

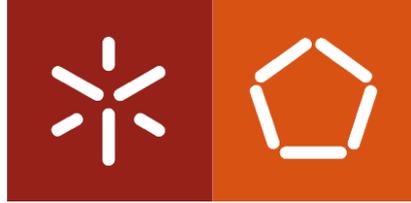


Universidade do Minho
Escola de Engenharia

Sibelle Moreira Pereira

Aplicação de Simulação para a resolução de um problema de escalonamento da produção em máquinas paralelas e redução do tempo total de processamento

Novembro de 2022



Universidade do Minho
Escola de Engenharia

Sibelle Moreira Pereira

Aplicação de Simulação para a resolução de um problema de escalonamento da produção em máquinas paralelas e redução do tempo total de processamento

Dissertação de Mestrado em Engenharia Industrial
Especialização em Gestão Industrial

Trabalho efetuado sob a orientação do
Professora Doutora Maria Leonilde Rocha Varela
Professor Marcelo Filipe Nunes Henriques

Novembro de 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição

CC BY

<https://creativecommons.org/licenses/by/4.0/>

AGRADECIMENTOS

Este trabalho significa um passo muito importante e desejado na minha vida. Agradeço primeiramente a Deus, pois me conduziu por todo o caminho até chegar aqui.

Meus sinceros agradecimentos à minha orientadora, professora Maria Leonilde Rocha Varela, que me apoiou, motivou, e se disponibilizou a me ajudar na realização deste trabalho.

Ao meu coorientador professor Marcelo Filipe Nunes Henriques, pela disponibilidade, suporte, paciência, e motivação que me deu durante o desenvolvimento deste trabalho.

À Luciene Kapisch por ter me incentivado e dado o apoio para que eu desse esse passo tão importante.

À minha família que sempre me incentiva a buscar crescimento profissional e pessoal.

Um agradecimento especial ao professor José Carlos Reston Filho (*in memoriam*), por ter sido uma inspiração durante a realização do mestrado, e pelo exemplo de profissional e ser humano que foi.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Aplicação de simulação para a resolução de um problema de escalonamento da produção em máquinas paralelas e redução do tempo total de processamento

RESUMO

A competitividade do mercado faz com que a busca por métodos que promovam a eficiência da utilização dos recursos, melhoria de processos e rapidez na entrega de resultados, se torne um desafio cada vez maior entre as empresas. A utilização de métodos de melhoria para resolução de problemas de escalonamento da produção em máquinas paralelas tem ganhado foco na literatura devido a sua complexidade. A simulação computacional é uma ferramenta que demonstra eficiência e grande adaptabilidade, desde os processos mais simples, aos mais complexos.

Esta dissertação apresenta a modelagem e simulação de um problema de escalonamento em máquinas paralelas não relacionadas e tempos de *setup* dependentes, tendo como objetivo principal a redução do tempo total de processamento (*Makespan*).

Como referência para este trabalho, foram utilizados os cenários estudados por Rabadi, et al. (2014), que considera grandes instâncias de problema (até 12 máquinas e 120 trabalhos). Também foram utilizadas cinco diferentes regras de sequenciamento que através de simulação computacional, no *software* de simulação SIMIO, foi realizada a modelagem e simulação de todos os cenários propostos. Em seguida foi feita uma comparação com os valores para buscar o menor tempo de processamento total.

Os experimentos mostraram que a regra de sequenciamento que prioriza os trabalhos com o menor tempo de *setup* (*Shortest Setup Time – SST*), apresentou resultados satisfatórios para os cenários propostos neste trabalho, mostrando que através da simulação computacional é possível obter resultados que auxiliam na tomada de decisão de problemas de escalonamento da produção.

PALAVRAS-CHAVE

Máquinas paralelas, Escalonamento da produção, Tempos de *setup*, Simulação, *Software* SIMIO.

Simulation application to solve a production scheduling problem on parallel machines and reduction of total processing time

ABSTRACT

The competitiveness of the market makes the search for methods that promote the efficient use of resources, process optimization and speed in delivering results, become an increasing challenge among companies. The use of optimization methods to solve production scheduling problems on parallel machines has gained focus in the literature due to its complexity. Computational simulation is an optimization tool that shows efficiency and great adaptability, from the simplest to the most complex processes.

This dissertation presents the modeling and simulation of a scheduling problem on unrelated parallel machines and dependent setup times, having as main goal the reduction of the total processing time (Makespan).

As a reference for this work, the scenarios studied by Rabadi, et al. (2014) were used, which considers large problem instances (up to 12 machines and 120 jobs). Five different sequencing rules were also used that through computer simulation, in the simulation software SIMIO, the modeling and simulation of all the proposed scenarios was performed. Then a comparison was made with the values to search for the shortest total processing time.

The experiments showed that the sequencing rule that prioritizes the jobs with the shortest setup time (SST), presented satisfactory results for the scenarios proposed in this work, showing that through computer simulation it is possible to obtain results that help in the decision making process of production scheduling problems.

KEYWORDS

Parallel machines, Process scheduling, Setup times, Simulation, Software SIMIO.

ÍNDICE

Agradecimentos.....	iv
Resumo.....	vi
Abstract.....	vii
Índice.....	viii
Índice de Figuras.....	x
Índice de Tabelas.....	xii
Índice de Equações.....	xiv
Índice de Gráficos.....	xv
Lista de Abreviaturas, Siglas e Acrónimos.....	xvi
1. Introdução.....	1
1.1 Motivação.....	3
1.2 Objetivos.....	3
1.3 Estrutura.....	4
2. Revisão Bibliográfica.....	5
2.1 Escalonamento da Produção.....	5
2.1.1 Classificações dos problemas de escalonamento da produção.....	6
2.1.2 Complexidade dos problemas de escalonamento.....	10
2.2 Métodos de otimização.....	12
2.2.1 Métodos Exatos e Métodos Aproximados.....	13
2.2.2 Meta-Heurísticas.....	14
2.2.3 Regras de sequenciamento.....	15
2.3 Modelagem e simulação.....	18
2.3.1 Principais conceitos da simulação.....	20
2.3.2 Vantagens e desvantagens da simulação.....	21
2.4 Software de simulação SIMIO.....	23
2.4.1 Facility.....	25
2.4.2 Definitions.....	29

2.4.3	Experiments	31
3.	Estudo de caso	31
3.1	Tratamento de dados	32
3.1.1	Cenários, tempos de processamento e tempos de setup	33
3.1.2	Regras de sequenciamento propostas para o modelo	37
3.2	Construção do modelo de simulação	38
3.2.1	Desenho conceitual do problema	39
3.2.2	Criação de Elements, Properties, States, Tokens e Tables	40
3.2.3	Processo principal do modelo	49
3.2.4	Processo da leitura das matrizes de tempos.....	50
3.2.5	Processos das regras de sequenciamento.....	51
3.2.6	Processo de entrada e saída dos servidores (Server)	59
4.	Análise dos resultados	63
4.1	Definição dos cenários	63
4.2	Definição do número de replicações	64
4.3	Análise do ACO II com regras de sequenciamento	66
4.4	Análise entre as regras de sequenciamento	74
5.	Conclusão	82
5.1	Simulação.....	82
5.2	Regras de sequenciamento	82
5.3	Trabalhos futuros	83
	Referências Bibliográficas	84
	Apêndice 1 – Resultados de desvio percentual médio para 5 replicações.....	92
	Apêndice 2 – Resultados de desvio percentual médio para 10 replicações	93
	Apêndice 3 – Resultados de desvio percentual médio para 15 replicações	94
	Apêndice 4 – Resultados de desvio percentual médio para 20 replicações	95
	Apêndice 5 – Resultados de desvio percentual médio para 25 replicações	96

ÍNDICE DE FIGURAS

Figura 1 – Métodos de resolução de problemas de otimização combinatória	11
Figura 2 – Janela Facility para criação de um novo modelo	25
Figura 3 – Processo de construção de um modelo de simulação no software SIMIO	38
Figura 4 – Tela principal de simulação (Facility)	39
Figura 5 – Janela Definitions do modelo (Elements)	41
Figura 6 – Janela Definitions (Properties)	42
Figura 7 – Janela Definitions (States)	42
Figura 8 – Janela Definitions (Tokens).....	44
Figura 9 – Janela Definitions (States em ModelEntity)	46
Figura 10 – Janela Data (T_Dados).....	46
Figura 11 – Janela Data (T_Conexões).....	47
Figura 12 – Janela Data (T_Trabalhos)	47
Figura 13 – Janela Data (T_Servidores)	48
Figura 14 – Janela Processes (Processos do modelo)	48
Figura 15 – Processo de Inicialização (OnRunInicialized)	49
Figura 16 – Janela Definitions (States)	50
Figura 17 – Processo de escrever matrizes	51
Figura 18 – Processo de sequenciamento aleatório.....	52
Figura 19 – Processo prioriza maior tempo de processamento.....	53
Figura 20 – Processo prioriza menor tempo de processamento.....	55
Figura 21 – Processo prioriza maior tempo de setup.....	56
Figura 22 – Processo prioriza menor tempo de setup.....	58
Figura 23 – Processo de entrada no servidor	59
Figura 24 – Processo de entrada e saída dos servidores	60
Figura 25 – Processos realizados antes e depois do tempo de processamento	61
Figura 26 – Processo de saída do servidor	62
Figura 27 – Processo de execução do tempo de processamento	62
Figura 28 – Experimentos e replicações para os sequenciamentos.....	63
Figura 29 – Comparação de desvio percentual médio para 5 replicações	92

Figura 30 – Comparação de desvio percentual para 10 replicações	93
Figura 31 – Comparação de desvio percentual médio para 15 replicações	94
Figura 32 – Comparação de desvio percentual médio para 20 replicações	95
Figura 33 – Comparação de desvio percentual médio para 25 replicações	96

ÍNDICE DE TABELAS

Tabela 1 – Definições das Regras de Sequenciamento	16
Tabela 2 – Principais objetos da Standard Library	26
Tabela 3 – Steps comumente utilizadas, na construção da lógica por processo	28
Tabela 4 – Cenário para tempo de setup maior que tempo de processamento (Setup Dominant).....	33
Tabela 5 – Cenários para tempo de processamento e setup balanceados (Balanced)	35
Tabela 6 – Regras de sequenciamento propostas.....	37
Tabela 7– Descrição de variáveis States.....	43
Tabela 8 – Descrição de variáveis States do Token.....	45
Tabela 9 – Média total do desvio dos sequenciamentos (Balanced)	64
Tabela 10 – Média total do desvio dos sequenciamentos (Setup Dominant).....	64
Tabela 11 – Comparação de ACO II e Sequenciamento Balanced para 20 trabalhos	66
Tabela 12 – Comparação de ACO II e Sequenciamento Balanced para 40 trabalhos	66
Tabela 13– Comparação de ACO II e Sequenciamento Balanced para 60 trabalhos.....	67
Tabela 14 – Comparação de ACO II e Sequenciamento Balanced para 80 trabalhos	67
Tabela 15 – Comparação de ACO II e Sequenciamento Balanced para 100 trabalhos	67
Tabela 16 – Comparação de ACO II e Sequenciamento Balanced para 120 trabalhos	68
Tabela 17 – Comparação de ACO II e Sequenciamento Setup Dominant para 20 trabalhos	69
Tabela 18 – Comparação de ACO II e Sequenciamento Setup Dominant para 40 trabalhos	69
Tabela 19 – Comparação de ACO II e Sequenciamento Setup Dominant para 60 trabalhos	69
Tabela 20 – Comparação de ACO II e Sequenciamento Setup Dominant para 80 trabalhos	70
Tabela 21 – Comparação de ACO II e Sequenciamento Setup Dominant para 100 trabalhos	70
Tabela 22 – Comparação de ACO II e Sequenciamento Setup Dominant para 120 trabalhos	70
Tabela 23 – Comparação de ACO II e os sequenciamentos de características Balanced	72
Tabela 24 – Comparação de ACO II e os sequenciamentos de características Setup Dominant.....	73
Tabela 25 – Comparação entre os Sequenciamentos Balanced para 20 trabalhos.....	74
Tabela 26 – Comparação entre os Sequenciamentos Balanced para 40 trabalhos.....	75
Tabela 27 – Comparação entre os Sequenciamentos Balanced para 60 trabalhos.....	75
Tabela 28 – Comparação entre os Sequenciamentos Balanced para 80 trabalhos.....	75
Tabela 29 – Comparação entre os Sequenciamentos Balanced para 100 trabalhos.....	76
Tabela 30 – Comparação entre os Sequenciamentos Balanced para 120 trabalhos.....	76

Tabela 31 – Comparação entre os Sequenciamentos de Setup Dominant para 20 trabalhos.....	77
Tabela 32 – Comparação entre os Sequenciamentos de Setup Dominant para 40 trabalhos.....	77
Tabela 33 – Comparação entre os Sequenciamentos de Setup Dominant para 60 trabalhos.....	77
Tabela 34 – Comparação entre os Sequenciamentos de Setup Dominant para 80 trabalhos.....	78
Tabela 35 – Comparação entre os Sequenciamentos de Setup Dominant para 100 trabalhos	78
Tabela 36 – Comparação entre os Sequenciamentos de Setup Dominant para 120 trabalhos	78
Tabela 37– Comparação de tempos de processamento total entre os sequenciamentos (Balanced)...	80
Tabela 38– Comparação de tempos de processamento total entre os sequenciamentos (Setup Dominant)	81

ÍNDICE DE EQUAÇÕES

Equação 1- Cálculo do desvio médio dos tempos	64
---	----

ÍNDICE DE GRÁFICOS

Gráfico 1– Gráfico comparativo do sequenciamento (Balanced).....	65
Gráfico 2– Gráfico comparativo do sequenciamento (Setup Dominant)	65

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

ACO – *Ant Colony Optimization*

AG – Algoritmos Genéticos

API – *Application Programming Interface*

EDD – *Early Due Date*

EP – Escalonamento da produção

ERD – *Early Release Date*

FIFO – *First In First Out*

LIFO – *Last In First Out*

LPT – *Longest Processing Time*

LST – *Last Slack Time*

Meta-RaPS – *Metaheuristic for Randomized Priority Search*

NP-Complete – *Nondeterministic Polynomial Time Complete*

NP-Hard – *Nondeterministic Polynomial Time Hard*

OP – Ordens de produção

P – *Polynomial Time*

PFS – *Parallel Flow Shop*

PH – *Partitioning Heuristic*

RSA – *Restrictive Simulated Annealing*

SD – *Setup Dominant*

SIRO – *Service In Random Order*

SPT – *Shortest Processing Time*

SQ – *Shortest Queue*

SST – *Shortest Setup Time*

TS – *Tabu Search*

1. INTRODUÇÃO

No ambiente de negócios atual, onde as mudanças são constantes, as empresas precisam responder rapidamente à flutuação da demanda e gerenciar suas capacidades de maneira mais eficiente, (Monden, 2012). Uma das áreas mais problemáticas em vários setores industriais é o planejamento/escalonamento otimizados de linhas de produção.

Os métodos de escalonamento têm uma aplicação bastante alargada à resolução de uma grande variedade de problemas de escalonamento, quer na prática industrial para produção de bens, quer também para os serviços, (Varela, 2007).

Escalonamento é um processo decisório, regularmente empregado em indústrias e serviços, que lidam com alocação de recursos para execução de tarefa durante um determinado período de tempo e com objetivos específicos. As tarefas e os recursos podem tomar diferentes formas nas organizações, levando a diferentes restrições de escalonamento. Um escalonamento pode ainda ser feito com diferentes objetivos, dependendo da aplicação, (Pinedo, 2004). Problemas de escalonamento produtivo são amplamente estudados devido à sua complexidade científica e aplicabilidade na indústria.

Resolver um problema de escalonamento envolve a resolução de dois sub-problemas:

- Alocação: recursos que serão disponibilizados para a realização de cada tarefa;
- Sequenciamento: interessa saber em que instante a tarefa será realizada;

Uma classificação típica dos problemas de escalonamento é quanto ao número de máquinas a alocar: uma só máquina ou mais do que uma máquina. Os problemas que envolvem mais do que uma máquina podem ainda subdividir-se em: problemas de máquinas paralelas, e problemas de múltiplas operações.

A configuração em linhas de fabrico ou de produção paralelas (*Parallel Flow Shop*) pode ser definido simplesmente como a replicação de várias linhas de fabrico clássicas, i.e., ao invés de ter apenas uma linha clássica ou uma linha de fabrico simples (*Permutation Flow Shop*), poderá ter-se várias linhas em paralelo, e todas elas integrando um único sistema de produção, (Al-Salem, 2004).

Os benefícios que se podem ser obtidos com um bom escalonamento e sequenciamento de tarefas ao longo de uma linha de produção são:

- Melhor cumprimento dos prazos de entrega;
- Redução de *stock*;

- Aumento da produtividade utilizando os mesmos recursos;
- Redução de paragens improdutivas;
- Melhorar a informação ao cliente;

Para a manufatura em particular, a eficiência da utilização dos recursos é amplamente determinada pelo *makespan* da produção, (Portougal & Trietsch, 1998). No entanto, com a complexidade cada vez maior na organização da produção e a aleatoriedade na demanda, reduzir o *makespan* se tornou um grande desafio, (Zhao, et al., 2018). Minimizar o *makespan* geralmente implica em uma boa utilização dos recursos produtivos, tipicamente máquinas ou trabalhadores.

Makespan pode, por isso, ser definido como a diferença de tempo entre a primeira tarefa e aquela que determina o final de uma sequência de tarefas ou trabalhos. Isto é, o mesmo que dizer: o tempo a que a última tarefa de uma sequência de trabalhos está completa, (Blazewicz, et al., 2004).

Segundo Lustosa, et al. (2008), os problemas de sequenciamento de produção representam uma das decisões mais importantes abordadas pela programação da produção.

De acordo com Bartosiewicz (2014), os problemas de sequenciamento de produção determinam uma ordem (sequência) em que as tarefas (itens) – entre as disponíveis para o processamento – devem ser executadas em uma determinada máquina, e em que momento este sequenciamento deve ser realizado.

Muitas empresas ainda utilizam métodos de análise tradicionais, com funcionalidade e desempenho muito limitados. A exigência constante pela diminuição dos prazos de entrega faz com que métodos mais modernos sejam desenvolvidos, com o objetivo de tornar a entrega mais eficiente.

Devido a dificuldade de caracterização e de resolução de problemas mais complexos, e urgência nas respostas para a produção, surge a ferramenta de simulação que permite construir um modelo de simulação adaptado ao sistema produtivo específico, por forma a prever resultados, podendo ser adaptado para diversos cenários, permitindo melhores resultados no processo de escalonamento da linha de produção.

A simulação computacional refere-se aos métodos para estudar diversos sistemas reais ou abstratos, em modelos de avaliação numérica, recorrendo a *softwares* projetados para replicar um sistema de operações, com características específicas, normalmente por um período de tempo. De forma prática, significa que é um processo de criação de um modelo computadorizado, tendo como objetivo a realização de experimentos numéricos, de tal forma que proporcione uma compreensão desse sistema submetido a um determinado conjunto de condições, (Kelton, Sadowski & Sadowski, 2000).

A simulação faz o uso de conceitos e métodos que envolvem probabilidade e estatística, análise numérica e ferramentas de suporte lógico da informática, seja usando linguagens de programação genéricas, ou linguagens específicas de simulação. Poucos são os métodos que são flexíveis e populares como a simulação.

De fato, a simulação tem sido uma das ferramentas mais utilizadas na área manufatureira, uma vez que esta proporciona meios para identificar, analisar e melhorar os parâmetros de produção e de processo, tais como: análise de quantidade de maquinário e operadores; tempos de processo; avaliação de desempenho e de procedimentos operacionais, bem como avaliar o sequenciamento de produção e fluxo produtivos, proporcionando um conhecimento dos pontos em que o sistema possibilita uma maior flexibilidade de produção (Costa & Jungles, 2006), (Pontes, Yamada & Porto, 2007).

1.1 Motivação

A motivação para realização deste trabalho surgiu da importância e relevância que os métodos de simulação vêm tendo no cenário atual, mostrando sua eficácia na resolução de problemas escalonamento.

Diversos métodos heurísticos e meta-heurísticos vêm sendo aplicados para este tipo de problema, mas é possível encontrar na simulação um método mais prático, de resposta visual, e adaptável a diversos tipos de cenários, para a resolução de problemas num nível mais complexo.

A busca por métodos inovadores e mais próximos da realidade em termos de praticidade, adaptabilidade e rapidez que o mercado exige, se torna cada vez mais necessário. Diversos estudos vêm sendo realizados para que se consiga unir métodos computacionais com a complexidade de problemas de escalonamento da produção.

1.2 Objetivos

O objetivo principal dessa dissertação consiste no desenvolvimento através do *software* de simulação SIMIO, de um modelo de simulação para a resolução de escalonamento de um conjunto de trabalhos a serem executados, para um conjunto variável de dados como: números de trabalhos a serem executados, tempos de processamento, tempo de *setup* dependentes, para um determinado número de máquinas, dentro do sistema de máquinas paralelas.

Por meio da simulação, será feita uma reconstituição do comportamento de um sistema de máquinas paralelas, possibilitando assim simular diversos cenários alternativos, realizando testes que normalmente são difíceis de serem aplicados no próprio sistema físico.

Serão avaliados critérios para o sequenciamento no sistema de máquinas paralelas, identificando alternativas que possam conduzir a minimização do tempo total de processamento, o *Makespan*.

1.3 Estrutura

Esta dissertação é constituída de 5 capítulos, sendo que o primeiro composto pela introdução, motivação e os objetivos deste trabalho. O segundo capítulo apresenta a revisão bibliográfica, contextualizando o escalonamento, descrevendo suas classificações e a complexidade, como também a contextualização sobre os métodos de otimização e descreve a modelagem e a simulação, abordando em específico o *software* de simulação SIMIO, mostrando suas principais funções para a modelagem e simulação de processos.

No terceiro capítulo é apresentado o caso de estudo e a descrição de todo o processo de modelagem do problema objetivo deste trabalho. O quarto capítulo mostra os resultados obtidos através da simulação, comparando com estudos anteriormente realizados.

Por fim, as conclusões deste trabalho são apresentadas no capítulo 5, seguindo-se as referências bibliográficas, e os apêndices referentes aos resultados obtidos com a simulação.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo, a revisão bibliográfica aborda os principais temas deste trabalho que são: o escalonamento e sequenciamento da produção, métodos de otimização, modelagem e simulação.

O *software* escolhido para desenvolvimento do trabalho foi o *software* de simulação SIMIO, portanto, este capítulo descreve suas principais funções e descrições para a modelagem de um processo.

A revisão bibliográfica teve como fontes primárias: dissertações e teses, e como fonte secundária: jornais, artigos científicos, revistas científicas e livros.

2.1 Escalonamento da Produção

O escalonamento da produção (EP) é um processo de tomada de decisão usado regularmente em muitos setores de manufatura e serviços. Este processo lida com a alocação de recursos para a realização de um dado conjunto de trabalhos, que podem integrar uma ou mais tarefas ou operações, durante um determinado período de tempo, e seu objetivo é otimizar uma ou mais medidas de desempenho, garantindo que as tarefas sejam executadas no momento exato, com o recurso certo, (Pinedo, 2016). Portanto, pretende-se que a capacidade de cada recurso seja utilizada de forma eficiente, de modo a que se cumpram as ordens de fabrico e melhore o desempenho do sistema de produção subjacente ao problema de EP em causa.

Para que isso ocorra de maneira eficiente, é necessário que se faça a distribuição de tarefas pelos recursos disponíveis, e esse processo é a atividade de escalonamento (*Scheduling*).

Os sistemas produtivos envolvem muitas decisões a serem tomadas a nível de planeamento, especialmente nos casos em que existem um elevado número de recursos a serem geridos de forma a garantir não apenas o cumprimento de prazos de entrega, bem como a minimização de custos e tempos associados ao processo produtivo de um conjunto mais ou menos alargado de trabalhos.

Segundo Pinedo (2012), o sequenciamento da produção consiste num subprocesso de decisão do escalonamento, que procura fazer uso eficiente dos recursos de produção e assegurar a rápida execução dos trabalhos e suas tarefas, com vista a minimizar e ou maximizar um conjunto de critérios ou medidas de desempenho.

O escalonamento possui três elementos fundamentais:

- A designação dos recursos (máquinas em um chão de fábrica, pistas de aeroporto, processadores em um sistema de computação, caminhões de distribuição, etc.), que envolve a seleção de um conjunto de recursos apropriados, para uma atividade conhecida;
- O sequenciamento de atividades (operações em um processo de produção, pousos e decolagens em um aeroporto, execução de *softwares* em um sistema de computação, encomendas a serem entregues), que define a ordem de execução das atividades designadas para os recursos;
- A determinação do tempo de utilização dos recursos pelas respectivas atividades, ou seja, a determinação dos tempos de início e término de cada atividade.

O sequenciamento usualmente visa minimizar o tempo de conclusão de um conjunto de tarefas (*completion time*), o atraso (*tardness*), a conclusão adiantada das tarefas (*earliness*), o número total de tarefas atrasadas (*tardy jobs*) ou uma combinação destes objetivos. Tais objetivos podem ser ligados a um fator de priorização (*weight*), que assume a forma de um fator de importância ou um fator representativo de custo da tarefa, (Morton & Pentico, 1993), (Pinedo, 2008).

2.1.1 Classificações dos problemas de escalonamento da produção

Inúmeras pesquisas foram realizadas, devido ao grande número de aspectos a serem considerados entre os diversos tipos de problemas de escalonamento da produção (EP) existentes. Pinedo (2012), Mundim & Fuchigami (2017) são alguns dos autores que apresentam nomenclaturas de classificação de problemas de EP, seguindo uma estrutura geral do tipo $\alpha | \beta | \gamma$.

Os ambientes de produção (α) representam características relativas ao tipo de sistema de produção, bem como à quantidade de recursos disponíveis, e às configurações e fluxos de produção dos sistemas de produção. Os tipos de ambiente de produção mais conhecidos são:

- Ambiente de máquina única (1):** trata-se de um ambiente de produção simples que integra uma única máquina, dando origem a um problema de sequenciação de um conjunto de tarefas na máquina, sendo considerado como um caso particular de todos os outros ambientes mais complexos;
- Ambiente de máquinas paralelas:** trata-se de uma generalização do ambiente de máquina única, onde há um conjunto de recursos ou máquinas capazes de processarem qualquer trabalho ou tarefa. Podem ser classificadas como:

- Máquinas paralelas idênticas (Pm): ambiente de produção onde existem m máquinas idênticas e cada tarefa pode ser processada em qualquer uma das máquinas com a mesma velocidade;
 - Máquinas paralelas uniformes (Qm): várias máquinas podem executar as mesmas tarefas com velocidades de processamento diferentes, mas que seguem uma relação conhecida entre si;
 - Máquinas paralelas não relacionadas (Rm): generalização do ambiente anterior, no qual as velocidades de processamento são diferentes em cada máquina, não seguindo uma relação conhecida entre elas. Portanto, a velocidade de processamento depende da tarefa a ser executada.
- c) **Ambiente *Flow Shop*** (Fm): ambiente de produção que possui m máquinas especializadas em série ou sequência, onde cada um dos n trabalhos deve ser processado em todas as máquinas, seguindo a mesma sequência, isto é, uma tarefa não pode ser entregue à próxima máquina antes de todo o processamento na máquina corrente estar concluído. Depois de concluir a operação em uma máquina, as tarefas podem ou não entrar numa fila de espera antes de iniciarem seu processamento na máquina seguinte;
- d) **Ambiente *Flexible Flow Shop*** (FFc): é uma generalização do *Flow Shop* e do ambiente de máquinas paralelas, em que existem estágios de processamento em série, com um número de máquinas em paralelo em cada um. Todas as tarefas seguem o mesmo roteiro e devem passar por uma das máquinas de todos os estágios em sequência;
- e) **Ambiente *Job Shop*** (Jm): existem n tarefas que devem ser processadas por m máquinas. Cada tarefa é formada por uma sequência de operações consecutivas, havendo uma rota própria através das máquinas para cada tarefa. Portanto, o *Job Shop* é caracterizado por permitir diferentes fluxos de tarefas entre as máquinas e diferentes números de operações por tarefa;
- f) **Ambiente *Flexible Job Shop*** (FJc): é uma extensão do *Job Shop*, sendo composto por vários centros de trabalho. Cada um dos centros de trabalho existem várias máquinas capazes de executar a mesma operação. Portanto, existe uma rota própria para cada tarefa ao longo dos centros de trabalho, mas as operações são realizadas em apenas uma das máquinas de casa centro;

- g) **Open Shop** (Om): neste ambiente cada tarefa deve ser processada em várias máquinas, mas não necessariamente em todas elas. Além disso não existem roteiros de processamento pré-estabelecidos para as tarefas, isto é, não existe nenhuma sequência obrigatória de processamento das tarefas ao longo das máquinas.

As características de restrições de processamento (β) especificam as limitações no sistema de produção.

Neste campo pode-se incluir múltiplas entradas, Pinedo (2012) descreve da seguinte forma:

- a) **Release Dates** (rj): indica que as tarefas possuem uma data mínima a partir da qual sua execução pode ser iniciada. Portanto, nenhuma tarefa pode entrar em processo antes da sua data de liberação;
- b) **Preemptions** ($prmp$): significa que o processamento de uma tarefa pode ser interrompido e continuado mais tarde em qualquer máquina. Para esse caso, a quantidade de processamento que a tarefa interrompida já recebeu não é perdida. No caso de máquinas paralelas, tarefa interrompida não é obrigada a finalizar na mesma máquina que começou, e durante esse tempo de interrupção pode executar outra tarefa nessa máquina que ficou parada;
- c) **Precedence Constrains** ($prec$): indica que existe uma ordem de execução entre as tarefas, ou seja, uma determinada tarefa só pode iniciar o seu processamento após o término da tarefa anterior;
- d) **Sequence dependente setup times** (sjk): indica que o tempo de preparação das máquinas depende tanto da tarefa a ser processada quanto daquela que já foi processada imediatamente antes numa mesma máquina;
- e) **Job Families** ($fmls$): indica que os n trabalhos pertencem a determinadas famílias, portanto, os trabalhos que pertencem a mesma família podem ser processados em uma máquina, um após o outro, sem exigir nenhum tempo de *setup* (preparação) intermediário. No entanto, se a máquina alternar de uma família para outra, será necessário realizar *setup*. Esse tempo de *setup* pode depender de ambas as famílias e ser dependente da sequência, ou depender apenas da família que esta prestes a iniciar, ou não depender da família;
- f) **Batch Processing** ($batch$ (b)): indica que uma máquina pode processar vários trabalhos ao mesmo tempo, os trabalhos podem ter tempos de processamento diferentes e o trabalho só é finalizado quando todos forem concluídos;

- g) **Breakdown** (*brkdwn*): quando ocorre avarias na máquina, significa que pode não estar continuamente disponível. Esses períodos podem ser considerados fixos, como por exemplo no caso de turnos ou manutenção programada. Às vezes as quebras de máquinas também são chamadas de restrições de disponibilidade da máquina;
- h) **Machine eligibility restrictions** (*M*): em alguns problemas de máquinas paralelas, pode acontecer de nem todas as máquinas serem capazes de processar um determinado trabalho;
- i) **Permutation** (*prmu*): Essa restrição pode aparecer num ambiente *Flow Shop*, onde as filas para cada máquina operam de acordo com a sequência *First In First Out* (FIFO). Isso implica que a ordem de execução das tarefas (ou permutação) é mantida a mesma durante todo o sistema;
- j) **Blocking** (*block*): é um fenômeno que ocorre em ambientes *Flow Shop*, quando um trabalho que concluiu seu processamento em uma determinada máquina não pode sair da máquina se o trabalho anterior ainda não tiver concluído seu processamento na próxima máquina, dessa forma, o trabalho bloqueado também impede (ou bloqueia) o próximo trabalho de iniciar seu processamento na máquina em questão;
- k) **No-wait** (*nwt*): esse comportamento ocorre em um ambiente *Flow Shop*, onde os trabalhos não podem esperar entre duas máquinas sucessivas. Isso implica que o instante de início de um trabalho na primeira máquina deve ser atrasado para garantir que o trabalho possa passar pelo *Flow Shop* sem ter que esperar por nenhuma máquina;
- l) **Recirculação** (*rcrc*): pode ocorrer em um *Job Shop* ou em um *Job Shop* flexível, quando um trabalho pode visitar uma máquina ou centro de trabalho mais de uma vez.

O termo (γ), se refere ao objetivo a ser minimizado, que podem estar em função dos tempos de conclusão dos trabalhos, que obviamente, dependem do sequenciamento proposto, mas também, podem ser uma função das datas de vencimento, (Pinedo, 2012).

Pinedo (2016), Zarei, et al. (2016), Habiba, et al. (2018), Bender & Rabin (2002), Zhou, et al. (2016), descrevem esses critérios como:

- a) **Lateness** (*Lmax*): representa a diferença entre a data de término de uma tarefa e sua data de entrega, e é definido como positivo quando o trabalho é concluído com atraso efetivo e negativo quando é concluído mais cedo, relativamente à sua data de conclusão;

- b) *Maximum Lateness*: é definido como a maior violação de tempo de entrega entre todas as tarefas;
- c) *Tardness*: representa a maior diferença entre o instante de término de uma tarefa e a previsão de sua data de entrega ou conclusão, portanto, é uma diferença positiva no mesmo cálculo de *Lateness*;
- d) *Earliness*: representa o valor de adianto ou avanço na conclusão de um trabalho relativamente à sua data de conclusão;
- e) *Makespan (Cmax)*: mede o tempo total para a conclusão de um conjunto de trabalhos ou tarefas. É equivalente ao instante de tempo de conclusão do último trabalho ou tarefa, em uma sequência de produção. Um valor mínimo de *Makespan* normalmente significa uma boa utilização das máquinas do correspondente sistema de produção;
- f) *Maximum utilization*: indica a utilização máxima de recursos (ou máquinas). Uma vez aperfeiçoado este valor, para todas as máquinas de um sistema de produção, ocorre a redução de perdas de produção no correspondente sistema. Está muitas vezes vinculado ou considera uma medida de desempenho em conjunto com a minimização do *Makespan*, porém minimizar o *Makespan* não equivale a maximizar a utilização dos recursos;
- g) *Total weighted completion time* ($\sum w_j C_j$): é a soma ponderada dos instantes de término de cada trabalho ou tarefa, de acordo com um peso atribuído a cada uma delas.

O planeamento e sequenciamento podem ser atividades difíceis por um variado número de razões. Uma é que na realidade uma operação precisa lidar com variações nos tempos de preparação, tempos de processamento, interrupções, modificações no conjunto de tarefas a serem produzidas. Outra razão reside no facto de não haver nenhum método que identifique um planeamento ou sequenciamento ideal sendo virtualmente impossível a organização entre um vasto número de possibilidades de alternativas para a obtenção do valor ideal. Porém, a introdução das Tecnologias de Informação veio trazer novas possibilidades no planeamento, tornando a execução em tempo real uma possibilidade.

2.1.2 Complexidade dos problemas de escalonamento

O sequenciamento é uma técnica capaz de auxiliar os planeadores de produção ao tomar decisões rápidas enquanto negociam com seus clientes as datas de entrega, (Silva & Fuchigami, 2018). Os objetivos dessa técnica incluem minimização do tempo de conclusão das tarefas, dos tempos de

estoques intermediários ou do número de tarefas atrasadas, impondo regras de sequenciamento, que considerem os recursos disponíveis e restrições de processo, (Tavares, 2015).

O escalonamento da produção abarca dois tipos de problemas: problemas de decisão e problemas de otimização. Os problemas de decisão apresentam respostas binárias, enquanto os problemas de otimização têm respostas mais complexas que correspondem a uma solução que otimiza uma medida de desempenho.

Os problemas de escalonamento são muito complexos e de resolução computacionalmente difícil devido a sua diversidade, à natureza dinâmica e a sua dimensão. Normalmente são caracterizados por serem problemas de otimização, ou seja, problemas em que o objetivo é a minimização ou maximização de um ou mais critérios de otimização. Por norma, os problemas de escalonamento são primeiramente formulados através do problema de decisão, com o intuito de se obter respostas binárias de alocação de trabalhos numa determinada máquina do sistema, (Varela, 2007).

Os problemas de sequenciamento em ambientes industriais enquadram-se numa classe de problemas de otimização combinatória devido a sua elevada complexidade em termos computacionais. Um problema de otimização combinatória surge quando é necessário determinar as soluções ótimas em tempo considerado útil, respeitando um determinado conjunto de restrições, (Verissimo, 2016).

Schrijver (2003) descreve que em 1960 Edmonds defendeu a ideia de classificar o método como eficiente se seu tempo de execução estiver limitado por um polinómio (P), porém em 1970, Cook e Karp descobriram que vários outros problemas de otimização combinatória são mais difíceis, criando, assim, uma nova classe denominada tempo polinomial Não Determinístico (NP). Dessa forma, é possível dividir a eficiência dos métodos em três tipos, com mostrado na Figura 1, adaptado de Schrijver (2003) e Verissimo (2016).

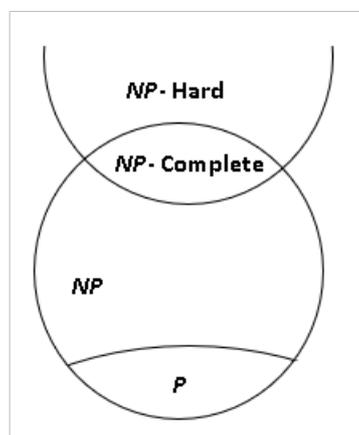


Figura 1 – Métodos de resolução de problemas de otimização combinatória

- **P**: conjunto de problemas com os quais existe um algoritmo eficiente que cresce em tempo polinomial;
- **NP-Hard (Nondeterministic Polynomial Time)**: é um tipo de problema para o qual não existem algoritmos eficientes que o resolva em tempo polinomial, ou seja, torna-se impossível testar todas as soluções possíveis em tempo computacional útil;
- **NP-Complete (Nondeterministic Polynomial Time Complete)**: Subconjunto de *NP*, em que caso seja encontrado um algoritmo polinomial, conseqüentemente existe um algoritmo polinomial para qualquer problema *NP*.

Xiao, et al. (2016) descreve que um problema de sequenciamento de tarefas em um sistema com várias máquinas, são tratados como problemas combinatórios complexos do tipo *NP-hard*, ou seja, requerem um tempo de processamento não polinomial (exponencial) para obtenção da solução ótima à medida que o número de máquinas ou tarefas é aumentado.

Para resolver esse tipo de problema é necessário conhecer todas as características que envolvem o sequenciamento estudado, tais como o ambiente de máquina, as restrições de processamento e o objetivo a ser minimizado.

Por conta disto, diversos algoritmos e heurísticas têm sido propostos para gerar soluções satisfatórias em tempos reduzidos de processamento computacional.

2.2 Métodos de otimização

A otimização matemática é uma área da ciência computacional que busca responder à pergunta “O que é melhor?” para problemas em que a qualidade de uma resposta pode ser medida por um número. Para resolver um problema é preciso primeiro formulá-lo, então a “função objetivo” representa cada ponto no espaço de soluções a um número real.

Um algoritmo de otimização é chamado determinístico se for possível prever todos os seus passos conhecendo seu ponto de partida, portanto, dada uma determinada entrada, sempre será produzida a mesma saída. A solução é dependente do ponto de partida fornecido e pode-se provar a convergência destes métodos para uma solução ótima, porém sem evidências de que ela é global, (Izmailov & Solodov, 2014).

Os algoritmos conhecidos como estocásticos ou probabilísticos tem como principal característica a busca pelo ideal através de regras de probabilidade trabalhando de maneira “aleatória”. Essas técnicas

ganharam popularidade com a evolução dos computadores, já que requerem um grande número de avaliações da função objetivo e das restrições, um exemplo desses algoritmos estocásticos são os Algoritmos Genéticos (AG), (Holland, 1973).

2.2.1 Métodos Exatos e Métodos Aproximados

Os métodos exatos de resolução de problemas de escalonamento compreendem os métodos eficientes, a programação matemática e os métodos de ramificação e limite (*Branch and Bound*), (Gomes, 2007).

Os métodos exatos devem ser aplicados em problemas de pequena dimensão, pois, para problemas de elevado grau de complexidade, a complexidade cresce exponencialmente com a dimensão do problema, pelo que estes algoritmos não conseguem encontrar um ideal global num intervalo de tempo útil, requerendo também um grande esforço computacional. Assim, para problemas de pequena dimensão, este tipo de método apresenta um bom desempenho. Os métodos exatos são particularmente importantes para a validação de eficácia obtida pelos procedimentos de aproximação, (Pinedo, 2008).

Pereira (2003) descreve que, para a obtenção de solução ótima de um problema de otimização combinatória, os métodos exatos baseiam-se em técnicas de enumeração exaustiva, onde pode-se considerar:

- O método de enumeração explícita, que consistem na pesquisa de todas as soluções do problema em causa;
- Os métodos de enumeração implícita: *Branch and Bound*, o método dos limites superior/inferior, a programação dinâmica, entre outros;
- As técnicas baseadas na estrutura do problema, através de formulações em programação inteira e posterior aplicação de métodos enumerativos.

Segundo Morton & Pentico (1993), os métodos de aproximação tentam resolver cada instância de um problema de forma aproximada, não garantem a obtenção da solução ótima, mas permitem encontrar soluções satisfatórias que em muitas situações são próximas da ótima. Esta classe de métodos pode ser aplicada a problemas de grande dimensão, uma vez que são rápidos e fáceis de implementar.

As heurísticas construtivas consistem na geração da solução utilizando regras ou procedimentos que determinam a ordem de processamento (sequência) das tarefas, (French, 1982). Esses métodos iniciam com uma solução vazia e expande iterativamente a solução atual até que uma solução completa seja construída, (Cui, et al., 2019).

Exemplos destas heurísticas são: Regras de sequenciamento (Pinedo, 2008), algoritmo *Shifting Bottleneck*, (Adams & Zawack, 1988), e o algoritmo *Giffler e Thompson*, (Baker, 1974).

Um algoritmo de pesquisa local (*Local Search*) consiste em partir de uma solução inicial, que pode ser gerada a partir de uma regra de sequenciamento. Depois é gerada a vizinhança dessa solução, sendo que o objetivo do algoritmo é escolher a melhor solução dentro da vizinhança e fazer dela a solução principal. A forma mais simples de escolher uma nova solução consiste em selecionar uma que melhore o valor da função objetivo. O processo se repete enquanto existir, dentro da vizinhança, uma solução que melhore a principal, (Marques, 2015).

2.2.2 Meta-Heurísticas

O propósito das meta-heurísticas é produzir uma solução para um problema complexo em um tempo admissível. A complexidade do problema de interesse torna possível a procura de cada solução possível ou combinação, pois, o objetivo é encontrar uma solução plausível num espaço de tempo aceitável, mas não há garantia de que a solução ótima seja encontrada. Portanto, o propósito é ter um algoritmo que permita atingir soluções com uma boa qualidade, ou seja, não há garantia de que a solução ótima seja atingida, mas as soluções encontradas são próximas desta, (Yang, 2010b).

Blum & Roli (2003) descrevem que os comportamentos das meta-heurísticas são avaliados pelos conceitos de intensidade e diversificação. O termo diversificação geralmente refere-se à exploração do espaço de procura, enquanto a intensidade é a exploração da experiência da pesquisa acumulada. Ou seja, diversificação significa a geração de diversas soluções como explorar o espaço de pesquisa em uma escala global, enquanto intensidade é a focalização da procura numa região local.

Reis (2020) cita, que de acordo com Gendreau & Potvin (2010), Glover & Kochenberger (2003), Osman (2011), as meta-heurísticas podem ser caracterizadas em grupos.

- 1) Algoritmos Evolucionários: Procedimentos que tentam reproduzir a evolução das espécies. Especificamente estes algoritmos simulam os processos biológicos que permitem que as gerações consecutivas em uma população se adaptem ao seu ambiente. O processo de adaptação é aplicado principalmente pela herança genética de pais para filhos e pela sobrevivência do mais apto. Exemplo desse tipo de algoritmo: Algoritmos genéticos;
- 2) Algoritmos baseados em memória: a principal característica destas heurísticas é o uso da memória adaptativa e exploração responsiva. O papel da memória adaptativa é impedir que a

pesquisa fique presa nas soluções ótimas locais e direcionar a pesquisa para processos de diversificação e intensidade mais efetivos. Exemplo desse tipo de algoritmo: *Tabu Search*;

- 3) Pesquisa de Vizinhança: Parte de uma solução candidata e depois move-se para uma solução vizinha. No espaço de pesquisa são definidas uma relação e estrutura de vizinhança;
- 4) *Swarm Intelligence*: São técnicas de inteligência artificial que estudam e simulam comportamentos coletivos e sistemas auto-organizados de exames de animais. Exemplo: *Ant Colony Optimization* e *Particle Swarm Optimization*;
- 5) Algoritmo baseado em probabilidades: Determinam se a solução atual é substituída por um novo ponto de avaliação com base numa probabilidade dependendo da diferença entre os seus valores da função. Exemplo: *Simulated Annealing*;
- 6) Métodos Híbridos: Existem muitas possibilidades para compor as heurísticas híbridas. O uso de métodos de pesquisa local dentro de uma meta-heurística é a maneira mais eficaz de superar a morosidade. Além disso, a inicialização múltipla de um método de pesquisa local é outro esquema que compõe as meta-heurísticas híbridas. Exemplo: *Memetic Algorithms*.

2.2.3 Regras de sequenciamento

Uma regra de sequenciamento é uma regra que prioriza todos os trabalhos que estão aguardando processamento em uma máquina. O esquema de priorização pode levar em conta os atributos das tarefas, os atributos das máquinas, bem como o tempo atual. Sempre que uma máquina é liberada, uma regra de sequenciamento seleciona entre os trabalhos em espera, o trabalho com a prioridade mais alta. Há décadas que pesquisas de regras de sequenciamento vem sido realizadas e muitas regras diferentes foram desenvolvidas e estudadas na literatura, (Pinedo, 2009).

A Tabela 1 mostra algumas regras de sequenciamento para a tomada de decisão num sequenciamento de tarefas, descritas por Pinedo (2016), Teixeira, et al. (2014), Barbosa, et al. (2010), Baker & Trietsch (2009).

Tabela 1 – Definições das Regras de Sequenciamento

Regras de sequenciamento	Descrição
SPT <i>(Shortest Processing Time)</i>	Prioriza os trabalhos na ordem crescente de seus tempos de processamento.
LPT <i>(Longest Processing Time)</i>	Prioriza os trabalhos na ordem decrescente de seus tempos de processamento.
FIFO <i>(First In First Out)</i>	Prioriza o pedido pela ordem de chegada. Neste sistema os primeiros a chegar são os primeiros a sair.
LIFO <i>(Last In First Out)</i>	Prioriza o pedido pela ordem de chegada. Neste sistema os últimos a chegar são os primeiros a sair.
SIRO <i>(Service In Random Order)</i>	A ordem de escolha do trabalho é feita de forma aleatória, considerando que todos os trabalhos na fila possuem a mesma probabilidade de serem selecionados.
ERD <i>(Early Release Date)</i>	Sequência os trabalhos a partir da sua hora de chegada. Essa sequência tenta minimizar os tempos de espera.
EDD <i>(Early Due Date)</i>	Organiza os trabalhos a partir de suas datas de vencimento crescentes. Esta regra funciona muito bem ao considerar critérios de desempenho baseados em atrasos, pois é capaz de minimizar o atraso máximo.
LST <i>(Last Slack Time)</i>	Prioriza medindo a urgência do trabalho pelo seu tempo de folga. Esse tempo é calculado pela diferença entre sua data de vencimento e seu tempo de processamento. Prioriza o trabalho com menor folga, pois representa maior urgência.

Regras de sequenciamento	Descrição
SQ <i>(Shortest Queue)</i>	Prioriza a primeira tarefa de uma sequência para a primeira máquina que está pronta para processá-la. Esse método minimiza a ociosidade das máquinas, garantindo que uma vez finalizado o trabalho, haja outro para iniciar o processamento.
SST <i>(Shortest Setup Time)</i>	Esta regra implica que sempre que um trabalho é concluído, o trabalho com menor tempo de <i>setup</i> é selecionado para ser executado em seguida (Pinedo, 2014).

As regras de sequenciamento são voltadas para a seleção da tarefa que terá prioridade de processamento, a partir das ordens de produção (OP) e/ou das condições do sistema produtivo, (Tubino, 2007).

As variações no tempo de chegada das tarefas classificam estas regras em dinâmicas (tarefas chegam em tempos diferentes) ou estáticas (tarefas chegam simultaneamente), (Moreira, 2011), (Silva & Fuchigami, 2018). As regras estáticas permitem uma visão completa no sistema produtivo, de modo que o sequenciamento busque encontrar uma tarefa adequada para um determinado recurso. Enquanto que, nas regras dinâmicas, o sequenciamento faz a mesma busca toda vez que uma nova tarefa é inserida no sistema, (Mendis, Ausdley & Indrusiak, 2017).

Os métodos de sequenciamento apresentados anteriormente são consideravelmente mais fáceis de implementar, mas permitem a obtenção de soluções razoáveis. Em problemas mais simples, podem até alcançar uma boa solução, por exemplo: a regra STP garante menor tempo de fluxo total e EDD menor atraso máximo. Porém, quando se trata de um problema mais complexo, as regras de sequenciamento só serão úteis se incorporadas a meta-heurísticas, as quais não garantem o resultado ideal, (Abreu & Pereira, 2019).

2.3 Modelagem e simulação

A simulação computacional refere-se aos métodos para estudar diversos modelos reais ou abstratos em sistemas de avaliação numérica utilizando-se de *software* específico, projetado para replicar um sistema de operações, com características específicas, normalmente por um determinado período de tempo. De forma prática, significa que é um processo de criação de um modelo computadorizado tendo como objetivo a realização de experimentos numéricos, de tal forma que proporcione uma compreensão desse sistema submetido a um determinado conjunto de condições, (Kelton, Sadowski & Sadowski, 2000).

Shannon (1998) define que a simulação é o processo de projetar um modelo de um sistema real, e realizar experimentos com esse modelo, com o objetivo de compreender o comportamento do sistema e /ou avaliar várias estratégias para a operação do sistema.

Banks (1999) diz que a simulação é um método experimental com modelação detalhada de um sistema real, onde, através da utilização de modelos visuais /ou animação gráfica, permite determinar como o sistema irá responder às mudanças na sua estrutura, ambiente ou pressupostos subjacentes.

Pidd (1998) declara que “Os métodos de simulação computacional têm se desenvolvido desde o início da década de ‘60, e podem ser uma das ferramentas analíticas mais comumente utilizadas no gerenciamento de sistemas. Os princípios básicos são suficientemente simples. O analista constrói um modelo de simulação do sistema de interesse, e cria os programas computacionais que incorporam o método, e usa o computador para replicar o comportamento do sistema quando sujeito a uma variedade de políticas operacionais. Desta maneira, a política mais desejável poderá ser selecionada.”

Como citado por Poça (2015), segundo Law & Kelton (1991) e Paiva (2005) os modelos de simulação podem ser classificados da seguinte forma:

- **Abertos versus fechados:** os modelos de simulação são classificados como abertos quando o meio envolvente afeta o comportamento do sistema. Por outro lado, os modelos de simulação são classificados como fechados quando não são afetados pelo meio envolvente, nem agem em função deste;
- **Estáticos versus dinâmicos:** os modelos de simulação são considerados como estáticos quando não envolvem a variável tempo e são considerados como dinâmicos quando o modelo tem em conta as alterações de estado do sistema ao longo do tempo de simulação, envolvendo a variável tempo;

- **Contínuos versus discretos:** os modelos de simulação contínuos alteram-se continuamente ao longo do tempo, ou seja, os valores das variáveis de estado podem mudar continuamente ao longo do tempo de simulação. De outra forma, os modelos de simulação discretos alteram-se em pontos específicos no tempo, isto é, os valores das variáveis de estado podem mudar em espaços temporais específicos no tempo de simulação;
- **Determinísticos versus estocásticos:** os modelos de simulação determinísticos não têm distribuições aleatórias, ou seja, os valores de entrada e saída são fixos. Nestes modelos os resultados serão sempre iguais, pois o comportamento do sistema está determinado. Por outro lado, nos modelos de simulação estocásticos, as distribuições devolvem resultados aleatórios, tendo uma base estatística.

A simulação tem apresentado crescente utilização em situações que envolvem eventos discretos, de forma a auxiliar a tomada de decisão, e é aplicável a qualquer tipo de sistema, uma vez que diversas metodologias de pesquisa em simulação têm sido desenvolvidas, comprovando sua eficácia na resolução de problemas, (Freitas Filho, 2008).

White, et al. (1999) define o modelo de simulação como a representação realizada através da necessidade de entender, mudar e gerenciar o sistema em estudo. Para que a simulação possua validade, esta deve ser estruturada de forma cuidadosa e possuir detalhes suficientes para que a tomada de decisão seja válida para a realidade estruturada.

Loch (2010) e Banks (1998) descrevem os passos para a construção do modelo de simulação, que são os seguintes:

- **Definição do problema:** para iniciar uma simulação, deve-se analisar quais as necessidades do sistema no qual o modelo será aplicado, assim como suas limitações;
- **Definição de objetivos e plano de projeto:** os objetivos representam qual é a meta que a simulação pretende alcançar, enquanto o plano de projeto é constituído por um cronograma de execução da proposta em conjunto com os recursos necessários em cada etapa da simulação;
- **Elaboração do modelo conceitual:** desenvolve-se por meio de equações matemáticas e conceitos lógicos que representem a estrutura do problema e, conseqüentemente, do sistema;
- **Coleta de dados:** os dados necessários para o modelo de simulação podem já existir ou não. No caso em que tais informações já são conhecidas, estas podem ser insuficientes. Nessas

circunstâncias, assim como no caso da não existência, uma nova coleta de dados do sistema será necessária;

- **Tradução do modelo:** nesta fase, o modelo descrito anteriormente é codificado e implementado na linguagem de programação, ou num dos *softwares* de simulação específico;
- **Verificação do modelo:** analisa-se o funcionamento do modelo. Nesta etapa, recomenda-se averiguar a execução do modelo elaborado;
- **Validação do modelo:** após a verificação, a validação do modelo investiga se os resultados encontrados representam adequadamente o esperado em um sistema em análise;
- **Projeto experimental:** nesta etapa são estabelecidos os cenários que serão simulados e o número de replicações de cada um deles;
- **Replicações e análises:** nesta etapa os testes são efetuados e as análises de desempenho de cada cenário são avaliadas estatisticamente;
- **Replicações adicionais:** caso os dados retirados da etapa anterior sejam insuficientes, novas replicações deverão ser realizadas e posteriormente analisadas.

2.3.1 Principais conceitos da simulação

Banks (1999) e Freitas (2008), descrevem os conceitos mais utilizados na simulação, que servem para a formulação no desenvolvimento de um modelo de simulação.

- **Entidades:** as entidades podem ser classificadas como dinâmicas ou estáticas. As entidades dinâmicas movem-se através do sistema, e.g., clientes; enquanto que as entidades estáticas têm a função de servir as outras entidades dinâmicas, e.g., máquinas. As entidades também poderão ser classificadas como temporárias quando entram, percorrem o modelo e abandonam o mesmo; ou permanentes, quando executam a sua função sem abandonar o modelo;
- **Variáveis:** a variável contém informações necessárias acerca do sistema num determinado instante de tempo. E.g., o número de pessoas, de peças ou de tarefas nas filas de espera são consideradas variáveis do modelo. As variáveis podem ser alteradas pelas entidades;
- **Atributos:** os atributos são características próprias que definem cada entidade no modelo, e.g., a idade ou a altura dos clientes;

- **Acontecimentos:** os acontecimentos ocorrem num determinado instante de tempo (discreto), e.g., a chegada de um cliente do modelo. Os acontecimentos podem alterar atributos ou variáveis do modelo;
- **Recursos:** os recursos são entidades estáticas que prestam serviço às entidades dinâmicas, podendo representar pessoas ou equipamentos. Neste sentido, o recurso é um “local” onde uma entidade fica a receber um serviço durante um determinado período de tempo;
- **Filas:** as filas são locais no modelo onde as entidades dinâmicas esperam enquanto aguardam por um recurso. Normalmente, as filas de espera adotam a estrutura FIFO, i.e., a primeira entidade a chegar à fila, será a primeira a ser processada;
- **Sistema:** um sistema é um conjunto de entidades que interagem entre si para a realização de um fim lógico;
- **Modelo:** um modelo é uma representação de um sistema real, ou abstrato, com o intuito de estudar e analisar esse mesmo sistema.

2.3.2 Vantagens e desvantagens da simulação

Para a maioria das empresas, os benefícios do uso da simulação vão além de simplesmente fornecer uma visão do futuro. Banks, et al. (1996), Saliby (1989), Pegden, et al. (1995), Law & Kelton (2000), detalha que a simulação possui as seguintes vantagens:

- A simulação permite testar todos os aspetos de uma alteração proposta ao sistema, sem obrigar a aquisição de recursos para conduzir esses testes. Isto é relevante, uma vez que após a realização de uma alteração, as correções poderão ser extremamente dispendiosas;
- O processo de modelagem evolutivo, pois, inicia-se um modelo simples e aumenta-se a sua complexidade conforme o modelo vai sendo ajustado ao sistema, real ou abstrato, identificando de maneira mais clara as particularidades do problema;
- A simulação permite acelerar ou desacelerar os fenómenos de forma que seja possível analisá-los. E.g., a simulação possibilita a análise de um turno inteiro em questão de minutos, ou o mesmo turno poderá ser analisado à medida que cada acontecimento ocorre, i.e., cada vez que há uma alteração ao estado do modelo;
- Geralmente procura-se saber o porquê de certos fenómenos ocorrem em um sistema real. Através da simulação é possível conseguir as respostas a essas questões, reconstituindo os

cenários, e fazendo um exame minucioso do modelo, por forma a determinar por qual razão esse fenômeno ocorre. Tal não é possível fazer num sistema real, uma vez que esse pode não ser controlado, ou mesmo visto, na sua totalidade. No caso de um sistema abstrato, este ainda não existe fisicamente;

- Uma das maiores vantagens do uso de *software* de simulação é que, uma vez desenvolvido um modelo de simulação, verificado e validado, este pode ser utilizado para explorar novas políticas, procedimentos operacionais ou métodos – sem os gastos associados com a implementação física, e a interrupção consequente do sistema real, para levar a cabo uma modificação. As modificações são incorporadas no modelo de simulação, podendo observar os efeitos dessas alterações em ambiente virtual, e não no sistema real;
- Com a evolução do *hardware* e do *software*, atualmente é possível ter uma solução muitas vezes em poucos minutos. Aliado ao alto nível de competitividade entre organizações do mercado atual, esse é um diferencial positivo que a simulação apresenta;
- Muitas pessoas ainda operam com a filosofia do uso de *layouts* computadorizados, escrever relatórios complexos para convencer os outros sobre a validade de um projeto. Em muitos casos, esses projetos são basados no pensamento de alguém sobre a maneira como o sistema opera, e não na análise. Os estudos de simulação ajudam a fornecer compreensão sobre como um sistema realmente funciona, em vez de indicar as previsões de alguém sobre como um sistema funcionará;
- Através de treinamentos, a equipe e seus membros poderão aprender como trabalhar melhor através de erros e acertos realizados na simulação, num processo interativo virtual;
- O custo típico de um estudo de simulação é substancialmente inferior a 1% do valor total gasto para a implementação de um projeto ou redesenho. Como o custo de uma mudança ou modificação de um sistema após a instalação é tão elevado, a simulação é um investimento inteligente.

Por outro lado, Banks (1998), Centeno & Carrillo (2001), Carson II (2004) citam as seguintes desvantagens do uso da simulação:

- Necessidade de treinamento especializado para desenvolver modelos de simulação. A técnica é aprendida e aperfeiçoada com o tempo, e precisa de experiência;
- Os resultados obtidos podem ser mal interpretados ou até mesmo difíceis de serem analisados;

- A fase de desenvolvimento do modelo de simulação, e da sua análise, pode consumir muito tempo e apresentar um custo elevado;
- Pode ser utilizada inadequadamente, e.g., quando uma solução analítica é exequível;
- Os resultados da simulação podem ser de difícil implementação, pois podem precisar de outros investimentos, alterações muito grandes do sistema físico, entre outros;
- A programação de um modelo de simulação pode se tornar altamente dispendiosa e desgastante se os recursos computacionais não forem os adequados.

Na realidade, se a solução analítica for relativamente simples, esta será sempre preferível à simulação, uma vez que considera o modelo geral. Contudo, existem muitos sistemas que geram problemas que não podem ser resolvidos de forma simples, então recorre-se à simulação. Por vezes, o sistema que se pretende analisar nem existe, i. e., sistema abstrato; caso em que a simulação ou o modelo qualitativo seria o ideal.

A simulação não substitui, por si só, outras formas de experimentação ou julgamento subjetivo, mas é uma solução alternativa conveniente quando o modelo é altamente complexo, ou o número de variáveis é elevado. De qualquer modo, o treinamento e a experiência, aliados a um conhecimento profundo dos fenómenos do sistema, são ingredientes essenciais para o sucesso dos modelos de simulação, (Mota, et al., 2017).

2.4 Software de simulação SIMIO

Pegden (2007) fundador da empresa, descreve que o *software* SIMIO é um *software* de simulação de acontecimentos discretos projetados em torno de princípios básicos de orientação a objetos. Permite construir e executar modelos 2D e 3D dinâmicos de uma ampla variedade de sistemas. Este permite que o utilizador visualize os seus modelos concebidos em execução, antes de implementar ou modificar o sistema real. É um sistema de modelagem relativamente novo (criado em 2006), sendo projetado para atingir três objetivos principais.

- **Simples para iniciantes:** permitindo construir modelos rapidamente, com a facilidade de arrastar e largar os objetos, tornando a modelagem mais fácil, enquanto fornece um novo paradigma orientado a objeto, que altera radicalmente a forma como os objetos são construídos e utilizados;

- **Poderoso para especialistas:** apesar de ser possível começar como iniciante, à medida que adquire experiência em modelagem, o utilizador poderá modelar sistemas maiores e mais complexos. Isto é possível através da elevada flexibilidade da ferramenta, vários paradigmas de programação, como: *Facility*, *Process*, *Data tables*, e utilizando a sua API (*Application Programming Interface*);
- **Adicionar modelos 3D:** a animação 3D faz parte do processo natural de modelagem no *software* SIMIO, permitindo ter uma experiência 3D mesmo para iniciantes. Adicionalmente, é também possível adicionar *layouts* ao modelo e adicionar modelos 3D concebidos externamente, e. g., através do *Google Warehouse* (SIMIO LLC, 2018).

Pegden & Sturrock (2014) referem que o *software* de simulação SIMIO oferece ao modelador dois modos de execução, nomeadamente, o modo interativo (*Facility*) que permite criar e visualizar o modelo de simulação para posterior validação e o modo experimental (*Experiments*), que permite definir uma ou mais propriedades a serem alteradas para posteriormente se verificar o impacto dessas alterações no desempenho do modelo.

Sturrock, et al. (2018) descreve que o *software* de simulação SIMIO é uma nova estrutura de modelagem baseada nos princípios básicos da modelagem orientada a objetos, sendo único das seguintes formas:

1. A estrutura do *software* SIMIO é uma estrutura de modelagem gráfica orientada a objetos – em oposição a apenas um conjunto de classes numa linguagem de programação orientada a objetos – que são úteis para modelagem de simulação. A estrutura de modelagem gráfica do *software* SIMIO suporta totalmente os princípios básicos da modelagem orientada a objetos, sem exigir habilidades de programação, para adicionar novos objetos ao sistema;
2. O *software* SIMIO suporta animação 3D como parte natural do processo de modelagem. Este fornece uma *interface* direta, como *Google Warehouse*, para facilitar a adição de símbolos 3D realistas em seu modelo;
3. A estrutura do *software* SIMIO é neutra em termos de domínio e permite a construção de objetos que suportam muitas áreas de aplicação diferentes. A modelagem por processos do *software* SIMIO possibilita a criação de objetos customizados, com comportamento complexo, sem a necessidade de dominar linguagens de programação (e.g., C#, ou Java);

4. A estrutura do *software* SIMIO suporta vários paradigmas de modelagem. A estrutura suporta a modelagem de sistemas discretos e contínuos por acontecimentos, por processos, orientado a objetos, dinâmica de sistemas, e *agent-based*, permitindo uma boa visualização em todas elas.

2.4.1 Facility

O *Facility* é a tela de visualização principal, e é nesta que se define o fluxo e a lógica principal do modelo de simulação, em termos de objeto e animação. A *interface* inicial do *software* SIMIO, de forma genérica, é composta pela janela *Facility* ao centro, as bibliotecas de objetos à esquerda, e o painel de navegação e o acesso às propriedades dos objetos selecionados à direita.

As bibliotecas disponíveis são *Standard Library*, *Flow Library* e *Project Library*. Sempre que a janela *Facility* de um modelo de simulação está ativada, o painel bibliotecas à esquerda exibe as bibliotecas que estão abertas e disponíveis para modelagem do modelo. Ver Figura 2.

A *Standard Library*, ou Biblioteca Padrão, é uma biblioteca de propósito geral, de objetos desenvolvidos pela empresa-mãe do *software* SIMIO, permitindo modelar várias funcionalidades de sistemas. A *Flow Library* é a biblioteca para modelagem de sistemas de líquidos, utilizando objetos específicos. A *Project Library* é a biblioteca de objetos do projeto atualmente aberto, que poderá conter vários modelos e objetos dentro de um só Projeto.

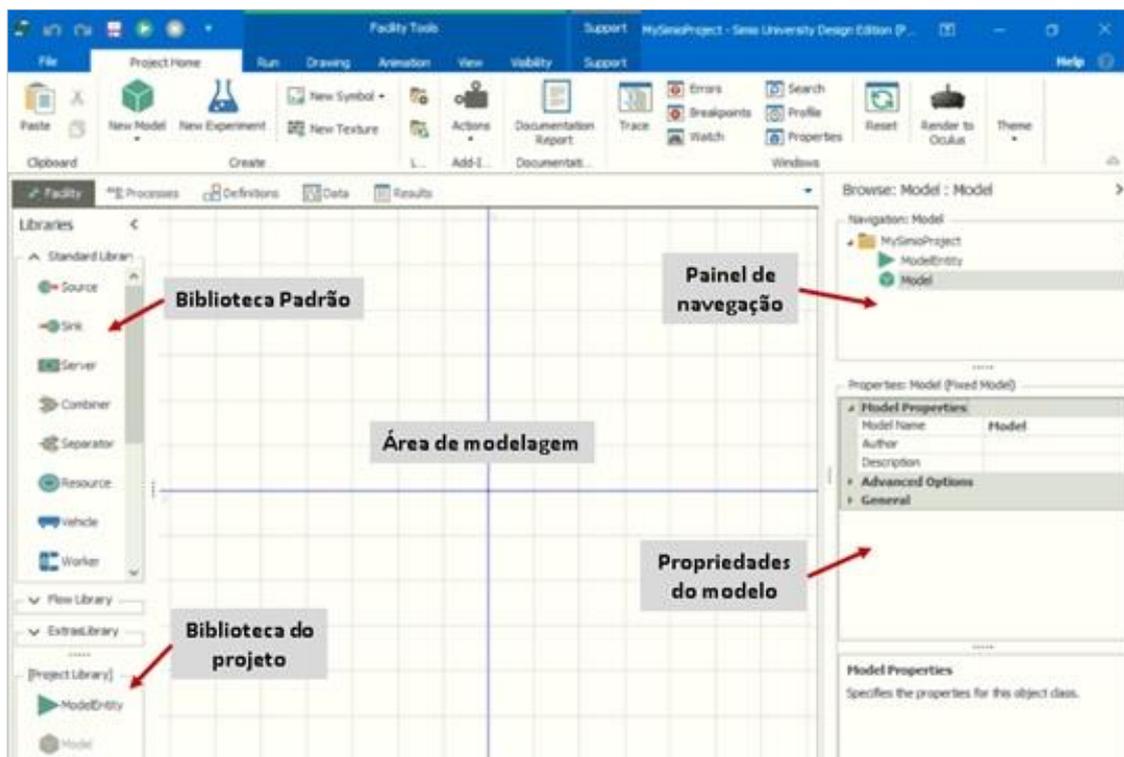


Figura 2 – Janela *Facility* para criação de um novo modelo

A biblioteca padrão do *software* SIMIO permite ao usuário iniciar a modelagem imediatamente, mas permite também que sejam construídos objetos personalizados. A Tabela 2 apresenta as principais funções:

Tabela 2 – Principais objetos da Standard Library (adaptado de SIMIO LLC (2018)).

Ícone	Objetos	Descrição
	<i>Source</i>	Permite definir a chegada de entidades no modelo.
	<i>Sink</i>	Permite a destruição das entidades que concluíram o seu fluxo no modelo.
	<i>Server</i>	Representa um processo capacitado, e.g., uma máquina, uma operação ou um serviço.
	<i>Combiner</i>	Junta uma ou mais entidades filho, com uma entidade pai, para formar um <i>Batch</i> , e.g., um palete; acumulando as mesmas funcionalidades do objeto <i>Server</i> .
	<i>Resource</i>	Um recurso genérico devendo ser ocupado e libertado para realização de uma operação por outros objetos, e.g., Entidades, mas não se movimenta.
	<i>Worker</i>	Um recurso e transportador, que se pode movimentar, devendo ser ocupado e libertado para a realização de uma operação ou transporte por outros objetos, e.g., Entidades.
	<i>Vehicle</i>	Transportador de entidades, de um ponto para outro, podendo seguir uma rota fixa, e.g., <i>milk run</i> ou ônibus, ou <i>on demand</i> , e.g., táxi.
	<i>Separator</i>	Separa um grupo de entidades formadas em <i>Batch</i> , copia entidades, ou cria várias entidades, mediante a entrada de uma só entidade, acumulando as mesmas funcionalidades do objeto <i>Server</i> .

Ícone	Objetos	Descrição
	<i>Connector</i>	<i>Link</i> entre dois pontos, com um tempo de deslocação igual a zero.
	<i>Path</i>	<i>Link</i> entre dois pontos, em que o tempo de deslocação será a relação da distância física entre os pontos (em metros) e a velocidade do objeto.
	<i>Time Path</i>	<i>Link</i> entre dois pontos, que tem um tempo de deslocação definido diretamente.
	<i>Conveyor</i>	<i>Link</i> entre dois pontos, em que a deslocação irá respeitar o método <i>push</i> ou <i>pull</i> .
	<i>Basic Node</i>	Modela uma interseção simples entre dois ou mais <i>links</i> .
	<i>Transfer Node</i>	Modela uma interseção complexa entre dois ou mais <i>links</i> , permitindo a transferência de Entidades para o mesmo.
	<i>Model Entity*</i>	Representa uma pessoa, componente, objeto, ou processo abstrato que é criado dinamicamente, flui pelo sistema e poderá ser destruída. *A <i>ModelEntity</i> , ou apenas <i>Entity</i> , faz parte da biblioteca do projeto, mas foi desenvolvida pela empresa do <i>software</i> SIMIO, e permite uma fácil edição dos seus atributos e lógica.

A Entidade (*Entity* ou *ModelEntity*) é um objeto, podendo ter o seu próprio comportamento, possuindo as suas próprias propriedades, tal como os restantes objetos do *software* SIMIO.

As Entidades podem ser criadas e destruídas dinamicamente, movimentar-se por uma rede de *links*, e *nodes*, e pelo espaço livre (*Free Space*), entrar e sair de objetos fixos (e.g., *Server*), ocupar e libertar recursos (*Worker* e *Resource*), ou ser transportadas (*Worker* e *Vehicle*).

O *Process (es)* ou Processo (s) permite a definição de lógica adicional ao modelo, utilizando um paradigma de fluxo de processo tradicional. Estes são criados em fluxogramas gráficos, sem a necessidade de programação com linguagem específica.

Um processo é uma sequência de etapas (apelidadas de *steps*), que é percorrido por um *token*, que poderá alterar o estado de um ou mais elementos, e.g., variável, ou atributo. À medida que o *token* percorre o processo, este executa as ações específicas em cada *step*.

Existem basicamente três tipos de processos no *software* SIMIO:

- **Standard Process:** é um processo definido e despotelado (*triggered*) pelo próprio *software* SIMIO, uma vez que é executado pelos próprios objetos do *software* SIMIO. Estes dispensam a necessidade de especificação de um acontecimento de disparo (*trigger*), uma vez que a lógica já foi desenvolvida pela empresa do *software*;
- **Add-on-Process:** é um processo criado pelo usuário que pode ser despotelado em instantes discretos pré-definidos, utilizando as propriedades dos objetos de simulação, ou executados através de outro processo. Estes permitem ao usuário definir a sua própria lógica de processo customizada no modelo de simulação, em instantes específicos, aumentando consideravelmente a flexibilidade de modelagem, estendendo a lógica pré-definida dos objetos de simulação;
- **Event-Triggered Process:** é um processo despotelado por um acontecimento (*event*) que ocorra discretamente no modelo, e.g., a chegada de uma entidade.

A componente lógica na criação do modelo de simulação no *software* SIMIO é feita nas janelas *Facility* e *Process*.

Para realizar a modelagem por processos é relevante conhecer as funcionalidades dos vários *steps* disponíveis, e como combiná-los no fluxo por processo. A Tabela 3 mostra os *steps* do processo que são comumente utilizadas, na construção de lógica adicional nos modelos de simulação.

Tabela 3 – *Steps* comumente utilizadas, na construção da lógica por processo (adaptado de SIMIO LLC (2018)).

Nome	Ação
<i>Assign</i>	Atribui um valor a uma, ou várias, variáveis ou atributos.
<i>Decide</i>	Permite uma bifurcação, baseada ou numa expressão, ou em probabilidade.
<i>Delay</i>	Realiza um atraso de tempo ao <i>token</i> .
<i>Execute</i>	Executa outro processo, permitindo a interoperabilidade entre processos diferentes.

Nome	Ação
<i>Find</i>	Examina uma expressão lógica, usando uma variável auxiliar, para encontrar o valor correspondente, mínimo, ou máximo, e.g., num vetor, matriz ou tabela.
<i>Search</i>	Pesquisa uma coleção de itens, incluindo linhas de tabela ou objetos em uma lista ou fila.
<i>Create</i>	Cria um novo objeto dinâmico ou criar uma cópia de um objeto dinâmico.
<i>Destroy</i>	Destrói objetos dinâmicos, incluindo entidades e transportadores.
<i>Wait</i>	Segura ao <i>token</i> no ponto do processo onde está colocado até o acontecimento (<i>event</i>) associado ser despoletado (<i>step Fire</i>). Permite coordenação entre dois ou mais pontos lógicos do modelo ou objeto.
<i>Release</i>	Libera a capacidade de um Recurso que foi ocupado.
<i>Transfer</i>	Transfere uma entidade dinâmica entre <i>nodes</i> , filas ou localizações.
<i>EndTransfer</i>	Conclui a transferência da entidade dinâmica ente <i>nodes</i> , filas, ou localizações, ordenada originalmente por um <i>step Transfer</i> .
<i>EndRun</i>	Ordena o término da simulação no instante atual da simulação.
<i>Notify</i>	Gera uma notificação ao usuário, ou através de um erro, aviso ou registro no <i>Model Trace</i> .

2.4.2 Definitions

O separador *Definitions* (Definição) possui cinco componentes principais: propriedades, estados, acontecimentos, visualização externa e lógica, com as seguintes especificidades.

- As propriedades (*Properties*) de um modelo são usadas para: apontar para propriedades de objetos que constam nesse modelo, ou fazer parte da lógica do próprio modelo. Estas permanecem estáticas durante a execução do modelo, mas permitem a sua especificação em diferentes cenários, ou ao colocar uma instância desse modelo, como objeto, num segundo modelo;

- Os estados (*States*) são definidos dentro de um objeto para guardar o valor de algo que pode mudar durante a execução do modelo. Existem duas categorias de estados: Discreto e Contínuo. Um estado discreto pode mudar por meio de uma atribuição apenas em instantes discretos no tempo. Um estado contínuo mudará contínua – e automaticamente quando a sua taxa ou aceleração for diferente de zero. Os *States* podem ser de vários tipos, sendo que os comumente usados são:
 - **Real (real)**: Um estado discreto que pode assumir qualquer valor numérico;
 - **Integer (inteiro)**: Um estado discreto que pode assumir apenas números inteiros;
 - **Boolean (booleano)**: Um estado discreto que pode assumir apenas ou o valor de *True* (Verdadeiro / 1) ou de *False* (Falso / 0);
 - **Date Time (tempo e data)**: Um estado discreto que pode ser atribuído a um valor em um formato *Date Time*;
 - **List (Lista)**: Um valor inteiro correspondente a uma das várias entradas em uma lista de *strings*;
 - **String**: Um estado discreto que pode ser atribuído a um valor de *string*, i.e., uma sequência de caracteres, e.g., “*Hello*”.
- Acontecimentos (*Events*) são ocorrências lógicas em um instante no tempo, e.g., uma entidade entrando em uma estação ou saindo de um nó. Os objetos podem definir e disparar seus próprios acontecimentos para notificar outros objetos de que algo ocorreu;
- A visão externa (*External view*) permite definir as componentes do modelo / objeto que serão visíveis quando este é instanciado noutro modelo;
- O *token* é um conceito abstrato, caracterizado aqui por ser um “representante” do objeto que executa os *steps* colocados num Processo. Um *token*, por sua vez, pode ter os seus próprios estados, definidos pelo usuário, que guardam informações passo a passo. Um *token* é criado no início de cada processo, e é destruído no final desse mesmo processo. À medida que o *token* se move pelo processo, este realiza as ações definidas pelo usuário em cada *step* do processo. Um único processo pode ter vários *tokens* ativos, movendo-se em paralelo, e um único objeto pode ter múltiplos *tokens* representando-o.

2.4.3 Experiments

A funcionalidade *Experiments* do *software* de simulação SIMIO permite executar o modelo de simulação construído para um número de replicações, especificado pelo usuário, onde o *software* SIMIO garante que as sementes da aleatoriedade geradas são diferentes, permitindo que as replicações sejam estatisticamente independentes umas das outras, uma vez que os números aleatórios subjacentes não se sobrepõem de uma replicação para a outra.

Existem duas formas básicas de executar experimentos no *software* SIMIO. A primeira é o modo Interativo, onde é possível assistir à execução do modelo animado e visualizar as informações dinâmicas do modelo, e.g., gráficos que estão de acordo com o comportamento do modelo de simulação. Esta forma é útil para construir e validar o modelo, bem como obter uma visão geral de como o sistema funcionará. Após a verificação e validação do modelo de simulação, a próxima etapa é definir os cenários específicos para testar com esse modelo. Neste caso, não há o interesse na animação, mas sim em replicar cada cenário para levar em conta a variabilidade subjacente no sistema e chegar a conclusões estatisticamente válidas do modelo.

3. ESTUDO DE CASO

Rabadi, et al. (2010) abordou um problema de escalonamento de máquinas paralelas não preemptivas e não relacionadas, com tempo de *setup* dependentes da máquina e da sequência. Para esse problema considera-se que todos os trabalhos estão disponíveis no tempo zero, todos os tempos são determinísticos e o objetivo é minimizar o *Makespan*.

Um algoritmo *Ant Colony Optimization* (ACO) foi apresentado e aplicado para este problema caracterizado como *NP-hard*.

As combinações foram em particular 2 máquinas e (40, 60, 80, 100, 120) trabalhos; 4 máquinas e (60, 80, 100, 120) trabalhos; 6 máquinas e (100, 120) trabalhos e 8 máquinas e 120 trabalhos. Os tempos de processamento e *setup* foram balanceados, gerados aleatoriamente a partir de uma distribuição uniforme $U(50, 100)$, (*Scheduling Research*, 2005).

Seu desempenho foi avaliado e comparado com outras heurísticas existentes: *Partitioning Heuristic* (PH) de Al-Salem (2004), *Tabu Search* (TS) de Helal, et al. (2006) e *Metaheuristic for Randomized Priority Search* (Meta-RaPS) de Rabadi, et al. (2006), e os testes provaram a superioridade do *Ant Colony Optimization* (ACO).

Rabadi, et al. (2014) fez uma extensão do trabalho anterior, testou o ACO sob processamento ou tempos de *setup* dominante, onde as distribuições são diferentes e a distribuição dominante possui valores mais altos.

O estudo foi realizado utilizando algoritmos matemáticos ACO aprimorado, no qual chamou de ACO II, tendo como referência os dados encontrados em Helal, et al. (2006), Rabadi, et al. (2006), Rabadi, et al. (2010), e Ying, et al. (2010), onde os tempos de processamento e *setup* foram gerados aleatoriamente a partir de duas distribuições uniformes: U (50,100) e U (125, 175), para grandes instâncias, ou seja, (2, 4, 6, 8, 10, 12) máquinas e (20, 40, 60, 80, 100, 120) trabalhos.

Os resultados obtidos com o algoritmo ACO II foram comparados com o TS de Helal, et al. (2006), Meta-RaPS de Rabadi, et al. (2006), ACO I de Rabadi, et al. (2010), *Simulated Annealing* (SA) e *Restrictive Simulated Annealing* (RSA) de Ying, et al. (2010). Os testes comprovaram a superioridade do ACO II.

Diversos autores utilizam abordagens meta heurísticas para análise de problema escalonamento de máquinas paralelas não relacionadas. Nos últimos anos, esse tipo de problema vem recebendo maior atenção devido a sua complexidade e importância.

Para este trabalho, propõe-se realizar a análise de um problema de escalonamento de máquinas paralelas não relacionadas com tempo de *setup* dependentes da sequência para minimizar o tempo total de processamento (*Makespan*), através de simulação computacional, utilizando o *software* de simulação SIMIO, considerando os cenários e valores de tempo de processamento e *setup* utilizados por Rabadi, et al. (2014).

A análise estará voltada para duas condições principais: tempo de processamento e tempo de *setup* balanceados, e tempo de *setup* dominante. O objetivo é minimizar o *Makespan*, portanto, serão utilizadas regras de sequenciamento e, ao final, será feita uma comparação com os valores obtidos de tempo de processamento por Rabadi, et al. (2014), com o ACO II.

3.1 Tratamento de dados

Para a construção do modelo, foi feita a recolha dos dados de tempo de processamento e tempo de *setup* através do *Scheduling Research* (2005), usado por Rabadi, et al. (2014). Realizou-se a padronização dos dados para se adaptassem ao *software* SIMIO, para que posteriormente essas informações fossem transferidas para o *software* e assim realizada a modelação.

3.1.1 Cenários, tempos de processamento e tempos de setup

Duas condições foram escolhidas para serem analisadas: 1) tempos de processamento e tempos de *setup* balanceados (*Balanced*) e, 2) tempos de *setup* dominante ao tempo de processamento, ou seja, os tempos de *setup* são maiores que os tempos de processamento (*Setup Dominant*).

Para cada um desses cenários, foram gerados aleatoriamente os tempos de processamento e tempos de *setup*, a partir das duas distribuições uniformes: U (50,100) e U (125,175), e cada combinação de máquina foi testada com 15 instâncias do problema, (Rabadi, et al., 2014). Portanto:

- **Balanced:** Tempos de processamento e tempos de *setup* [U (50, 100)];
- **Setup Dominant:** os tempos de processamento [U (50, 100)] e tempo de *setup* [U (125, 175)].

A variação da quantidade de máquina é (2, 4, 6, 8, 10, 12) e a quantidade de trabalhos é de (20, 40, 60, 80, 100, 120), portanto são no total de 72 cenários, sendo 36 com características *Setup Dominant*, como mostrado na Tabela 4, e outros 36 cenários com características *Balanced*, como mostrado na Tabela 5. Considerando a quantidade de máquina e o tipo de cenário, foi determinado também um número de conexão, dividindo então todos esses cenários em 12 conjuntos.

Para cada conexão, foi gerada uma planilha de cálculo com todos os tempos de processamento e tempos de *setup*, considerando as 15 instâncias de cada um dos cenários.

Tabela 4 – Cenário para tempo de *setup* maior que tempo de processamento (*Setup Dominant*)

Setup Dominant			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
2	20	2m20t_SD	1
	40	2m40t_SD	
	60	2m60t_SD	
	80	2m80t_SD	
	100	2m100t_SD	
	120	2m120t_SD	
4	20	4m20t_SD	2
	40	4m40t_SD	

Setup Dominant			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
	60	4m60t_SD	
	80	4m80t_SD	
	100	4m100t_SD	
	120	4m120t_SD	
6	20	6m20t_SD	3
	40	6m40t_SD	
	60	6m60t_SD	
	80	6m80t_SD	
	100	6m100t_SD	
	120	6m120t_SD	
8	20	8m20t_SD	4
	40	8m40t_SD	
	60	8m60t_SD	
	80	8m80t_SD	
	100	8m100t_SD	
	120	8m120t_SD	
10	20	10m20t_SD	5
	40	10m40t_SD	
	60	10m60t_SD	
	80	10m80t_SD	
	100	10m100t_SD	
	120	10m120t_SD	

Setup Dominant			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
12	20	12m20t_SD	6
	40	12m40t_SD	
	60	12m60t_SD	
	80	12m80t_SD	
	100	12m100t_SD	
	120	12m120t_SD	

Tabela 5 – Cenários para tempo de processamento e *setup* balanceados (*Balanced*)

Balanced			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
2	20	2m20t_Balanced	7
	40	2m40t_Balanced	
	60	2m60t_Balanced	
	80	2m80t_Balanced	
	100	2m100t_Balanced	
	120	2m120t_Balanced	
4	20	4m20t_Balanced	8
	40	4m40t_Balanced	
	60	4m60t_Balanced	
	80	4m80t_Balanced	
	100	4m100t_Balanced	
	120	4m120t_Balanced	

Balanced			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
6	20	6m20t_Balanced	9
	40	6m40t_Balanced	
	60	6m60t_Balanced	
	80	6m80t_Balanced	
	100	6m100t_Balanced	
	120	6m120t_Balanced	
8	20	8m20t_Balanced	10
	40	8m40t_Balanced	
	60	8m60t_Balanced	
	80	8m80t_Balanced	
	100	8m100t_Balanced	
	120	8m120t_Balanced	
10	20	10m20t_Balanced	11
	40	10m40t_Balanced	
	60	10m60t_Balanced	
	80	10m80t_Balanced	
	100	10m100t_Balanced	
	120	10m120t_Balanced	
12	20	12m20t_Balanced	12
	40	12m40t_Balanced	
	60	12m60t_Balanced	
	80	12m80t_Balanced	

Balanced			
Nº Máquinas	Nº Trabalhos	Cenário	Conexão
	100	12m100t_Balanced	
	120	12m120t_Balanced	

3.1.2 Regras de sequenciamento propostas para o modelo

Para a resolução desse problema de escalonamento, foram definidas regras de sequenciamento para os trabalhos e alocação nas máquinas. As propostas se baseiam nas regras conhecidas SPT (*Shortest Processing Time*), LPT (*Longest Processing Time*), SIRO (*Service In Random Order*) e SQ (*Shortest Queue*).

Para os tempos de *setup*, as propostas baseiam-se na regra *Shortest Setup Time* (SST) e alguns detalhes são adaptações baseadas nas heurísticas apresentadas por Gupta & Tunc (1994) e Li (1997).

As regras de sequenciamento propostas para a construção da análise desse problema estão descritas na Tabela 6, a seguir.

Tabela 6 – Regras de sequenciamento propostas

Índice	Regras de sequenciamento
S0	Sequência das tarefas realizada de forma aleatória.
S1	Sequência das tarefas pela ordem decrescente do valor de tempo de processamento.
S2	Sequência das tarefas pela ordem crescente do valor de tempo de processamento.
S3	Sequência de tarefas pela ordem decrescente do tempo de <i>setup</i> .
S4	Sequência de tarefas pela ordem crescente do tempo de <i>setup</i> .

Nos modelos de sequenciamento 3 e 4, o tempo de *setup* depende tanto da tarefa a ser processada como também da tarefa que foi executada imediatamente antes na mesma máquina, por isso chamado dependente da sequência.

Outros parâmetros também devem ser considerados:

- m é o número de máquina;
- não há preempção durante o sequenciamento, ou seja, as tarefas não podem ser interrompidas durante a execução do modelo;
- cada máquina processa um trabalho por vez;
- no instante zero, todos os trabalhos estão disponíveis;
- no instante zero, todas as máquinas estão disponíveis;
- cada trabalho possui um tempo de processamento dependente da máquina;
- cada máquina possui um tempo de *setup* dependente do trabalho.

3.2 Construção do modelo de simulação

O processo de construção do modelo de simulação no *software* SIMIO não é estritamente sequencial, e muitas vezes se torna iterativo, (Sturrock, et al., 2018). A Figura 3 mostra o fluxograma para a construção do modelo de simulação no *software* SIMIO.

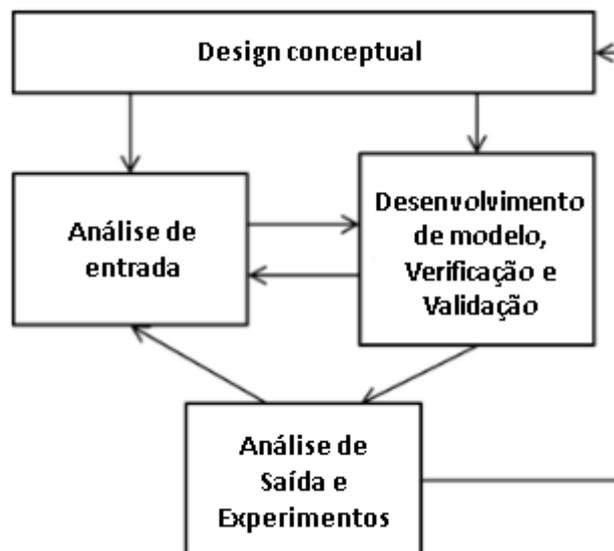


Figura 3 – Processo de construção de um modelo de simulação no *software* SIMIO

O *design* conceitual do modelo no *software* SIMIO pode ser realizado de forma muito simples, onde o foco principal é verificar os detalhes do problema e as possíveis abordagens para a modelagem. A análise de entrada envolve a caracterização das entradas do sistema e, em seguida, o desenvolvimento do modelo para gerar observações sobre as variáveis e processos aleatórios de entrada.

Todos os dados de tempo de processamento e tempo de *setup* foram acessados em uma planilha de cálculo, anteriormente criado de todas as 15 instâncias dos cenários, e foram criados processos para a simulação de cada um dos 5 tipos de sequenciamentos sugeridos anteriormente na Tabela 6, para então analisar o comportamento dos diferentes cenários.

O desenvolvimento do modelo de simulação é o processo de codificação pelo qual o modelo conceitual é convertido em um modelo de simulação executável. Os *steps* de verificação e validação garantem que o modelo cumpra com o que é pretendido.

3.2.1 Desenho conceitual do problema

Na primeira etapa de construção do modelo de simulação, mostrado na janela principal *Facility*, como mostra a Figura 4, foi construído o cenário geral base para a quantidade máxima de máquinas que se deseja realizar o experimento. Para este modelo de simulação, foi utilizado o objeto *Server*, da *Standard Library*, que representa as várias máquinas, totalizando 12 máquinas. Utilizou-se também a *Model Entity*, da *Project Library*, que irá representar os trabalhos.

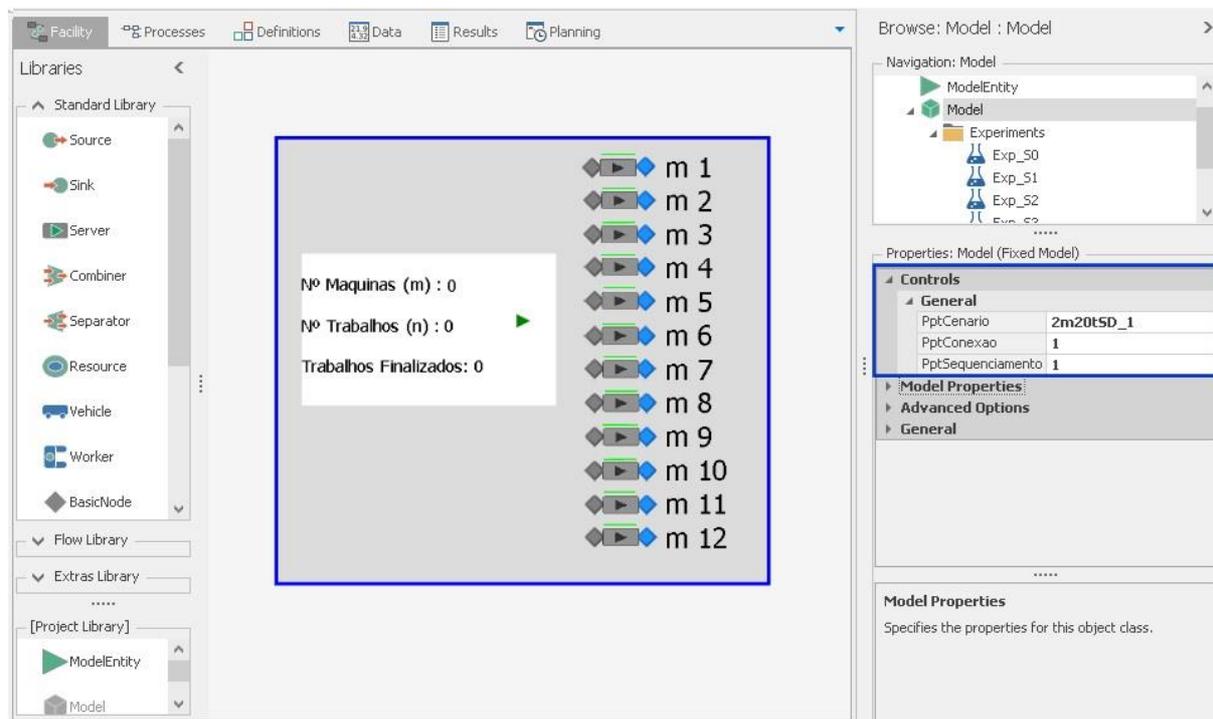


Figura 4 – Tela principal de simulação (*Facility*)

O modelo tem a capacidade de simular um dos cenários anteriormente detalhados, individualmente. Foram criados 3 tipos de propriedades, que determinam qual o cenário a ser executado, qual a conexão a estabelecer, e qual o tipo de sequenciamento que será utilizado, que são: *PptCenário*, *PptConexão*, *PptSequenciamento*, respectivamente, de acordo com o que o usuário escolher, como pode ser visualizado no lado direito da Figura 4.

3.2.2 Criação de Elements, Properties, States, Tokens e Tables

A janela *Definitions*, apresenta todos os *Elements*, *Properties*, *States*, e *Token* que foram definidos para a execução dos processos.

Em *Elements*, foram incluídos 12 *ExcelConnect*, em que cada um representa uma conexão, pois cada cenário possui 15 instâncias, que foram organizadas de acordo com a quantidade de máquina, quantidade de trabalho e característica dos tempos de processamento e *setup*.

Dessa forma é feita a ligação das 12 planilhas de cálculo com os dados principais para alimentar o modelo de simulação, os quais serão lidos na execução do processo, que é despoletado no início de cada execução da simulação (*simulation run*).

A escolha dessa função se torna mais viável, uma vez que o *software* SIMIO não deve ser utilizado como banco de dados, então desta forma o arquivo *CSV é acessado de acordo com o cenário escolhido para executar a simulação. Além disto, ao colocar os dados numa planilha de cálculo, é facilitado o acesso e manipulação destes dados por pessoas que não sejam especialistas em simulação ou não possuam o *software* de simulação SIMIO. A Figura 5 mostra a conexão entre o *software* SIMIO e as planilhas de cálculo.

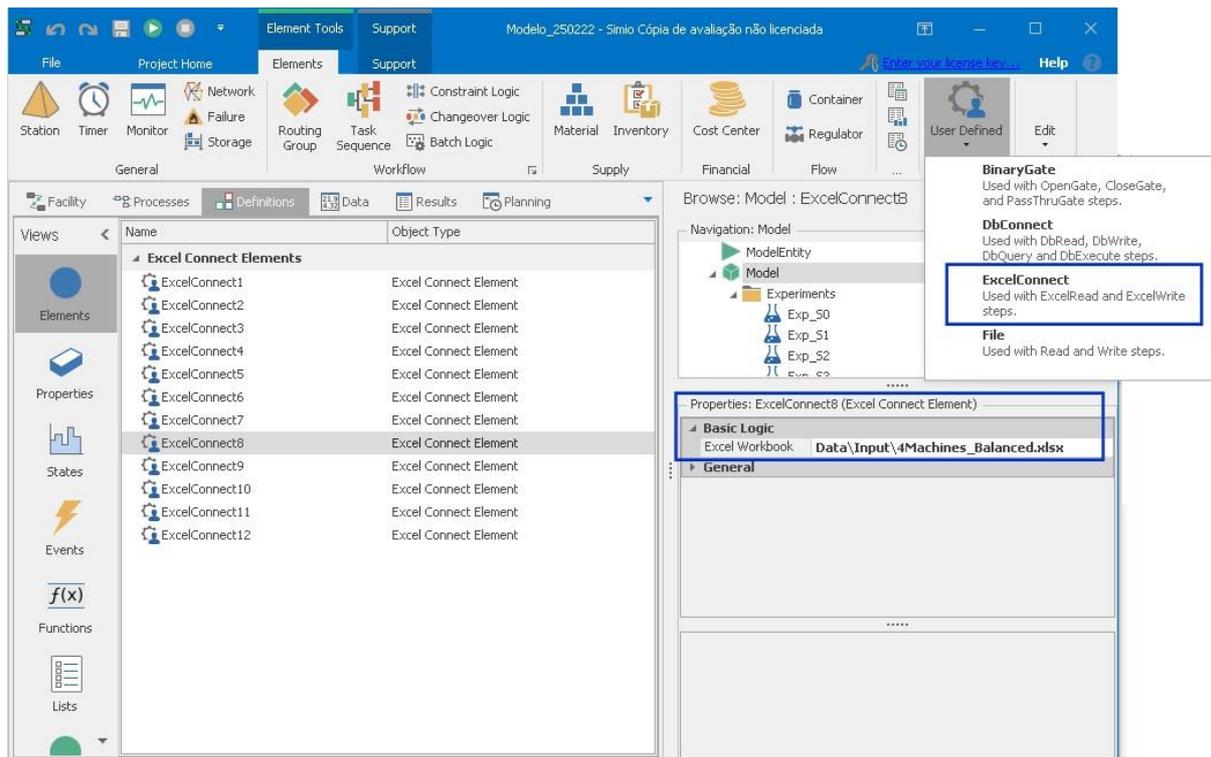


Figura 5 – Janela *Definitions* do modelo (*Elements*)

Como mencionado em anteriormente, foram criadas as 3 propriedades gerais para funcionamento do modelo, essas propriedades foram criadas na janela *Definitions* (em *Properties*), e são fundamentais para o funcionamento deste modelo de simulação:

- *PptCenário* é do tipo *String Property*, então, trata-se de um valor de texto que aceita letras e números, por exemplo, “**2m20t_SD_1**”, que significa que são 2 máquinas, 20 trabalhos, de característica *Setup Dominant*, da instância 1;
- *PptConexão* e *PptSequenciamento* são do tipo *Integer Property*, que é um valor numérico inteiro. Necessário identificar a qual conexão o cenário pertence (1a12) – ver Figura 6 e Figura 7 – e qual tipo de sequenciamento é desejado analisar (0 a 5) – ver Figura 6.

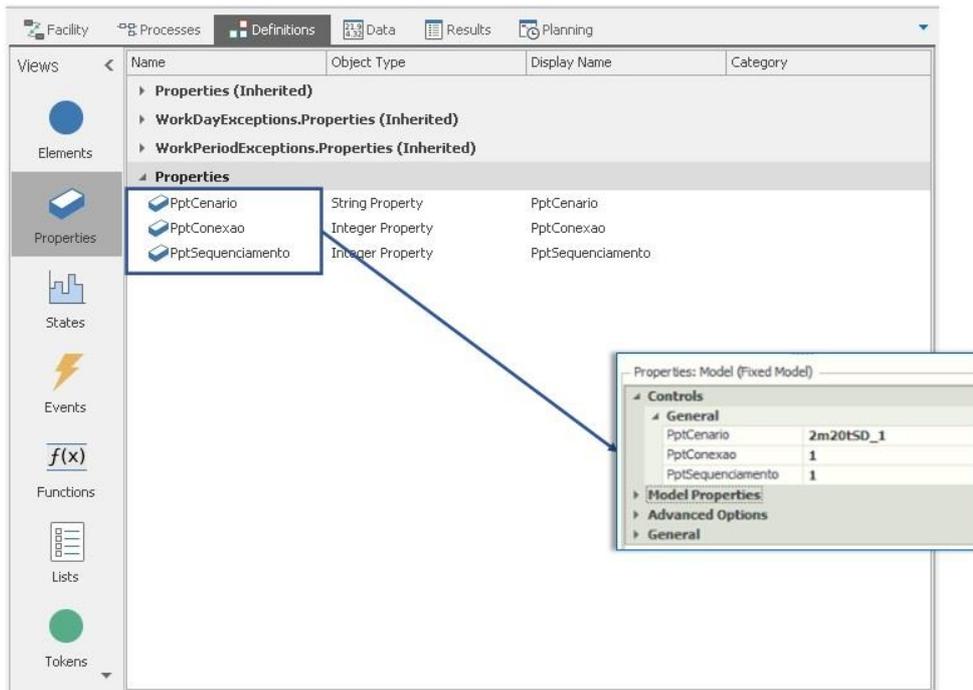


Figura 6 – Janela *Definitions (Properties)*

A Figura 7, mostra os *States* que foram criados para o modelo, e que serão utilizados e alterados no decorrer da simulação.

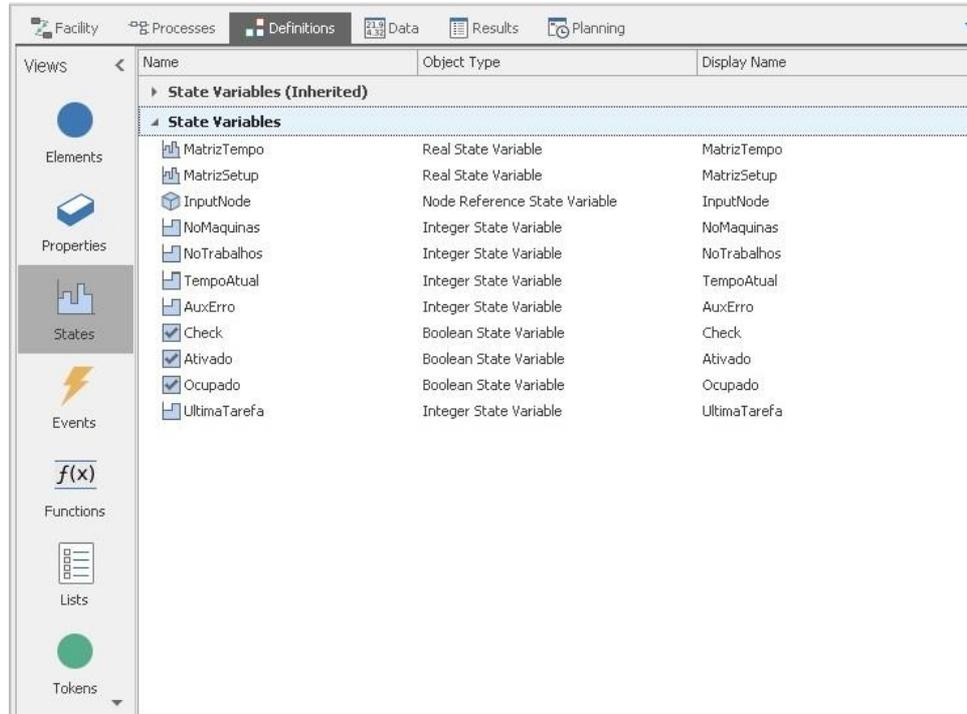


Figura 7 – Janela *Definitions (States)*

A Tabela 7 descreve as variáveis *State* definidas no modelo, indicando o que cada uma representa no mesmo.

Tabela 7 – Descrição de variáveis *States*

Variáveis	Tipo	Descrição
NoMaquinas	<i>Integer State</i>	Representa o número de máquinas.
NoTrabalhos	<i>Integer State</i>	Representa o número de trabalhos.
AuxErro	<i>Integer State</i>	Usada no processo Inicialização, para reconhecer se os dados de conexão e cenário estão corretos.
TempoAtual	<i>Integer State</i>	Usado nos processos Sequenciamento, para atualização e comparação dos valores de tempo.
UltimaTarefa	<i>Integer State</i>	Usada no processo “Máquinas”, para reconhecer o último trabalho executado na máquina.
MatrizTempo	<i>Real State</i>	Usado nos processos Matriz de Tempos, Sequenciamento, e na lógica do <i>Server</i> , para a leitura dos valores das matrizes de tempo de processamento.
MatrizSetup	<i>Real State</i>	Usado nos processos Matriz de Tempos, Sequenciamento, e Máquinas, para a leitura dos valores das matrizes de tempo de <i>setup</i> .
Check	<i>Boolean State</i>	Usado nos processos Sequenciamento, para identificar quais trabalhos já foram executados.
Ativado	<i>Boolean State</i>	Usado nos processos Inicialização e Sequenciamento, para identificar quais máquinas estão ativadas para uso de acordo com o NoMaquinas do cenário analisado.

Variáveis	Tipo	Descrição
Ocupado	<i>Boolean State</i>	Usado nos processos Sequenciamento e Máquinas, para identificar quais máquinas estão ocupadas ou desocupadas.
InputNode	<i>Node Reference State</i>	Usado nos processos Sequenciamento, para enviar o trabalho para o servidor (<i>Server</i>).

Como referido anteriormente, a execução dos processos é realizada por um *token*. O *token* representa o modelo ou um objeto, que pode ter as suas próprias variáveis de estado (*States*), mas não é o mesmo que uma entidade ou objeto. Os *tokens* executam os *steps* de um processo em nome do modelo ou de um objeto, normalmente o objeto cujo comportamento está sendo modificado pelo processo ou onde o processo reside chamado de Objeto Pai, ou o objeto que causa a modificação, chamado de Objeto Associado, (Pegden & Sturrock, 2014).

A Figura 8 mostra o *token* definido para a execução dos processos e apelidado de *MyToken1*, bem como as suas variáveis (*State*).

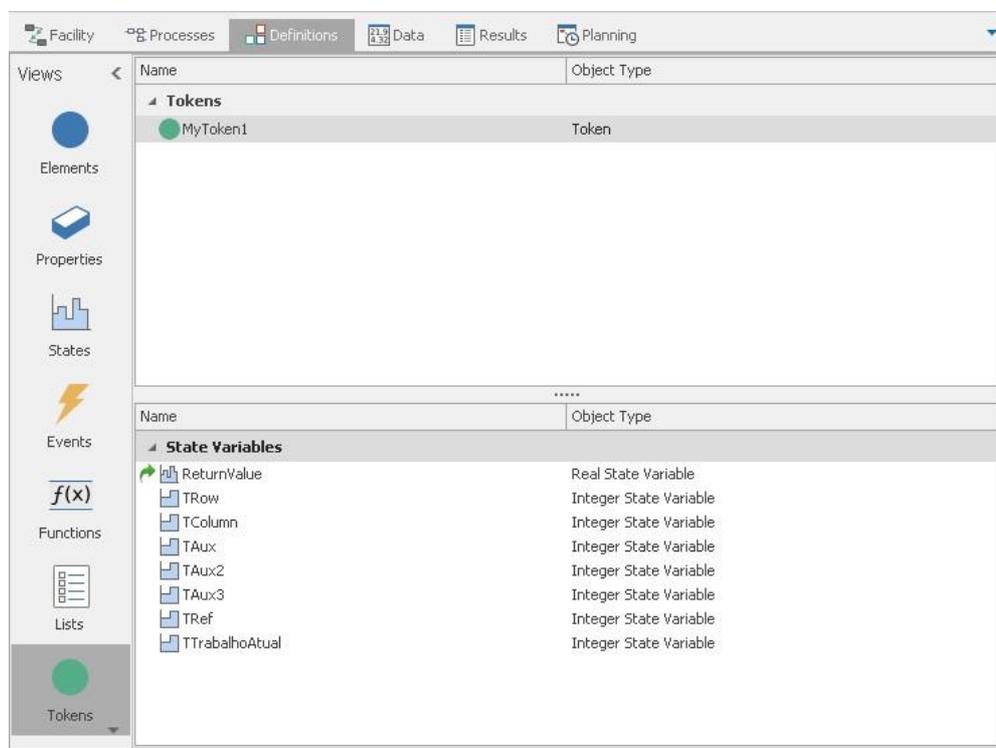


Figura 8 – Janela *Definitions (Tokens)*

A Tabela 8 descreve cada uma das variáveis (*States*) do *Mytoken1*, indicando onde serão utilizadas no decorrer da simulação.

Tabela 8 – Descrição de variáveis *States* do *Token*

Variáveis	Tipo	Descrição
TRow	<i>Integer State</i>	Usado no processo Matrizes de Tempos, como auxiliar da leitura das linhas das matrizes.
TColumn	<i>Integer State</i>	Usado no processo Matrizes de Tempos, como auxiliar da leitura das colunas das matrizes.
TAux	<i>Integer State</i>	Usado no processo Matrizes de Tempos, como auxiliar da leitura das linhas das matrizes.
TAux2	<i>Integer State</i>	Usado no processo Matrizes de Tempos, como auxiliar da leitura das colunas das matrizes.
TAux3	<i>Integer State</i>	Usada no processo Máquinas, para reconhecer o último trabalho executado na máquina.
TRef	<i>Integer State</i>	Usado no processo Matrizes de Tempos, como auxiliar para início da leitura da matriz de <i>setup</i> .
TTrabalhoAtual	<i>Integer State</i>	Usado no processo Sequenciamento, como auxiliar identificação do último trabalho que foi realizado.

Para o *ModelEntity* foi criada uma única *State* do tipo *Integer* apelidada de “EStaTrabalho”, utilizada nos processos de Sequenciamento, sendo responsável pelo envio do trabalho selecionado de acordo com a

regra de sequenciamento, para o servidor (máquina) escolhido. A Figura 9 apresenta onde essa variável é inserida no modelo.

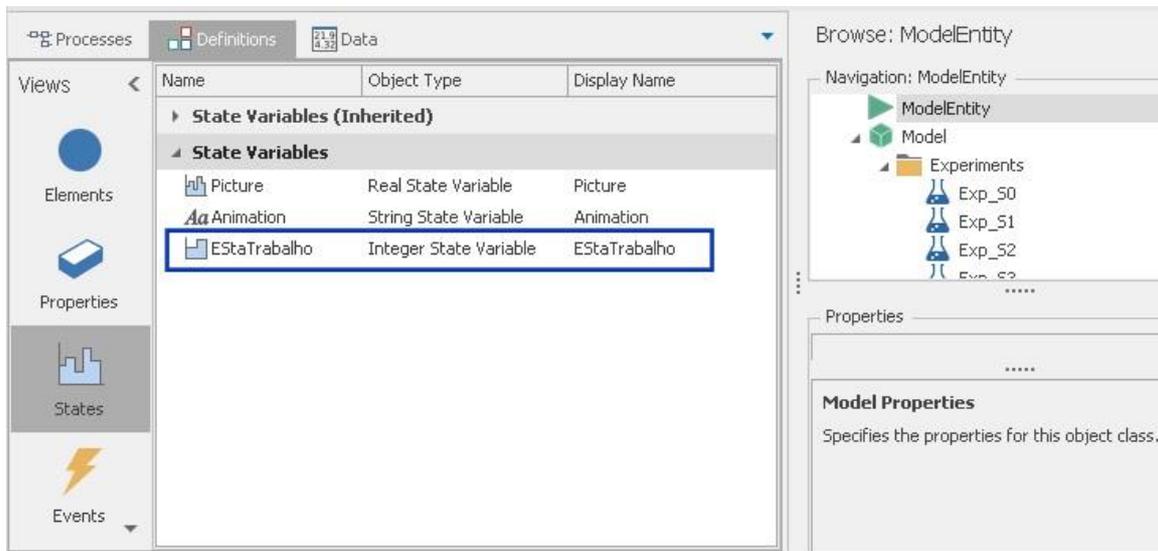


Figura 9 – Janela Definitions (*States* em *ModelEntity*)

No separador *Data*, foram construídas 4 tabelas, que representam as seguintes informações:

- T_Dados: representa a tabela que faz a relação da informação de Cenário com a Conexão definida para os dados de tempos de processamento e tempos de *setup*, como mostra a Figura 10.

The screenshot shows the 'Data' window with a table titled 'T_Dados'. The table has two columns: 'Cenário' and 'Conexão'. The rows are numbered 1 to 20. The 'Cenário' column contains values like '2m20t5D_1' through '2m40t5D_5'. The 'Conexão' column contains the value '1' for all rows.

	Cenário	Conexão
1	2m20t5D_1	1
2	2m20t5D_2	1
3	2m20t5D_3	1
4	2m20t5D_4	1
5	2m20t5D_5	1
6	2m20t5D_6	1
7	2m20t5D_7	1
8	2m20t5D_8	1
9	2m20t5D_9	1
10	2m20t5D_10	1
11	2m20t5D_11	1
12	2m20t5D_12	1
13	2m20t5D_13	1
14	2m20t5D_14	1
15	2m20t5D_15	1
16	2m40t5D_1	1
17	2m40t5D_2	1
18	2m40t5D_3	1
19	2m40t5D_4	1
20	2m40t5D_5	1

Figura 10 – Janela *Data* (T_Dados)

- T_Conexões: representa a ligação do processo criado com a tabela de tempos, para leitura dos dados, como mostra a Figura 11.

	Processo
1	Escrever_matrizes_1
2	Escrever_matrizes_2
3	Escrever_matrizes_3
4	Escrever_matrizes_4
5	Escrever_matrizes_5
6	Escrever_matrizes_6
7	Escrever_matrizes_7
8	Escrever_matrizes_8
9	Escrever_matrizes_9
10	Escrever_matrizes_10
11	Escrever_matrizes_11
12	Escrever_matrizes_12
*	

Figura 11 – Janela Data (T_Conexões)

- T_Trabalhos: representa a tabela que está relacionando à quantidade máxima de trabalhos propostos para esse problema (120), como mostra a Figura 12.

	Indice
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20

Figura 12 – Janela Data (T_Trabalhos)

- T_Servidores: representa a ligação com cada um dos servidores (*Server*) definidos, que está relacionada com a quantidade máxima de máquinas propostas para este problema (12). Como mostra a Figura 13.

	Indice	Input
1	1	Input@MyServer1
2	2	Input@MyServer2
3	3	Input@MyServer3
4	4	Input@MyServer4
5	5	Input@MyServer5
6	6	Input@MyServer6
7	7	Input@MyServer7
8	8	Input@MyServer8
9	9	Input@MyServer9
10	10	Input@MyServer10
11	11	Input@MyServer11
12	12	Input@MyServer12
*		

Figura 13 – Janela *Data* (T_Servidores)

A Figura 14 mostra o separador *Processes*, onde foram criados quatro processos, que contêm a maior parte da sequência lógica que o modelo de simulação irá executar.

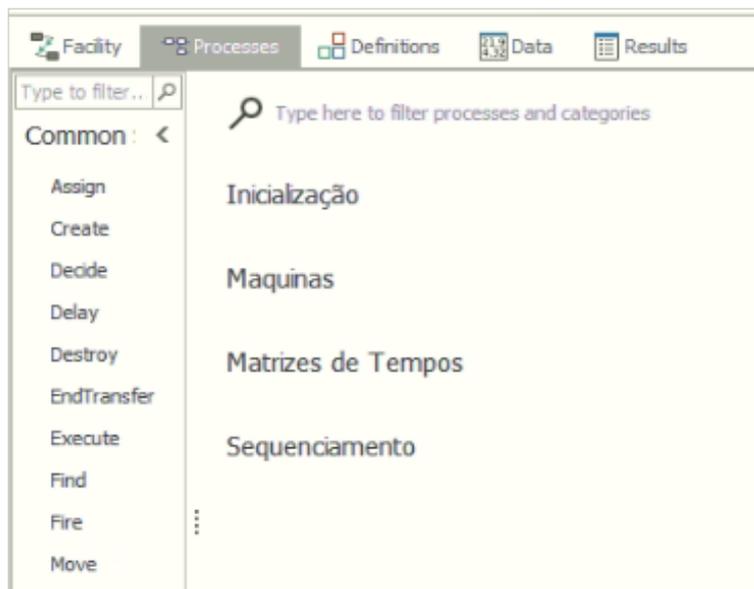


Figura 14 – Janela *Processes* (Processos do modelo)

A seguir, será feita a descrição de como foi modelado cada um destes processos.

3.2.3 Processo principal do modelo

O processo principal do modelo é o de Inicialização (*OnRunInicialized*), onde também são executados os restantes processos, através do *step Execute*, identificados pela cor vermelha na Figura 15, que mostra também, o fluxograma completo do processo principal.

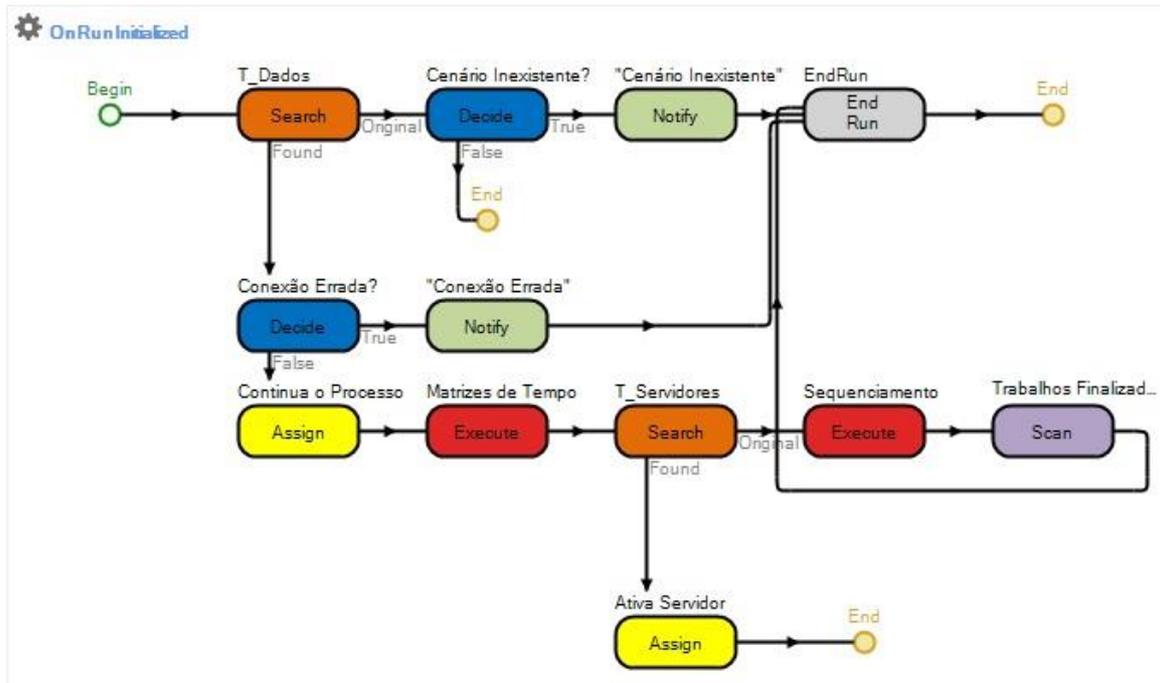


Figura 15 – Processo de Inicialização (*OnRunInicialized*)

Este inicia com um *step Search* (*T_Dados*), que busca o cenário que se deseja analisar, de acordo com o Cenário escolhido no *PptCenário* na janela *Facility*, apresentado anteriormente na Figura 4.

De seguida, tem-se o *step Decide* (*Cenário Inexistente?*) (*Conexão Errada?*), que verifica se as informações que o usuário inseriu em *PptConexão* e *PptCenário* estão corretas, pois, caso seja inserido um valor incorreto ou inexistente, a simulação termina e o usuário é informado disso mesmo, através de um *step Notify*.

Confirmadas as informações, segue para o *step Execute* (*Matrizes de Tempos*), onde entra no processo “*Matrizes de Tempos*”, que realiza a leitura da matriz dos valores de tempo de processamento [*NoTrabalhos* x *NoMaquinas*] e tempo de *setup* [*NoTrabalhos* x *NoTrabalhos*] x *NoMaquinas*] referente ao cenário escolhido.

Finalizada a execução desse processo, segue para o *step Search* (*T_Servidores*), onde são verificados e ativados os servidores de acordo com o número de máquinas necessário (*NoMaquinas*), determinado de

acordo com o cenário escolhido. Para este modelo, no tempo zero, todas as máquinas estão disponíveis e são ativadas em sequência, e não de forma aleatória.

Após a ativação das máquinas, segue para o *step Execute* (Sequenciamento) que é referente ao tipo de sequenciamento escolhido para análise do cenário. Para cada trabalho que passou pelo processo de “Sequenciamento”, quando sai, entra no *step Scan* (Trabalhos Finalizados), e a entidade trabalho “EntTrabalho” é destruída, e contabilizada, pois, quando a quantidade de “EntTrabalho” que passar pelo *step Scan*, for igual ao número de trabalhos total do cenário analisado, a simulação finaliza.

3.2.4 Processo da leitura das matrizes de tempos

O processo “Matrizes de Tempo” busca a planilha de cálculo relacionada ao cenário que se deseja analisar. Cada cenário tem uma conexão definida, portanto, são 12 arquivos diferentes e cada um deles possui o seu próprio processo [*Escrever_matrizes*] que está dentro do processo “Matrizes de Tempo”.

Esse processo é composto pelos recursos: *ExcelConnect*, (ver Figura 5); *States*: *MatrizTempo* e *MatrizSetup*, ambos do tipo *Real State Variable*, que formam as matrizes de tempo de processamento e tempo de *setup*, (ver Figura 16); e *ExcelRead*. O processo completo de escrever matrizes pode ser visualizado na Figura 17.

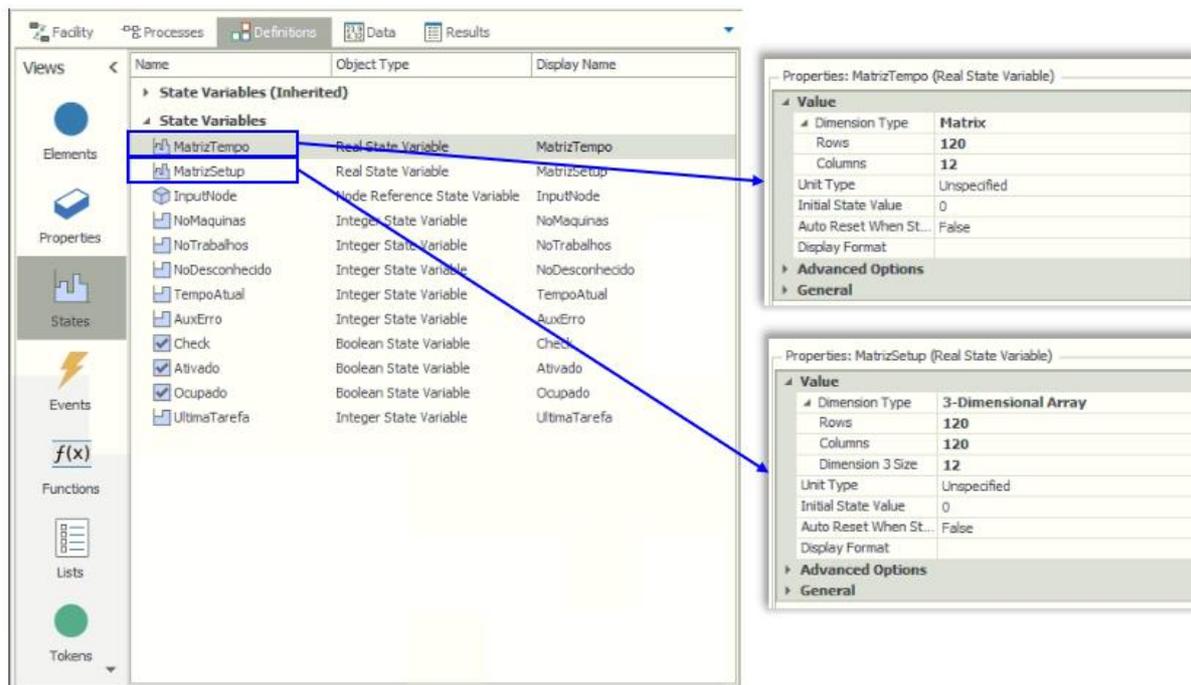


Figura 16 – Janela *Definitions (States)*

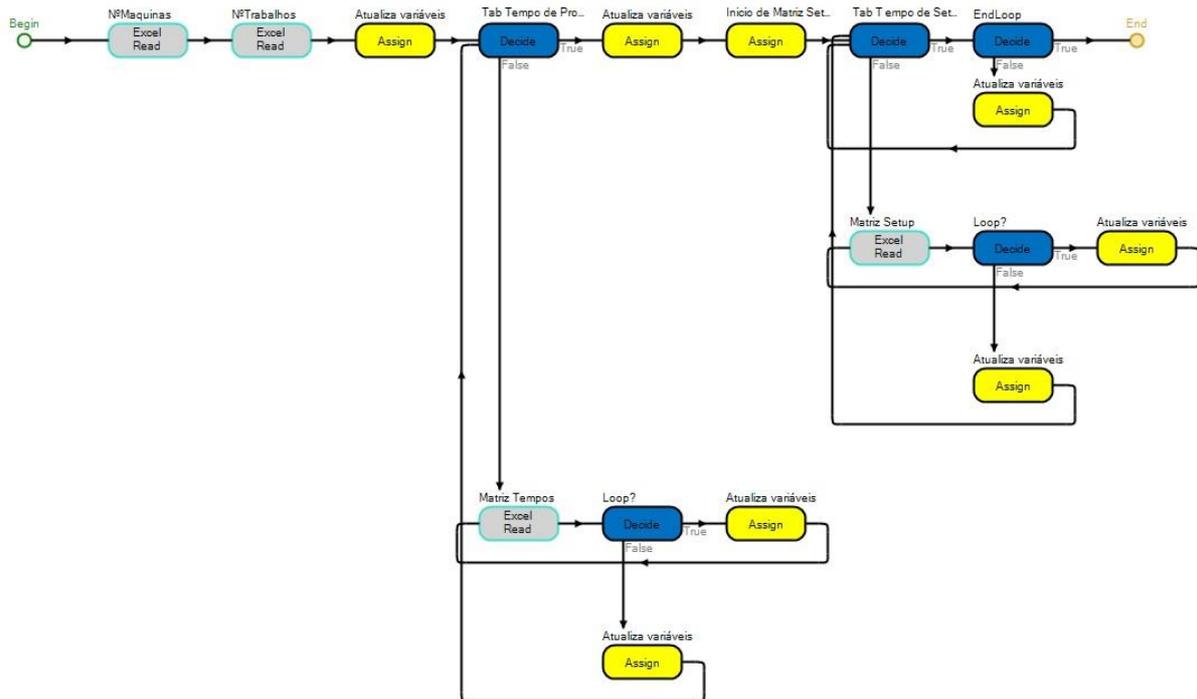


Figura 17 – Processo de escrever matrizes

Este processo inicia fazendo a leitura do número de máquinas (*NoMaquinas*) e número de trabalhos (*NoTrabalhos*) do cenário em questão, depois o *step Assign* identifica qual linha e coluna deve começar a leitura dos tempos na planilha de cálculo. Então, percorre todos os valores, iniciando pela matriz de tempo de processamento [$(NoTrabalhos \times NoMaquinas)$], e depois que finaliza, os valores das variáveis são novamente atualizados no *step Assign*, e segue para a leitura de todos os valores das matrizes de tempos de *setup* [$(NoTrabalhos \times NoTrabalhos) \times NoMaquinas$]. É feita uma gravação local dos valores, que fica salva enquanto o processo estiver sendo executado.

Após a leitura de todas as matrizes, o *token* segue para o *step Search* (T_Servidores), no processo principal, onde os servidores serão ativados de acordo com o número de máquinas indicado (*NoMaquinas*). Após a ativação dos servidores, o *token* segue para o próximo processo entrando no *step Execute* (Sequenciamento), como pode ser visto no fluxograma da Figura 15. Neste processo, será determinado qual tipo de sequenciamento será executado, de acordo com a escolha realizada em *PptSequenciamento*, na janela *Facility*, mostrado na Figura 4.

3.2.5 Processos das regras de sequenciamento

A construção do modelo para execução das regras de sequenciamento seguirá a ordem descrita na Tabela 6.

O primeiro sequenciamento é o S0, que representa a regra *Service In Random Order* (SIRO), que segue uma sequência aleatória. A Figura 18 mostra o fluxograma da lógica do processo de sequenciamento aleatório.

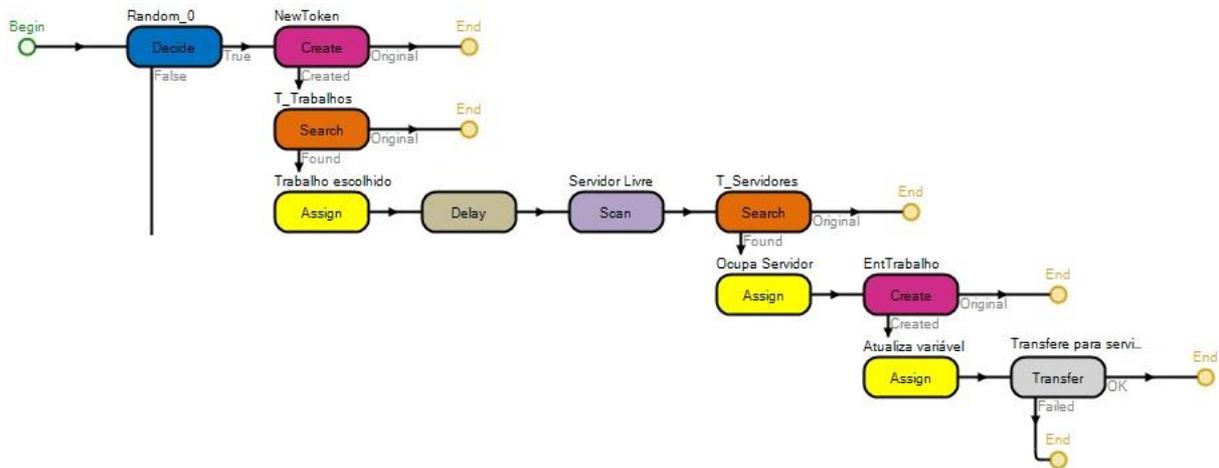


Figura 18 – Processo de sequenciamento aleatório

O processo inicia no *step Decide*, que identifica qual a regra de sequenciamento foi escolhida em *PptSequenciamento*, como mostrado na Figura 4. Se o valor escolhido for zero (0), segue para o *step Create*, com a criação de novos *tokens*, de acordo com o número de trabalhos indicado (*NoTrabalhos*). Depois cada um dos *tokens* entra individualmente no *step Search* (*T_Trabalhos*), buscando quais trabalhos estão disponíveis para serem executados. Segue para o *step Assign* (*Trabalho escolhido*), onde identifica o trabalho escolhido para ser executado. Entra no *step Delay* de tempo "*Math.Epsilon*", que devolve o valor real mais pequeno próximo de zero. Este é necessário devido à natureza da simulação discreta, por forma a que o relógio da simulação avance, e as alterações ao modelo ocorram. Em seguida, vai para o *step Scan* (*Servidor Livre*), que verifica se há pelo menos um servidor (máquina) livre para o qual o trabalho possa seguir. Enquanto todos os servidores (máquinas) estiverem ocupados, o *token* fica preso, e apenas quando algum servidor estiver livre, é que o *token* segue para o *step Search* (*T_Servidores*), buscando qual servidor que está ativo e disponível para ser utilizado.

O próximo *step Assign* (*Ocupa Servidor*) aponta o servidor que será ocupado pela entidade (trabalho) em análise e atualiza uma variável (*InputNode*), identificando qual é o nó de entrada do servidor (máquina) para onde este trabalho será enviado. Segue-se o *step Create* onde é criada uma entidade trabalho (*EntTrabalho*), e no *step Assign* seguinte, atualiza a variável da entidade "EStTrabalho" com o valor do trabalho que será executado, e por fim transfere essa informação para o nó, indicando o servidor (máquina) onde este trabalho será processado.

Todos os servidores (máquinas) e todas as entidades (trabalhos) estão disponíveis no tempo zero, portanto, à medida que um trabalho sai e é enviado para o servidor (máquina), qualquer um outro trabalho que esteja disponível entra imediatamente no processo descrito acima, seguindo para a máquina que está disponível.

Conforme mostrado na Tabela 6, o próximo é o sequenciamento S1 que representa a regra *Longest Process Time* (LPT), onde a prioridade é o trabalho que possui maior tempo de processamento. A Figura 19 mostra o fluxograma com a lógica para a execução deste sequenciamento.

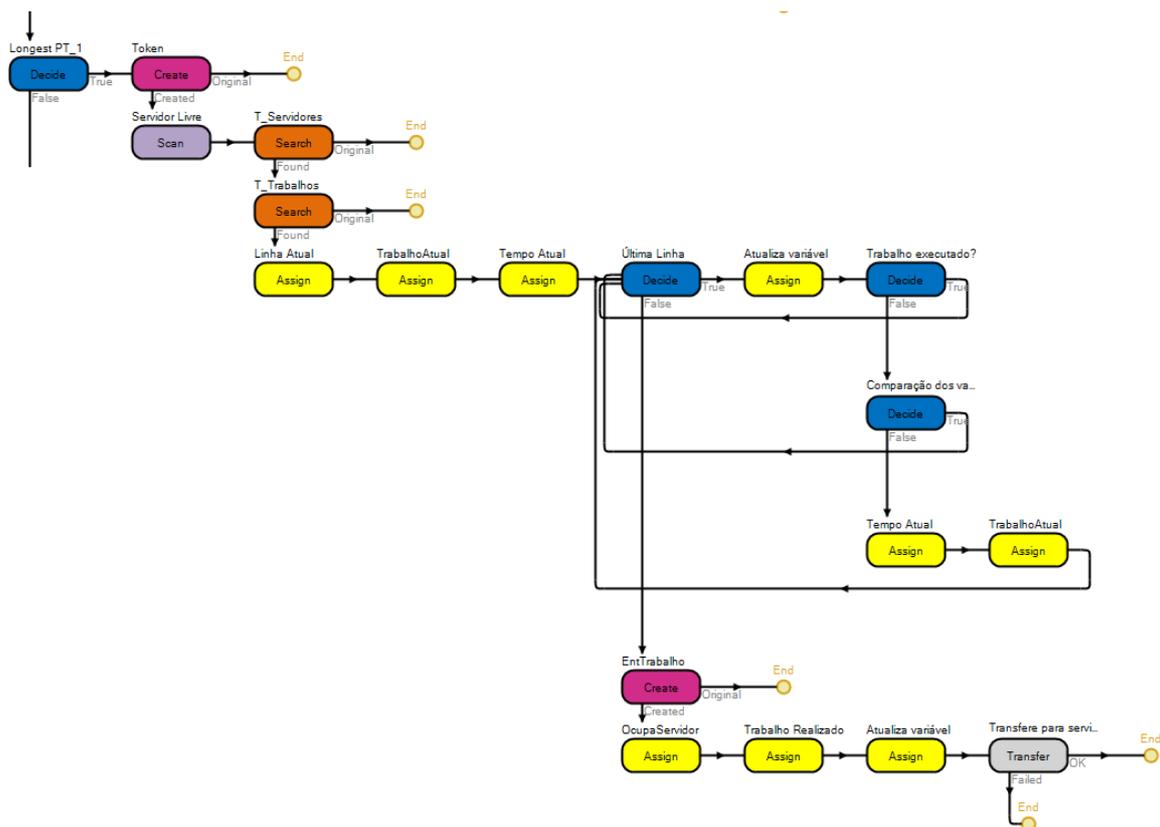


Figura 19 – Processo prioriza maior tempo de processamento

O processo inicia no *step Decide*, que identifica qual a regra de sequenciamento foi escolhida em *PptSequenciamento*, como mostrado na Figura 4. Se o valor escolhido for um (1), segue para o *step Create*, com a criação de novos *tokens*, de acordo com o número de trabalhos indicado (NoTrabalhos). Em seguida vai para o *step Scan* (Servidor Livre), que identifica se tem pelo menos um servidor (máquina) livre para que o trabalho possa seguir. Enquanto todos os servidores estiverem ocupados, o *token* fica esperando liberar, e quando algum estiver livre, o *token* segue para o *step Search* (T_Servidores), buscando qual servidor que está ativado e disponível para uso, então vai para o próximo *step Search* (T_Trabalhos), buscando os trabalhos que estão disponíveis para serem executados.

Depois de selecionado o trabalho e o servidor, o *token* segue para os *steps Assign* (Linha Atual, Trabalho Atual e Tempo Atual), em que as variáveis são atualizadas indicando qual linha da matriz na planilha de cálculo está sendo iniciada a leitura dos tempos, qual o trabalho que está sendo analisado, e qual o valor de tempo de processamento deste trabalho.

No *step Decide* (*Última Linha*), verifica se todas as linhas já foram analisadas, e enquanto o valor da variável que conta as linhas (*TRow*) for menor que o número de trabalho (*NoTrabalhos*), o processo continua com a análise dos dados até que a última linha seja verificada, pois na matriz de tempo de processamento, o número de linhas é igual ao número de trabalhos.

O *step Assign* seguinte faz o incremento do valor da variável (*TRow*) que percorre as linhas. Depois entra no *step Decide* (Trabalho executado?) e verifica se o próximo trabalho a ser verificado já foi executado ou não. Se ainda não tiver sido executado, então entra em um outro *step Decide* (Comparação dos valores) e verifica se o valor de tempo da linha que está sendo analisada é o maior do que os valores de tempos analisados nas linhas anteriores, e mesmo que seja, só termina de analisar quando todas as linhas tiveres sido percorridas, pois nas próximas linhas, pode ser que encontre um valor de tempo maior.

Depois de analisadas todas as linhas e encontrado o maior valor de tempo de processamento, segue para o *step Create*, e é criada uma entidade trabalho (*EntTrabalho*). Em seguida essa entidade vai para um próximo *step Assign* (Ocupa Servidor), identifica o servidor que será ocupado e atualiza uma variável (*InputNode*), identificando qual é o nó de entrada do servidor (máquina) que esse trabalho será enviado. No *step Assign* seguinte, atualiza a variável da entidade “*EStaTrabalho*” com o valor do trabalho que será executado, e por fim, transfere essa informação para o nó, indicando o servidor (máquina) que este trabalho será executado.

Seguindo os dados da Tabela 6, o próximo a ser modelado é o sequenciamento S2, que utiliza a regra *Shortest Process Time* (SPT), onde a prioridade é o trabalho que possui menor tempo de processamento. A Figura 20 mostra o fluxograma para esse sequenciamento.

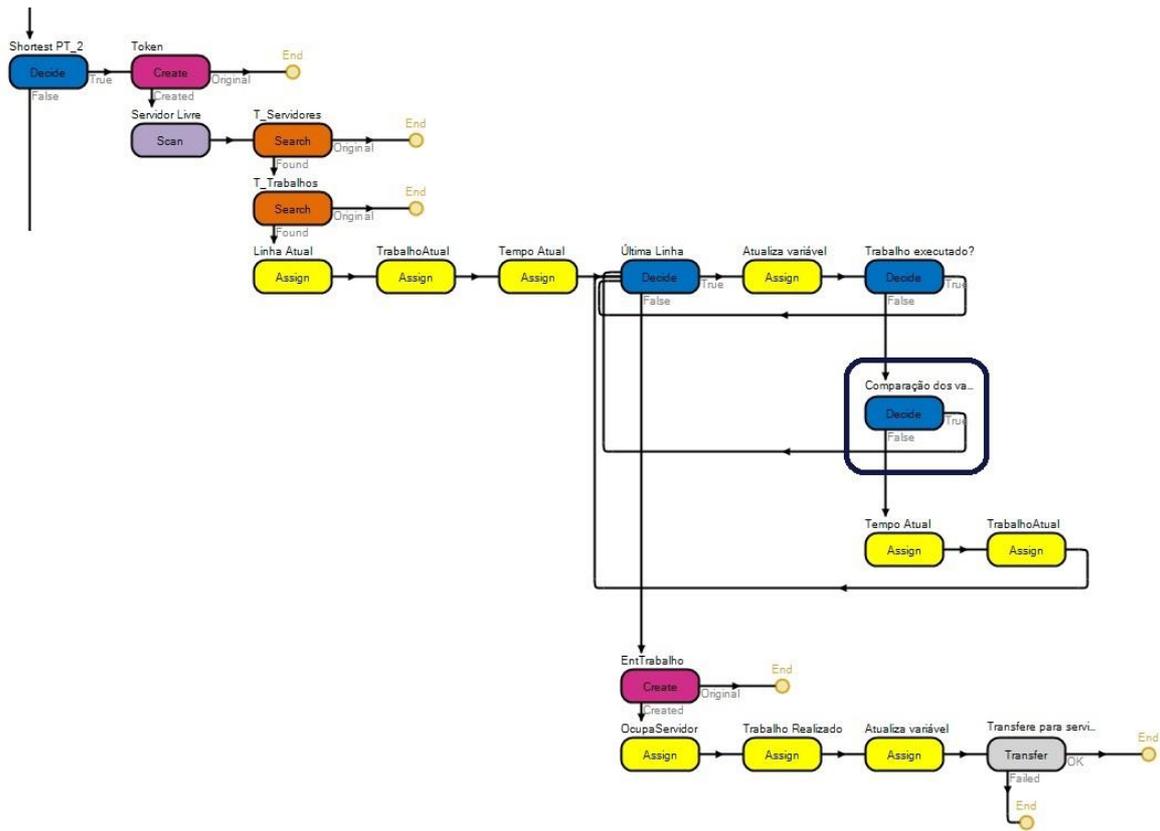


Figura 20 – Processo prioriza menor tempo de processamento

Esse sequenciamento segue a mesma linha lógica do S1, porém, o que diferencia os fluxogramas lógicos é que neste sequenciamento, o primeiro *step Decide* verifica se o valor escolhido em *PptSequenciamento* foi o número dois (2), que representa o sequenciamento que prioriza o menor tempo de processamento, e no *step Decide* (Comparação dos valores) que está destacado ao lado direito na Figura 20, ao invés de buscar pelo maior tempo de processamento como foi feito em S1, neste caso busca o menor tempo de processamento.

Seguindo os dados da Tabela 6, a Figura 21 representa o sequenciamento S3, que utiliza a regra de restrição *Longest Setup Time*, que a prioriza o trabalho que possui maior tempo de *setup*, dependente da máquina. Esta regra é uma adaptação da regra *Shortest Setup Time* (SST).

Da mesma forma que nas regras anteriores, no tempo zero, todos os trabalhos e todas as máquinas estão disponíveis; e não é considerado o tempo de *setup* para o primeiro trabalho de cada máquina.

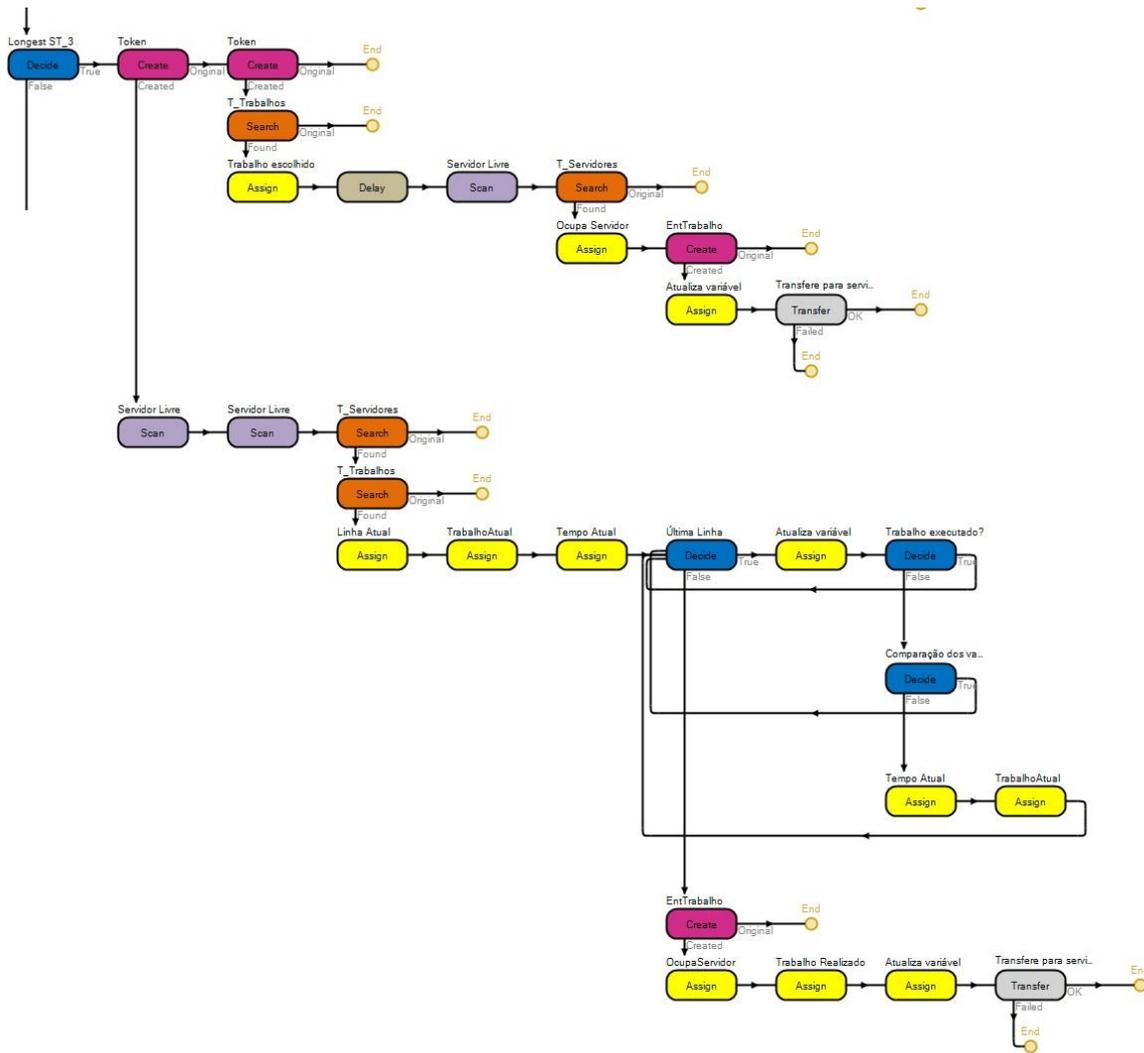


Figura 21 – Processo prioriza maior tempo de *setup*

O processo inicia no *step Decide*, que identifica qual a regra de sequenciamento foi escolhida em *PptSequenciamento*, como mostrado na Figura 4. Se o valor escolhido for três (3), segue para o *step Create* em que cria novos *tokens*, e a quantidade de *tokens* criada é: $NoTrabalhos - NoMaquinas$. Esse valor está relacionado a quantidade de trabalho que precisarão considerar o tempo de *setup*, pois, como dito anteriormente, o primeiro trabalho de cada máquina não precisa executar o tempo de *setup*, e esses trabalhos estão representados no segundo *step Create*, pois são criados novos *tokens* de acordo com o número de máquinas ($NoMaquinas$).

A primeira parte do fluxograma representa todas as primeiras atividades de cada máquina, nas quais não é necessário considerar o tempo de *setup*. Portanto, cada um dos *tokens* segue individualmente para o *step Search* ($T_Trabalhos$), buscando quais trabalhos estão disponíveis para serem executados.

Em seguida, vai para o *step Assign* (Trabalho escolhido), onde identifica qual trabalho foi escolhido, e então entra no *step Delay* de tempo "*Math.Epsilon*", que devolve o valor real mais pequeno próximo de zero. Este é necessário devido à natureza da simulação discreta, por forma a que o relógio da simulação avance, e as alterações ao modelo ocorram, e segue para o *step Scan* (Servidor Livre), que identifica se tem pelo menos um servidor (máquina) livre para que o trabalho possa seguir. Enquanto todos os servidores estiverem ocupados, o *token* fica esperando liberar, e quando algum servidor estiver livre, o *token* segue para o *step Search* (T_Servidores), buscando qual servidor que está ativado e disponível para uso.

O próximo *step Assign* (Ocupa Servidor) identifica o servidor ocupado e atualiza uma variável (*InputNode*), identificando qual é o nó de entrada do servidor (máquina) que esse trabalho será enviado. Cria-se uma entidade trabalho (EntTrabalho), e no *step Assign* seguinte atualiza a variável da entidade "EStaTrabalho" com o valor do trabalho que será executado, e por fim, transfere essa informação para o nó, indicando o servidor (máquina) que este trabalho será executado.

Essa primeira parte do processo segue a mesma sequência lógica do sequenciamento zero (0), pois todas as máquinas e todos os trabalhos estarão disponíveis no tempo zero, e como não é necessário considerar o tempo de *setup* para o primeiro trabalho da máquina, o trabalho foi escolhido de forma aleatória.

No momento em que o primeiro servidor (máquina) for liberado, o primeiro *step Scan* (Servidor Livre) da sequência lógica é acionado, e o *token* segue para o segundo *step Scan* (Servidor Livre) para confirmar que há pelo menos um servidor disponível.

Em seguida entra no *step Search* (T_Servidores) buscando qual servidor que está ativado e disponível para uso. Em seguida vai para o próximo *step Search* (T_Trabalhos), buscando os trabalhos que estão disponíveis para serem executados.

Depois de selecionado o trabalho e o servidor, o *token* segue para os 3 *step Assign* (Linha Atual, Trabalho Atual e Tempo Atual), em que as variáveis são atualizadas, apontando qual linha da matriz na planilha de cálculo indica o trabalho escolhido; qual o trabalho que está sendo analisado; e qual o valor de tempo de *setup* deste trabalho para o servidor livre em questão.

No *step Decide* (Última Coluna), é realizado o controle para que todas as colunas sejam analisadas, e o *step Assign* seguinte faz o incremento do valor da variável que percorre as colunas. O *step Decide* (Trabalho executado?) analisa se o próximo trabalho a ser verificado já foi executado ou não. Se ainda não tiver sido executado, então entra em um outro *step Decide* (Comparação dos valores) e verifica se

os valores de tempo de *setup* do trabalho atual, para a máquina escolhida, é maior que o tempo de *setup* do próximo trabalho livre.

Para esse processo, é analisado o tempo de *setup* de todos os trabalhos que estão livres para serem executados no servidor escolhido, e busca o maior valor. Quando esse maior tempo de *setup* é encontrado, o *token* segue para o *step Create* onde é criada uma entidade trabalho (EntTrabalho). Em seguida essa entidade vai para o próximo *step Assign* (Ocupa Servidor), e identifica o servidor que será ocupado, e atualiza a variável (*InputNode*), identificando qual é o nó de entrada do servidor (máquina) que esse trabalho será enviado. No *step Assign* seguinte, atualiza a variável da entidade “EStoTrabalho” com o valor do trabalho que será executado e, por fim, transfere essa informação para o nó, indicando o servidor (máquina) que este trabalho será executado.

A Figura 22 mostra o fluxograma do sequenciamento S4, que utiliza a regra de restrição *Shortest Setup Time* (SST), conforme descrição da Tabela 6, onde a prioridade é o trabalho que possui menor tempo de *setup*, dependente da máquina.

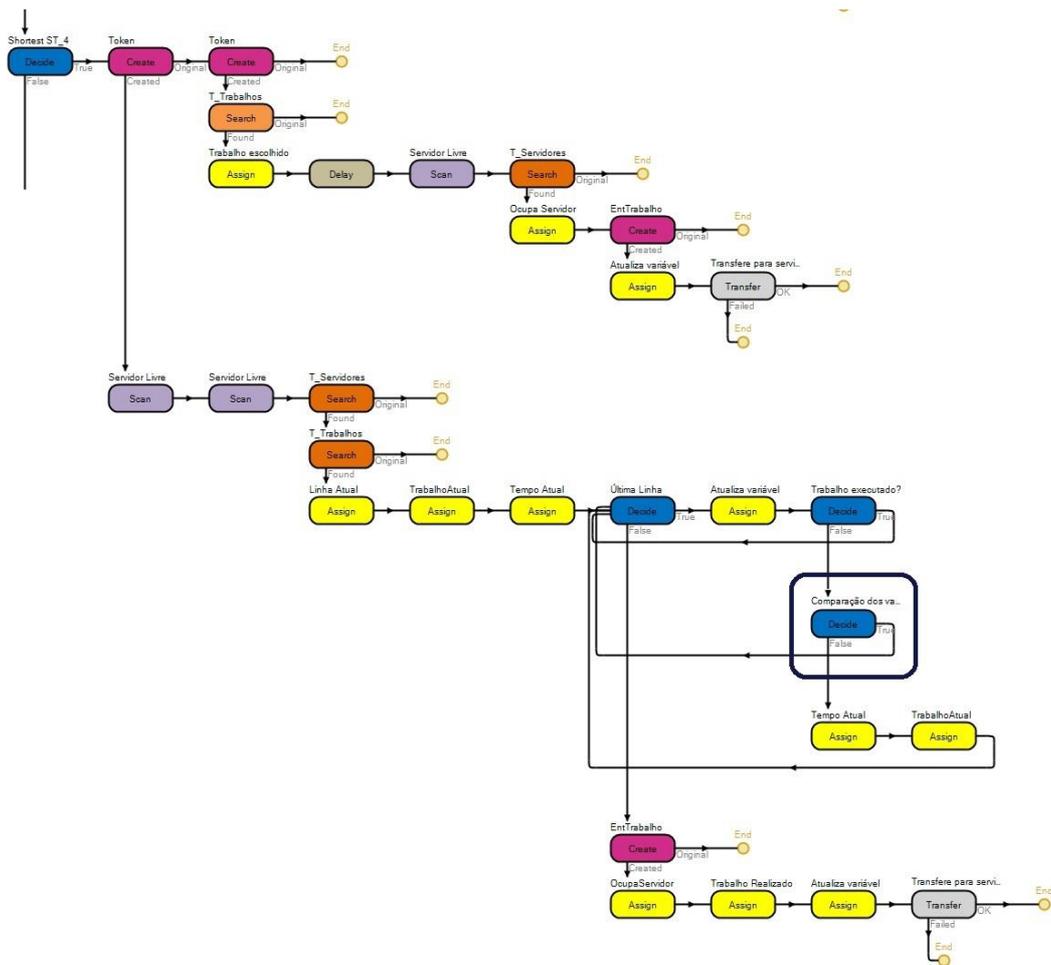


Figura 22 – Processo prioriza menor tempo de *setup*

Esse sequenciamento segue a mesma linha lógica do S3, porém, o que diferencia os fluxogramas lógicos é que para este sequenciamento, o primeiro *step Decide* verifica se o valor escolhido em *PptSequenciamento* foi o número quatro (4), que representa o sequenciamento que prioriza o menor tempo de *setup*, e no *step Decide* (Comparação dos valores) destacado ao lado direito na Figura 22, ao invés de buscar pelo maior tempo de *setup* como foi feito anteriormente, neste caso busca o menor tempo de *setup*.

3.2.6 Processo de entrada e saída dos servidores (Server)

Quando o processo “Sequenciamento” finaliza, continua com o envio do trabalho para o servidor (máquina) definido na variável “*InputNode*”, como descrito anteriormente.

O trabalho é transferido para um nó de entrada (*Input@MyServer*), e cada servidor possui sua entrada própria, como pode ser visualizado na Figura 23.

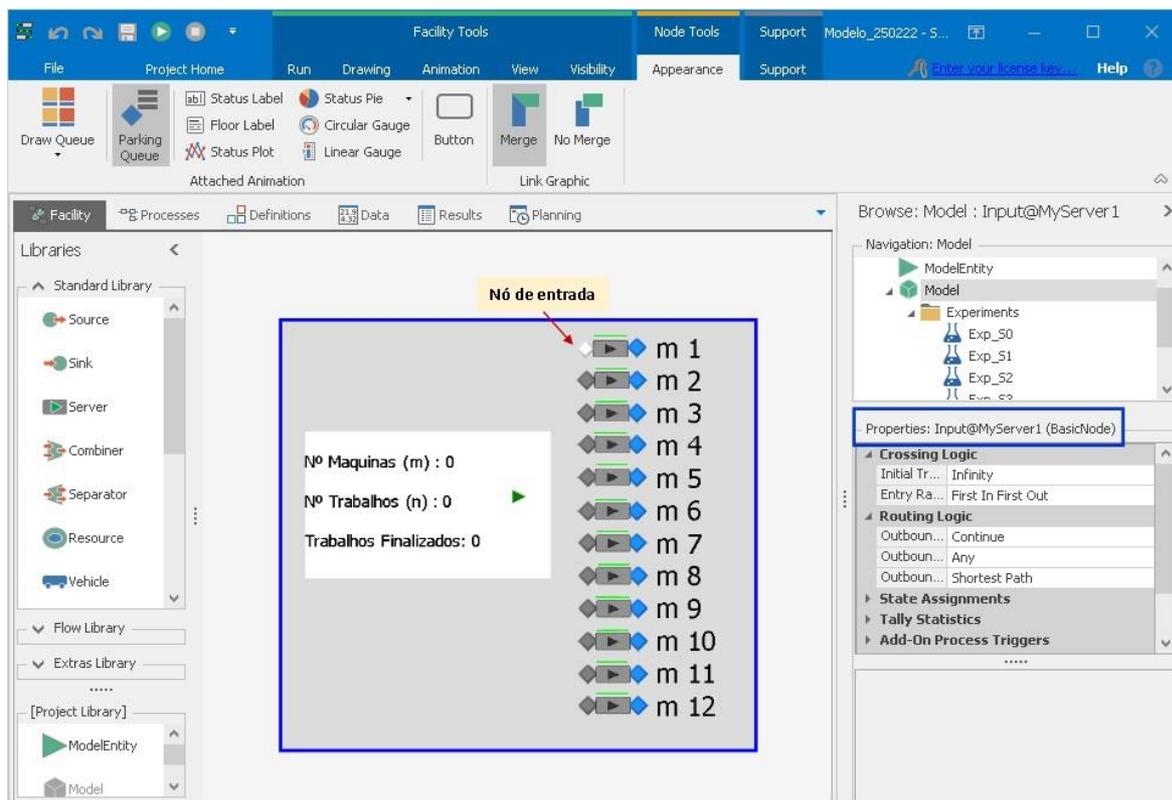


Figura 23 – Processo de entrada no servidor

A Figura 24 mostra os processos criados para serem executados antes e depois do tempo de processamento. Para esses processos, o *Server* possui uma extensão lógica em *Add-On Process Triggers*, onde esses processos são inseridos.

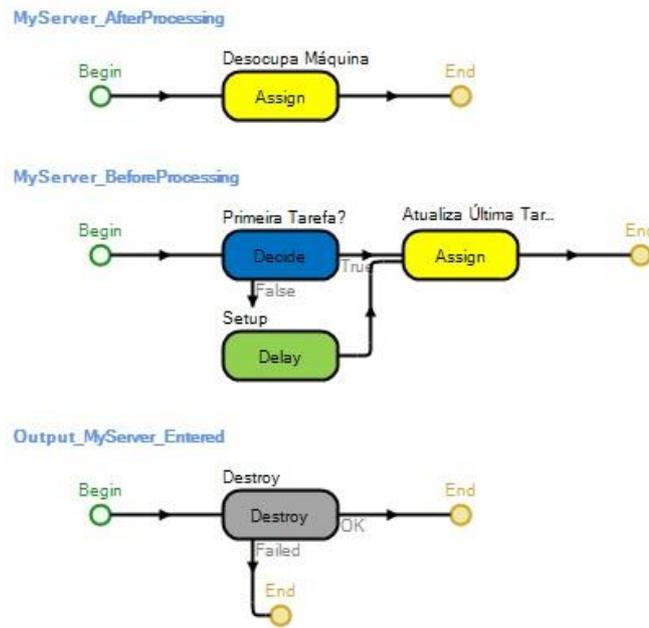


Figura 24 – Processo de entrada e saída dos servidores

Em *[MyServer_BeforeProcessing]*, o *step Decide* (Primeira Tarefa?) verifica se essa tarefa que está entrando para ser executada, é a primeira que o servidor recebe ou não. Caso seja a primeira, a *State* “Ultima Tarefa” é atualizada com o valor do trabalho que está entrando, caso contrário, segue para o *step Delay* (*Setup*), e esse valor de espera é equivalente ao valor da variável “MatrizSetup”, que é o tempo de *setup* dependente do trabalho executado anteriormente e da máquina. Depois a variável do servidor “UltimaTarefa” é atualizada e o processo segue.

O *[MyServer_AfterProcessing]* é realizado após o servidor executar o tempo de processamento, e é representado apenas por um *step Assign* (Desocupa Máquina), que atualiza a variável “Ocupado” de 1 para 0, ou seja, identifica que a máquina está livre para receber o próximo trabalho. Esses dois processos inseridos no servidor, podem ser visualizados no destaque do lado direito da Figura 25.

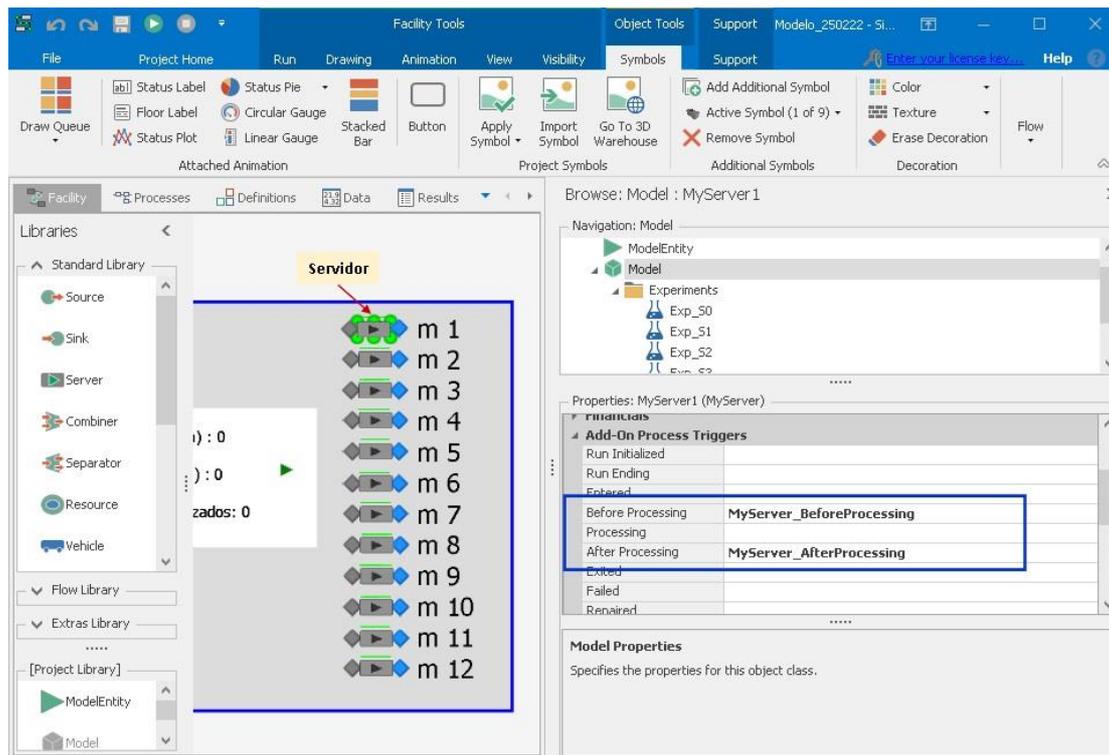


Figura 25 – Processos realizados antes e depois do tempo de processamento

Em `[Output_MyServer_Entered]`: utiliza o *step Destroy*, pois todas as vezes que uma entidade sai do processo ela é destruída, fazendo ligação com o *step Scan* do processo “Inicialização”, que contabiliza as entidades que são destruídas, e quando essa quantidade for igual ao número de trabalhos, a simulação finaliza. Destacado do lado direito da Figura 26, verifica-se onde é inserido este processo.

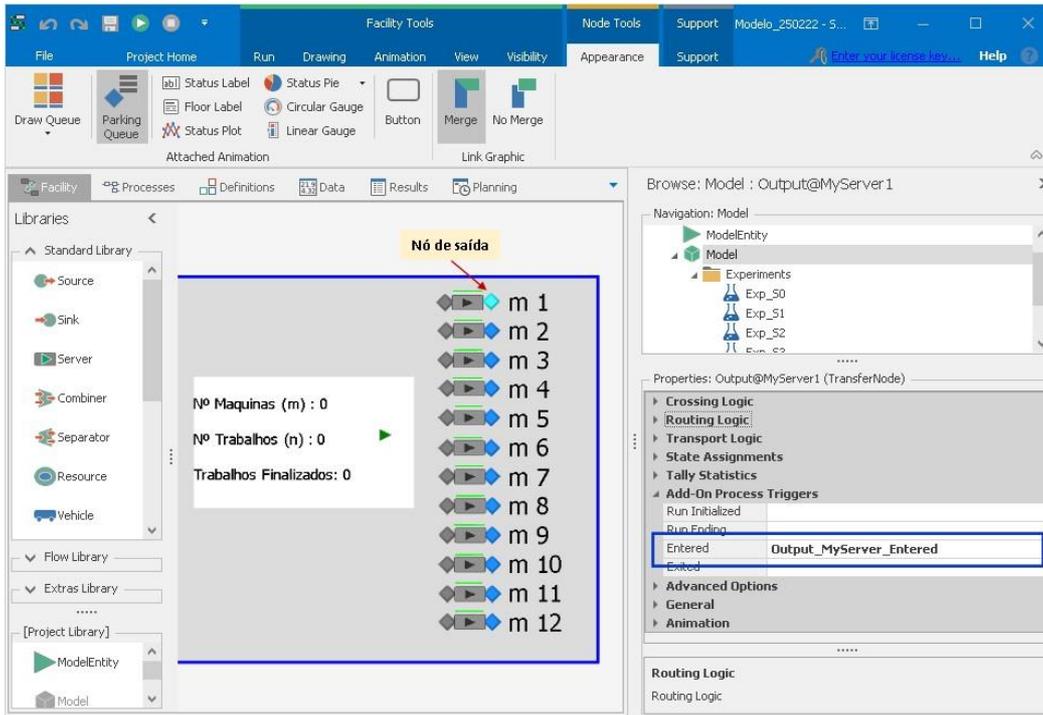


Figura 26 – Processo de saída do servidor

A lógica para a execução do Tempo de Processamento é realizada dentro do *Server*. Na lógica do processo (*Process Logic*), em *Processing Time*, é executado o valor da variável “MatrizTempo” que relaciona o trabalho com a máquina. A Figura 27 mostra destacado à direita, onde é inserido o valor do tempo de processamento.

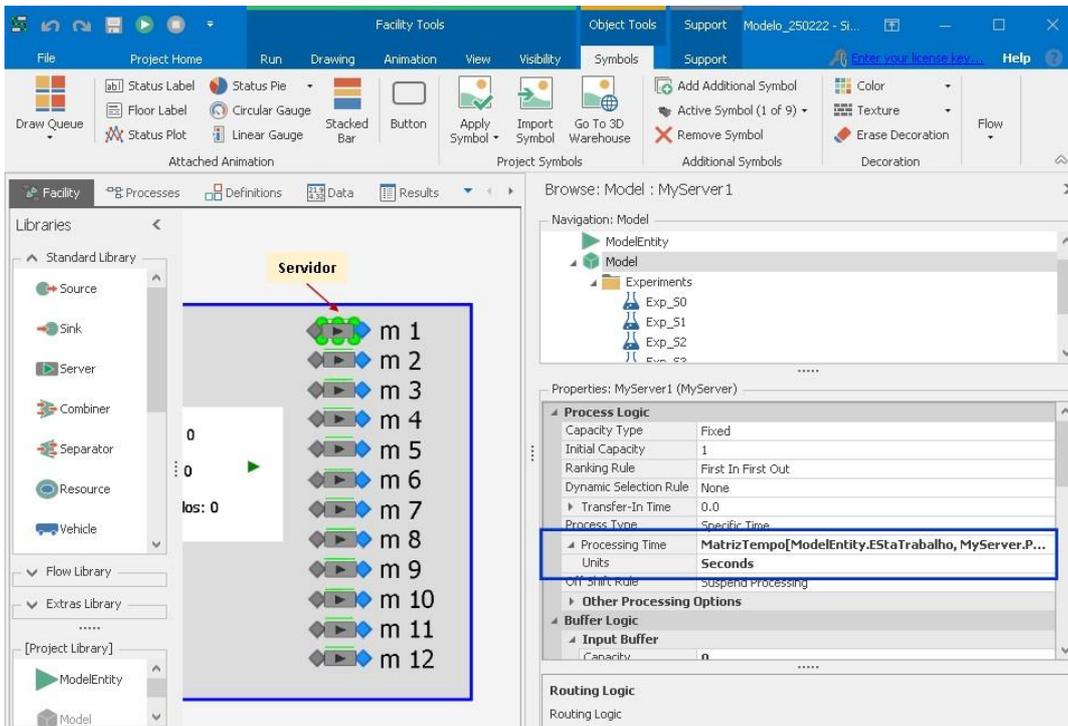


Figura 27 – Processo de execução do tempo de processamento

4. ANÁLISE DOS RESULTADOS

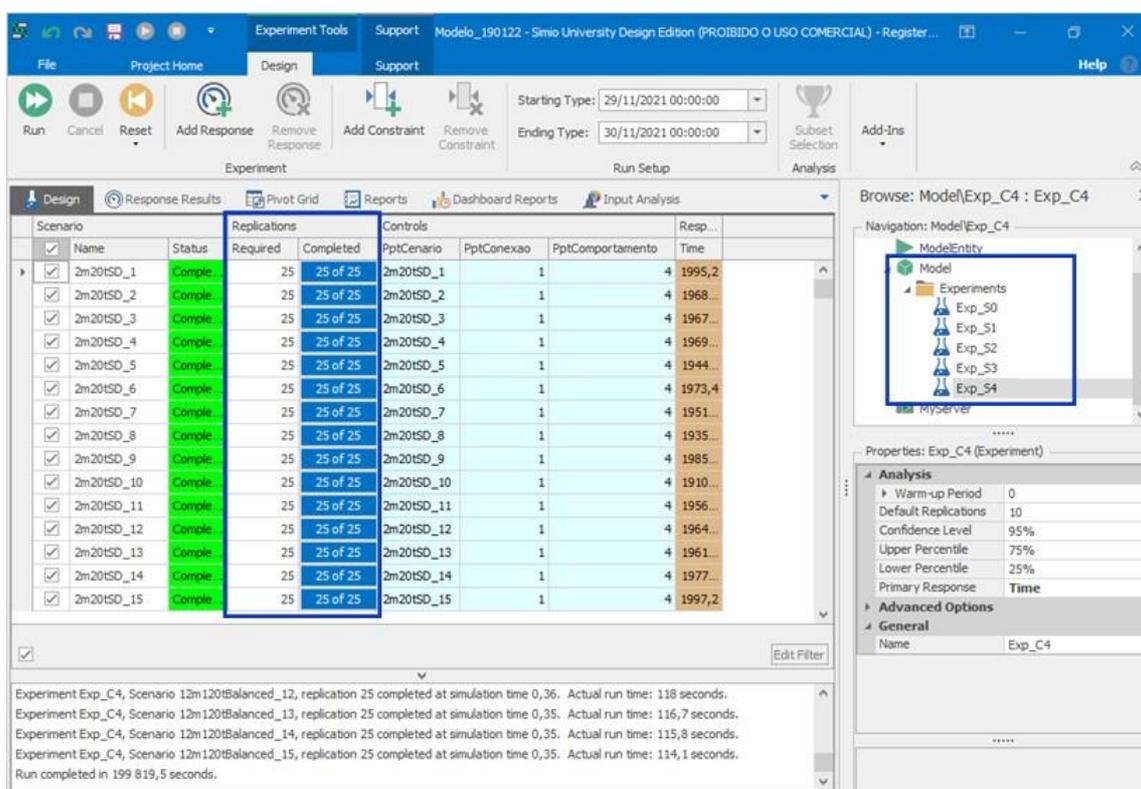
Os 5 sequenciamentos propostos foram simulados, considerando todas as 15 instâncias de cada cenário, conforme o estudo de Rabadi, et al. (2014).

O *software* SIMIO permite que sejam realizados experimentos, então foram feitas análises através de replicações de experimentos baseados nos métodos de sequenciamento determinados para esta simulação, dessa forma a comparação dos resultados foi mais assertiva.

4.1 Definição dos cenários

Foram realizadas (5, 10, 15, 20, 25) replicações de todos os cenários, considerando os mesmos valores determinado para cada instância. Depois foi feita uma comparação dos valores de tempo de processamento total de cada sequenciamento, com os valores de tempo de processamento total do resultado do ACO II de Rabadi, et al. (2014).

A Figura 28, mostra a tela de Experimentos do *software* SIMIO, que foram realizados para análise dos tempos de processamento total dos sequenciamentos propostos.



Scenario	Name	Status	Required	Completed	PptCenario	PptConexao	PptComportamento	Time
✓	2m20ISD_1	Complete	25	25 of 25	2m20ISD_1	1	4	1995,2
✓	2m20ISD_2	Complete	25	25 of 25	2m20ISD_2	1	4	1968...
✓	2m20ISD_3	Complete	25	25 of 25	2m20ISD_3	1	4	1967...
✓	2m20ISD_4	Complete	25	25 of 25	2m20ISD_4	1	4	1969...
✓	2m20ISD_5	Complete	25	25 of 25	2m20ISD_5	1	4	1944...
✓	2m20ISD_6	Complete	25	25 of 25	2m20ISD_6	1	4	1973,4
✓	2m20ISD_7	Complete	25	25 of 25	2m20ISD_7	1	4	1951...
✓	2m20ISD_8	Complete	25	25 of 25	2m20ISD_8	1	4	1935...
✓	2m20ISD_9	Complete	25	25 of 25	2m20ISD_9	1	4	1985...
✓	2m20ISD_10	Complete	25	25 of 25	2m20ISD_10	1	4	1910...
✓	2m20ISD_11	Complete	25	25 of 25	2m20ISD_11	1	4	1956...
✓	2m20ISD_12	Complete	25	25 of 25	2m20ISD_12	1	4	1964...
✓	2m20ISD_13	Complete	25	25 of 25	2m20ISD_13	1	4	1961...
✓	2m20ISD_14	Complete	25	25 of 25	2m20ISD_14	1	4	1977...
✓	2m20ISD_15	Complete	25	25 of 25	2m20ISD_15	1	4	1997,2

Experiment Exp_C4, Scenario 12m120tBalanced_12, replication 25 completed at simulation time 0,35. Actual run time: 118 seconds.
Experiment Exp_C4, Scenario 12m120tBalanced_13, replication 25 completed at simulation time 0,35. Actual run time: 116,7 seconds.
Experiment Exp_C4, Scenario 12m120tBalanced_14, replication 25 completed at simulation time 0,35. Actual run time: 115,8 seconds.
Experiment Exp_C4, Scenario 12m120tBalanced_15, replication 25 completed at simulation time 0,35. Actual run time: 114,1 seconds.
Run completed in 199 819,5 seconds.

Figura 28 – Experimentos e replicações para os sequenciamentos

Foram criados 5 experimentos, cada um representando um tipo de sequenciamento, e dentro dessa função é possível configurar a quantidade de replicações a serem realizadas, em (*Replications - Required*).

Para todas as instâncias foi calculado o desvio percentual médio dos valores dos sequenciamentos comparando com os valores de ACO II, de acordo com a Equação 1, a seguir:

Equação 1- Cálculo do desvio médio dos tempos

$$\frac{C \max(S) - C \max(ACO II)}{C \max(ACO II)} * 100\% \quad (1)$$

4.2 Definição do número de replicações

Foram realizados experimentos considerando (5, 10, 15, 20, 25) replicações, para que se pudesse chegar em um valor satisfatório. A Tabela 9 e a Tabela 10 a seguir, mostram os valores médios de desvio percentual das replicações realizadas para cada sequenciamento, comparado com os valores de ACO II.

Tabela 9 – Média total do desvio dos sequenciamentos (*Balanced*)

Pi j, Si j balanced					
R	S0	S1	S2	S3	S4
5R	0,258	0,273	0,104	0,284	0,093
10R	3,034	4,092	1,560	4,260	1,361
15R	0,202	0,273	0,104	0,284	0,092
20R	0,258	0,273	0,104	0,284	0,093
25R	0,258	0,273	0,104	0,284	0,093

Tabela 10 – Média total do desvio dos sequenciamentos (*Setup Dominant*)

Si j dominant					
R	S0	S1	S2	S3	S4
5R	0,104	0,144	0,014	0,152	0,006
10R	1,311	2,155	0,203	2,273	0,092
15R	0,087	0,144	0,014	0,152	0,005
20R	0,104	0,144	0,014	0,152	0,005
25R	0,104	0,144	0,014	0,152	0,005

Através desses valores, é possível identificar valores que sejam satisfatórios para o resultado desejado. Portanto, verifica-se que tanto para os cenários *Balanced* como para os cenários *Setup Dominant*, quando se chega em 20 replicações que os valores se estabilizam.

O Gráfico 1 e o Gráfico 2 mostram a variação dos valores, de característica *Balanced* e *Setup Dominant*, respectivamente, onde é possível comparar os valores apresentados nas tabelas anteriores, encontrados através dos experimentos e replicações, e através dos gráficos é possível identificar o ponto de estabilização em 20 replicações.

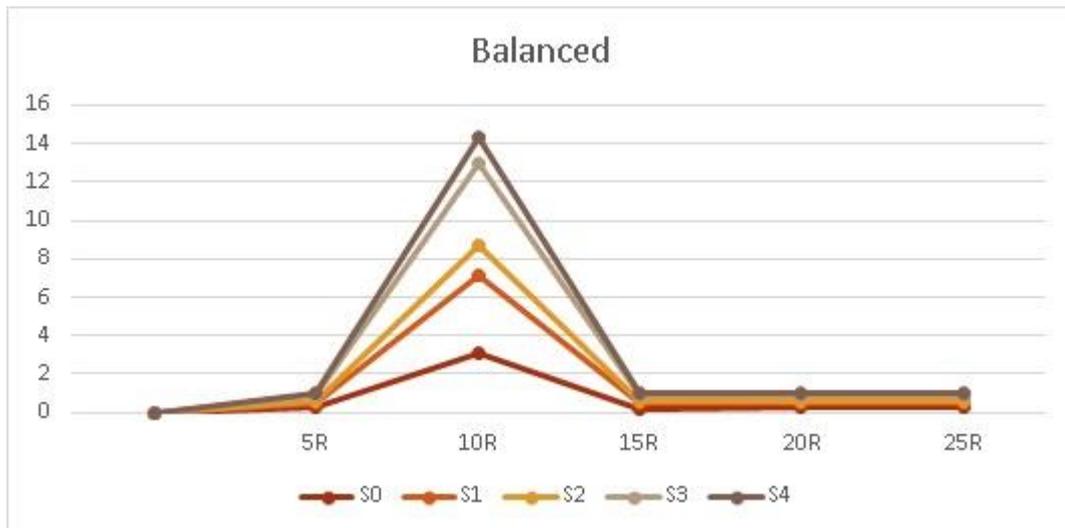


Gráfico 1- Gráfico comparativo do sequenciamento (*Balanced*)

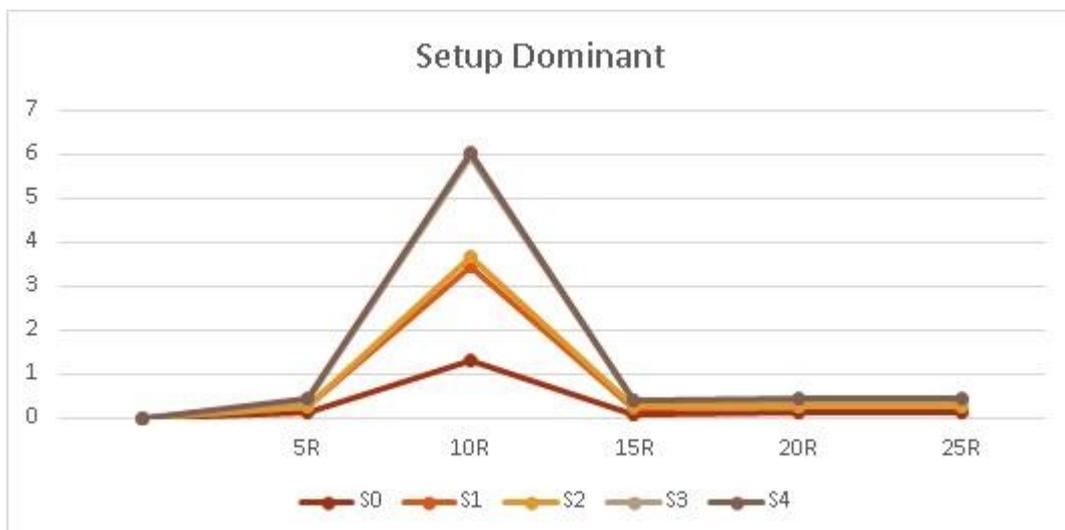


Gráfico 2- Gráfico comparativo do sequenciamento (*Setup Dominant*)

O sequenciamento S4 que representa a regra de sequenciamento que prioriza os trabalhos com menor tempo de *setup*, apresentou o menor desvio comparado com os valores de ACO II.

Dado este cenário, foi analisado também os valores de tempo de processamento, obtidos com os valores das 20 replicações da simulação de todos os cenários, para todos os tipos sequenciamento sugeridos, na busca de encontrar os menores valores de tempo total de processamento (*Makespan*) para cada cenário.

4.3 Análise do ACO II com regras de sequenciamento

Realizando a comparação dos valores de tempo total de processamento, é possível verificar que para a condição de cenários *Balanced*, os tempos de processamento total encontrados através do método ACO II, na maioria dos cenários, apresentaram valores mais satisfatórios, como pode ser visualizado nas (Tabela 12 a Tabela 16). No entanto, quando considerado 20 trabalhos para (8, 10, 12) máquinas, o sequenciamento S2 que prioriza o menor tempo de processamento, apresentou valores mais satisfatórios que o ACO II, como pode ser visualizado na Tabela 11.

Tabela 11 – Comparação de ACO II e Sequenciamento *Balanced* para 20 trabalhos

Balanced						
20 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	1235	1456	1526	1354	1610	1291
4	609	719	770	657	774	650
6	446	508	562	461	555	472
8	340	381	421	337	407	356
10	246	261	285	223	268	253
12	234	249	276	215	254	239

Tabela 12 – Comparação de ACO II e Sequenciamento *Balanced* para 40 trabalhos

Balanced						
40 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	2395	2947	3089	2812	3332	2559
4	1167	1467	1597	1332	1640	1287
6	792	987	1091	882	1097	872
8	581	730	799	632	792	653
10	476	579	630	494	622	523
12	437	516	588	460	570	470

Tabela 13– Comparação de ACO II e Sequenciamento *Balanced* para 60 trabalhos

Balanced						
60 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	3565	4458	4689	4199	5070	3811
4	1726	2224	2442	2008	2513	1919
6	1148	1480	1624	1312	1663	1286
8	880	1122	1247	984	1264	982
10	689	887	977	782	971	780
12	573	737	806	629	800	653

Tabela 14 – Comparação de ACO II e Sequenciamento *Balanced* para 80 trabalhos

Balanced						
80 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	4723	5940	6262	5624	6800	5067
4	2289	2967	3225	2677	3386	2546
6	1530	1982	2191	1757	2260	1710
8	1131	1483	1652	1297	1673	1284
10	903	1189	1336	1044	1324	1034
12	773	996	1119	862	1115	870

Tabela 15 – Comparação de ACO II e Sequenciamento *Balanced* para 100 trabalhos

Balanced						
100 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	5882	7422	7848	6966	8532	6301
4	2838	3707	4070	3341	4258	3171
6	1882	2477	2748	2195	2837	2123
8	1422	1859	2089	1635	2121	1599
10	1126	1491	1677	1293	1681	1285
12	969	1249	1412	1087	1423	1085

Tabela 16 – Comparação de ACO II e Sequenciamento *Balanced* para 120 trabalhos

Balanced						
120 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	7073	8930	9364	8422	10290	7568
4	3390	4451	4895	4021	5126	3790
6	2234	2972	3307	2620	3400	2541
8	1670	2234	2488	1954	2546	1914
10	1336	1786	2002	1560	2028	1534
12	1113	1494	1688	1294	1683	1289

Para a condição de cenário em que os valores de tempo de *setup* são maiores que os valores de tempo de processamento (*Setup Dominant*), os tempos de processamento total utilizando o método ACO II, na maioria dos cenários, também apresentaram valores satisfatórios.

No entanto, a Tabela 17 mostra que no cenário de 20 trabalhos considerando 2 e 4 máquinas, o sequenciamento S4 apresentou melhores resultados, e para (6, 8, 10, 12) máquinas o sequenciamento S2 que apresentou os menores valores de tempo total de processamento.

A Tabela 18 mostra que considerando 40 trabalhos os resultados satisfatórios de menor tempo total de processamento foi quando se utilizou para (2, 4, 6) máquinas o ACO II, para 8 máquinas o sequenciamento S4 e para (10 e 12) máquinas o sequenciamento S2.

A Tabela 19 mostra que considerando 60 trabalhos, os resultados satisfatórios de menor tempo total de processamento foi quando se utilizou para (2, 4, 6, 8 e 10) máquinas o ACO II, e para 12 máquinas o sequenciamento S2.

A Tabela 20, Tabela 21 e Tabela 22 mostram os cenários de (80, 100 e 120) trabalhos, respectivamente, e para essa condição os resultados de menor tempo de processamento total foi encontrado em ACO II.

Tabela 17 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 20 trabalhos

Setup Dominant						
20 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	1983	2128	2190	2041	2278	1964
4	984	1016	1062	953	1072	948
6	745	731	787	692	779	694
8	568	530	571	491	555	505
10	400	337	356	301	342	327
12	383	324	350	292	330	314

Tabela 18 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 40 trabalhos

Setup Dominant						
40 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	3888	4385	4516	4206	4762	3983
4	1921	2152	2276	2011	2315	1963
6	1319	1437	1541	1327	1547	1324
8	961	1032	1095	958	1092	952
10	778	805	856	726	847	746
12	738	740	812	684	795	696

Tabela 19 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 60 trabalhos

Setup Dominant						
60 Trabalhos						
<i>m</i>	ACOII	S0	S1	S2	S3	S4
2	5815	6628	6858	6387	7247	6002
4	2862	3291	3512	3072	3575	2979
6	1904	2169	2313	1994	2340	1968
8	1479	1641	1771	1513	1785	1503
10	1132	1268	1360	1145	1348	1157
12	950	1039	1109	929	1102	955

Tabela 20 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 80 trabalhos

Setup Dominant						
80 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	7707	8860	9189	8543	9711	7984
4	3793	4407	4674	4131	4819	3986
6	2565	2942	3165	2727	3233	2678
8	1889	2177	2329	1991	2347	1972
10	1501	1726	1842	1578	1850	1565
12	1297	1448	1574	1310	1565	1323

Tabela 21 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 100 trabalhos

Setup Dominant						
100 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	9712	11110	11541	10727	12217	9994
4	4718	5539	5902	5183	6069	4994
6	3155	3675	3949	3378	4029	3323
8	2402	2761	2993	2538	3024	2506
10	1877	2182	2347	1975	2357	1972
12	1636	1844	2017	1691	2024	1685

Tabela 22 – Comparação de ACO II e Sequenciamento *Setup Dominant* para 120 trabalhos

Setup Dominant						
120 Trabalhos						
m	ACOII	S0	S1	S2	S3	S4
2	11648	13360	13854	12845	14727	12018
4	5646	6651	7074	6214	7309	5989
6	3741	4419	4749	4096	4840	3984
8	2807	3308	3552	3023	3606	2981
10	2244	2635	2828	2390	2858	2381
12	1861	2186	2353	1966	2359	1974

Através dos resultados apresentados nas tabelas anteriores, é possível identificar que para os cenários de características *Balanced*, o ACOII representa 92% dos cenários com o menor tempo total de processamento, enquanto o sequenciamento S2 apresenta 8%.

Para os cenários de características *Setup Dominant*, o ACOII representa 72% dos cenários com o menor tempo total de processamento, enquanto o sequenciamento S2 representa 19% e o sequenciamento S4 representa 8%.

Portanto, de forma geral é possível analisar que o menor tempo total de processamento foi encontrado em ACOII com 82% em cenários, seguido do sequenciamento S2 em 14%, e o sequenciamento S4 com 4%.

A Tabela 23 e Tabela 24 mostram todos os valores dos tempos de processamento total. É possível identificar através das cores, divididos numa escala de cores variando de vermelho → amarelo → verde, onde o verde escuro representa os menores valores de tempo de processamento total e o vermelho os maiores valores de tempo de processamento total. Nota-se que os valores encontrados no sequenciamento S4 estão mais próximos dos valores de ACO II, seguido do sequenciamento S2.

Tabela 23 – Comparação de ACO II e os sequenciamentos de características *Balanced*

Balanced							
Nº Máquinas	Nº Trabalhos	ACOII	S0	S1	S2	S3	S4
2	20	1235	1456	1526	1354	1610	1291
	40	2395	2947	3089	2812	3332	2559
	60	3565	4458	4689	4199	5070	3811
	80	4723	5940	6262	5624	6800	5067
	100	5882	7422	7848	6966	8532	6301
	120	7073	8930	9364	8422	10290	7568
4	20	609	719	770	657	774	650
	40	1167	1467	1597	1332	1640	1287
	60	1726	2224	2442	2008	2513	1919
	80	2289	2967	3225	2677	3386	2546
	100	2838	3707	4070	3341	4258	3171
	120	3390	4451	4895	4021	5126	3790
6	20	446	508	562	461	555	472
	40	792	987	1091	882	1097	872
	60	1148	1480	1624	1312	1663	1286
	80	1530	1982	2191	1757	2260	1710
	100	1882	2477	2748	2195	2837	2123
	120	2234	2972	3307	2620	3400	2541
8	20	340	381	421	337	407	356
	40	581	730	799	632	792	653
	60	880	1122	1247	984	1264	982
	80	1131	1483	1652	1297	1673	1284
	100	1422	1859	2089	1635	2121	1599
	120	1670	2234	2488	1954	2546	1914
10	20	246	261	285	223	268	253
	40	476	579	630	494	622	523
	60	689	887	977	782	971	780
	80	903	1189	1336	1044	1324	1034
	100	1126	1491	1677	1293	1681	1285
	120	1336	1786	2002	1560	2028	1534
12	20	234	249	276	215	254	239
	40	437	516	588	460	570	470
	60	573	737	806	629	800	653
	80	773	996	1119	862	1115	870
	100	969	1249	1412	1087	1423	1085
	120	1113	1494	1688	1294	1683	1289

Tabela 24 – Comparação de ACO II e os sequenciamentos de características *Setup Dominant*

Setup Dominant							
Nº Máquinas	Nº Trabalhos	ACOII	S0	S1	S2	S3	S4
2	20	1983	2128	2190	2041	2278	1964
	40	3888	4385	4516	4206	4762	3983
	60	5815	6628	6858	6387	7247	6002
	80	7707	8860	9189	8543	9711	7984
	100	9712	11110	11541	10727	12217	9994
	120	11648	13360	13854	12845	14727	12018
4	20	984	1016	1062	953	1072	948
	40	1921	2152	2276	2011	2315	1963
	60	2862	3291	3512	3072	3575	2979
	80	3793	4407	4674	4131	4819	3986
	100	4718	5539	5902	5183	6069	4994
	120	5646	6651	7074	6214	7309	5989
6	20	745	731	787	692	779	694
	40	1319	1437	1541	1327	1547	1324
	60	1904	2169	2313	1994	2340	1968
	80	2565	2942	3165	2727	3233	2678
	100	3155	3675	3949	3378	4029	3323
	120	3741	4419	4749	4096	4840	3984
8	20	568	530	571	491	555	505
	40	961	1032	1095	958	1092	952
	60	1479	1641	1771	1513	1785	1503
	80	1889	2177	2329	1991	2347	1972
	100	2402	2761	2993	2538	3024	2506
	120	2807	3308	3552	3023	3606	2981
10	20	400	337	356	301	342	327
	40	778	805	856	726	847	746
	60	1132	1268	1360	1145	1348	1157
	80	1501	1726	1842	1578	1850	1565
	100	1877	2182	2347	1975	2357	1972
	120	2244	2635	2828	2390	2858	2381
12	20	383	324	350	292	330	314
	40	738	740	812	684	795	696
	60	950	1039	1109	929	1102	955
	80	1297	1448	1574	1310	1565	1323
	100	1636	1844	2017	1691	2024	1685
	120	1861	2186	2353	1966	2359	1974

No entanto, as regras que são utilizadas no ACO II seguem um padrão diferente dos que são apresentados nas regras dos sequenciamentos que foram propostos para este trabalho. Realizou-se, então, uma análise considerando apenas as cinco regras de sequenciamento descritas na Tabela 6.

4.4 Análise entre as regras de sequenciamento

Para os cenários com características de tempo de processamento e tempo de *setup* balanceados (*Balanced*), os valores de menor tempo total de processamento estão distribuídos entre o sequenciamento S2 que prioriza o menor tempo de processamento, e o sequenciamento S4 que prioriza o menor tempo de *setup*.

A Tabela 25 mostra que para 20 trabalhos, os resultados foram satisfatórios quando se utilizou para 2 e 4 máquinas o sequenciamento S4, e para (6, 8, 10, 12) máquinas sequenciamento S2.

A Tabela 26 mostra que para 40 trabalhos, os resultados foram satisfatórios quando se utilizou para (2, 4, 6) máquinas, o sequenciamento S4 e para (8, 10, 12) máquinas, o sequenciamento S2.

A Tabela 27 e Tabela 28 que representam 60 e 80 trabalhos, respectivamente, mostram que os resultados satisfatórios de menor tempo total de processamento foram quando se utilizou para (2, 4, 6, 8, 10) máquinas, o sequenciamento S4, e para 12 máquinas o sequenciamento S2.

A Tabela 29 e Tabela 30 que apresentam 100 e 120 trabalhos, respectivamente, mostra que nessa condição, para todas as quantidades de máquinas (2, 4, 6, 8, 10, 12), o sequenciamento S4 apresentou os resultados mais satisfatórios.

Tabela 25 – Comparação entre os Sequenciamentos *Balanced* para 20 trabalhos

Balanced					
20 Trabalhos					
<i>m</i>	S0	S1	S2	S3	S4
2	1456	1526	1354	1610	1291
4	719	770	657	774	650
6	508	562	461	555	472
8	381	421	337	407	356
10	261	285	223	268	253
12	249	276	215	254	239

Tabela 26 – Comparação entre os Sequenciamentos *Balanced* para 40 trabalhos

Balanced					
40 Trabalhos					
m	S0	S1	S2	S3	S4
2	2947	3089	2812	3332	2559
4	1467	1597	1332	1640	1287
6	987	1091	882	1097	872
8	730	799	632	792	653
10	579	630	494	622	523
12	516	588	460	570	470

Tabela 27 – Comparação entre os Sequenciamentos *Balanced* para 60 trabalhos

Balanced					
60 Trabalhos					
m	S0	S1	S2	S3	S4
2	4458	4689	4199	5070	3811
4	2224	2442	2008	2513	1919
6	1480	1624	1312	1663	1286
8	1122	1247	984	1264	982
10	887	977	782	971	780
12	737	806	629	800	653

Tabela 28 – Comparação entre os Sequenciamentos *Balanced* para 80 trabalhos

Balanced					
80 Trabalhos					
m	S0	S1	S2	S3	S4
2	5940	6262	5624	6800	5067
4	2967	3225	2677	3386	2546
6	1982	2191	1757	2260	1710
8	1483	1652	1297	1673	1284
10	1189	1336	1044	1324	1034
12	996	1119	862	1115	870

Tabela 29 – Comparação entre os Sequenciamentos *Balanced* para 100 trabalhos

Balanced					
100 Trabalhos					
m	S0	S1	S2	S3	S4
2	7422	7848	6966	8532	6301
4	3707	4070	3341	4258	3171
6	2477	2748	2195	2837	2123
8	1859	2089	1635	2121	1599
10	1491	1677	1293	1681	1285
12	1249	1412	1087	1423	1085

Tabela 30 – Comparação entre os Sequenciamentos *Balanced* para 120 trabalhos

Balanced					
120 Trabalhos					
m	S0	S1	S2	S3	S4
2	8930	9364	8422	10290	7568
4	4451	4895	4021	5126	3790
6	2972	3307	2620	3400	2541
8	2234	2488	1954	2546	1914
10	1786	2002	1560	2028	1534
12	1494	1688	1294	1683	1289

Para os cenários em que os valores de tempo de *setup* são maiores que os valores de tempo de processamento (*Setup Dominant*), os resultados apresentados estão mais distribuídos que os cenários anteriores.

A Tabela 31 que representa o cenário com 20 trabalhos, mostra que os resultados satisfatórios de menor tempo total de processamento foram quando se utilizou para 2 e 4 máquinas o sequenciamento S4, e para (6, 8, 10 e 12) máquinas o sequenciamento S2.

A Tabela 32 e Tabela 33 que representam os cenários com 40 e 60 trabalhos, respectivamente, mostram que os resultados satisfatórios foram quando se utilizou para (2, 4, 6, 8) máquinas o sequenciamento S4, e para (10 e 12) máquinas o sequenciamento S2.

A Tabela 34 e a Tabela 36 que representam os cenários com 80 e 120 trabalhos, respectivamente, mostram que os resultados satisfatórios foram quando se utilizou para (2, 4, 6, 8, 10) máquinas o sequenciamento S4, e para 12 máquinas o sequenciamento S2.

A Tabela 35 que representa o cenário com 100 trabalhos, somente o sequenciamento S4 que prioriza o menor tempo total de processamento, apresentou resultados satisfatórios.

Tabela 31 – Comparação entre os Sequenciamentos de *Setup Dominant* para 20 trabalhos

Setup Dominant					
20 Trabalhos					
m	S0	S1	S2	S3	S4
2	2128	2190	2041	2278	1964
4	1016	1062	953	1072	948
6	731	787	692	779	694
8	530	571	491	555	505
10	337	356	301	342	327
12	324	350	292	330	314

Tabela 32 – Comparação entre os Sequenciamentos de *Setup Dominant* para 40 trabalhos

Setup Dominant					
40 Trabalhos					
m	S0	S1	S2	S3	S4
2	4385	4516	4206	4762	3983
4	2152	2276	2011	2315	1963
6	1437	1541	1327	1547	1324
8	1032	1095	958	1092	952
10	805	856	726	847	746
12	740	812	684	795	696

Tabela 33 – Comparação entre os Sequenciamentos de *Setup Dominant* para 60 trabalhos

Setup Dominant					
60 Trabalhos					
m	S0	S1	S2	S3	S4
2	6628	6858	6387	7247	6002
4	3291	3512	3072	3575	2979
6	2169	2313	1994	2340	1968
8	1641	1771	1513	1785	1503
10	1268	1360	1145	1348	1157
12	1039	1109	929	1102	955

Tabela 34 – Comparação entre os Sequenciamentos de *Setup Dominant* para 80 trabalhos

Setup Dominant					
80 Trabalhos					
<i>m</i>	S0	S1	S2	S3	S4
2	8860	9189	8543	9711	7984
4	4407	4674	4131	4819	3986
6	2942	3165	2727	3233	2678
8	2177	2329	1991	2347	1972
10	1726	1842	1578	1850	1565
12	1448	1574	1310	1565	1323

Tabela 35 – Comparação entre os Sequenciamentos de *Setup Dominant* para 100 trabalhos

Setup Dominant					
100 Trabalhos					
<i>m</i>	S0	S1	S2	S3	S4
2	11110	11541	10727	12217	9994
4	5539	5902	5183	6069	4994
6	3675	3949	3378	4029	3323
8	2761	2993	2538	3024	2506
10	2182	2347	1975	2357	1972
12	1844	2017	1691	2024	1685

Tabela 36 – Comparação entre os Sequenciamentos de *Setup Dominant* para 120 trabalhos

Setup Dominant					
120 Trabalhos					
<i>m</i>	S0	S1	S2	S3	S4
2	13360	13854	12845	14727	12018
4	6651	7074	6214	7309	5989
6	4419	4749	4096	4840	3984
8	3308	3552	3023	3606	2981
10	2635	2828	2390	2858	2381
12	2186	2353	1966	2359	1974

Portanto, para os cenários apresentados nas tabelas anteriores que consideram somente os sequenciamentos propostos na Tabela 6, é possível identificar que para os cenários de características

Balanced, o sequenciamento S4 representa 75% dos cenários com o menor tempo total de processamento, enquanto o sequenciamento S2 apresenta 25%.

Para os cenários de características *Setup Dominant*, o sequenciamento S4 representa 72% dos cenários com o menor tempo total de processamento, enquanto o sequenciamento S2 representa 28%.

A Tabela 37 e Tabela 38 mostram todos os valores dos tempos de processamento total, e é possível identificar através da mesma escala de cores usada anteriormente, que varia de vermelho → amarelo → verde, onde o verde escuro representa os menores valores de tempo de processamento total e o vermelho os maiores valores de tempo de processamento total. Nota-se que os sequenciamentos S2 e S4 possuem os menores valores de tempo total de processamento.

Portanto, de forma geral é possível identificar que o sequenciamento S4 apresentou 74% dos menores valores de tempo total de processamento, e sequenciamento S2 apresentou 26%, ou seja, os sequenciamentos com as regras de restrição *Shortest Processing Time* (SPT) e *Shortest Setup Time* (SST) apresentaram resultados satisfatórios para a proposta deste trabalho.

Tabela 37– Comparação de tempos de processamento total entre os sequenciamentos (Balanced)

Balanced						
Nº Máquinas	Nº Trabalhos	S0	S1	S2	S3	S4
2	20	1456	1526	1354	1610	1291
	40	2947	3089	2812	3332	2559
	60	4458	4689	4199	5070	3811
	80	5940	6262	5624	6800	5067
	100	7422	7848	6966	8532	6301
	120	8930	9364	8422	10290	7568
4	20	719	770	657	774	650
	40	1467	1597	1332	1640	1287
	60	2224	2442	2008	2513	1919
	80	2967	3225	2677	3386	2546
	100	3707	4070	3341	4258	3171
	120	4451	4895	4021	5126	3790
6	20	508	562	461	555	472
	40	987	1091	882	1097	872
	60	1480	1624	1312	1663	1286
	80	1982	2191	1757	2260	1710
	100	2477	2748	2195	2837	2123
	120	2972	3307	2620	3400	2541
8	20	381	421	337	407	356
	40	730	799	632	792	653
	60	1122	1247	984	1264	982
	80	1483	1652	1297	1673	1284
	100	1859	2089	1635	2121	1599
	120	2234	2488	1954	2546	1914
10	20	261	285	223	268	253
	40	579	630	494	622	523
	60	887	977	782	971	780
	80	1189	1336	1044	1324	1034
	100	1491	1677	1293	1681	1285
	120	1786	2002	1560	2028	1534
12	20	249	276	215	254	239
	40	516	588	460	570	470
	60	737	806	629	800	653
	80	996	1119	862	1115	870
	100	1249	1412	1087	1423	1085
	120	1494	1688	1294	1683	1289

Tabela 38– Comparação de tempos de processamento total entre os sequenciamentos (*Setup Dominant*)

Setup Dominant						
Nº Máquinas	Nº Trabalhos	S0	S1	S2	S3	S4
2	20	2128	2190	2041	2278	1964
	40	4385	4516	4206	4762	3983
	60	6628	6858	6387	7247	6002
	80	8860	9189	8543	9711	7984
	100	11110	11541	10727	12217	9994
	120	13360	13854	12845	14727	12018
4	20	1016	1062	953	1072	948
	40	2152	2276	2011	2315	1963
	60	3291	3512	3072	3575	2979
	80	4407	4674	4131	4819	3986
	100	5539	5902	5183	6069	4994
	120	6651	7074	6214	7309	5989
6	20	731	787	692	779	694
	40	1437	1541	1327	1547	1324
	60	2169	2313	1994	2340	1968
	80	2942	3165	2727	3233	2678
	100	3675	3949	3378	4029	3323
	120	4419	4749	4096	4840	3984
8	20	530	571	491	555	505
	40	1032	1095	958	1092	952
	60	1641	1771	1513	1785	1503
	80	2177	2329	1991	2347	1972
	100	2761	2993	2538	3024	2506
	120	3308	3552	3023	3606	2981
10	20	337	356	301	342	327
	40	805	856	726	847	746
	60	1268	1360	1145	1348	1157
	80	1726	1842	1578	1850	1565
	100	2182	2347	1975	2357	1972
	120	2635	2828	2390	2858	2381
12	20	324	350	292	330	314
	40	740	812	684	795	696
	60	1039	1109	929	1102	955
	80	1448	1574	1310	1565	1323
	100	1844	2017	1691	2024	1685
	120	2186	2353	1966	2359	1974

5. CONCLUSÃO

O principal objetivo deste trabalho é estudar o problema de escalonamento de máquinas paralelas não relacionadas com tempo de processamento e tempo de *setup* dependente, com o auxílio do *software* de simulação SIMIO, considerando dois cenários: o primeiro caracterizado por de tempos de processamento e tempos de *setup* balanceados (*Balanced*); e o segundo caracterizado por tempo de *setup* maior que o tempo de processamento (*Setup Dominant*), tendo como referência os estudos feitos por Rabadi, et al. (2014).

Na revisão da literatura, foi apresentada as definições sobre o escalonamento e sua complexidade, métodos de otimização, regras de sequenciamento, modelagem e simulação.

5.1 Simulação

Este mesmo problema foi anteriormente analisado por outros autores utilizando diferentes métodos como: o *Tabu Search* de Helal, et al. (2006), *Meta-RaPS* de Rabadi, et al. (2006), *ACO I* de Rabadi, et al. (2010), *Simulated Annealing* (SA), e *Restrictive Simulated Annealing* (RSA) de Ying, et al. (2010).

Para este trabalho, foi escolhido o *software* de simulação SIMIO para realizar a simulação destes dados, possibilitando a modelação de diferentes cenários, tornando os resultados confiáveis.

A modelação no *software* SIMIO foi desenvolvida, considerando os métodos de sequenciamento comumente usados, como o LPT (*Longest Process Time*), SPT (*Shortest Process Time*), SIRO (*Service Random in Order*) e SST (*Shortest Setup Time*). O número de máquinas, número de trabalhos, valores de tempo de processamento e tempos de *setup* foram os mesmos utilizados por Rabadi, et al. (2014), por forma a permitir a comparação.

5.2 Regras de sequenciamento

As cinco regras de sequenciamento selecionadas para o desenvolvimento deste trabalho foram: S0- Sequenciamento aleatório; S1- Sequenciamento que prioriza o maior tempo de processamento dos trabalhos; S2- Sequenciamento que prioriza o menor tempo de processamento dos trabalhos; S3- Sequenciamento que prioriza o maior tempo de *setup* das máquinas; S4- Sequenciamento que prioriza o menor tempo de *setup* das máquinas.

Como resultado da simulação, identificou-se que os valores de tempo total de processamento encontrados através do algoritmo ACO II (Rabadi, et al., 2014) foram menores quando comparados com

os cinco sequenciamentos propostos. No entanto, a diferença foi pequena comparada com os valores encontrados no sequenciamento S4, que prioriza os trabalhos com o menor tempo de *setup* (SST).

Fez-se, então, uma comparação entre os cinco sequenciamentos escolhidos para este trabalho, e foi identificado que o sequenciamento que prioriza os trabalhos com menor tempo de *setup* (SST) continuou apresentando resultados satisfatórios.

A exceção foi vista nos cenários que possuem o maior número de máquinas (12), onde o sequenciamento que prioriza o trabalho com o menor tempo de processamento S2 apresentou melhores resultados.

5.3 Trabalhos futuros

Para problemas de escalonamento, os *softwares* de simulação apresentam uma ampla gama de possibilidades de modelação que os tornam cada vez mais eficientes e competitivos no mercado. A facilidade de adaptação para diversos cenários dá ao usuário a liberdade de simular processos em diferentes condições, de forma simplificada se comparado com métodos matemáticos.

Futuramente, seria interessante analisar os cenários para a condição de tempo de processamento maiores que o tempo de *setup* (*Process Dominante*), e definir outras regras de sequenciamento para os primeiros trabalhos que entram na máquina que não dependem de tempo de *setup*, com o intuito de encontrar a melhor combinação de regras de sequenciamento para redução do tempo total de processamento.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abreu, J. C.; Pereira, A. A. S. (2019) Meta-heurística multiobjetivo para sequenciamento de máquinas paralelas não relacionadas com tempos de preparação dependentes da sequência. *Revista Científica FAGOC-Multidisciplinar*, v. 3, n.1, p. 31-43.
- Adams, J., Balas, E., Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34, 391–401.
- Al-Salem, A. (2004). A heuristic to minimize makespan in proportional parallel flow shops. *International Journal of Computing & Information Sciences*, 2(2), 98.
- Al-Salem, A. (2004). Scheduling to minimize makespan on unequal parallel machines with sequence dependent setup times. *Engineering Journal of the University of Qatar*, 17, 177–187.
- Baker, K. R. (1974). *Introdução ao sequenciamento e programação*. Retrieved from https://openlibrary.org/books/OL5047415M/Introduction_to_sequencing_and_scheduling
- Baker, K. R., Trietsch, D. (2009). *Principles of Sequencing and Scheduling*, (1st Ed) John Wiley.
- Banks, J.; Carson II, J.S. & Nelson, B.L. (1996) *Discrete-event system simulation*. 2. ed., New Jersey: Prentice Hall.
- Banks, J. (1998). *Handbook of Simulation – Principles, methodology, advances, applications and practice*. New York: John Wiley & Sons.
- Banks, J. (1999). Introduction to simulation. Paper presented at the Winter Simulation Conference Proceedings.
- Barbosa, E., Costa, M. G., Fatima, M., Souza, D., Pereira, F. H. (2010), 'Evaluation of Production Sequencing Rules in Job Shop and Flow Shop Environment through Computer Simulation', *Icieceom*.

- Bartosiewicz Netto, F. (2014). Otimização do tempo total de trabalho na programação da produção em oficina de máquinas com tempo de preparação dependente da sequência. 2014. 66 f. Tese (Doutorado em Métodos Numéricos em Engenharias) – Universidade Federal do Paraná, Curitiba.
- Bender, M. A.; Rabin, M. (2002). Online scheduling of parallel programs on heterogeneous systems with applications to cilk. *Theory of Computing Systems*, v. 35, n. 3, p. 289-304.
- Błażewicz, J., Machowiak, M., Węglarz, J., Kovalyov, M. Y. & Trystram, D. (2004). Scheduling Malleable Tasks on Parallel Processors to Minimize the Makespan. *Annals of Operations Research*, 129, 65-80.
- Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 268–308. Retrieved from https://www.researchgate.net/profile/Christian_Blum/publication/221900771_Metaheuristics_in_Combinatorial_Optimization_Overview_and_Conceptual_Comparison/links/02bfe510587dd1ca65000000.pdf
- Carson II, J. S. (2004). Introduction to modeling and simulation. In: *Winter Simulation Conference*, 2004.
- Centeno, M. A.; Carrillo, M. (2001). Challenges of introducing simulation as a decision making tool. In: *Winter Simulation Conference*.
- Costa, A. C. F.; Jungles, A. E. (2006). O mapeamento do fluxo de valor aplicado a uma fábrica de montagem de canetas simulada. In: *Encontro nacional de engenharia de produção*. 26., 09 a 11 de outubro de 2006. Anais..., Fortaleza, CE: ABEPRO.
- Cui, M., Bai, R., Lu, Z., Li, X., Aickelin, U., & Ge, P. (2019). Regular Expression Based Medical Text Classification Using Constructive Heuristic Approach. *IEEE Access*.
- Freitas Filho, P. J. (2008). *Introdução a Modelagem e Simulação de Sistemas com Aplicações em Arena*. 2.ed. Florianópolis: Visual Books.
- French, S. (1982). *Sequencing and Scheduling: An introduction to the Mathematics of the Job Shop*. Chichester: Ellis Horwood.
- Gendreau, M., & Potvin, J. (2010). *Handbook of Metaheuristics* (3 ed). Springer.

- Glover, F., Kochenberger, G. A. (2003). Handbook of Metaheuristics. International Series in Operations Research and management Science. Stanford University. Springer.
- Gomes. (2007). Reactive Scheduling in make to order production systems: An Optimization based approach. Dissertação para a Obtenção do grau de Doutor em Engenharia de Sistemas. Instituto Superior Técnico, Universidade Técnica de Lisboa.
- Gupta, J.N.D.; Tunc, E.A. (1994). Scheduling a two-stage hybrid flowshop with separable setup and removal times. European Journal of Operational Research, v. 77, p. 415-428.
- Habiba, H.; Ahmed, H.; Souad, T.; Zaki, S. (2018). Minimizing Makespan on Identical Parallel Machines. Electrotehnica, Electronica, Automatica, v. 66, n. 1, p. 180-185.
- Helal, M., Rabadi, G., & Al-Salem, A. (2006). A tabu search algorithm to minimize the makespan for unrelated parallel machines Scheduling problem with setup times. International Journal of Operations Research, 3(3), 182–192.
- Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. SIAM J. Comput.2, 88-105.
- Izmailov, A. F., Solodov, M. V. (2014). Newton-type methods for optimization and variational problems. Springer serie in operation research and finalcial engineering. Springer, heidelberg, New York..
- Kelton, W. D.; Sadowski, R., P.; Sadowski, D. A. (2000). Simulation with ARENA. 2nd. ed. Columbus, Ohio: McGraw Hill.
- Law, A. M., Kelton, W. D. (1991). Simulation Modeling & Analysis (2nd ed.). Singapore: McGraw-Hill.
- Law, A.M. & Kelton, W.D. (2000). Simulation Modeling and Analysis (3. ed.), New York: McGraw-Hill.
- Li, S. (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. European Journal of Operational Research, v. 102, p. 142-156.
- Loch, G. V. (2010). Mensuração e previsão do custo da falha interna via modelo de simulação. 2010 143 f. Dissertação (Mestrado) - Curso de Pósgraduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba.
- Lustosa L.; Mesquita, M. A.; Quelhas, O.; Oliveira, R. (2008). Planejamento e Controle da Produção. Editora Elsevier.

- Marques, J. A. de S. (2015). Heurísticas de pesquisa local para problemas de máquina única. Dissertação de mestrado. Instituto Superior de tecnologia do Porto.
- Mendis, H. R.; Audsley, N. C.; Indrusiak, L. S. (2017). Dynamic and static task allocation for hard real-time video stream decoding on NoCs. LITES: Leibniz Transactions on Embedded Systems, v. 4, n. 2, p. 1-25.
- Monden, Y. (2012). Toyota Production System: An Integrated Approach to Just-intime. CRC Press.
- Moreira, D. (2011). Administração da Produção e Operações. São Paulo: Cengage Learning.
- Morton, T. E., Pentico, D. W. (1993). Heuristic Scheduling Systems. Proceedings of the IEEE, Vol. 84, p. 324. <https://doi.org/10.1109/jproc.1996.482234>.
- Mota, I. F.; Guasch, A.; Mota, M.M.;Piera, M. A. (2017) Robust Modelling and Simulation. Integration of SIMIO with Coloured Petri Nets. Springer International Publishing AG.
- Mundim, L. R.; Fuchigami, H. Y. (2017). Uma heurística robusta para programação de máquinas paralelas com tempos de setup dependentes da sequência. Revista Produção Online, v. 17, n. 2, p. 463-481.
- Osman, M. (2011). Designing Machine Learning Tools Based on Meta-Heuristic Programming. Retrieved from https://www.researchgate.net/publication/278404748_Designing_Machine_Learning_Tools_Based_on_Meta-Heuristic_Programming.
- Paiva, A. (2005). Geração automática de modelos de simulação de uma linha de produção na indústria têxtil. Dissertação de mestrado, Universidade do Minho, Portugal.
- Pedgen, C.D. (2007). SIMIO: a new simulation system based on intelligent objects, in: Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best Is Yet to Come, pp. 2293–2300.
- Pedgen, C. D.; Shannon, R.E. & Sadowski, R.P. (1995). Introduction to Simulation Using SIMAN. 2. ed., New York: McGraw-Hill.
- Pedgen, C. D.; Sturrock, D. T. (2014). Rapid Modeling Solutions: Introduction to Simulation and SIMIO 1. ed.

- Pereira, A.M.D.M. (2003). Aplicação de meta-heurística ao problema de escalonamento em ambiente dinâmico de produção discreta. Tese de doutoramento. Universidade do Minho.
- Pidd, M. (1998). *Computer Simulation in Management Science*. 4th Edition. New York: John Wiley & Sons Ltd., 280p.
- Pinedo, M. L. (2004). Planning and scheduling in manufacturing and services. *Project Planning and Sheduling*, Capítulo 4, p. 51–79. Springer, Nova Iorque.
- Pinedo, M. L. (2008). *Scheduling: Theory, Algorithms and Systems*. New York: Springer.
- Pinedo, M. L., (2009). *Planning and Scheduling in Manufacturing and Services*. New York: Springer, 2. ed.
- Pinedo, M. L. (2012). *Scheduling*. New York: Springer, 4. ed.
- Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* Fifth ed. Cham: Springer International Publishing.
- Poça, P.S.F. (2015). Utilização de Modelos de Simulação para a Melhoria de um Sistema de Restauração – Estudo de Caso de uma Cantina Universitária. Dissertação de mestrado, Universidade do Minho, Portugal.
- Pontes, H. L. J.; Yamada, M. C.; Porto, A. J. V. (2007). Análise do arranjo físico de uma linha de montagem em uma empresa do setor de componentes automotivos utilizando simulação. In: Congresso ibero americano de engenharia mecânica. 8., 23 a 25 de outubro de 2007. Anais... Cuzco, Peru: CIBIM.
- Portugal, V., Trietsch, D. (1998). Makespan-related criteria for comparing schedules in stochastic environments. *J. Oper. Res. Soc.* 49 (11), 1188–1195
- Rabadi, G., Moraga, R., & Al-Salem, A. (2006). Heuristics for the Unrelated Parallel Machine Scheduling Problem with Setup Times. *Journal of Intelligent Manufacturing*, 17, 85–97.

- Rabadi, G., Arnaout, J. P., & Musa, R. (2010). A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 21(6), 693–701. <https://doi.org/10.1007/s10845-009-0246-1>.
- Rabadi, G., Arnaout, J. P., & Musa, R. (2014). A two-stage Ant Colony optimization algorithm to minimize the makespan on unrelated parallel machines - Part II: Enhancements and experimentations. *Journal of Intelligent Manufacturing*, 25(1), 43–53. <https://doi.org/10.1007/s10845-012-0672-3>.
- Reis, P. C. S. O. (2020). Ferramenta de apoio ao escalonamento da produção. Dissertação de mestrado. Instituto Superior de Engenharia do Porto.
- Saliby, E. (1989). Repensando a simulação: a amostragem descritiva. São Paulo: Atlas.
- Schrijver, A. (2003). Otimização combinatória: poliedros e eficiência. Springer Science & Business Media, v. 24.
- SchedulingResearch. (2005). <http://SchedulingResearch.com>. Accessed in November, 2021.
- Shannon, R. E. (1998). Introduction to the Art and Science of Simulation. Em Proceedings of the 30th Conference on Winter Simulation (pp. 7–14). Los Alamitos, CA, USA: IEEE Computer Society Press. Obtido de <http://dl.acm.org/citation.cfm?id=293172.293175>.
- Silva, T. C.; Fuchigami, H. Y. (2018). Desenvolvimento e análise de regras de sequenciamento da produção para minimização de medidas de atraso em Flow Shop com tempos de setup. *Revista Produção Online*. Florianópolis, SC, v. 18, n. 2, p. 454-478.
- SIMIO LLC. (2018). Obtido de <https://www.SIMIO.com/> <https://www.SIMIO.com/resources/white-papers/Introduction-to-SIMIO>.
- Sturrock, D. T., Smith, J. S., Kelton, W. D. (2018). *Simio and Simulation: Modeling, Analysis, Applications*, 5ed.
- Tavares, H. M. G. (2015). Estudo e Análise do Sequenciamento de Tarefas de Produção: Job Shop scheduling. Dissertação (Mestrado). Programa de Pós-Graduação em Engenharia e Gestão Industrial, Instituto Politécnico do Porto, Vila do Conde, Portugal.

- Teixeira, F. d. O., Oliveira, L., Varela, L. R. (2014). Comparative analysis of Scheduling rules through arena for parallel machines', 2014 6th World Congress on Nature and Biologically Inspired Computing, NaBIC 2014 pp. 170{175.
- Tubino, D. F. (2007). Planejamento e Controle da Produção: Teoria e Prática. São Paulo: Atlas.
- Varela, M. L. R. (2007). Uma contribuição para o escalonamento da produção baseado em métodos globalmente distribuídos. Tese de doutoramento (270 pgs), Universidade do Minho.
- Veríssimo, J. M. F. (2016). Loteamento e sequenciamento da produção: um caso de estudo na indústria de injeção de plásticos. Dissertação (Mestrado), Programa de Pós-graduação em Engenharia e Gestão Industrial, Universidade de Coimbra.
- White, R. E., Pearson, J. N., Wilson, J. R., Pidd, M. (1999). "Systems just modeling through: a rough guide to modeling", In- terfaces Archive, Vol.29, No.2.
- Xiao, L.; Song, S.; Chen, X.; Coit, D. W. (2016). Joint optimization of production scheduling and machine group preventive maintenance. Reliability Engineering & System Safety, v. 146, p. 68-78.
- Yang, X.-S. (2010b). Nature-inspired Metaheuristic Algorithms (L. Press, Ed.). Retrieved from https://www.researchgate.net/profile/Xin-She_Yang/publication/235979455_Nature-Inspired-Metaheuristic-Algorithms/links/0c96051520ff6131a0000000/Nature-Inspired-Metaheuristic-Algorithms.pdf
- Ying, K.-C., Lee, Z.-J., & Lin, S.-W. (2010). Makespan minimization for scheduling unrelated parallel machines with setup times. Journal of Intelligent Manufacturing (online first).
- Zarei, M.; Dawari, M.; Kolahan, F.; Wong, K. (2016). Simultaneous selection and scheduling with sequence-dependent setup times, lateness penalties, and machine availability constraint: Heuristic approaches. International Journal of Industrial Engineering Computations, v. 7, n. 1, p. 147-160.
- Zhao, Y., Xu, X., Li, H., Liu, Y. (2018). Stochastic customer order scheduling with setup times to minimize expected cycle time. Int. J. Prod. Res. 56 (7), 2684– 2706.

Zhou, S.; Liu, M.; Chen, H.; LI, X. (2016). An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. *International Journal of Production Economics*, v. 179, p. 1-11.

APÊNDICE 1 – RESULTADOS DE DESVIO PERCENTUAL MÉDIO PARA 5 REPLICAÇÕES

<i>m</i>	<i>n</i>	<i>P_{ij}, S_{ij} balanced</i>					<i>S_{ij} dominant</i>				
		<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>
2	20	0,179	0,190	0,087	0,231	0,041	0,074	0,095	0,028	0,130	-0,009
	40	0,231	0,224	0,148	0,282	0,064	0,128	0,139	0,075	0,183	0,025
	60	0,253	0,240	0,151	0,296	0,063	0,140	0,152	0,090	0,198	0,032
	80	0,259	0,246	0,160	0,306	0,068	0,150	0,161	0,098	0,206	0,035
	100	0,262	0,250	0,156	0,311	0,066	0,144	0,158	0,095	0,205	0,029
	120	0,262	0,245	0,160	0,312	0,065	0,148	0,159	0,093	0,209	0,031
4	20	0,172	0,208	0,070	0,212	0,063	0,029	0,073	-0,033	0,083	-0,037
	40	0,255	0,269	0,124	0,287	0,092	0,121	0,156	0,044	0,168	0,020
	60	0,286	0,293	0,140	0,314	0,100	0,149	0,185	0,068	0,200	0,041
	80	0,297	0,290	0,145	0,323	0,101	0,161	0,188	0,082	0,213	0,047
	100	0,309	0,303	0,150	0,334	0,105	0,172	0,201	0,090	0,222	0,056
	120	0,313	0,307	0,157	0,338	0,105	0,177	0,202	0,091	0,227	0,057
6	20	0,122	0,206	0,031	0,195	0,053	-0,024	0,053	-0,079	0,042	-0,072
	40	0,249	0,274	0,102	0,277	0,094	0,089	0,144	0,005	0,147	0,002
	60	0,288	0,293	0,125	0,311	0,105	0,138	0,176	0,045	0,187	0,032
	80	0,294	0,301	0,129	0,322	0,104	0,147	0,189	0,060	0,206	0,042
	100	0,317	0,315	0,142	0,336	0,112	0,165	0,201	0,066	0,217	0,049
	120	0,330	0,324	0,147	0,343	0,120	0,183	0,212	0,087	0,227	0,060
8	20	0,122	0,191	-0,013	0,168	0,048	-0,065	0,004	-0,157	-0,021	-0,121
	40	0,257	0,272	0,080	0,270	0,112	0,077	0,123	-0,003	0,121	-0,009
	60	0,276	0,293	0,105	0,304	0,103	0,106	0,165	0,022	0,171	0,018
	80	0,312	0,315	0,128	0,324	0,118	0,152	0,189	0,051	0,195	0,042
	100	0,306	0,319	0,130	0,330	0,113	0,149	0,198	0,053	0,206	0,042
	120	0,336	0,328	0,145	0,344	0,127	0,177	0,210	0,071	0,222	0,057
10	20	0,065	0,139	-0,107	0,087	0,028	-0,157	-0,123	-0,330	-0,173	-0,226
	40	0,218	0,244	0,035	0,236	0,092	0,037	0,090	-0,073	0,080	-0,041
	60	0,296	0,295	0,119	0,292	0,115	0,122	0,167	0,011	0,160	0,020
	80	0,314	0,324	0,135	0,319	0,125	0,150	0,185	0,049	0,188	0,041
	100	0,328	0,328	0,128	0,330	0,122	0,164	0,200	0,049	0,204	0,049
	120	0,338	0,332	0,143	0,343	0,131	0,175	0,206	0,061	0,215	0,057
12	20	0,059	0,152	-0,094	0,079	0,014	-0,161	-0,095	-0,316	-0,164	-0,223
	40	0,181	0,256	0,048	0,234	0,080	0,001	0,092	-0,078	0,073	-0,058
	60	0,288	0,289	0,088	0,285	0,126	0,094	0,143	-0,023	0,137	0,008
	80	0,288	0,309	0,103	0,306	0,113	0,119	0,176	0,010	0,170	0,021
	100	0,290	0,314	0,108	0,319	0,110	0,129	0,189	0,033	0,190	0,030
	120	0,343	0,340	0,139	0,339	0,134	0,174	0,209	0,053	0,211	0,056
		0,258	0,273	0,104	0,284	0,093	0,104	0,144	0,014	0,152	0,006

Figura 29 – Comparação de desvio percentual médio para 5 replicações

APÊNDICE 2 – RESULTADOS DE DESVIO PERCENTUAL MÉDIO PARA 10 REPLICAÇÕES

<i>m</i>	<i>n</i>	<i>P_{ij}, S_{ij} balanced</i>					<i>S_{ij} dominant</i>				
		<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>
2	20	2,261	2,853	1,305	3,540	0,626	1,019	1,421	0,424	1,896	-0,168
	40	2,832	3,367	2,219	4,284	1,006	1,688	2,084	1,128	2,756	0,393
	60	3,013	3,594	2,261	4,481	0,966	1,839	2,280	1,343	2,979	0,517
	80	3,080	3,685	2,400	4,564	1,007	1,954	2,417	1,466	3,079	0,511
	100	3,107	3,756	2,333	4,654	0,987	1,895	2,376	1,418	3,050	0,457
	120	3,121	3,669	2,403	4,681	1,032	1,915	2,388	1,397	3,128	0,444
4	20	2,231	3,123	1,054	2,966	0,846	0,404	1,090	-0,499	1,269	-0,618
	40	3,077	4,032	1,856	4,313	1,452	1,610	2,334	0,663	2,498	0,292
	60	3,351	4,397	2,106	4,704	1,461	1,967	2,771	1,022	2,982	0,604
	80	3,433	4,353	2,171	4,838	1,452	2,087	2,825	1,226	3,186	0,636
	100	3,532	4,542	2,256	5,012	1,542	2,226	3,009	1,346	3,326	0,861
	120	3,580	4,611	2,354	5,071	1,543	2,267	3,027	1,370	3,413	0,814
6	20	1,768	3,089	0,472	2,931	0,496	-0,325	0,793	-1,180	0,647	-1,086
	40	2,983	4,109	1,525	4,043	1,331	1,217	2,157	0,078	2,194	0,026
	60	3,360	4,392	1,875	4,628	1,603	1,822	2,646	0,668	2,775	0,450
	80	3,408	4,521	1,928	4,836	1,522	1,919	2,842	0,893	3,075	0,629
	100	3,610	4,723	2,133	5,045	1,655	2,118	3,018	0,993	3,242	0,710
	120	3,730	4,865	2,204	5,148	1,788	2,319	3,183	1,298	3,402	0,904
8	20	1,609	2,865	-0,189	2,545	0,612	-1,045	0,060	-2,352	-0,390	-1,952
	40	3,065	4,087	1,199	4,036	1,619	1,048	1,840	-0,051	1,823	-0,029
	60	3,230	4,402	1,579	4,555	1,596	1,456	2,470	0,331	2,557	0,282
	80	3,561	4,727	1,913	4,898	1,870	1,978	2,839	0,772	2,929	0,587
	100	3,526	4,787	1,955	4,935	1,629	1,938	2,963	0,802	3,049	0,587
	120	3,785	4,927	2,178	5,212	1,864	2,273	3,145	1,072	3,307	0,850
10	20	0,855	2,089	-1,610	1,345	0,008	-2,789	-1,844	-4,956	-2,255	-3,480
	40	2,667	3,657	0,528	3,589	1,459	0,506	1,356	-1,091	1,257	-0,422
	60	3,396	4,427	1,778	4,313	1,712	1,590	2,503	0,163	2,397	0,447
	80	3,569	4,854	2,021	4,711	1,812	1,947	2,775	0,730	2,758	0,624
	100	3,674	4,924	1,925	4,946	1,672	2,103	3,004	0,741	3,035	0,709
	120	3,779	4,986	2,146	5,107	1,937	2,221	3,094	0,910	3,239	0,880
12	20	0,863	2,273	-1,417	1,163	0,335	-2,783	-1,419	-4,745	-2,509	-3,287
	40	2,314	3,840	0,718	3,445	1,312	0,044	1,375	-1,176	1,113	-0,724
	60	3,321	4,330	1,320	4,292	1,887	1,264	2,144	-0,348	2,046	0,237
	80	3,371	4,636	1,545	4,611	1,683	1,569	2,640	0,148	2,532	0,330
	100	3,347	4,705	1,624	4,794	1,590	1,687	2,829	0,488	2,877	0,441
	120	3,816	5,104	2,092	5,109	2,089	2,232	3,138	0,801	3,172	0,841
		3,034	4,092	1,560	4,260	1,361	1,311	2,155	0,203	2,273	0,092

Figura 30 – Comparação de desvio percentual para 10 replicações

APÊNDICE 3 – RESULTADOS DE DESVIO PERCENTUAL MÉDIO PARA 15 REPLICAÇÕES

<i>m</i>	<i>n</i>	<i>P_{ij}, S_{ij} balanced</i>					<i>S_{ij} dominant</i>				
		<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>
2	20	0,153	0,190	0,087	0,233	0,044	0,068	0,095	0,028	0,130	-0,008
	40	0,188	0,224	0,148	0,281	0,064	0,113	0,139	0,075	0,184	0,024
	60	0,200	0,240	0,151	0,297	0,064	0,122	0,152	0,090	0,197	0,032
	80	0,206	0,246	0,160	0,305	0,068	0,130	0,161	0,098	0,206	0,034
	100	0,207	0,250	0,156	0,311	0,066	0,126	0,158	0,095	0,205	0,028
	120	0,208	0,245	0,160	0,313	0,066	0,128	0,159	0,093	0,209	0,031
4	20	0,152	0,208	0,070	0,212	0,062	0,029	0,073	-0,033	0,083	-0,038
	40	0,204	0,269	0,124	0,288	0,093	0,107	0,156	0,044	0,169	0,021
	60	0,223	0,293	0,140	0,314	0,100	0,131	0,185	0,068	0,200	0,040
	80	0,229	0,290	0,145	0,324	0,101	0,140	0,188	0,082	0,213	0,048
	100	0,235	0,303	0,150	0,334	0,106	0,148	0,201	0,090	0,223	0,056
	120	0,238	0,307	0,157	0,338	0,106	0,151	0,202	0,091	0,227	0,057
6	20	0,119	0,206	0,031	0,196	0,054	-0,022	0,053	-0,079	0,043	-0,073
	40	0,198	0,274	0,102	0,278	0,092	0,081	0,144	0,005	0,148	0,003
	60	0,225	0,293	0,125	0,310	0,107	0,122	0,176	0,045	0,186	0,033
	80	0,228	0,301	0,129	0,323	0,104	0,128	0,189	0,060	0,207	0,042
	100	0,240	0,315	0,142	0,337	0,114	0,142	0,201	0,066	0,217	0,051
	120	0,249	0,324	0,147	0,343	0,120	0,154	0,212	0,087	0,227	0,061
8	20	0,106	0,191	-0,013	0,165	0,046	-0,071	0,004	-0,157	-0,020	-0,121
	40	0,205	0,272	0,080	0,267	0,112	0,069	0,123	-0,003	0,121	-0,009
	60	0,216	0,293	0,105	0,303	0,103	0,098	0,165	0,022	0,172	0,016
	80	0,238	0,315	0,128	0,325	0,120	0,132	0,189	0,051	0,195	0,042
	100	0,235	0,319	0,130	0,330	0,111	0,129	0,198	0,053	0,206	0,041
	120	0,253	0,328	0,145	0,344	0,126	0,152	0,210	0,071	0,222	0,058
10	20	0,059	0,139	-0,107	0,084	0,027	-0,187	-0,123	-0,330	-0,172	-0,226
	40	0,177	0,244	0,035	0,235	0,092	0,033	0,090	-0,073	0,081	-0,043
	60	0,225	0,295	0,119	0,290	0,117	0,108	0,167	0,011	0,160	0,022
	80	0,239	0,324	0,135	0,318	0,124	0,130	0,185	0,049	0,189	0,041
	100	0,245	0,328	0,128	0,330	0,123	0,140	0,200	0,049	0,204	0,048
	120	0,252	0,332	0,143	0,342	0,129	0,148	0,206	0,061	0,215	0,057
12	20	0,059	0,152	-0,094	0,079	0,018	-0,184	-0,095	-0,316	-0,161	-0,222
	40	0,154	0,256	0,048	0,233	0,073	0,003	0,092	-0,078	0,071	-0,060
	60	0,221	0,289	0,088	0,284	0,122	0,085	0,143	-0,023	0,137	0,005
	80	0,225	0,309	0,103	0,307	0,111	0,104	0,176	0,010	0,171	0,020
	100	0,224	0,314	0,108	0,319	0,108	0,112	0,189	0,033	0,191	0,029
	120	0,255	0,340	0,139	0,339	0,136	0,149	0,209	0,053	0,212	0,057
		0,202	0,273	0,104	0,284	0,092	0,087	0,144	0,014	0,152	0,005

Figura 31 – Comparação de desvio percentual médio para 15 replicações

APÊNDICE 4 – RESULTADOS DE DESVIO PERCENTUAL MÉDIO PARA 20 REPLICAÇÕES

<i>m</i>	<i>t</i>	<i>P_{ij}, S_{ij} balanced</i>					<i>S_{ij} dominant</i>				
		<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>
2	20	0,179	0,190	0,087	0,233	0,043	0,073	0,095	0,028	0,130	-0,010
	40	0,230	0,224	0,148	0,281	0,064	0,128	0,139	0,075	0,183	0,024
	60	0,250	0,240	0,151	0,297	0,065	0,140	0,152	0,090	0,198	0,031
	80	0,258	0,246	0,160	0,305	0,068	0,150	0,161	0,098	0,206	0,035
	100	0,262	0,250	0,156	0,311	0,067	0,144	0,158	0,095	0,205	0,028
	120	0,263	0,245	0,160	0,313	0,065	0,147	0,159	0,093	0,209	0,031
4	20	0,180	0,208	0,070	0,212	0,062	0,032	0,073	-0,033	0,082	-0,038
	40	0,258	0,269	0,124	0,289	0,093	0,120	0,156	0,044	0,170	0,021
	60	0,289	0,293	0,140	0,313	0,101	0,150	0,185	0,068	0,199	0,039
	80	0,296	0,290	0,145	0,324	0,101	0,162	0,188	0,082	0,213	0,048
	100	0,306	0,303	0,150	0,333	0,105	0,174	0,201	0,090	0,223	0,055
	120	0,313	0,307	0,157	0,339	0,106	0,178	0,202	0,091	0,228	0,057
6	20	0,140	0,206	0,031	0,197	0,054	-0,019	0,053	-0,079	0,043	-0,074
	40	0,246	0,274	0,102	0,278	0,092	0,089	0,144	0,005	0,147	0,003
	60	0,289	0,293	0,125	0,310	0,108	0,139	0,176	0,045	0,186	0,032
	80	0,295	0,301	0,129	0,323	0,105	0,147	0,189	0,060	0,207	0,042
	100	0,316	0,315	0,142	0,336	0,113	0,165	0,201	0,066	0,217	0,051
	120	0,330	0,324	0,147	0,343	0,121	0,181	0,212	0,087	0,227	0,061
8	20	0,119	0,191	-0,013	0,163	0,045	-0,067	0,004	-0,157	-0,023	-0,124
	40	0,257	0,272	0,080	0,266	0,111	0,074	0,123	-0,003	0,120	-0,009
	60	0,275	0,293	0,105	0,303	0,104	0,109	0,165	0,022	0,172	0,016
	80	0,311	0,315	0,128	0,324	0,119	0,153	0,189	0,051	0,195	0,042
	100	0,307	0,319	0,130	0,330	0,111	0,150	0,198	0,053	0,206	0,042
	120	0,338	0,328	0,145	0,344	0,127	0,179	0,210	0,071	0,222	0,058
10	20	0,063	0,139	-0,107	0,084	0,028	-0,157	-0,123	-0,330	-0,170	-0,225
	40	0,216	0,244	0,035	0,234	0,090	0,034	0,090	-0,073	0,081	-0,043
	60	0,289	0,295	0,119	0,291	0,117	0,119	0,167	0,011	0,160	0,021
	80	0,317	0,324	0,135	0,318	0,126	0,150	0,185	0,049	0,189	0,041
	100	0,323	0,328	0,128	0,330	0,124	0,163	0,200	0,049	0,204	0,048
	120	0,336	0,332	0,143	0,341	0,129	0,174	0,206	0,061	0,215	0,057
12	20	0,063	0,152	-0,094	0,080	0,020	-0,154	-0,095	-0,316	-0,161	-0,222
	40	0,181	0,256	0,048	0,233	0,071	0,003	0,092	-0,078	0,072	-0,060
	60	0,286	0,289	0,088	0,283	0,121	0,093	0,143	-0,023	0,137	0,005
	80	0,289	0,309	0,103	0,307	0,112	0,116	0,176	0,010	0,171	0,020
	100	0,289	0,314	0,108	0,319	0,107	0,127	0,189	0,033	0,192	0,029
	120	0,342	0,340	0,139	0,339	0,136	0,175	0,209	0,053	0,211	0,057
		0,258	0,273	0,104	0,284	0,093	0,104	0,144	0,014	0,152	0,005

Figura 32 – Comparação de desvio percentual médio para 20 replicações

APÊNDICE 5 – RESULTADOS DE DESVIO PERCENTUAL MÉDIO PARA 25 REPLICAÇÕES

<i>m</i>	<i>n</i>	<i>P_{ij}, S_{ij} balanced</i>					<i>S_{ij} dominant</i>				
		<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>
2	20	0,178	0,190	0,087	0,233	0,042	0,073	0,095	0,028	0,130	-0,010
	40	0,231	0,224	0,148	0,281	0,064	0,127	0,139	0,075	0,183	0,023
	60	0,250	0,240	0,151	0,297	0,064	0,140	0,152	0,090	0,198	0,031
	80	0,258	0,246	0,160	0,305	0,068	0,150	0,161	0,098	0,206	0,034
	100	0,262	0,250	0,156	0,310	0,066	0,144	0,158	0,095	0,205	0,028
	120	0,262	0,245	0,160	0,313	0,065	0,147	0,159	0,093	0,209	0,031
4	20	0,181	0,208	0,070	0,213	0,063	0,032	0,073	-0,033	0,082	-0,038
	40	0,257	0,269	0,124	0,288	0,093	0,121	0,156	0,044	0,170	0,021
	60	0,290	0,293	0,140	0,313	0,100	0,150	0,185	0,068	0,199	0,039
	80	0,296	0,290	0,145	0,324	0,101	0,162	0,188	0,082	0,213	0,048
	100	0,306	0,303	0,150	0,333	0,105	0,174	0,201	0,090	0,222	0,055
	120	0,313	0,307	0,157	0,339	0,105	0,178	0,202	0,091	0,228	0,057
6	20	0,139	0,206	0,031	0,197	0,054	-0,019	0,053	-0,079	0,044	-0,074
	40	0,246	0,274	0,102	0,278	0,091	0,089	0,144	0,005	0,147	0,002
	60	0,288	0,293	0,125	0,310	0,108	0,138	0,176	0,045	0,186	0,033
	80	0,294	0,301	0,129	0,323	0,105	0,147	0,189	0,060	0,206	0,042
	100	0,316	0,315	0,142	0,336	0,113	0,165	0,201	0,066	0,217	0,051
	120	0,331	0,324	0,147	0,343	0,121	0,182	0,212	0,087	0,227	0,061
8	20	0,120	0,191	-0,013	0,163	0,045	-0,067	0,004	-0,157	-0,023	-0,125
	40	0,259	0,272	0,080	0,266	0,111	0,073	0,123	-0,003	0,120	-0,010
	60	0,274	0,293	0,105	0,303	0,103	0,109	0,165	0,022	0,171	0,016
	80	0,312	0,315	0,128	0,324	0,119	0,153	0,189	0,051	0,195	0,042
	100	0,307	0,319	0,130	0,330	0,110	0,149	0,198	0,053	0,206	0,042
	120	0,338	0,328	0,145	0,344	0,127	0,179	0,210	0,071	0,222	0,058
10	20	0,065	0,139	-0,107	0,084	0,030	-0,157	-0,123	-0,330	-0,170	-0,224
	40	0,217	0,244	0,035	0,234	0,091	0,035	0,090	-0,073	0,080	-0,043
	60	0,288	0,295	0,119	0,290	0,118	0,119	0,167	0,011	0,159	0,022
	80	0,317	0,324	0,135	0,318	0,126	0,151	0,185	0,049	0,189	0,041
	100	0,323	0,328	0,128	0,330	0,124	0,163	0,200	0,049	0,203	0,048
	120	0,337	0,332	0,143	0,341	0,130	0,174	0,206	0,061	0,215	0,057
12	20	0,062	0,152	-0,094	0,079	0,019	-0,155	-0,095	-0,316	-0,163	-0,223
	40	0,179	0,256	0,048	0,233	0,070	0,002	0,092	-0,078	0,072	-0,061
	60	0,286	0,289	0,088	0,284	0,121	0,093	0,143	-0,023	0,137	0,005
	80	0,288	0,309	0,103	0,306	0,112	0,117	0,176	0,010	0,172	0,020
	100	0,289	0,314	0,108	0,319	0,108	0,127	0,189	0,033	0,191	0,029
	120	0,341	0,340	0,139	0,339	0,136	0,175	0,209	0,053	0,211	0,058
		0,258	0,273	0,104	0,284	0,093	0,104	0,144	0,014	0,152	0,005

Figura 33 – Comparação de desvio percentual médio para 25 replicações