

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**JUAN DAVID ORDÓÑEZ MUTIZ**

**SISTEMAS MULTI-AGENTES E  
COORDENAÇÃO BIO-INSPIRADA  
APLICADOS A SISTEMAS DE  
AUTOMAÇÃO DE MANUFATURA**

Porto Alegre  
2017

**JUAN DAVID ORDÓÑEZ MUTIZ**

**SISTEMAS MULTI-AGENTES E  
COORDENAÇÃO BIO-INSPIRADA  
APLICADOS A SISTEMAS DE  
AUTOMAÇÃO DE MANUFATURA**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

CO-ORIENTADOR: Prof. Dr. Renato Ventura Bayan Henriques

Porto Alegre  
2017

**JUAN DAVID ORDÓÑEZ MUTIZ**

**SISTEMAS MULTI-AGENTES E  
COORDENAÇÃO BIO-INSPIRADA  
APLICADOS A SISTEMAS DE  
AUTOMAÇÃO DE MANUFATURA**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira , UFRGS

Doutor pela Stuttgart University – Stuttgart, Alemanha

Banca Examinadora:

Prof. Dr. Rafael Heitor Bordini, PUC-RS

Doutor pela University College London – Londres, Reino Unido

Prof. Dr. Edison Pignaton de Freitas, UFRGS

Doutor pela Universidade de Halmstad – Halmstad, Suécia e pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Walter Fetter Lages, UFRGS

Doutor pelo Instituto Tecnológico de Aeronáutica – São José dos Campos, Brasil

Coordenador do PPGEE: \_\_\_\_\_

Prof. Dr. Valner João Brusamarello

Porto Alegre, dezembro de 2017.

## DEDICATÓRIA

À minha família, especialmente aos meus pais **Ana Mercedes** e **Luis Carlos**, minha irmã **María Alejandra** e minha avó **Ana** pelo seu apoio e sacrifícios...

Aos futuros interessados nesta área de pesquisa. Possa este trabalho ser de ajuda, esclarecer e contribuir nos seus conhecimentos. "*Ne frustra vixisse videar!*" *Tycho Brahe*.

## **AGRADECIMENTOS**

Aos professores Carlos Eduardo Pereira e Renato Ventura Bayan Henriques pelo seu apoio e orientações durante o mestrado.

Aos professores de engenharia química da Universidad Nacional de Colombia Paulo César Narváez e Juan Guillermo Cadavid Estrada e ao professor da Universidad Militar Nueva Granada Oscar Fernando Aviles pela inspiração para iniciar e continuar com meus estudos na área de automação industrial e ter me referenciado ao ingresso no PPGEE da UFRGS.

Aos professores do PPGEE, especialmente Sérgio Haffner e Edison Pignaton de Freitas pela orientação nos conceitos referentes a algoritmos bio-inspirados.

Aos colegas de estudo e demais amigos que me acompanharam durante o mestrado pelo seu auxílio e acolhida em Rio Grande do Sul. A João Alvarez Peixoto pelas informações referentes a sistemas Multi-Agentes em manufatura.

À CAPES pela provisão da bolsa durante o meu segundo ano de mestrado...

## RESUMO

A globalização econômica é um fenômeno que tem gerado um mercado que demanda produtos customizados, de alta qualidade e com custos de produção reduzidos para se manter competitivo. Este cenário exige o desenvolvimento de sistemas de produção do tipo "*mass customization*" que são sistemas que garantem um custo baixo de produção (como os tradicionais sistemas de produção em massa) e ao mesmo tempo possuem a flexibilidade requerida que permite a personalização individual dos produtos. A fim de implementar esse tipo de sistema, dispositivos com mais autonomia e capazes de se autogerenciar passam a ser desejados em vez dos sistemas convencionais com programação centralizada e arranjo de funcionalidades fixas. O conceito de sistemas multi-agentes propõe a descentralização do controle que o mercado requer e fornece aos componentes da planta uma variedade de novas funcionalidades que permitem otimizar seu desempenho. Isso permite que os equipamentos passem a se comportar como agentes de manufatura e possam negociar entre si os processos de manufatura. Porém, o desempenho destes sistemas está fortemente relacionado à coordenação entre os agentes e a qualidade dos dados usados na tomada de decisões, portanto, uma otimização bio-inspirada é proposta a fim de garantir um bom desempenho entre os dispositivos. Buscando fornecer dados para a análise de desempenho e potencialidades desta proposta, um estudo de caso de uma planta de manufatura sob uma plataforma multi-agente usando otimização bio-inspirada é estudada.

**Palavras-chave:** Mass customization, manufatura inteligente, automação e controle, sistemas multi-agentes, coordenação bio-inspirada.

## **ABSTRACT**

Economic globalization is a phenomenon that has created a market that demands customized products, with high quality at low prices to stay competitive. This situation requires the development of mass customization production systems which are systems who ensure a production low price (as the traditional mass production systems) and at the same time, owns the required flexibility to allow an individual product customization. In order to implement those systems, devices with more autonomy and able to manage themselves are desirable instead of the conventional systems with centralized programming and fixed functionalities. The concept of multi-agent systems, proposes the decentralization that the market requires, and offers to the plant devices a wide range of new features that allow to improve its performance, including that those devices behave as manufacturing agents allowing them to negotiate the required manufacturing processes among themselves. However, the performance of those systems is strongly related to the coordination among agents and the quality of the data used to make decisions, therefore, a bio-inspired optimization is proposed in order to guarantee a well performance among devices. Looking to provide data to performance analysis and potential of the proposal, a case study of a manufacturing plant using a multi-agent platform and bio-inspired optimization is studied.

**Keywords: Mass customization, intelligent manufacturing, automation and control, multi-agent systems, bio-inspired coordination.**

# SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES</b> . . . . .	9
<b>LISTA DE TABELAS</b> . . . . .	11
<b>LISTA DE ABREVIATURAS</b> . . . . .	12
<b>1 INTRODUÇÃO</b> . . . . .	13
1.1 <b>Motivação</b> . . . . .	13
1.2 <b>Objetivo Geral</b> . . . . .	15
1.3 <b>Objetivos Especificos</b> . . . . .	15
1.4 <b>Delimitação da Pesquisa</b> . . . . .	16
1.5 <b>Organização do Trabalho</b> . . . . .	16
<b>2 METODOLOGIA</b> . . . . .	17
2.1 <b>Esclarecimento da Pesquisa</b> . . . . .	19
2.2 <b>Estudo Descritivo I</b> . . . . .	19
2.3 <b>Estudo Prescritivo</b> . . . . .	20
2.4 <b>Estudo Descritivo II</b> . . . . .	21
<b>3 REFERENCIAL TEÓRICO</b> . . . . .	22
3.1 <b>Sistemas Agentes e Multi-agentes (MAS)</b> . . . . .	22
3.1.1 <b>Agentes de Software</b> . . . . .	22
3.1.2 <b>Holons</b> . . . . .	24
3.1.3 <b>Sistemas Multi-Agentes (MAS)</b> . . . . .	25
3.1.4 <b>Tecnologias de Suporte ao Desenvolvimento de Sistemas Agentes e Multi-Agentes (MAS)</b> . . . . .	26
3.2 <b>Coordenação Bio-Inspirada</b> . . . . .	30
3.2.1 <b>Estigmergia</b> . . . . .	30
3.2.2 <b>Algoritmo de Colônia de Formigas (ACO)</b> . . . . .	31
<b>4 TRABALHOS RELACIONADOS</b> . . . . .	33
4.1 <b>Estruturas Multiagente</b> . . . . .	33
4.1.1 <b>PROSA</b> . . . . .	33
4.1.2 <b>ADACOR</b> . . . . .	34
4.1.3 <b>PABADIS</b> . . . . .	34
4.1.4 <b>HCBA</b> . . . . .	34
4.1.5 <b>KASA</b> . . . . .	35
4.1.6 <b>Rockwell</b> . . . . .	35
4.1.7 <b>Peixoto</b> . . . . .	36

4.1.8	Resumo . . . . .	36
4.2	Otimizações implementadas em MAS . . . . .	38
<b>5</b>	<b>PROPOSTA DE USO DE MAS E ACO PARA SISTEMAS DE MANUFATURA . . . . .</b>	<b>40</b>
5.1	Visão Geral . . . . .	40
5.2	Arquitetura de Agentes . . . . .	40
5.3	Interação entre Agentes . . . . .	42
5.4	Mecanismo de Tomada de Decisões . . . . .	44
5.4.1	Alocação de Produtos . . . . .	45
5.4.2	Disseminação de Informações . . . . .	46
5.4.3	Exemplo do funcionamento do mecanismo de tomada de decisões . . . . .	47
5.5	Planejamento dos Estudos de Caso . . . . .	51
<b>6</b>	<b>IMPLEMENTAÇÃO, SIMULAÇÃO E VALIDAÇÃO EXPERIMENTAL . . . . .</b>	<b>53</b>
6.1	Implementação . . . . .	53
6.1.1	Agente <i>Staff</i> . . . . .	53
6.1.2	Agente Recurso . . . . .	64
6.1.3	Holon Produto . . . . .	64
6.2	Simulação . . . . .	65
6.3	Validação Experimental . . . . .	67
6.3.1	Métricas de comparação . . . . .	67
6.3.2	Análise dos resultados . . . . .	76
<b>7</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>77</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>78</b>
	<b>ANEXO A CONCENTRAÇÃO DE FEROMÔNIOS AO FINAL DOS TESTES. . . . .</b>	<b>82</b>
A.1	Testes Cenário I, Produto $p_1$ . . . . .	82
A.2	Testes Cenário I, Produto $p_2$ . . . . .	84
A.3	Testes Cenário II, Produto $p_1$ . . . . .	85
A.4	Testes Cenário II, Produto $p_2$ . . . . .	86

## LISTA DE ILUSTRAÇÕES

1	Trabalhos relacionados e proposta (Fonte: Próprio autor) . . . . .	14
2	Problema de Agendamento de Tarefas (JSP) (Fonte: Próprio autor) . .	15
3	Metodologia de Pesquisa em Projetos (traduzido de (BLESSING; CHAKRABARTI, 2009)) . . . . .	18
4	Agente. (UHRMACHER; WEYNS, 2009) . . . . .	23
5	Agente e Ambiente. (UHRMACHER; WEYNS, 2009) . . . . .	23
6	Arquitetura de um holon. (Fonte: Próprio autor) . . . . .	24
7	Comunicação entre agentes para executar o agendamento de tarefas (Fonte: Próprio autor) . . . . .	25
8	Representação da ontologia de gerenciamento de agentes. (BELLI- FEMINE; CAIRE; GREENWOOD, 2007) . . . . .	27
9	Inicialização da plataforma JADE. (Fonte: Próprio autor) . . . . .	29
10	Interface gráfica do RMA de JADE. (Fonte: Próprio autor) . . . . .	29
11	Configuração Experimental e Seleção de caminhos menores por uma colônia de <i>Linepitema humile</i> (DORIGO; BONABEAU; THERAU- LAZ, 2000). . . . .	31
12	Agentificação Proposta (Fonte: Próprio autor). . . . .	41
13	Estrutura do Holon Produto proposto (Fonte: Próprio autor). . . . .	42
14	Diagrama de sequência UML da Interação Proposta (Fonte: Próprio autor). . . . .	42
15	Interação proposta para alocação de produtos (Fonte: Próprio autor). .	45
16	Interação proposta para disseminação de informações (Fonte: Próprio autor). . . . .	47
17	Topologia demonstrativa de planta de manufatura com tempos de transporte entre recursos [s] (Fonte: Próprio autor). . . . .	47
18	Roleta para escolher entre os recursos candidatos para executar a ta- refa $T_1$ , partindo do recurso $R_0$ (Fonte: Próprio autor). . . . .	49
19	Topologia da planta de manufatura (Fonte: Próprio autor). . . . .	52
20	Console de gerenciamento de agentes (GUI do RMA) com agentes da proposta (Fonte: Próprio autor). . . . .	54
21	Interface gráfica do agente <i>staff</i> (Fonte: Próprio autor). . . . .	54
22	Informações apresentadas na interface gráfica (Fonte: Próprio autor). .	55
23	Posições estabelecidas na planta de manufatura simulada (Fonte: Pró- prio autor). . . . .	55
24	Passo 1: Mensagem de inicialização dos recursos (Fonte: Próprio autor). . . . .	56

25	Passo 2: Interface gráfica do agente <i>Staff</i> quando um produto é lançado nele (Fonte: Próprio autor). . . . .	57
26	Passo 3: Interface gráfica do agente <i>Staff</i> com informação do recurso alocado atualizada (Fonte: Próprio autor). . . . .	57
27	Passo 4: Interface gráfica do agente <i>Staff</i> com dois produtos na esteira central (Fonte: Próprio autor). . . . .	58
28	Passo 5: Interface gráfica do agente <i>Staff</i> com um recurso executando uma tarefa (Fonte: Próprio autor). . . . .	59
29	Passo 6: Interface gráfica do agente <i>Staff</i> com um recurso terminando uma tarefa (Fonte: Próprio autor). . . . .	60
30	Passo 7: Interface gráfica do agente <i>Staff</i> com dois recursos executando tarefas ao mesmo tempo. (Fonte: Próprio autor). . . . .	61
31	Passo 8: Interface gráfica do agente <i>Staff</i> depois de uma disseminação de informações. (Fonte: Próprio autor). . . . .	62
32	Passo 9: Interface gráfica do agente <i>Staff</i> depois de manufaturar os dois produtos $P1$ e $P2$ . (Fonte: Próprio autor). . . . .	63
33	Máquina de estados finita do agente recurso (Fonte: Próprio autor). . . . .	65
34	<i>Makespans</i> [s] obtidos no cenário I, produto $p_1$ , teste 1 (Fonte: Próprio autor). . . . .	71
35	<i>Makespans</i> [s] obtidos no cenário I, produto $p_2$ , teste 1 (Fonte: Próprio autor). . . . .	72
36	<i>Makespans</i> [s] obtidos no cenário I, produto $p_2$ , teste 2 (Fonte: Próprio autor). . . . .	73
37	<i>Makespans</i> [s] obtidos no cenário II, produto $p_1$ , teste 1 (Fonte: Próprio autor). . . . .	74
38	<i>Makespans</i> [s] obtidos no cenário II, produto $p_2$ , teste 1 (Fonte: Próprio autor). . . . .	76

## LISTA DE TABELAS

1	Arquiteturas MAS em Manufatura e tipos de agente implementados . . .	37
2	Otimizações em manufatura . . . . .	39
3	Parâmetros definidos para exemplificar o mecanismo de tomada de decisões. . . . .	47
4	Condições iniciais dos tempos de transporte e processamento na planta demostrativa [s]. . . . .	48
5	Cenários concebidos para validar a proposta. . . . .	52
6	Parâmetros definidos para o mecanismo de tomada de decisões. . . .	66
7	Condições iniciais dos tempos de transporte entre recursos [s]. . . . .	66
8	Condições iniciais dos tempos de processamento nos cenários conce- bidos [s]. . . . .	66
9	Métricas de comparação. . . . .	68
10	Efeitos no sistema com retirada de estações. . . . .	69
11	Melhores soluções cenário I, produto tipo $p_1$ , $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ . . .	70
12	Resultados cenário I, produto $p_1$ . . . . .	70
13	Melhores soluções cenário I, produto tipo $p_2$ , $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ . . .	71
14	Resultados cenário I, produto $p_2$ . . . . .	72
15	Melhores soluções cenário II, produto tipo $p_1$ , $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ . . .	73
16	Resultados cenário II, produto $p_1$ . . . . .	74
17	Melhores soluções cenário II, produto tipo $p_2$ , $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ . . .	75
18	Resultados cenário II, produto $p_2$ . . . . .	75
19	Análise das métricas observadas. . . . .	76

## LISTA DE ABREVIATURAS

ACL	Agent Communications Language (Linguagem de Comunicação entre Agentes)
ACO	Ant Colony Optimization (Otimização por Colônia de Formigas)
AID	Agent Identifier (Identificador do Agente)
AMS	Agent Management System (Gerenciamento do Sistema de Agentes)
DF	Directory Facilitator (Postagem de Serviços)
DRM	Design Research Methodology (Metodologia de Pesquisa em Projetos)
FMS	Flexible Manufacturing System (Sistema de Manufatura Flexível)
JADE	JAVA Agent Development Framework (Plataforma de Desenvolvimento de Agentes)
JSP	Job Shop Scheduling Problem (Problema de Agendamento de Tarefas)
MAS	Multi-Agent System (Sistema Multi-Agente)
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
RMA	Remote Monitoring Agent (Agente de Monitoramento Remoto)
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)

# 1 INTRODUÇÃO

Os sistemas de controle de manufatura tradicionais possuem estruturas de controle centralizadas e hierárquicas que garantem uma ótima produção em larga escala. Estes sistemas se caracterizam pelo uso de grandes lotes de produção de peças idênticas e padronizadas, linhas com máquinas e processos que possuem altos custos de instalação e configuração, e limitações de robustez e flexibilidade. Durante anos, essas limitações foram ignoradas pois o mercado no qual atuavam não era dinâmico e garantia às indústrias a venda dos seus produtos com maior antecipação. No entanto, essas limitações vão se tornando cada vez mais críticas num mundo globalizado onde os usuários têm acesso a uma maior diversidade de produtos e estão na busca constante de produtos de baixo custo, altamente personalizados, de alta qualidade, e com curtos ciclos de vida. Já no final dos anos noventa, (PARUNAK, 1998) previa sobre o incremento na variedade e complexidade dos produtos demandados num mundo globalizado e as exigências que implicariam na indústria do futuro.

## 1.1 Motivação

O cenário descrito anteriormente demanda o desenvolvimento de sistemas de produção do tipo "*mass customization*", os quais são sistemas de manufatura que ao mesmo tempo que têm custo baixo de produção como os sistemas de produção em massa, são flexíveis de modo a permitir a customização individual dos produtos. Como apontam (LEITÃO; RESTIVO, 2002; LEITÃO, 2009; PEIXOTO, 2012) o paradigma de sistemas multi-agentes (MAS) foi desenvolvido a fim de atingir os níveis de flexibilidade, robustez e customização exigidos pelo mercado adicionando vantagens de modularidade, descentralização e autonomia nos componentes de controle da planta de manufatura. No entanto, este paradigma per se, não garante uma boa produção em massa e, dependendo da estrutura de controle implementada nele, também pode possuir um alto tráfego de dados que impeça a comunicação eficiente entre os agentes e incremente o consumo de energia necessário para efetuar essa comunicação.

A fim de executar uma busca ótima de soluções em uma linha de montagem, que procure a diminuição ou maximização de uma função objetivo, diversos tipos de otimizações heurísticas ou meta-heurísticas têm sido concebidos no estado da arte incluindo: Heurística gulosa (PEIXOTO, 2012), recozido simulado (AYDIN; FOGARTY, 2004), algoritmos genéticos (LIM; ZHANG, 2012), otimização multi-objetivo (JANA et al., 2013), abordagens bio-inspiradas (COLORNI et al., 1994; VALCKENAERS et al., 2004; XIANG; LEE, 2008; FREITAS, 2011; ZHANG; WONG, 2017)... etc. Destas abordagens, aquelas que implementam algum tipo de otimização meta-heurística garantem produções em massa ótima.

Do mesmo modo, para solucionar o problema de tráfego de dados associado aos sistemas multi-agentes, uma estrutura de controle heterárquica e um mecanismo de comunicação indireto entre agentes conhecido no estado da arte pelo nome de estigmergia podem ser implementados. Os trabalhos de (FREITAS, 2011; VALCKENAERS et al., 2004) aproveitam o mecanismo de estigmergia a fim de evitar problemas de tráfego de dados fazendo uma tomada de decisões com informações locais.

Pode-se dizer então, que é necessária a implementação de um sistema que procure solucionar os três problemas anteriormente mencionados, a saber: O problema de flexibilidade através do uso do paradigma de sistemas multi-agentes (MAS); O problema de otimização através do uso de uma meta-heurística (META-HEUR) e o problema de tráfego de dados associado aos sistemas multi-agentes através do mecanismo de estigmergia (ESTIG). A relação entre alguns dos trabalhos anteriormente citados pode ser visualizada na figura 1.

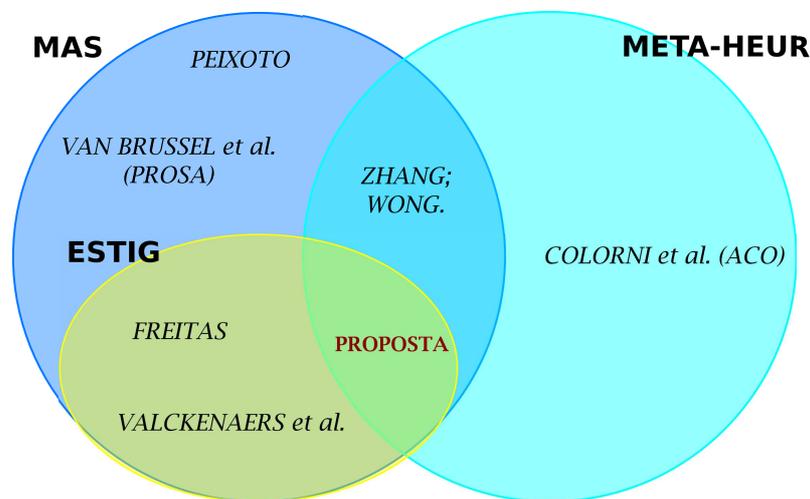


Figura 1: Trabalhos relacionados e proposta (Fonte: Próprio autor)

Um problema clássico de otimização em manufatura tratado neste trabalho é o Problema de Agendamento de Tarefas em Manufatura (JSP). O problema genérico de agendamento de tarefas em manufatura é associado à alocação de tarefas no chão de fábrica como é exibido na Figura 2. Considere um produto cuja produção consista de  $n$  tarefas ( $T_1, T_2, \dots, T_n$ ) que devem ser executadas em um conjunto de  $m$  recursos ( $R_1, R_2, \dots, R_m$ ) de um chão de fábrica os quais transformam um material simples em um produto acabado, sendo uma restrição que a quantidade de recursos disponíveis no chão de fábrica deve ser menor ou igual a  $m$ . Cada tarefa  $T_i$  deve ser alocada a um dos recursos, e devem ser executadas nos produtos uma por uma na sequência determinada de forma que as restrições de precedência não sejam violadas. Assumindo um chão de fábrica configurado de maneira que possua  $m$  recursos e um produto que precise de  $n$  tarefas, o máximo número de escolhas de recursos no problema é de  $m^n$  o qual aumenta exponencialmente com o número de tarefas requeridas pelo produto. Conseqüentemente, o problema a ser resolvido é fornecer uma estratégia para alocar todas as tarefas requeridas pelo produto nos recursos adequados pretendendo a otimização de uma função objetivo. Essa função objetivo, no caso do JSP, procura a diminuição do tempo total de manufatura ou *Makespan* em inglês.

Devido à finalidade que o problema possui de minimizar uma função objetivo, os métodos clássicos de programação dinâmica e linear poderiam ser considerados como os mais adequados. Porém, se sabe que o problema JSP possui uma complexidade exponen-

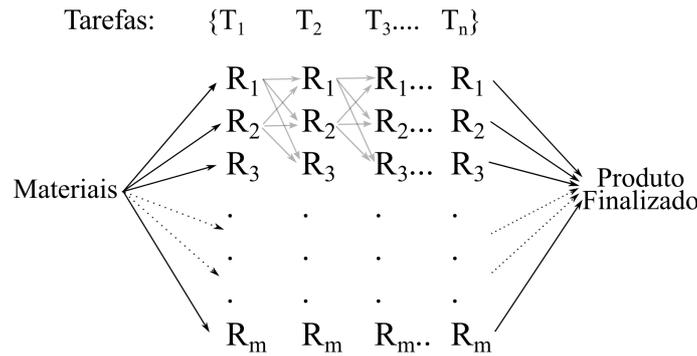


Figura 2: Problema de Agendamento de Tarefas (JSP) (Fonte: Próprio autor)

cial ( $m^n$ ) e é catalogado como um problema NP-difícil, isto faz com que esses métodos de programação per se precisem de tempos de execução polinomiais. Se o espaço de busca do problema for pequeno, os métodos de programação dinâmica e linear são satisfatórios, caso contrário, é necessária a integração desses métodos de programação a algoritmos de aproximação adequados que permitam atingir soluções ótimas sem a necessidade de explorar todo o espaço de busca (CORMEN, 2009). A proposta apresentada nesta dissertação usa uma meta-heurística bio-inspirada chamada de Otimização por Colonia de Formigas (ACO) devido a que pode ser implementada a fim de executar um método de aproximação chamado de "técnica de arredondamento" para lidar com problemas cujo espaço de busca é grande. Adicionalmente, pode ser aproveitado o fato que os métodos bio-inspirados podem ser associados e integrados facilmente aos métodos de programação dinâmica, linear, e ao mecanismo de estigmergia (PINTEA, 2014). Essas vantagens permitem executar uma tomada de decisões descentralizada e autônoma em cada um dos recursos aumentando o desempenho dadas as restrições de tempo e comunicação, e incrementando adicionalmente a fiabilidade do sistema (FREITAS, 2011).

## 1.2 Objetivo Geral

O presente trabalho tem como objetivo geral a apresentação de uma abordagem bio-inspirada para solucionar o problema de agendamento de tarefas derivado dos sistemas de manufatura flexível (FMS). Essa abordagem reúne os paradigmas de sistemas multi-agentes (MAS) e os mecanismos bio-inspirados que podem dar solução aos problemas de flexibilidade, otimização e tráfego de dados que a indústria e o mercado requer.

## 1.3 Objetivos Específicos

A fim de atingir o objetivo geral foram definidos os seguintes objetivos específicos:

- Estudar os principais conceitos associados aos paradigmas de MAS e mecanismos bio-inspirados.
- Pesquisar as características necessárias para concepção e simulação de uma planta de manufatura.
- Implementar uma solução ao problema de agendamento de tarefas (JSP) derivado dos sistemas de manufatura flexível (FMS).

- Validar a solução usando um software que permita simular sistemas de manufatura flexível e de suporte os paradigmas envolvidos na proposta.
- Analisar e avaliar itens de mensuração de desempenho nas plantas de manufatura.
- Avaliar e discutir os resultados encontrados.

## **1.4 Delimitação da Pesquisa**

Como foi descrito anteriormente, o objetivo geral da dissertação é propor uma abordagem bio-inspirada para solucionar o problema de agendamento de tarefas derivado dos sistemas de manufatura flexível (FMS) usando paradigmas que são necessários na indústria. Problemas como segurança nas redes de comunicação industrial (KNAPP; LANGILL, 2014), detecção de distúrbios em ambientes agressivos de montagem (CHIANG; RUSSELL; BRAATZ, 2000) e detalhamento de sistemas de transporte de materiais com suas respectivas características e tecnologias (CAVALCANTE, 2012) são assuntos importantes no chão de fábrica, porém esses assuntos estão fora do escopo da dissertação e ficarão abertos a trabalhos futuros.

## **1.5 Organização do Trabalho**

Esta dissertação está organizada da seguinte forma: Inicialmente é exposta a metodologia adotada para a execução do trabalho no Capítulo 2 e os seus estágios de desenvolvimento são descritos, posteriormente é abordado o referencial teórico no Capítulo 3 com ênfase nos paradigmas MAS e algoritmos bio-inspirados. No Capítulo 4 é feita descrição dos trabalhos relacionados que foram analisados durante o desenvolvimento da dissertação. Na sequência apresenta-se a proposta de uso de MAS e ACO para sistemas de manufatura no capítulo 5. No Capítulo 6 são implementados e simulados os diferentes estudos de caso propostos e os seus resultados. Finalmente as conclusões e trabalhos futuros são abordados no Capítulo 7.

## 2 METODOLOGIA

A metodologia adotada para execução deste trabalho de dissertação é descrita em (BLESSING; CHAKRABARTI, 2009), e é chamada de Metodologia de Pesquisa em Projetos (DRM). Foi desenvolvida pelos autores devido à ausência de terminologia comum, ferramentas investigação padronizadas e conseqüentemente uma metodologia normalizada de pesquisa em projetos. Esses problemas remetem à necessidade de criação de uma metodologia que garanta aos pesquisadores atingir objetivos como:

- Usar um *framework* de pesquisa em projetos tanto para pesquisadores individuais como para grupos de pesquisa.
- Ajudar na identificação de áreas de pesquisa e projetos que tenham mais probabilidade de ser promissórios e realistas em contextos acadêmicos e práticos.
- Fornecer uma ampla variedade de abordagens de investigação e métodos.
- Proporcionar diretrizes para o planejamento sistemático da pesquisa.
- Proporcionar diretrizes para uma pesquisa rigorosa.
- Ajudar no desenvolvimento de uma sólida linha de argumentação.
- Fornecer novos métodos e indicadores à metodologias existentes para executar os estágios do processo de pesquisa.
- Ajudar na seleção de métodos adequados e combinação de métodos.
- Proporcionar um contexto para posicionar os projetos de pesquisa e programas em relação a outros projetos de pesquisa.
- Incentivar a reflexão sobre a abordagem aplicada.

A Metodologia de Pesquisa em Projetos (DRM) possui como objetivos específicos os itens citados acima. Com essa finalidade, foram concebidos quatro estágios de desenvolvimento os quais podem ocorrer de maneira iterativa e cíclica conforme mostra a Figura 3. Cada um desses estágios possui entradas e saídas, as entradas correspondentes às ferramentas metodológicas usadas no estágio são exibidas à esquerda da Figura 3, e as saídas correspondentes aos resultados obtidos em cada estágio são apresentadas à direita dela.

O estágio de *Esclarecimento da Pesquisa* foca-se na busca de evidências ou indicações que apoiem a hipótese a fim de formular objetivos de pesquisa realistas e promissórios.

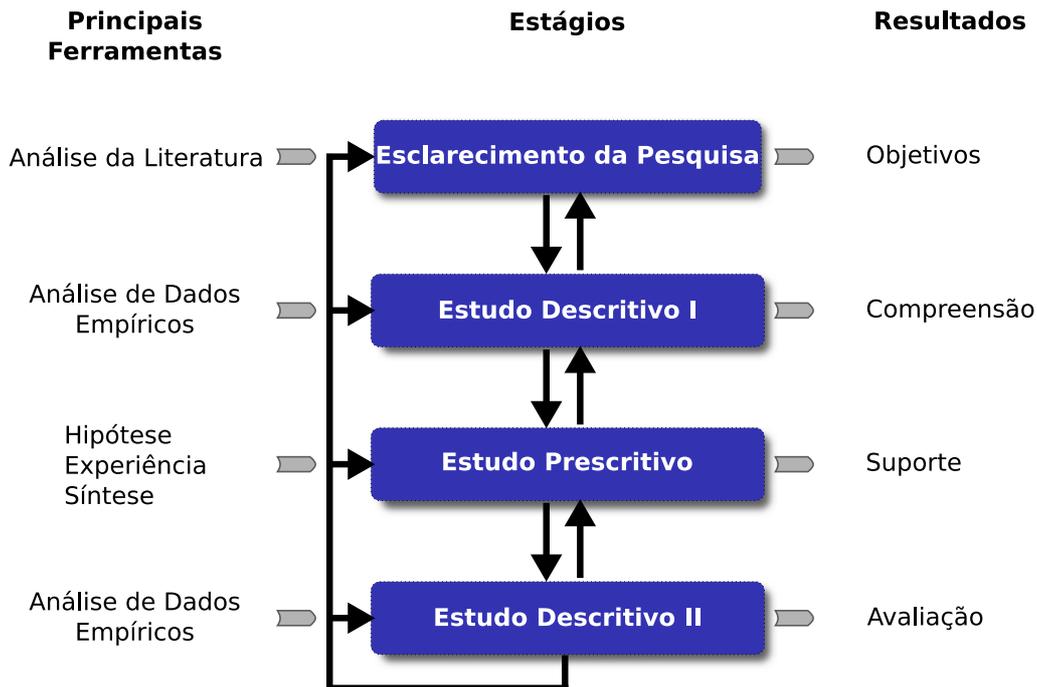


Figura 3: Metodologia de Pesquisa em Projetos (traduzido de (BLESSING; CHAKRA-BARTI, 2009))

Para isso, baseia-se na análise da literatura que apoie o esclarecimento das tarefas envolvidas e o sucesso do projeto. Com base na análise anterior, faz-se uma descrição inicial da situação existente e da situação desejada para estruturar a hipótese.

No estágio de *Estudo Descritivo I*, possuindo objetivo e foco bem definidos, é feita uma revisão da literatura com vista à busca de fatores mais determinantes para elaborar uma descrição ainda mais detalhada da situação existente, desse jeito, podem-se estabelecer de forma efetiva e eficiente quais são os fatores que devem ser abordados para melhorar o esclarecimento das tarefas. Contudo, não são achadas evidências suficientes na literatura para determinar claramente esses fatores e decide-se observar e interrogar outros pesquisadores da área para obter uma melhor compreensão da situação existente antes de avançar para o estágio seguinte e começar o desenvolvimento de suporte que aborde esses fatores.

No *Estudo Prescritivo* é usado o conhecimento acrescentado da situação existente para corrigir e elaborar a descrição inicial da situação desejada. Essa descrição representa a visão de como, abordando um ou mais fatores da situação existente pode-se atingir uma situação desejada e melhorá-la. Vários cenários são desenvolvidos através da variação do(s) fator(es) envolvido(s). Desse modo é decidido focar o trabalho no melhoramento da qualidade da definição do problema dependendo dos fatores mais promissórios diminuindo assim o número de modificações que serão necessárias o qual diminuirá eventualmente o "*time-to-market*" do projeto e incrementará o sucesso dele. É nesta etapa, que se possui a confiança necessária para começar o desenvolvimento sistemático de um suporte para melhorar a qualidade da definição do problema. Depois de um esclarecimento de tarefas e uma etapa de desenvolvimento conceitual, se possui o conceito de uma ferramenta de software (o suporte pretendido) o qual espera-se que apoie e suporte na definição do problema como é projetada. É decidido focar os esforços na base deste suporte, pois isso deverá ser suficiente para avaliar os conceitos e verificar as hipóteses subjacentes

Posteriormente é efetuado o estágio de *Estudo Descritivo II* para avaliar o impacto do suporte e a sua habilidade para atingir a situação desejada. São assumidos dois estudos empíricos para obter conhecimento do uso atual do suporte. O primeiro estudo avalia a aplicabilidade do suporte, a pergunta principal que pretende-se responder é se o software pode ser usado para apoiar uma definição de alta qualidade do problema. O segundo estudo avalia a utilidade, i.e., o sucesso do software baseado nos critérios definidos anteriormente. Estes estudos mostram que o suporte é aplicável, porém a utilidade é menor da esperada porque o suporte desenvolvido atualmente só inclui uma parte do suporte pretendido. Desse jeito são observados vários efeitos que não tinham sido antecipados, tais como a grande quantidade de tempo necessário para manter atualizada a definição do problema. Finalmente se conclui que o conceito é promissor, mas são necessários estudos adicionais da situação existente e que a descrição da situação desejada necessita ser adaptada adequadamente. Para isso é recomendado recomençar o estágio de *Estudo Descritivo I*.

Em continuidade, faz-se um detalhamento de cada um dos estágios da DRM e os seus resultados. As ferramentas metodológicas e os objetivos específicos de cada estágio podem ser consultados em (BLESSING; CHAKRABARTI, 2009).

## 2.1 Esclarecimento da Pesquisa

Como mencionado anteriormente, o estágio de *Esclarecimento da Pesquisa* foca-se na formulação dos objetivos de pesquisa realistas e promissores. Segundo (BLESSING; CHAKRABARTI, 2009), tem-se observado que muitos estudantes pesquisadores visam suas metas em critérios muito abstratos ou de longo prazo, o qual resulta em objetivos, questões de pesquisa e planos de projeto pouco realistas. Isto acontece em parte pelo fato que em muitas publicações de pesquisa são apresentados os objetivos finais da pesquisa, por exemplo, redução do tempo de execução, melhoramento da qualidade do produto, etc. Porém, estes objetivos não são relacionados com os focos atuais do projeto de pesquisa, por exemplo, melhorar a comunicação dos membros do projeto, assistir a eventos internacionais para obter melhores informações do assunto etc. Isto faz que os projetos apresentem lacunas importantes na linha de argumentação entre os fatores mencionados nos objetivos e aqueles que são estudados ou abordados.

Os produtos deste estágio podem ser agrupados em dois conjuntos:

- Conhecimentos Atuais e Expectativas: Modelo de Referência Inicial; Modelo de Impacto Inicial e Critérios Preliminares.
- Plano de Pesquisa Geral: Objetivos e focos da pesquisa; Problemas, perguntas principais e hipóteses da pesquisa; Áreas relevantes a ser consultadas; Abordagem (tipo de pesquisa, estágios principais e métodos); Contribuição esperada e resultados; Cronograma.

## 2.2 Estudo Descritivo I

O estágio de *Estudo Descritivo I* tem por finalidade ampliar o conhecimento do projeto e os seus fatores de sucesso a través da revisão da literatura sobre investigação empírica, empreendimento de investigação empírica e adicionalmente, através do raciocínio. O ponto de partida é o Modelo de Referência Inicial elaborado no estágio de *Esclarecimento da Pesquisa* e os Critérios Preliminares. Para isso, apoia-se em uma série de métodos e

diretrizes que dependem fortemente da disciplina ou área envolvida na pesquisa como: Métodos de coleta de dados, análise de ambientes de laboratório e industriais, análise estatística, etc. No entanto, esses métodos nunca irão substituir a importância da consulta de especialistas nas disciplinas relevantes para garantir que sejam escolhidos os métodos mais adequados e eles sejam aplicados corretamente.

Como resultados da execução deste estágio, podem-se mencionar:

- Um modelo de referência terminado, critérios de sucesso, critérios de sucesso mensuráveis e fatores-chave que:
  - Descrevem a situação existente e destacam os problemas;
  - Mostram a relevância do tópico de pesquisa;
  - Esclarecem e ilustrar a linha principal de argumentação;
  - Indicam os fatores mais adequados para abordar a fim de melhorar a situação;
- Um modelo de impacto inicial atualizado.
- Implicações das descobertas devido ao desenvolvimento do suporte e/ou para a avaliação do suporte existente.

### 2.3 Estudo Prescritivo

Em última instância a pesquisa de projetos trata-se de desenvolver o suporte para o melhoramento de projetos embora esse não seja o foco de um projeto individual. O desenvolvimento de suporte de projetos possui uma longa tradição, no entanto, existe pouca evidência do uso extensivo de dados empíricos válidos provavelmente pelo fato que as comunidades de pesquisa desenvolveram os seus suportes de maneira relativamente independente das demais, e porque muitos estudos empíricos não estabelecem as relações entre os fatores influentes e o sucesso do projeto.

Para o desenvolvimento do suporte é importante apresentar a visão subjacente do projeto a ser desenvolvido, i.e., a visão que possui o pesquisador sobre a situação desejada e a função do suporte. As suposições nas quais o suporte está baseado devem ser apresentadas como fatos, isto é importante pois vai influenciar o desenvolvimento do suporte e as suas probabilidades de sucesso. Também deve-se dar especial importância a concepção do projeto, pois como aponta (BLESSING; CHAKRABARTI, 2009), nesta etapa é usualmente observado que uma grande quantidade de tempo é investido nos detalhes do suporte -particularmente se ele implica o desenvolvimento de software- em vez de sua concepção, embora a contribuição principal da pesquisa muitas vezes dependa dela. Como produtos deste estágio podem ser citados:

- Documentação do suporte pretendido:
  - Descrição do suporte pretendido: O que é e como funciona.
  - Plano de início pretendido: Como iniciar, instalar, customizar, usar e manter o suporte assim como as suas condições organizacionais, técnicas e infra-estruturais.
  - Modelo de impacto pretendido.

- Suporte atual: Apostila, lista de tarefas, software, etc.
- Documentação do suporte atual:
  - Descrição do suporte atual.
  - Plano de início atual.
  - Modelo de impacto atual.
- Resultados da avaliação do suporte.
- Esboço do plano de avaliação.

## 2.4 Estudo Descritivo II

O estágio de *Estudo Descritivo II* foca-se na avaliação do suporte. Como aponta (BLESSING; CHAKRABARTI, 2009), em muitos estudos são feitos esclarecimentos sobre o uso do suporte e não sobre a sua avaliação, isto é em parte porque se espera que o suporte do projeto seja usado eventualmente na abordagem das necessidades e problemas que motivaram o seu desenvolvimento. Além disso, o suporte desenvolvido é usualmente avaliado usando produtos ou processos previamente existentes, isto é, produtos ou processos já conhecidos. A fim de observar o efeito de um suporte, torna-se necessário a sua aplicação sem o conhecimento do resultado. Geralmente, os projetos de pesquisa raramente tratam estes problemas, conseqüentemente, a maioria das avaliações dificilmente revelam os problemas reais do uso do suporte no projeto. Existem muitas maneiras nas quais o suporte do projeto pode ser avaliado, mas a criatividade é requerida para estabelecer uma avaliação apropriada que adapte os objetivos e restrições do projeto e que ao mesmo tempo, proporcione a fiabilidade necessária no suporte proposto. Isto torna o estágio de *Estudo Descritivo II* uma tarefa desafiante, mas não impossível. Os resultados obtidos depois da execução do *Estudo Descritivo II* são:

- Resultados da avaliação da aplicação.
- Resultados da avaliação do sucesso.
- Conseqüências e sugestões para o melhoramento de:
  - O suporte atual.
  - O suporte pretendido, seu conceito, elaboração e pressupostos subjacentes.
  - O plano atual e pretendido de apresentação incluindo introdução, instalação, customização, uso e problemas de manutenção.
  - O modelo de impacto atual e pretendido.
  - O modelo de referência.
  - Os critérios usados.

## 3 REFERENCIAL TEÓRICO

Este capítulo apresenta a visão geral dos tópicos importantes que embasaram o projeto desta dissertação e que permitem sua compreensão. Esses tópicos podem ser resumidos em duas áreas: Sistemas agentes e multi-agentes, e coordenação bio-inspirada.

### 3.1 Sistemas Agentes e Multi-agentes (MAS)

#### 3.1.1 Agentes de Software

Os conceito de agente de software tem sido um tópico de grande interesse no ambiente acadêmico pois ele oferece um paradigma que possui a vantagem de dividir uma tarefa complexa em várias sub-tarefas menores e mais simples. Uma vez que cada indivíduo ou agente pode ser desenvolvido para oferecer uma solução especializada a uma sub-tarefa específica, podem-se construir sistemas que exibam comportamentos complexos por meio do uso de um conjunto de agentes relativamente simples.

No entanto, como aponta (FRANKLIN; GRAESSER, 1996), no estado da arte existem várias definições do conceito de agente de software, mesmo assim, alguns conceitos comuns podem ser identificados na maioria destas definições, os quais possibilitam afirmar que um agente de software é uma abstração de software ou entidade executável que apresenta as seguintes propriedades:

- **Persistência:** O código de um agente de software é executado como uma tarefa constante uma vez que ele é iniciado até que ele decide autonomamente terminar sua própria execução.
- **Autonomia:** Os agentes possuem a capacidade de tomar decisões sem intervenção de nenhuma entidade externa incluindo o seu próprio autocontrole.
- **Habilidade Social:** Os agentes possuem a capacidade de interagir e cooperar com outros agentes através de algum tipo de linguagem de comunicação de agentes.
- **Reatividade:** Os agentes possuem a capacidade de perceber e responder às mudanças no seu ambiente adequadamente.

Uma propriedade adicional que não foi incluída na lista porque depende do tipo de estrutura de controle exibida pelos agentes, é a pro-atividade, isto é, que além de possuir a capacidade de perceber e responder às mudanças no ambiente, os agentes são capazes de apresentar um comportamento orientado a objetivos comuns.

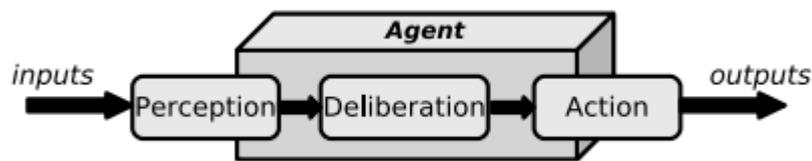


Figura 4: Agente. (UHRMACHER; WEYNS, 2009)

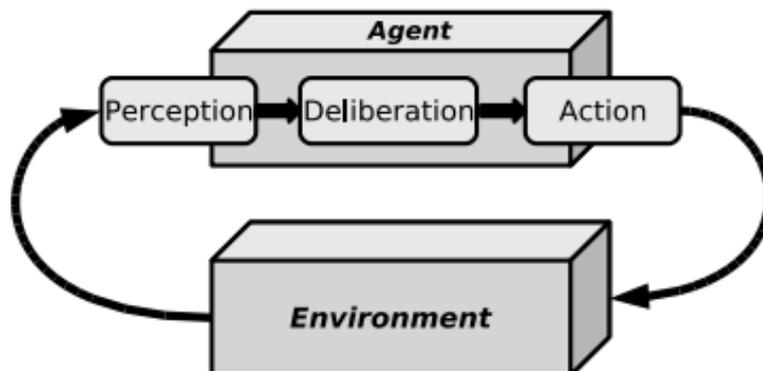


Figura 5: Agente e Ambiente. (UHRMACHER; WEYNS, 2009)

Em consequência, um agente simples pode ser modelado como um processo cíclico que possui três fases: Percepção, deliberação e ação como é apresentado por (UHRMACHER; WEYNS, 2009) na Figura 4

Inicialmente o agente recebe informações do ambiente que são processadas na fase de percepção. Posteriormente, na etapa de deliberação, é definido como o agente usa a percepção obtida para atualizar sua própria representação do mundo de interesse. Finalmente na fase de ação, os agentes fazem a tomada das decisões mais adequadas dependendo das percepções anteriormente obtidas e a sua representação atualizada do mundo de interesse, a maioria das vezes, as ações são dirigidas à modificação direta do ambiente. Desse modo, o ambiente define as condições nas quais os agentes existem e ele faz parte integral do sistema como é representado na Figura 5.

Dependendo das características e dinâmicas do ambiente, (RUSSELL; NORVIG, 1995) considera a seguinte classificação de ambientes:

- **Acessível vs. inacessível:** O ambiente é definido como acessível se o sistema de sensoramento do agente possui acesso a todos os aspectos que são relevantes para definir as possíveis ações. Um ambiente acessível é conveniente porque o agente não precisa necessariamente de um estado interno que represente o mundo de interesse.
- **Determinístico vs. não determinístico:** Se o seguinte estado do ambiente for totalmente determinado por seu estado atual e as ações selecionadas pelos agentes, o ambiente é dito determinístico.
- **Episódico vs. não episódico:** Nos ambientes episódicos o comportamento do agente é dividido em episódios (uma série de percepção e ação do agente) os quais não dependem das ações selecionadas em episódios anteriores. Os ambientes episódicos são mais simples pois neles os agentes não precisam se antecipar ao ambiente.

- Estático vs. dinâmico: Se o ambiente puder sofrer modificações durante a deliberação de um agente, o ambiente é dito dinâmico para esse agente, caso contrário, é dito estático. Nos ambientes estáticos os agentes não precisam monitorar o ambiente durante sua deliberação de ações nem precisam lidar com problemas pela passagem do tempo, o que facilita a operação. Se o ambiente não muda com o decorrer do tempo, mas o desempenho do agente é afetado, o ambiente é dito de semi-dinâmico.
- Discreto vs. contínuo: Um ambiente discreto possui uma quantidade limitada e claramente definida de percepções e ações.

### 3.1.2 Holons

Os Holons e os agentes de software são conceitos similares, porém, não são equivalentes. Tanto os holons como os agentes possuem as características anteriormente mencionadas de persistência, autonomia, habilidade social e reatividade, porém a maior diferença, como apontam (VERSTRAETE et al., 2008; GIRET; BOTTI, 2004) é que os holons são associados a uma parte física no sistema e possuem uma separação explícita entre a parte de processamento de informações e a parte do processamento físico. Essa separação não é explícita nos agentes de software. Desse modo, um holon pode ser considerado como um tipo especial de agente de software que possui necessariamente uma parte física correspondente e uma estrutura interna de duas capas.

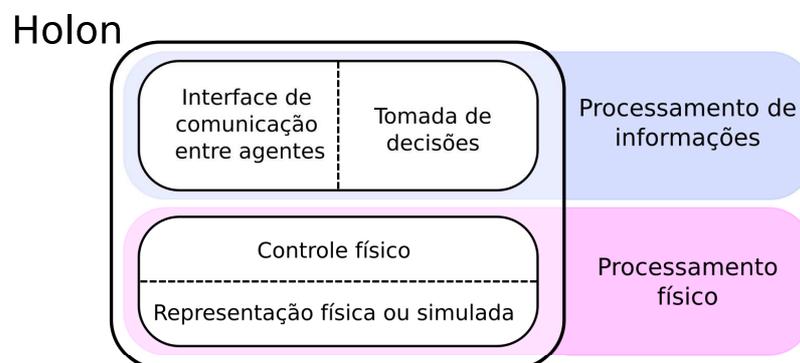


Figura 6: Arquitetura de um holon. (Fonte: Próprio autor)

A arquitetura de um holon pode ser visualizada na Figura 6. Nela, pode ser visualizada claramente a separação entre as duas partes do holon. A parte superior chamada de **processamento de informações** é responsável pela comunicação entre o holon com outros agentes que possam interagir no MAS sejam eles agentes, holons ou humanos envolvidos no MAS, além disso, é encarregada pelo agendamento de tarefas ou tomada de decisões que o holon executa. A capa inferior, correspondente ao **processamento físico**, é dividida em duas sub-capas: A primeira, chamada de controle físico, responsável pelas informações que permitem executar o controle sobre a representação física ou simulada, e a sub-capa inferior, encarregada de representar a parte física ou simulada do holon.

Como aponta (VERSTRAETE et al., 2008), os agentes em um sistema de manufatura holônico podem trocar informações diretamente a través de mensagens usando sua capa de processamento de informações, ou através do ambiente usando a capa de processamento físico.

### 3.1.3 Sistemas Multi-Agentes (MAS)

Uma das características importantes que os agentes possuem é a habilidade social. Essa habilidade permite que os agentes possam ter comportamentos de coordenação e cooperação entre eles. Assim, pode-se construir sistemas que exibam comportamentos complexos por meio do uso de um conjunto de agentes simples que interajam entre si para atingir um objetivo individual ou coletivo. Estes sistemas recebem o nome de sistemas multi agente (MAS).

Usualmente é dito que o ambiente em um MAS é inerentemente dinâmico desde o ponto de vista de cada agente, devido à presença de outros agentes que modificam o ambiente (VLASSIS, 2007). Além disso, devido ao fato que cada agente é desenvolvido para oferecer uma solução especializada à sub-tarefa no ambiente, os agentes possuem acesso unicamente às informações relevantes para definir suas próprias ações, e não à totalidade de informações que o ambiente pode gerar ou às decisões dos demais agentes do sistema.

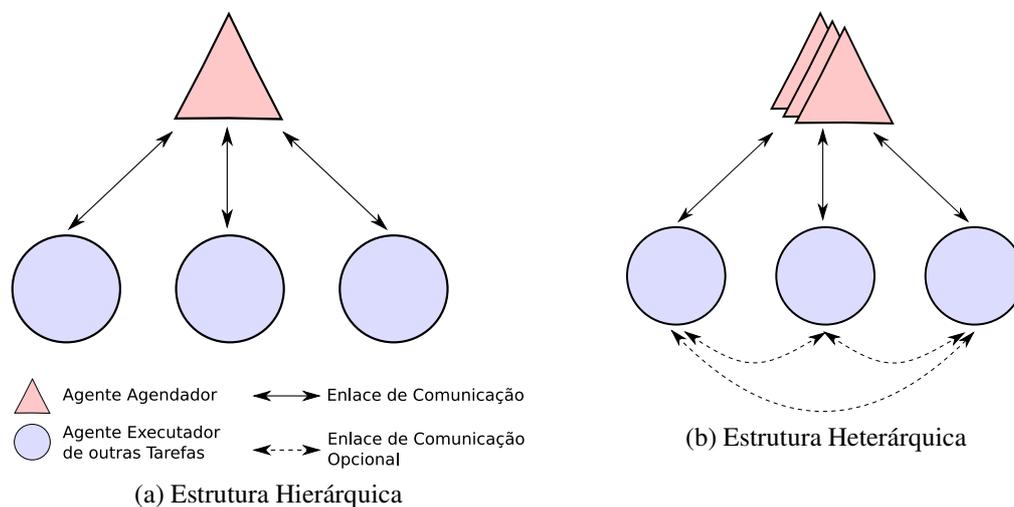


Figura 7: Comunicação entre agentes para executar o agendamento de tarefas (Fonte: Próprio autor)

Existem vários tipos de estruturas de controle clássicas em MAS: Estruturas hierárquicas, descentralizadas ou heterárquicas, e híbridas (BARBOSA et al., 2015). Nas estruturas hierárquicas existe um único agente encarregado do agendamento das tarefas no MAS, desse modo, os demais agentes do sistema devem-se comunicar com ele caso seja requerido um agendamento a outro agente do sistema. O agente agendador possui uma visão centralizada e geral de todas as informações relevantes que permitem aos agentes do MAS executar um agendamento eficiente e atingir um desempenho ótimo, porém, os MAS hierárquicos não possuem uma boa resposta a distúrbios (BARBOSA et al., 2015) e possuem um alto tráfego de dados pelo fato de possuir um agente centralizado que deve receber e enviar informações referentes a todos os agendamentos no MAS (VALCKENAERS et al., 2004). Consequentemente, as estruturas heterárquicas ou descentralizadas foram desenvolvidas para lidar com esses problemas. Os MAS heterárquicos possuem vários agentes de agendamento de tarefas que não têm uma visão global do sistema, isto faz que, dependendo do tipo de dados solicitados pelo mecanismo de agendamento de tarefas, os demais agentes do MAS devam em alguns casos, comunicar entre si informações do sistema para executar o agendamento. Finalmente, os MAS que possuem estruturas híbridas usam um agente centralizado ou vários dependendo das condições no ambiente,

desse modo exibem os dois comportamentos. Uma visão de como os agentes nestes tipos de estruturas interagem para executar o agendamento de tarefas é descrito na Figura 7.

### 3.1.4 Tecnologias de Suporte ao Desenvolvimento de Sistemas Agentes e Multi-Agentes (MAS)

Vários tipos de plataformas que permitem projetar e desenvolver MAS têm sido propostas no estado da arte. Como aponta (NIKOLAI; MADEY, 2009), essas plataformas possuem diversos tipos de disponibilidade, ferramentas de trabalho, protocolos de comunicação entre agentes, e abordagens acadêmicas ou comerciais. Na proposta apresentada nesta dissertação foi escolhida a plataforma JADE a qual cumpre as especificações definidas em FIPA.

#### 3.1.4.1 FIPA

FIPA (*Foundation for Intelligent Physical Agents*) foi estabelecida no ano 1996 como uma associação internacional sem fins lucrativos para o desenvolvimento de um conjunto de normas relacionadas à tecnologia de agentes de software. Naquele momento, os agentes de software já eram bem conhecidos na comunidade acadêmica, porém, as empresas comerciais tinham pouco interesse nessa tecnologia e só abordavam ela desde uma perspectiva exploratória (BELLIFEMINE; CAIRE; GREENWOOD, 2007).

Conseqüentemente, o consórcio decidiu elaborar normas que definissem o fundamento de uma nova indústria fazendo ela implementável através de um grande conjunto de aplicações.

Como fundamentos de FIPA estão os seguintes princípios:

- As tecnologias de agentes proporcionam um novo paradigma que solucionam problemas novos e antigos.
- Algumas tecnologias de agente têm atingido um grau considerável de maturidade.
- Para ser implementadas, algumas tecnologias de agentes precisam de ser padronizadas.
- A padronização de tecnologias genéricas é possível e têm se demonstrado que proporciona resultados efetivos a través de outras formas de padronização.
- A padronização da mecânica interna dos próprios agentes não é o assunto principal, mas sim a infraestrutura e linguagem requerida para uma execução aberta.

Decidiu-se então padronizar a linguagem dos agentes. FIPA-ACL *Agent Communication Language* (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (FIPA), 2002) define a linguagem padronizada que os agentes usam para intercambiar informações. FIPA-ACL define vários protocolos através dos quais a comunicação pode ser efetuada, alguns desses protocolos são: Protocolo *FIPA-Request*, protocolo *FIPA-Cancel-Meta* e protocolo *FIPA-Contract Net*. Além da comunicação, o segundo aspecto fundamental dos MAS abordados nas especificações FIPA é o gerenciamento de agentes. Portanto, foi desenvolvido um framework com as normativas que permitem aos agentes FIPA sua criação, registro, localização, comunicação, migração e execução dos agentes. O modelo de referência para o gerenciamento de agentes consiste dos componentes exibidos na figura 8. Nela são descritos os agentes principais definidos em FIPA para a execução do MAS, esses agentes são:

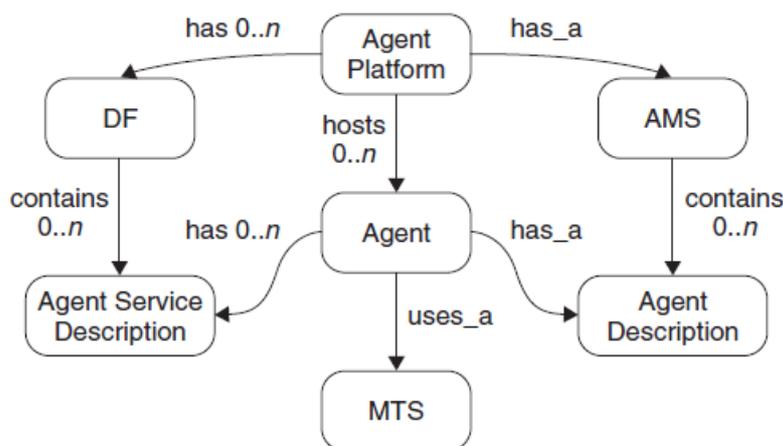


Figura 8: Representação da ontologia de gerenciamento de agentes. (BELLIFEMINE; CAIRE; GREENWOOD, 2007)

- **Agente:** É uma instância da classe *Agent* e representa a base que permite a definição dos agentes e oferece todas as interações básicas e métodos que facilitam a implementação deles.
- **Agent Management System (AMS):** É o encarregado do gerenciamento de agentes na plataforma. Todos os agentes devem ser registrados no AMS e devem ser identificados a través de um identificador AID. Além disso o AMS exerce controle sobre o acesso e uso da plataforma.
- **Directory Facilitator (DF):** É o agente encarregado de oferecer o serviço de páginas amarelas da plataforma, i.e., onde os demais agentes registram suas habilidades e serviços para que estejam a disposição dos demais agentes da plataforma.

#### 3.1.4.2 Plataforma JADE

JADE (*Java Agent DEvelopment Framework*) é um *framework* de software totalmente implementado em linguagem Java que foi inicialmente desenvolvido por *Telecom Italia* no final do ano 1998 para validar as primeiras especificações FIPA (*Foundation for Intelligent Physical Agents*). Os sistemas baseados em JADE podem ser distribuídos através de várias máquinas (sem a necessidade de possuir o mesmo sistema operacional) e a configuração pode ser controlada usando uma interface gráfica remota ou durante sua execução através de agentes móveis de uma máquina à outra quando for necessário. JADE simplifica a implementação de MAS através de um conjunto de ferramentas gráficas que permitem o monitoramento de mensagens, depuração, criação e eliminação de agentes no MAS e naturalmente, através da execução das especificações FIPA (BELLIFEMINE; CAIRE; GREENWOOD, 2007).

A interface gráfica desenvolvida em JADE que permite o gerenciamento da plataforma está controlada por um agente chamado de *Remote Monitoring Agent* ou RMA. Esse agente permite visualizar o estado da plataforma, o estado dos agentes que estão interagindo nela, e oferece as ferramentas que permitem solicitar serviços administrativos do agente AMS, e as tarefas de depuração e teste anteriormente mencionadas. Vários agentes RMA podem ser executados na mesma plataforma desde que cada instância possua um AID diferente.

### 3.1.4.2.1 Instalação e execução da Plataforma

Todo o software relacionado a JADE pode ser descarregado do site de JADE em <http://jade.tilab.com>. O software está dividido em duas seções a distribuição principal e os complementos. Os complementos incluem módulos independentes que implementam características ampliadas como interpretadores para determinadas linguagens. Adicionalmente, a distribuição principal é composta de cinco arquivos com o seguinte conteúdo:

- **jadeBin.zip** contém os arquivos .jar pre-compilados prontos para ser usados.
- **jadeDoc.zip** possui a documentação, incluindo os manuais de administrador e programador. Documentos que também estão disponíveis diretamente no site.
- **jadeExamples.zip** contém o código fonte de vários exemplos.
- **jadeSrc.zip** contém todas as fontes de JADE.
- **jadeAll.zip** possui os quatro arquivos anteriormente indicados.

Depois de descompactar os arquivos anteriormente mencionadas é achada a pasta *jade/lib*, ela deve ser incluída no CLASSPATH de Java para iniciar a plataforma. Consequentemente, antes de iniciar a plataforma o usuário deve definir seu CLASSPATH local de Java, i.e. o conjunto de diretórios onde a Máquina Virtual Java procurará o bytecode ou seja, os arquivos .class e .jar.

Depois de definido o CLASSPATH a interface gráfica de JADE pode ser iniciada usando o comando:

```
prompt> java jade.Boot -gui
```

O resultado é mostrado na figura 9 e a opção de comando *-gui* adiciona o efeito de iniciar a interface gráfica JADE apresentada na figura 10.

```

dic 07, 2017 12:25:45 PM jade.core.Runtime beginContainer
INFORMACIÓN: -----
                This is JADE 4.4.0 - revision 6778 of 21-12-2015 12:24:43
                downloaded in Open Source, under LGPL restrictions,
                at http://jade.tilab.com/
                -----
dic 07, 2017 12:25:46 PM jade.imtp.leap.LEAPIMTPManager initialize
INFORMACIÓN: Listening for intra-platform commands on address:
- jicp://192.168.0.10:39965

dic 07, 2017 12:25:46 PM jade.core.BaseService init
INFORMACIÓN: Service jade.core.management.AgentManagement initialized
dic 07, 2017 12:25:46 PM jade.core.BaseService init
INFORMACIÓN: Service jade.core.messaging.Messaging initialized
dic 07, 2017 12:25:46 PM jade.core.BaseService init
INFORMACIÓN: Service jade.core.resource.ResourceManagement initialized
dic 07, 2017 12:25:46 PM jade.core.BaseService init
INFORMACIÓN: Service jade.core.mobility.AgentMobility initialized
dic 07, 2017 12:25:46 PM jade.core.BaseService init
INFORMACIÓN: Service jade.core.event.Notification initialized
dic 07, 2017 12:25:46 PM jade.mtp.http.HTTPServer <init>
INFORMACIÓN: HTTP-MTP Using XML parser com.sun.org.apache.xerces.inter
nal.jaxp.SAXParserImpl$JAXPSAXParser
dic 07, 2017 12:25:46 PM jade.core.messaging.MessagingService boot
INFORMACIÓN: MTP addresses:
http://DESKTOP-3NLG9IC:7778/acc
dic 07, 2017 12:25:46 PM jade.core.AgentContainerImpl joinPlatform
INFORMACIÓN: -----
Agent container Main-Container@192.168.0.10 is ready.
                -----

```

} Aviso inicial JADE  
 } Inicialização de Serviços  
 } Endereços MTP  
 } Nome do contêiner dos agentes

Figura 9: Inicialização da plataforma JADE. (Fonte: Próprio autor)

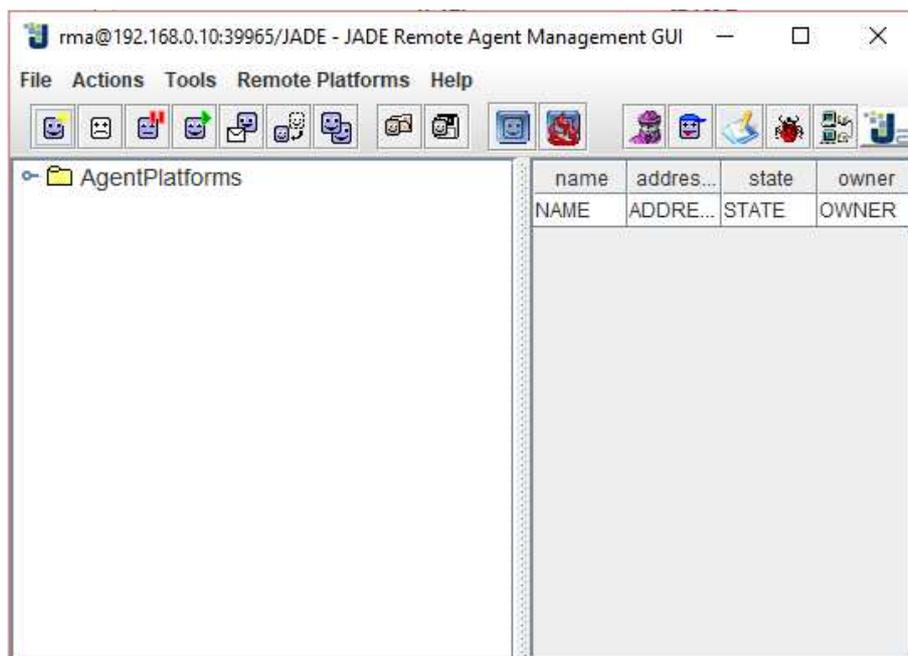


Figura 10: Interface gráfica do RMA de JADE. (Fonte: Próprio autor)

## 3.2 Coordenação Bio-Inspirada

### 3.2.1 Estigmergia

O conceito de estigmergia (do grego stigma: sinal e ergon: trabalho) foi estabelecido pelo biólogo Pierre-Paul Grassé em 1959. Ele define o mecanismo de coordenação heterárquica que possuem as interações entre animais, especificamente, o modo no qual a comunicação indireta ocorre entre indivíduos que pertencem a uma sociedade de insetos através de mudanças induzidas no ambiente. Com este mecanismo, as formigas são capazes de achar o menor caminho entre o seu formigueiro e as fontes de comida próximas. (THERAULAZ; BONABEAU, 1999)

O comportamento das formigas forrageiras pode ser resumido em dois passos:

1. Seleção da trilha: Ocorre quando as formigas forrageiras saem do formigueiro e começam a busca de fontes de alimento no seu ambiente. Para executar a busca de comida, cheiram uma substância química no ambiente chamada feromônio. Existem diferentes trilhas com concentrações distintas de de feromônio, quanto maior for concentração na trilha, mais atrativa será ela para as formigas. Caso não existam feromônios inseridos no ambiente, cada formiga realiza uma busca aleatória por comida.
2. Disseminação de informações: Quando uma formiga achar comida, ela carrega uma parte dela, e deposita feromônios no caminho de retorno ao seu formigueiro modificando dessa maneira o ambiente para as demais formigas e criando uma trilha de feromônios. Uma característica importante das trilhas de feromônios é o fato de sofrer evaporação a uma taxa estabelecida, assim, quando uma fonte de comida estiver esgotada, as formigas não terão a capacidade de carregar mais comida e a trilha de feromônios que conduz à fonte de comida esgotada evaporará.

Os rastros depositados pelas formigas criam uma memória coletiva compartilhada por todos os indivíduos da colônia fazendo que informações globais possam ser disponibilizadas localmente e refletindo dados relevantes que são úteis para o mecanismo de tomada de decisões. Caso duas ou mais trilhas de feromônios sejam achadas à mesma fonte de comida, a menor vai ser a mais atrativa devido ao fato que estará menos evaporada pois as formigas precisarão de um tempo menor para percorrer essa trilha, desse modo, o tráfego na trilha menor é mais atrativo e o rastro de feromônios nela é reforçado por novas formigas. O comportamento descrito anteriormente pode ser visualizado na Figura 11 onde uma ponte com dois caminhos foi colocada entre um formigueiro e uma área de comida. À esquerda, a Figura 11 mostra o comportamento da colônia no minuto 4 e à direita o comportamento aos 8 minutos do iniciado o experimento. A figura mostra como no início do experimento as formigas executam uma busca aleatória para chegar à fonte de comida, e aos 8 minutos, todas as formigas da colônia seguem o caminho menor.

Consequentemente, a estigmergia apresenta um estado transitório inicial, quando as formigas realizam uma busca aleatória da comida, e um estado estacionário final quando as formigas têm achado um caminho ótimo e toda a colônia segue esse mesmo caminho.

O algoritmo anteriormente descrito é simples, e a sua robustez e flexibilidade baseia-se na volatilidade das informações da estigmergia (evaporação de feromônios) e o seguimento estocástico de rotas existentes, também no seu comportamento heterárquico baseado na cooperação de formigas individuais.

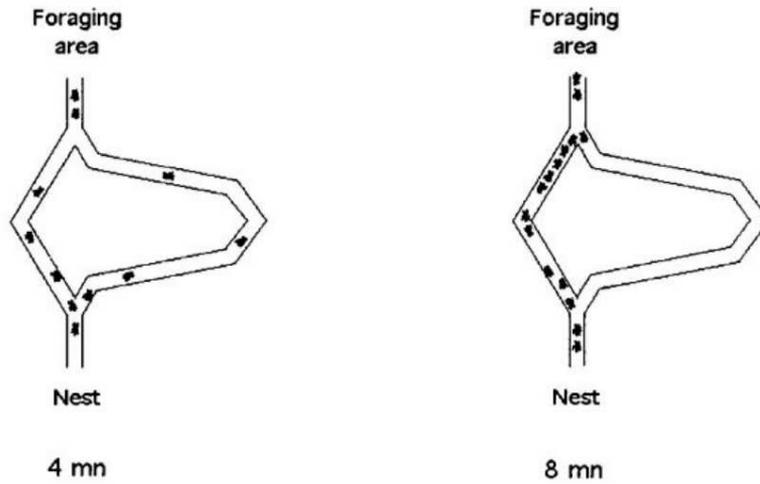


Figura 11: Configuração Experimental e Seleção de caminhos menores por uma colônia de *Linepithema humile* (DORIGO; BONABEAU; THERAULAZ, 2000).

### 3.2.2 Algoritmo de Colônia de Formigas (ACO)

O algoritmo de otimização por colônia de formigas (ACO) foi concebido por Marco Dorigo em sua tese de PhD (DORIGO, 1992). Ele define uma meta-heurística que usa o conceito de estigmergia para resolver problemas computacionais que envolvem procura de bons caminhos em grafos. O problema inicial concebido em (DORIGO, 1992) foi o problema do caixeiro-viajante, porém diversas aplicações têm emergido da ideia inicial. Já em (COLORNI et al., 1994) foi usado por primeira vez para resolver o problema de agendamento de tarefas em manufatura com o objetivo de diminuir o tempo de manufatura.

Em (COLORNI et al., 1994) são implementadas formigas virtuais encarregadas de solucionar o problema de agendamento de tarefas em manufatura, e os nós do grafo são associados ao processamento de um trabalho por recurso. Além disso, define o seguinte método para executar a seleção de rotas e disseminação de informações:

1. Seleção de rotas: Cada produto se move do nó  $i$  ao nó  $j$  com a probabilidade descrita em 1. Nela,  $\tau_{ij}$  representa a concentração de feromônios entre o nó  $i$  e  $j$ ;  $\eta_{ij}$  é uma medida heurística da desejabilidade de escolher a rota entre  $i$  e  $j$ ;  $\mathbf{A}$  representa a série de nós adjacentes a  $i$ ;  $\alpha$  e  $\beta$  são parâmetros que permitem ao usuário equilibrar a importância oferecida à heurística gulosa (parâmetro  $\beta$ ) em relação à concentração de feromônios entre nós (parâmetro  $\alpha$ )

$$Prob(i,j) = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in \mathbf{A}} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} & \text{se } j \in \mathbf{A} \\ 0 & \text{Caso contrário} \end{cases} \quad (1)$$

2. Disseminação de informações: Depois que as formigas virtuais achem uma solução possível, disseminam informações através da atualização de feromônios no ambiente, essa atualização é executada segundo 2.

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad (2)$$

Onde  $\rho$  é a taxa de evaporação de feromônios e  $\Delta\tau_{ij}$  é a quantidade de feromônios inseridos por uma formiga na rota entre  $i$  e  $j$ . Essa quantidade possui dois possíveis valores os quais são  $Q/L$  caso a rota entre  $i$  e  $j$  fosse escolhida pela formiga e 0 caso contrário como apresentado em 3. Onde os parâmetros  $Q$  e  $L$  representam uma constante do sistema e o valor da função objetivo achado pela formiga, respectivamente. Deste modo, quanto maior for o valor  $L$  atingido pela formiga na função objetivo, menor vai ser a quantidade de feromônios inseridos na trilha.

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{L} & \text{se } ij \text{ foi escolhido} \\ 0 & \text{Caso contrário} \end{cases} \quad (3)$$

Conseqüentemente, antes da execução do algoritmo ACO é importante definir a função objetivo. Do mesmo modo, devem ser estabelecidos os demais parâmetros relacionados às equações de seleção de rotas e disseminação de informações, a saber, a concentração inicial de feromônios  $\tau_{ij}$ , o parâmetro  $\alpha$ , as heurísticas gulosas  $\eta_{ij}$ , o parâmetro  $\beta$ , a taxa de evaporação  $\rho$  e o parâmetro de constante  $Q$ .

## 4 TRABALHOS RELACIONADOS

### 4.1 Estruturas Multiagente

Esta secção apresenta uma lista dos trabalhos mais relevantes que aplicam o paradigma de MAS em manufatura e seus agentes concebidos são identificados a fim de descrever resumidamente as funções que cada um possui no MAS.

#### 4.1.1 PROSA

PROSA *Product-Resource-Order-Staff Architecture* (VAN BRUSSEL et al., 1998) é uma arquitetura de referência para controle de manufatura desenvolvida na divisão de engenharia de produção, projeto de máquinas e automação da Universidade Católica de Louvain na Bélgica. Provê um *framework* de controle multi-agente onde os paradigmas bio-inspirados podem ser aplicados para controlar sistemas de manufatura complexos através de arquiteturas hierárquicas, heterárquicas ou híbridas. Esta arquitetura identifica os holons necessários para implementar qualquer sistema de manufatura, as suas responsabilidades e a estrutura das interações na qual eles cooperam.

A arquitetura consiste de três tipos de holons básicos: Holons de petição (*Order*), holons de produto (*Product*), e holons de recurso (*Resource*). Também podem ser adicionados holons opcionais centralizados de *Staff* para auxiliar aos agentes básicos com dados especializados. Estes agentes fornecem um alto nível de modularidade, a qual reduz a complexidade para incluir novos componentes e permite uma fácil reconfiguração do sistema. Cada agente é responsável por um aspecto do controle de manufatura:

1. Holon de Produto: Possui a descrição de um tipo de produto e fornece aos holons de petição o conhecimento necessário sobre rotas disponíveis e as fórmulas ou tarefas requeridas para criar produto adequado.
2. Holon de Recurso: Responsáveis pela operação de recursos e sistemas de transporte. Possui os dados sobre as suas funções, as tarefas em execução, sub recursos, buffers, e um histórico das suas atividades.
3. Holon de Petição: Os holons de petição ou pedido, fazem uma busca de soluções e selecionam a mais atrativa entre as possíveis candidatas para se tornar a sua intenção. Além disso fazem um rastreamento do estado do produto físico, o progresso das tarefas executadas, e os dados históricos das tarefas. É o holon responsável pela logística interna do sistema.
4. Holon de *Staff*: É um holon opcional que auxilia aos holons básicos com dados especializados, permitindo a presença de elementos e funcionalidades centralizadas na arquitetura.

Os holons descritos possuem uma exposição limitada às propriedades do sistema que não pertencem a seu domínio de especialização e são estruturadas usando conceitos de programação orientada a objetos como agregação e especialização.

#### 4.1.2 ADACOR

O trabalho ADACOR (LEITÃO; RESTIVO, 2006) e a sua evolução, ADACOR<sup>2</sup> (BARBOSA et al., 2015), apresentam agentes similares aos propostos em PROSA e fazem ênfase em uma estrutura híbrida e a capacidade que possui o seu agente centralizado de monitorar o processo de manufatura e inserir produtos na planta, deste modo, o agente centralizado concebido em ADACOR e ADACOR<sup>2</sup> não é opcional. Os agentes que compõem esta proposta são quatro, a saber, *product holon*, *task holon*, *operational holon* e *supervisor holon*. Cada um desses agentes está encarregado dos seguintes aspectos no controle do processo de manufatura:

1. *Product holon*: Representa os tipos de produto que podem ser manufaturados na planta e o conhecimento necessário para produzi-los.
2. *Task holon*: Responsáveis da logística na execução das petições de produção no chão de fábrica.
3. *Operational holon*: Representam aos recursos do sistema, e.g. robôs, estações de trabalho, humanos, etc.
4. *Supervisor holon*: Agente centralizado responsável pela otimização e monitoramento do sistema, além de inserir novas petições de manufatura nele.

#### 4.1.3 PABADIS

O trabalho PABADIS *Plant Automation Based on Distribution Systems* (KLOSTERMEYER; KLEMM, 2003) concebe três tipos diferentes de agente: *Product agent*, *residential agent* e *plant management agent*. Esses agentes são descritos nos itens a seguir:

1. *Product agent*: É um tipo de agente móvel que representa uma petição de manufatura e é o encarregado do controle e execução da petição de manufatura.
2. *Residential agent*: Proporciona a comunicação entre a rede da planta e recursos particulares nela.
3. *Plant management agent*: É uma interface entre os demais agentes do MAS. Basicamente implementa o sistema SCADA da planta o qual controla, monitora e adquire dados do sistema durante sua execução.

#### 4.1.4 HCBA

O trabalho desenvolvido em HCBA *Holonic Component-Based Approach to Reconfigurable Manufacturing Control Architecture* (CHIRN; MCFARLANE, 2000), descreve um MAS aplicado em manufatura que possui quatro tipos diferentes de agente, a saber, *product holon*, *work-in-progress agent*, *resource holon* e *coordinator*. Os componentes desta arquitetura têm as seguintes características:

1. *Product holon*: O holon de produto possui uma parte dinâmica e outra estática. A estática possui a informação necessária que permite definir a lista de tarefas requeridas pelo produto, a parte dinâmica é chamada de *work-in-progress agent*.

2. *Work-in-progress agent*: A missão de este agente é acompanhar o processo de um produto que está sendo manufaturado para executar as suas tarefas de manufatura.
3. *Resource holon*: É o componente encarregado executar tarefas como fabricação, montagem, transporte, teste, etc.
4. *Coordinator*: A função dele é criar e rastrear o estado dos agentes *work-in-progress* de modo que eles possam ser monitorados.

#### 4.1.5 KASA

Para implementar MAS em manufatura, (MERDAN et al., 2006) propõe uma arquitetura chamada *Knowledge-based Multi-agent Architecture*, a qual executa três tipos de agentes diferentes: *Order agent*, *transport agent* e *machine agent*. Cada uma de elas possui as seguintes características:

1. *Order agent*: Representa a petição do cliente (não existe um agente por cada petição), é responsável pela aquisição de dados de petição dos clientes, negociação de preços, data prevista de entrega, e possível anulação ou modificação do pedido.
2. *Transport agent*: Encarregado do controle do processo de manufatura em uma instância de produto e de transportar a instância através da topologia da planta.
3. *Machine agent*: Representa os recursos de manufatura ou estações de trabalho que oferecem processos, serviços ou tarefas particulares.

#### 4.1.6 Rockwell

É uma proposta de MAS em manufatura desenvolvida pela empresa Rockwell Automation. Descrita em (VRBA et al., 2011), possui quatro tipos diferentes de agentes, a saber, *work station/ equipment agent*, *production plan agent*, *product agent* e *order agent*. A descrição de cada agente concebido é apresentada nos itens a seguir:

1. *Work station/ equipment agent*: É o encarregado de um grupo de estações de trabalho e ferramentas que oferecem tarefas de manufatura e/ou recursos materiais.
2. *Production plan agent*: Possui o plano de produção que descreve a série de operações ou tarefas que devem ser executadas na ordem correta para obter o produto final.
3. *Product agent*: Incorpora o conceito de um produto "ativo" ou "inteligente" que administra suas próprias informações e executa decisões autônomas que permitem a execução de um plano de produção.
4. O *Order agent*: É o responsável da recepção de petições dos níveis de controle superiores do sistema. Se encarrega de dividir a petição nos produtos particulares e designar esse trabalho aos *product agents*.

#### 4.1.7 Peixoto

(PEIXOTO, 2012) implementa três tipos diferentes de agente encarregados de simular o processo de manufatura: Agente produto/peça, agente montagem/recurso e agente monitor. Cada um deles está encarregado dos seguintes aspectos no controle do processo de manufatura:

- Agente produto/peça: Encarregado da logística e representação de cada petição de manufatura realizada.
- Agente montagem/recurso: Representa cada um dos recursos de manufatura ou estações de trabalho disponíveis no chão de fábrica.
- Agente monitor: Responsável pelo monitoramento do sistema e inserção de novas petições de manufatura nele.

#### 4.1.8 Resumo

Desse modo, os conceitos principais abordados em todos os trabalhos anteriormente citados, podem ser descritos a través dos itens a seguir:

1. Petição (*Order*): Descreve a petição feita por por níveis de controle superiores do sistema (usuários) e encarregado da comunicação com eles.
2. Tipo de produto (*Product type*): Encarregado do catálogo dos possíveis produtos que podem ser manufaturados na planta.
3. Instância de produto (*Product instance*): Encarregado de representar cada um dos produtos que estão sendo manufaturados no chão de fábrica.
4. Recurso (*Resource/Machine*): Representa cada uma das estações de trabalho que podem executar tarefas de manufatura
5. Agente Supervisor (*Supervisor*): Encarregado do monitoramento centralizado da planta de manufatura.

Na tabela 1 são relacionados cada um dos conceitos anteriormente expostos e os tipos de agente estudados neste capítulo.

<b>Arquitetura MAS</b>	<b>Referência</b>	<b>Petição</b>	<b>Tipo de Produto</b>	<b>Instância de Produto</b>	<b>Recurso/Máquina</b>	<b>Supervisório</b>
PROSA	(VAN BRUSSEL et al., 1998)		product holon	order holon	resource holon	staff holon
ADACOR <sup>2</sup>	(BARBOSA et al., 2015)		product holon	task holon	operational holon	supervisor holon
PABADIS	(KLOSTERMEYER; KLEMM, 2003)			product agent	residential agent	plant management agent
HCBA	(CHIRN; MC-FARLANE, 2000)		product holon	work-in-progress agent	resource holon	coordinator
KASA	(MERDAN et al., 2006)	order agent		transport agent	machine agent	
Rockwell	(VRBA et al., 2011)	order agent	production plan agent	product agent	work station/equipment agent	
Peixoto	(PEIXOTO, 2012)			Product Agent	Resource Agent	Monitor Agent

Tabela 1: Arquiteturas MAS em Manufatura e tipos de agente implementados

## 4.2 Otimizações implementadas em MAS

Alguns dos mecanismos de agendamento que implementam otimizações em sistemas multi-agentes (MAS) são apresentados na Tabela 2, sendo (ZHANG; WONG, 2017) o trabalho mais relacionado à proposta neste aspecto. Isso devido a o seu uso direto do método de Otimização por Colônia de Formigas (ACO) e a sua implementação de um sistema multi-agente (MAS) próprio.

Os trabalhos apresentados na Tabela 2 podem ser divididos em três grupos:

- Os que implementam otimizações heurísticas.
- Os que podem suportar vários tipos de otimizações, e conseqüentemente, não especificam um tipo único de otimização.
- Os que implementam otimizações meta-heurísticas.

Os trabalhos que implementam algum tipo de otimização heurística possuem a desvantagem de não garantir a obtenção de uma solução ótima ao problema de otimização abordado. O trabalho (BARBOSA et al., 2015) implementa um MAS que pode suportar vários tipos de otimizações centralizadas no seu agente supervisorio. Devido ao fato que as otimizações são executadas neste agente centralizado, os agentes ou holones produto devem executar a sua tomada de decisões com informações que não são locais, apresentando desse modo a desvantagem de possuir problemas no tráfego de dados. Os trabalhos que implementam meta-heurísticas, como (COLORNI et al., 1994; WU; WU; WANG, 2017) não integram o conceito de MAS possuindo desse modo, problemas de flexibilidade e robustez. Finalmente os trabalhos de (AYDIN; FOGARTY, 2004; LIM; ZHANG, 2012; ZHANG; WONG, 2017) abordam os problemas de flexibilidade e otimização através da integração de algum tipo de otimização meta-heurística e o paradigma de sistemas multi-agentes (MAS). Porém, nenhum dos trabalhos anteriormente citados aproveita o uso da estigmergia para abordar o problema de tráfego de dados no MAS. Esta vantagem pode ser explorada usando o mecanismo de estigmergia, evitando a inundação convencional de informações na rede e fazendo uma tomada de decisões com informações locais em cada um dos nós do grafo, como é descrito em (FREITAS, 2011).

<b>Referência</b>	<b>Arquitetura MAS</b>	<b>Tipo de Otimização</b>	<b>Otimização</b>
(PEIXOTO, 2012)	Própria	Heurística	Gulosa
(VALCKENAERS et al., 2004)	PROSA	Heurística	Bio-inspirada
(FREITAS, 2011)	Própria	Heurística	Bio-inspirada
(BARBOSA et al., 2015)	ADACOR <sup>2</sup>	Sem especificar	Sem especificar
(COLORNI et al., 1994)	Não aplica	Meta-heurística	Colônia de Formigas (ACO)
(AYDIN; FOGARTY, 2004)	Própria	Meta-heurística	Recozido simulado modular (MSA)
(LIM; ZHANG, 2012)	Própria	Meta-heurística	Algoritmo Genético
(WU; WU; WANG, 2017)	Não aplica	Meta-heurística	Colônia de Formigas híbrido
(ZHANG; WONG, 2017)	Própria	Meta-heurística	Colônia de Formigas (ACO)

Tabela 2: Otimizações em manufatura

## 5 PROPOSTA DE USO DE MAS E ACO PARA SISTEMAS DE MANUFATURA

### 5.1 Visão Geral

Neste capítulo a proposta de uso de MAS e ACO é apresentada. Inicialmente são descritas a arquitetura e interação entre os agentes propostos (MAS), e finalmente o mecanismo de tomada de decisões baseado em ACO concebido para solucionar o problema de agendamento de tarefas é descrito.

### 5.2 Arquitetura de Agentes

Os agentes e holons implementados nesta proposta foram três: Holon Produto, Agente Recurso e Agente *Staff*. Esses agentes baseiam-se nos agentes concebidos em (PEIXOTO, 2012), isto devido ao fato de ser um trabalho do mesmo grupo de pesquisa e ter a possibilidade de integrar ambos projetos. Porém, algumas modificações foram contempladas nele a fim de integrar os conceitos de estigmergia aplicados em (FREITAS, 2011) e as noções de MAS desenvolvidas em (VAN BRUSSEL et al., 1998; VALCKENAERS et al., 2004).

As adaptações feitas à arquitetura desenvolvida em (PEIXOTO, 2012) foram:

- **Adaptação 1:** O *agente monitor* é trocado por um *agente staff*, o qual, além de ser um agente supervisor e servir como plataforma para inserir novos produtos na planta, atua como agente centralizado de transporte de todas as peças inseridas no chão de fábrica. Isto devido a que a topologia implementada em (PEIXOTO, 2012) usa uma única esteira centralizada de transporte de materiais e o agente monitor recebe a informação da posição de todos os produtos e recursos no chão de fábrica e representa eles numa tela, o qual facilita a implementação do processo de transporte neste agente. É importante salientar que a implementação deste agente não afeta a estrutura descentralizada do sistema devido a que o *agente staff* não influencia no processo de tomada de decisões no sistema.
- **Adaptação 2:** Do mesmo modo que no trabalho de Peixoto, os *agentes recurso* estão relacionados às estações de manufatura da planta. Porém, os *agentes recurso* propostos só podem executar uma tarefa de manufatura por vez e possuem dois tipos de buffer: buffer de transporte e buffer de processamento. O buffer de transporte (TB) consiste de produtos que foram alocados no recurso e cujos produtos físicos estão sendo transportados na esteira central, e o buffer de manufatura (MB)

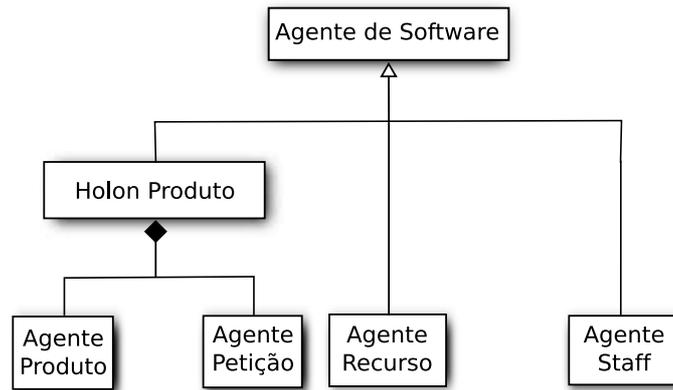


Figura 12: Agentificação Proposta (Fonte: Próprio autor).

consiste de dos produtos físicos que chegaram ao recurso e ao mesmo tempo estão à espera de ser processados.

- **Adaptação 3:** O agente produto foi trocado por um **Holon Produto**. Como descrito na secção 3.1.2, os holons possuem uma arquitetura interna de duas capas. Dese modo, cada holon produto é composto de um agente produto (*product agent*) e um agente petição (*order agent*) encarregados da capa de processamento físico e da capa de processamento de informações respetivamente.
  - O agente produto é a entidade ambiental do holon produto, atua como um servidor encarregado do processamento físico dele e possui a informação do estado de uma instância de produto, i.e., o modelo do estado do produto manufaturado, além disso, possui a lista de tarefas requeridas, a área na qual o processo de manufatura deve ser efetuado e um cronômetro que permite registrar o tempo de manufatura decorrido.
  - O agente petição proposto é o encarregado do processamento de informações do holon produto, i.e., da disseminação de informações, a busca e alocação/agendamento de tarefas, portanto, é o agente responsável pela logística interna do seu produto correspondente e não possui informações detalhadas do processo de manufatura.

A agentificação proposta é apresentada na Figura 12.

O holon produto descrito anteriormente, possui uma estrutura inspirada em (FREITAS, 2011) e é apresentada na Figura 13. O primeiro campo da estrutura é o identificador do holon produto, um número que identifica exclusivamente a cada holon produto. O campo seguinte possui informação sobre a Área de Missão (MA), que é definida pelo conjunto de recursos que podem ser escolhidos para garantir a conclusão do processo de manufatura, isto é os que garantem que dependendo do recurso que é escolhido para executar a tarefa de manufatura  $T_i$ , exista na sua vizinhança um subconjunto de recursos com a habilidade de executar a tarefa de manufatura  $T_{i+1}$  e os meios para o transporte de materiais para esse subconjunto. O terceiro campo define quais tarefas de manufatura ou habilidades são requeridas para realizar um determinado produto. O quarto campo apresenta o critério de avaliação para o mecanismo de tomada de decisões, ele descreve a qualidade que os recursos possuem para executar a tarefa, isto é, as probabilidades que os recursos têm de ser escolhidos pelo holon produto. O campo seguinte é o modelo do

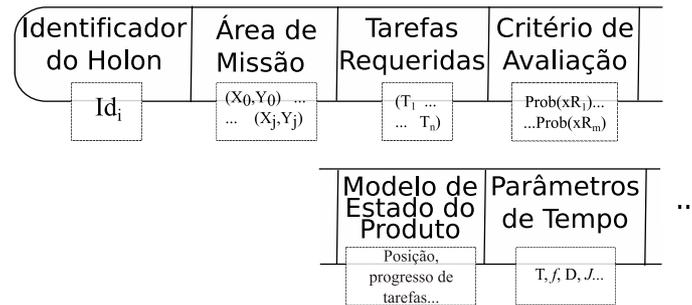


Figura 13: Estrutura do Holon Produto proposto (Fonte: Próprio autor).

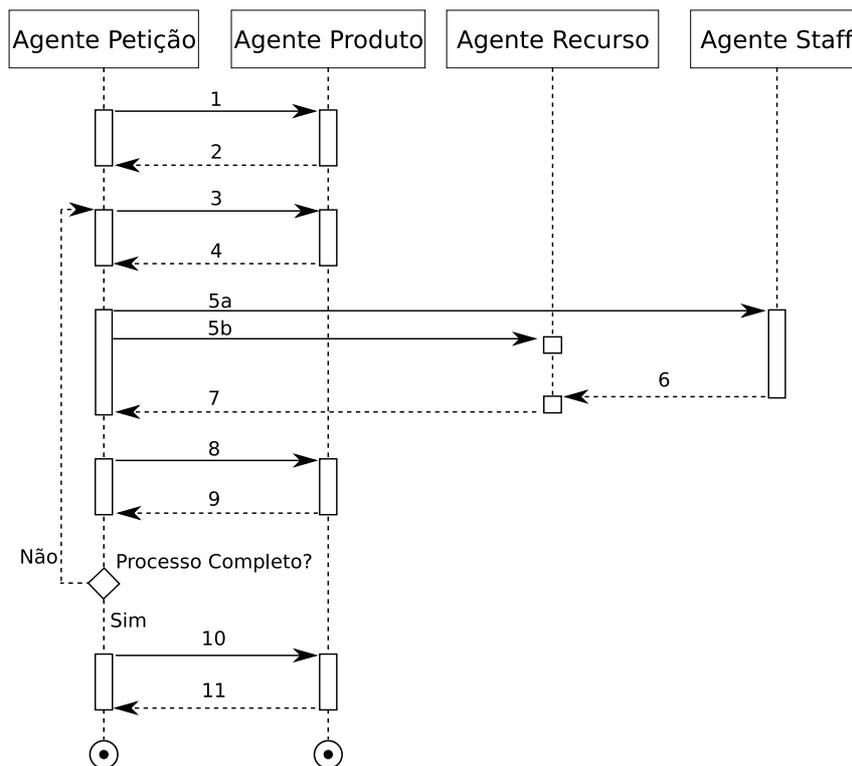


Figura 14: Diagrama de sequência UML da Interação Proposta (Fonte: Próprio autor).

estado da instância do produto físico que está sendo manufaturado, esse modelo possui a posição do produto, o progresso das tarefas executadas no produto, o registro das tarefas, etc. O sexto campo pode especificar os parâmetros de tempo, tais como períodos ou frequências nas quais as medições devem ser realizadas, atrasos, cronômetros, etc. Outros parâmetros adicionais podem ser definidos dependendo das necessidades de um produto específico.

O primeiro campo de identificação pertence tanto ao agente produto como ao agente petição. Os campos de área de missão (MA), tarefas de manufatura requeridas, modelo do estado do produto, e parâmetros de tempo são inseridos no agente produto. Finalmente o parâmetro de critério de avaliação é vinculado ao agente de petição.

### 5.3 Interação entre Agentes

A interação proposta entre agentes é inspirada na interação descrita na arquitetura PROSA (VALCKENAERS, 2009) e é apresentada na Figura 14.

A informação que é enviada entre os diferentes agentes é explicada detalhadamente nos itens a seguir:

- **Mensagens 1 e 2:** Quando um holon de produto é inserido na planta de manufatura, os seus respectivos agentes de petição e produto são criados. Para conferir que os agentes foram criados exitosamente e iniciar o cronômetro interno do agente de produto, o agente de petição (*order*) envia a mensagem 1 ao seu agente de produto correspondente. Como resposta, a mensagem 2 é enviada ao agente de petição. Essa resposta confere que o agente produto foi criado e o seu cronômetro interno iniciado.
- **Mensagens 3 e 4:** Na sequencia, o agente de petição solicita ao agente produto a tarefa seguinte a ser executada na mensagem 3. Posteriormente o agente de produto envia o nome da tarefa requerida ao agente de petição na mensagem 4. Desse modo, o agente de petição não possui informações detalhadas do processo de manufatura, porém, é capaz de executar sua logística a través da comunicação constante com os demais agentes do MAS.
- **Mensagens 5a e 5b:** Como apresentado na Figura 14, as mensagens 5 são duas e estão encarregadas de iniciar a execução de uma tarefa de manufatura no produto. A mensagem 5a cujo destino é o agente *staff* tem por objetivo enviar ao agente encarregado do transporte (*staff agent*) a identificação do produto e a identificação do agente recurso selecionado, desse modo, o agente *staff* recebe as informações necessárias para executar o transporte do material até o recurso selecionado. A mensagem 5b é destinada ao agente recurso selecionado, ela possui a identificação do produto junto com o nome da tarefa requerida pelo produto. Quando o agente de recurso receber a mensagem 5b, insere a identificação do produto no seu buffer de transporte (TB) e fica à espera da chegada do produto físico.
- **Mensagens 6 e 7:** Quando o agente *staff* terminar a execução de um transporte de materiais envia a mensagem 6 ao agente recurso informando a identificação do produto que foi transportado. Posteriormente o agente de recurso remove do buffer de transporte (TB) o registro desse produto e insere ele no buffer de processamento (MB). Os materiais registrados no buffer de processamento ficam à espera de ser processados pelo recurso, tendo em conta que cada recurso só pode executar uma tarefa de manufatura por vez. Finalmente, quando o agente de recurso terminar a execução da tarefa de manufatura requerida pelo produto, remove do buffer de processamento o registro do produto correspondente e envia a mensagem 7 para informar ao holon de produto que a tarefa requerida foi executada adequadamente no seu agente produto.
- **Mensagens 8 e 9:** A mensagem 8 informa ao agente produto que o seu modelo do estado do produto deve ser atualizado segundo a informação recebida na mensagem 7. Seguidamente, o agente produto envia a mensagem 9 indicando que o seu modelo foi atualizado adequadamente e informando se todas as tarefas de manufatura do produto foram totalmente executadas. Caso o produto precisar da execução de outra tarefa de manufatura, as mensagens 3 a 9 são novamente efetuadas, caso contrário, são executadas as mensagens 10 e 11.
- **Mensagens 10 e 11:** A mensagem 10 tem por objetivo finalizar o cronômetro interno do agente de produto que foi anteriormente iniciado na mensagem 1. Como

resposta, o agente produto para o seu cronômetro e envia o tempo total de manufatura do produto na mensagem 11. Desse modo, finaliza o ciclo de vida do holon produto.

## 5.4 Mecanismo de Tomada de Decisões

Para resolver o problema de agendamento de tarefas foi escolhida uma abordagem baseada na meta-heurística de otimização por colônia de formigas (ACO) (COLORNI et al., 1994; DORIGO; BONABEAU; THERAULAZ, 2000). Na proposta apresentada nesta dissertação os holons produto atuam como formigas e os recursos como o ambiente, portanto, os holons produto devem imitar o comportamento das formigas e viajar através dos recursos inserindo rastros de feromônio que depois podem ser seguidos por outros holons produto. Deste modo, os recursos devem fornecer posições onde os holons produto possam residir, um mecanismo para armazenar e manter os feromônios artificiais nessas posições e um mecanismo de propagação que permita aos holons a movimentação através da topologia do chão de fábrica, a leitura e atualização de feromônios nos recursos.

Como função objetivo, o problema de agendamento de tarefas proposto procura a minimização do tempo de manufatura. Esse tempo de manufatura, chamado de *Makespan* em inglês, deve ser definido detalhadamente para que se conheçam os fatores que o influenciam.

Nos sistemas de manufatura cada recurso possui a habilidade técnica de executar diferentes tarefas de manufatura e os meios para o transporte de materiais a novos recursos, portanto, para cada tarefa requerida pelo produto existe um recurso alocado para executá-la. Seja  $\mathbf{T}_{prod}$  o vetor que representa a lista de tarefas requeridas pelo produto  $prod$  de esquerda para a direita, e  $\mathbf{R}_{prod}$  o vetor que representa a lista do recurso onde o produto é inserido seguido dos recursos onde as tarefas requeridas são executadas de esquerda para a direita, a função objetivo pode ser descrita por 4.

$$\mathbf{min}: Makespan_{prod} = \sum_{i=1}^{n_{T_{prod}}} (t_t[R_{prod}(i), R_{prod}(i+1)] + t_p[R_{prod}(i+1), T_{prod}(i)]) \quad (4)$$

Onde  $Makespan_{prod}$  é o tempo de manufatura decorrido para que o produto  $prod$  seja totalmente manufaturado no chão de fábrica,  $n_{T_{prod}}$  é o número de tarefas requeridas pelo produto  $prod$ ,  $i$  é o índice dos vetores,  $t_t[R_{prod}(i), R_{prod}(i+1)]$  é o tempo de transporte entre o recurso no qual foi feita a alocação de recursos  $R_{prod}(i)$  e o recurso selecionado  $R_{prod}(i+1)$  para executar a tarefa  $T_{prod}(i)$  no produto  $prod$ , e  $t_p[R_{prod}(i+1), T_{prod}(i)]$  é o tempo de processamento que o recurso  $R_{prod}(i+1)$  precisa para executar a tarefa  $T_{prod}(i)$ .

Devido ao fato que os recursos só podem executar uma tarefa de manufatura por vez e às restrições de transporte de materiais, os recursos precisam de usar dois tipos de buffer para executar as suas tarefas: Buffer de manufatura ou processamento (**MB**) e buffer de transporte (**TB**). Os tempos de manufatura obtidos na planta vão depender, desse modo, dessas restrições de manufatura e transporte.

Já com a função objetivo esclarecida, os mecanismos de alocação e disseminação de informações, associados aos mecanismos de seleção de trilha e disseminação de informações explicados na Seção 3.2, podem ser especificados.

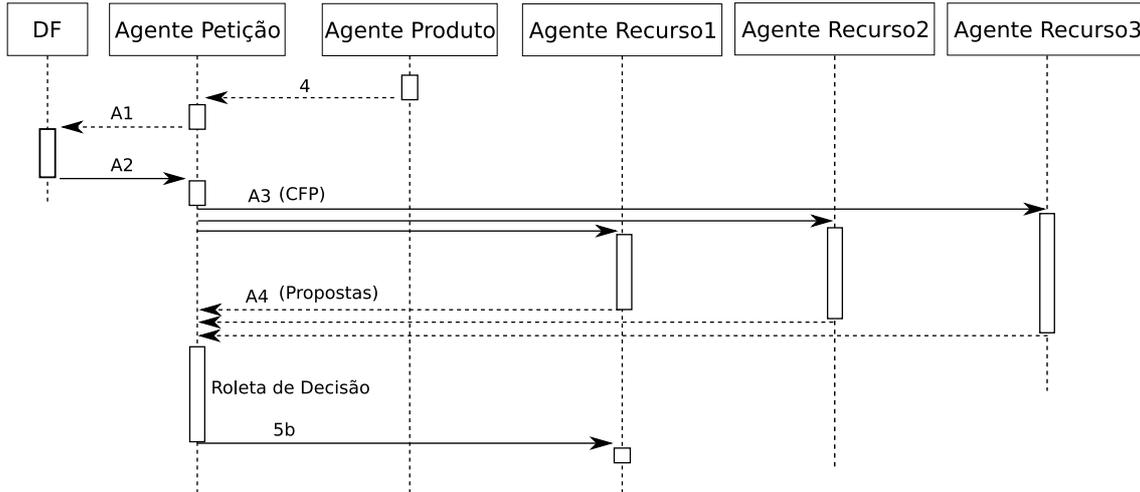


Figura 15: Interação proposta para alocação de produtos (Fonte: Próprio autor).

#### 5.4.1 Alocação de Produtos

Acontece quando um holon de produto localizado no recurso  $x$  escolhe novos recursos através da leitura dos rastros de feromônio apropriados, a saber, os valores de feromônio da tarefa requerida depositados no recurso atual e nos recursos vizinhos. Para efetuar esse processo, foi proposta a interação descrita na Figura 15.

Inicialmente, o agente de petição (*order agent*) envia a mensagem A1 para solicitar ao agente de páginas amarelas (DF) a lista dos recursos que possuem a habilidade de executar a tarefa que foi anteriormente solicitada pelo produto na mensagem 4. O agente de páginas amarelas envia a lista de possíveis candidatos na mensagem A2. Seguidamente, o agente petição solicita aos agentes recurso candidatos a concentração de feromônios relacionada à tarefa de manufatura requerida pelo produto nas mensagens A3. Os recursos candidatos enviam as suas propostas e o agente de petição espera a recepção de todas elas, para finalmente, calcular a probabilidade de selecionar os recursos candidatos conforme a seguinte fórmula descrita em (DORIGO; BONABEAU; THERAULAZ, 2000):

$$Prob(R_x, R_y, T_z) = \begin{cases} \frac{\tau(R_y, T_z)\eta(R_x, R_y)}{\sum_{j \in \mathfrak{R}} \tau(R_j, T_z)\eta(R_x, R_j)} & \text{Se } R_y \in \mathfrak{R} \\ 0 & \text{Caso contrário} \end{cases} \quad (5)$$

Onde  $Prob(R_x, R_y, T_z)$  é a probabilidade de escolher o recurso  $R_y$  partindo do recurso  $R_x$  para executar a tarefa  $T_z$ .  $\mathfrak{R}$  é o conjunto de recursos candidatos possíveis, a saber, os recursos vizinhos que possuem a habilidade de executar a tarefa requerida e também pertencem à Área de Missão (MA).  $\tau(R_y, T_z)$  é o valor da concentração de feromônios depositado no recurso  $R_y$  relacionado com a tarefa requerida  $T_z$  e  $\eta(R_x, R_y)$  é uma heurística de distância inversamente proporcional ao tempo mínimo de transporte factível entre os recursos, i.e.,  $\eta(R_x, R_y) = 1/(t_t[R_x, R_y])$  onde  $t_t[R_x, R_y]$  é o tempo de transporte mínimo que pode ser atingido para transportar materiais desde o recurso  $R_x$  até o possível recurso candidato  $R_y$ .

Finalmente, o agente de petição seleciona o seguinte recurso usando o método da roleta de decisão (*decision wheel*). Uma roleta virtual é criada e a cada recurso candidato lhe é atribuída uma ranhura com uma largura definida. Quanto maior a percentagem que o recurso tem de ser selecionado, maior a largura da sua ranhura correspondente.

Finalmente um giro aleatório da roleta é feito e uma ranhura específica nela é selecionada, o giro é simulado pela geração de um número aleatório com distribuição uniforme entre 0 e 1 e a seleção é simulada pela comparação desse número com as percentagens calculadas para cada recurso. Desse modo, um recurso do MAS é escolhido e o agente petição pode enviar a mensagem 5b ao recurso selecionado.

#### 5.4.2 Disseminação de Informações

A disseminação de informações é executada depois da mensagem 11. Nessa mensagem, o agente petição recebe o tempo total de manufatura (*Makespan*) do agente produto. Depois de obtido, são efetuadas as interações apresentadas na Figura 16. O agente petição envia o tempo de manufatura obtido aos agentes recurso que pertencem à área de missão e ao agente centralizado *staff* na mensagens D1. O agente *staff* obtém esse tempo de manufatura e armazena ele no seu banco de dados, os demais agentes de recurso executam a atualização de feromônios.

A atualização de feromônios ocorre com o depósito de feromônios em todos os recursos e tarefas que foram escolhidas pelo agente petição, caso contrário, os feromônios são evaporados. A atualização de feromônios, segundo (DORIGO; BONABEAU; THERAU-LAZ, 2000), pode ser escrita como:

$$\tau(R_x, T_z) \leftarrow (1 - \rho)\tau(R_x, T_z) + \Delta\tau(R_x, T_z) \quad (6)$$

Onde  $\tau(R_x, T_z)$  é a concentração de feromônio armazenado no recurso  $R_x$  para executar a tarefa  $T_z$ ,  $\rho$  é a taxa de evaporação de feromônios, a qual deve ser ajustada empiricamente antes da execução do algoritmo e  $\Delta\tau(R_x, T_z)$  possui dois possíveis valores:

$$\Delta\tau(R_x, T_z) = \begin{cases} \frac{Q}{Makespan} & \text{se } R_x, T_z \text{ foi escolhido} \\ 0 & \text{Caso contrário.} \end{cases} \quad (7)$$

Desse modo, se o recurso  $R_x$  foi escolhido para executar a tarefa  $T_z$ , o valor de  $\Delta\tau(R_x, T_z)$  é  $Q/Makespan$  onde  $Q$  é uma constante que deve ser definida antes da execução da tomada de decisões e está relacionada à quantidade de feromônios inseridos por cada formiga. Desse modo, o valor de  $Q$  deve ser calibrado para que seja similar a um valor médio de *Makespan* esperado na planta de manufatura.

Caso contrário, se o recurso  $R_x$  não foi escolhido para executar a tarefa  $T_z$ , o valor de  $\Delta\tau(R_x, T_z)$  é 0 e a concentração de feromônios  $\tau(R_x, T_z)$  é evaporada como expressado em 7. Por tanto, quanto menor o tempo de manufatura *Makespan*, maior a quantidade de feromônios inseridos na trilha e maior a probabilidade de ser ela escolhida por futuros produtos inseridos na planta.

Finalmente, o agente petição recebe as mensagens D2 verificando que todos os agentes receberam a mensagem D1. Quando o agente petição receber todas as mensagens D2, espera durante um segundo e termina a sua execução no MAS.

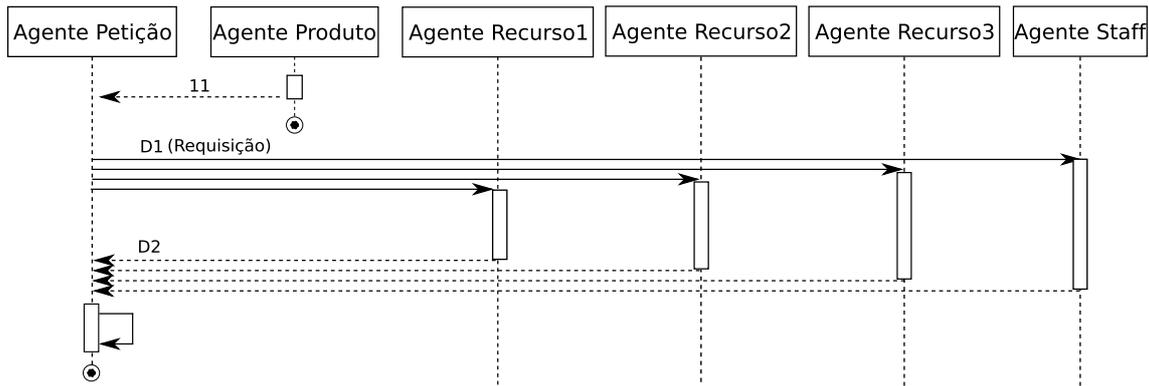


Figura 16: Interação proposta para disseminação de informações (Fonte: Próprio autor).

### 5.4.3 Exemplo do funcionamento do mecanismo de tomada de decisões

Para ilustrar o mecanismo de tomada de decisões, considere uma planta cuja topologia pode ser representada pelo grafo da figura 17 onde  $R_0$ ,  $R_1$  e  $R_2$  representam os recursos da planta de manufatura. Nela, vai se manufaturar um produto  $d$  cujo vetor de tarefas requeridas é  $\mathbf{T}_d = (T_1, T_2)$ .

Para executar o processo de tomada de decisões, as condições iniciais de tempos de transporte entre recursos e de tempos de processamento deles devem ser definidos além dos parâmetros inerentes ao mecanismo de tomada de decisões. Esses parâmetros são mostrados nas tabelas 3, 4a e 4b.

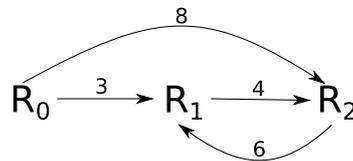


Figura 17: Topologia demonstrativa de planta de manufatura com tempos de transporte entre recursos [s] (Fonte: Próprio autor).

Parâmetro	Valor
Concentração de Feromônio Inicial	10
Taxa de evaporação ( $\rho$ )	0.05
Constante ( $Q$ )	15

Tabela 3: Parâmetros definidos para exemplificar o mecanismo de tomada de decisões.

#### 5.4.3.1 Alocação de Produtos

Inicialmente o holon produto é inserido no recurso  $R_0$ , aí, o agente petição solicita ao agente produto o nome da tarefa a ser executada (Mensagem 3). Na sequência o agente produto responde ao agente petição na mensagem 4 o nome da tarefa que ele solicita. Como nenhuma tarefa da sua lista  $\mathbf{T}_d = (T_1, T_2)$  ainda foi executada nele, o agente produto responde com  $T_d(1) = T_1$ . Já com o nome da tarefa definido, o agente de petição passa a solicitar ao agente DF a lista de recursos que possuem a habilidade de executar a tarefa  $T_1$  na planta de manufatura (Mensagens A1 e A2). Posteriormente, o agente

from \ to	$R_1$	$R_2$
$R_0$	3	8
$R_1$	0	4
$R_2$	6	0

(a) Condições iniciais tempos de transporte.

$R_i$ \ Tasks	$T_1$	$T_2$
$R_0$	-	2
$R_1$	3	-
$R_2$	2	3

(b) Condições iniciais tempos de processamento.

Tabela 4: Condições iniciais dos tempos de transporte e processamento na planta demonstrativa [s].

de petição solicita aos recursos que possuem a habilidade de executar  $T_1$  a concentração de feromônios relacionados à tarefa solicitada, e eles respondem com essa concentração (Mensagens A3 e A4). Quando todas as mensagens A4 são recebidas pelo agente de petição é feito o cálculo de probabilidades para selecionar os recursos candidatos.

As probabilidades de selecionar os recursos neste caso são:

- $Prob(R_0, R_0, T_1) = 0$  Representa a probabilidade de selecionar o recurso  $R_0$  partindo do recurso  $R_0$  para executar a tarefa  $T_1$ . Essa probabilidade equivale neste caso a 0%, pois o recurso  $R_0$  não possui a capacidade de executar a tarefa  $T_1$  como pode ser visualizado na Tabela 4b, conseqüentemente,  $R_0$  não pertence ao conjunto de recursos candidatos.
- $Prob(R_0, R_1, T_1)$  Representa a probabilidade de selecionar o recurso  $R_1$  partindo do recurso  $R_0$  para executar a tarefa  $T_1$ .
- $Prob(R_0, R_2, T_1)$  Representa a probabilidade de selecionar o recurso  $R_2$  partindo do recurso  $R_0$  para executar a tarefa  $T_1$ .

Sabendo que todas as concentrações de feromônio  $\tau(R_x, T_z)$  possuem um valor inicial de 10 como definido na Tabela 3, e que as heurísticas de distância entre recursos dependem dos tempos de transporte definidos na tabela 4a, as anteriores probabilidades podem ser calculadas usando 5 do seguinte modo:

$$\begin{aligned}
 \bullet \text{ } Prob(R_0, R_1, T_1) &= \frac{\tau_{(R_1, T_1)} \eta_{(R_0, R_1)}}{\tau_{(R_1, T_1)} \eta_{(R_0, R_1)} + \tau_{(R_2, T_1)} \eta_{(R_0, R_2)}} \\
 &= \frac{10 \cdot (1/3)}{10 \cdot (1/3) + 10 \cdot (1/8)} \approx 0.73 \\
 \bullet \text{ } Prob(R_0, R_2, T_1) &= \frac{\tau_{(R_2, T_1)} \eta_{(R_0, R_2)}}{\tau_{(R_1, T_1)} \eta_{(R_0, R_1)} + \tau_{(R_2, T_1)} \eta_{(R_0, R_2)}} \\
 &= \frac{10 \cdot (1/8)}{10 \cdot (1/3) + 10 \cdot (1/8)} \approx 0.27
 \end{aligned}$$

Para finalizar o processo de alocação, deve-se definir o recurso que vai executar a tarefa  $T_1$  através do método da roleta. É criada uma roleta virtual com uma quantidade de ranhuras equivalentes ao conjunto de recursos candidatos e as larguras correspondentes. Deste modo, a ranhura correspondente ao candidato  $R_1$ , possui uma probabilidade

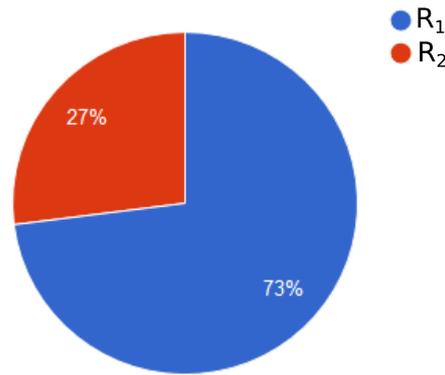


Figura 18: Roleta para escolher entre os recursos candidatos para executar a tarefa  $T_1$ , partindo do recurso  $R_0$  (Fonte: Próprio autor).

aproximada de 73% de ser escolhida e o candidato  $R_2$ , de 27% como mostrado na figura 18. Finalmente é gerado um número aleatório com distribuição uniforme entre 0 e 1 que simula um giro da roleta e indica uma posição nela. Por exemplo, caso o número aleatório for menor de 0.73 o recurso escolhido vai ser  $R_1$  e caso contrário, o agente de petição vai escolher o recurso  $R_2$ .

Supondo neste caso, que o número aleatório gerado foi 0.85, o agente petição correspondente escolhe o recurso  $R_2$  para executar a tarefa  $T_1$  e envia a mensagem 5b ao recurso  $R_2$  e a mensagem 5a ao agente (*Staff*) encarregado do transporte de materiais a  $R_2$ . Quando o recurso  $R_2$  terminar a execução da tarefa  $T_1$ , o agente petição atualiza o estado do produto, e percebe através da mensagem 9 que o seu produto ainda não está totalmente manufaturado, portanto, outra alocação do produto é necessária e devem ser repetidas as mensagens 3 e 4. Nessas mensagens é identificado que a nova tarefa requerida pelo produto é  $T_d(2) = T_2$ , e as novas probabilidades são calculadas:

- $Prob(R_2, R_0, T_2) = 0$  Representa a probabilidade de escolher o recurso  $R_0$  partindo do recurso  $R_2$  para executar a tarefa  $T_2$ . Neste caso, a probabilidade de escolher o recurso  $R_0$  é de 0%, porque mesmo que recurso  $R_0$  possa executar a tarefa  $T_2$  como mostrado na Tabela 4b, ele não pertence à área de missão devido que não existem meios para o transporte de materiais desde o recurso  $R_2$  ao recurso  $R_0$ .
- $Prob(R_2, R_1, T_2) = 0$  Representa a probabilidade de escolher o recurso  $R_1$  partindo do recurso  $R_2$  para executar a tarefa  $T_2$ . Neste caso, a probabilidade de selecionar o recurso  $R_1$  é de 0%, porque o recurso  $R_1$  não possui a habilidade de executar a tarefa  $T_2$  como mostrado na tabela 4b. Desse modo, o recurso  $R_1$  não pertence ao conjunto de recursos candidatos.
- $Prob(R_2, R_2, T_2)$  Representa a probabilidade de escolher o recurso  $R_2$  partindo do recurso  $R_2$  para executar a tarefa  $T_2$ . Neste caso, a probabilidade de selecionar o recurso  $R_2$  pode ser calculada usando 5 do seguinte modo:

$$\blacktriangleright Prob(R_2, R_2, T_2) = \frac{\tau_{(R_2, T_2)} \eta_{(R_2, R_2)}}{\tau_{(R_2, T_2)} \eta_{(R_2, R_2)}} = \frac{10 \cdot (1/0.001)}{10 \cdot (1/0.001)} = 1$$

Note-se que foi utilizado o número 0.001 em vez de 0 inicialmente indicado na Tabela 4a para representar o tempo de transporte entre o recurso  $R_2$  e o mesmo recurso  $R_2$ , isto, para evitar problemas de divisão por 0.

Neste caso, o único recurso disponível para fazer a tarefa  $T_2$  é  $R_2$  e possui a probabilidade de 100% de ser escolhido. Finalmente, quando o recurso  $R_2$  terminar a execução da tarefa  $T_2$ , o processo de manufatura é totalmente terminado e o processo de disseminação de informações passa a ser executado.

#### 5.4.3.2 Disseminação de Informações

Os vetores de recursos  $\mathbf{R}_d$  e o vetor de tarefas  $\mathbf{T}_d$  correspondentes ao exemplo descrito anteriormente são  $\mathbf{R}_d = (R_0, R_2, R_2)$  e  $\mathbf{T}_d = (T_1, T_2)$ . Eles representam a lista dos recursos onde foi alocado o produto  $d$  incluindo no início da lista o recurso no qual foi inserido o produto no chão de fábrica e a lista das tarefas requeridas pelo produto respectivamente.

O tempo de manufatura  $Makespan_d$  do produto  $d$  inserido na planta pode ser facilmente estimado devido a que não foi inserido nenhum outro produto no chão de fábrica durante o seu processo de manufatura. Desse modo, não foram afetados nem os buffers dos recursos nem as possíveis restrições de transporte de materiais que a planta possa possuir, portanto, os tempos de processamento e transporte correspondem aos indicados nas condições iniciais nas Tabelas 4a e 4b. Lembrando que o vetor de tarefas requeridas pelo produto é  $\mathbf{T}_d = (T_1, T_2)$ , o número de tarefas requeridas pelo produto é  $n_{T_d} = 2$ , e que os tempos de transporte  $t_t$  e processamento  $t_p$  podem ser consultados nas tabelas 4a e 4b, o tempo de manufatura do produto pode ser estimado usando 4 do seguinte modo:

$$\begin{aligned}
 Makespan_d &= \sum_{i=1}^2 (t_t[R_d(i), R_d(i+1)] + t_p[R_d(i+1), T_d(i)]) \\
 &= t_t[R_d(1), R_d(2)] + t_p[R_d(2), T_d(1)] \\
 &\quad + t_t[R_d(2), R_d(3)] + t_p[R_d(3), T_d(2)] \\
 &= t_t[R_0, R_2] + t_p[R_2, T_1] + t_t[R_2, R_2] + t_p[R_2, T_2] \\
 &= 8 + 2 + 0 + 3 = 13 \text{ s}
 \end{aligned}$$

O processo de disseminação começa com o envio das mensagens D1 aos recursos que pertencem à área de missão e ao agente *Staff*, deste modo, a mensagem D1 é enviada unicamente aos recursos  $R_1$  e  $R_2$ . Ali, cada um deles atualiza a sua concentração de feromônios usando o tempo de manufatura obtido da mensagem D1, e os parâmetros associados à ACO os quais são  $\rho$  e  $Q$ , esses parâmetros foram definidos na Tabela 3 como 0.05 e 15 respectivamente. Para ilustrar o mecanismo de atualização de feromônios é necessário usar 6 e inicializar as variáveis que armazenam a concentração de feromônios, desse modo,  $\tau_{(R_1, T_1)} \leftarrow 10$ ;  $\tau_{(R_2, T_1)} \leftarrow 10$ ; e  $\tau_{(R_2, T_2)} \leftarrow 10$ . Posteriormente é executada a atualização. Supondo que o tempo de manufatura enviado pela mensagem D1 foi o estimado anteriormente de  $Makespan_d = 13 \text{ s}$ , a atualização de feromônios pode ser exemplificada da seguinte maneira:

- No recurso  $R_1$ :

- $\tau_{(R_1, T_1)} \leftarrow (1 - \rho)\tau_{(R_1, T_1)} + \Delta\tau_{(R_1, T_1)}$
- $\tau_{(R_1, T_1)} \leftarrow (1 - 0.05) \cdot 10 + 0$
- $\tau_{(R_1, T_1)} \leftarrow 9.5$

- No recurso  $R_2$ :

- $\tau_{(R_2, T_1)} \leftarrow (1 - \rho)\tau_{(R_2, T_1)} + \Delta\tau_{(R_2, T_1)}$   
 $\tau_{(R_2, T_1)} \leftarrow (1 - 0.05) \cdot 10 + 15/13$   
 $\tau_{(R_2, T_1)} \leftarrow 10.65$
- $\tau_{(R_2, T_2)} \leftarrow (1 - \rho)\tau_{(R_2, T_1)} + \Delta\tau_{(R_2, T_1)}$   
 $\tau_{(R_2, T_2)} \leftarrow (1 - 0.05) \cdot 10 + 15/13$   
 $\tau_{(R_2, T_2)} \leftarrow 10.65$

É importante salientar que a concentração de feromônios referente a  $\tau_{(R_1, T_2)}$  não existe porque o recurso  $R_1$  não possui a capacidade de executar a tarefa  $T_2$  como é ilustrado pelo símbolo “-” na tabela 4b.

Desse modo, se um novo produto é inserido na planta de manufatura, o seu agente de petição executa a tomada de decisões com essas novas concentrações de feromônio atualizadas, as quais são:  $\tau_{(R_1, T_1)} = 0.95$ ,  $\tau_{(R_2, T_1)} \approx 10.65$  e  $\tau_{(R_2, T_2)} \approx 10.65$ .

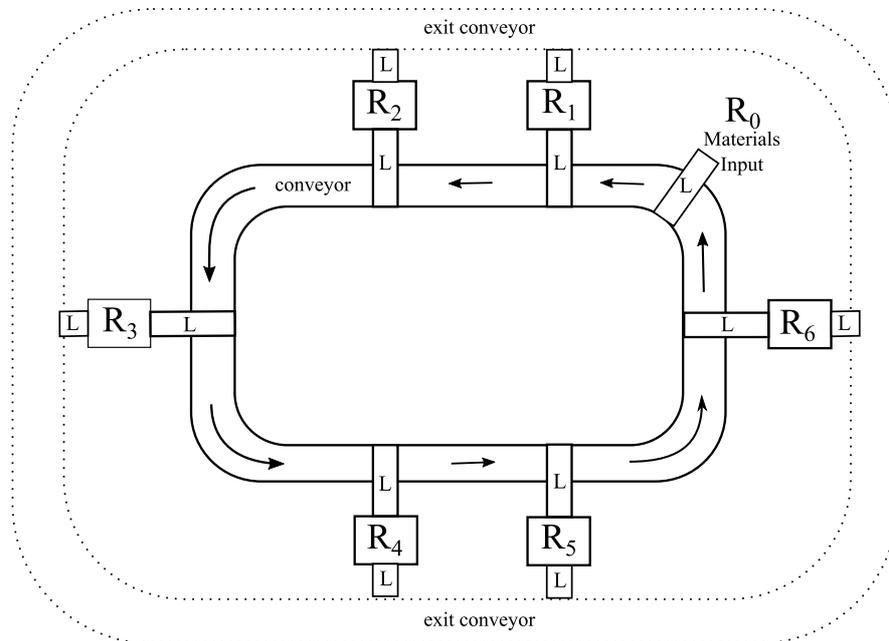
## 5.5 Planejamento dos Estudos de Caso

Para validar-se a arquitetura proposta, bem como para verificar-se que a interação entre agentes e o mecanismo de tomada de decisões permitem solucionar o problema de agendamento de tarefas derivado dos sistemas de manufatura flexível (FMS), devem ser planejados e implementados alguns estudos de caso. Eles permitirão validar a agentificação proposta, sua capacidade de simular sistemas de manufatura flexíveis e integrar no seu agendamento os conceitos de coordenação bio-inspirada incluídos no mecanismo de tomada de decisões que diminuam efetivamente tempo total de manufatura dos produtos.

O esquema da planta de manufatura que pretende-se simular, corresponde à mesma topologia usada para validação da tese de doutorado de Peixoto (PEIXOTO, 2016) e é apresentado na Figura 19. Como mostrado na figura, a topologia da planta consiste em uma esteira circular e sete recursos onde  $R_0$  é simplesmente o ponto de entrada de materiais e a saída pode ser qualquer um dos outros seis recursos dependendo da tarefa final requerida pelo produto. Deste modo, quando o recurso executar a ultima tarefa do produto ele envia o produto à esteira externa de saída. Cada recurso possui um carregador ( $L$ ) para a inserção/expulsão dos produtos aos recursos e as suas próprias habilidades. No estudo de caso definido as habilidades disponíveis no chão de fábrica são:  $T_1$ ,  $T_2$  ou  $T_3$ . Cada habilidade ou tarefa de manufatura pode ser no ambiente industrial uma tarefa de perfuração, trituração, extrusão, etc.

Dois cenários foram concebidos para simular e validar a proposta. O primeiro cenário, similar ao descrito em (PEIXOTO, 2012), cujas tarefas de manufatura estão descritas na Tabela 5a e o segundo cenário, onde as tarefas de manufatura foram modificadas e estão descritas na Tabela 5b. Nestas tabelas, o símbolo “-” indica que o recurso  $R_i$  não possui a habilidade de executar essa tarefa de manufatura e “✓” indica que a tarefa pode ser realizada no recurso  $R_i$ .

Desse modo, qualquer tipo de produto que precise das tarefas “ $T_1$ ”, “ $T_2$ ” ou “ $T_3$ ” pode ser manufaturado na planta. Como estudo de caso, se planeja a produção de dois tipos de produto:  $p_1$  e  $p_2$  cuja lista de tarefas requeridas é  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$  e  $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$  respectivamente. Ambos tipos de produto solicitam as tarefas da sua lista de esquerda à direita.



L=Loader  
R=Resource

Figura 19: Topologia da planta de manufatura (Fonte: Próprio autor).

$R_i$	$Tasks$		
	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	✓	-	✓
$R_2$	-	✓	✓
$R_3$	✓	✓	-
$R_4$	✓	-	✓
$R_5$	-	✓	✓
$R_6$	✓	✓	-

(a) Cenário I

$R_i$	$Tasks$		
	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	✓	✓	-
$R_2$	✓	✓	-
$R_3$	-	✓	✓
$R_4$	✓	-	✓
$R_5$	-	✓	✓
$R_6$	✓	-	✓

(b) Cenário II

Tabela 5: Cenários concebidos para validar a proposta.

## 6 IMPLEMENTAÇÃO, SIMULAÇÃO E VALIDAÇÃO EXPERIMENTAL

Já planejados os estudos de caso, a implementação, simulação e validação experimental podem ser efetuadas.

### 6.1 Implementação

Para implementar as funcionalidades do sistema flexível de manufatura pretendido, foi utilizada uma plataforma no *framework* JADE. O console de gerenciamento de plataformas é apresentado na Figura 20, nela pode ser visualizada a plataforma implementada, o agente de *staff* (*M*), os de recurso (*R1*, *R2*, *R3*, *R4*, *R5*, *R6*) e demais agentes necessários que permitem o correto funcionamento da plataforma (AMS, DF e rma).

#### 6.1.1 Agente *Staff*

Como descrito na Seção 5.2 o agente *staff* é um agente supervisor centralizado que ao mesmo tempo permite o transporte dos produtos inseridos no chão de fábrica. Deste modo, o agente *staff* deve possuir uma interface gráfica a fim de visualizar e proporcionar informação referente ao estado dos produtos físicos simulados e o estado das estações de trabalho (recursos). Além disso, o agente *staff* deve permitir a inserção de produtos na planta e armazenar a informação referente aos tempos de manufatura obtidos nela.

##### 6.1.1.1 Interface Gráfica

A interface gráfica desenvolvida é apresentada na Figura 21, e os dados visualizados nela são explicados nas Figuras 22a e 22b. Para a obtenção e visualização desses dados, os agentes de recurso e de produto enviam ao agente *staff* os dados que precisam ser visualizados a uma taxa de uma vez por segundo. Como apresentado na Figura 21, a interface gráfica desenvolvida para o agente *staff* possui três botões. O botão *Launch new Mission* permite inserir manualmente produtos na planta de manufatura, o botão *Stop* termina o processo de manufatura e finalmente o botão *Print Data* imprime na tela de Java a quantidade de produtos e recursos na planta junto com a lista dos produtos que foram totalmente manufaturados e o seu tempo de manufatura, além disso, gera um arquivo chamado de *Resultados.txt* com todos os dados obtidos pelo agente supervisor.

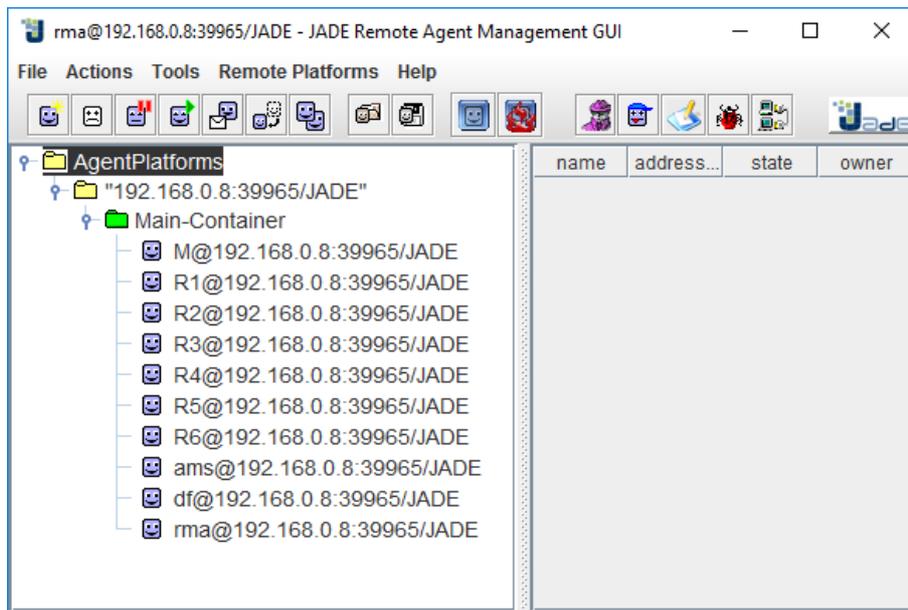


Figura 20: Console de gerenciamento de agentes (GUI do RMA) com agentes da proposta (Fonte: Próprio autor).

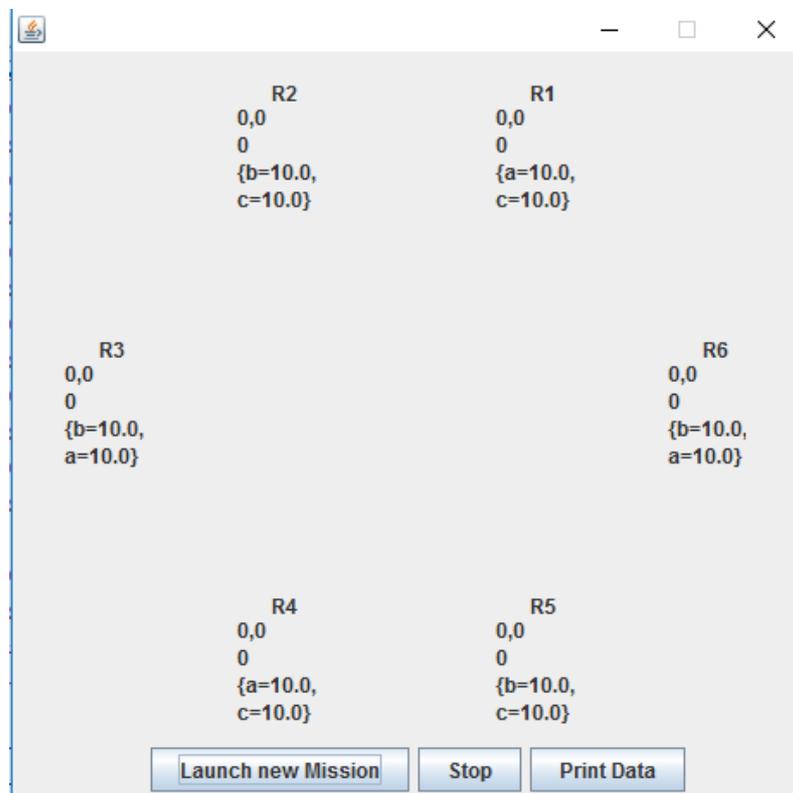


Figura 21: Interface gráfica do agente *staff* (Fonte: Próprio autor).

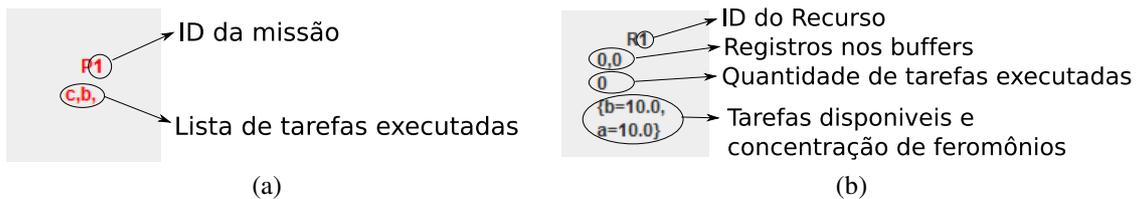


Figura 22: Informações apresentadas na interface gráfica (Fonte: Próprio autor).

### 6.1.1.2 Serviço de Transporte

Para o transporte de materiais foram definidas posições discretas e restrições de transporte de materiais. As possíveis posições que os produtos podem possuir dentro do chão de fábrica podem-se agrupar em três regiões: Na esteira central, dentro de algum recurso, ou na esteira externa de saída como apresentado na Figura 23.

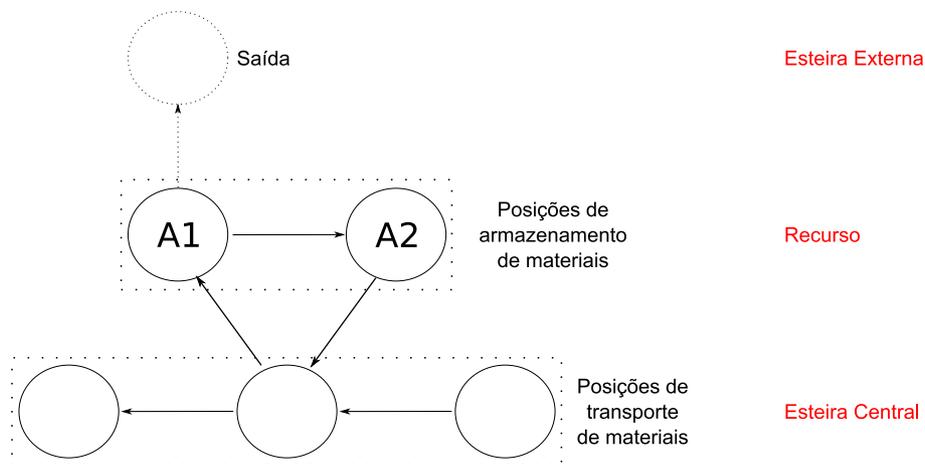


Figura 23: Posições estabelecidas na planta de manufatura simulada (Fonte: Próprio autor).

As restrições estabelecidas para o transporte de materiais foram:

- A quantidade de posições na esteira de transporte central é limitada a 26.
- Só um produto pode estar locado em uma posição que pertença á esteira de transporte central.
- A quantidade de produtos que podem estar locadas dentro dos recursos é ilimitada.

Para a movimentação dos produtos, o agente *staff* envia a cada segundo a todos os produtos no chão de fábrica sua próxima posição. Quando o agente de produto receber a mensagem de movimentação muda sua posição à indicada pelo agente *staff*. Para calcular a seguinte posição que cada produto deve ter, o agente *staff* leva em conta as restrições anteriormente descritas e soluciona possíveis concorrências de transporte a través de uma pila FIFO, deste modo, se uma posição da esteira central é requerida por dois produtos, o produto que foi inserido primeiro na planta de manufatura tem prioridade para se situar na próxima posição e o outro fica à espera na mesma posição anterior.

Como descrito nas restrições anteriormente mencionadas, as posições de armazenamento possuem uma capacidade infinita, porém a capacidade da posição A1 mostrada na

```

INFORMACIÓN: -----
Agent container Main-Container@192.168.0.3 is ready.
-----
R6, locado em (9,5) Possui os seguintes serviços: T1 T2
R3, locado em (2,5) Possui os seguintes serviços: T1 T2
R1, locado em (7,2) Possui os seguintes serviços: T1 T3
R2, locado em (4,2) Possui os seguintes serviços: T2 T3
R4, locado em (4,8) Possui os seguintes serviços: T1 T3
R5, locado em (7,8) Possui os seguintes serviços: T2 T3

```

Figura 24: Passo 1: Mensagem de inicialização dos recursos (Fonte: Próprio autor).

Figura 23 foi limitada a 10 produtos. Na posição A1 estão localizados os produtos que estão à espera de ser processados pelo recurso correspondente. Quando o processo de manufatura seja executado pelo recurso, é avaliado se todas as tarefas de manufatura foram efetuadas nele, nesse caso, o produto é movimentado na esteira externa. Se o produto precisar de mais uma tarefa é efetuado o agendamento dessa tarefa e é movimentado à posição A2 se a seguinte tarefa precisa ser executada em outro recurso diferente ao atual. Finalmente, os produtos localizados na posição A2 requerem uma posição na esteira central para ser transportados ao seguinte recurso alocado.

#### 6.1.1.3 Exemplo de lançamento de produtos na planta de manufatura

A seguir são detalhadas as informações mostradas na interface gráfica e no painel de saída com um exemplo no qual são lançados produtos na planta de manufatura correspondente ao estudo de caso. As informações são explicadas passo a passo.

- **Passo 1:** Quando a plataforma é iniciada, os agentes recurso imprimem na tela de saída uma mensagem inicial que indica que eles foram corretamente criados, além disso, a mensagem indica a sua posição e a lista de serviços/tarefas que eles podem executar como mostrado na Figura 24.
- **Passo 2:** Seguidamente, as interfaces gráficas dos agentes RMA e o agente *Staff* são visualizadas. A interface do agente *Staff* é apresentada na Figura 25, nela se mostra que quando o botão *Launch new Mission* é apertado, um novo produto chamado de  $P1$  é inserido na planta de manufatura na posição  $R_0$ . Esse produto  $P1$  requer uma lista de tarefas  $\mathbf{T}_{P1} = (T_1, T_2, T_3)$  que ainda não tem sido executadas nele. Além disso é mostrado que todas as concentrações de feromônios  $\tau_{rz}$  são inicialmente estabelecidas com um valor de 10.
- **Passo 3:** A seguir, o produto  $P1$  é alocado a um dos recursos da planta de manufatura para executar a tarefa  $T_1$ . O recurso alocado pelo algoritmo neste caso, foi  $R_4$ , como mostrado na Figura 26. Nela, se mostra que o buffer de transporte (TB) do recurso  $R_4$  aumenta a um registro, e na parte inferior, se mostra uma mensagem do agente  $O1$  indicando que ele alocou o produto  $P1$  ao recurso  $R_4$  para executar a tarefa  $T_1$ . O agente  $O1$  é o nome do agente de petição correspondente ao produto  $P1$  inserido.

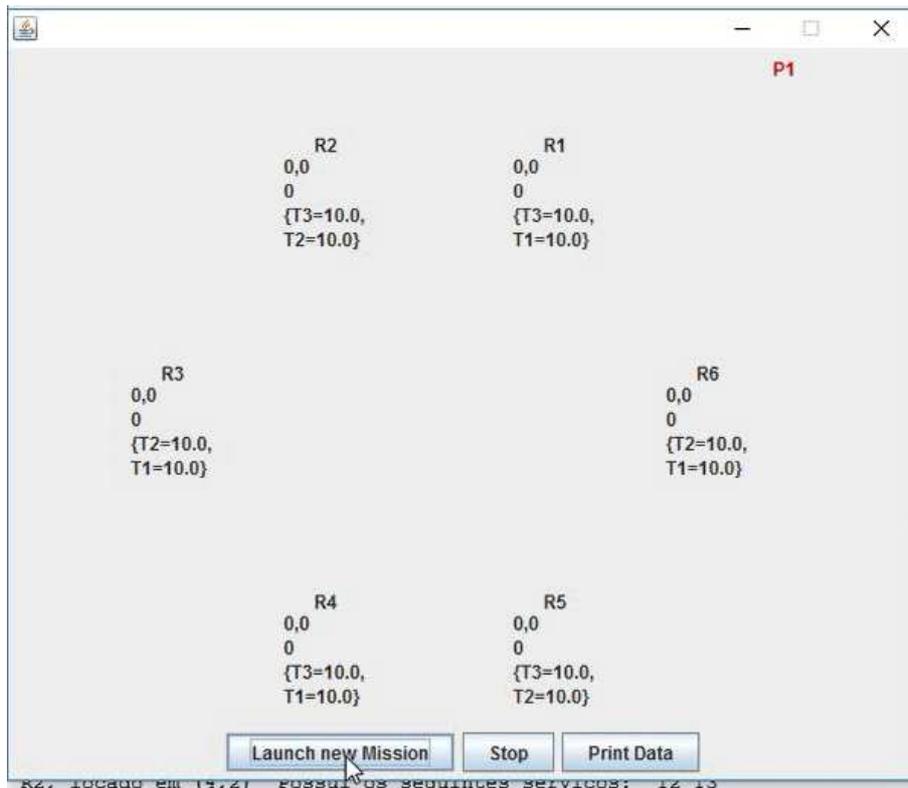


Figura 25: Passo 2: Interface gráfica do agente *Staff* quando um produto é lançado nele (Fonte: Próprio autor).

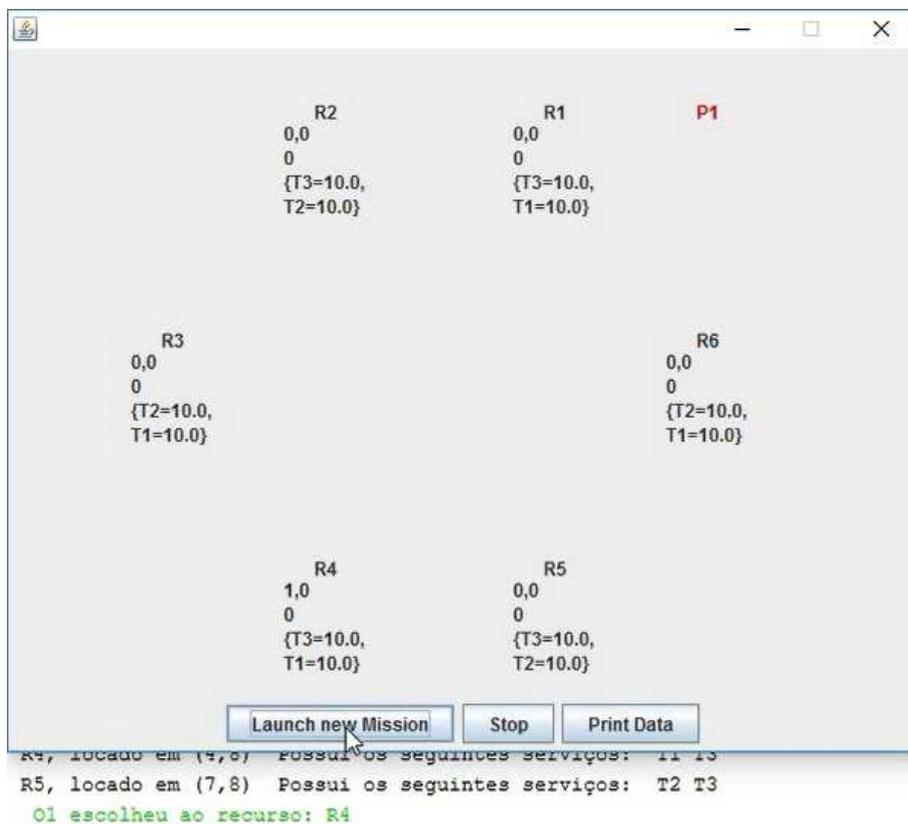


Figura 26: Passo 3: Interface gráfica do agente *Staff* com informação do recurso alocado atualizada (Fonte: Próprio autor).

- Passo 4:** Mais produtos podem ser inseridos na planta de manufatura em qualquer momento da execução dela, por exemplo, na Figura 27 se mostra que um produto adicional chamado de  $P2$  foi inserido na planta simulada. Esse produto requer a mesma lista de tarefas de  $P1$ , desse modo,  $T_{P2} = (T_1, T_2, T_3)$ . Na figura 27 também pode ser visualizada uma atualização de dados no buffer de transporte (**TB**) do recurso  $R_1$  devido a que o produto  $P2$  foi alocado nele. É importante visualizar também que o produto  $P1$  continuou o seu caminho na esteira central sem ser afetado pela inserção do novo produto  $P2$ .

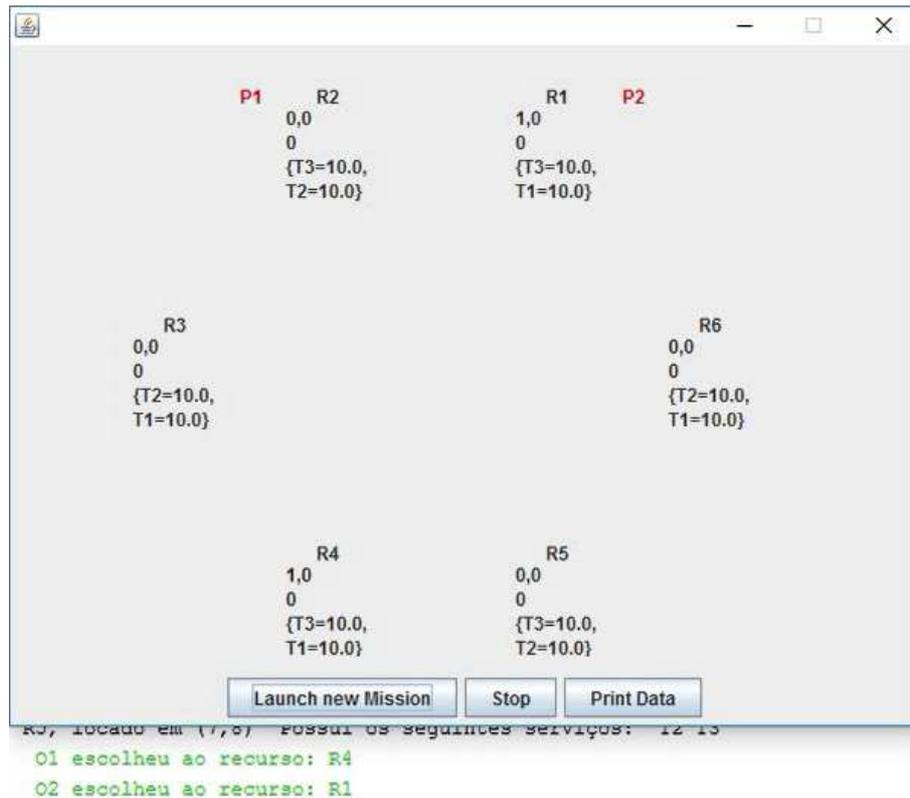


Figura 27: Passo 4: Interface gráfica do agente *Staff* com dois produtos na esteira central (Fonte: Próprio autor).

- **Passo 5:** Na Figura 28 se visualiza que o produto  $P2$  entra na posição A1 do recurso  $R_1$ , aí, o seu registro no buffer de transporte (TB) do recurso  $R_1$  é deletado e é adicionado ao buffer de manufatura (MB) do recurso  $R_1$ . Ao mesmo tempo o recurso  $R_1$  indica, na parte inferior da figura, que a tarefa  $T_1$  está sendo executada. O produto  $P1$  continua o seu caminho sem ser afetado.

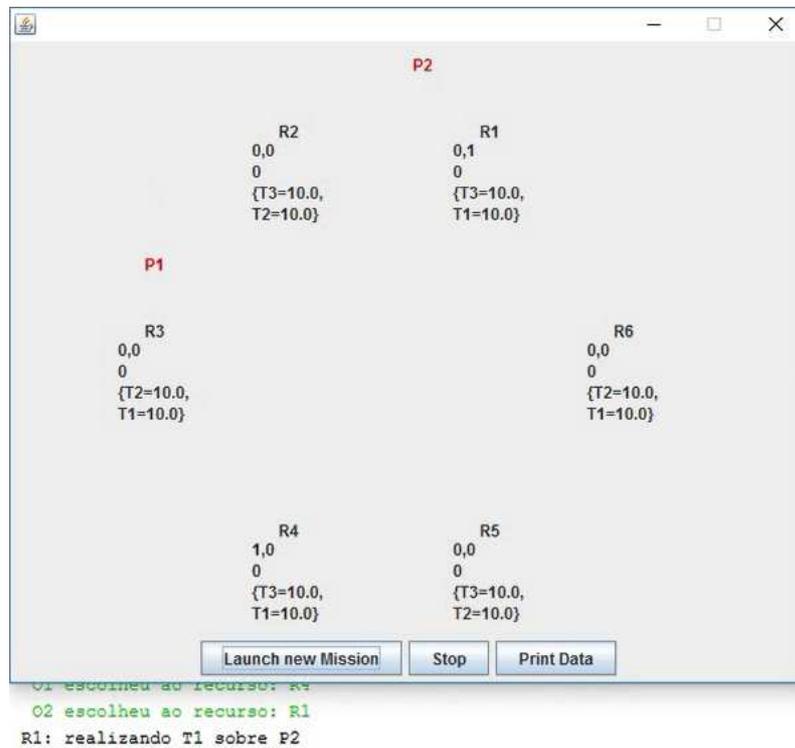


Figura 28: Passo 5: Interface gráfica do agente *Staff* com um recurso executando uma tarefa (Fonte: Próprio autor).

- **Passo 6:** A Figura 29 apresenta a situação da interface gráfica quando a tarefa  $T_1$  é finalmente executada no produto  $P2$ . Nela pode-se visualizar que o produto  $P2$  está na posição A2 do recurso  $R_1$ , que o seu registro no buffer de manufatura (MB) do recurso  $R1$  é eliminado, e que a execução da tarefa  $T_1$  no produto  $P2$  foi adicionada no registro de tarefas executadas pelo recurso  $R_1$ .

O agente de petição do produto  $P2$ , chamado de  $O2$ , alocou o seu produto no recurso  $R_2$  para executar a tarefa  $T_2$ , isso pode ser visualizado na mensagem impressa na parte inferior pelo agente  $O2$  e pelo fato que foi adicionado um registro no buffer de transporte (TB) do recurso  $R_2$ . Finalmente, se mostra que o produto  $P1$  continua o seu caminho na esteira central sem ser afetado, e está próximo ao recurso  $R_3$ .

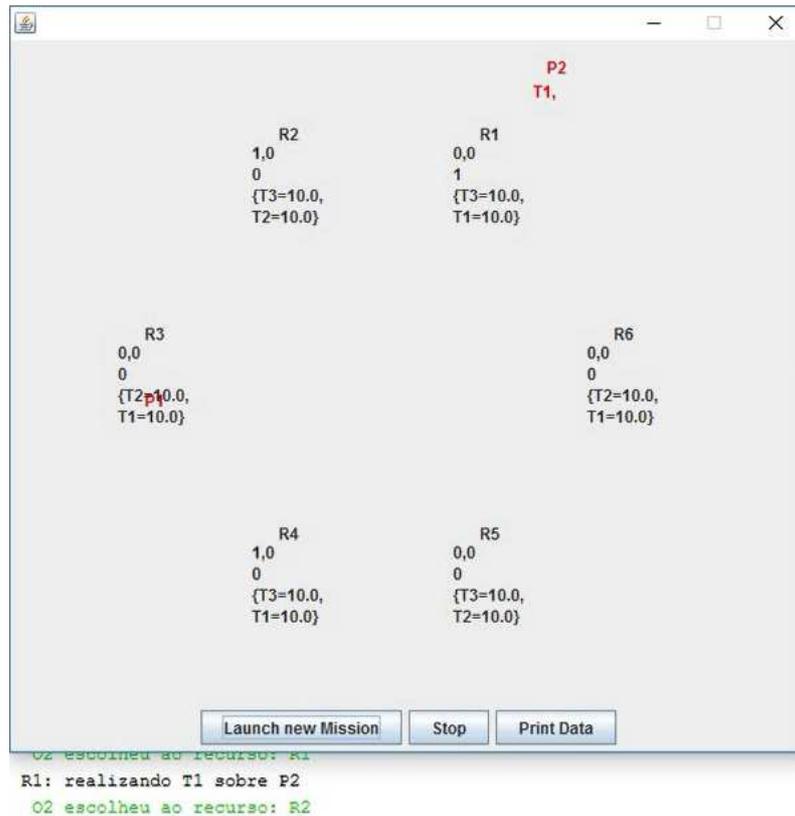


Figura 29: Passo 6: Interface gráfica do agente *Staff* com um recurso terminando uma tarefa (Fonte: Próprio autor).

- Passo 7:** A Figura 30 Mostra que tanto o produto  $P1$  como o produto  $P2$  chegaram aos recursos nos quais foram alocados  $R_4$  e  $R_2$  respectivamente. Desse modo, os buffers de transporte (**TB**) desses recursos diminuem e os de processamento (**MB**) aumentam em um registro. Neste caso, tanto o recurso  $R_4$  como o recurso  $R_2$  estão executando as tarefas correspondentes nos produtos ao mesmo tempo.
- Passo 8:** A manufatura totalmente finalizada do produto  $P2$  pode ser visualizada na Figura 31. Quando um produto finaliza o seu processo de manufatura, é executada a disseminação de informações, a qual envia aos recursos que pertencem à área de missão o tempo de manufatura do recurso que foi manufaturado  $Makespan_{P2}$ . Como a planta de manufatura possui uma topologia circular, a área de missão é composta por todos os recursos da planta, desse modo, o tempo de manufatura  $Makespan_{P2}$  é enviado a todos os recursos para que eles executem a atualização de feromônios. A atualização aumenta a concentração de feromônios em aqueles recursos e tarefas que foram selecionados pelo holon do produto  $P2$  e diminui as demais concentrações de feromônio da planta.

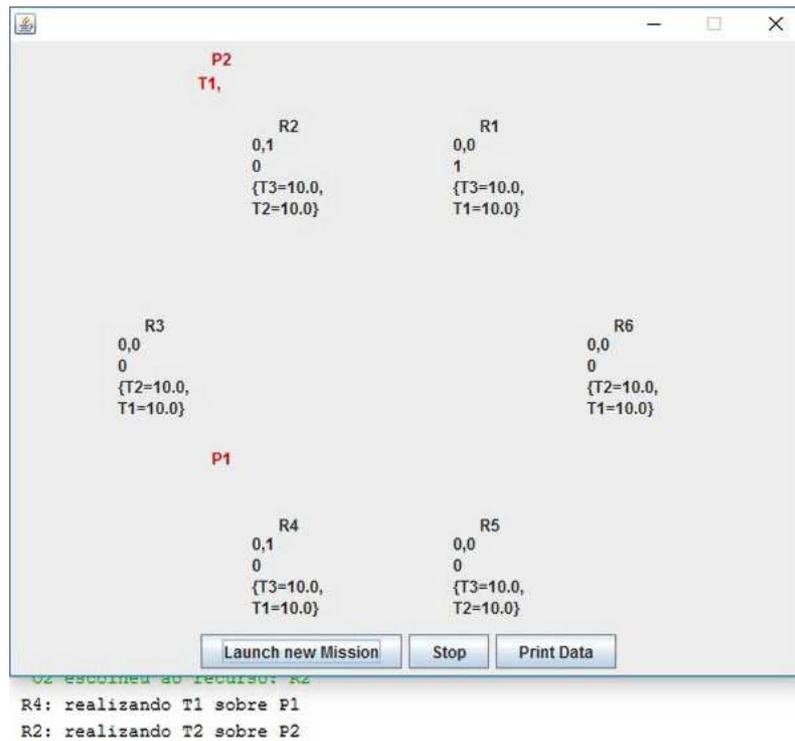


Figura 30: Passo 7: Interface gráfica do agente *Staff* com dois recursos executando tarefas ao mesmo tempo. (Fonte: Próprio autor).

Na parte inferior da Figura 31 pode-se observar que o tempo de manufatura do produto  $P2$  obtido na planta simulada foi de aproximadamente  $Makespan_{P2} \approx 15.4$  s e que as concentrações de feromônio correspondentes aos recursos e tarefas selecionadas pelo holon do produto  $P2$ , a saber,  $\tau_{(R_1, T_1)}$ ,  $\tau_{(R_2, T_2)}$  e  $\tau_{(R_2, T_3)}$  aumentaram a um valor aproximado de 19.5 e que as demais concentrações da planta diminuíram a um valor de 9.5.

Finalmente, pode-se perceber que antes da disseminação, o produto  $P1$  foi alocado ao recurso  $R_3$  devido a que foi adicionado um registro no seu buffer de transporte (TB).

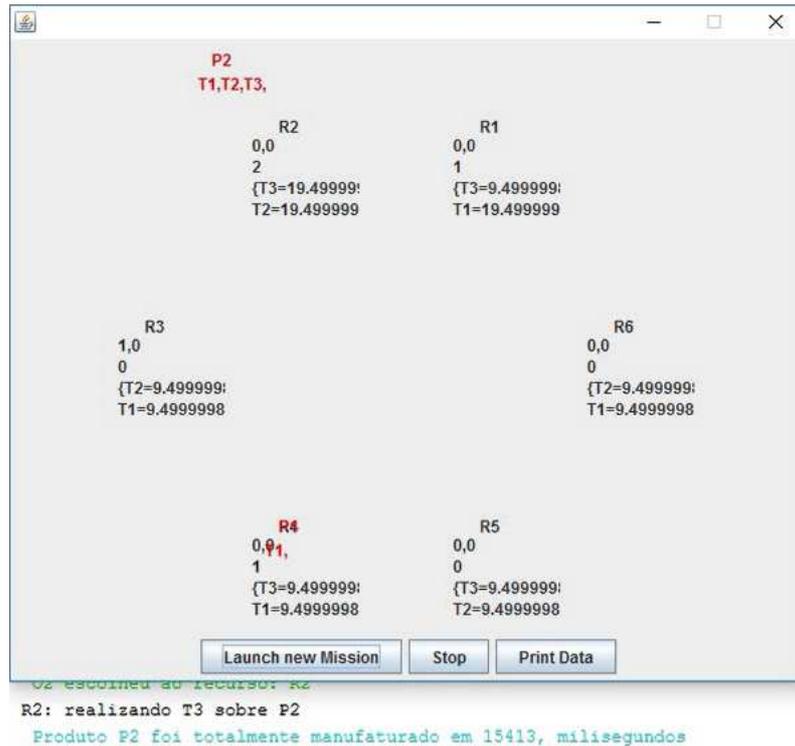


Figura 31: Passo 8: Interface gráfica do agente *Staff* depois de uma disseminação de informações. (Fonte: Próprio autor).

- **Passo 9:** Depois que os produtos estiverem totalmente manufaturados eles são enviados à esteira externa, isto é representado eliminando o produto manufaturado da interface gráfica. A Figura 32 apresenta a interface gráfica depois que os produtos  $P1$  e  $P2$  foram totalmente manufaturados. O vetor solução que representa os recursos nos quais foram executadas as tarefas do produto  $P2$  é  $\mathbf{R}_{P2} = (R_1, R_2, R_2)$ , e o vetor solução do produto  $P1$  é  $\mathbf{R}_{P1} = (R_4, R_3, R_2)$ .

Para concluir, pode-se observar na parte inferior da Figura 32 que o tempo de manufatura  $Makespan_{P1}$  obtido foi de aproximadamente  $70.4s$  e que as concentrações de feromônio no final da execução são maiores em aqueles recursos e tarefas selecionadas pelos holons de produto, os quais neste caso foram:

- $\tau_{(R_1, T_1)} \approx 18.52$ : Devido a que o recurso  $R_1$  executou tarefa  $T_1$  no produto  $P2$ .
- $\tau_{(R_2, T_2)} \approx 18.52$ : Devido a que o recurso  $R_2$  executou tarefa  $T_2$  no produto  $P2$
- $\tau_{(R_2, T_3)} \approx 19.24$ : Devido a que o recurso  $R_2$  executou tarefa  $T_3$  nos produtos  $P2$  e  $P1$ .
- $\tau_{(R_3, T_2)} \approx 9.74$ : Devido a que o recurso  $R_3$  executou tarefa  $T_2$  no produto  $P1$
- $\tau_{(R_4, T_1)} \approx 9.74$ : Porque o recurso  $R_4$  executou tarefa  $T_1$  no produto  $P1$

As demais concentrações de feromônios possuem uma quantidade aproximada de 9.02, como pode ser conferido na Figura 32.

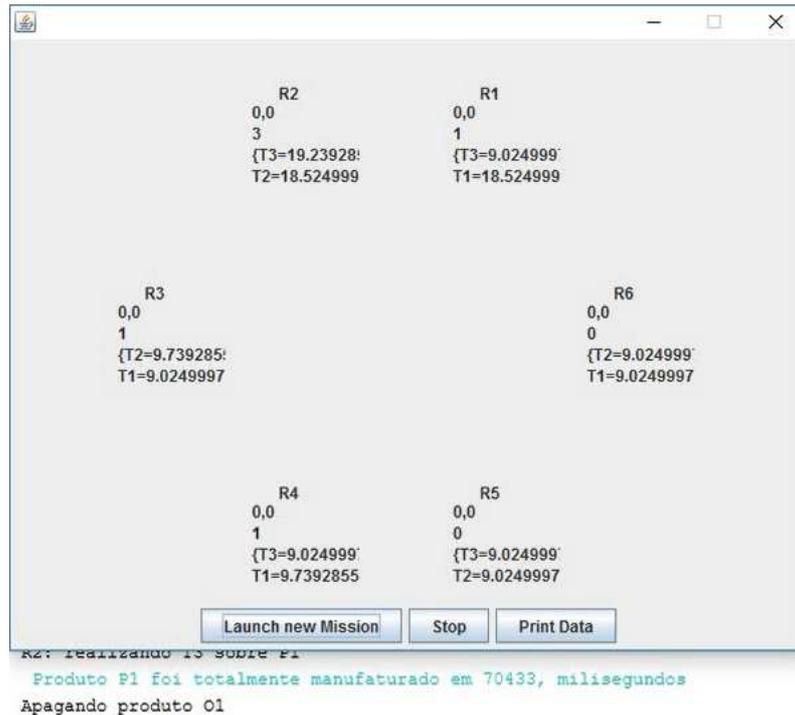


Figura 32: Passo 9: Interface gráfica do agente *Staff* depois de manufaturar os dois produtos *P1* e *P2*. (Fonte: Próprio autor).

#### 6.1.1.4 Comportamentos Implementados

Para implementar os serviços e ideias concebidas no agente *staff*, foram programados os seguintes comportamentos nele:

- **Comportamento CriarNovoProduto:** Do tipo *OneShotBehaviour*, é ativado quando o botão *Launch new Mission* estiver apertado. Permite a inserção manual de produtos na planta simulada.
- **Comportamento Start:** É um comportamento opcional do tipo *TickerBehaviour* que permite inserir automaticamente produtos na planta a tempos periodicamente estabelecidos.
- **Comportamento ObterDados:** Do tipo *CyclicBehaviour* encarregado da escuta das mensagens (INFORM) que permitem ao agente *staff* representar os dados atualizados no seu GUI.
- **Comportamento AtualizarPosições:** Comportamento tipo *TickerBehaviour*, encarregado do processo de transporte de materiais seguindo as restrições anteriormente descritas. É executado a cada segundo.
- **Comportamento AtualizarRecursoAlocado:** É um *CyclicBehaviour* encarregado da escuta das mensagens 5(ACCEPT) e obter os dados que ela envia para o correto transporte de materiais.
- **Comportamento FinManuf:** É um *CyclicBehaviour* encarregado da escuta das mensagens D1(REQUEST), obtenção dos tempos de manufatura enviados por esses mensagens,

### 6.1.2 Agente Recurso

Para implementar o caso de estudo, devem ser usadas seis instâncias do agente recurso, devido ao fato que o recurso  $R_0$  planejado no estudo de caso não oferece serviço nenhum. Deste modo, só basta garantir que os produtos sejam inseridos na posição estabelecida para  $R_0$ .

Para implementar as ideias concebidas na proposta foram formulados os seguintes comportamentos JADE no recurso:

- **Comportamento Monitor:** É um *TickerBehaviour* encarregado de enviar cada segundo os dados de visualização do recurso para que o agente *staff* possa representá-lo no seu GUI.
- **Comportamento DemandaOfertaServidor:** É um *CyclicBehaviour* encarregado da escuta da mensagem A3 (CFP), e de responder a ela com a mensagem A4 no processo de alocação de produtos.
- **Comportamento AtualizarBufferT:** É um *CyclicBehaviour* encarregado da escuta da mensagem 5 (ACCEPT). Atualiza o buffer de transporte (TB) e armazena em outro espaço de memória o ID do produto que solicita a tarefa de manufatura e o nome da tarefa que foi solicitada.
- **Comportamento AtualizarBufferM:** É um *CyclicBehaviour* encarregado da escuta da mensagem 6 (INFORM) e atualizar os buffers de transporte (TB) e processamento (MB).
- **Comportamento Manufaturar:** É um *CyclicBehaviour* que está avaliando constantemente o tamanho do buffer de processamento (MB) para determinar o estado do recurso (ocioso ou ocupado) e garante a execução de uma tarefa de manufatura por vez.
- **Comportamento FIPRequestResponder:** Do tipo *OneShotBehaviour*, encarregado da escuta da mensagem D1 (REQUEST), a atualização de feromônios e o envio da mensagem D2 (INFORM) no processo de disseminação de informações.

Como evidenciado no comportamento *Manufaturar* descrito anteriormente, o recurso foi modelado através de uma máquina de estados finita apresentada na Figura 33. Desse modo, quando o recurso detecta que a quantidade de registros no seu buffer de manufatura/processamento (MB) é maior a zero, passa do seu estado inicial de "ocioso" ao estado "ocupado". No estado ocupado, escolhe um dos registros do buffer de manufatura e é manufaturado. Posteriormente, quando o recurso terminar de executar a tarefa de manufatura requerida pelo produto, passa do estado "ocupado" ao "ocioso" e a quantidade de registros no buffer de manufatura (MB) é avaliada de novo.

### 6.1.3 Holon Produto

Como foi concebido na proposta, para implementar o holon produto dois agentes foram desenvolvidos: O agente produto e o agente petição. O agente produto atua como um servidor de informações especializadas do processo de manufatura e o agente petição como o encarregado da sua logística. Deste modo, os comportamentos implementados neles foram:

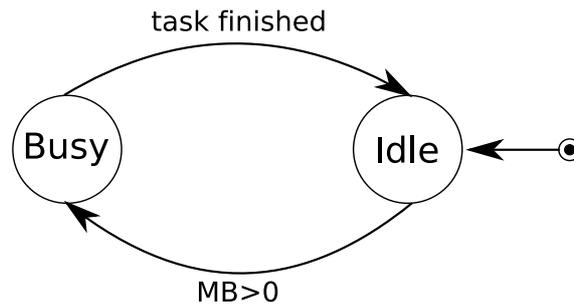


Figura 33: Máquina de estados finita do agente recurso (Fonte: Próprio autor).

#### • Agente Produto

- **Comportamento ProductServer:** É um *CyclicBehaviour* encarregado da escuta de todas as mensagens e execução das possíveis tarefas que pode desenvolver. Tais como: Atualizar posição, atualizar lista de tarefas executadas, início/parada do cronômetro etc.
- **Comportamento Finalizar:** Do tipo *WakerBehaviour* encarregado de deletar o agente produto quando o produto estiver totalmente manufaturado.

#### • Agente Petição

Inicialmente, o agente de petição envia a mensagem 1 e recebe a 2. Depois, são executados os seguintes comportamentos:

- **Comportamento PedirServiço:** É um comportamento do tipo *Behaviour* encarregado da comunicação das mensagens 3, 4, A1 e A2.
- **Comportamento ExecutarTarefa:** Do tipo *Behaviour* encarregado do envio ou escuta das mensagens envolvidas no FIPA-ContractNet-Protocol as quais são A3, A4, 5 e 7. E encarregado das mensagens 8 e 9, além de executar da roleta de decisão no processo de alocação.
- **Comportamento AtualizarFeromonios:** É um *Behaviour* encarregado da comunicação das mensagens 10, 11, D1 e D2 no processo de disseminação de informações.
- **Comportamento Finalizar:** Do tipo *WakerBehaviour* encarregado de deletar o agente petição quando o produto estiver totalmente manufaturado e todas as mensagens D2 recebidas.

## 6.2 Simulação

Para simular o estudo de caso, devem-se definir os parâmetros relativos ao mecanismo de tomada de decisões e as condições iniciais da planta. Os parâmetros e condições definidas são apresentados nas Tabelas 6, 7 e 8. Nas Tabelas 7 e 8 apresentam em segundos os tempos mínimos factíveis que a planta de manufatura pode oferecer para transportar e processar os produtos sendo as Tabelas 8a e 8b as correspondentes ao primeiro e segundo cenário respetivamente.

Parâmetro	Valor
Concentração de Feromônio Inicial	10
Taxa de evaporação ( $\rho$ )	0.05
Constante ( $Q$ )	50

Tabela 6: Parâmetros definidos para o mecanismo de tomada de decisões.

from \ to	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$
$R_0$	3	6	11	16	19	24
$R_1$	0	6	11	16	19	24
$R_2$	26	0	8	13	16	21
$R_3$	21	24	0	8	11	16
$R_4$	16	19	24	0	6	11
$R_5$	13	16	21	26	0	8
$R_6$	8	11	16	21	24	0

Tabela 7: Condições iniciais dos tempos de transporte entre recursos [s].

$R_i$ \ Tasks	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	2	-	2
$R_2$	-	2	2
$R_3$	2	2	-
$R_4$	2	-	2
$R_5$	-	2	2
$R_6$	2	2	-

(a) Condições iniciais cenário I

$R_i$ \ Tasks	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	2	2	-
$R_2$	2	2	-
$R_3$	-	2	2
$R_4$	2	-	2
$R_5$	-	2	2
$R_6$	2	-	2

(b) Condições iniciais cenário II

Tabela 8: Condições iniciais dos tempos de processamento nos cenários concebidos [s].

Para a obtenção de resultados pode ser manufaturado qualquer tipo de produto que precise das tarefas " $T_1$ ", " $T_2$ " ou " $T_3$ ". Na simulação decidiu-se manufaturar os tipos de produto  $p_1$  e  $p_2$  os quais requerem uma lista de três tarefas de manufatura, a saber,  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$  e  $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ . Deste modo, a planta de manufatura oferece uma quantidade máxima de  $6^3$  possíveis soluções a este tipo de produtos, porém, devido ao fato que nos dois cenários concebidos cada tarefa de manufatura só pode ser executada por quatro possíveis recursos, a quantidade de possíveis soluções nos cenários é reduzida a  $4^3$ . Essas possíveis soluções ao problema de agendamento podem ser representadas por uma lista dos recursos que foram escolhidos para executar cada um dos processos de manufatura requeridos pelo produto deste modo, as soluções para manufaturar um produto cuja lista de tarefas requeridas seja  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ , são um conjunto de três recursos que executam cada tarefa de esquerda à direita incluindo, no início da lista, o recurso no qual o produto é inserido na planta. (e.g.  $\mathbf{R}_{p_1} = (R_0, R_1, R_2, R_4)$  onde  $R_0$  é o recurso no qual o produto foi inserido na planta,  $R_1$  o recurso que executa a tarefa  $T_1$ ,  $R_2$  o recurso que executa a tarefa  $T_2$  e  $R_4$  o recurso que executa a tarefa  $T_3$ ).

## 6.3 Validação Experimental

### 6.3.1 Métricas de comparação

Para validar o mecanismo de agendamento, arquitetura e interação propostas, devem ser estabelecidas algumas métricas que dimensionem ou qualifiquem as características ou requisitos do sistema a fim de permitir uma análise comparativa. Certas métricas foram listadas e definidas como não pertinentes para a análise proposta, como: *Manter produção com inserção de estações* (pelo fato de possuir um agente centralizado de transporte a inserção requer reprogramação nele), *variação do tempo de manufatura no início da execução* (pois o mecanismo de tomada de decisões possui um comportamento aleatório no início da sua execução), *custo*, entre outros.

As métricas definidas como adequadas para a análise são similares as propostas em (PEIXOTO, 2012) e são apresentadas na Tabela 9, junto com sua justificativa técnica, forma de obtenção e critério de avaliação. Elas são: *Manter produção com retirada de estações*, *produzir produtos customizados* e *obter tempos de manufatura mínimos*. A natureza das duas primeiras métricas são qualitativas e a terceira métrica possui uma natureza de tipo quantitativo.

<b>Métricas</b>	<b>Justificativa técnica</b>	<b>Forma de obtenção</b>	<b>Critério de avaliação</b>
Manter produção com retirada de estações.	Em caso de falha nas estações de manufatura (recursos) o MAS deve continuar o processamento de materiais.	Desligar estações de trabalho durante a execução de um teste de manufatura.	Conferir se o MAS possui a modularidade suficiente para continuar o processamento de materiais com menos estações de trabalho nas concebidas nos estudos de caso sem requerer alteração na programação.
Produzir produtos customizados.	A produção de produtos customizados é necessária em um mercado fortemente flexível	Produzir produtos com diferentes listas de tarefas requeridas.	Conferir se o sistema de manufatura simulado possui a capacidade de produzir produtos com diferentes listas de tarefas requeridas.
Obtenção de tempos de manufatura mínimos	A diminuição de tempos de manufatura permite maior agilidade e competitividade no mercado	Montar o sistema e fazer testes cronometrando o tempo de manufatura dos produtos	Avaliar se as soluções atingidas pelo mecanismo de tomada de decisões baseado em ACO são ótimas ou sub-ótimas.

Tabela 9: Métricas de comparação.

### 6.3.1.1 Manter produção com retirada de estações.

Como plano de teste para obtenção dos dados foi usado o cenário I e produtos do tipo  $p_1$  cuja lista de tarefas é  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ . Inicialmente é inserido de forma manual um produto do tipo  $P_1$  com todas as estações ligadas. Posteriormente, foram eliminadas uma a uma, as instâncias das quatro estações que possuem a habilidade de realizar a tarefa " $T_2$ ", as quais são  $R_2, R_3, R_5$  e  $R_6$ . Depois de cada eliminação de instância é inserido mais um produto e é avaliado o efeito que a retirada de estações tem na planta de manufatura.

A Tabela 10 apresenta os resultados observados para esta métrica. Segundo esses resultados, conclui-se então, que a retirada de estações do MAS não prejudica a continuidade de produção desde que exista pelo menos um recurso disponível que ofereça o processamento de cada uma das tarefas requeridas.

<b>Estações Retiradas</b>	<b>Efeito no observado MAS</b>
-	O algoritmo ofereceu a solução $\mathbf{R}_N=(R_0, R_4, R_2, R_2)$
$R_2$	O algoritmo ofereceu a solução $\mathbf{R}_N=(R_0, R_3, R_3, R_5)$ . Não houve necessidade de reprogramação.
$R_2$ e $R_3$	O algoritmo ofereceu a solução $\mathbf{R}_N=(R_0, R_1, R_6, R_1)$ . Não houve necessidade de reprogramação.
$R_2, R_3$ e $R_5$	O algoritmo ofereceu a solução $\mathbf{R}_N=(R_0, R_1, R_6, R_1)$ . Não houve necessidade de reprogramação.
$R_2, R_3, R_5$ e $R_6$	O algoritmo não consegue oferecer uma solução. O produto inserido agendou o recurso $R_1$ para efetuar a tarefa " $T_1$ ", posteriormente o produto correspondente ficou parado na posição A1 do recurso $R_1$ pois a planta não oferece nenhuma estação disponível para executar a tarefa " $T_2$ ". Não houve necessidade de reprogramação.

Tabela 10: Efeitos no sistema com retirada de estações.

### 6.3.1.2 Produzir produtos customizados e Obtenção de tempos de manufatura mínimos.

Para a obtenção de resultados nestas métricas, dois tipos de produtos foram manufaturados:  $p_1$  e  $p_2$ , sendo a lista de tarefas requeridas por eles  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$  e  $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$  respectivamente.

Os resultados obtidos no primeiro cenário são apresentados nas Tabelas 12 e 14. E os obtidos no segundo cenário nas Tabelas 16 e 18. Cada tabela apresenta 10 testes de simulação, cada teste representa uma simulação na qual foram manufaturados 100 produtos do mesmo tipo e inseridos no chão de fábrica a uma taxa fixa de uma cada 4 segundos. A solução obtida em cada teste, apresentada na última coluna das tabelas, é aquela que possui a maior concentração de feromônios no final do teste. As concentrações finais de feromônios obtidas em cada teste, podem ser consultadas no Anexo A.

Como esperado, os tempos de manufatura médios obtidos em todos os testes (*Mean makespan*), são maiores do que o tempo de manufatura atingido pelo mecanismo de tomada de decisões devido à busca aleatória realizada pelos produtos do teste no estado transitório inicial da otimização por colônia de formigas (ACO). A continuação são apresentados e detalhados os resultados obtidos para cada um dos casos estudados.

Lembrando que a planta pode oferecer 64 possíveis soluções, e que os tempos de manufatura (*Makespan*) de cada solução podem ser estimados usando 4, as cinco melhores soluções que a planta pode oferecer para manufaturar um produto do tipo  $p_1$  no cenário I são apresentadas na Tabela 11 junto com seus respectivos tempos estimados de manufatura em ordem ascendente. Como mostrado na Tabela 11, a solução ótima no caso do cenário I, produto  $p_1$ , é  $(R_0, R_1, R_2, R_2)$  e o seu melhor tempo de manufatura é 15 segundos. Os resultados obtidos neste caso, apresentados na Tabela 12, mostram que o mecanismo de tomada de decisões consegue atingir a solução ótima em todos os testes planejados.

Vetor Solução $\mathbf{R}_{p_1}$	Melhor makespan estimado [s]
$(R_0, R_1, R_2, R_2)$	15
$(R_0, R_3, R_3, R_4)$	25
$(R_0, R_1, R_2, R_4)$	28
$(R_0, R_1, R_3, R_4)$	28
$(R_0, R_1, R_5, R_5)$	28

Tabela 11: Melhores soluções cenário I, produto tipo  $p_1$ ,  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ .

Número de Teste	Duração do Teste [s]	Melhor makespan [s]	Média aritmética dos Makespans [s]	Solução Atingida $\mathbf{R}_{p_1}$
1	421.6	16.0	19.7	$(R_0, R_1, R_2, R_2)$
2	421.6	15.9	18.7	$(R_0, R_1, R_2, R_2)$
3	421.6	15.8	24.3	$(R_0, R_1, R_2, R_2)$
4	421.6	15.9	19.9	$(R_0, R_1, R_2, R_2)$
5	421.5	16.0	22.1	$(R_0, R_1, R_2, R_2)$
6	421.5	15.4	21.3	$(R_0, R_1, R_2, R_2)$
7	420.9	15.4	19.7	$(R_0, R_1, R_2, R_2)$
8	420.4	15.4	19.5	$(R_0, R_1, R_2, R_2)$
9	420.4	15.3	19.6	$(R_0, R_1, R_2, R_2)$
10	420.3	15.4	21.5	$(R_0, R_1, R_2, R_2)$
Mean	421.14	15.6	20.6	

Tabela 12: Resultados cenário I, produto  $p_1$ .

A Figura 34 mostra detalhadamente os tempos de manufatura ou *Makespans* obtidos para cada um dos 100 produtos que foram fabricados no teste 1 do caso cenário I, produto  $p_1$ . Na figura, observa-se que cada produto inserido é associado a uma iteração do algoritmo de colônia de formigas implementado, além disso, observa-se o comportamento aleatório que o algoritmo possui no início da sua execução em busca de soluções ótimas, e no final da execução do teste, observa-se o comportamento estável esperado onde todos

os produtos inseridos na planta seguem a solução ótima atingida pelo algoritmo proposto. Mesmo atingindo a solução ótima de  $\mathbf{R}_{p_1}=(R_0,R_1,R_2,R_2)$  os tempos obtidos na planta não correspondem ao melhor tempo estimado de 15 s devido a que 4 não considera diretamente os tempos de processamento, comunicação nem os possíveis atrasos devidos às restrições de manufatura e transporte da planta.

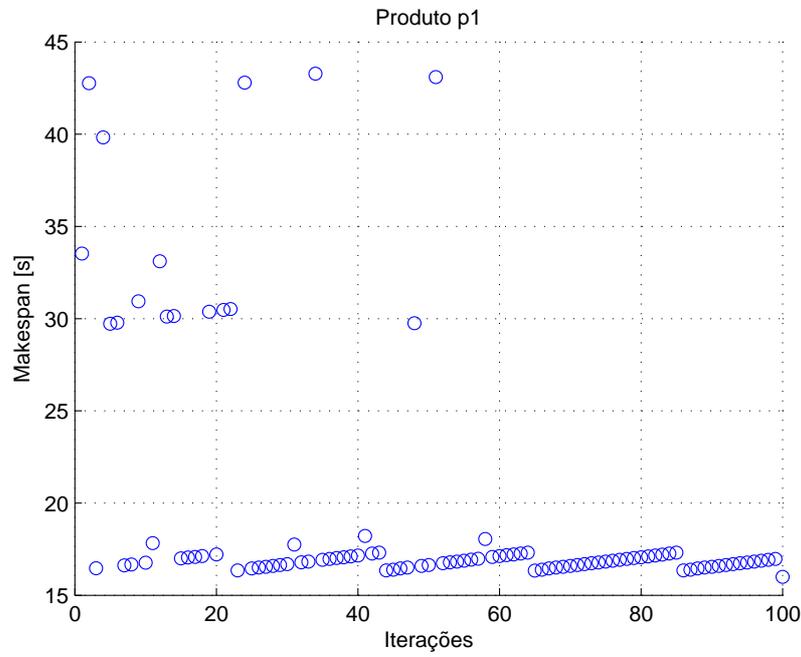


Figura 34: *Makespans* [s] obtidos no cenário I, produto  $p_1$ , teste 1 (Fonte: Próprio autor).

As cinco melhores soluções estimadas para o caso do cenário I, produto  $p_2$ , são apresentadas na Tabela 13 e os resultados correspondentes são apresentados na Tabela 14. Os resultados mostram que em todos os testes a solução atingida foi  $(R_0,R_1,R_2,R_3)$ , a qual é uma solução sub-ótima cujo tempo mínimo de manufatura estimado é de 23 segundos. As soluções ótimas neste caso são  $(R_0,R_1,R_3,R_3)$ ,  $(R_0,R_2,R_2,R_3)$  ou  $(R_0,R_2,R_3,R_3)$  todas elas possuem um tempo mínimo de manufatura estimado de 20 segundos. É importante salientar que em todos os testes o melhor tempo de manufatura obtido foi menor que 23 segundos, isso sugere que pelo menos uma das soluções ótimas foi escolhida por um produto do teste. Porém, devido a que o resultado obtido pode ser considerado muito próximo ao valor pretendido, variações de possíveis fatores de ajuste no mecanismo de tomada de decisões não foram analisadas.

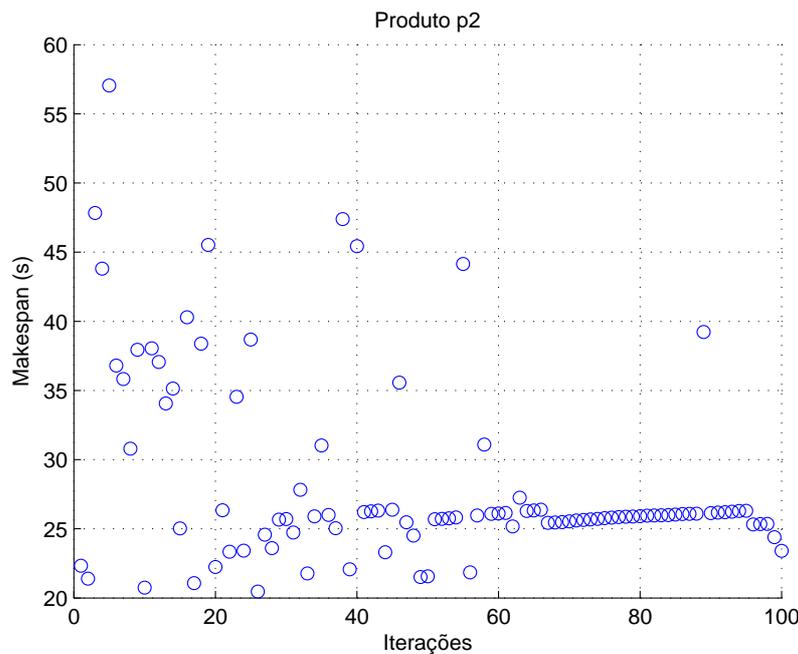
Vetor Solução $\mathbf{R}_{p_2}$	Melhor makespan estimado [s]
$(R_0,R_1,R_3,R_3)$	20
$(R_0,R_2,R_2,R_3)$	20
$(R_0,R_2,R_3,R_3)$	20
$(R_0,R_1,R_2,R_3)$	23
$(R_0,R_2,R_2,R_4)$	25

Tabela 13: Melhores soluções cenário I, produto tipo  $p_2$ ,  $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ .

Número de Teste	Duração do Teste [s]	Melhor makespan [s]	Média aritmética dos Makespans [s]	Solução Atingida $R_{p_2}$
1	429.2	20.5	28.4	$(R_0, R_1, R_2, R_3)$
2	453.3	20.4	28.1	$(R_0, R_1, R_2, R_3)$
3	428.8	21.1	27.3	$(R_0, R_1, R_2, R_3)$
4	429.3	20.6	29.2	$(R_0, R_1, R_2, R_3)$
5	429.2	21.5	29.6	$(R_0, R_1, R_2, R_3)$
6	428.9	20.5	28.6	$(R_0, R_1, R_2, R_3)$
7	428.4	20.4	28.9	$(R_0, R_1, R_2, R_3)$
8	428.4	20.4	26.5	$(R_0, R_1, R_2, R_3)$
9	428.2	20.4	26.4	$(R_0, R_1, R_2, R_3)$
10	444.2	20.6	33.1	$(R_0, R_1, R_2, R_3)$
Mean	432.8	20.6	28.6	

Tabela 14: Resultados cenário I, produto  $p_2$ .

De forma similar ao anterior caso, as Figuras 35 e 36 mostram detalhadamente os tempos de manufatura dos 100 produtos inseridos no teste 1 e no teste 2 respectivamente correspondentes ao caso do cenário I, produto tipo  $p_2$ . Observa-se, como esperado, o comportamento aleatório inicial e a forma como o algoritmo se estabiliza ao final da sua execução atingindo valores próximos aos estimados, neste caso de 23 s. Nota-se uma dificuldade maior do algoritmo de se estabilizar no teste 2 devido a que, neste teste, o algoritmo explora em várias iterações a solução ótima de  $(R_0, R_2, R_2, R_3)$ , isto pode ser observado tanto na Figura 36 como na tabela correspondente no Anexo A.2. Mesmo assim, o teste 2 finaliza com uma quantidade maior de feromônios na solução  $(R_0, R_1, R_2, R_3)$ .

Figura 35: *Makespans* [s] obtidos no cenário I, produto  $p_2$ , teste 1 (Fonte: Próprio autor).

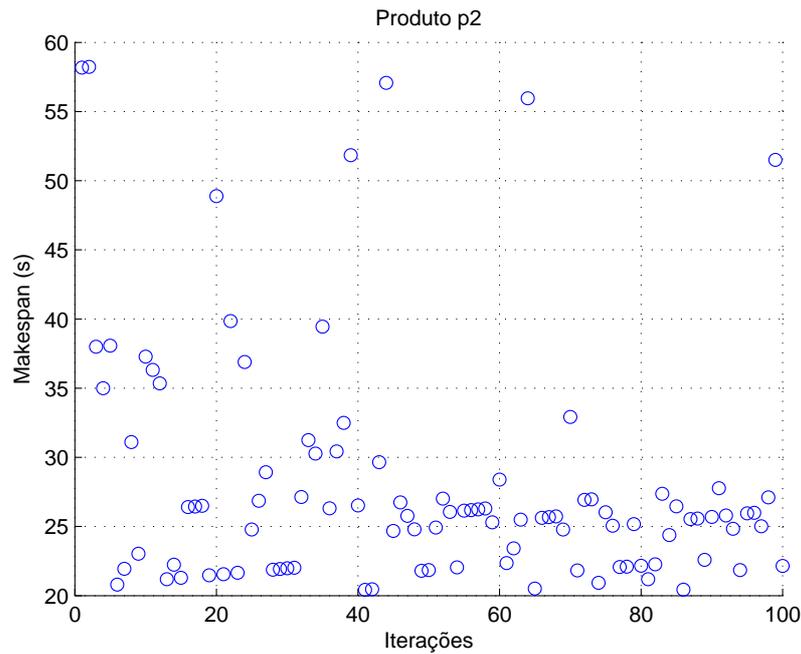


Figura 36: *Makespans* [s] obtidos no cenário I, produto  $p_2$ , teste 2 (Fonte: Próprio autor).

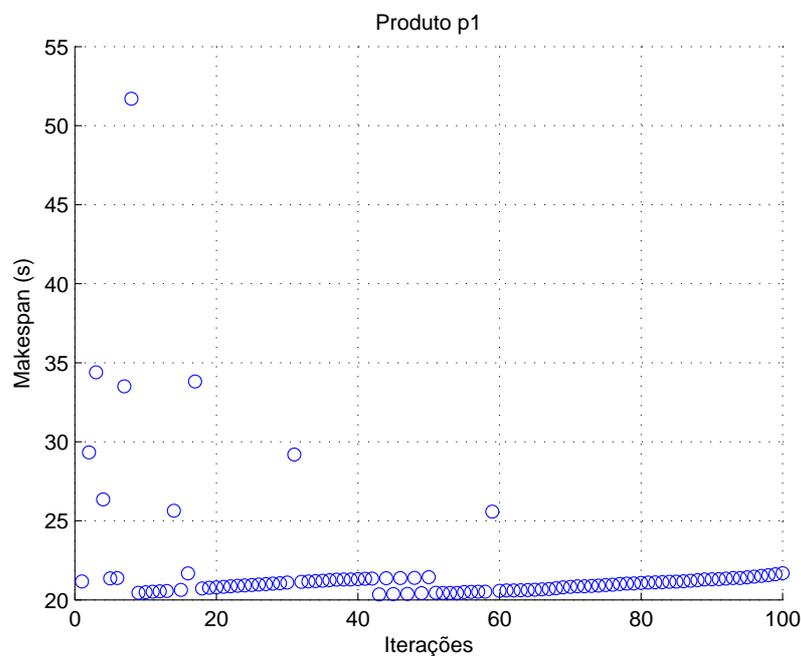
No cenário II, produto  $p_1$  as melhores soluções que a planta pode oferecer são listadas na tabela 15. Na Tabela 16 observa-se que no cenário II, produto  $p_1$ , todos os testes atingiram a solução  $(R_0, R_1, R_1, R_3)$ , essa solução é uma das várias soluções ótimas do caso, sendo o seu tempo de manufatura mínimo de 20 segundos. Uma solução sub-ótima do caso, é  $(R_0, R_1, R_2, R_3)$  cujo tempo de manufatura mínimo é 23 segundos.

A Figura 37 mostra os tempos de manufatura correspondentes ao teste 1 do caso cenário II, produto tipo  $p_1$ , os comportamentos aleatório inicial e estável final são observados na figura, assim como o fato esperado que os tempos de manufatura obtidos no teste não atingem o melhor valor estimado de 20 segundos correspondente a solução  $(R_0, R_1, R_1, R_3)$  devido aos tempos de processamento, comunicação e possíveis atrasos pelas restrições da planta.

Vetor Solução $\mathbf{R}_{p_1}$	Melhor makespan estimado [s]
$(R_0, R_1, R_1, R_3)$	20
$(R_0, R_1, R_3, R_3)$	20
$(R_0, R_2, R_2, R_3)$	20
$(R_0, R_2, R_3, R_3)$	20
$(R_0, R_1, R_2, R_3)$	23

Tabela 15: Melhores soluções cenário II, produto tipo  $p_1$ ,  $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ .

Número de Teste	Duração do Teste [s]	Melhor makespan [s]	Média aritmética dos Makespans [s]	Solução Atingida $\mathbf{R}_{p_1}$
1	424.6	20.3	22.0	$(R_0, R_1, R_1, R_3)$
2	424.9	20.3	22.1	$(R_0, R_1, R_1, R_3)$
3	424.9	20.3	22.5	$(R_0, R_1, R_1, R_3)$
4	424.8	20.4	23.6	$(R_0, R_1, R_1, R_3)$
5	424.9	20.3	23.7	$(R_0, R_1, R_1, R_3)$
6	424.6	20.0	23.9	$(R_0, R_1, R_1, R_3)$
7	424.6	20.3	22.2	$(R_0, R_1, R_1, R_3)$
8	424.8	20.2	23.1	$(R_0, R_1, R_1, R_3)$
9	425.1	20.4	23.7	$(R_0, R_1, R_1, R_3)$
10	424.8	20.4	21.8	$(R_0, R_1, R_1, R_3)$
Mean	424.8	20.3	22.9	

Tabela 16: Resultados cenário II, produto  $p_1$ .Figura 37: *Makespans* [s] obtidos no cenário II, produto  $p_1$ , teste 1 (Fonte: Próprio autor).

Finalmente, os resultados apresentados na Tabela 18 correspondentes ao cenário II, produto  $p_2$ , mostram que em todos os testes o resultado atingido pelo mecanismo de tomada de decisões é  $(R_0, R_3, R_3, R_4)$ , esse resultado é a solução ótima do caso, como pode ser conferido na tabela 17. Do mesmo modo que nos anteriores casos, a Figura 38 mostra os tempos de manufatura ou *Makespans* obtidos no teste 1 do caso. Na figura podem ser visualizados os dois comportamentos do algoritmo e a forma como ele estagna na solução ótima atingindo resultados próximos aos estimados, neste caso, de 25 segundos.

Vetor Solução $\mathbf{R}_{p_2}$	Melhor makespan estimado [s]
$(R_0, R_3, R_3, R_4)$	25
$(R_0, R_3, R_3, R_6)$	33
$(R_0, R_5, R_5, R_6)$	33
$(R_0, R_3, R_5, R_6)$	36
$(R_0, R_4, R_5, R_6)$	36

Tabela 17: Melhores soluções cenário II, produto tipo  $p_2$ ,  $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ .

Número de Teste	Duração do Teste [s]	Melhor makespan [s]	Média aritmética dos Makespans [s]	Solução Atingida $\mathbf{R}_{p_2}$
1	429.1	25.4	34.8	$(R_0, R_3, R_3, R_4)$
2	428.6	25.2	34.5	$(R_0, R_3, R_3, R_4)$
3	432.7	25.2	37.8	$(R_0, R_3, R_3, R_4)$
4	432.7	25.2	30.5	$(R_0, R_3, R_3, R_4)$
5	428.7	25.3	32.8	$(R_0, R_3, R_3, R_4)$
6	428.7	25.3	30.0	$(R_0, R_3, R_3, R_4)$
7	428.8	25.4	32.6	$(R_0, R_3, R_3, R_4)$
8	448.7	25.0	30.3	$(R_0, R_3, R_3, R_4)$
9	428.7	25.3	31.1	$(R_0, R_3, R_3, R_4)$
10	428.7	25.3	31.6	$(R_0, R_3, R_3, R_4)$
Mean	431.5	25.3	32.6	

Tabela 18: Resultados cenário II, produto  $p_2$ .

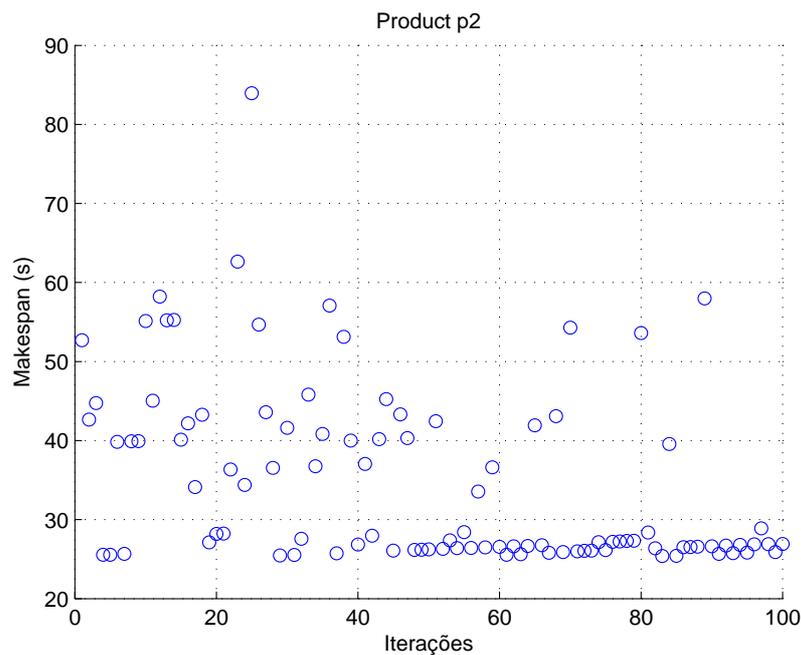


Figura 38: *Makespans* [s] obtidos no cenário II, produto  $p_2$ , teste 1 (Fonte: Próprio autor).

### 6.3.2 Análise dos resultados

Para concluir, é apresentada a análise dos resultados obtidos na planta de manufatura simulada. A Tabela 19 resume as observações em cada uma das métricas indicadas na Tabela 9.

Métricas	Análise
Manter produção com retirada de estações.	A continuidade de produção no MAS não é afetada pela retirada de estações de trabalho nele desde que exista pelo menos algum recurso disponível que ofereça o processamento das tarefas requeridas.
Produzir produtos customizados.	A planta de manufatura simulada conseguiu manufaturar com sucesso pelo menos dois tipos diferentes de produto, a saber, um produto do tipo $p_1$ cuja lista de tarefas é $\mathbf{T}_{p_1} = (T_1, T_2, T_3)$ e outro do tipo $p_2$ cuja lista de tarefas é $\mathbf{T}_{p_2} = (T_3, T_2, T_1)$ .
Obtenção de tempos de manufatura mínimos	O mecanismo de tomada de decisões baseado em ACO resolve efetivamente o problema de agendamento de tarefas em manufatura e atinge as soluções ótimas ou sub-ótimas das 64 possíveis soluções que a planta pode oferecer.

Tabela 19: Análise das métricas observadas.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Um controle descentralizado foi proposto para resolver o problema de agendamento de tarefas derivado dos sistemas de manufatura flexível (FMS). O controle integra os paradigmas de MAS e ACO a fim de resolver os problemas de flexibilidade, robustez e otimização apresentados em sistemas industriais. O desempenho do controle foi avaliado numa simulação onde uma arquitetura baseada nos agentes concebidos em (PEIXOTO, 2012) foi adotada. Finalmente, os resultados mostram que a otimização ACO logra atingir soluções ótimas ou sub-ótimas e o MAS possui a flexibilidade necessária para produzir diversos tipos de produto e manter a produção mesmo com a retirada de estações de trabalho nele.

Como trabalho futuro pretende-se a implementação de um sistema que evite a inundação convencional de informações na rede e faça uma tomada de decisões local em cada um dos nós do grafo. Desse modo, pode ser aproveitado o mecanismo de stigmergia implementado a fim de solucionar o problema de tráfego de dados. Além disso, pretende-se integrar os conceitos apresentados nas diferentes arquiteturas multi-agente estudadas. Tudo isto com a intenção de implementar um ambiente mais similar ao observado em uma planta de manufatura real.

Para implementar o MAS pretendido, devem ser identificados diferentes tipos de agentes e plataformas ou nós. Deste modo, podem ser identificados agentes estáticos que podem representar aos recursos e agentes móveis que podem representar a cada um dos holons produto, os quais, devem executar um agendamento de tarefas que precise unicamente de informações locais.

Outro assunto importante que pode ser tratado em trabalhos futuros, é a minimização do impacto dos distúrbios. Sabe-se que a implementação de um holon centralizado que detecte distúrbios e propague essa informação através do MAS, adiciona um comportamento híbrido que minimiza efetivamente o impacto dos distúrbios (BARBOSA et al., 2015). Desse modo a ACO pode-se comportar como no seu estado inicial transitório em caso de distúrbios ou como no seu estado final estacionário conforme o exigido pela planta, sem a necessidade de executar a otimização no holon centralizado.

Finalmente, os trabalhos futuros também podem ser dirigidos ao estudo de mecanismos que permitam a maximização do rendimento das estações de trabalho nos FMS, procurando desse modo, a utilização de mais de um caminho possível no processo de manufatura. O paradigma de Otimização Multimodal pode ser implementado a fim de utilizar mais de uma solução ótima do sistema (PREUSS, 2015).

## REFERÊNCIAS

- AYDIN, M. E.; FOGARTY, T. C. A simulated annealing algorithm for multi-agent systems: a job-shop scheduling application. **Journal of Intelligent Manufacturing**, Dordrecht, v.15, n.6, p.805–814, 2004.
- BARBOSA, J. et al. Dynamic self-organization in holonic multi-agent manufacturing systems: the adacor evolution. **Computers in Industry**, [S.l.], v.66, p.99–111, 2015.
- BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. **Developing multi-agent systems with JADE**. Chichester: John Wiley & Sons, 2007. v.7.
- BLESSING, L. T.; CHAKRABARTI, A. **DRM, a design research methodology**. Londres: Springer Science & Business Media, 2009.
- CAVALCANTE, A. L. D. **Arquitetura baseada em agentes e auto-organizável para a manufatura**. 2012. 138 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.
- CHIANG, L. H.; RUSSELL, E. L.; BRAATZ, R. D. **Fault detection and diagnosis in industrial systems**. Londres: Springer Science & Business Media, 2000.
- CHIRN, J.-L.; MCFARLANE, D. C. A holonic component-based approach to reconfigurable manufacturing control architecture. In: INTERNATIONAL WORKSHOP ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, 11, 2000, Londres. **Proceedings...** Piscataway NJ: IEEE, 2000. p.219–223.
- COLORNI, A. et al. Ant system for job-shop scheduling. **Belgian Journal of Operations Research, Statistics and Computer Science**, Bruxelas, v.34, n.1, p.39–53, 1994.
- CORMEN, T. H. **Introduction to algorithms**. Cambridge, MA, USA: MIT press, 2009.
- DORIGO, M. **Optimization, Learning and Natural Algorithms (in Italian)**. 1992. 140 p. Tese (Doutorado em Engenharia Elétrica) — Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.
- DORIGO, M.; BONABEAU, E.; THERAULAZ, G. Ant algorithms and stigmergy. **Future Generation Computer Systems**, Amsterdam, v.16, n.8, p.851–871, 2000.
- FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (FIPA). **Fipa acl message structure specification**. Disponível em <<http://www.fipa.org/specs/fipa00061/SC00061G.html>>. Acesso em: 17 Set.2017.

FRANKLIN, S.; GRAESSER, A. Is it an Agent, or just a Program?: a taxonomy for autonomous agents. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 1996, Budapest. **Proceedings...** Berlin: Springer, 1996. p.21–35.

FREITAS, E. P. de. **Cooperative context-aware setup and performance of surveillance missions using static and mobile wireless sensor networks**. 2011. 261 p. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

GIRET, A.; BOTTI, V. Holons and agents. **Journal of Intelligent Manufacturing**, Dordrecht, v.15, n.5, p.645–659, 2004.

JANA, T. K. et al. Dynamic schedule execution in an agent based holonic manufacturing system. **Journal of Manufacturing Systems**, Madison, Wisconsin, v.32, n.4, p.801–816, 2013.

KLOSTERMEYER, A.; KLEMM, E. PABADIS—an agent based flexible manufacturing concept. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 1, 2003, Banff, Canada. **Proceedings...** Piscataway NJ: IEEE, 2003. p.286–293.

KNAPP, E. D.; LANGILL, J. T. **Industrial Network Security**: securing critical infrastructure networks for smart grid, scada, and other industrial control systems. Waltham, MA, USA: Syngress, 2014.

LEITÃO, P. Agent-based distributed manufacturing control: a state-of-the-art survey. **Engineering Applications of Artificial Intelligence**, Auburn, Washington, USA, v.22, n.7, p.979–991, 2009.

LEITÃO, P.; RESTIVO, F. Holonic adaptive production control systems. In: IEEE ANNUAL CONFERENCE OF THE INDUSTRIAL ELECTRONICS SOCIETY, 28, 2002, Sevilla, Spain. **Proceedings...** Piscataway NJ: IEEE, 2002. v.4, p.2968–2973.

LEITÃO, P.; RESTIVO, F. ADACOR: a holonic architecture for agile and adaptive manufacturing control. **Computers in Industry**, [S.l.], v.57, n.2, p.121–130, 2006.

LIM, M.; ZHANG, Z. A multi-agent system using iterative bidding mechanism to enhance manufacturing agility. **Expert Systems with Applications**, Louisiana, USA, v.39, n.9, p.8259–8273, 2012.

MERDAN, M. et al. Knowledge-based multi-agent architecture. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR MODELLING, CONTROL AND AUTOMATION, AND INTERNATIONAL CONFERENCE ON INTELLIGENT AGENTS, WEB TECHNOLOGIES AND INTERNET COMMERCE, 2006, Sydney, Australia. **Proceedings...** [S.l.: s.n.], 2006. p.55–55.

NIKOLAI, C.; MADEY, G. Tools of the trade: a survey of various agent based modeling platforms. **Journal of Artificial Societies and Social Simulation**, [S.l.], v.12, n.2, p.2, 2009.

- PARUNAK, H. V. D. What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC. In: INTERNATIONAL WORKSHOP ON COOPERATIVE INFORMATION AGENTS, 1998, Paris, France. **Proceedings...** Berlin: Springer Berlin Heidelberg, 1998. p.1–18.
- PEIXOTO, J. A. **Desenvolvimento de Sistemas de Automação da Manufatura Usando Arquiteturas Orientadas a Serviço e Sistemas Multi-agentes**. 2012. 135 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.
- PEIXOTO, J. A. **Sistema minimamente invasivo baseado em agentes aplicado em controladores lógicos programáveis**. 2016. 224 p. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.
- PINTEA, C.-M. **Advances in bio-inspired computing for combinatorial optimization problems**. Berlin: Springer, 2014.
- PREUSS, M. **Multimodal optimization by means of evolutionary algorithms**. Berlin: Springer, 2015.
- RUSSELL, S.; NORVIG, P. **Artificial Intelligence: a modern approach**. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995. 905 p. v.25.
- THERAULAZ, G.; BONABEAU, E. A brief history of stigmergy. **Artificial life**, Portland, OR, USA, v.5, n.2, p.97–116, 1999.
- UHRMACHER, A. M.; WEYNS, D. **Multi-Agent systems: simulation and applications**. Boca Raton, Florida, USA: CRC press, 2009.
- VALCKENAERS, P. **The PROSA Reference Architecture**. Disponível em <<https://www.mech.kuleuven.be/en/pma/research/MACC/research/prosaresearch/>>. Acesso em: 17 Set.2017.
- VALCKENAERS, P. et al. Multi-agent coordination and control using stigmergy. **Computers in industry**, [S.l.], v.53, n.1, p.75–96, 2004.
- VAN BRUSSEL, H. et al. Reference architecture for holonic manufacturing systems: prosa. **Computers in industry**, [S.l.], v.37, n.3, p.255–274, 1998.
- VERSTRAETE, P. et al. Engineering manufacturing control systems using PROSA and delegate MAS. **International Journal of Agent-Oriented Software Engineering**, Olney, Bucks, UK, v.2, n.1, p.62–89, 2008.
- VLASSIS, N. A concise introduction to multiagent systems and distributed artificial intelligence. **Synthesis Lectures on Artificial Intelligence and Machine Learning**, Williston, VT, USA, v.1, n.1, p.1–71, 2007.
- VRBA, P. et al. Rockwell automation's holonic and multiagent control systems compendium. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, [S.l.], v.41, n.1, p.14–30, 2011.
- WU, J.; WU, G.; WANG, J. FLEXIBLE JOB-SHOP SCHEDULING PROBLEM BASED ON HYBRID ACO ALGORITHM. **International Journal of Simulation Modelling (IJSIMM)**, Vienna, v.16, n.3, p.497–505, 2017.

XIANG, W.; LEE, H. Ant colony intelligence in multi-agent dynamic manufacturing scheduling. **Engineering Applications of Artificial Intelligence**, Amsterdam, v.21, n.1, p.73–85, 2008.

ZHANG, S.; WONG, T. N. Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid mas/aco approach. **International Journal of Production Research**, [S.l.], v.55, n.11, p.3173–3196, 2017.

## ANEXO A CONCENTRAÇÃO DE FEROMÔNIOS AO FINAL DOS TESTES.

### A.1 Testes Cenário I, Produto $p_1$

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	58.44	-	0.07
$R_2$	-	58.13	58.35
$R_3$	0.06	0.18	-
$R_4$	0.08	-	0.16
$R_5$	-	0.27	0.29
$R_6$	0.29	0.29	-

Test 1

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	58.56	-	0.06
$R_2$	-	58.20	58.45
$R_3$	0.22	0.46	-
$R_4$	0.06	-	0.08
$R_5$	-	0.16	0.31
$R_6$	0.06	0.07	-

Test 2

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	54.72	-	0.06
$R_2$	-	51.76	53.12
$R_3$	0.20	3.28	-
$R_4$	0.23	-	1.97
$R_5$	-	0.07	0.07
$R_6$	0.07	0.12	-

Test 3

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	58.35	-	0.08
$R_2$	-	58.22	58.22
$R_3$	0.08	0.29	-
$R_4$	0.44	-	0.08
$R_5$	-	0.36	0.54
$R_6$	0.06	0.06	-

Test 4

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	56.4	-	0.36
$R_2$	-	55.22	56.02
$R_3$	0.24	1.35	-
$R_4$	0.19	-	0.19
$R_5$	-	0.25	0.37
$R_6$	0.09	0.12	-

Test 5

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	57.34	-	0.11
$R_2$	-	56.95	57.42
$R_3$	0.69	0.92	-
$R_4$	0.07	-	0.19
$R_5$	-	0.16	0.45
$R_6$	0.07	0.14	-

Test 6

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	56.84	-	0.06
$R_2$	-	55.74	56.71
$R_3$	0.07	1.05	-
$R_4$	0.08	-	0.09
$R_5$	-	0.15	0.19
$R_6$	0.06	0.11	-

Test 7

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	59.32	-	0.06
$R_2$	-	58.58	59.12
$R_3$	0.08	0.66	-
$R_4$	0.06	-	0.15
$R_5$	-	0.12	0.19
$R_6$	0.06	0.16	-

Test 8

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	59.31	-	0.06
$R_2$	-	59.03	59.13
$R_3$	0.09	0.18	-
$R_4$	0.06	-	0.14
$R_5$	-	0.19	0.23
$R_6$	0.09	0.15	-

Test 9

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	58.31	-	0.18
$R_2$	-	56.94	57.61
$R_3$	0.14	0.69	-
$R_4$	0.13	-	0.07
$R_5$	-	0.64	0.76
$R_6$	0.06	0.35	-

Test 10

## A.2 Testes Cenário I, Produto $p_2$

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.36	-	37.49
$R_2$	-	37.08	0.58
$R_3$	36.39	0.63	-
$R_4$	0.33	-	0.12
$R_5$	-	0.32	0.08
$R_6$	1.19	0.24	-

Test 1

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.13	-	23.74
$R_2$	-	36.47	13.87
$R_3$	36.39	2.30	-
$R_4$	2.43	-	1.39
$R_5$	-	0.24	0.13
$R_6$	0.17	0.12	-

Test 2

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.06	-	34.90
$R_2$	-	37.04	2.83
$R_3$	35.30	0.70	-
$R_4$	0.54	-	0.31
$R_5$	-	0.12	0.06
$R_6$	2.20	0.24	-

Test 3

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.19	-	31.01
$R_2$	-	36.31	6.72
$R_3$	37.28	0.60	-
$R_4$	0.46	-	0.10
$R_5$	-	1.27	0.45
$R_6$	0.35	0.09	-

Test 4

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.19	-	27.22
$R_2$	-	33.26	7.02
$R_3$	33.12	1.88	-
$R_4$	2.27	-	1.81
$R_5$	-	0.85	0.11
$R_6$	0.60	0.17	-

Test 5

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.17	-	36.41
$R_2$	-	34.56	1.82
$R_3$	34.41	3.54	-
$R_4$	2.65	-	0.14
$R_5$	-	0.20	0.12
$R_6$	1.26	0.18	-

Test 6

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.20	-	35.62
$R_2$	-	37.41	3.07
$R_3$	37.90	0.88	-
$R_4$	0.07	-	0.19
$R_5$	-	0.50	0.07
$R_6$	0.78	0.14	-

Test 7

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.11	-	37.08
$R_2$	-	38.45	2.19
$R_3$	38.38	0.84	-
$R_4$	0.20	-	0.11
$R_5$	-	0.07	0.06
$R_6$	0.74	0.08	-

Test 8

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.28	-	34.97
$R_2$	-	37.51	4.97
$R_3$	39.29	2.29	-
$R_4$	0.17	-	0.12
$R_5$	-	0.07	0.07
$R_6$	0.38	0.26	-

Test 9

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	2.67	-	30.55
$R_2$	-	32.59	4.41
$R_3$	30.89	0.08	-
$R_4$	0.20	-	0.17
$R_5$	-	2.73	0.36
$R_6$	1.74	0.10	-

Test 10

### A.3 Testes Cenário II, Produto $p_1$

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	68.47	68.48	-
$R_2$	2.86	2.86	-
$R_3$	-	0.06	70.87
$R_4$	0.07	-	0.35
$R_5$	-	0.06	0.12
$R_6$	0.06	-	0.11

Test 1

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	63.86	63.86	-
$R_2$	1.47	1.47	-
$R_3$	-	0.06	64.61
$R_4$	0.13	-	0.06
$R_5$	-	0.13	0.21
$R_6$	0.06	-	0.64

Test 2

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	68.02	68.06	-
$R_2$	2.56	2.56	-
$R_3$	-	0.06	67.14
$R_4$	0.06	-	1.78
$R_5$	-	0.06	1.76
$R_6$	0.10	-	0.06

Test 3

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	54.63	54.79	-
$R_2$	2.01	2.01	-
$R_3$	-	0.06	55.61
$R_4$	0.19	-	0.77
$R_5$	-	0.06	0.24
$R_6$	0.09	-	0.29

Test 4

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	83.68	83.68	-
$R_2$	3.54	3.54	-
$R_3$	-	0.06	84.81
$R_4$	0.07	-	1.50
$R_5$	-	0.06	0.63
$R_6$	0.06	-	0.39

Test 5

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	64.00	64.03	-
$R_2$	0.51	0.52	-
$R_3$	-	0.06	64.06
$R_4$	0.06	-	0.21
$R_5$	-	0.07	0.22
$R_6$	0.11	-	0.19

Test 6

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	84.25	84.25	-
$R_2$	1.03	1.04	-
$R_3$	-	0.06	85.15
$R_4$	0.09	-	0.07
$R_5$	-	0.09	0.13
$R_6$	0.07	-	0.10

Test 7

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	75.94	76.10	-
$R_2$	0.31	0.31	-
$R_3$	-	0.06	75.97
$R_4$	0.21	-	0.16
$R_5$	-	0.06	0.31
$R_6$	0.07	-	0.09

Test 8

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	70.77	70.77	-
$R_2$	0.21	0.21	-
$R_3$	-	0.06	63.57
$R_4$	0.06	-	6.28
$R_5$	-	0.06	0.91
$R_6$	0.06	-	0.32

Test 9

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	81.22	81.22	-
$R_2$	24.80	24.80	-
$R_3$	-	0.06	105.63
$R_4$	0.06	-	0.21
$R_5$	-	0.06	0.24
$R_6$	0.06	-	0.06

Test 10

#### A.4 Testes Cenário II, Produto $p_2$

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	1.29	0.06	-
$R_2$	0.72	0.15	-
$R_3$	-	51.87	51.09
$R_4$	51.53	-	1.38
$R_5$	-	2.10	1.56
$R_6$	0.63	-	0.15

Test 1

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.31	0.13	-
$R_2$	0.55	0.10	-
$R_3$	-	57.21	55.66
$R_4$	56.52	-	1.62
$R_5$	-	0.22	0.17
$R_6$	0.28	-	0.21

Test 2

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	2.14	0.10	-
$R_2$	0.13	0.06	-
$R_3$	-	44.17	43.69
$R_4$	44.69	-	1.64
$R_5$	-	5.40	4.02
$R_6$	2.78	-	0.39

Test 3

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.16	0.16	-
$R_2$	0.33	0.07	-
$R_3$	-	54.97	54.75
$R_4$	54.82	-	0.30
$R_5$	-	0.19	0.16
$R_6$	0.08	-	0.17

Test 4

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.11	0.09	-
$R_2$	0.26	0.06	-
$R_3$	-	72.87	71.89
$R_4$	71.66	-	0.49
$R_5$	-	1.12	0.82
$R_6$	2.11	-	0.94

Test 5

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.25	0.08	-
$R_2$	0.30	0.07	-
$R_3$	-	61.91	61.85
$R_4$	61.14	-	0.17
$R_5$	-	0.43	0.40
$R_6$	0.80	-	0.07

Test 6

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.80	0.10	-
$R_2$	0.24	0.09	-
$R_3$	-	60.55	59.78
$R_4$	58.90	-	0.78
$R_5$	-	0.07	0.07
$R_6$	0.87	-	0.18

Test 7

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.11	0.08	-
$R_2$	0.08	0.08	-
$R_3$	-	35.01	34.74
$R_4$	36.43	-	0.30
$R_5$	-	2.24	2.23
$R_6$	0.79	-	0.14

Test 8

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.15	0.10	-
$R_2$	0.14	0.12	-
$R_3$	-	50.37	50.14
$R_4$	49.57	-	0.15
$R_5$	-	0.28	0.22
$R_6$	1.01	-	0.35

Test 9

$R_i \backslash Tasks$	$T_1$	$T_2$	$T_3$
$R_0$	-	-	-
$R_1$	0.20	0.08	-
$R_2$	0.32	0.15	-
$R_3$	-	66.87	66.72
$R_4$	66.29	-	0.69
$R_5$	-	0.74	0.37
$R_6$	1.03	-	0.07

Test 10