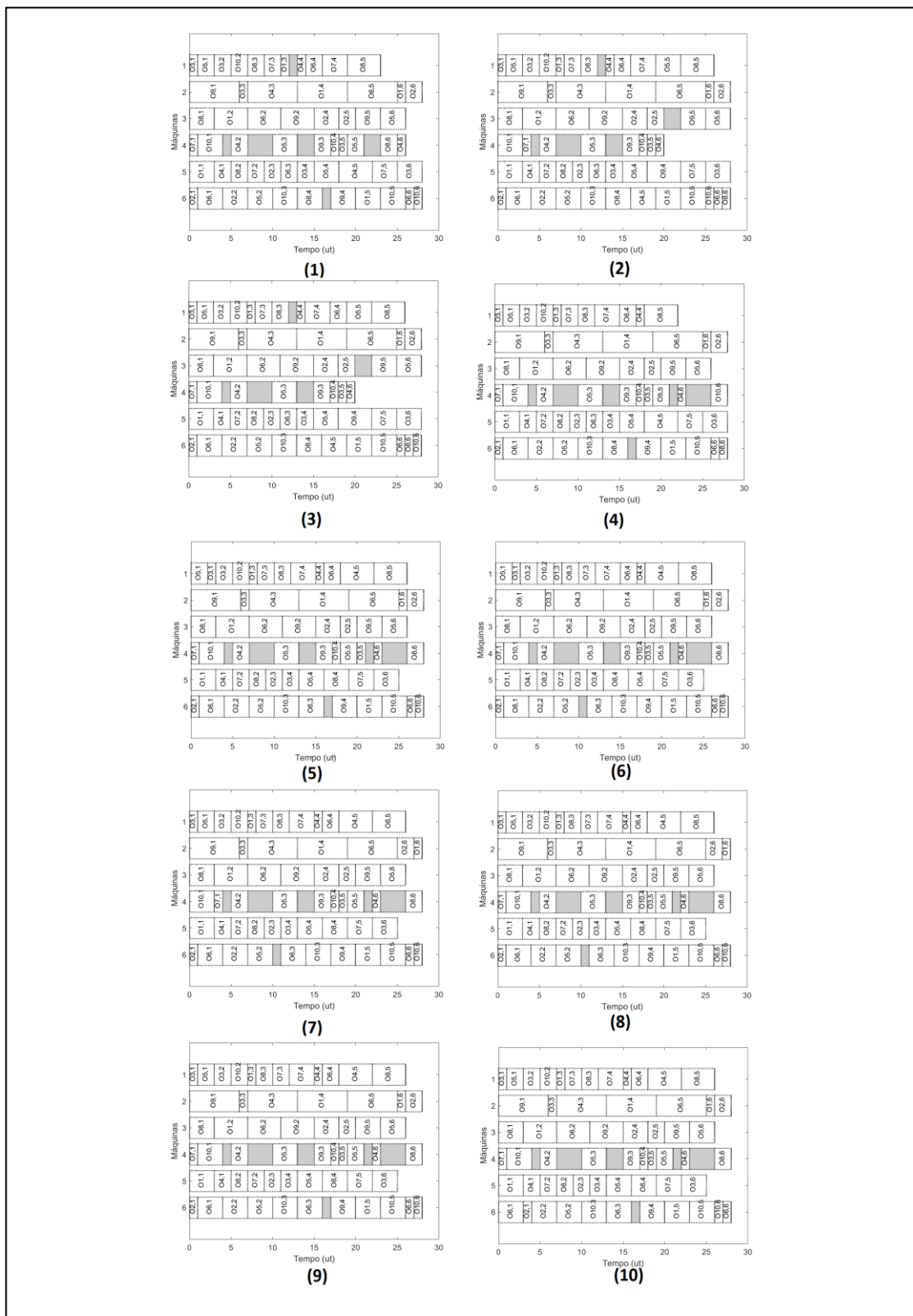
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.15.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para Bradimarte MK2 caso 2 são mostrados na Figura 5.16.

Nos experimentos relacionados à instância MK2 caso 2, foram colocados baixíssima prioridade (BB) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e altíssima prioridade (AA) para o menor tempo ocioso entre as máquinas em todas as máquinas.

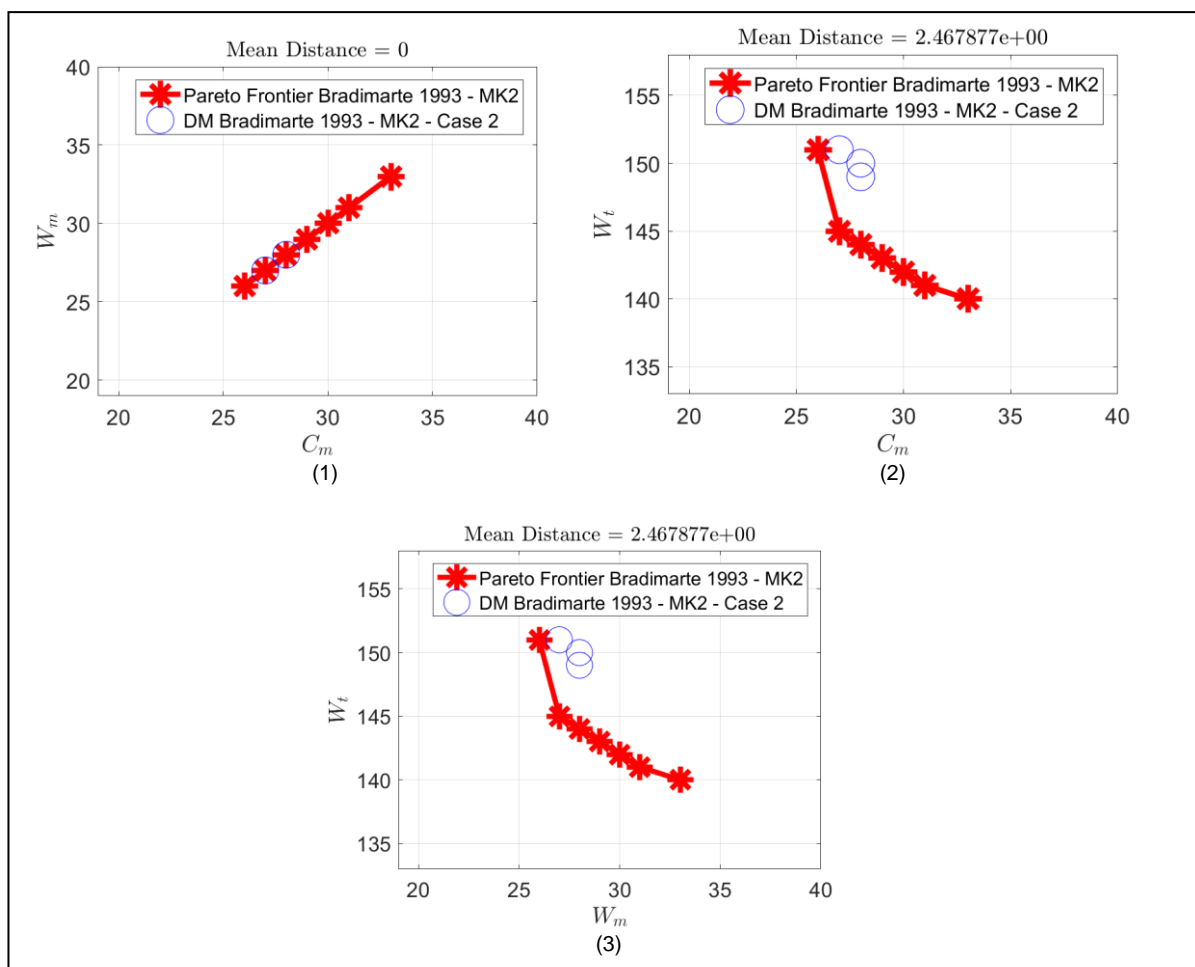
A ideia de verificar soluções boas ranqueadas para o DM sem se afastar em demasia das soluções da Fronteira de Pareto, buscando não modificar muito o desempenho da produção.

Figura 5.15- Gráficos de Gantt da instância MK2 caso 2 (cenário 1).



Fonte – Próprio autor.

Figura 5.16- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ instância MK2 caso 2 (cenário 1).



Fonte – Próprio autor.

Verificou-se que embora selecionado prioridade baixíssima para C_m , W_t e W_m as soluções apresentadas pelo DM se encontram relativamente perto da Fronteira de Pareto, mostrando-se mais distante em relação a W_t , conforme Figura 5.16. Ou seja, verificou-se um distanciamento da Fronteira de Pareto maior em relação a C_m e W_t sem que o critério W_m se afaste da Fronteira de Pareto.

Pode-se observar novamente que, com a proximidade das soluções perto da Fronteira de Pareto, menor é a diversidade de programação da produção representada por gráficos de Gantt, o que faz com que os resultados se distanciem das soluções não dominadas.

Vale ressaltar que visualmente na Figura 5.15 as soluções encontradas pelo DM possuem, para essa instância, pouco ou nenhum tempo ocioso entre as operações.

5.1.9 Bradimarte MK4 caso 3

Para a instância Bradimarte MK4 caso 3 a solução do DM é apresentada na Tabela 5.17 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.17 – Resultado do DM para Bradimarte MK4 caso 3 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP	CCI
	A	AA	BB	A	M	AA	M	B	MB	AA	BB			
1	78	348	75	0	46	6	23	30	47	7	25	5,374	1,020	0,841
2	78	348	75	0	46	6	24	30	47	7	25	5,369	1,025	0,840
3	77	349	75	0	40	7	24	27	48	7	25	5,361	1,033	0,838
4	77	349	75	0	46	7	19	33	48	7	24	5,359	1,035	0,838
5	76	350	75	0	41	8	19	31	46	7	24	5,355	1,039	0,838
6	77	349	75	0	42	7	24	29	47	7	24	5,354	1,039	0,837
7	77	349	75	0	43	7	23	30	47	7	25	5,354	1,039	0,837
8	77	349	75	0	43	7	23	30	47	7	25	5,354	1,039	0,837
9	77	349	75	0	43	7	23	30	47	7	25	5,354	1,039	0,837
10	77	349	75	0	43	7	23	30	47	7	25	5,354	1,039	0,837

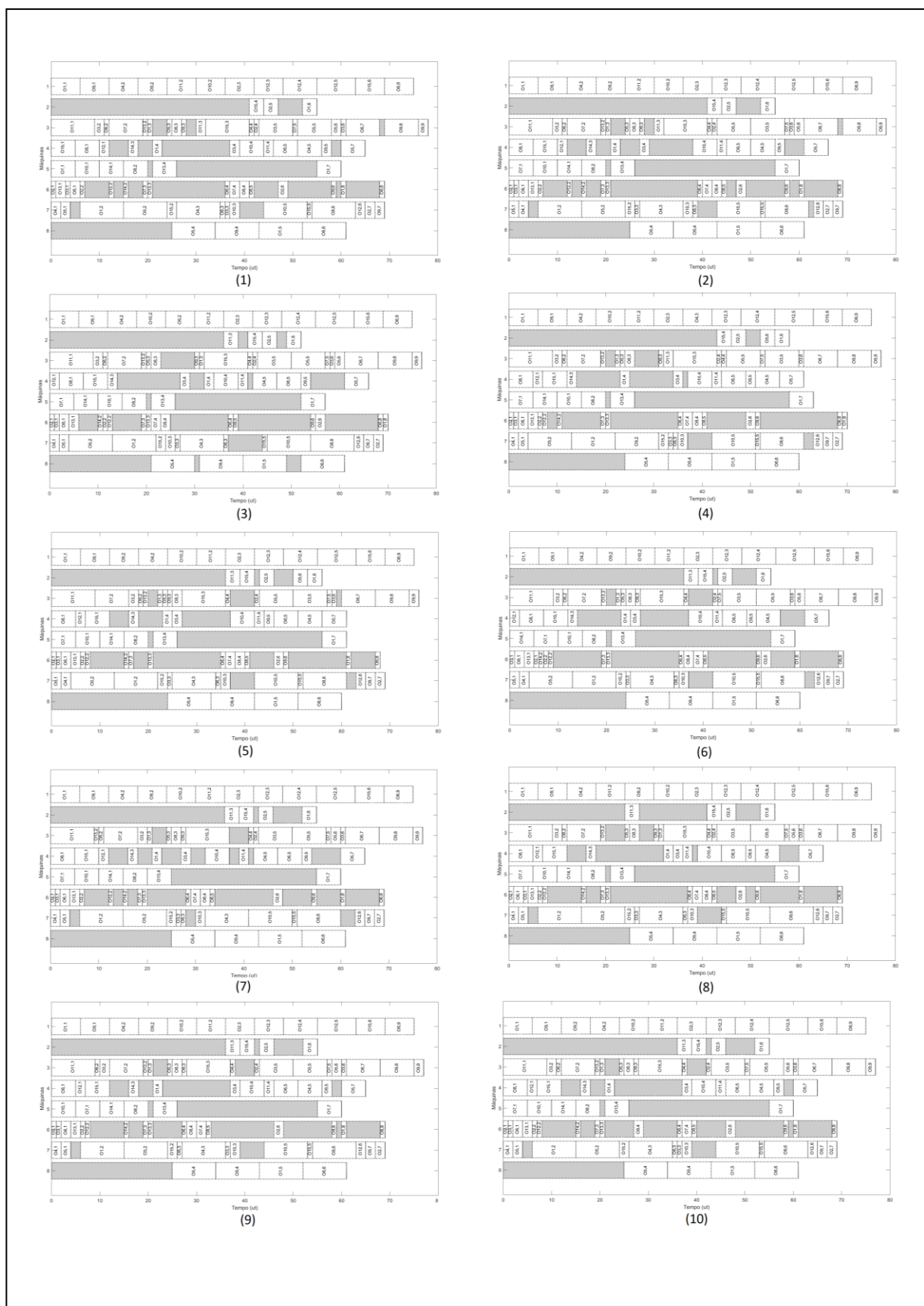
Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.17.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK4 caso 3 são mostrados na Figura 5.18.

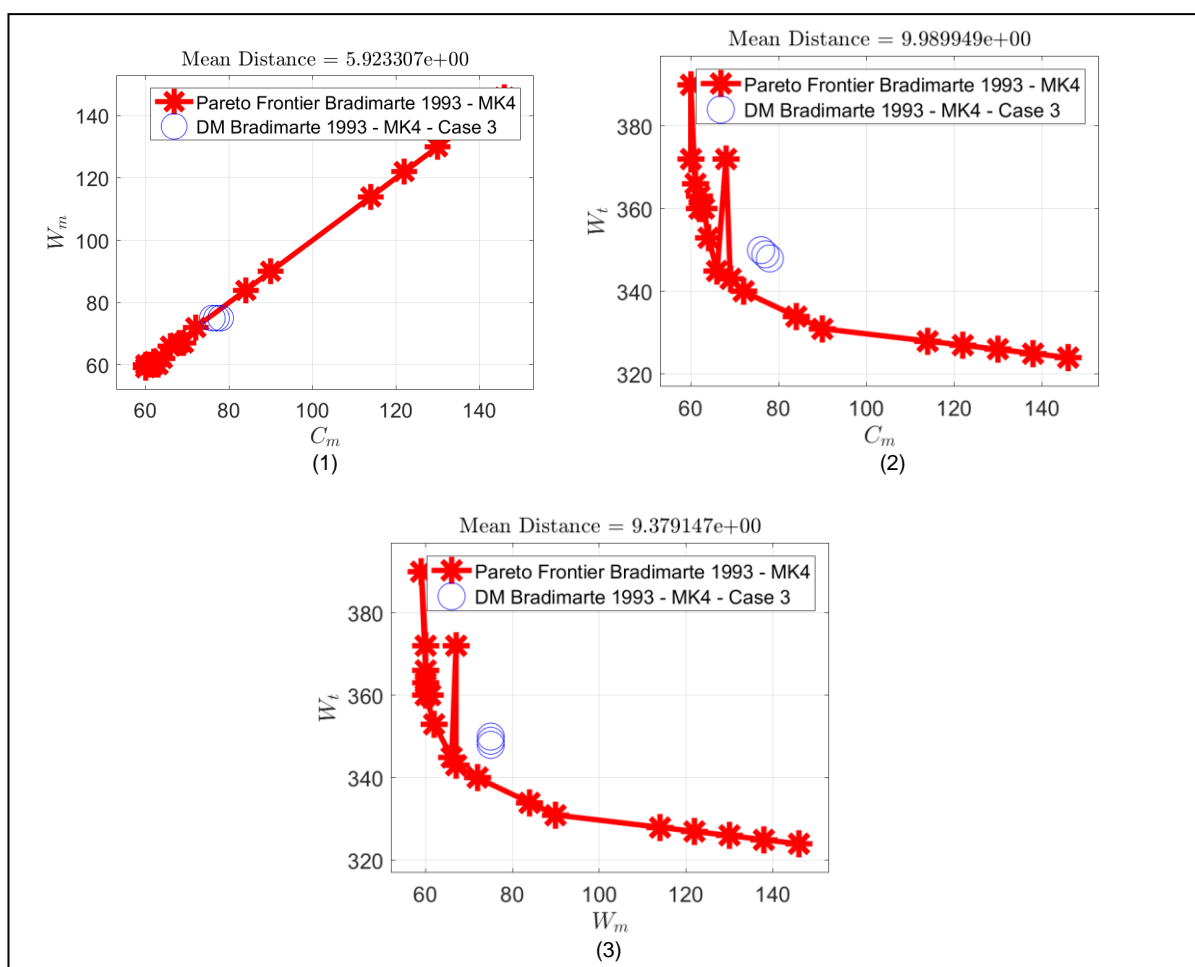
Nos experimentos relacionados à instância MK4 caso 3, foram colocados uma miscelânea de prioridades aos critérios do tomador de decisão (DM) a ideia é verificar um ajuste mais fino pelo usuário, de acordo com suas necessidades apontadas no chão de fábrica.

Figura 5.17- Gráficos de Gantt da instância MK4 caso 3 (cenário 1).



Fonte – Próprio autor.

Figura 5.18- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK4 caso 3 (cenário 1).



Fonte – Próprio autor.

Como pode ser observado nas soluções apresentadas na Tabela 5.17 buscou-se manter um equilíbrio para os critérios C_m , W_t e W_m (A, AA e BB, respectivamente) e selecionar algumas máquinas com prioridade altíssima (AA) para minimizar os tempos ociosos entre suas operações. As distâncias médias das soluções encontradas pelo DM, ilustradas na Figura 5.18 mostram valores na casa de unidades, que proporcionalmente às magnitudes de cada critério podem ser consideradas relativamente pequenas.

Assim, se é possível utilizar soluções balanceadas para os critérios C_m , W_t e W_m dentro as soluções fornecidas pelo arquivo de Pareto do algoritmo FJSP-ABC.

5.1.10 Bradimarte MK4 caso 4

Para a instância Bradimarte MK4 caso 4 a solução do DM é apresentada na Tabela 5.18 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.18 – Resultado do DM para Bradimarte MK4 caso 4 (cenário 1).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP
	MA	M	MA	BB	AA	M	A	M	AA	MA	AA		
1	78	349	75	0	41	17	19	21	47	16	13	6,451	1,046
2	78	349	75	0	43	17	19	21	47	16	13	6,439	1,058
3	76	349	75	0	45	15	20	29	46	7	15	6,434	1,063
4	77	352	75	0	45	25	19	29	47	8	6	6,419	1,078
5	78	349	75	0	41	16	21	21	48	16	15	6,409	1,088
6	78	349	75	0	45	17	20	29	47	7	15	6,405	1,092
7	78	349	75	0	45	17	20	29	47	7	15	6,405	1,092
8	78	350	75	1	42	19	20	29	48	7	15	6,388	1,109
9	78	349	75	1	42	17	17	36	48	7	17	6,386	1,111
10	76	350	75	0	41	8	19	31	46	7	24	6,385	1,112

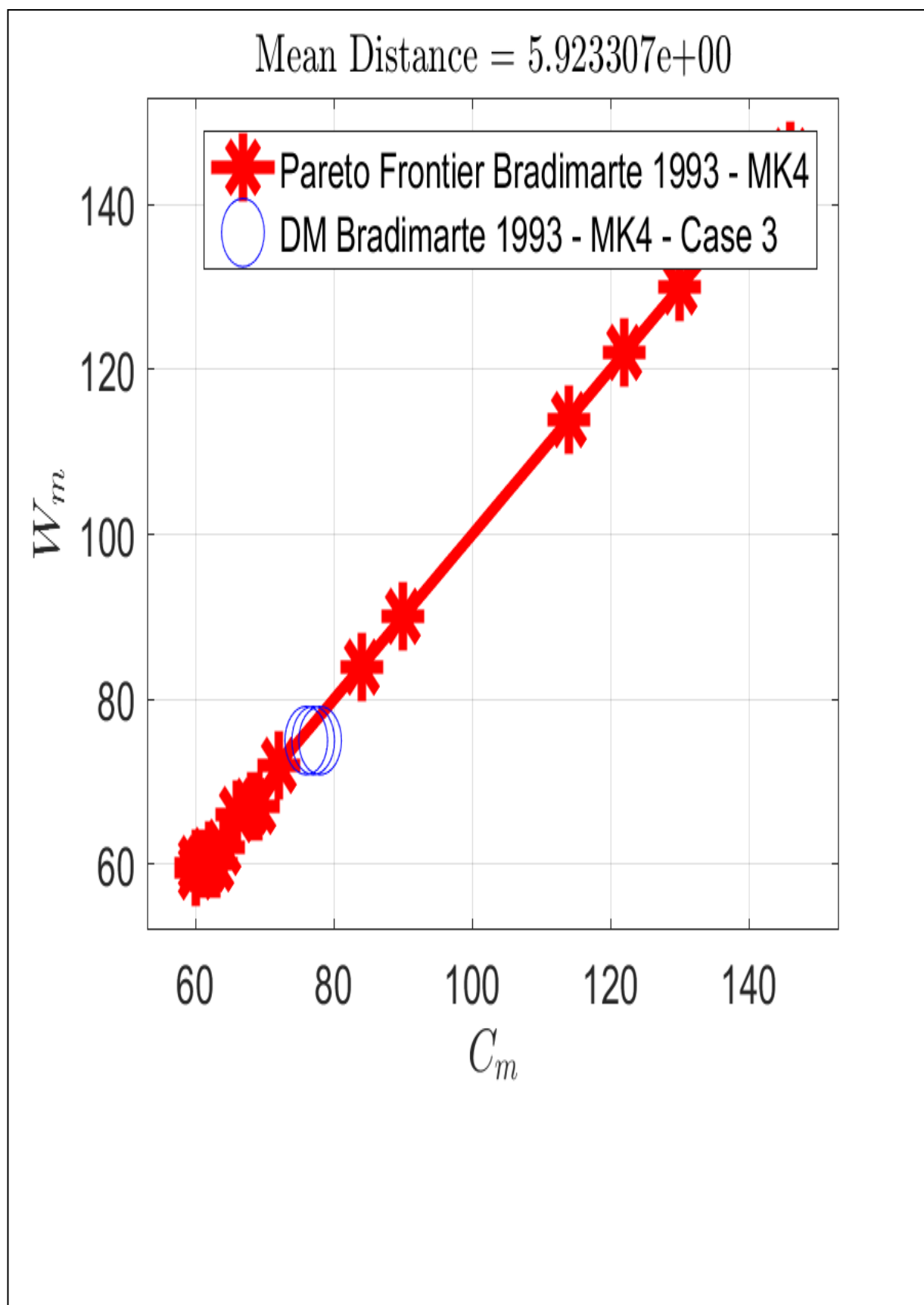
Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.19.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK4 caso 4 são mostrados na Figura 5.20.

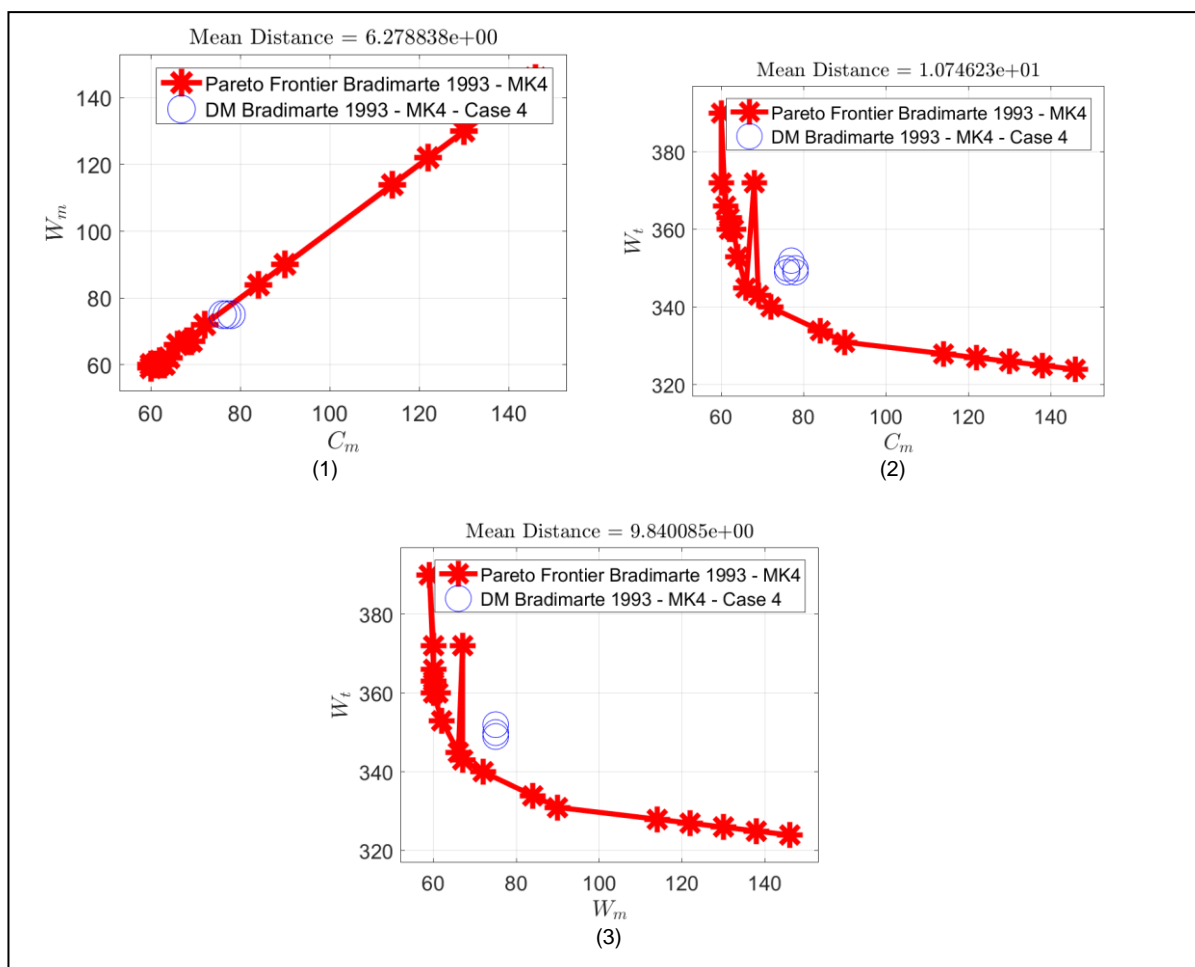
Nos experimentos relacionados à instância MK4 caso 4, semelhante ao MK4 caso 3 (item 5.1.9) foram colocados uma miscelânea de prioridades aos critérios do tomador de decisão (DM) para se verificar outra situação de um ajuste mais fino pelo usuário, de acordo com suas necessidades apontadas no chão de fábrica.

Figura 5.19- Gráficos de Gantt da instância MK4 caso 4 (cenário 1).



Fonte – Próprio autor.

Figura 5.20- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK4 caso 4 (cenário 1).



Fonte – Próprio autor.

Da mesma forma que o item 5.1.9, pode ser observado nas soluções apresentadas na Tabela 5.18 que os critérios C_m , W_t e W_m (MA, M e MA, respectivamente) podem ser selecionados para buscar soluções de acordo com situações diferentes das condições de otimização direta destes parâmetros, fornecendo alternativas de programação da produção ao usuário de acordo com os critérios estabelecidos para as máquinas.

Embora essas soluções tendam a se afastar da Fronteira de Pareto, as soluções encontradas pelo DM podem atender as restrições estabelecidas para o sistema produtivo de forma satisfatória.

A Figura 5.20 ilustra as distâncias médias das soluções, as quais são semelhantes as distâncias médias encontradas para a situação anterior (item 5.1.9).

5.2 Cenário 2: Prioridade para maior ociosidade entre operações

Para o cenário com prioridade para a máquina com maior ociosidade entre operações, o usuário pode estabelecer os critérios de decisão para buscar as programações da produção onde determinada máquina pode ter a sua carga de trabalho minimizada.

O algoritmo do DM foi executado para seis casos diferentes, onde os pesos de cada critério foram alterados de forma a ilustrar uma situação real possível no sistema de fabricação. Os resultados utilizam as instâncias de Bradimarte (1993) nos seis casos estudados, conforme ilustrado nas tabelas 5.19, 5.20 e 5.21.

As análises foram realizadas para todos os cenários e casos descritos nas tabelas 5.19, 5.20, e 5.21 e nas seções seguintes deste documento serão apresentados os casos em destaque para cada cenário.

Tabela 5.19 – Pesos dos critérios para Bradimarte MK2 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	AA	BB	BB	BB	BB	BB
caso 3	AA	AA	AA	A	BB	BB	BB	BB	BB
caso 4	AA	AA	AA	M	BB	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	AA	BB	BB
caso 6	AA	AA	AA	BB	BB	BB	A	BB	BB

Fonte – Próprio autor.

Tabela 5.20 – Pesos dos critérios para Bradimarte MK4 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	BB	AA	BB	BB	BB	BB	BB	BB
caso 3	AA	AA	AA	BB	A	BB	BB	BB	BB	BB	BB
caso 4	AA	AA	AA	BB	M	BB	BB	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	BB	BB	BB	AA	BB
caso 6	AA	AA	AA	BB	BB	BB	BB	BB	BB	A	BB

Fonte – Próprio autor.

Tabela 5.21 – Pesos dos critérios para Bradimarte MK8 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
caso 1	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	BB	BB
caso 2	AA	AA	AA	BB	BB	AA	BB	BB	BB	BB	BB	BB	BB
caso 3	AA	AA	AA	BB	BB	A	BB	BB	BB	BB	BB	BB	BB
caso 4	AA	AA	AA	BB	BB	M	BB	BB	BB	BB	BB	BB	BB
caso 5	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	AA	BB
caso 6	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	A	BB

Fonte – Próprio autor.

5.2.1 Bradimarte MK2 caso 1

Para a instância Bradimarte MK2 caso 1 a solução do DM é apresentada na Tabela 5.22 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.22 – Resultado do DM para Bradimarte MK2 caso 1 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB	BB			
1	28	146	27	1	0	1	12	1	1	2,450	0,847	0,743
2	29	146	27	2	1	2	17	1	2	2,419	0,878	0,734
3	29	146	27	2	1	2	17	1	2	2,419	0,878	0,734
4	29	146	27	1	1	2	17	1	2	2,416	0,882	0,733
5	29	146	27	1	1	2	17	1	2	2,416	0,882	0,733
6	29	146	27	2	1	2	13	2	2	2,416	0,882	0,733
7	30	145	27	2	1	0	13	1	3	2,414	0,883	0,732
8	31	145	27	6	2	3	17	4	4	2,414	0,883	0,732
9	31	145	27	6	2	3	16	4	4	2,412	0,886	0,731
10	29	146	27	2	1	2	13	1	2	2,411	0,887	0,731

Fonte – Próprio autor.

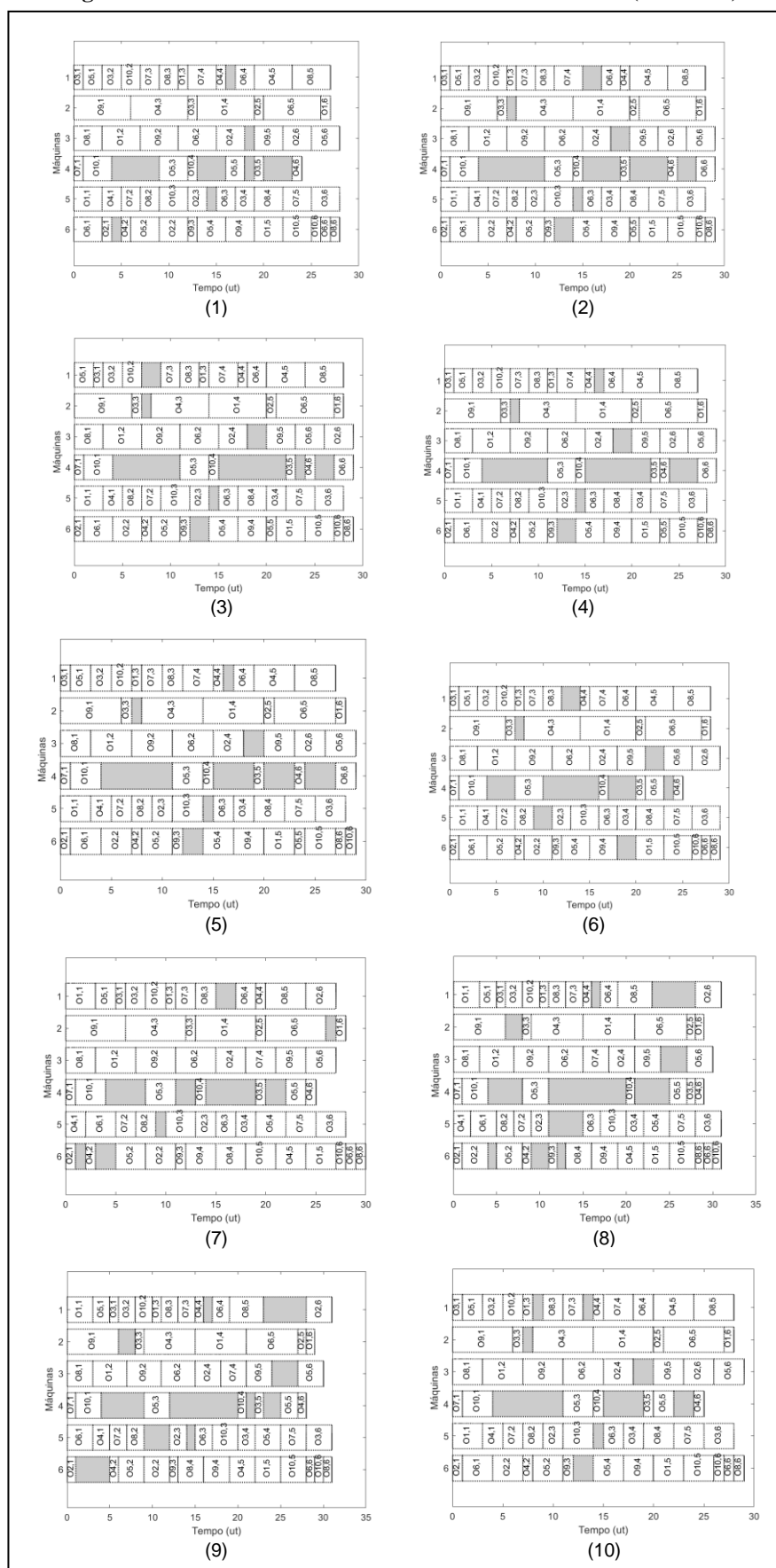
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.21.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK2 caso 1 são mostrados na Figura 5.22.

Nos experimentos relacionados à instância MK2 caso 1, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m , sendo que aos outros critérios, referente às máquinas, foram atribuídas baixíssima prioridade (BB).

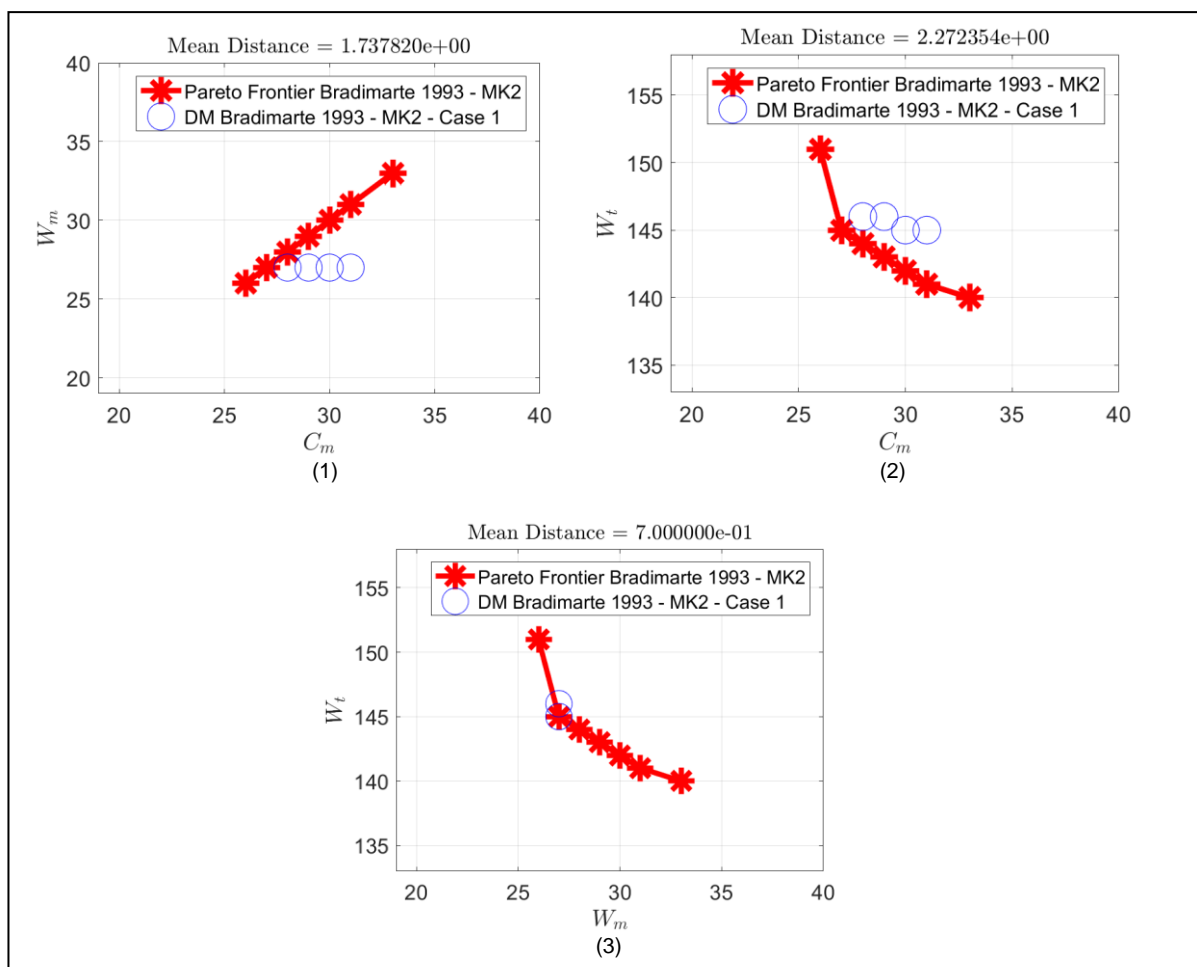
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que o DM forneça resultados o mais próximo possível da Fronteira de Pareto, ou seja, os melhores resultados da otimização realizada pelo algoritmo FJSP-ABC. E para as máquinas atribuídas com prioridade BB a ideia é de que não haja influência significativa da busca de menor ociosidade das máquinas pelo DM, deixando os resultados serem conduzidos pelos critérios C_m , W_t e W_m .

Figura 5.21- Gráficos de Gantt da instância MK2 caso 1 (cenário 2).



Fonte – Próprio autor.

Figura 5.22- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK2 caso 1 (cenário 2).



Fonte – Próprio autor.

O ranqueamento das soluções do DM deve estabelecer resultados próximos à Fronteira de Pareto e semelhantes aos melhores resultados otimizados pelo algoritmo FJSP-ABC, em concordância com os pesos dos outros critérios estabelecidos de prioridade para maior tempo de ociosidade entre as máquinas, isto é, o DM pondera um maior número maior de variáveis de otimização, fazendo com que os resultados possam não ser os mesmos.

Observa-se na Figura 5.22 que as soluções ranqueadas pelo DM possuem uma distância média pequena em relação às soluções não dominados da Fronteira de Pareto validando o funcionamento do algoritmo do DM.

5.2.2 Bradimarte MK2 caso 2

Para a instância Bradimarte MK2 caso 2 a solução do DM é apresentada nas Tabela 5.23 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.23 – Resultado do DM para Bradimarte MK2 caso 2 (cenário 2).

	m	t	m	1	2	3	4	5	6	IN	IP	CI
	AA	AA	AA	AA	BB	BB	BB	BB	BB			
1	33	143	31	15	0	5	18	0	7	2,992	1,231	0,709
2	34	142	32	16	0	4	21	1	6	2,983	1,240	0,706
3	36	143	29	14	2	3	15	8	8	2,959	1,264	0,701
4	34	143	32	16	0	5	22	5	4	2,936	1,287	0,695
5	36	143	29	12	3	6	20	8	8	2,864	1,359	0,678
6	32	143	32	13	0	1	20	2	5	2,837	1,386	0,672
7	31	144	29	9	0	2	18	3	4	2,830	1,393	0,670
8	32	144	30	11	0	3	20	2	4	2,817	1,407	0,667
9	31	144	29	9	0	0	17	2	4	2,815	1,408	0,667
10	31	144	29	9	0	0	13	3	4	2,812	1,411	0,666

Fonte – Próprio autor.

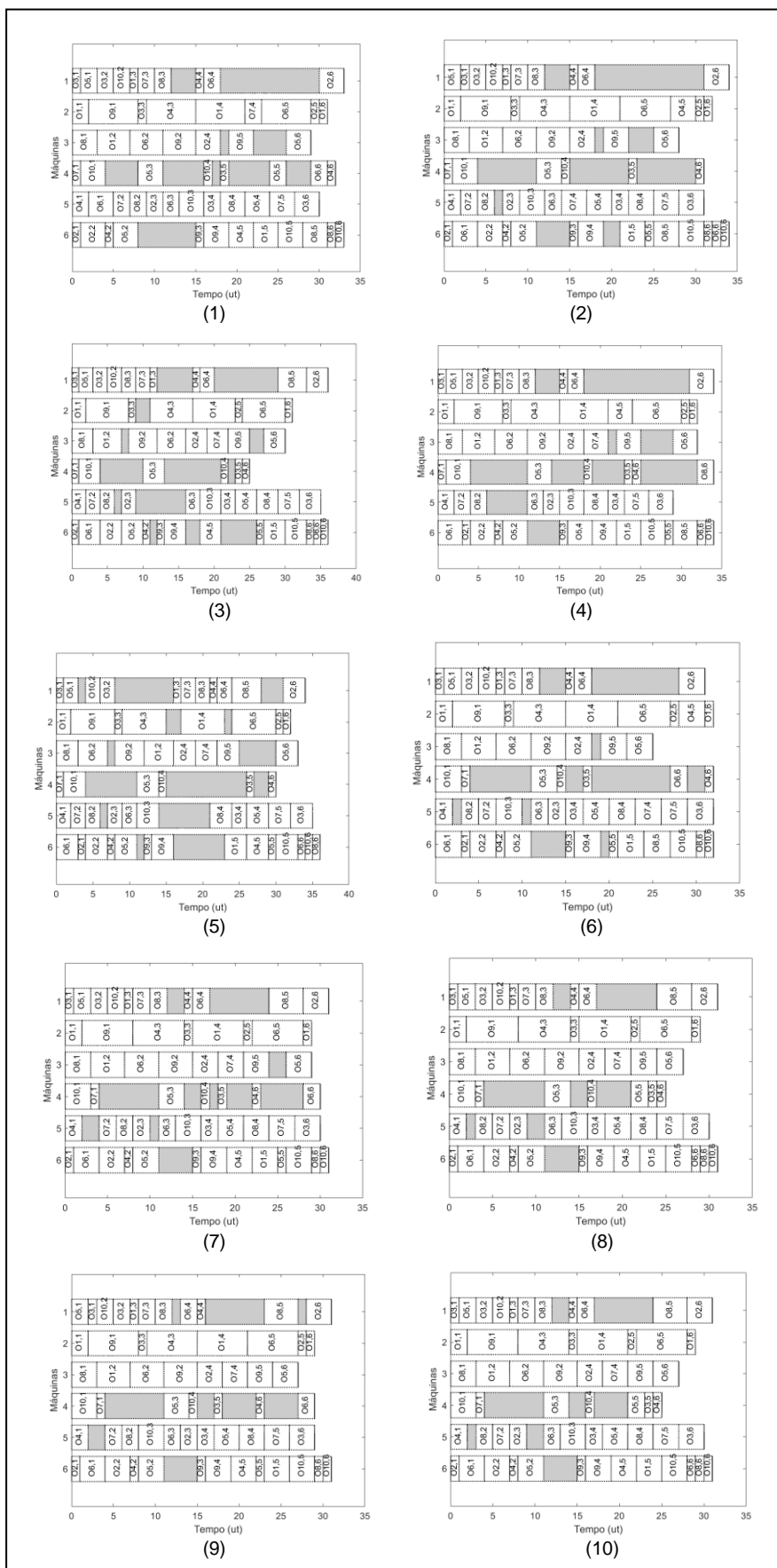
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.23.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK2 caso 2 são mostrados na Figura 5.24.

Nos experimentos relacionados à instância MK2 caso 2, foram alocados altíssima prioridade (AA) aos tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e M1. Para o restante das máquinas da instância foram colocadas baixíssima prioridade (BB) para o maior tempo ocioso entre suas operações.

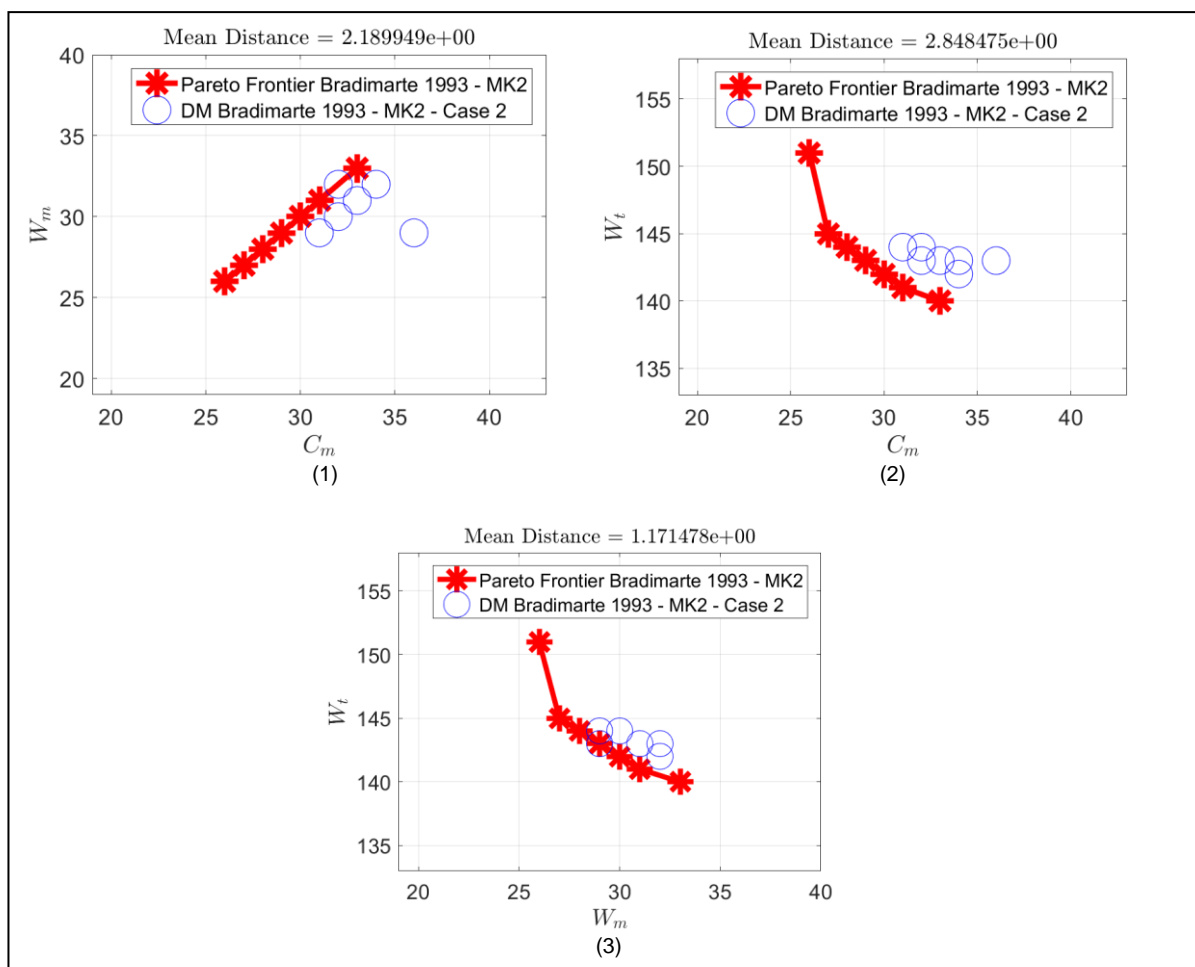
A ideia é de se verificar soluções boas ranqueadas para o DM sem se afastar em demasia das soluções da Fronteira de Pareto e buscar programações da produção que possuam, para a máquina M1, maiores intervalos de tempos ociosos entre suas operações. Pode-se verificar na Tabela 5.23 (em negrito) que as somas dos tempos ociosos são elevadas para cada cenário.

Figura 5.23- Gráficos de Gantt da instância MK2 caso 2 (cenário 2).



Fonte – Próprio autor.

Figura 5.24- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK2 caso 2 (cenário 2).



Fonte – Próprio autor.

A Figura 5.23 ilustra graficamente as programações da produção escolhidas pelo DM para as prioridades para os critérios escolhidos. Visualmente é possível se validar o funcionamento do DM.

As distâncias médias das soluções escolhidas pelo DM e a Fronteira de Pareto são relativamente pequenas ao porte da instância MK2, o que consolida as soluções alcançadas pelo DM.

5.2.3 Bradimarte MK4 caso 3

Para a instância Bradimarte MK4 caso 3 a solução do DM é apresentada nas Tabela 5.24 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.24 – Resultado do DM para Bradimarte MK4 caso 3 (cenário 2).

	C_m	W_t	W_m	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP	CCI
	AA	AA	AA	BB	A	BB	BB	BB	BB	BB	BB			
1	122	328	122	0	118	44	58	72	91	79	28	2,662	1,569	0,629
2	127	328	122	2	124	59	60	69	96	85	25	2,659	1,571	0,629
3	143	326	138	2	140	75	76	90	113	100	93	2,605	1,625	0,616
4	94	348	63	31	88	28	39	35	64	17	12	2,598	1,632	0,614
5	105	344	66	37	79	40	45	58	76	42	26	2,598	1,633	0,614
6	94	348	63	31	88	28	39	33	64	11	10	2,594	1,637	0,613
7	94	348	63	31	88	28	40	29	63	11	10	2,592	1,639	0,613
8	82	344	73	9	66	12	32	44	54	11	13	2,591	1,639	0,613
9	94	348	63	4	88	28	40	29	63	32	10	2,569	1,662	0,607
10	94	348	63	4	88	28	39	29	64	32	10	2,569	1,662	0,607

Fonte – Próprio autor.

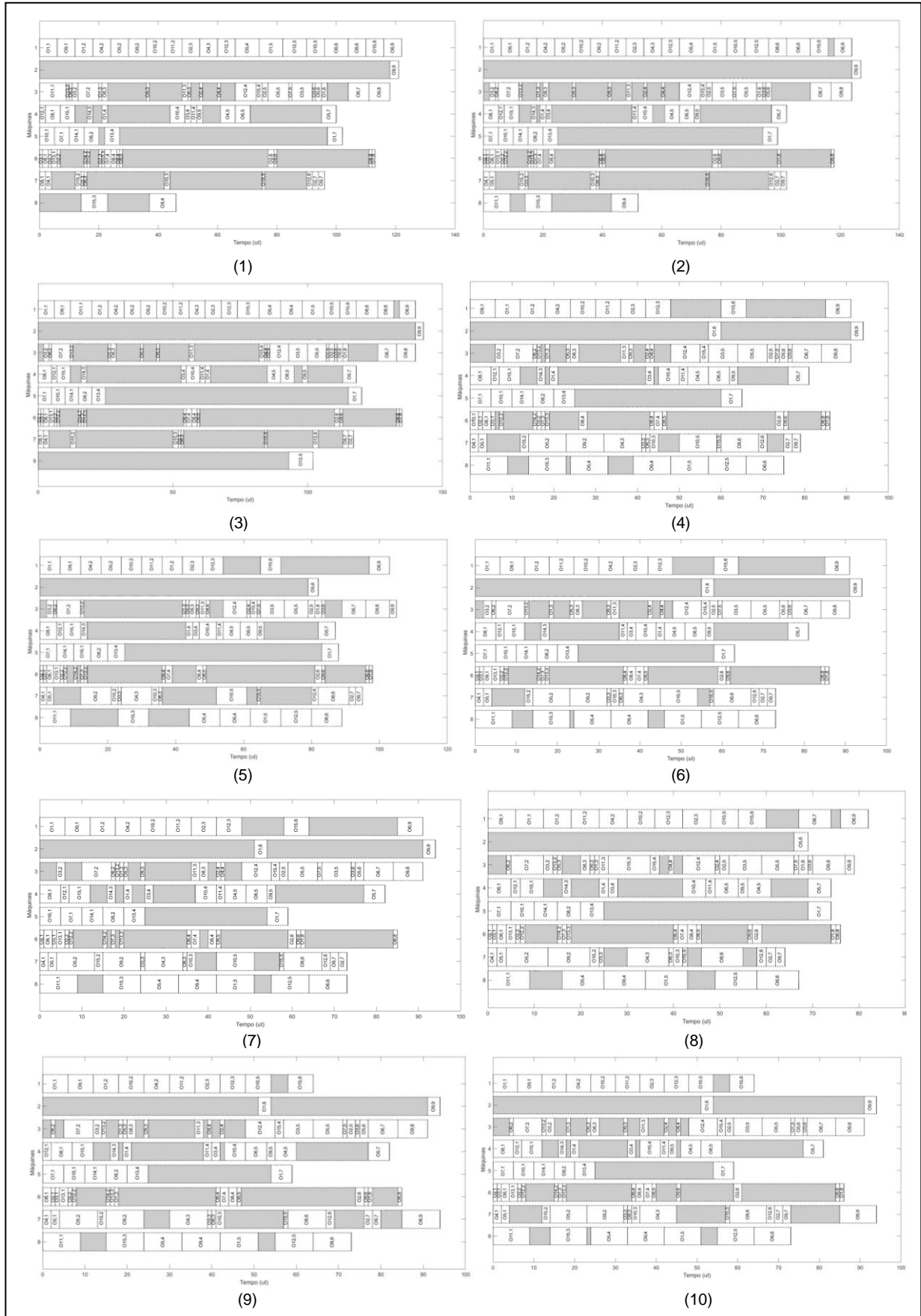
Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.25.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK4 caso 3 são mostrados na Figura 5.26.

Nos experimentos relacionados à instância MK4 caso 3 foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e alta prioridade para M2, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

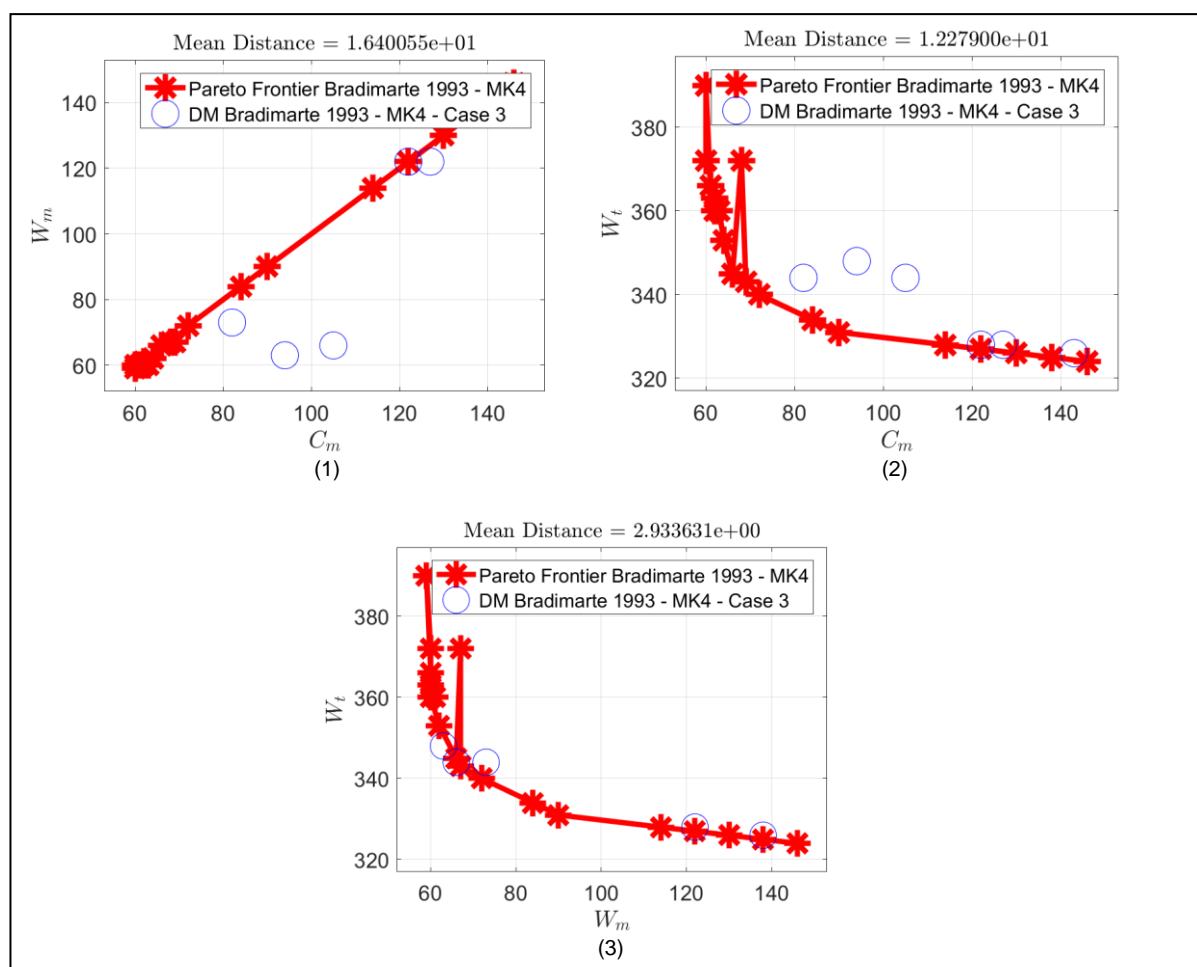
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção. E a alocação de prioridade alta (A) somente para a máquina M2, enquanto as outras ficam com prioridade baixíssima (BB) é de testar a sensibilidade da prioridade alta (A).

Figura 5.25- Gráficos de Gantt da instância MK4 caso 3 (cenário 2).



Fonte – Próprio autor.

Figura 5.26- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK4 caso 3 (cenário 2).



Fonte – Próprio autor.

É esperado que no ranqueamento do DM sejam estabelecidos cenários onde a máquina M2 possua alto tempo ocioso entre as operações e que as soluções apontadas pelo DM fiquem próximas à fronteira de Pareto.

Pode-se observar na Figura 5.25 que a máquina M2 possui grandes intervalos entre operações e que na Figura 5.26 e os valores relacionados como o makespan (C_m) possuem maior discrepâncias da distância média em relação à Fronteira de Pareto. Essa maior distância média de soluções em relação ao C_m está relacionada com a busca do algoritmo por soluções que atendam os requisitos de prioridade dos critérios estabelecido para o ensaio.

5.2.4 Bradimarte MK4 caso 4

Para a instância Bradimarte MK4 caso 4 a solução do DM é apresentada nas Tabela 5.25 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.25 – Resultado do DM para Bradimarte MK4 caso 4 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	SIN	SIP	CCI
	AA	AA	AA	BB	M	BB	BB	BB	BB	BB	BB			
1	82	344	73	9	66	12	32	44	54	11	13	2,453	1,428	0,632
2	105	344	66	37	79	40	45	58	76	42	26	2,432	1,449	0,627
3	122	328	122	0	118	44	58	72	91	79	28	2,415	1,466	0,622
4	94	348	63	31	88	28	39	35	64	17	12	2,414	1,467	0,622
5	94	348	63	31	88	28	39	33	64	11	10	2,410	1,471	0,621
6	94	348	63	31	88	28	40	29	63	11	10	2,407	1,474	0,620
7	96	345	66	28	65	33	34	53	66	33	17	2,404	1,477	0,620
8	81	343	81	0	62	14	28	39	52	17	19	2,404	1,477	0,619
9	81	343	81	0	62	14	28	39	51	17	19	2,403	1,478	0,619
10	89	347	65	27	69	24	35	47	59	7	9	2,403	1,478	0,619

Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na Figura 5.27.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK4 caso 4 são mostrados na Figura 5.28.

Nos experimentos relacionados à instância MK4 caso 4, foram colocados altíssima prioridade (AA) para o tomador de decisão (DM) nos critérios de seleção C_m , W_t , W_m e prioridade média (M) para a máquina M2, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

Os critérios C_m , W_t e W_m com prioridade AA atribuídos ao DM irá fazer o papel de não alterar significativamente o desempenho da produção quanto ao Makespan e os critérios de carga total das máquinas e carga individual das máquinas.

Figura 5.27- Gráficos de Gantt da instância MK4 caso 4 (cenário 2).

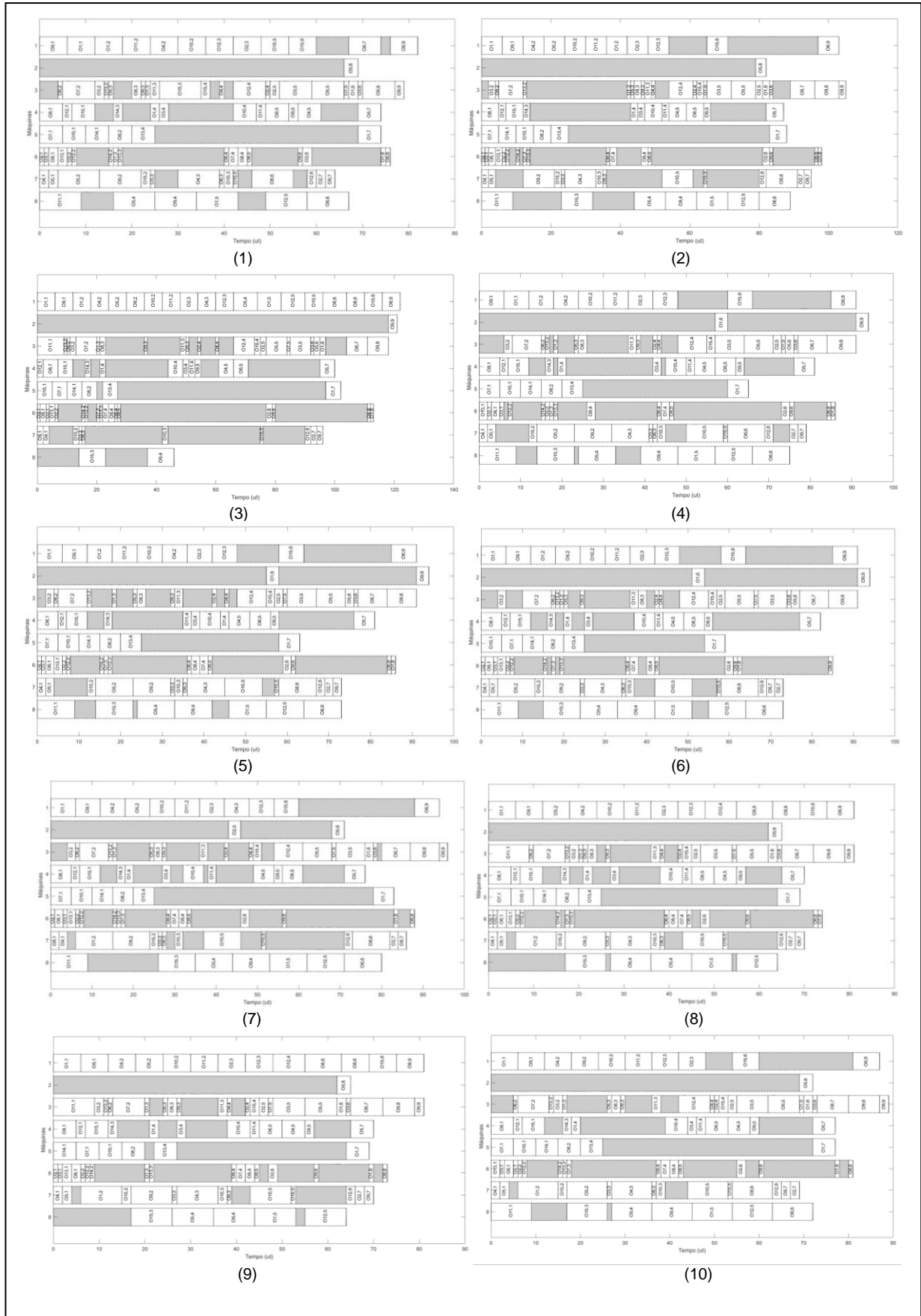
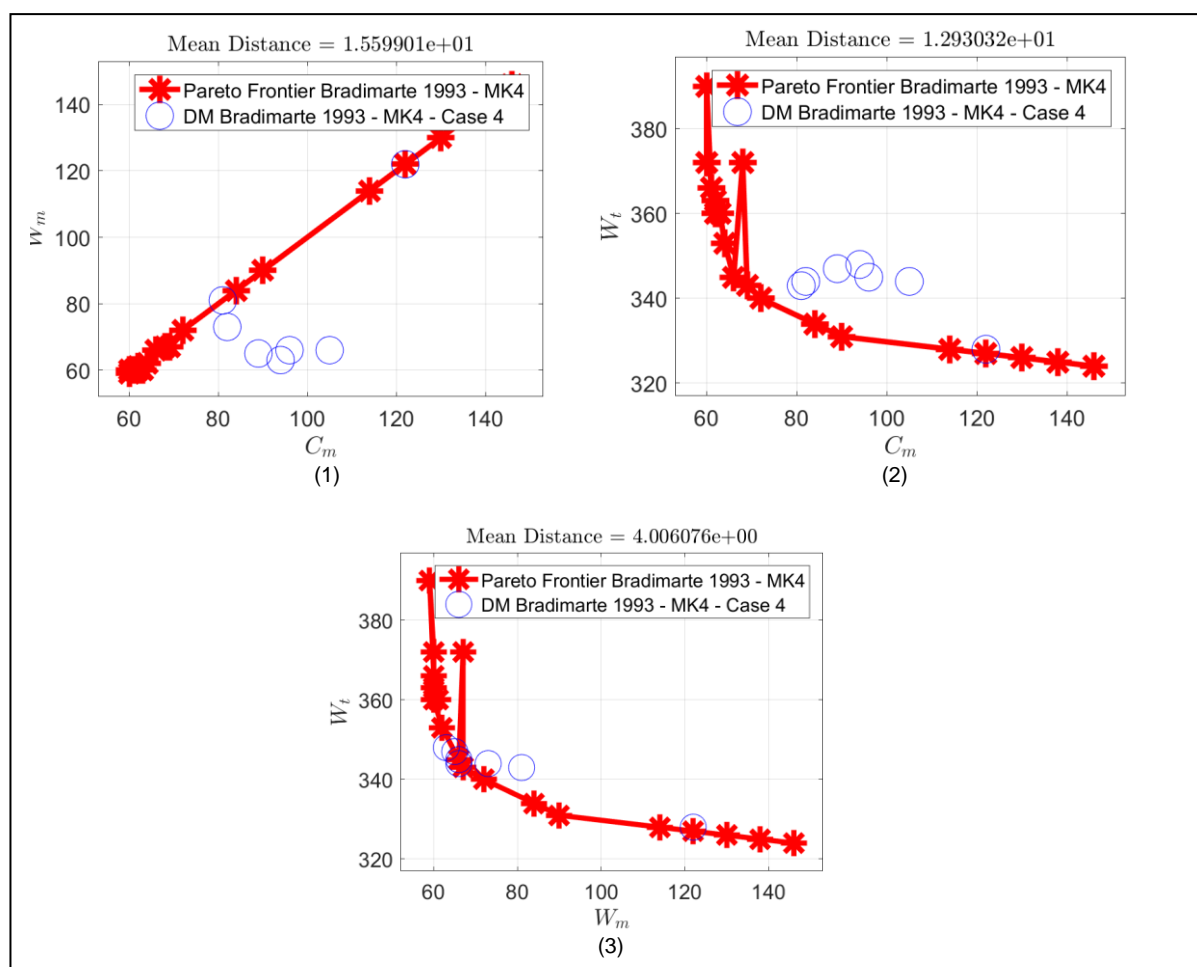


Figura 5.28- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK4 caso 4 (cenário 2).



Fonte – Próprio autor.

Já o critério de médio (M) para a maior quantidade de tempo ocioso entre as operações para a máquina M2 pode ser comparado com os valores da seção anterior (5.2.3) onde são relativamente menores em todas as soluções apontadas pelo DM. Esse era o comportamento esperado para as soluções do DM com o critério M atribuído a M2.

As soluções da Tabela 5.25 são ilustradas na Figura 5.27 na forma de gráficos de Gantt onde duas das soluções do algoritmo FJSP-ABC produzem mais de uma programação da produção (em negrito na Tabela 5.25), as quais são ranqueadas em conjunto com os critérios estabelecidos.

Na Figura 5.28 observa-se novamente que as soluções encontradas pelo DM estão próximas ou sobrepondo-se à Fronteira de Pareto (valores não dominados) e possuem uma distância média equivalente a encontrada na seção anterior (seção 5.2.3).

5.2.5 Bradimarte MK8 caso 5

Para a instância Bradimarte MK8 caso 5 a solução do DM é apresentada nas Tabela 5.26 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.26 – Resultado do DM para Bradimarte MK8 caso 5 (cenário 2).

	C_m	W_t	W_m	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	AA	BB			
1	605	2514	533	63	373	199	285	238	0	210	178	349	72	2,914	1,483	0,663
2	585	2519	524	44	370	180	266	205	0	205	129	330	61	2,877	1,519	0,654
3	578	2514	533	36	362	177	266	205	0	180	138	322	45	2,788	1,608	0,634
4	605	2524	523	63	358	197	281	198	0	238	146	342	90	2,778	1,619	0,632
5	605	2519	524	64	381	184	266	228	0	218	141	334	81	2,775	1,621	0,631
6	612	2509	542	71	372	162	266	230	0	218	189	346	70	2,742	1,654	0,624
7	645	2514	533	104	375	195	279	215	0	272	206	363	112	2,726	1,670	0,620
8	574	2514	533	32	379	168	245	189	0	200	129	313	41	2,718	1,678	0,618
9	585	2519	524	43	364	143	254	207	0	187	133	318	61	2,712	1,684	0,617
10	607	2514	533	65	372	190	266	198	0	223	160	335	74	2,704	1,692	0,615

Fonte – Próprio autor.

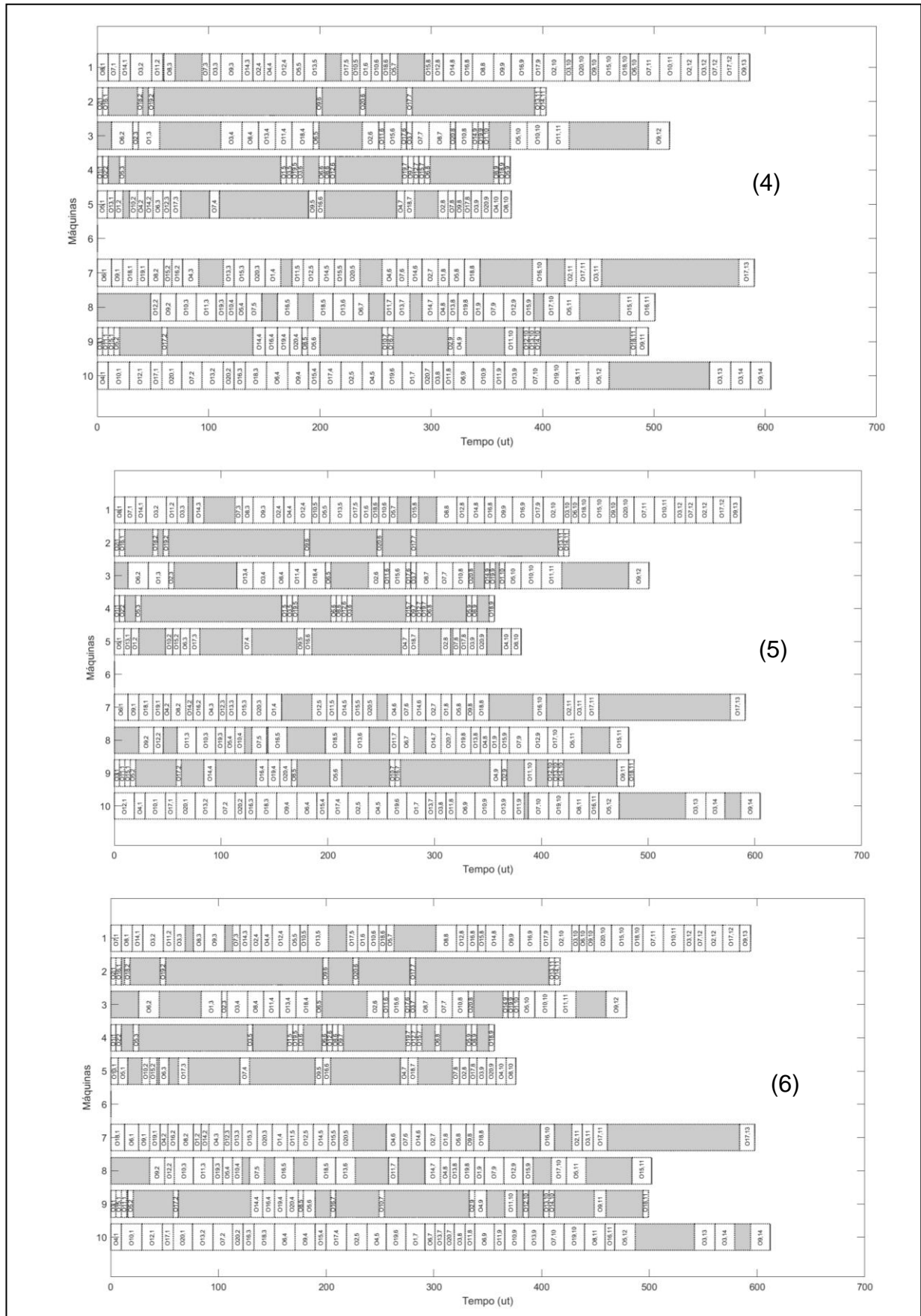
Os gráficos de Gantt para as 9 melhores soluções ranqueadas pelo DM são ilustradas nas Figuras 5.29, 5.30 e 5.31.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK8 caso 5 são mostrados na Figura 5.32.

Nos experimentos relacionados à instância MK8 caso 5, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e também para M9, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

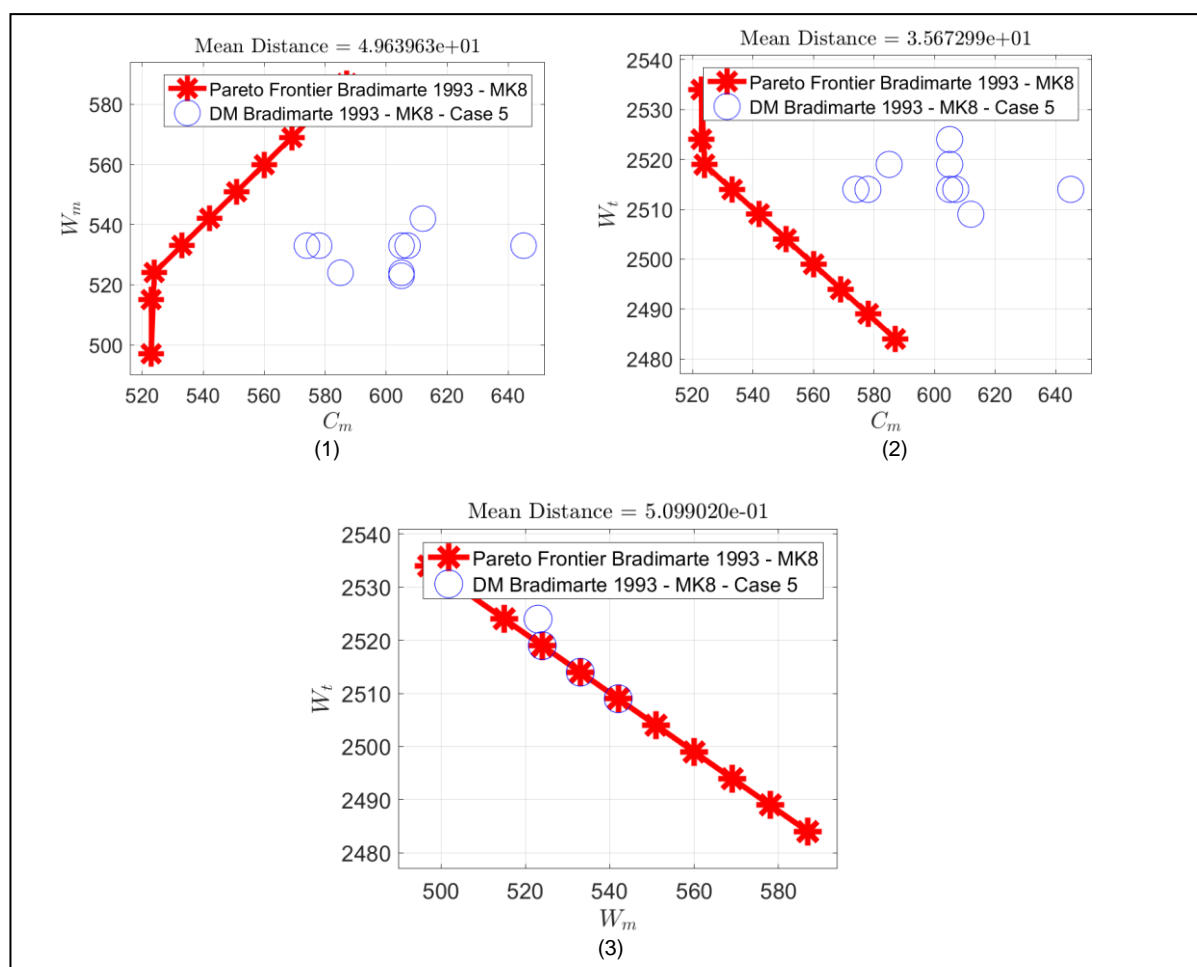
A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção.

Figura 5.30- Gráficos de Gantt da instância MK8 caso 5 – parte 2 (cenário 2).



Fonte – Próprio autor.

Figura 5.32- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK8 caso 5 (cenário 2).



Fonte – Próprio autor.

A partir dessa escolha de critérios é esperado que no ranqueamento do DM sejam estabelecidos cenários onde a máquina M9 possua alto tempo ocioso entre as operações, ou seja, a somatória dos tempos ociosos entre as operações da máquina M9 seja relativamente alto. Isto pode ser observado na Figuras 5.29, 5.30 e 5.31, onde são escolhidos cenários com altos tempos de ociosidade para a máquina M9.

Observa-se na Figura 5.32 que as distâncias médias não são pequenas em relação as soluções da Fronteira de Pareto, o que é justificado pela busca de soluções pelo DM de acordo com os parâmetros de peso estabelecidos para os critérios e pela magnitude de C_m , W_t e W_m do problema.

5.2.6 Bradimarte MK8 caso 6

Para a instância Bradimarte MK8 caso 6 a solução do DM é apresentada nas Tabela 5.30 onde na primeira linha um são apresentados os critérios utilizados, os coeficientes de proximidade e de máximo e mínimo, os quais compuseram o decisor. Na segunda linha as variáveis linguísticas para cada critério, e nas linhas a posteriori são apresentadas as dez melhores soluções para cada caso, com seus respectivos coeficientes de proximidade e de máximo e mínimo.

Tabela 5.27 – Resultado do DM para Bradimarte MK8 caso 6 (cenário 2).

	Cm	Wt	Wm	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	SIN	SIP	CCI
	AA	AA	AA	BB	BB	BB	BB	BB	BB	BB	BB	H	BB			
1	605	2514	533	63	373	199	285	238	0	210	178	349	72	2,822	1,466	0,658
2	585	2519	524	44	370	180	266	205	0	205	129	330	61	2,808	1,480	0,655
3	578	2514	533	36	362	177	266	205	0	180	138	322	45	2,729	1,560	0,636
4	605	2519	524	64	381	184	266	228	0	218	141	334	81	2,702	1,587	0,630
5	605	2524	523	63	358	197	281	198	0	238	146	342	90	2,695	1,594	0,628
6	574	2514	533	32	379	168	245	189	0	200	129	313	41	2,670	1,619	0,623
7	582	2519	524	41	389	166	254	236	0	178	123	311	58	2,658	1,631	0,620
8	585	2519	524	43	364	143	254	207	0	187	133	318	61	2,657	1,631	0,620
9	612	2509	542	71	372	162	266	230	0	218	189	346	70	2,654	1,634	0,619
10	582	2514	533	41	372	144	259	198	0	216	159	316	49	2,640	1,648	0,616

Fonte – Próprio autor.

Os gráficos de Gantt para as 10 melhores soluções ranqueadas pelo DM são ilustradas na figuras 5.33, 5.34 e 5.35.

Os gráficos que ilustram os resultados encontrados pelo DM e a Fronteira de Pareto para caso função de $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ para MK8 caso 6 são mostrados na Figura 5.36.

Nos experimentos relacionados à instância MK8 caso 6, foram colocados altíssima prioridade (AA) para os tomadores de decisão (DM) nos critérios de seleção C_m , W_t , W_m e alta prioridade (A) para a máquina M9, sendo que às outras máquinas foram atribuídas baixíssima prioridade (BB).

A ideia de manter os critérios C_m , W_t e W_m com prioridade AA é a de que esses critérios não sofram mudanças significativas, buscando não modificar muito o desempenho da produção.

Figura 5.33- Gráficos de Gantt da instância MK8 caso 6 – parte 1 (cenário 2).

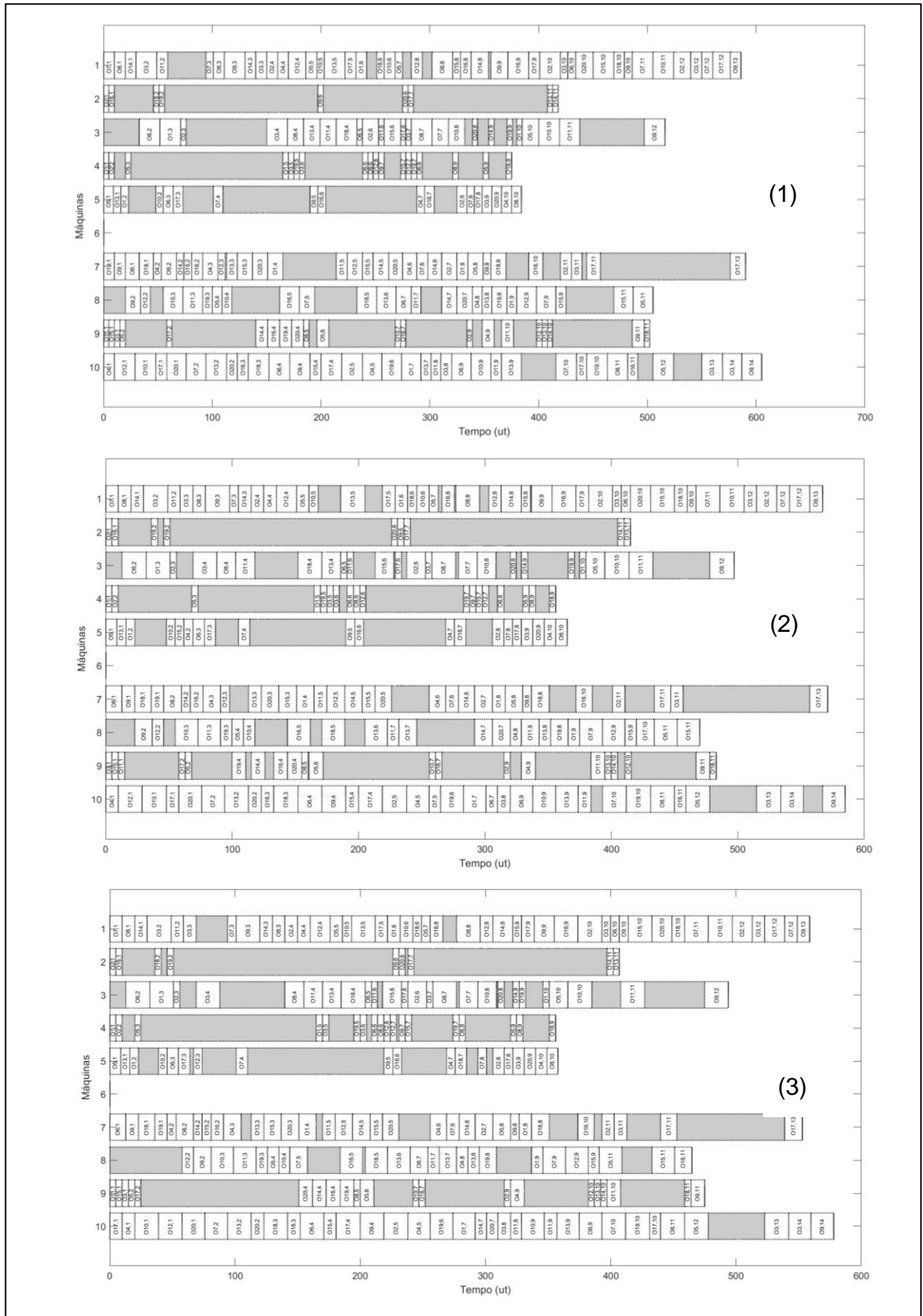
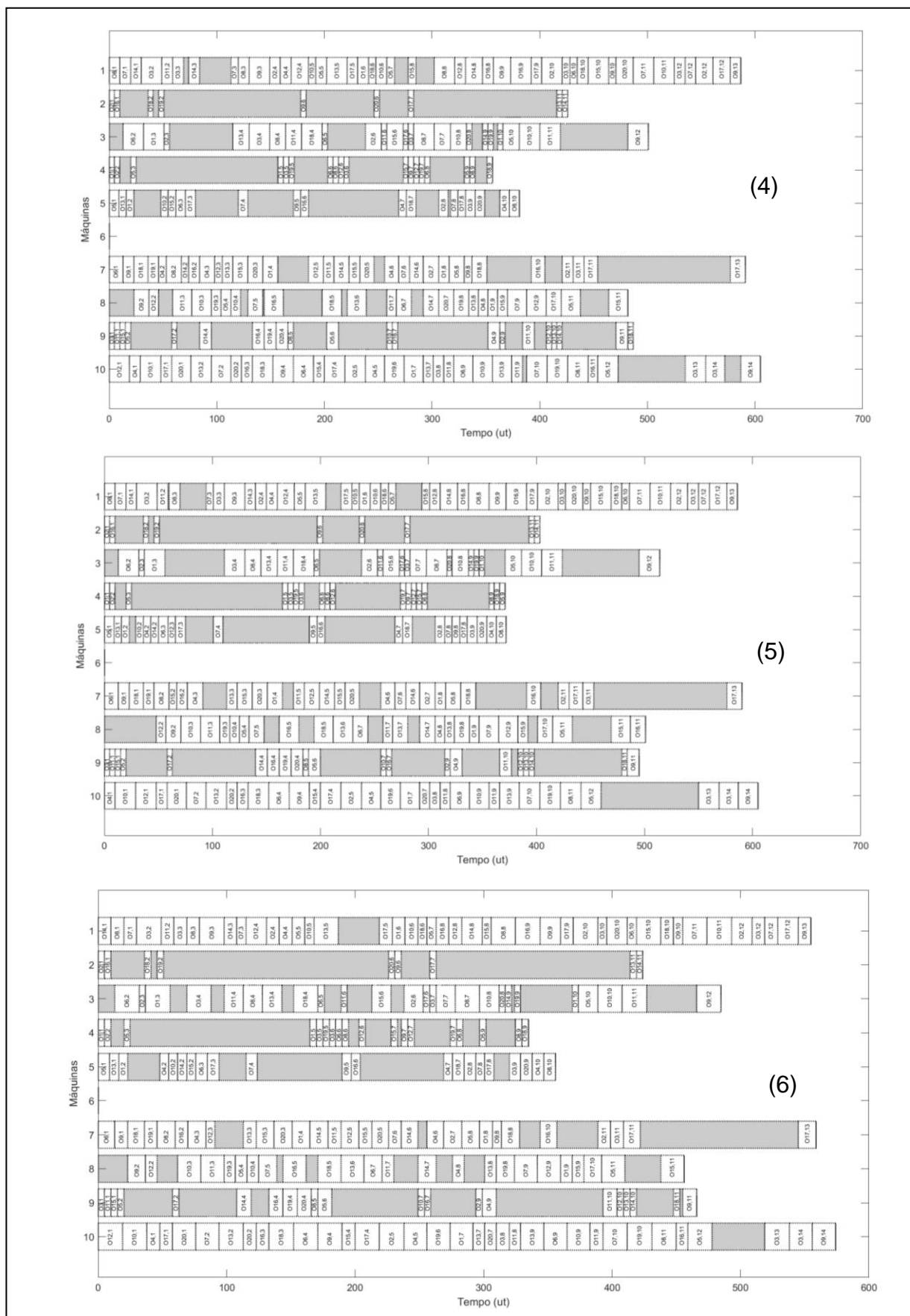
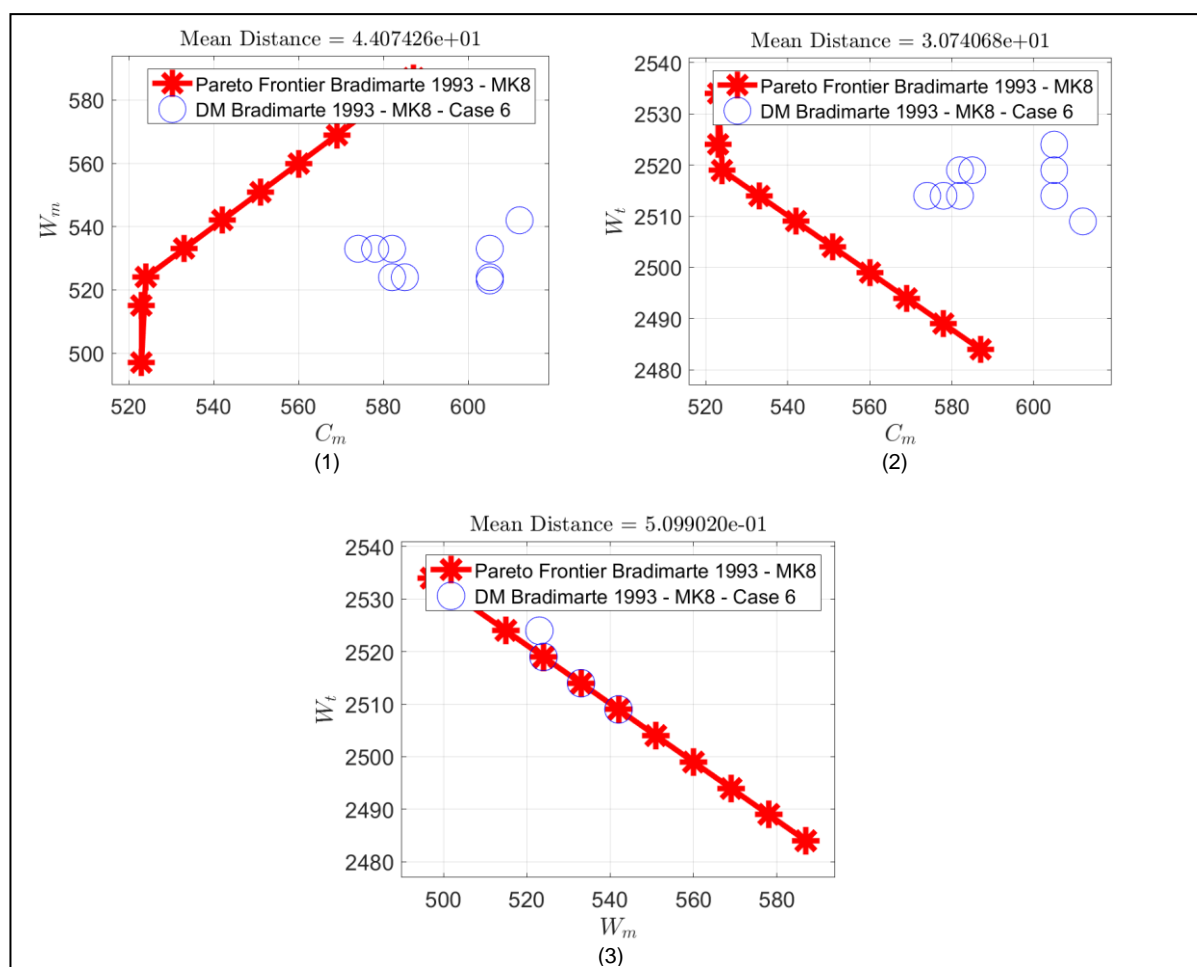


Figura 5.34- Gráficos de Gantt da instância MK8 caso 6 – parte 2 (cenário 2).



Fonte – Próprio autor.

Figura 5.36- Gráficos $C_m \times W_t$, $C_m \times W_m$ e $W_t \times W_m$ da instância MK8 caso 6 (cenário 2).



Fonte – Próprio autor.

Utilizando a prioridade alta (A) para que a máquina M9 deva possuir uma quantidade de tempo ocioso alto entre suas operações espera-se analisar a sensibilidade em relação ao parâmetro de altíssima prioridade (AA) estabelecido neste cenário na seção anterior (seção 5.2.5).

Pode-se observar na Tabela 5.27 que há uma ligeira diminuição dos tempos ociosos para a máquina 9 em relação a Tabela 5.26, principalmente nas soluções do DM ranqueadas acima da terceira posição. Isto mostra que a troca dos parâmetros de AA para A foi válida para o DM.

Na Figura 5.36 observa-se que as soluções encontradas pelo DM estão muito próximas aos resultados da Figura 5.32, ou seja, estabelecendo distâncias médias semelhantes para os critérios C_m , W_t e W_m .

5.3 Discussão Final

Foram realizados experimentos para o DM em dois cenários diferentes, um no qual foi priorizado o menor tempo ocioso entre operações de uma máquina (Cenário 1) e outro onde foi priorizado o maior tempo ocioso entre operações de uma máquina (Cenário 2). Para isto foram utilizadas as instâncias de Kacem, Mammadi e Bourne (2002) e Bradimarte (1993). Em ambos os cenários foram apresentados seis casos diferentes de configuração das prioridades dos critérios de decisão do DM. Foram apresentadas as 10 primeiras soluções ranqueadas pelo DM para cada caso, seus gráficos de Gantt de programação da produção e os gráficos comparativos da Fronteira de Pareto com as soluções encontradas pelo DM.

Na maioria dos experimentos utilizou-se a altíssima prioridade (AA) para os critérios de seleção C_m , W_t e W_m com o objetivo de que os resultados selecionados pelo DM em seu ranqueamento ficassem o mais perto possível da Fronteira de Pareto, buscando não modificar muito o desempenho da produção.

Para os ensaios onde C_m , W_t e W_m , possuíam prioridades diferentes de AA, as soluções se afastaram da Fronteira de Pareto, ou seja, é possível se considerar soluções úteis longe da Fronteira de Pareto, dependendo do ajuste dos parâmetros dos pesos do critério de desempenho escolhido. que pode ser explicado devido a diversidade de Gráficos de Gantt para cada solução de C_m , W_t e W_m fornecida pelo algoritmo FJSP-ABC.

A medida da distância média das soluções encontradas pelo DM em relação à Fronteira de Pareto depende da qualidade dos resultados fornecidos pelo algoritmo FJSP-ABC, ou seja, o quanto os resultados do algoritmo se aproximam da Fronteira de Pareto (resultados ótimos) de cada instância e da diversidade de soluções de gráficos de Gantt para cada solução encontrada.

Pode-se observar que para cada uma das soluções encontradas de C_M , W_T e W_M fornecidas pelo algoritmo FJSP-ABC, pode haver gráficos de Gantt diferentes, ou seja, muitas soluções ranqueadas pelo DM apresentaram-se com os mesmos valores C_M , W_T e W_M , no entanto, umas se apresentaram melhores que outras dependendo da configuração escolhida das prioridades das máquinas da instância.

Na maioria dos experimentos as soluções encontradas pelo DM estão muito próximas ou sobrepondo-se à Fronteira de Pareto (valores não dominados), para as instâncias de pequeno e médio porte, possuindo uma distância média muito baixa, confirmando-se o que era esperado para esta maneira de escolha dos pesos dos critérios de seleção dos cenários. Já nos experimentos com instâncias maiores, proporcionalmente a magnitude do problema as distâncias médias

podem ser consideradas satisfatórias no sentido de se atenderem as prioridades dos critérios estabelecidos.

Nos testes de sensibilidade do algoritmo do DM realizados para a verificação dos pesos dos critérios nas soluções fornecidas, foram comparados instâncias e casos diferentes com a mudança do peso dos critérios de AA para A e para M, sendo que as soluções da programação da produção reagiram de maneira satisfatória com o desejado.

Também foram realizados testes comparativos com as trocas das prioridades para máquina a máquina na mesma instância, sendo verificado o funcionamento adequado do algoritmo do DM. Assim com uma miscelânea de trocas, inclusive para o C_m , W_t e W_m e o algoritmo funcionou de forma satisfatória no ranqueamento das soluções.

Capítulo 6

CONCLUSÕES

Conforme descrito, o objetivo desse trabalho foi o de implementar um método de escolha da melhor programação da produção de um FJSP multiobjetivo, na forma de um gráfico de Gantt, para um sistema produtivo.

Para isto foi utilizado um algoritmo programador da produção multiobjetivo utilizando uma meta-heurística com estratégias de busca diversas, baseada na Colônia de Abelhas Artificiais (FJSP - ABC).

O algoritmo de otimização FJSP - ABC fornece um arquivo com todas as soluções encontradas na otimização de três objetivos, minimização de tempo de conclusão de todas as operações Makespan (C_m), minimização da carga total de todas as máquinas (W_t) e minimização da carga atribuída a máquina mais carregada (W_m). As soluções encontradas possuem estes objetivos e seus gráficos de Gantt de forma a fornecerem uma alta diversidade de resultados na fronteira ou perto da fronteira de Pareto.

Neste trabalho foi implementado um algoritmo de um Tomador de Decisão (DM) baseado em Fuzzy-TOPSIS para escolher a partir das programações da produção geradas no algoritmo FJSP - ABC a melhor programação de acordo outras variáveis do sistema produtivo. O DM utiliza variáveis relacionadas as máquinas de forma individual permitindo que a tomada de decisão da melhor programação da produção seja mais aderente com as condições atuais, do momento, do sistema produtivo. Nesse trabalho foi utilizada a variável incorporada de ociosidade de cada máquina, ou seja, a ideia é otimizar o tempo ocioso de cada máquina.

Dois cenários foram implementados, sendo um que tratava de minimizar o tempo ocioso interoperações de uma máquina (Cenário 1) e outro que tratava de maximizar o tempo ocioso

interoperações de uma máquina (Cenário 2). Os dois cenários utilizaram os benchmarks clássicos para o problema encontrados na literatura, ou seja, Kacem, Hammadi e Bourne (2002) e Bradimarte (1993).

Para a validação do algoritmo do DM implementado foram realizados testes utilizando os dois cenários citados, onde foi verificado e validado em função da representação de sua solução nos gráficos de Gantt e de um gráfico comparativo das soluções encontradas pelo DM e das soluções das instâncias na Fronteira de Pareto o funcionamento do algoritmo. Conclui-se que a partir dos resultados dos testes realizados para os dois cenários que o DM funcionou de forma satisfatória no ranqueamento dos melhores cenários frente as necessidades estabelecidas pelo usuário de acordo com as condições de interesse do sistema produtivo.

Dessa forma os objetivos deste trabalho foram atingidos com sucesso, fornecendo um método baseado em um algoritmo de tomada de decisão que envolve a escolha da melhor programação da produção de um FJSP multiobjetivo de acordo com as condições ambientais (relacionados à utilização das máquinas) e/ou necessidades de funcionamento de um sistema produtivo.

6.1 Trabalhos Futuros

Pretende-se como próximos passos, realizar essas modificações e refinamentos, de forma a se utilizar vários tipos de benchmarks e cenários do sistema produtivo para análise das soluções, de forma a se obter uma validação mais robusta.

Outros mecanismos de tomada de decisão também podem ser implementados para verificar e melhorar o desempenho do DM, de acordo com as necessidades do sistema produtivo, por exemplo cenários com a exclusão de alguma máquina etc.

Outra possibilidade é a utilização do DM para outros sistemas de produção ou mesmo o FJSP mono objetivo, de forma a prover uma ferramenta efetiva para ser utilizada nos ambientes dos sistemas produtivos reais.

REFERÊNCIAS BIBLIOGRÁFICAS

ABD, K.; ABHARY, K.; MARIAN, R. *Development of fuzzy ahp and topsis approach for multi-objective scheduling problems in robotic flexible assembly cells*. In: *DAAAM INTERNATIONAL SCIENTIFIC BOOK*, 2017. v. 16, cap. 20, p. 265–278. ISBN 9783902734129. Disponível em: <<https://doi.org/10.2507/daaam.scibook.2017.20>>.

BRANDIMARTE, P. *Routing and scheduling in a flexible job shop by tabu search*. *Applications In Production And Scheduling*, v. 41, p. 157–183, 1993. Disponível em: <<https://doi.org/10.1007/BF02023073>>.

CAO, Y.; SHI, H. *An adaptive multi-population artificial bee colony algorithm for multi-objective flexible job shop scheduling problem*. In: **2019 Chinese Control And Decision Conference (CCDC)**. [s.n.], 2019. p. 3822–3827. Disponível em: <<https://doi.org/10.1109/CCDC.2019.8833005>>.

CHAKLADAR, N. D.; CHAKRABORTY, S. *A combined topsis-ahp-method-based approach for non-traditional machining processes selection*. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, v. 222, n. 12, p. 1613–1623, 2008. Disponível em: <<https://doi.org/10.1243/09544054JEM1238>>.

CHANG, Y.; MATSUO, H.; SULLIVAN, R. S. *A bottleneck-based beam search for job scheduling in a flexible manufacturing system*. *The International Journal of Production Research*, Taylor & Francis, v. 27, n. 11, p. 1949–1961, 1989. Disponível em: <<https://doi.org/10.1080/00207548908942666>>.

CHAO, L.-P.; LIN, T.-C.; LU, C.-L. *Flexible job-shop scheduling based on hybrid artificial bee colony algorithm with diversity index search for multiple decisions making in manufacturing system*. *International Journal of Advanced Engineering and Management Research*, v. 2, n. 5, p. 1403–1417, 2017.

CHEN, C.-T. *Extensions of the topsis for group decision-making under fuzzy environment*. *Fuzzy Sets and Systems*, v. 114, n. 1, p. 1–9, 2000. ISSN 0165-0114. Disponível em: <[https://doi.org/10.1016/S0165-0114\(97\)00377-1](https://doi.org/10.1016/S0165-0114(97)00377-1)>.

CHEN, H.; IHLOW, J.; LEHMANN, C. *A genetic algorithm for flexible job-shop scheduling*. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. [s.n.], 1999. v. 2, p. 1120–1125 vol.2. Disponível em: <<https://doi.org/10.1109/ROBOT.1999.772512>>.

CUI, Y. et al. *Review: Multi-objective optimization methods and application in energy saving*. *Energy*, v. 125, p. 681–704, 2017. ISSN 0360-5442. Disponível em: <<https://doi.org/10.1016/j.energy.2017.02.174>>.

DEB, K. *Multi-objective optimization using evolutionary algorithms*. [S.l.]: John Wiley & Sons, 2001. v. 16.

FERREIRA, J. C.; FONSECA, C. M.; GASPAR-CUNHA, A. *Methodology to select solutions from the pareto-optimal set: A comparative study*. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: Association for Computing Machinery, 2007. (GECCO '07), p.789–796. ISBN 9781595936974. Disponível em: <<https://doi.org/10.1145/1276958.1277117>>.

FERREIRA, J. H. I. **Um modelo de otimização multiobjetivo aplicado no dimensionamento da capacidade instalada de uma Pequena Central Hidrelétrica**. Tese (Doutorado) — Universidade Federal de Uberlândia, 01/2020. Disponível em: <<http://doi.org/10.14393/ufu.te.2020.70>>.

FLORÊNCIO, G. F. **Algoritmo baseado em colônia de abelhas artificiais para resolução do problema de programação de um job shop flexível multiobjetivo**. Dissertação (Mestrado) — Universidade Federal de São Carlos, 02/2019. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/11793>>

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. *The complexity of flowshop and job-shop scheduling*. *mathematics of operations research*. *Mathematics of operations research, INFORMS*, v. 1, n. 2, p. 117–129, 1976. Disponível em: <<https://doi.org/10.1287/moor.1.2.117>>.

GOLDBARG, M. C.; GOLDBARG, E. G.; LUNA, H. P. L. **Otimização Combinatória e Meta-Heurísticas: Algoritmos e Aplicações**. 1. ed. Rio de Janeiro: Elsevier, 2016. ISBN 978-85-352-7812-5.

HUANG, S. et al. *Multi-objective flexible job-shop scheduling problem using modified discrete particle swarm optimization*. **SpringerPlus**, *SpringerOpen*, v. 5, n. 1, p. 1–22, 2016. Disponível em: <<https://doi.org/10.1186/s40064-016-3054-z>>.

HUANG, X.; GUAN, Z.; YANG, L. *An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem*. **Advances in Mechanical Engineering**, v. 10, n. 9, 2018. Disponível em: <<https://doi.org/10.1177/1687814018801442>>.

HWANG, C.-L.; MASUD, A. S. M. ***Multiple Objective Decision Making — Methods and Applications***. Springer Berlin Heidelberg, 1979. Disponível em: <<https://doi.org/10.1007%2F978-3-642-45511-7>>.

HWANG, C.-L.; YOON, K. ***Multiple Attribute Decision Making***. Springer Berlin Heidelberg, 1981. Disponível em: <<https://doi.org/10.1007%2F978-3-642-48318-9>>.

JIA, S.; HU, Z.-H. *Path-relinking tabu search for the multi-objective flexible job shop scheduling problem*. **Comput. Oper. Res.**, Elsevier Science Ltd., GBR, v. 47, p. 11–26, jul. 2014. ISSN 0305-0548. Disponível em: <<https://doi.org/10.1016/j.cor.2014.01.010>>.

KACEM, I.; HAMMADI, S.; BORNE, P. *Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems*. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 32, n. 1, p. 1–13, 2002. Disponível em: <<https://doi.org/10.1109/TSMCC.2002.1009117>>.

KARABOGA, D. ***An Idea Based on Honey Bee Swarm for Numerical Optimization***. Kayseri/Turquia, 2005. TR06.

KATO, E. R. R., ARANHA, G. D., TSUNAKI, R. H., *A new approach to solve the flexible job shop problem based on a hybrid particle swarm optimization and Random-Restart Hill Climbing*, **Computers & Industrial Engineering**, 2018, no.125, pp. 178-189.

KEFALAS, M. et al. *A tabu search-based memetic algorithm for the multi-objective flexible job shop scheduling problem*. In: **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. New York, NY, USA: Association for Computing Machinery, 2019. (GECCO'19), p. 1254–1262. ISBN 9781450367486. Disponível em: <<https://doi.org/10.1145/3319619.3326817>>.

KONAK, A.; COIT, D. W.; SMITH, A. E. *Multi-objective optimization using genetic algorithms: A tutorial*. **Reliability Engineering System Safety**, v. 91, n. 9, p.

992–1007, 2006. ISSN 0951-8320. *Special Issue - Genetic Algorithms and Reliability*. Disponível em: <<https://doi.org/10.1016/j.res.2005.11.018>>.

KUMAR, S.; KUMAR, S.; BARMAN, A. G. *Supplier selection using fuzzy topsis multi criteria model for a small scale steel manufacturing unit*. **Procedia Computer Science**, v. 133, p. 905–912, 2018. ISSN 1877-0509. *International Conference on Robotics and Smart Manufacturing (RoSMa2018)*. Disponível em: <<https://doi.org/10.1016/j.procs.2018.07.097>>.

LI, J.-Q.; PAN, Q.-K.; GAO, K.-Z. *Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems*. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 55, n. 9, p. 1159–1169, 2011. Disponível em: <<https://doi.org/10.1007/s00170-010-3140-2>>.

LI, J.-Q.; PAN, Q.-K.; TASGETIREN, M. F. *A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities*. **Applied Mathematical Modelling**, v. 38, n. 3, p. 1111–1132, 2014. ISSN0307-904X. Disponível em: <<https://doi.org/10.1016/j.apm.2013.07.038>>.

Li J.Q., Pan Q.K., Liang Y.C., (2010) An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Comput Indust Eng.*, 59(4): p.647–662.

LIMA JUNIOR, F. R.; CARPINETTI, L. C. R. *Uma comparação entre os métodos topsis e fuzzy-topsis no apoio à tomada de decisão multicritério para seleção de fornecedores*. **Gestão Produção**, Scielo, v. 22, p. 17 – 34, 03 2015. ISSN 0104-530X. Disponível em: <<http://dx.doi.org/10.1590/0104-530X1190>>.

LOBATO, F. S. **Otimização multi-objetivo para o projeto de sistemas de engenharia**. Tese (Doutorado) — Universidade Federal de Uberlândia, 9 2008. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14677>>.

LUUKKA, P. Luukka, p.: *Feature selection using fuzzy entropy measures with similarity classifier*. *expert syst. appl.* 38(4), 4600-4607. **Expert Systems with Applications**, v. 38, p. 4600–4607, 04 2011. Disponível em: <<https://doi.org/10.1016/j.eswa.2010.09.133>>.

MACCARTHY, B. L.; J.LIU. *Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling*. **The International Journal of Production Research**, Taylor & Francis, v. 31, n. 1, p. 59–79, 1993. Disponível em: <<https://doi.org/10.1080/00207549308956713>>.

MENG, G.-J. et al. *A hybrid artificial bee colony algorithm for multi-objective flexible job-shop scheduling problem*. *Journal of Computers*, v. 31, n. 5, p. 224–235, 2020. Disponível em: <<https://doi.org/10.3966/199115992020103105017>>.

PERERA, A. et al. *A hybrid tool to combine multi-objective optimization and multi-criterion decision making in designing standalone hybrid energy systems*. *Applied Energy*, v. 107, p. 412–425, 2013. ISSN 0306-2619. Disponível em: <<https://doi.org/10.1016/j.apenergy.2013.02.049>>.

PREVIERO, W. D. **Estratégias de resolução para o problema de Job- Shop Flexível**. Tese (Doutorado) — Universidade de São Paulo, Instituto de Matemática e Estatística, 2016. Disponível em: <<https://doi.org/10.11606/T.45.2016.tde-20102016-123243>>.

SAATY, T. L. *Analytic Heirarchy Process*. American Cancer Society, 2014. ISBN 9781118445112. Disponível em: <<https://doi.org/10.1002/9781118445112.stat05310>>.

SALIH, M. M. et al. *Survey on fuzzy topsis state-of-the-art between 2007 and 2017*. *Computers Operations Research*, v. 104, p. 207–227, 2019. ISSN 0305-0548. Disponível em: <<https://doi.org/10.1016/j.cor.2018.12.019>>.

SHARMA, P.; SINGHAL, S. *Implementation of fuzzy topsis methodology in selection of procedural approach for facility layout planning*. *Int J Adv Manuf Technol*, n. 88, p. 1485–1493, 02 2017. Disponível em: <<https://doi.org/10.1007/s00170-016-8878-8>>.

SLACK, N. et al. **Administração da produção**. 1. ed. São Paulo: Atlas, 2006. ISBN 85-224-2171-4.

TAY, J. C.; WIBOWO, D. *An effective chromosome representation for evolving flexible job shop schedules*. In: DEB, K. (Ed.). *Genetic and Evolutionary Computation – GECCO 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 210–221. ISBN 978-3-540-24855-2. Disponível em: <<https://doi.org/10.1080/00207549308956713>>.

TUBINO, D. **Planejamento E Controle Da Produção - Teoria e prática**. São Paulo: Atlas, 2007. ISBN 9788597013054.

VAN LAARHOVEN, P.; PEDRYCZ, W. *A fuzzy extension of saaty's priority theory*. *Fuzzy Sets and Systems*, v.11, n.1, p. 229–241, 1983. ISSN 0165-0114. Disponível em: <[https://doi.org/10.1016/S0165-0114\(83\)80082-7](https://doi.org/10.1016/S0165-0114(83)80082-7)>.

WANG, L. et al. *A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. International Journal of Production Research*, Taylor Francis, v.51, n. 12, p. 3593–3608, 2013. Disponível em: <<https://doi.org/10.1080/00207543.2012.754549>>.

YANG, X.-S. *Engineering optimization: an introduction with metaheuristic applications*. 1. ed. [S.l.]: John Wiley & Sons, 2010. ISBN -10 0470582464.

WANG, L, CAI, J., LI, M., LIU, Z., *Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization, Scientific Programming*, v 2017, Article ID 9016303, 11 pages

XIA, W. e WU, Z., *An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers & Industrial Engineering*, v. 48, n. 2, p. 409–425