

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

**O Problema da Alocação de Tarefas sob o Aspecto
da Escolha Social: uma Análise Arroviana de
Sistemas Multirrobo**

Wallace Pereira Neves dos Reis

Itajubá, Dezembro de 2017

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Wallace Pereira Neves dos Reis

O Problema da Alocação de Tarefas sob o Aspecto da Escolha Social: uma Análise Arroviana de Sistemas Multirrobo

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Área de Atuação: Automação e Sistemas Elétricos Industriais.

Orientador: Professor Dr. Guilherme Sousa Bastos

Dezembro de 2017

Itajubá

Wallace Pereira Neves dos Reis

O Problema da Alocação de Tarefas sob o Aspecto da Escolha Social: uma Análise Arroviana de Sistemas Multirrobo/ Wallace Pereira Neves dos Reis. – Itajubá, Dezembro de 2017-

154 p. : il. (algumas color.) ; 30 cm.

Orientador: Professor Dr. Guilherme Sousa Bastos

Dissertação (Mestrado) – UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA, Dezembro de 2017.

1. Alocação de Tarefas. 2. Escolha Social. I. Professor Dr. Guilherme Sousa Bastos. II. Universidade Federal de Itajubá. III. Faculdade de Engenharia Elétrica. IV. O Problema da Alocação de Tarefas sob o Aspecto da Escolha Social: uma Análise Arroviana de Sistemas Multirrobo

CDU 02:141:005.7

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Wallace Pereira Neves dos Reis

O Problema da Alocação de Tarefas sob o Aspecto da Escolha Social: uma Análise Arroviana de Sistemas Multirrobo

Dissertação aprovada por banca examinadora em 14 de dezembro de 2017 conferindo ao autor o título de Mestre em Ciências em Engenharia Elétrica.

Professor Dr. Guilherme Sousa Bastos
Orientador - UNIFEI

Professor Dr. Edson Prestes e Silva Júnior
UFRGS

Professor Dr. Edmílson Marmo Moreira
UNIFEI

Itajubá
Dezembro de 2017

A Giselle, minha vida, e Gláucia, minha luz.

Agradecimentos

A Deus pelas oportunidades, por ter me guiado e me ter possibilitado chegar até aqui.

Aos meus amigos do IFRJ campus Volta Redonda que acompanharam essa jornada, principalmente aos meus, hoje colegas, que já foram meus professores.

Aos meus amigos irmãos que me acompanham desde muito tempo e, mesmo de longe, me deram força e incentivo. Agradeço à nossa amizade de longa data.

Aos colegas do Laboratório de Robótica da UNIFEI, pela ajuda e por todas as conversas sobre esse universo que é a robótica. Foram muito importantes.

Ao meu orientador e amigo, Guilherme Bastos, pela paciência e dedicação, por ter aceitado essa orientação e ter me guiado e me cobrado ao longo desses anos.

Aos membros da banca, pelos apontamentos, correções e sugestões.

A minha família, sem eles eu não estaria aqui.

Em especial, agradeço minha noiva, Giselle, pelo companheirismo de todos esses anos, pelo porto seguro, pelo incentivo e por não me deixar desanimar. Eu te amo!

A natureza colocou a humanidade sob o domínio de dois mestres soberanos, a dor e o prazer. Só eles podem mostrar o que devemos fazer, bem como determinar o que faremos... Eles nos governam em tudo o que fazemos, em tudo o que dizemos, em tudo o que pensamos...

Jeremy Bentham, 1780

Resumo

A pesquisa em sistemas multirrobo é uma área fértil, que tem recebido muita atenção nas últimas décadas. Com os avanços de *hardware* e *software*, como o desenvolvimento do *Robot Operating System* - ROS, a aplicação real desses sistemas tem se tornado cada vez mais factível. Porém, novos desafios emergem, principalmente, a coordenação dos robôs e a alocação de tarefas para o correto cumprimento de uma missão. Apesar das muitas propostas apresentadas na literatura, a alocação de tarefas ainda não é um assunto esgotado, tanto em aplicações quanto em teorias. Assim, este trabalho propõe uma visão do problema de alocação de tarefas em sistemas multirrobo baseada na Teoria da Escolha Social, mais especificamente, na estrutura proposta por Kenneth J. Arrow em seu famoso Teorema da Impossibilidade. As condições impostas por Arrow visam criar um mecanismo agregador de preferências padrão por meio da análise axiomática. Essa análise é transportada para o domínio multirrobo. Classes de problemas com robôs mono-tarefa e algumas arquiteturas são analisadas desse ponto de vista. Além disso, é discutida a comparação de utilidades cardinais entre robôs e as vantagens da utilização de preferências ordinais. Também são propostos e analisados algoritmos para a agregação de preferências ordinais.

Palavras-chaves: Sistemas Multirrobo. Alocação de Tarefas. Escolha Social. Teorema de Arrow. Agregação de preferências.

Abstract

Research in multi-robot systems is a fertile area, which has received much attention in recent decades. With hardware and software advances, such as the development of the Robot Operating System - ROS, the actual application of these systems has become increasingly feasible. However, new challenges emerge. Mainly, the robot coordination and tasks allocation to achieve a correct mission accomplishment. Despite many proposals presented in the literature, the task allocation is not yet an exhausted matter, both in applications as in theories. Thus, this dissertation proposes a Social Choice Theory based vision of the task allocation problem in multi-robot systems, more specifically, based on the structure proposed by Kenneth J. Arrow in his famous Impossibility Theorem. The conditions imposed by Arrow aim to create a standard preference aggregation mechanism through axiomatic analysis. This analysis is transported to the multi-robot domain. Mono-task robots problems and some architectures are analyzed from this point of view. In addition, we discuss the comparison of cardinal utilities between robots and advantages of using ordinal preferences. Algorithms for ordinal preferences aggregation are also proposed and analyzed.

Key-words: Multi-robot System. Task Allocation. Social Choice. Arrow's Theorem. Preference Aggregation.

Lista de ilustrações

Figura 2.1 –Representação gráfica para exemplificação de ordenamentos de preferência de pico único e de pico não único para cinco alternativas.	35
Figura 3.1 –Estrutura em eixos da taxonomia de coordenação de sistemas multirrobo. . .	46
Figura 3.2 –Divisão em três eixos da taxonomia de arquiteturas de alocação de tarefas. .	52
Figura 3.3 –Classificação de tarefas e dependências entre tarefas na <i>iTax</i>	53
Figura 3.4 –Estrutura típica de camadas de uma arquitetura baseada em comportamento.	59
Figura 3.5 –Arquitetura ALLIANCE.	60
Figura 3.6 –Diagramas de estado do protocolo M+ para a alocação de tarefas.	72
Figura 5.1 –Níveis de motivação interna dos robôs durante avaliação da arquitetura ALLIANCE em experimento de localização e contenção de lixo tóxico.	121
Figura 5.2 –Níveis de motivação interna dos robôs durante a mesma missão representada na Figura 5.1, porém, com uma falha do robô <i>GREEN</i> na localização do lixo.	123
Figura 5.3 –Níveis de motivação interna dos robôs durante a mesma missão representada na Figura 5.1, porém, com uma falha total no robô <i>GOLD</i>	124
Figura 5.4 –Níveis de motivação interna dos robôs durante a simulação apresentada por Reis e Bastos (2016).	125
Figura 5.5 –Níveis de motivação interna dos robôs durante uma simulação apresentada por Reis e Bastos (2016) em que há uma falha parcial em um robô.	126
Figura A.1 –Os quatro perfis de preferências usados na prova do Teorema de Arrow. . . .	146

Lista de tabelas

Tabela 1 – Contagem de Borda para três alternativas em uma situação de escolha cíclica.	26
Tabela 2 – Perfil de ordenamentos de preferências individuais.	37
Tabela 3 – Perfil de ordenamentos de preferências individuais ligeiramente modificado.	38
Tabela 4 – Classificação das arquiteturas de acordo com a taxonomia de sistemas multirrobo.	85
Tabela 5 – Classificação das arquiteturas de acordo com a taxonomia para problemas de alocação de tarefas em sistemas multirrobo.	86
Tabela 6 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte I . . .	139
Tabela 7 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte II . . .	140
Tabela 8 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte III . .	141

Lista de abreviaturas e siglas

FBS	Função de bem-estar social
FBS-MR \mathcal{T}	Função de bem-estar social multirrobô sobre o conjunto \mathcal{T}
FBS-M $\mathcal{R}\mathcal{T}$	Função de bem-estar social multirrobô sobre os conjuntos \mathcal{R} e \mathcal{T}
FES	Função ou regra de escolha social
FES-MR \mathcal{T}	Função de escolha social multirrobô sobre o conjunto \mathcal{T}
FES-M $\mathcal{R}\mathcal{T}$	Função de escolha social multirrobô sobre os conjuntos \mathcal{R} e \mathcal{T}
IA	Tarefa de alocação Instantânea (<i>Instantaneous Assignment</i>)
IIA	<i>Independence of Irrelevant Alternatives</i> - Independência a Alternativas Irrelevantes
MARA	<i>Multi Agent Resource Allocation</i> - Alocação de recursos em SMA
MRTA	<i>Multi-Robot Task Allocation</i> - Alocação de tarefas em SMR
MU	<i>Marginal Utility</i> - Utilidade Marginal
MR	Tarefa de múltiplos robôs (<i>Multi robot task - MR</i>)
MT	Robô de tarefas múltiplas (<i>Multi task robot</i>)
ND	<i>Nondictatorship</i> - Não-ditadura
SMA	Sistema Multiagente
SMR	Sistema Multirrobô
SO	<i>Social Ordering</i> - Ordenamento social
SR	Tarefa de único robô (<i>Single robot task</i>)
ST	Robô de tarefa única (<i>Single task robot</i>)
OAP	<i>Optimal Assignment Problem</i>
TA	Tarefa de alocação com tempo prolongado (<i>Time-extended Assignment</i>)
UD	<i>Unrestricted Domain</i> - Domínio Irrestrito
WP	<i>Weak Pareto</i> - Princípio de Pareto ou Unanimidade

Lista de símbolos

X	Conjunto de alternativas
x_i	Alternativa i do conjunto de alternativas X
\mathcal{U}	Conjunto de funções de utilidade contínuas sobre X
u	Função de utilidade individual
p ou q	Probabilidades
\succeq	Preferência fraca, quando $x_1 \succeq x_2$ / Ordenamento social
\sim	Indiferença
\succ	Preferência estrita
\mathcal{L}	Conjunto de loterias simples sobre o conjunto de alternativas X
$P_\ell(x_i)$	Probabilidade da alternativa x_i ser selecionada pela loteria ℓ
α	Peso, definido entre $[0, 1]$, de uma alternativa x_n
\succeq_i	Relação fraca ou não-estrita de preferências do indivíduo j / Ordenamento fraco ou não-estrito de preferências do indivíduo j
\succ_i	Relação estrita de preferências do indivíduo j / Ordenamento estrito de preferências do indivíduo j
N	Conjunto de indivíduos ou agentes membros da sociedade
P	Conjunto de relações de preferências possíveis
R	Conjunto de relações de preferências racionais, ou ordenamentos de preferências, possíveis em P
$[\succeq]$	Conjunto de ordenamentos de preferências dos membros da sociedade: Perfil de Preferências
C	Função de escolha social
$C(R)$	Função de escolha social sobre o conjunto R
W	Função de bem-estar social
\succ_W	Preferência Social
S	Ordenamento forte de preferências
$B(x_1, x_2, x_3)$	Intermediação das preferências entre as alternativas x_1, x_2 e x_3

O	Dotação social de bens, ou objetos, indivisíveis disponíveis para alocação à uma sociedade
\mathcal{R}	Conjunto de m robôs
r_m	Robô m
\mathcal{T}	Conjunto de n tarefas
τ_n	Tarefa n
$Q_{r_m\tau_n}$	Qualidade com que o robô r_m executa a tarefa τ_n
$C_{r_m\tau_n}$	Custo para o robô r_m executar a tarefa τ_n
$U_{r_m\tau_n}$	Utilidade do robô r_m para a execução da tarefa τ_n
R	ALLIANCE: conjunto de n robôs
T	ALLIANCE: conjunto de m tarefas
A_i	ALLIANCE: conjuntos de comportamentos ou funções de realização de tarefas de alto nível do robô i
a_{i1}	ALLIANCE: comportamento de alto nível 1 do robô i
$h_i(a_{ik})$	ALLIANCE: função de ativação que retorna a tarefa em T a qual o robô r_i está executando quando o conjunto de comportamentos a_{ik} é ativado
θ	ALLIANCE: limiar de ativação do conjunto de comportamentos
ρ_i	ALLIANCE: taxa com que o robô r_i envia sua atividade corrente
τ_i	ALLIANCE: máximo período de tempo permitido por r_i à um companheiro de equipe deixar de enviar sua atividade
$\phi_{ij}(k, t)$	ALLIANCE: período de tempo o qual o robô r_i está disposto a permitir que as mensagens do robô r_k influenciem seu cálculo de motivação
$\delta_{slow_{ij}}(t)$	ALLIANCE: taxa de variação lenta da impaciência do robô i para o conjunto de comportamentos j
$\delta_{fast_{ij}}(t)$	ALLIANCE: taxa de variação rápida da impaciência do robô i para o conjunto de comportamentos j
$\psi_{ij}(t)$	ALLIANCE: período de tempo em que o robô r_i mantém o comportamento a_{ij} ativo, antes de permitir que o robô r_k o ative
$\lambda_{ij}(t)$	ALLIANCE: período de tempo em que r_i manterá o comportamento a_{ij} ativo, antes de desistir deste e partir para outro comportamento
m_{ij}	ALLIANCE: motivação do comportamento a_{ij}
$q(a_{ij})$	ALLIANCE: qualidade da ação a_{ij}

U_i	ALLIANCE: conjunto de tarefas que o robô r_i irá executar durante a missão corrente
T_k	M+: tarefa k
R_i	M+: robô i
s	M+: quantidade de tempo a qual um estado é mantido <i>verdadeiro</i>
$\mu(i)$	ASYMTRE: utilidade do robô R_i para j -ésimo sistema sensorial ou computacional que o mesmo necessita utilizar
P_j	ASYMTRE: probabilidade de sucesso para o j -ésimo sistema sensorial ou computacional utilizado
C_j	ASYMTRE: custo de sensoriamento de dado sensor para o j -ésimo sistema sensorial ou computacional utilizado
w	ASYMTRE: peso de combinação entre a probabilidade P_j e o custo C_j
$\succeq_i^{\tau_k}$	Preferência do robô r_i quanto a execução da tarefa τ_k
\succeq_{r_i}	Ordenamento de preferências não estritas do robo r_i sobre as tarefas do conjunto \mathcal{T}
$\succeq_{r_i}^{\tau_k}$	Ordenamento de preferências do robô r_i sobre o conjunto de robôs \mathcal{R} para a execução de determinada tarefa τ_k
$[\succeq]_{\mathcal{R}}$	Perfil de preferências: conjunto dos ordenamentos de preferências individuais dos robôs em relação à execução das tarefas
$[\succeq]_{\mathcal{R}}^{\mathcal{T}}$	Perfil de preferências: conjunto de ordenamentos de preferências individuais dos robôs sobre os robôs do conjunto \mathcal{R} para a execução das tarefas do conjunto \mathcal{T}
\mathcal{B}	Subconjunto do conjunto de tarefas \mathcal{T}
\mathcal{B}_{r_i}	Subconjunto de tarefas do conjunto de tarefas \mathcal{T} alocadas ao robô r_i
$\Pi^{\mathcal{R}}$	Conjunto de permutações possíveis sobre elementos do conjunto \mathcal{R}
π	Permutação ou política, elemento do conjunto $\Pi^{\mathcal{R}}$
$RSD()$	Mecanismo ditadura aleatória em série
$RSDTA()$	Mecanismo ditadura aleatória em série para alocação de tarefas escalonadas no tempo
p_{τ_k}	Número de postos, robôs, necessários para a execução de uma tarefa em um problema MR
\mathcal{P}	Conjunto de postos das tarefas do conjunto \mathcal{T}
$2^{\mathcal{R}}$	Conjunto potência do conjunto \mathcal{R} de robôs

Sumário

1	Introdução	17
1.1	Motivação	17
1.2	Objetivos	18
1.3	Contribuições	18
1.4	Estrutura do Trabalho	19
2	Utilidade, Teoria da Escolha Social e o Teorema da Impossibilidade de Arrow	20
2.1	Teoria da Utilidade: um breve histórico	20
2.1.1	Conceitos fundamentais e Axiomatização	22
2.2	Teoria da Escolha Social	25
2.2.1	Modelo Formal	27
2.3	O Teorema da Impossibilidade de Arrow	28
2.4	Relaxando as condições de Arrow	34
2.4.1	Relaxando o domínio irrestrito: preferências de pico único	34
2.4.2	Relaxando a condição de Pareto	35
2.4.3	Relaxando a Independência de Alternativas Irrelevantes	36
2.5	O Teorema de Gibbard-Satterthwaite	38
2.6	Aplicações do Teorema da Impossibilidade de Arrow	39
2.7	Escolha Social Computacional	42
2.7.1	O problema da alocação de objetos	43
3	Sistemas Multirrobo	45
3.1	Taxonomia para Sistemas Multirrobo	45
3.2	Taxonomia de Arquiteturas de Alocação de Tarefas em SMR	51
3.3	O Problema de Alocação de Tarefas em SMR	54
3.4	Arquiteturas de Alocação de Tarefas em SMR	56
3.4.1	Arquiteturas Baseadas em Comportamento	56
3.4.1.1	ALLIANCE	56
3.4.1.1.1	O problema de eficiência do ALLIANCE	63
3.4.2	Arquiteturas Baseadas em Mercado	65
3.4.2.1	MURDOCH	66
3.4.2.2	Protocolo M+	70
3.4.2.2.1	M+CTA	73
3.4.3	Outras Arquiteturas Estudadas	75

3.4.4	Comparação	79
3.5	Discussão	83
4	A visão arroviana do problema MRTA	87
4.1	Bem-estar social em SMR	87
4.2	Modelo Formal da Escolha Social para problemas MRTA	88
4.2.1	Funções de escolha e bem-estar social	90
4.2.2	As condições arrovianas para o problema MRTA	92
4.2.3	Preferências de longo e curto prazo	96
4.3	Análise arroviana da Taxonomia de problemas MRTA	98
4.3.1	ST-SR-IA	99
4.3.2	ST-SR-TA	105
4.3.3	ST-MR-IA	108
4.3.4	ST-MR-TA	113
4.3.5	O problema MT: <i>multi task robots</i>	114
4.4	Discussão	116
5	Análise arroviana das arquiteturas de MRTA	118
5.1	ALLIANCE	118
5.1.1	O Problema de Eficiência do ALLIANCE sob a visão arroviana	131
5.2	MURDOCH	133
5.3	M+CTA	135
6	Conclusão	142
6.1	Considerações Finais	142
6.2	Trabalhos Futuros	143
	APÊNDICE A Prova do Teorema da Impossibilidade de Arrow	144
	Referências	147

1 Introdução

1.1 Motivação

Nas últimas décadas, sistemas multirrobô (SMR) têm sido extensivamente estudados e aplicados nas mais diferentes áreas. Essas aplicações envolvem, por exemplo, o uso para cuidado de pacientes idosos em unidades de saúde, segurança patrimonial, limpeza, resgate de vítimas de desastres, transportes, entre outras. Nesses sistemas, um problema crítico é a correta alocação de tarefas para que o objetivo final seja atingido com sucesso a partir da (possível) cooperação entre os robôs envolvidos.

Segundo [Gerkey e Mataric \(2004\)](#), a complexidade dos sistemas estudados tem aumentado, principalmente, devido a dois fatores: (a) grupos de robôs cada vez maiores e (b) maior heterogeneidade entre os robôs e entre as tarefas a serem executadas. Não obstante, muitas razões explicam o interesse nesse tipo de sistema, por exemplo, a redução da complexidade e do custo de desenvolvimento, a eficiência, segundo [Cao et al. \(1995\)](#), e, no futuro, a cooperação entre humanos e robôs em um mesmo ambiente, o qual é um objetivo a longo prazo, ainda segundo [Gerkey e Mataric \(2004\)](#).

Adicionalmente, [Cao et al. \(1995\)](#) enfatizam que também pode-se derivar, dos experimentos com SMR, compreensão envolvendo ciências sociais (Teoria da organização e econômica), ciências biológicas (modelos biológicos, etologia animal) e ciência cognitiva (psicologia, aprendizagem, inteligência artificial).

O trabalho em equipe envolvendo um SMR é complexo, consistindo na divisão de tarefas, na alocação das tarefas, na coordenação e na comunicação entre os robôs ([CHOUDHURY, 2009](#)). Um ponto sensível de um sistema como este é a correta alocação de tarefas, de modo que a missão seja executada com determinado grau de eficiência.

A literatura apresenta diversas abordagens para tratar do problema de alocação de tarefas em SMR (*Multi-robot Task Allocation - MRTA*). As arquiteturas para MRTA englobadas neste estudo incluem abordagens baseadas em comportamentos ([PARKER, 1996](#)) e baseadas em mercado ou na teoria econômica ([GERKEY; MATARIĆ, 2002](#); [BOTELHO; ALAMI, 1999](#)).

A aplicação da teoria da escolha social em sistemas multiagente se faz muito clara, pois, segundo [Brandt et al. \(2016\)](#), se um sistema multiagente for visto como uma sociedade de agentes de *software* autônomos, com suas particularidades e capacidades, então, deve-se definir um mecanismo claro para agregar suas respectivas preferências a fim de se tomar uma decisão coletiva. A principal diferença entre sistemas multiagente e multirrobô é a característica física do segundo. Complementarmente, de acordo com [Stirling e Nokleby \(2009\)](#), duas abordagens

são dominantes em relação à teoria da decisão multiagente: a teoria dos jogos não-cooperativos e a teoria da escolha social.

Segundo [Arrow, Sen e Suzumura \(2011\)](#), o processo de escolha social é equivalente a um sistema composto de vários elementos, incluindo o membro da sociedade, suas interações e os resultados dessas interações. Os autores ainda afirmam que “como um sistema adaptativo, as interações deste podem ser resumidas como comunicações, expressões das capacidades específicas dos membros da sociedade” ([ARROW; SEN; SUZUMURA, 2011](#)). Em um SMR, os robôs devem agir de modo a refletir suas propriedades e capacidades. E assim como [Arrow, Sen e Suzumura \(2011\)](#) associaram a Escolha Social com um ecossistema e com um sistema adaptativo, é possível associá-la à um SMR.

1.2 Objetivos

O objetivo geral deste trabalho é fazer um paralelo entre a Teoria da Escolha Social e o problema de alocação de tarefas em sistemas multirrobô. Mais especificamente, o trabalho visa abordar a estrutura proposta no Teorema da Impossibilidade de Arrow, e, por afinidade, no Teorema da Impossibilidade de Gibbard-Satterthwaite, a fim de propor uma análise de mecanismos de decisão social e de arquiteturas de alocação de tarefas propostas na literatura. Para isso, são discutidos os axiomas e as condições necessários para a existência de funções de escolha e de bem-estar social, com o objetivo de maximizar o bem-estar social.

Além disso, um objetivo específico é estabelecer uma estrutura ordinal para comparação de preferências individuais, ao invés da comparação da utilidade, normalmente, um valor escalar calculado por cada robô. A axiomatização proposta por [Arrow \(1951\)](#) não é uma estrutura de alocação de tarefas, mas sim um “padrão” de verificação para saber quando uma determinada estrutura atende aos requisitos básicos de uma escolha social democrática e que maximiza o bem-estar coletivo.

1.3 Contribuições

É proposta a comparação ordinal de preferências sobre o conjunto de alternativas para a tomada de decisão social em uma sociedade de robôs. Essa tomada de decisão reflete a alocação de tarefas. Essa consideração baseia-se em duas justificativas complementares. Primeiramente, na dificuldade reconhecida e explorada por [Arrow \(1951\)](#) na definição de um valor (utilidade) para determinada alternativa e a posterior comparação entre esses valores por indivíduos de uma sociedade. De forma complementar, na dificuldade exposta por [Gerkey e Matarić \(2004\)](#) na medição da utilidade em sistemas robóticos, sendo afetada por ruídos de medição, métodos de computação e métricas utilizadas.

São propostos mecanismos sociais para a alocação de tarefas que atendem, mesmo que de forma relaxada, as condições arrowianas. Essa proposição se restringe aos problemas em que

os robôs executam apenas uma tarefa por vez. Os algoritmos genéricos de cada mecanismo também são apresentados, relacionados com outros algoritmos propostos por [Gerkey \(2003\)](#), [Parker \(1996\)](#) e [Botelho e Alami \(2000\)](#), principalmente, para a resolução do problema MRTA.

1.4 Estrutura do Trabalho

Esta dissertação está estruturada da seguinte forma: o capítulo 2 é destinado à revisão da Teoria da Utilidade, apresentando um histórico sobre o assunto, e à apresentação do domínio arroviano necessário às análises posteriores, ou seja, apresentar o formalismo, as condições, as funções de escolha e de bem-estar social e os teoremas, que serão transportados para o domínio do problema de alocação de tarefas em sistemas multirrobô. Além disso, o capítulo apresenta algumas aplicações do Teorema da Impossibilidade em diferentes áreas do conhecimento e introduz o conceito de Escolha Social Computacional. O capítulo 3 faz uma revisão do conceito de Sistema Multirrobô e da taxonomia proposta por [Iocchi, Nardi e Salerno \(2000\)](#) e [Farinelli, Iocchi e Nardi \(2004\)](#) para esses sistemas, apresenta a taxonomia proposta por [Gerkey e Matarić \(2004\)](#) para o problema de alocação de tarefas e define o referido problema. Ainda nesse capítulo, são apresentadas e classificadas arquiteturas de alocação de tarefas em SMR, fazendo-se, ao final, uma breve comparação entre as três arquiteturas estudadas mais a fundo neste trabalho: ALLIANCE, de [Parker \(1994\)](#), MURDOCH, de [Gerkey e Matarić \(2002\)](#) e M+, de [Botelho e Alami \(1999\)](#). O capítulo 4 define o domínio arroviano para o problema MRTA. Neste capítulo, são estabelecidas as condições para existência das funções de escolha e de bem-estar social no domínio de SMR. Será realizada uma análise arroviana da taxonomia de problemas MRTA, a partir da taxonomia de [Gerkey e Matarić \(2004\)](#), para as classes de problemas em que o robô executa apenas uma tarefa por vez. O capítulo 5 destina-se à análise arroviana das arquiteturas ALLIANCE, MURDOCH e M+. Por fim, o capítulo 6 discute as considerações finais e as propostas para trabalhos futuros.

2 Utilidade, Teoria da Escolha Social e o Teorema da Impossibilidade de Arrow

Essa dissertação se baseia em conceitos da Teoria Econômica, particularmente das áreas de estudo da Teoria da Utilidade e da Teoria da Escolha Social, com foco no movimento *New Welfare Economics* e sua *função de bem-estar social*. Essa linha de pensamento foi inaugurada por Kenneth J. Arrow, em 1951. Segundo [Beltrame \(2009\)](#), “em linhas gerais se preocupa com as análises sobre os processos de decisão coletiva e sobre como nesses processos devem ser consideradas as opiniões individuais”. Ainda sobre o trabalho de Arrow, que resultou em seu famoso Teorema da Impossibilidade, [Hansson \(2005\)](#) afirma que este foi o começo da Teoria Moderna de Decisão Social. Este capítulo discute fundamentos e conceitos que serão aplicados às análises de arquiteturas de alocação de tarefas em sistemas multirrobo.

2.1 Teoria da Utilidade: um breve histórico

A Teoria da Utilidade como escola do pensamento econômico, iniciada por Jeremy Bentham na segunda metade do século XVIII, propôs que a felicidade de um indivíduo poderia ser medida e, além disso, maximizada. A teoria, ainda, define a felicidade como meio de bem-estar individual ([BELTRAME, 2009](#)).

A proposta de Bentham se opõe ao pensamento liberalista de Adam Smith, que afirmava que o aspecto social e econômico da riqueza era reflexo da livre iniciativa individual, carregada de um sentimento egoísta de felicidade, mas que, guiada por uma *mão invisível*, essa iniciativa individual e egoísta levava a resultados coletivos e sociais jamais imaginados pelos membros da sociedade ([BELTRAME, 2009](#); [HUNT; LAUTZENHEISER, 2005](#)).

Segundo [Hunt e Lautzenheiser \(2005\)](#), Bentham definiu *utilidade* como a propriedade de qualquer objeto que tenda a produzir algum bem, vantagem ou felicidade ou a impedir algum mal, danos ou infelicidade à parte cujo interesse esteja sendo considerado. Com isso, Bentham “achava que tinha encontrado a chave da elaboração de uma ciência do bem-estar ou da felicidade humana que pudesse ser expressa matematicamente” ([HUNT; LAUTZENHEISER, 2005](#)).

Dessa forma, as escolhas dos indivíduos passaram a ser vistas como decisões calculadas, racionais, ponderando todos os benefícios, prazeres, a serem obtidos com determinada ação, deduzindo todos os custos, dor, causados por esta e, assim, escolhendo racionalmente a ação que maximizasse a felicidade sobre a dor ([HUNT; LAUTZENHEISER, 2005](#); [WAGNER, 2009](#)).

Três publicações no início da década de 1870 revolucionaram o pensamento econômico, sendo considerado um período divisor de águas: as obras de William Stanley Jevons, Carl Menger e Léon Walras. A maior contribuição desses pensadores foi a introdução do rigor lógico e matemático no conceito que foi chamado de Teoria da Utilidade Marginal. “O marginalismo permitiu que a visão utilitarista da natureza humana, que era considerada somente uma maximização racional e calculista da utilidade, fosse formulada em termos de cálculo diferencial” (HUNT; LAUTZENHEISER, 2005).

Entretanto, muitas críticas e contestações foram feitas à Teoria da Utilidade. As duas principais críticas eram quanto à mensuração da utilidade e à suposição hedonista da teoria (HUNT; LAUTZENHEISER, 2005; CUSINATO, 2003). Segundo Cusinato (2003), em relação à primeira crítica, não estava claro como se media a utilidade ou como ela poderia ser medida. De que forma se mede prazer ou felicidade?

Quanto ao hedonismo inerente à teoria, esta supõe que os indivíduos são maximizadores de prazer ou felicidade, sendo os melhores juízes de seus próprios atos (CUSINATO, 2003). Contudo, antes de tomada a decisão, não há qualquer certeza de que essa seja mesmo a melhor, já que os resultados não são conhecidos antecipadamente.

Até esse momento, pensava-se em utilidade como uma quantidade, ou seja, uma medida que expressa o quão bem aquele objeto faz ao indivíduo. Todavia, o ponto fraco da teoria era, justamente, o procedimento de medição dessa quantidade, que deve ser feito pelo próprio indivíduo. Contribuindo para essa discussão, além de muitas contribuições no ferramental matemático, Vilfredo Pareto constatou nas análises de curvas de indiferença que as relações entre dois bens que mantinham o indivíduo no mesmo nível de bem-estar, quando quantificadas por uma função de utilidade, mantinham um ordenamento entre as possíveis combinações de consumo. Dessa forma, ele percebeu que a simples ordenação entre as alternativas era suficiente à Teoria da Utilidade, eliminando o conceito de cardinalidade (CUSINATO, 2003; READ, 2004). Com isso, eliminou-se a crítica à medição da utilidade.

A segunda principal crítica à teoria foi superada na década de 1930, com a apresentação de uma nova abordagem de John Hicks e R. G. D. Allen. Essa nova abordagem propôs uma mudança na interpretação da utilidade, não como a “causa das preferências, mas uma descrição das preferências” (CUSINATO, 2003).

Hicks (1956 apud BARBERÀ; HAMMOND; SEIDL, 2004a) explica em seu trabalho que a suposição sobre o comportamento dos consumidores pode ser traduzida a partir de uma escala de preferências. A alternativa de maior preferência é posicionada acima das demais, e assim por diante. Em determinada condição de mercado, o consumidor toma uma decisão, em outra condição, toma outra decisão, porém, as escolhas que o mesmo faz expressam sempre o mesmo ordenamento e devem, então, ser consistentes.

Com essa interpretação moderna da Teoria da Utilidade, a função de utilidade deixa de guiar o comportamento do indivíduo, tendo o objetivo de emular o comportamento de escolhas do mesmo, independente dos processos mentais individuais utilizados para realizar as escolhas (CUSINATO, 2003).

Segundo Barberà, Hammond e Seidl (2004a), provavelmente, apenas com a publicação de *Social Choice and Individual Values* (ARROW, 1951), o trabalho seminal de Kenneth J. Arrow, o formalismo de relações de preferências binárias foi popularizado e inteiramente reconhecido como uma distinta entidade técnica.

No caso da utilidade cardinal, as alternativas são ordenadas de forma decrescente, de acordo com o quanto o bem causa felicidade no indivíduo, sendo $u = \mathcal{U}(x_1, x_2, \dots, x_n)$ a função de utilidade do indivíduo sob o consumo dos bens x_i , com $i = 1, 2, \dots, n$. Entretanto, como visto anteriormente, havia grande dificuldade de se medir a utilidade de cada alternativa e de se obter as funções de utilidade. Isso foi superado com a axiomatização da utilidade ordinal.

A teoria moderna da Utilidade Esperada com risco foi desenvolvida por von Neumann e Morgenstern em sua obra *Teoria dos Jogos e Comportamento Econômico* (*Theory of Games and Economic Behavior*), em 1944. Os próprios autores afirmam que, em relação ao problema do comportamento racional, “as dificuldades conceituais e práticas da noção de utilidade e, particularmente as tentativas de descrevê-la como um número, são conhecidas” (NEUMANN; MORGENSTERN, 1953), não sendo esse o objetivo primário de seu trabalho. Entretanto, o mesmo trouxe respostas, por meio das bases axiomáticas desenvolvidas, para os questionamentos sobre a real aplicação da maximização da utilidade esperada na representação da tomada de decisão e por não utilizar outros princípios para determinar as preferências (CUSINATO, 2003).

Escolhas consistentes são aquelas feitas a partir de um conjunto de axiomas logicamente equivalentes à uma função de utilidade. Axiomas são princípios simples, que, considerados, tornam o tomador de decisões racional. Read (2004) afirma que “se as preferências do agente obedecem a esses axiomas, então, elas podem ser descritas por uma função de utilidade, que se origina de uma transformação linear”. Como a medição da utilidade proposta apenas resume a preferência individual, o conceito vai ao encontro da definição de utilidade proposta por Benthan. Com a nova abordagem da Utilidade Esperada, o estudo das preferências ganha um maior aporte matemático. A próxima seção discute conceitos importantes para o entendimento da Teoria da Utilidade e da Teoria da Escolha Social.

2.1.1 Conceitos fundamentais e Axiomatização

A abordagem de von Neumann e Morgestern é a de maior influência, sendo as funções de utilidade que seguem seus axiomas denominadas *funções de utilidade von Neumann-Morgenstern* (ou funções de utilidade vN-M). A fim de caracterizar essa função de utilidade, serão apresentados os conceitos chaves e axiomas, que, em grande parte, serão utilizados para agregar preferências sociais.

Primeiramente, é importante definir a notação utilizada para expressar as preferências de um indivíduo. Baseando-se na notação utilizada por Shoham e Leyton-Brown (2008) e por Norvig e Russell (2013), dadas duas alternativas $x_1, x_2 \in X$, sendo X o conjunto de alternativas (ou resultados), se:

- $x_1 \succeq x_2$, entende-se que a alternativa x_1 é fracamente preferível à x_2 , o que significa que x_1 é preferível à x_2 ou o indivíduo é indiferente às alternativas.
- $x_1 \sim x_2$, entende-se que a alternativa x_1 é tão boa quanto à x_2 , o indivíduo é indiferente.
- $x_1 \succ x_2$, entende-se que o indivíduo prefere estritamente a alternativa x_1 à x_2 .

Ainda segundo [Shoham e Leyton-Brown \(2008\)](#), apesar da conveniência das últimas duas notações, pode-se representar preferências usando apenas a primeira, sendo que:

- $x_1 \sim x_2$ pode ser representado por “ $x_1 \succeq x_2$ e $x_2 \succeq x_1$ ”
- $x_1 \succ x_2$ pode ser representado por “ $x_1 \succeq x_2$ e não $x_2 \succeq x_1$ ”

À relação entre as ações a serem escolhidas pelo indivíduo e os possíveis resultados, dá-se o nome de loterias. Formalmente, uma loteria é uma distribuição de probabilidades sobre os resultados, na forma $[p_1 : x_1, \dots, p_n : x_n]$, onde $x_i \in X$, com $i = (1, 2, \dots, n)$; as probabilidades $p_i \geq 0$, sendo $\sum_{i=1}^n p_i = 1$. O conjunto de loterias denotado por \mathcal{L} é o conjunto de todas as loterias simples sobre o conjunto de alternativas X . A partir desse conjunto, pode-se definir a relação fraca, ou não-estrita, de preferências \succeq , um subconjunto do produto cartesiano $\mathcal{L} \times \mathcal{L}$ ([SHOHAM; LEYTON-BROWN, 2008](#); [CUSINATO, 2003](#)).

A partir das definições anteriores, pode-se iniciar a apresentação dos axiomas da Teoria da Utilidade.

Axioma 2.1 (Completeness ou Ordenabilidade). $\forall x_1, x_2$, ou $x_1 \succeq x_2$ ou $x_2 \succ x_1$ ou $x_1 \sim x_2$.¹

O axioma da Completeness ([SHOHAM; LEYTON-BROWN, 2008](#)) ou Ordenabilidade ([NORVIG; RUSSELL, 2013](#)) estabelece que o indivíduo deve ordenar as alternativas, sendo permitido haver empates. O indivíduo não pode evitar a decisão.

Axioma 2.2 (Transitividade). Sendo $x_1, x_2, x_3 \in X$, se $x_1 \succeq x_2$ e $x_2 \succeq x_3$, então $x_1 \succeq x_3$.²

A transitividade é um requisito razoável para se evitar a escolha cíclica. Se a preferência do indivíduo é não-transitiva, então pode haver uma relação entre as alternativas do seguinte modo: $x_1 \succeq x_2$, $x_2 \succeq x_3$, mas $x_3 \succeq x_1$. Segundo [Shoham e Leyton-Brown \(2008\)](#), esse comportamento não pode ser chamado de racional. Como exemplo, pode-se pensar no caso de que cada alternativa seja um bem econômico. Nesse caso, com a relação cíclica, o indivíduo estará sempre disposto a trocar o bem atual por outro.

Axioma 2.3 (Substitutibilidade). Se $x_1 \sim x_2$, então, para todo o conjunto de alternativas em X e o conjuntos de probabilidade p tem-se $[p : x_1, (1 - p) : x_i] \sim [p : x_2, (1 - p) : x_i]$.²

¹ Cf. ([SHOHAM; LEYTON-BROWN, 2008](#); [NORVIG; RUSSELL, 2013](#))

Esse axioma relaciona duas alternativas, que são indiferentes ao indivíduo, às loterias as quais as alternativas são oferecidas. Dessa forma, a troca entre alternativas não muda o resultado da loteria.

Usando a notação $P_\ell(x_i)$ para a probabilidade da alternativa x_i ser selecionada pela loteria ℓ , apresenta-se o próximo axioma, o qual reflete que o indivíduo é indiferente à loterias que inferem a mesma probabilidade sobre as alternativas, não importando se a probabilidade é expressa através de apenas uma loteria ou de uma loteria de loterias (SHOHAM; LEYTON-BROWN, 2008). Em outras palavras, duas loterias consecutivas podem ser compactadas em uma loteria equivalente, chamada de “nada de diversão no jogo” (NORVIG; RUSSELL, 2013), pois o número de vezes em que o dado é lançado não afeta a preferência do apostador (SHOHAM; LEYTON-BROWN, 2008).

Axioma 2.4 (Decomponibilidade). Se $\forall x_i \in X, P_{\ell_1} = P_{\ell_2}$, então, $\ell_1 = \ell_2$.²

Axioma 2.5 (Monotonicidade). Se $x_1 \succ x_2$ e seja p, q probabilidades tal que $p > q$, então, $[p : x_1, 1 - p : x_2] \succ [q : x_1, 1 - q : x_2]$.²

Basicamente, a Monotonicidade expressa a preferência do indivíduo em desejar mais do melhor resultado (SHOHAM; LEYTON-BROWN, 2008). Ou seja, se duas loterias tem as mesmas alternativas x_1, x_2 e $x_1 \succ x_2$, então a loteria de maior probabilidade para x_1 será a escolhida (NORVIG; RUSSELL, 2013). Shoham e Leyton-Brown (2008) explicam que o axioma tem esse nome, pois não depende de um valor numérico para as probabilidades, quanto mais peso, ou seja, quanto maior a probabilidade, de ocorrer x_1 , mais o indivíduo estará feliz.

Axioma 2.6 (Continuidade). Se $x_1 \succ x_2$ e $x_2 \succ x_3$, então, $\exists p \in [0, 1]$ tal que $x_2 \sim [p : x_1, 1 - p : x_3]$.²

Tendo o indivíduo racional a preferência $x_1 \succ x_2 \succ x_3$, há uma probabilidade p de que o mesmo seja indiferente entre escolher x_2 , por garantia, ou escolher a loteria em que x_1 tenha probabilidade p e x_3 , $(1 - p)$ (NORVIG; RUSSELL, 2013). Por fim, o Axioma da Independência, segundo Cusinato (2003), foi assumido apenas implicitamente e, segundo Castro e Faro (2005), impõe uma importante estrutura normativa à representação da utilidade v.N-M.

Axioma 2.7 (Axioma da Independência). Dadas as alternativas $x_1, x_2, x_3 \in X$ e a probabilidade $\alpha \in [0, 1]$, então, $x_1 \succeq x_2 \Leftrightarrow \alpha x_1 + (1 - \alpha)x_3 \succeq \alpha x_2 + (1 - \alpha)x_3$.³

Este axioma expressa que misturando a preferência de duas alternativas x_1, x_2 com uma terceira, x_3 , então, a preferência ainda é definida apenas por aquela dada entre x_1 e x_2 , independente do peso α da terceira alternativa adotada (CASTRO; FARO, 2005). Segundo Wagner (2009), o axioma 2.7 preenche uma lacuna na definição entre racionalidade e ordenamento.

² Cf. (SHOHAM; LEYTON-BROWN, 2008)

³ Cf. (CASTRO; FARO, 2005; CUSINATO, 2003)

Mantidas as condições de racionalidade, ou seja, respeitando-se os axiomas 2.1, 2.2 e 2.6, e supondo um ordenamento bem comportado entre dois elementos, um terceiro elemento comum não modifica o conjunto.

Então, sendo satisfeitos os axiomas de 2.1 a 2.7, existem funções de utilidade unidimensionais cujos valores os indivíduos esperam maximizar, sendo estabelecido o Teorema 2.1.

Teorema 2.1 (von Neumann e Morgenstern, 1944). Se uma relação de preferências \succeq satisfaz os axiomas da completude, transitividade, substitutibilidade, decomponibilidade, monotonicidade, continuidade e da independência, então, existe a função $u : \mathcal{L} \mapsto [0, 1]$ com as propriedades:

- 1 $u(x_1) \geq u(x_2)$ se e somente se $x_1 \succeq x_2$, e
- 2 $u([p_i : x_i, \dots, p_n : x_n]) = \sum_{i=1}^n p_i u(x_i)$.

A utilidade cardinal é utilizada pra expressar o risco na forma de distribuição de probabilidades. A prova do teorema 2.1 pode ser encontrada na literatura, mais especificamente em Shoham e Leyton-Brown (2008), Castro e Faro (2005), Cusinato (2003).

Entretanto, até o momento, apenas foi discutida a decisão individual. Em muitas aplicações, a decisão a ser tomada afeta a utilidade de vários indivíduos, devendo ser tomada coletivamente. A próxima seção tratará da Teoria da Escolha Social e o desafio de agregar-se preferências individuais em uma decisão coletiva.

2.2 Teoria da Escolha Social

A Teoria da Escolha Social trata da análise do processo de tomada de decisão por um coletivo, uma sociedade. Segundo Gaertner (2009), essa teoria tem como objetivo obter um veredito ou homologação coletiva partindo de opiniões e valores dos membros de uma dada comunidade ou cidadãos de uma sociedade. Para isso, as preferências individuais devem ser agregadas em uma preferência social, que deve refletir a opinião geral da sociedade.

O debate acerca de mecanismos de agregação de preferências, sendo os sistemas de votação os mais clássicos, remonta à Roma Antiga e à eleição dos senadores, segundo McLean (1990). Isso se deve pois não há registros escritos da Grécia Antiga sobre decisões com mais de três alternativas, sendo as decisões da Assembleia Ateniense julgadas par a par, ou seja, de forma binária (MCLEAN, 1990).

Muitos trabalhos importantes, como os de Ramon Lull (século XIII), Nicolas de Cusa (século XV) e Samuel von Pufendorf (1672) são colocados de lado em detrimento aos trabalhos mais conhecidos de Jean-Charles de Borda (1781) e de Marie Jean Antoine Nicolas de Caritat, o Marquês de Condorcet (1785), apesar de antecederem o debate e propor ideias similares aos últimos autores (MCLEAN, 1990; GAERTNER, 2009). O Marquês de Condorcet é tido como fundador da Teoria da Escolha Social.

O resultado mais expressivo de Condorcet é o paradoxo que leva seu nome. O Paradoxo de Condorcet observa que a decisão coletiva pode ser irracional (especialmente intransitiva), mesmo que as decisões individuais não o sejam.

Como exemplo, considere a situação em que uma assembleia municipal deve decidir sobre o investimento de um recurso financeiro e três alternativas são apresentadas: *Saúde*, *Educação* e *Urbanização*. Para um terço dos membros da assembleia, as alternativas são ordenadas como $Saúde \succ Educação \succ Urbanização$. Para outro um terço dos participantes, $Educação \succ Urbanização \succ Saúde$ e para o restante, $Urbanização \succ Saúde \succ Educação$. Com as preferências definidas, percebe-se que a regra da maioria não pode ser aplicada apenas aos “primeiros lugares”, visto que cada terço da assembleia prefere uma alternativa distinta. Dessa forma, comparando as alternativas de forma binária, $Saúde \succ Educação$ por uma maioria de dois terços. Entretanto, a mesma proporção ocorrem os resultados $Educação \succ Urbanização$ e $Urbanização \succ Saúde$. Dessa forma, a preferência da assembleia é cíclica, e intransitiva, já que $Saúde \succ Educação \succ Urbanização \succ Saúde$, o que é paradoxal.

Para uma alternativa ser classificada como Vencedora de Condorcet (*Condorcet Winner*), se houver alguma, ela deve ser pelo menos tão boa quanto cada uma das demais (GAERTNER, 2009). No exemplo anterior, pode-se observar que não há uma alternativa vencedora.

Borda (1781), por sua vez, propõe um método, chamado posteriormente de Contagem de Borda, que evita o Paradoxo de Condorcet, mas que viola uma das condições do Teorema da Impossibilidade de Arrow, a condição da Independência das Alternativas Irrelevantes, que será visto na próxima seção.

A Contagem de Borda, basicamente, confere à alternativa de maior preferência o peso n , sendo n o número total de alternativas, e para as preferências subsequentes, o peso é sempre descontado de uma unidade. Dessa forma, a alternativa de menor preferência tem peso unitário. Soma-se, então, os valores das preferências de cada indivíduo, sendo vencedora a alternativa com maior número de pontos. Esse método analisa todas as alternativas do ponto de vista de cada membro da comunidade.

Por exemplo, para a mesma situação anterior com o número de alternativas $n = 3$, a contagem de Borda ficaria como o mostrado na Tabela 1. Observa-se que a decisão cíclica permanece com o empate entre todas as alternativas, visto que o número de indivíduos para cada perfil de preferências é igual.

Tabela 1 – Contagem de Borda para três alternativas em uma situação de escolha cíclica.

Assembleia	Saúde	Educação	Urbanização
1º terço de indivíduos	3	2	1
2º terço de indivíduos	1	3	2
3º terço de indivíduos	2	1	3
Contagem total	6	6	6

Fonte: Próprio autor.

Teorema da Impossibilidade de Arrow é o marco da teoria moderna da Escolha Social cujo resultado teve grande influência após sua publicação, mas, sobretudo, trouxe contribuição devido ao método axiomático utilizado na análise (LIST, 2013). Arrow (1951) é defensor da análise ordinal da utilidade, pois “parece não fazer sentido somar a utilidade de um indivíduo, uma magnitude psíquica de sua mente, com a utilidade de outro indivíduo”. Ainda segundo o autor, “mesmo que as preferências dos indivíduos possam ser medidas (utilidade) ela é inter-pessoalmente incomparável”.

Como visto nos exemplos anteriores, desenvolver abordagens de escolha social é difícil. Com qualquer método e apenas três alternativas podem ser apresentados exemplos de sistemas de escolha de forma a lançar dúvidas sobre sua confiabilidade. Ao invés de decretar que *nenhum sistema de votação é justo*, o resultado de Arrow é um comentário negativo sobre a metodologia (SAARI, 2016). Antes de apresentar os fundamentos do Teoria da Impossibilidade de Arrow, serão apresentados conceitos relativos à Escolha Social.

2.2.1 Modelo Formal

Seja $N = \{1, 2, \dots, k\}$ o número de j indivíduos ou agentes, com $j = (1, \dots, k)$, ou seja, o conjunto fixo de membros participantes da sociedade, sendo $k \geq 2$. E seja $X = \{x_1, x_2, \dots, x_n\}$ o conjunto finito de alternativas ou resultados (SHOHAM; LEYTON-BROWN, 2008; MOULIN, 1991; GAERTNER, 2009).

Segundo Shoham e Leyton-Brown (2008), pode-se expandir para o caso multiagentes a notação de preferência definida na seção 2.1.1. Dessa forma, para o indivíduo $j = (1, \dots, k)$, se x_1 é tão bom quanto a alternativa x_2 , então $x_1 \succeq_j x_2$. Analogamente, se x_1 é estritamente preferível à alternativa x_2 , logo, $x_1 \succ_j x_2$. E se o agente j é indiferente em relação as alternativas, $x_1 \sim_j x_2$.

Sendo as preferências transitivas, uma relação de preferências de um agente induz ao ordenamento (não-estrito) de preferências. Assim como no caso individual, denota-se \succeq como o ordenamento de preferências. Quando apresentada sem o índice, significa o ordenamento de preferências da sociedade e o ordenamento de preferências do indivíduo j é dada por \succeq_j (GAERTNER, 2009). Dá-se o nome de conjunto de relações de preferências P , conjunto de preferências possíveis sobre X , isto é, $P = X \times X$. Um subconjunto de P é o conjunto R , $R \subset P$, o conjunto das preferências racionais possíveis, portanto, ordenamentos de preferências, sobre X . E entende-se \succeq_i como elemento de R . O conjunto $[\succeq]$ de ordenamentos de preferências $\{\succeq_1, \succeq_2, \dots, \succeq_k\} \in R^n$ é chamado de perfil de preferências (*preference profile*) e contém o ordenamento de preferências de cada agente (CASTRO; FARO, 2005; SHOHAM; LEYTON-BROWN, 2008).

Segundo Arrow (1950), “o objetivo da sociedade então descrita é maximizar a utilidade social, ou o bem-estar social, para quaisquer limitações tecnológicas ou de recursos relevantes, escolhendo o estado social que reflete o maior bem-estar possível aos membros da sociedade”. A partir das notações apresentadas, pode-se definir as funções de escolha e de bem-estar social.

Definição 2.1 (Função de escolha social⁴). Uma função de escolha social (FES), sobre N e X , é uma função $C : R^n \mapsto X$.

Definição 2.2 (Função de bem-estar social⁵). Uma função de bem-estar social (FBS) significa um processo ou uma regra a qual, para cada relação de preferências dos estados sociais alternativos $\succeq_1, \dots, \succeq_k$ individual (sendo uma ordem por indivíduo), resulta em um ordenamento social correspondente às alternativas, $W : R^n \mapsto R$.

Segundo Shoham e Leyton-Brown, funções de bem-estar social “produzem ordenamentos totais sobre o conjunto de alternativas” (SHOHAM; LEYTON-BROWN, 2008, p. 256). Uma FBS mapeia os perfis de preferência individuais em um ordenamento social. As condições para a existência de uma FBS serão examinadas na seção 2.3.

A principal diferença entre FBS e FES é que a primeira realiza um mapeamento completo das alternativas individuais em um ordenamento social, enquanto a segunda “simplesmente seleciona uma das alternativas, ou, em uma versão mais genérica, algum subconjunto” (SHOHAM; LEYTON-BROWN, 2008, p. 255). As condições de existência de uma função de escolha social serão examinadas na seção 2.5.

2.3 O Teorema da Impossibilidade de Arrow

Para se atingir o resultado da impossibilidade observado por Arrow, necessita-se conhecer as condições elencadas para a construção de uma função de bem-estar social. As condições impostas por Arrow (1951) são, individualmente, condições razoáveis para a construção de uma regra de agregação de preferências. Contudo, consideradas juntas, levam ao resultado negativo. Serão apresentadas as condições originais e conceitos presentes na literatura, principalmente em Castro e Faro (2005), Shoham e Leyton-Brown (2008), Moulin (1991), Gaertner (2009), Maskin e Sen (2014) e Morreau (2014), a fim de tornar o entendimento mais claro.

Como apresentado no modelo formal na seção 2.2.1, um conjunto de alternativas é apresentado aos indivíduos que irão tomar a decisão. Segundo Arrow (1951, p. 12), “o indivíduo decisor considera, por sua vez, todos os pares de alternativas possíveis, digamos x e y , e para cada par, ele toma uma e somente uma decisão: $x \succ y$, $x \sim y$, ou $y \succ x$ ”.

As alternativas apresentadas ao decisor são comparáveis, seguindo o Axioma 2.1. Esse axioma é denominado Axioma I em Arrow (1951, p. 13). Além disso, o ordenamento proposto pelo indivíduo segue o expresso pelo Axioma 2.2 (ou Axioma II para Arrow), ou seja, as decisões são consideradas consistentes (ARROW, 1951, p. 13). Considerando decisões tomadas que satisfazem os Axiomas 2.1 e 2.2, então, estas decisões são racionais (ARROW, 1951, p. 19).

Diante da dificuldade de medir-se a utilidade e da necessidade de buscar a maximização, Arrow afirma:

⁴ Cf. (SHOHAM; LEYTON-BROWN, 2008, p. 255)

⁵ Cf. (ARROW, 1951, p. 23),(SHOHAM; LEYTON-BROWN, 2008, p. 256)

Como qualquer tipo de comportamento descrito pela maximização, a mensurabilidade do bem-estar social não precisa ser adotada; tudo o que importa é a existência de um ordenamento social que satisfaça os Axiomas I (2.1) e II (2.2). [...], todo o necessário para definição de tal ordenamento é saber o ranqueamento relativo de cada par de alternativas. (ARROW, 1951, p. 22)

As condições apresentadas a seguir são consideradas na situação em que o número k de indivíduos membros da sociedade é pelo menos dois, ou seja, $k \geq 2$. Além disso, é assumido que o conjunto de alternativas apresentadas aos indivíduos possui pelo menos três alternativas. Dessa forma, sendo $X = \{x_1, \dots, x_n\}$ o conjunto de alternativas e x_i a i -ésima alternativa, então, $n \geq 3$. O caso para duas alternativas pode ser verificado em (ARROW, 1951, p. 48-51).

Condição 2.1 (Domínio Irrestrito⁶). Entre todas as alternativas, há um conjunto R de três alternativas tal que, para qualquer conjunto de ordenamentos individuais $\{P_1, \dots, P_k\}$ de alternativas em R , há um conjunto admissível de ordenamentos individuais $\{\succeq_1, \dots, \succeq_k\}$ de todas as alternativas de forma que, para cada indivíduo j , $x_1 \succeq_j x_2$ se e somente se $x_1 \succeq_j x_2$ em P_i , $\forall x_1, x_2 \in R$.

A Condição 2.1, que na literatura passou a ser chamada de Domínio Irrestrito (UD - *Unrestricted Domain* (MASKIN; SEN, 2014)), é, segundo Arrow (1951, p. 25), “uma restrição na forma da função de bem-estar social, desde que [...] é requerido que para um número suficiente de conjuntos de ordenamento individuais a função de bem-estar social tenha como resultado um ordenamento social verdadeiro”.

A definição da condição UD pode ser colocada em outras palavras, de forma a torná-la mais simples. Segundo Maskin e Sen (2014, p. 34), ela pode ser colocada como a condição que “para cada conjunto logicamente possível de ordenamento individual de preferências, há um ordenamento social \succeq ”. Ou seja, quaisquer tipos de ordenamentos podem ser realizados em relação às alternativas apresentadas e, ainda assim, um único ordenamento social deve ser especificado.

É possível ainda encontrar definições da Condição 2.1 que se referem aos ordenamentos possíveis como ordenamentos fracos (*weak orderings*) do conjunto de alternativas X . O adjetivo *fraco* significa que o ordenamento permite e não exclui as relações de indiferença entre os pares de alternativas avaliadas.

Condição 2.2 (Monotonicidade⁷). Sejam $\{\succeq_1, \dots, \succeq_k\}$ e $\{\succeq'_1, \dots, \succeq'_k\}$ dois conjuntos de relações de ordenamento individuais; \succeq e \succeq' os ordenamentos sociais e \succ e \succ' as preferências sociais resultantes. Suponha que, para cada j , as relações de ordenamento individuais estão conectadas das seguintes formas:

- a. para x'_1 e x'_2 distintos de uma outra alternativa x_1 , $x'_1 \succeq'_j x'_2$ se e somente se $x'_1 \succeq_j x'_2$;

⁶ Cf. (ARROW, 1951, p. 24)

⁷ Cf. (ARROW, 1951, p. 26)

- b. $\forall x'_2, x_1 \succeq_j x'_2$ implica em $x_1 \succeq'_j x'_2$; e
- c. $\forall x'_2, x_1 \succ_j x'_2$ implica em $x_1 \succ'_j x'_2$.

Então, se $x_1 \succ_j x_2$, $x_1 \succ'_j x_2$.

A condição de Monotonicidade para o caso de múltiplos agentes é similar ao verificado no Axioma 2.5. Pode-se ainda entendê-la de outra forma: suponha um conjunto de ordenamentos individuais $\succeq_1, \dots, \succeq_k$ inicial e um ordenamento de preferência social \succeq . Se uma alternativa é ranqueada com maior preferência em um novo conjunto de ordenamentos individuais $\succeq'_1, \dots, \succeq'_k$ por todos os membros da sociedade e as demais alternativas permanecem inalteradas, então, essa alternativa deve ser ranqueada com maior preferência pela sociedade (ARROW, 1951, p. 25). Essa condição também é chamada de Associação Positiva (ARROW, 1951, p. 25), (CRAVEN, 1992, p. 44).

Condição 2.3 (Independência de Alternativas Irrelevantes⁸). Sejam $\succeq_1, \dots, \succeq_k$ e $\succeq'_1, \dots, \succeq'_k$ dois conjuntos de relações de ordenamento individuais e $C(R)$ e $C'(R)$ as funções de escolha social correspondentes. Se, para cada indivíduo j e todo x_1 e x_2 em determinado ambiente R , $x_1 \succeq_j x_2$ se e somente se $x_1 \succeq'_j x_2$, então, $C(R)$ e $C'(R)$ são iguais.

A condição da Independência de Alternativas Irrelevantes (IIA - *Independence of Irrelevant Alternatives*) é considerada por muitos pesquisadores a condição responsável pelo teorema da impossibilidade de Arrow (CATO, 2014). Ela estabelece que a relação de preferência entre duas alternativas depende apenas da relação entre essas mesmas duas alternativas, sendo independente de qualquer outra alternativa. De certa forma, a condição é criticada por barrar maiores informações do ordenamento, tornando-se muito restritiva (CATO, 2014; GAERTNER, 2009).

Entretanto, a condição foi estabelecida por Arrow de forma a excluir a comparação interpessoal de utilidade, ou por algum mecanismo de medição direta, ou por comparação com outros estados sociais alternativos (BARBERÀ; HAMMOND; SEIDL, 2004b, p. 1202), descartando informações como a intensidade da preferência, propositadamente. O exemplo mais utilizado para essa condição é uma eleição com três candidatos. Após os ordenamentos de preferências individuais feitos pelos membros da sociedade, caso um dos candidatos saia da disputa, o ordenamento entre os demais não pode mudar.

Além disso, Yuan (YUAN, 2015) afirma que “o importante objetivo de Arrow ao impor a condição IIA foi de eliminar a possibilidade de indivíduos se darem melhor sob um processo de decisão coletivo se eles não apresentarem suas verdadeiras preferências como entradas do procedimento de decisão coletiva”, ou seja, manipular a decisão final baseando-se nas preferências individuais.

As próximas definições e condições dizem respeito ao que Arrow chama de “condições de soberania dos cidadãos”, ou seja, a soberania dos indivíduos sobre suas decisões (ARROW,

⁸ Cf. (ARROW, 1951, p. 27)

1951, p. 28). Para Arrow, uma FBS imposta significa haver estados sociais em que os membros da comunidade não podem expressar suas preferências, independentemente das preferências individuais (ARROW, 1951, p. 28).

Definição 2.3 (Imposição⁹). Um função de bem-estar social é dita imposta se, para algum par de alternativas distintas x_1 e x_2 , $x_1 \succeq x_2$ para qualquer conjunto de ordenamentos individuais $\succeq_1, \dots, \succeq_k$, onde \succeq é o ordenamento social correspondente a $\succeq_1, \dots, \succeq_k$.

Condição 2.4 (Não-imposição¹⁰). A função de bem-estar social não deve ser imposta.

A condição de Não-imposição estabelece que nenhuma preferência deve ser imposta à sociedade, por exemplo, como cita Arrow, “[...] um tipo de escolha social na qual as decisões são tomadas de acordo com um código de conduta costumeiro” (ARROW, 1951, p. 28). Essa condição, então, argumenta sobre a possível imposição do ambiente sobras as escolhas individuais. Já a condição de Dominância reflete a dominação de um indivíduo da sociedade sobre as escolhas dos demais.

Definição 2.4 (Dominância¹¹). Uma função de bem-estar social é dita dominante, ou ditatorial, se existe um indivíduo j tal que, para todo x_1 e x_2 , $x_1 \succ_j x_2$ implica em $x_1 \succ x_2$ independentemente dos ordenamentos $\{\succeq_1, \dots, \succeq_k\}$ dos demais indivíduos além de j , onde \succ é a relação estrita de preferências social correspondente à $\{\succeq_1, \dots, \succeq_k\}$.

Condição 2.5 (Não-ditadura, Não-dominância¹²). A função de bem-estar social não deve ser ditatorial ou dominante.

Por fim, a Condição 2.5 (ND - *Nondictatorship*) é definida como uma forma de escolha social não-coletiva, em que as preferências da sociedade são baseadas nas preferências de apenas um membro, ou um pequeno grupo de membros (ARROW, 1951, p. 30). A Definição 2.4 classifica uma FBS em que as preferências individuais não são levadas em consideração e a Condição 2.5 é razoável, de modo que, com o objetivo de agregar preferências, a preferência de um indivíduo não pode subjugar as preferências dos demais. A partir das condições estabelecidas, o resultado negativo tem o enunciado do Teorema 2.2, chamado originalmente de Teorema da Possibilidade Geral de Arrow.

Teorema 2.2 (Teorema da Possibilidade Geral de Arrow ou Teorema da Impossibilidade de Arrow, 1951¹³). Se há, ao menos, três alternativas as quais os membros da sociedade são livres para ordenar de qualquer maneira (Domínio Irrestrito), então toda função de bem-estar social

⁹ (ARROW, 1951, p. 28)

¹⁰ Cf. (ARROW, 1951, p. 29)

¹¹ Cf. (ARROW, 1951, p. 30)

¹² Cf. (ARROW, 1951, p. 30)

¹³ Cf. (ARROW, 1951, p. 59)

que satisfaz as condições 2 (Monotonicidade) e 3 (Independência de Alternativas Irrelevantes) e que produz ordenamentos sociais que satisfazem os Axiomas I e II (Axiomas 2.1 e 2.2, respectivamente) são ou impostas ou ditatoriais.

Versões alternativas do teorema podem ser encontradas na literatura, diferindo, principalmente na quantidade de condições e nas próprias condições, mas sem diferir do resultado encontrado por Arrow. Shoham e Leyton-Brown (2008, p. 260) apresentam três condições: Eficiência de Pareto, IIA e ND. Castro e Faro (2005, p. 152) enunciam quatro condições: UD, Condição de Pareto ou Axioma da Unanimidade, ND e IIA. As mesmas quatro condições são enunciadas por Gaertner (2009, p. 20) e Maskin e Sen (2014, p. 5), entretanto, o autor nomeia a condição de Pareto como Princípio fraco de Pareto. Morreau (2014) enuncia cinco condições. Além das condições UD, IIA e ND, utiliza o Princípio fraco de Pareto e a condição chamada Ordenamento Social (SO - *Social Ordering*). Morreau (2014) afirma que a condição SO não foi estabelecida separadamente por Arrow na definição original das condições para a construção de uma função de bem-estar social. Essa condição é a combinação dos Axiomas I (2.1) e II (2.2) definidos por Arrow, a fim de que a FBS reflita uma escolha racional. Na definição dada pelo autor, a condição SO requer que o resultado da agregação de preferências seja um ordenamento fraco do conjunto de alternativas X , além de ser completo e transitivo (MORREAU, 2014).

As condições relativas ao Princípio de Pareto ou Unanimidade, a qual será abreviada por WP (*Weak Pareto*), merece uma atenção maior por refletir uma noção de otimização, o que será necessário para análises posteriores neste trabalho. Assim como a condição SO, WP não é uma das condições originais estabelecidas por Arrow. Entretanto, segundo Craven (1992, p. 44), ela pode ser derivada de duas outras, a saber: Não-imposição e Associação Positiva, também chamada de Monotonicidade.

Ainda segundo o autor, a condição de Não-imposição “assegura que há circunstâncias nas quais cada alternativa é escolhida de um par de alternativas” e a Monotonicidade “considera o efeito da mudança de preferência entre duas alternativas por alguns indivíduos” (CRAVEN, 1992). Em outras palavras, a Monotonicidade fortalece a alternativa com maior preferência. Garantida a Não-imposição, iniciando-se do caso individual, em que $x_1 \succ x_2$, e expandindo de forma que todo indivíduo prefere x_1 a x_2 , ou seja, fortalecendo a alternativa x_1 . Dessa forma, se todos preferem x_1 a x_2 , x_1 deve ser a escolha social (CRAVEN, 1992).

Shoham e Leyton-Brown (2008) tratam essa mesma condição como Eficiência de Pareto. Essa nomenclatura deriva do conceito de Ótimo ou Otimalidade de Pareto. Basicamente, esse conceito estabelece um limite de otimização, chamada de fronteira de Pareto, a qual é a solução ótima. Para o caso da escolha social, esse limite de otimização pode ser entendido como o conjunto solução em que a utilidade de qualquer membro da sociedade não pode aumentar sem que a utilidade de algum outro membro decresça (SHOHAM; LEYTON-BROWN, 2008, p. 61), sendo esse conjunto Pareto-dominante.

Definição 2.5 (Pareto-dominante¹⁴). O perfil de preferências $[\succeq] \in R$ é Pareto-dominante em

relação ao perfil de preferências $[\succeq]'$ se para todo $k \in N$, $u_k([\succeq]) \geq u_k([\succeq]')$ e existe algum indivíduo $m \in N$ para o qual $u_m([\succeq]) > u_m([\succeq]')$.

Definição 2.6 (Otimalidade de Pareto). O perfil de preferências $[\succeq]$ é Pareto-ótimo, ou Pareto-eficiente, se não existe outro perfil de preferências $[\succeq]' \in R$ que domina $[\succeq]$.¹⁴

Com isso, a condição WP pode ser enunciada como:

Condição 2.6 (Princípio de Pareto ou da Unanimidade¹⁵). A função de bem-estar social é Pareto-eficiente se para qualquer $x_1, x_2 \in X$, $\forall j x_1 \succ_j x_2$ implica em $x_1 \succ x_2$.

Com a descrição da condição 2.6 fica clara a nomenclatura “unanimidade”, ou seja, se todos os agentes concordam com a preferência entre duas alternativas, essa deve ser a preferência social entre essas alternativas. O maior problema dessa condição é quando não há concordância entre, por exemplo, dois agentes, levando o princípio à falha na comparação dos respectivos ordenamentos de preferência (MASKIN; SEN, 2014, p. 2). Além disso, segundo Morreau (2014), apesar de parecer uma condição inofensiva, em combinação com a condição UD, elas restringem fortemente as possibilidades de escolha social.

Definida a condição WP, o Teorema da Impossibilidade pode ser reescrito levando-se em consideração as condições UD, WP, IIA e ND.

Teorema 2.3 (Teorema da Impossibilidade de Arrow). Se há, ao menos, três alternativas, qualquer FBS que satisfaça as condições Domínio Irrestrito, Unanimidade e Independência de Alternativas Irrelevantes é ditatorial¹⁶.

A prova do teorema de Arrow pode ser encontrada em várias fontes na literatura. Em Arrow (1951, p. 51-59), há a demonstração inicial do resultado estabelecido. Craven (1992, p. 36-43) apresenta três provas do teorema chamadas: a epidemia, o coletivo e o ditador. Geanakoplos (2005) demonstra três breves provas e Yuan (2015) apresenta uma prova simples e elegante utilizando o conceito do Eleitor Principal (*Pivotal Voter*). Falik e Friedgut (2011) propõem uma prova algébrica para uma versão robusta do Teorema da Impossibilidade, “mostrando que a impossibilidade permanece mesmo quando as restrições são quase sempre satisfeitas”. Outros resultados podem ser encontrados em Gaertner (2009, p. 21-33), Maskin e Sen (2014, p. 35-37), Castro e Faro (2005, p. 153-157) e Brandt et al. (2016). Uma demonstração adaptada de Shoham e Leyton-Brown (2008) é encontrada no Apêndice A.

¹⁴ Cf. (SHOHAM; LEYTON-BROWN, 2008, p. 61)

¹⁵ Cf. (SHOHAM; LEYTON-BROWN, 2008, p. 260)

¹⁶ Cf. (SHOHAM; LEYTON-BROWN, 2008, p. 261)

2.4 Relaxando as condições de Arrow

A fim de que o resultado da estrutura proposta por Arrow (1951) seja evitado, basta que, pelo menos, uma das condições seja relaxada ou, ainda, que sejam abandonadas as restrições das entradas das regras de agregação e mais informações sejam utilizadas (LIST, 2013).

2.4.1 Relaxando o domínio irrestrito: preferências de pico único

Segundo Arrow (1951), p. 75), uma restrição à variedade de ordenamentos individuais possíveis foi proposta por Black (1948). Essa restrição trata da condição de preferências de pico único (*single-peakedness preferences*).

Baseado na análise política dos parlamentos da Europa no período anterior à II Guerra Mundial, onde havia uma clara organização Esquerda-Direita entre os partidos, Black (1948) propôs que a representação gráfica dos ordenamentos individuais devem ter um único pico. Sendo o número de indivíduos ímpar, essa consideração faz com que a regra de decisão pela maioria gere resultados possíveis (ARROW, 1951, p. 76), gerando um vencedor de Condorcet. Black (1948) provou que se o domínio da regra de agregação for restrito aos ordenamentos de preferência individuais que satisfaçam a condição do pico único, então a decisão cíclica da regra da maioria não irá ocorrer (LIST, 2013).

Um perfil de preferências $[\succeq] = \{\succeq_1, \dots, \succeq_k\}$ é de pico único se as alternativas podem ser alinhadas da esquerda para a direita, em alguma dimensão de conhecimento ou de ideologia, de forma que cada indivíduo tenha sua alternativa de maior preferência cercada de alternativas de preferência decrescente. A Figura 2.1 exemplifica as características de ordenamentos de pico único e de pico não único.

Formalmente, Arrow (1951, p. 77) define tal ordenamento como um Ordenamento Forte S , mostrado na Definição 2.7. Dessa definição, Arrow deriva outro conceito, o da intermediação entre as alternativas (*betweenness*), Definição 2.8.

Definição 2.7 (Ordenamento Total¹⁷). A relação de preferência S é um ordenamento total se,

- (a) $\forall x$, não há xSx , ou seja, não há preferência reflexiva;
- (b) $\forall x_1 \neq x_2$, ou x_1Sx_2 ou x_2Sx_1 ;
- (c) $\forall (x_1, x_2, x_3)$, x_1Sx_2 e x_2Sx_3 implica em x_1Sx_3 , ou seja, o ordenamento deve ser transitivo.

Definição 2.8 (Intermediação entre as alternativas¹⁸). Se a relação de preferência S é um ordenamento forte, define-se $B(x_1, x_2, x_3)$ com o significado de que uma das possibilidades ocorre: ou x_1Sx_2 e x_2Sx_3 ou x_3Sx_2 e x_2Sx_1 .

¹⁷ Cf. (ARROW, 1951, p. 77)]

¹⁸ Cf. (ARROW, 1951, p. 77)]

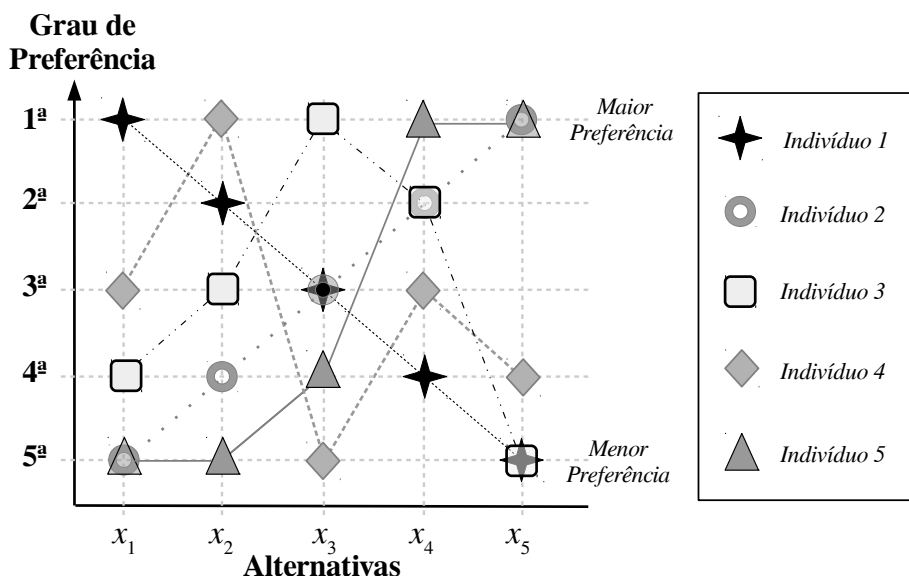


Figura 2.1 – Representação gráfica para exemplificação de ordenamentos de preferência de pico único e de pico não único para as alternativas x_1, x_2, x_3, x_4 e x_5 . Os indivíduos 1, 2 e 3 possuem preferências de pico único. Os indivíduos 4 e 5 possuem preferências de pico não único. No caso do indivíduo 4, a alternativa x_4 tem preferência maior que a alternativa x_3 , o que pode gerar decisões cíclicas. Para o indivíduo 5, as alternativas x_1 e x_2 empatadas com a menor preferência e as alternativas x_4 e x_5 empatadas com a maior preferência descaracterizam a condição de pico único. Fonte: Próprio autor.

De acordo com a Definição 2.8, $B(x_1, x_2, x_3)$ significa que x_2 está entre as alternativas x_1 e x_3 (ARROW, 1951, p. 77), intermediando a relação de preferência, independentemente da ordem de preferências entre as alternativas. A partir das Definições 2.7 e 2.8, Arrow estabelece que se x_1, x_2, x_3 são distintos, então uma e apenas uma das seguintes situações ocorre: $B(x_1, x_2, x_3), B(x_2, x_1, x_3), B(x_2, x_3, x_1)$. Black (1948 apud ARROW, 1951) propõe uma substituição da condição do Domínio Irrestrito:

Condição 2.7 (Domínio Restrito pela hipótese das preferências de pico único). Para todos os conjuntos de ordenamentos individuais $\{\succeq_1, \dots, \succeq_k\}$ que satisfaçam a hipótese de preferências de pico único, o ordenamento social correspondente \succeq deverá ser um ordenamento fraco.

Com essa consideração, a impossibilidade é contornada. A regra da maioria torna-se uma função de bem-estar social que satisfaz a nova condição proposta por Black (1948) e as condições 2.2-2.5 originais, para qualquer número de alternativas dado que o número de indivíduos é ímpar. A prova dessa afirmação pode ser encontrada em (ARROW, 1951, p.78-80).

2.4.2 Relaxando a condição de Pareto

Como trata-se de uma condição de eficiência, a condição de Pareto é difícil de se deixar de lado (LIST, 2013). De acordo com Campbell e Kelly (2002), o teorema de Arrow implica que se uma função de bem-estar social é não ditatorial, transitiva e satisfaz a condição IIA, então

essa função violará o critério de Pareto¹⁹. Ainda, segundo List (2013), uma regra de agregação de preferências que se torna possível rejeitando-se a condição de Pareto é imposta. Onde, seja o conjunto de perfis de preferências $\{\succeq_1, \succeq_2, \dots, \succeq_k\}$ em R , há um perfil $[\succeq_{imposto}]$ fixado antecipadamente. Apesar de ignorar totalmente as preferências individuais, tal regra de agregação satisfaz as demais condições de Arrow (LIST, 2013).

Campbell e Kelly (1993) afirmam que não há muito o que se ganhar relaxando o critério de Pareto. Nessa condição, as funções de bem-estar arrovianas ou dão a algum indivíduo poder ditatorial ou ordena socialmente muitos pares de alternativas sem a consulta a nenhum indivíduo. Os trabalhos (CAMPBELL; KELLY, 1993; CAMPBELL; KELLY, 2002) apresentam uma avaliação das funções de bem-estar social admissíveis quando o princípio de Pareto é relaxado. Essa avaliação utiliza a abordagem da Troca (*Trade-off*), que define uma metodologia de medição do quanto uma função de bem-estar social viola ou satisfaz o princípio de Pareto.

Sen (1970 apud LIST, 2013) mostra que o princípio de Pareto conflita com a noção de liberalismo. Esse conceito se refere à um sacrifício das preferências individuais em detrimento à preferência social. A situação exposta em (SEN, 1970) analisa o caso em que é dada a liberdade aos indivíduos da sociedade determinar pelo menos uma escolha social. Permanecendo as demais alternativas iguais para este indivíduo e para o resto da sociedade.

Entretanto, somando-se às condições do Domínio Irrestrito e do princípio de Pareto, há uma impossibilidade. Segundo Sen (1970), “o princípio de Pareto tem sido considerado como uma expressão da liberdade individual, mas em escolhas envolvendo mais de duas alternativas as consequências são profundamente não liberais”. Portanto, se a individualidade é respeitada, há um sacrifício na eficiência de Pareto (LIST, 2013).

2.4.3 Relaxando a Independência de Alternativas Irrelevantes

Assim como a condição de Pareto, se a condição de Independência de Alternativas Irrelevantes for deixada de lado, as demais condições podem ser atendidas, fugindo da impossibilidade. De acordo com Dryzek e List (2004), nesta situação, a regra de Borda e outras regras posicionais tornam-se regras que atendem às demais condições de Arrow.

Entretanto, tais regras são suscetíveis à manipulação por parte dos membros da sociedade. Seja o exemplo apresentado em (LIST, 2013): uma sociedade com $k = 15$ indivíduos e quatro alternativas $\{x_1, x_2, x_3, x_4\}$. Assim, com $n = 4$, a alternativa de maior preferência recebe o valor 4, a de segunda maior preferência o valor 3 e assim por diante. Considerando o número de indivíduos de cada grupo, a contagem de Borda dá o peso de cada alternativa dadas as preferências de todos os indivíduos da sociedade. As Tabelas 2 e 3 mostram dois perfis de ordenamentos de preferências com uma mudança sutil no ordenamento de apenas um indivíduo. Essa mudança, na preferência entre x_2 e x_1 e entre x_3 e x_4 , não altera a preferência do indivíduo 1 sobre o par x_1 e x_3 .

¹⁹ A prova da afirmação pode ser encontrada em (CAMPBELL; KELLY, 2002, p53-54).

Tabela 2 – Perfil de ordenamentos de preferências individuais.

	Indivíduo 1	Indivíduos 2 a 7	Indivíduos 8 a 15
1ª preferência	x_2	x_1	x_3
2ª preferência	x_1	x_3	x_1
3ª preferência	x_3	x_4	x_2
4ª preferência	x_4	x_2	x_4

Fonte: Adaptado de List (2013).

Considere contagem de Borda, considerando a alternativa de maior preferência com o valor k , sendo X o conjunto de alternativas e $k = |X|$. A alternativa de segunda maior preferência recebe o valor $k - 1$, e assim por diante. Dessa forma, a pontuação de Borda para a Tabela 2 fica:

- $x_1 : 6 \times 4 + 9 \times 3 = 51$
- $x_2 : 1 \times 4 + 8 \times 2 + 6 \times 1 = 26$
- $x_3 : 8 \times 4 + 6 \times 3 + 1 \times 2 = 52$
- $x_4 : 6 \times 2 + 9 \times 1 = 21$

resultando na seguinte preferência social: $x_3 \succeq x_1 \succeq x_2 \succeq x_4$. Com uma ligeira mudança na preferência do Indivíduo 1, que resulta na Tabela 3, a contagem de Borda fica:

- $x_1 : 7 \times 4 + 8 \times 3 = 52$
- $x_2 : 1 \times 3 + 8 \times 2 + 6 \times 1 = 25$
- $x_3 : 8 \times 4 + 6 \times 3 + 1 \times 1 = 51$
- $x_4 : 7 \times 2 + 8 \times 1 = 22$

e essa nova configuração leva à preferência social $x_1 \succeq x_3 \succeq x_2 \succeq x_4$. Apesar de diferença de ordenamento de preferência de apenas um indivíduo e, ainda, da preferência entre x_1 e x_3 para o mesmo indivíduo não se alterar, a preferência social final tem a ordem entre x_1 e x_3 inversa. Isso mostra a violação da condição de Independência de Alternativas Irrelevantes, pois a mudança no ordenamento entre x_1 e x_2 e entre x_4 e x_3 causou a mudança no ordenamento entre x_1 e x_3 para toda a sociedade.

Segundo List (2013), esse tipo de violação é comum nos sistemas de votação do mundo real, fazendo com que regras de agregação de preferência tornem-se potencialmente vulneráveis à manipulação, como será mostrado na seção 2.5.

Tabela 3 – Perfil de ordenamentos de preferências individuais ligeiramente modificado.

	Indivíduo 1	Indivíduos 2 a 7	Indivíduos 8 a 15
1ª preferência	x_1	x_1	x_3
2ª preferência	x_2	x_3	x_1
3ª preferência	x_4	x_4	x_2
4ª preferência	x_3	x_2	x_4

Fonte: Adaptado de [List \(2013\)](#).

2.5 O Teorema de Gibbard-Satterthwaite

[Gibbard \(1973\)](#) e [Satterthwaite \(1975\)](#) provaram, independentemente, que “qualquer sistema de votação não-ditatorial, com pelo menos três alternativas, está sujeito à manipulação”. Apesar de usar o termo sistema de votação, o teorema é aplicado à qualquer esquema que faça uma escolha social a partir das preferências individuais relativas às alternativas disponíveis ([GIBBARD, 1973](#)).

Seja um indivíduo com a preferência ($x_1 \succ x_2 \succ x_3$) e a preferência social ($x_3 \succ x_2 \succ x_1$), sendo a alternativa vencedora x_3 . A manipulação da escolha se deve ao fato desse indivíduo poder apresentar suas preferências de forma desonesta a fim de que uma alternativa com menor preferência vença, sendo um cenário melhor para esse indivíduo. Neste caso, se o ordenamento de preferências individuais ($x_2 \succ x_1 \succ x_3$) torna a preferência social ($x_2 \succ x_3 \succ x_1$), com a alternativa x_2 vencedora, então o indivíduo pode manipular o resultado final para um resultado individual mais vantajoso ou menos prejudicial.

A principal diferença entre o teorema de Arrow e o teorema de Gibbard-Satterthwaite é o resultado final da agregação de preferências. Enquanto o primeiro trabalho com funções de bem-estar social e ordenamentos completos sobre as alternativas, o segundo trabalha com funções de escolha social tendo como saída uma alternativa vencedora. Essa diferença também é apresentada por [Gibbard \(1973\)](#).

Apesar disso, os teoremas são paralelos e, guardadas as proporções, equivalentes. [Gartner \(2009, p. 81\)](#) afirma que as condições colocadas para o teorema de Gibbard-Satterthwaite são correspondentes àquelas do teorema de Arrow, adaptadas às funções de escolha social. [Reny \(2001\)](#) propõe uma prova unificada para ambos os teoremas, reforçando ainda mais a conexão entre os mesmos.

As notações utilizadas serão similares às já apresentadas na seção 2.2.1. Seja X o conjunto finito de alternativas e n o número de indivíduos. O conjunto de ordenamentos de preferências racionais é denotado por R . Como mostrado na Definição 2.1, a função de escolha social é representada por $C : R^n \mapsto X$. As condições para a existência de uma função de escolha social são apresentadas a seguir e são análogas às condições Princípio de Pareto, Independência de Alternativas Irrelevantes e Não-ditadura para funções de bem-estar social ([SHOHAM; LEYTON-BROWN, 2008, p. 263-264](#)).

Condição 2.8 (Eficiência de Pareto²⁰). A FES é Pareto-eficiente se para todo indivíduo j da sociedade a alternativa $x_1 \in X$ é classificada como a de maior preferência e isso implica em $C = x_1$.

Assim como o Princípio de Pareto apresentado para a FBS, a condição previne a FES de escolher uma alternativa dominada por outra. Ou seja, a FES deve respeitar a unanimidade entre os membros da sociedade.

Condição 2.9 (Monotonicidade). A FES é Monotônica²⁰ se:

- o ordenamento de preferências individuais \succeq sempre implicar $C(\succeq_1, \dots, \succeq_k) = x_1$, e
- para todo o indivíduo j e toda a alternativa x_2 o ordenamento \succeq'_j ter $x_1 \succ x_2$ se \succeq_j assim o for,

então, $C(\succeq'_1, \dots, \succeq'_k) = x_1$.

A condição 2.9 é equivalente à condição 2.5, entretanto, mantendo a posição da alternativa ganhadora x_n sobre todos os conjuntos de ordenamentos de preferência, em vez de manter um ordenamento de preferências completo.

Condição 2.10 (Não-ditadura). A FES é não-ditatorial²⁰ se não há indivíduo j tal que $C(\succeq_1, \dots, \succeq_k) = x_1$ se e somente se \succeq_j classifica a alternativa x_1 como a de maior preferência.

Seguindo as condições apresentadas, o Teorema de Gibbard-Satterthwaite pode ser enunciado:

Teorema 2.4 (Teorema da Impossibilidade de Gibbard-Satterthwaite). Se há, ao menos, três alternativas, e a FES $C : R^n \mapsto X$ é Pareto-eficiente e Monotônica, então C é ditatorial.

A prova do teorema pode ser encontrado em diversas fontes na literatura, sendo a proposta por [Reny \(2001\)](#) simples, além de oferecer o paralelo com o Teorema de Arrow.

Os teoremas e as condições aqui apresentadas serão examinadas no capítulo 4 de forma a estabelecer funções análogas às apresentadas para o domínio da alocação de tarefas em sistemas multirrobo e, posteriormente, no capítulo 5, as arquiteturas de MRTA serão analisadas sob a ótica da Escolha Social.

2.6 Aplicações do Teorema da Impossibilidade de Arrow

A abordagem axiomática do Teorema de Arrow possibilita a análise de vários problemas pelo mesmo ponto de vista: requisitos simples são colocados a fim de que um mecanismo

²⁰ Cf. [\(RENY, 2001\)](#)

decisor reflita a escolha do grupo, entretanto, quando combinados, os mesmos requisitos limitam a solução. Áreas como Sistemas de Recomendação à usuários, *Data Mining*, Robótica, Engenharia de Projeto e decisão em ambientes multiagente estão entre aquelas que possibilitam a análise utilizando os preceitos definidos por Arrow (1951).

Pennock et al. (2000) utiliza a abordagem axiomática da teoria da escolha social para a implementação de Filtragem Colaborativa em Sistemas de Recomendação. Um mecanismo de filtragem colaborativa tem o objetivo de prever preferências de um usuário, chamado usuário ativo, tendo como base as preferências de um grupo de usuários. “A ideia chave é que o usuário ativo preferirá itens que pessoas de opiniões parecidas preferem, ou mesmo o que pessoas [com opiniões] diferentes não preferem” (PENNOCK et al., 2000).

A construção do filtro inicia-se pelo estabelecimento das características ideais que o algoritmo deve possuir e o trabalho prova que, sob todas as condições, apenas uma estratégia de previsão é possível: as preferências do usuário ativo são determinadas com base nas preferências de apenas um outro usuário, chamado de “vizinho mais próximo”. Esse resultado vai ao encontro do obtido por Arrow, em que o vizinho mais próximo faz o papel do ditador. Além disso, as condições estabelecidas são similares às estabelecidas no Teorema da Impossibilidade de Arrow: *Domínio Universal e funcionalidade mínima*, que é similar à condição de Domínio Irrestrito; *Unanimidade*, também chamada de Princípio de Pareto ou, em inglês, *Weak Pareto*; *Independência a Alternativas Irrelevantes*, na qual as alternativas são as classificações de usuários sobre itens, como músicas, filmes, produtos de uma loja virtual; e *Invariância de escala*, princípio análogo à condição de Monotonicidade.

Kleinberg (2003) propõe uma análise axiomática para o problema de agrupamento de dados, também conhecido como *clustering*. Essa análise leva à impossibilidade do agrupamento de dados por uma função de agrupamento com base em três condições simples: *invariância de escala*, a qual estabelece que a função não deve ser sensível à variações de escala se as distâncias relativas entre os pontos ou vetores de atributos permanecem as mesmas; *riqueza*, que requer que qualquer subconjunto de pontos pode ser um conjunto de saída; e *consistência*, condição que se refere à produção de uma mesma partição de dados com distâncias relativas entre *cluster* aumentadas, mas distância relativa entre os pontos de um *cluster* diminuída.

Com isso, para um número de pontos maior que dois, não há função de agrupamento que respeite às três condições simultaneamente. Segundo o autor, o resultado negativo “indica uma série de compensações que são inerentes ao problema de agrupamento de dados” (KLEINBERG, 2003). Essas compensações são as relaxações feitas nas condições previamente estabelecidas para que haja uma função de agrupamento.

Uma estrutura de decisão coletiva baseada no Bem-estar Social aplicada para sociedade de robôs é apresentada por Stirling e Nokleby (2009). A estrutura proposta se baseia em dois conceitos de otimização: *satisficing* e condicionamento. O termo *satisficing* foi cunhado por Simon (1955 apud STIRLING; NOKLEBY, 2009) e significa “satisfazer e ser suficiente”, sendo uma combinação das palavras, em inglês, *satisfy* e *suffice*. Essa heurística é aplicada de forma a buscar uma decisão boa o suficiente, já que atingir o ótimo pode demandar capacidade com-

putacional elevada. Dessa forma, uma decisão satisfaz e é suficiente se a utilidade calculada é maior que um valor limiar de utilidade pré-definido (STIRLING; NOKLEBY, 2009, p. 57). O conceito de condicionamento estabelece uma relação entre a ação atual e a ação anterior, criando um laço de utilidade condicional.

O trabalho discute ainda a complexidade adicionada à teoria da escolha social com a definição dos conceitos acima, além dos impactos nos axiomas originais. Os autores afirmam que “na presença de preferências condicionais, [...] os axiomas tradicionais da teoria da escolha social, tal como a Independência a Alternativas Irrelevantes, tornam-se problemáticos” (STIRLING; NOKLEBY, 2009, p. 59).

Dentre as diversas áreas de aplicação possíveis, em (JACOBS; POEL; OSSEWEIJER, 2014) o Teorema de Arrow é modificado de forma a agregar desempenho, e não preferências, na escolha de projetos conceituais de engenharia. A aplicação em Engenharia de Projeto (*Engineering Design*) do teorema é uma proposta de resolução para o problema de decisão multicritério. Para tal, as condições originais são modificadas para atender o objetivo de agregar desempenho. A condição IIA original é adaptada para *Independência de Conceitos Irrelevantes*, que determina que o desempenho de dois conceitos em uma determinada estrutura de critérios depende apenas de seus respectivos desempenhos, e não de um desempenho de um terceiro conceito. A condição Escopo Irrestrito (relacionada à condição UD) estabelece que a estrutura de critérios é transitiva e completa. O princípio de Pareto, neste caso, define que, se um projeto conceito é estritamente melhor que outro em todos os critérios de desempenho, então, esta deve ser a estrutura de critérios global para esses dois projetos. E, como no Teorema de Arrow, não deve haver dominância, de forma que o desempenho global não deve ser determinado por uma única estrutura de critérios. A partir das condições vistas anteriormente, Jacobs, Poel e Osseweijer (2014, p. 6), para a impossibilidade Arrowiana para a agregação de desempenho em Engenharia de Projetos, afirmam que “se houver um número finito de critérios de avaliação e houver, pelo menos, três projetos conceitos alternativos, nenhum método de agregação pode, simultaneamente, satisfazer as quatro condições estabelecidas”.

Outros trabalhos relacionando a Teoria da Escolha Social, incluindo os princípios do bem-estar social, e estruturas de decisão em sistemas multiagente podem ser encontrados na literatura, como Roos e Rothe (2010). Meyer, Ghose e Chopra (2001) investigam a ligação entre a fusão de crenças baseada no contexto e as operações de agregação de preferências estudadas por Arrow (1951). Os trabalhos de Endriss e Maudet (2003) e de Chevaleyre et al. (2006) discutem os resultados da aplicação de funções de bem-estar social utilitaristas, em que o objetivo é maximizar a utilidade global somando as utilidades de cada agente, e de funções de bem-estar social igualitaristas, com a meta de maximizar a utilidade do agente que tem a menor utilidade, para o problema de alocação de recursos em sistemas multiagente (MARA - *Multi Agent Resource Allocation*). A aplicação em sistemas multirrobô pode ser visto em nos trabalhos de Kim, Kim e Lee (2012) e Kim, Baik e Lee (2015), utilizando o conceito de igualitarismo, da Teoria da Utilidade, para minimizar o consumo de recursos energéticos e maximizar a taxa de conclusão de tarefas.

2.7 Escolha Social Computacional

Mesmo como uma área de pesquisa relativamente recente, a Escolha Social Computacional tem seus precursores, principalmente, David Gale e Lloyd Shapley com o algoritmo para o problema de atribuição ótima (OAP - *Optimal Assignment Problem*), de 1962. Outros trabalhos são os de John Bartholdi, Craig Tovey e Michael Trick, com uma série de publicações discutindo a complexidade computacional como barreira para a manipulação estratégica em eleições, e de Bernard Monjardet e Olivier Hudry, estudando a complexidade computacional de processamento de agregação de preferências, ambos no fim da década de 1980 (BRANDT et al., 2016).

A pesquisa em escolha social computacional tem duas linhas de pesquisa principais. A primeira envolve aplicar técnicas computacionais, como inteligência artificial (IA), por exemplo, para obtenção de melhores análises de mecanismos de escolha social (CHEVALEYRE et al., 2007).

Chevaleyre et al. (2007) cita exemplos para essa aplicação, como a análise de protocolos de votação, das formas de especificação e verificação de processos sociais (como o da divisão justa, do inglês *fair division*) e a utilização de técnicas de IA e lógica para a representação compacta de preferências em domínios combinatoriais (como mecanismo de alocação de tarefas por meio de leilões, do inglês *auctions*).

A segunda linha se preocupa com aplicações na ordem inversa, ou seja, aplicações de conceitos da Teoria da Escolha Social para a resolução de problemas da ciência da computação e de IA (BRANDT et al., 2016; CHEVALEYRE et al., 2007). Brandt et al. (2016) sugere que a teoria da escolha social fornece ferramentas para que decisões conjuntas possam emergir de sistemas multiagentes povoados de agentes heterogêneos, possivelmente egoístas. Uma lista de tópicos interessantes à pesquisa em escolha social computacional é apresentada em (CHEVALEYRE et al., 2007):

- *Agregação de preferências*
- *Teoria da votação*
- *Alocação de recursos e divisão justa*
- *Formação de coalizões*
- *Agregação de julgamento e Fusão de crenças*
- *Sistemas de classificação ou ranking*

Para Endriss (2014), as três estruturas fundamentais à sistemas multiagente são: a alocação ou divisão justa de recursos, sistemas de votação e agregação de julgamentos. Outro ponto importante, citado por Aziz (2016), é a formação de coalizões em sistemas multiagente, principalmente em situações de preferências incertas. A incerteza das preferências é proveniente, principalmente, da limitação na comunicação e nas informações.

Portanto, a escolha social computacional é um campo de extensas possibilidades. Neste trabalho, o foco será na aplicação das limitações impostas pelo Teorema da Impossibilidade de Arrow na análise do problema de alocação de tarefas em sistemas multirrobo.

2.7.1 O problema da alocação de objetos

Brandt et al. (2016) modela um dos problemas clássicos do estudo da Teoria da Alocação de Recursos: o problema da alocação de objetos. Para Chevalleyre et al. (2006) o problema da alocação de tarefas pode ser reduzido à um problema de alocação de recursos. A limitação para tal consideração depende do grau de acoplamento entre as tarefas, ou seja, qual a dependência de execução de uma tarefa com base na situação das demais. Usando esse modelo como base, será definido o modelo social para o problema MRTA.

Segundo Hurwicz (1973), é natural que se exija que um mecanismo de alocação guie os agentes à situações, pelo menos, praticáveis, mesmo que haja dificuldade em se atingir tal resultado. Ainda segundo o autor, a viabilidade não é a única característica necessária, mas também a eficiência, ou otimalidade (introduzido pelo Princípio de Pareto), ou ainda, a maximização de uma função de bem-estar social (como definido por Arrow (1951)). Do ponto de vista de Hurwicz (1973), “esses diferentes atributos tem uma importante característica em comum: eles são independentes do mecanismo. Um critério de otimalidade que pressupõe um mecanismo particular não atende como um critério legítimo de comparação entre outros mecanismos.”

Seja uma dotação social O de bens, ou objetos, indivisíveis disponíveis para alocação a uma sociedade de k agentes. Cada agente $j \in N$, com $j = (1, \dots, k)$, pode consumir apenas um objeto e tem um ordenamento de preferências \succeq_j sobre O . Ainda, os objetos são os únicos recursos (BRANDT et al., 2016).

De forma mais geral, Brandt et al. (2016) definem que se cada agente $j \in N$ pode consumir mais de um objeto, então o conjunto $R_{[\succeq]} = \{\succeq_1, \dots, \succeq_k\}$ de preferências é definido sobre 2^O . Essa configuração pode ser classificada como uma alocação de recurso multiagente ordinal (*ordinal-MARA*) (BRANDT et al., 2016). A versão *cardinal* do problema substitui o conjunto de ordenamentos de preferências $R_{[\succeq]}$ por um conjunto $U = \{u_1, \dots, u_k\}$ de utilidades escalares.

Claramente, o caso mais simples é aquele em que o número de agentes $|N|$ é igual ao número de objetos $|O|$. Caso o número de objetos seja menor que o número de agentes, são adicionados tantos *objetos nulos* ao conjunto O quanto o necessário para igualar as quantidades para que cada agente receba ao menos um objeto, real ou nulo. Um objeto é *desejado* se o mesmo é preferido a um objeto nulo. Por outro lado, um objeto é *indesejado* se for preterido a um objeto nulo.

Um exemplo do problema é a divisão de tarefas domésticas (BRANDT et al., 2016). Podem haver tarefas (objetos) que não são desejadas, mas ainda assim devem ser executadas. Por exemplo, há uma preferência a um objeto nulo à lavar o banheiro. Portanto, as tarefas domésticas serão alocadas primeiro em relação aos objetos nulos do conjunto. A solução de tal problema é a alocação apropriada dos recursos aos agentes levando em consideração as pre-

preferências individuais de cada agente. Essa solução pode ser atingida de diferentes maneiras. [Brandt et al. \(2016\)](#) destaca: (i) regras baseadas na igualdade, proporcionalidade, prioridade, loterias e no mercado, destacando a importância na observação do mundo real; (ii) a análise axiomática, quando a definição de princípios de bom comportamento da solução são estabelecidos por meio de axiomas; (iii) a teoria dos jogos cooperativos; e (iv) a utilização de regras aplicadas a outras áreas de estudo.

Neste trabalho, será dada atenção à análise axiomática do problema. Mais especificamente, utilizando as mesmas premissas da análise de [Arrow \(1951\)](#). Apesar de ter intensa relação com a agregação de preferências em sistemas de votação, tal análise pode ser também utilizada para problemas de alocação de recursos e, por similaridade, à alocação de tarefas.

De forma análoga ao visto no problema da alocação de objetos, o problema MRTA será considerado como um problema MARA ordinal. Dessa forma, o conjunto de n tarefas disponíveis para alocação \mathcal{T} deve ser atribuído ao conjunto de m robôs \mathcal{R} . A análise axiomática proposta por Arrow será utilizada na verificação dos mecanismos de alocação utilizados nas arquiteturas analisadas.

3 Sistemas Multirrobo

Para um robô monolítico, ou seja, que dispõe de um número variado de ferramentas a fim de possibilitar a aplicação em diversas tarefas diferentes, pode ser muito complexo completar uma série de tarefas em um determinado intervalo de tempo, levando a uma redução de desempenho do sistema como um todo.

Este capítulo apresenta um estudo em SMR. Serão apontadas duas taxonomias, ou classificações, de SMR e suas possíveis tipologias. A primeira é baseada na taxonomia proposta por [Iocchi, Nardi e Salerno \(2000\)](#) e apresenta uma classificação de SMR em dois grupos de dimensões a fim de melhor descrever a estrutura desses sistemas e os aspectos envolvidos no projeto dos mesmos. As nomenclaturas e definições propostas inicialmente foram adaptadas de acordo com a literatura atual. A segunda, proposta por [Gerkey e Matarić \(2004\)](#), é uma classificação mais específica acerca da alocação de tarefas em SMR, caracterizando, assim, o problema de alocação de tarefas em sistemas multirrobo, mais conhecido pela sigla, em inglês, MRTA - *Multi-Robot Task Allocation*. Além disso, será apresentada brevemente a extensão da taxonomia de Gerkey, proposta por [Korsah, Stentz e Dias \(2013\)](#). Ainda neste capítulo, serão apresentadas as arquiteturas estudadas, destacando-se suas principais características e as classificando de acordo com as taxonomias propostas. Posteriormente, as mesmas serão analisadas do ponto de vista do Teorema da Impossibilidade de Arrow.

3.1 Taxonomia para Sistemas Multirrobo

Sistemas multirrobo diferem de outros sistemas multiagente (SMA) ou, do inglês *Multi Agent Systems* - MAS, por conta de sua concretização, em um equipamento de *hardware*, e atuação implícita em um ambiente real. Por consequência, necessitam de modelos mais complexos que os sistemas no domínio de *software*, como bancos de dados e redes de comunicação ([CAO et al., 1995](#)).

Para este trabalho, é importante definir quais sistemas serão contemplados quando citado um SMR, uma vez que um sistema robótico pode significar desde um único sensor, realizando a aquisição e o processamento de dados de um determinado local, até máquinas humanoides complexas que interagem com o ambiente de formas complexas ([FARINELLI; IOCCHI; NARDI, 2004](#)). Dessa forma, define-se SMR como um conjunto de sistemas robóticos móveis, operando no mesmo ambiente, equipados com sensores e atuadores, que se comunicam entre si e/ou com uma estação central e cooperam para a execução de uma tarefa ou missão (conjunto de tarefas).

Uma taxonomia para sistemas de coordenação de SMR foi proposta por Iocchi, Nardi e Salerno (2000) e revisada por Farinelli, Iocchi e Nardi (2004). Esta proposta compila e esquematiza, principalmente, as definições propostas por Cao et al. (1995) e Dudek et al. (1996), além de levar outros trabalhos em consideração, para classificações de SMR. No presente trabalho, a mesma foi adaptada de forma a explorar e exemplificar as características desses sistemas, focando na classificação de arquiteturas de cooperação entre os robôs. Na adaptação dos termos e definições utilizados na taxonomia original, buscou-se uma revisão da literatura atual sobre o assunto. A taxonomia se divide em dois eixos ou grupos de dimensões: *dimensões de coordenação* e *dimensões de sistema*.

Basicamente, o eixo de dimensões de coordenação tem o intuito de definir as características de coordenação que um SMR pode atingir, sendo dividido em quatro níveis: *Cooperação*, *Conhecimento*, *Coordenação* e *Organização* (FARINELLI; IOCCHI; NARDI, 2004). O eixo de dimensões de sistema apresenta as características do sistema que influenciam no projeto do SMR. Os eixos definidos são ortogonais, de forma que as características do sistema tem impacto na coordenação do SMR, independentemente das características de coordenação. São definidas quatro dimensões de sistema: *Comunicação*, *Composição da Equipe*, *Arquitetura do Sistema* e *Tamanho da Equipe*. A estrutura apresentada pode ser observada na Figura 3.1.

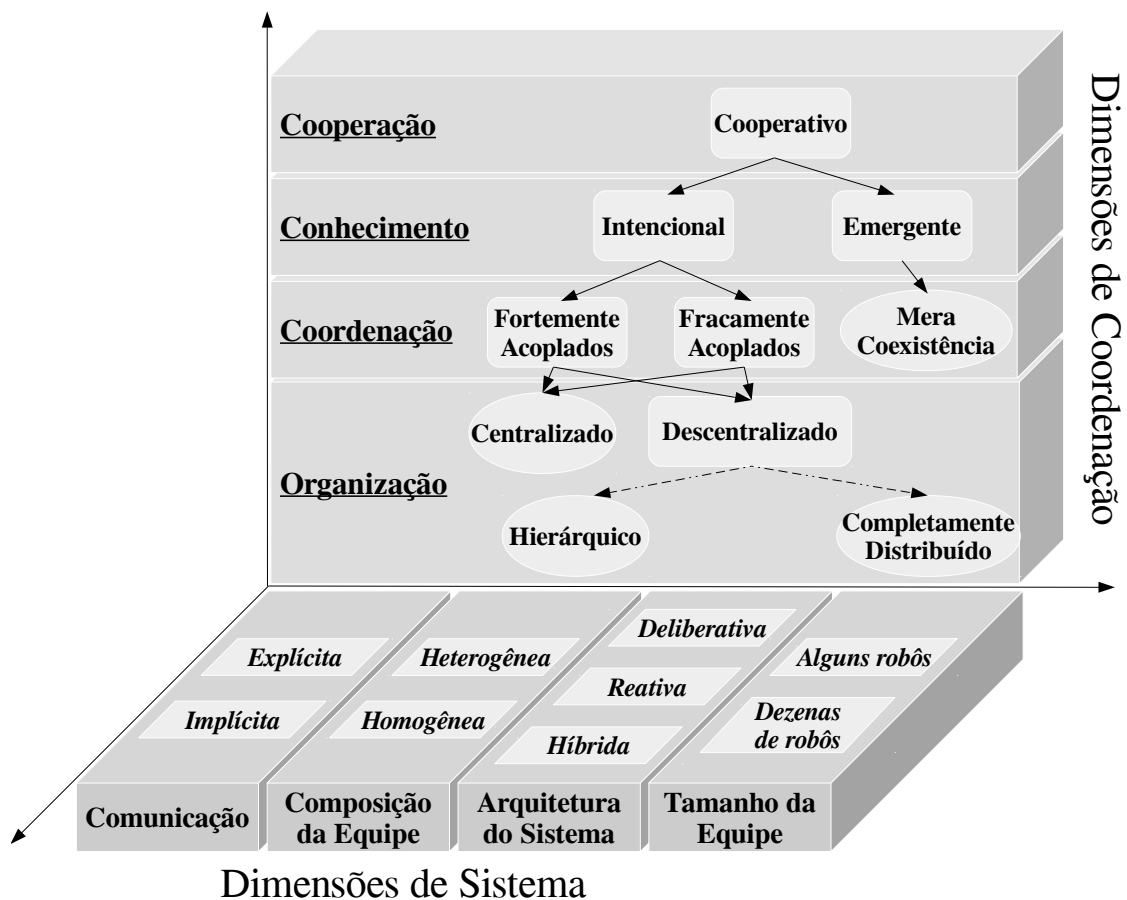


Figura 3.1 – Estrutura em eixos da taxonomia de coordenação de sistemas multirrobo proposta em (FARINELLI; IOCCHI; NARDI, 2004) com termos e definições adaptados. Fonte: Adaptado de Farinelli, Iocchi e Nardi (2004).

A dimensão de *Cooperação* é a camada que classifica a capacidade de cooperação do SMR. Segundo [Farinelli, Iocchi e Nardi \(2004\)](#), essa camada distingue os sistemas cooperativos dos não-cooperativos. Sendo que sistemas cooperativos são aqueles em que os robôs operam em conjunto para executar uma tarefa global ([FARINELLI; IOCCHI; NARDI, 2004](#)). Neste trabalho, é dado foco aos sistemas cooperativos. Sistemas não-cooperativos podem ser entendidos como sistemas compostos por agentes egoístas. O que não significa causar mal aos demais agentes. Apenas não se tem conhecimento do estado dos demais agentes do sistema ([SHOHAM; LEYTON-BROWN, 2008](#)) e isso não é levado em conta na atuação de cada robô.

Sendo o sistema cooperativo, os robôs envolvidos podem ter ou não ciência da presença dos demais membros da equipe, o que é definido na dimensão de *Conhecimento*. Pode-se dividir, então, a cooperação em intencional e emergente.

Na *cooperação intencional*, os robôs são conscientes da existência de seus colegas de equipe, de suas ações e estados ([PARKER, 2008](#)). A equipe coopera de forma intencional para atingir um nível de eficiência e completar a missão e, de acordo com [Gerkey e Mataric \(2004\)](#), normalmente, através de comunicação relacionada à tarefa ou à negociação. Outra característica importante é o fato desse tipo de cooperação trabalhar com uma equipe com número limitado de membros, tipicamente de robôs heterogêneos, desempenhando várias tarefas também distintas ([PARKER, 1998](#)).

A cooperação intencional pode ser cruamente dividida entre abordagens baseadas em comportamento, do inglês *behavior-based*, e baseadas em mercado, do inglês *market-based*. O trabalho de [Yan, Jouandeau e Cherif \(2013\)](#) apresenta uma diferenciação entre as abordagens baseadas em mercado. Neste trabalho, as abordagens são divididas em três tipos: baseadas em mercado, baseadas em leilões (*auction-based*) e baseadas em negócios (*trade-based*). Entretanto, as diferenças entre as classificações são bastante sutis. Por exemplo, abordagens baseadas em mercado calculam a oferta para a execução da tarefa levando em consideração o custo e os benefícios, enquanto abordagens baseadas em leilões utilizam apenas o custo estimado ([YAN; JOUANDEAU; CHERIF, 2013](#)). Entretanto, sob vários aspectos as três classificações são semelhantes, como, por exemplo, do ponto de vista da complexidade computacional. Por esse motivo, neste trabalho tais abordagens serão consideradas no mesmo grupo. Além das já discutidas, existem outras classificações possíveis para abordagens de alocação de tarefas em SMR, mas não é o objetivo cobrir, aqui, todo o assunto.

Em abordagens baseadas em comportamento, a alocação de tarefas ocorre, comumente, sem discussões explícitas entre o time de robôs ([PARKER, 2008](#)). Nessas arquiteturas, exemplos de cooperação completamente distribuída, os robôs utilizam do conhecimento do ambiente (a partir da leitura dos sensores), ações e o estado atual dos membros da equipe, capacidades previamente conhecidas dos membros da equipe e status da missão, além do conhecimento de parâmetros internos, para determinar qual robô deve executar determinada tarefa.

Diferentemente do visto em arquiteturas baseadas em comportamento, abordagens baseadas em mercado envolvem uma comunicação de duas vias entre os robôs, já que a alocação de tarefas requer uma negociação prévia ([PARKER, 2008](#)). Baseada na Teoria de Mercado da

economia, os robôs envolvidos na missão buscam otimizar a utilidade global baseando-se nas utilidades individuais que os robôs ofertam para desempenhar uma tarefa, tipicamente, utilizando um algoritmo guloso.

Por sua vez, na *cooperação emergente*, os robôs não têm ciência da presença ou da quantidade de membros na equipe. Apesar de apresentar um comportamento cooperativo, os indivíduos respondem apenas a estímulos de outros indivíduos ou do ambiente e não há conhecimento dos estados dos demais membros do enxame (GERKEY; MATARIĆ, 2004). Também chamada de robótica de enxame, em tradução livre a *swarm robotics*, lida com um grande número de robôs homogêneos e com limitações profundas em suas capacidades. Essa abordagem é adequada a aplicações em que o tempo de execução não é um ponto crítico e as tarefas executadas são amplamente repetidas em áreas relativamente grandes, como a limpeza de estações (PARKER, 1998). Todos os robôs do enxame têm as mesmas leis de controle que, usualmente, são baseadas em comunidades cooperativas biológicas.

A dimensão de *Coordenação* é a camada em que são classificados os mecanismos utilizados para a coordenação dos robôs no SMR. Farinelli, Iocchi e Nardi (2004) descrevem coordenação como cooperação em que os membros da equipe levam em consideração as ações dos demais, de forma a potencializar a operação, executando-a de forma coerente. Nessa dimensão são definidas as regras de interação dos robôs entre si e com o ambiente, um protocolo de coordenação.

O comportamento emergente visto na dimensão superior leva à uma coordenação chamada *mera coexistência*, quando, de acordo com Matarić (2014), mesmo trabalhando em uma mesma tarefa em um ambiente compartilhado, os robôs podem não se comunicar ou, até mesmo, nem reconhecer uns aos outros, tratando-se como obstáculos.

A cooperação intencional pode-se subdividir os sistemas em fracamente acoplados e fortemente acoplados. Segundo Matarić (2014), na abordagem *fracamente acoplada* “os robôs se reconhecem como membros de um grupo e podem até usar uma coordenação simples, como se afastar um do outro para dar mais espaço e minimizar a interferência”. Entretanto, do ponto de vista da tarefa, não há interdependência entre os robôs, ou seja, os membros do grupo não dependem uns dos outros para a execução da tarefa. Como vantagem, esse tipo de abordagem apresenta a robustez, devido a redundância e a rejeição a falhas dos indivíduos. Porém, apresentam uma coordenação mais difícil na execução de tarefas específicas (MATARIĆ, 2014).

Abordagens *fortemente acopladas* apresentam a cooperação entre robôs em tarefas específicas. De acordo com Matarić (2014), “geralmente, por meio de comunicação, em turnos, e outros meios de coordenação forte. Eles dependem uns dos outros, o que dá ao sistema um desempenho melhor do grupo”. No entanto, essa dependência diminui a robustez, já que a perda de um membro da equipe leva à diminuição do desempenho do sistema.

O quarto nível é definido como dimensão de *Organização*. Nela, são caracterizadas as formas de tomada de decisão das equipes de robôs em um SMR. Essa camada apresenta uma distinção entre duas formas de coordenação, a abordagem centralizada e a descentralizada (FARINELLI; IOCCHI; NARDI, 2004; CAO et al., 1995). Antes de defini-las, é importante

ressaltar a relação entre a terceira e a quarta dimensões, mais especificamente, entre os tipos de coordenação fortemente e fracamente acoplados com as abordagens de organização. Como são caracterizadas pela relação entre os robôs e as dependências das tarefas, ambas as formas de coordenação podem derivar sistemas organizados de forma centralizada ou descentralizada. Isso acarreta apenas a mudança do mecanismo de decisão.

Um sistema centralizado é caracterizado por um agente de controle único que observa todo o ambiente e tem acesso a toda informação obtida do sistema. Esse agente toma decisões para maximizar a utilidade, ou eficiência, global e é capaz de tomar decisões ótimas por ter a informação do ambiente como um todo, dentro de alguns limites de tamanho do ambiente em questão. Porém, essa estação central necessita de uma comunicação robusta e permanente com cada robô do sistema. E isso é também um gargalo, significando que não pode haver falhas centrais para que a missão seja cumprida com sucesso pela equipe de robôs.

Um sistema descentralizado não necessita de um único agente de controle e tais sistemas podem ser categorizados em hierárquicos ou distribuídos (CAO et al., 1995). Em arquiteturas hierárquicas, usualmente utilizadas em aplicações com um número grande de robôs no sistema, cada grupo define um dos robôs como o líder local, centralizando nessa figura as decisões do grupo. Ao passo que, em arquiteturas descentralizadas, todos os robôs são iguais em relação à tomada de decisões, mesmo quando são heterogêneos.

Os sistemas distribuídos possuem ainda uma subcategoria, as arquiteturas completamente descentralizadas, ou seja, nenhum robô é individualmente responsável pelo controle dos demais (PARKER, 1995).

Os dois últimos conceitos apresentados, sistemas distribuídos e sistemas completamente distribuídos, diferem, especificamente, na comunicação entre os robôs e na tomada de decisão. Em uma arquitetura completamente distribuída, os robôs se comunicam entre si por meio de comunicação por difusão (*broadcasting*), não há uma comunicação de duas vias estabelecida. Isso significa que mesmo que um robô utilize as informações dos demais membros da equipe, este não depende da aprovação dos demais para tomar uma decisão, por exemplo, iniciar a execução de uma tarefa.

Em relação às arquiteturas descentralizadas, os robôs necessitam de uma comunicação explícita de duas vias, estabelecida quando a decisão, por exemplo, a alocação de tarefas, envolve uma negociação (PARKER, 2008). Como resultado, um dos robôs assume o controle do poder de decisão para o processo de alocação de uma tarefa e torna-se responsável por todas as mensagens relacionadas a esta tarefa.

Arquiteturas descentralizadas têm muitas vantagens em relação às centralizadas. Um sistema descentralizado requer uma comunicação local entre os robôs e, se necessário, uma comunicação intermitente com uma estação central. Além disso, a descentralização confere ao sistema tolerância a falhas, redundância, confiabilidade e escalabilidade. Em arquiteturas centralizadas, atingir os últimos dois tópicos é um problema crítico.

Em relação às classificações anteriores, existem características inerentes a todos os sistemas multirrobo cooperativos relevantes ao projeto de tais sistemas. Por isso, foram agrupadas

quatro características em um eixo transversal ao eixo de coordenação, como mostrado na Figura 3.1, de forma a representar a interseção entre os eixos.

Em uma equipe, geralmente, há a necessidade de comunicação entre os membros, por isso o primeiro dos quatro aspectos a ser discutido é a *Comunicação*. Segundo [Matarić \(2014\)](#), existem boas razões para que os robôs em um SMR se comuniquem, entre elas:

- *melhora na percepção do ambiente*: usando a troca de informações como meio de reconhecimento indireto do ambiente;
- *sincronização das ações*: como os robôs não percebem instantaneamente os demais membros da equipe, a comunicação é importante para sinalização das ações que fizeram, não fizeram ou deixaram de fazer; e
- *habilitação da coordenação e negociação*: apesar de não ser necessária para exprimir um comportamento coordenado, a comunicação aumenta a capacidade dos robôs de cooperar e negociar, a fim de tomar decisões e realizar tarefas corretamente.

Os tipos de comunicação podem ser divididos em *comunicação explícita* e *comunicação implícita*. A comunicação explícita, também chamada de intencional ([MATARIĆ, 2014](#)), ocorre quando o robô age de forma proposital no envio de uma mensagem, seja ela por difusão (*broadcast*), por comunicação ponto a ponto (*peer-to-peer*) ou por comunicação publicar-inscrever (*publisher-subscriber*), muito parecida com uma lista de *e-mails* ou um fórum de discussões *online*. A comunicação implícita é conhecida como aquela em que o agente modifica o ambiente para passar a mensagem, não utilizando nenhum canal de comunicação explícito. Esse tipo de comunicação é conhecido também por estigmergia. Um exemplo prático muito comum é a trilha de feromônios que uma formiga deixa para que as demais saibam o caminho ([MATARIĆ, 2014](#)). Na literatura, pode-se encontrar a classificação dividida em comunicação direta e indireta ([IOCCHI; NARDI; SALERNO, 2000](#)), tendo a mesma definição da nomenclatura vista anteriormente.

Outro aspecto importante de um SMR é a *Composição da Equipe*, que está dividida principalmente em equipes *heterogêneas* e *homogêneas*. As equipes heterogêneas são compostas por robôs que diferem uns dos outros em suas características de *hardware* e/ou *software*. Por exemplo, robôs com diferentes sensores e atuadores trabalhando juntos. Mesmo que os robôs tenham as mesmas características físicas, se suas leis de controle ou funções forem diferentes e não intercambiáveis, podem ser considerados robôs heterogêneos do ponto de vista de um SMR ([MATARIĆ, 2014](#)). Já as equipes homogêneas são compostas por robôs que dispõem exatamente do mesmo *hardware* e *software*.

A *Arquitetura do Sistema* listada na taxonomia refere-se a uma arquitetura de controle de todo o sistema, não de um único robô ou agente. As arquiteturas podem ser divididas em três grupos: deliberativas, reativas e híbridas ([IOCCHI; NARDI; SALERNO, 2000](#); [FARINELLI; IOCCHI; NARDI, 2004](#); [MATARIĆ, 2014](#)) e têm uma forte relação com a tomada de decisão no

SMR. Um controle *deliberativo* é caracterizado pelo conceito clássico da malha SMPA (*Sense-Model-Plan-Act*). [Iocchi, Nardi e Salerno \(2000\)](#) definem, ainda, um controle deliberativo social, permitindo “ao sistema lidar com as mudanças do ambiente fornecendo uma estratégia que pode ser adotada para reorganizar todas as tarefas alocadas aos membros da equipe, a fim de que o uso dos recursos seja eficiente e o objetivo global seja atingido”. Como uma estratégia global deve ser proposta ao SMR, esse tipo de controle é mais adequado à sistemas centralizados. Assim, um robô ou agente reúne e trata todos os dados, processa o plano e o envia para os demais robôs ([MATARIĆ, 2014](#)).

Em oposição ao controle deliberativo, o controle *reativo* é aquele executado pelo próprio robô, que lida diretamente com as alterações do ambiente e encontra uma solução específica para reorganizar suas tarefas, visando completar o objetivo originalmente a ele atribuído. Nesse tipo de controle, nenhum plano global é traçado e a cooperação surge da interação entre os robôs da equipe, que podem se comunicar e cooperar com os demais quando necessário, segundo [Matarić \(2014\)](#). Ou seja, as decisões são tomadas de forma distribuída.

Para extrair as melhores características de ambas as arquiteturas de controle vistas, existe a arquitetura *híbrida*. Segundo [Matarić \(2014\)](#), mesmo sendo mais indicado para sistemas centralizados, essa arquitetura também pode ser utilizada em sistemas distribuídos. Basicamente, cada robô tem controle reativo das funções de baixo nível, como monitoramento de seus sensores. As decisões são tomadas em um nível mais alto (usualmente um planejador) que tem acesso às informações do sistema ([IOCCHI; NARDI; SALERNO, 2000](#)).

Por fim, e não menos importante, o *Tamanho da Equipe* é de grande relevância à arquitetura de controle, sendo um fator limitador. Quanto maior o número de robôs no mesmo ambiente, maior tende a ser a interferência, dificuldade de comunicação e mais elaborada deve ser a estratégia de controle ([FARINELLI; IOCCHI; NARDI, 2004](#)).

[Gerkey e Matarić \(2004\)](#) sugerem que um dos maiores desafios enfrentados em SMR é a definição eficiente da utilidade de uma ação em curso. Especialmente porque a maioria, se não todas, as abordagens de coordenação apoiam-se em alguma forma de utilidade, mesmo que com designações diferentes. Outrossim, de acordo com [Bastos, Ribeiro e Souza \(2008\)](#), a grande maioria de arquiteturas de alocação de tarefas em sistemas multirrobo utiliza um valor de utilidade não variável no tempo, embora no mundo real as tarefas tenham grau de prioridade e tempo de duração máxima. Diante disso, para melhores soluções, a modelagem do problema deve contemplar uma utilidade variável ou uma função de utilidade que traduza esses requisitos variáveis das tarefas.

3.2 Taxonomia de Arquiteturas de Alocação de Tarefas em SMR

Com o objetivo de formalizar a análise e o tratamento dos problemas de Alocação de Tarefas em sistemas multirrobo, [Gerkey e Matarić \(2004\)](#) descreveram uma taxonomia para

esses sistemas, sendo representada em três eixos, como mostrado na Figura 3.2. O eixo Tipo de Robô é dividido em duas categorias: robô de tarefa única (*Single task robot - ST*) e robô de tarefas múltiplas (*Multi task robot - MT*). Um robô ST é aquele que executa apenas uma tarefa por vez, ou seja, quando alocado para a execução de uma tarefa, não é capaz de realizar qualquer outra atividade simultaneamente. Já um robô MT é capaz de realizar várias tarefas de forma simultânea.

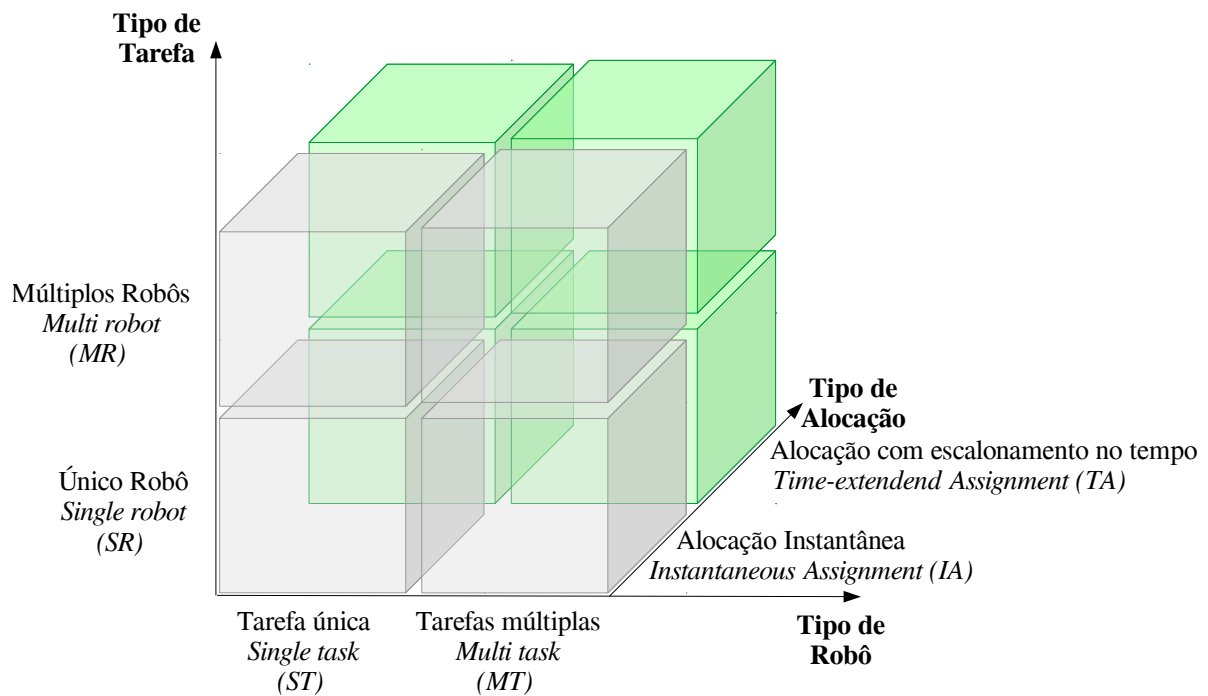


Figura 3.2 – Divisão em três eixos da taxonomia de arquiteturas de alocação de tarefas proposta por Gerkey e Mataric (2004). Fonte: Adaptado de Korsah, Stentz e Dias (2013).

A classificação do Tipo da Tarefa diz respeito ao número de robôs necessários para sua realização. É dividida em tarefas de único robô (*Single robot task - SR*) e tarefas de múltiplos robôs (*Multi robot task - MR*). Tarefas do tipo SR necessitam de apenas um robô para serem executadas, enquanto tarefas MR podem necessitar de mais de um robô para serem finalizadas.

O terceiro eixo da taxonomia refere-se ao Tipo de Alocação, como as tarefas são distribuídas aos membros da equipe ao longo do tempo. Essa alocação pode ser do tipo Alocação Instantânea (*Instantaneous Assignment - IA*), quando a alocação da tarefa implica o início da execução da mesma pelo agente determinado, ou do tipo Alocação escalonada no tempo (*Time-extended Assignment - TA*), quando as tarefas são agendadas ao longo da execução das demais. A principal diferença entre os tipos de alocação é a existência, ou não, de um planejamento. Na alocação IA, não há um planejamento futuro em relação às tarefas não alocadas, limitado, principalmente, pela falta de informações acerca do ambiente, dos robôs e das tarefas. Com mais informações disponíveis sobre o sistema, as tarefas podem ser alocadas conforme um planejamento das ações futuras. Dessa forma, mesmo que esteja executando determinada tarefa τ_1 , o robô pode já ter alocada uma tarefa τ_n para execução posterior.

Mesmo que o trabalho de [Gerkey e Mataric \(2004\)](#) descreva uma gama considerável de problemas, há limitações em relação às características das tarefas que foram consideradas, o que exclui uma coleção de problemas MRTA. Admitindo que as tarefas possuem utilidades independentes, a execução de uma tarefa não afeta a execução das demais. Entretanto, em aplicações reais, fica evidente que pode haver uma interdependência durante a execução das tarefas. Por isso, [Korsah, Stentz e Dias \(2013\)](#) expandiram a taxonomia proposta para casos mais complexos, chamando-a de *iTax*.

Essa proposta lida com utilidades inter-relacionadas e questões limitantes. O conceito chave da *iTax* é a decomposição das tarefas em quatro classes: tarefa elementar, ou atômica, tarefa simples, tarefa composta e tarefa complexa, como mostrado na parte superior da Figura 3.3. Além disso, classes de dependência entre as tarefas são estabelecidas: sem dependência, dependências de planejamento, dependências de planejamento cruzado e dependências complexas, conforme a parte inferior da Figura 3.3. Apesar da criação de novas classes, [Korsah, Stentz e Dias \(2013\)](#) adotam todos os demais conceitos propostos por [Gerkey e Mataric \(2004\)](#) para a classificação de tipos de tarefas, de robôs e de alocação.

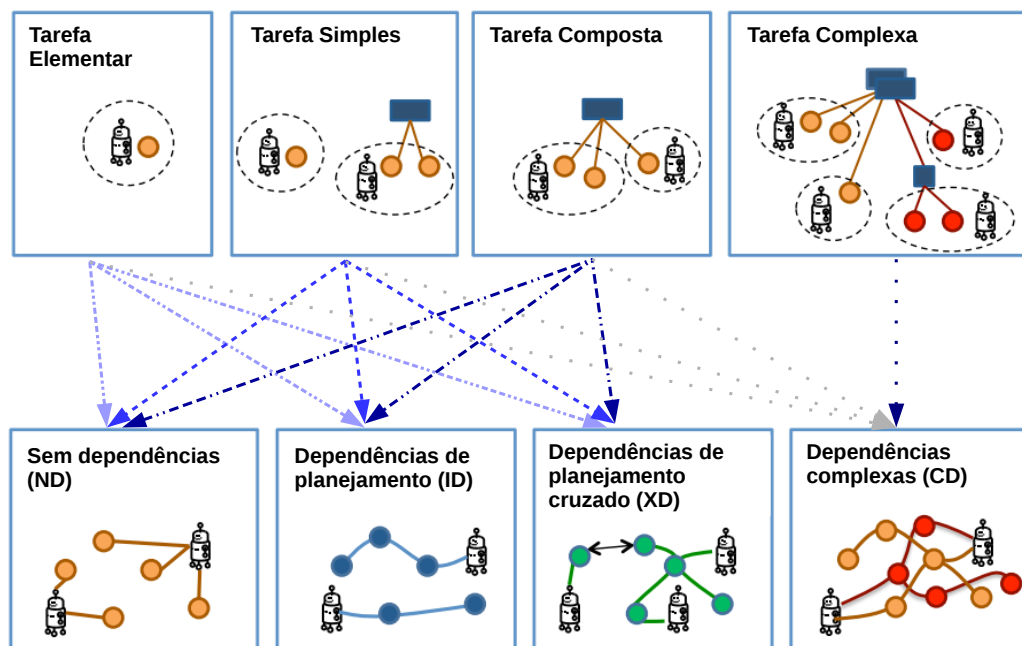


Figura 3.3 – Classificação de tarefas e dependências entre tarefas na *iTax*. Os círculos tracejados na parte superior da figura indicam a potencial alocação de tarefas entre os robôs envolvidos. Os círculos de cor sólida representam as tarefas elementares. Os retângulos de cor sólida representam as tarefas compostas, que podem ser decompostas em árvores de tarefas simples. Os retângulos sobrepostos significam as tarefas complexas, que podem ser decompostas em tarefas compostas e tarefas simples. As setas ligando as tarefas às classes de dependência exemplificam quais tipos de dependências são possíveis para cada tarefa. As setas pontilhadas mais claras indicam que as tarefas estão englobadas na tarefa complexa. Na parte inferior da figura, as linhas representam a rota de execução das tarefas pelos robôs. As setas entre as tarefas significam limitações na execução e as rotas superpostas significam as múltiplas possibilidades de decomposição de tarefas. Fonte: Adaptado de [Korsah, Stentz e Dias \(2013\)](#).

Os graus de interdependência dos problemas de alocação de tarefas vão desde a independência total (*ND - No Dependencies*) até a dependência complexa (*CD - Complex Dependencies*). Os problemas com independências têm as utilidades das tarefas totalmente independentes, tanto de outras tarefas quanto de outros agentes. A alocação de tarefas simples ou compostas pode ter um grau de dependência no planejamento da execução das mesmas pelo agente. Ou seja, a utilidade de uma tarefa depende da utilidade de outra tarefa alocada para o mesmo agente. À essa dependência, dá-se o nome de dependências de planejamento (*ID - In-Schedule Dependencies*). Quando há dependência entre as tarefas simples ou compostas executadas por agentes diferentes, diz-se que há dependências de planejamento cruzadas (*XD - Cross-Schedule Dependencies*). Nesse caso, a utilidade de uma tarefa executada por um agente interfere na utilidade de outra tarefa executada por outro agente. As dependências complexas (*CD - Complex Dependencies*) ocorrem quando tarefas complexas têm múltiplas possibilidades de decomposição, o que significa que há uma dependência forte entre as utilidades das tarefas para os agentes envolvidos.

Neste trabalho, o foco é dado em problemas compostos de tarefas simples e independentes ou com dependências de planejamento. Assim, considera-se a decomposição das tarefas previamente realizada.

3.3 O Problema de Alocação de Tarefas em SMR

O trabalho em equipe envolvendo um SMR é complexo, consistindo na divisão de tarefas, na alocação das tarefas, na coordenação e na comunicação entre os robôs (CHOUDHURY, 2009). Um ponto sensível de um sistema como esse é a correta alocação de tarefas, de modo que a missão seja executada com determinado grau de eficiência.

Primeiramente, será definido para esse trabalho o conceito de tarefa. Gerkey e Mataric (2004, p. 1) definem tarefa como “uma meta secundária necessária para se alcançar a meta global do sistema, que pode ser atingida independentemente de outras metas secundárias”. Além disso, os autores dividem tarefas em discretas ou contínuas. Uma tarefa discreta é aquela que tem início e fim bem definidos, por exemplo, coletar um objeto no ponto *A* e levar até o ponto *B*. Já uma tarefa contínua pode ser entendida como uma tarefa condicional, sendo exemplificada por uma tarefa como de monitoramento contra entrada de intrusos ou rastreamento de objetos.

A consideração de tarefas independentes, segundo Rohmüller (2011, p. 33), é basilar à maioria das abordagens MRTA e, de acordo com Gerkey e Mataric (2004), é um forte pressuposto, limitando o escopo do estudo. A independência entre as tarefas não permite a limitação do ordenamento de execução entre as mesmas, podendo ser alocadas e executadas de forma livre (GERKEY; MATARIC, 2004).

Seguindo, também, a classificação de tarefas vista na Seção 3.2, uma tarefa pode ser executada por um único robô (*SR*) ou por múltiplos robôs (*MR*), e pode-se definir tarefa de acordo com a Definição 3.1.

Definição 3.1 (Tarefa). Meta secundária que compõe uma missão, com execução independente de outras tarefas, a tarefa reúne um conjunto de ações já planejadas por uma camada de controle mais elevada, embarcada ou não no robô, podendo ser discreta ou contínua. Sua execução pode necessitar um ou mais robôs.

Com a definição de tarefa, pode-se apresentar uma definição formal do Problema de Alocação de Tarefas em SMR, dada na Definição 3.2.

Definição 3.2 (Alocação de Tarefas em Sistemas Multirrobo (MRTA)). Dados um conjunto $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ de m robôs, cada um capaz de executar uma ou mais tarefas simultaneamente, e um conjunto $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ de n tarefas possivelmente ponderadas, sendo necessário um ou mais robôs para a execução de cada tarefa. Dado, além disso, para cada robô uma estimativa não negativa da taxa de eficiência de seu desempenho para cada tarefa (se o robô é incapaz de executar determinada tarefa, então sua eficiência é nula). Então, o objetivo é atribuir robôs às tarefas de modo a maximizar o desempenho global esperado, levando em conta as tarefas prioritárias e a taxa de eficiência de cada robô.

Para Gerkey (2003, p. 13), o problema de MRTA pode ser resumido em maximizar o desempenho nas tarefas dados os recursos, as próprias tarefas e as métricas de desempenho, considerando-se, ainda, as limitações dos recursos. Ainda segundo o autor, o problema é essencialmente organizacional, tendo os primeiros esforços em seu estudo nas Ciências Econômicas (Edgeworth, 1881), embora tenha sido estendido para outras áreas como Teoria dos Jogos (von Neumann e Morgenstern, 1964), Pesquisa Operacional (Gale, 1960), Otimização Combinatória (Korte e Vygen 2000), entre outras, como citado em (GERKEY, 2003).

A estimativa de eficiência mencionada na Definição 3.2 é a utilidade do robô em relação às tarefas do sistema. Essa taxa também é chamada de *custo*, *aptidão*, *elegibilidade* e *recompensa* (GERKEY; MATARIĆ, 2004). A versão mais aceita da medição da utilidade na literatura é dada por Gerkey (2003). Sejam o robô r_m e a tarefa τ_n , a qualidade com que r_m executa a tarefa τ_n é Q_{mn} e o custo para essa execução é C_{mn} . A equação 3.1 mostra a medição não negativa da utilidade.

$$U_{mn} = \begin{cases} Q_{mn} - C_{mn}, & \text{se } r_m \text{ é capaz de} \\ & \text{executar } \tau_n \text{ e} \\ & Q_{mn} > C_{mn} \\ 0, & \text{caso contrário.} \end{cases} \quad (3.1)$$

Por diversos fatores, a medição da utilidade de um robô pode ser inexata (GERKEY, 2003). Esses fatores incluem, mas não se limitam à, ruídos de medição dos sensores, incertezas do sistema em geral, mudanças no ambiente, falhas de comunicação. Além disso, Gerkey (2003) afirma que não apenas fatores externos previamente estabelecidos, mas também o método de processamento pode influenciar a utilidade. Dada a grande dificuldade na medição da utilidade, este ainda é um problema para qualquer método de coordenação e para qualquer método de

medição ou processamento.

A utilidade escalar, pelo motivos citados, torna a comparação de utilidade entre dois robôs como um mecanismo de alocação de tarefas impraticável. Como discutido por [Arrow \(1951\)](#), a comparação de utilidades cardinais pode ser substituída pela comparação ordinal. Dessa forma, algumas características são essenciais ao modelo social de um mecanismo de alocação de tarefas. Essas características serão discutidas no Capítulo 4.

3.4 Arquiteturas de Alocação de Tarefas em SMR

Nessa seção¹ serão apresentadas e classificadas arquiteturas pesquisadas na literatura. Uma maior atenção será dedicada à três arquiteturas, entretanto, outras arquiteturas pesquisadas serão brevemente apresentadas.

3.4.1 Arquiteturas Baseadas em Comportamento

Como principais exemplos, têm-se a arquitetura totalmente distribuída e tolerante à falhas ALLIANCE ([PARKER, 1995](#); [PARKER, 1998](#)) e a abordagem *Broadcast of Local Eligibility* ou BLE ([WERGER; MATARIĆ, 2000](#)), na qual as tarefas são alocadas baseando-se na comparação da melhor elegibilidade global calculada com a elegibilidade local, ou seja, uma comparação entre a capacidade do próprio robô com a melhor capacidade calculada em todo grupo. Nestas, os robôs têm motivações internas que os levam a executar uma tarefa. As motivações internas podem ser modificadas por informações do ambiente, pelos estados dos demais robôs da equipe, pelo estado do próprio robô, mas não podem ser modificadas diretamente pelas motivações internas dos demais robôs.

Não há negociação explícita entre os robôs. Uma vez que um robô inicia a execução de uma tarefa, o restante dos robôs da equipe deve inibir a execução da mesma. A inibição pode basear-se em mensagens por difusão aberta, ou *broadcast*, previamente enviadas pelo robô executor ou no comportamento do mesmo robô. Dessa forma, a mesma tarefa não será atribuída a outro robô da equipe caso ela não permita que dois robôs a executem ao mesmo tempo. Em resumo, um robô envia uma mensagem aberta à equipe informando que uma tarefa já foi atribuída e não com uma oferta para execução da mesma, com o objetivo de iniciar uma negociação com os outros robôs.

3.4.1.1 ALLIANCE

ALLIANCE é uma arquitetura de cooperação de robôs heterogêneos totalmente distribuída e tolerante a falhas que utiliza uma seleção adaptativa de ações para atingir o controle cooperativo ([PARKER, 1995](#)). A seleção adaptativa de ações na arquitetura é “facilitada pelo

¹ As notações originais das arquiteturas aqui apresentadas serão mantidas, diferindo da notação utilizada nesse trabalho. A Lista de Símbolos traz cada notação referenciada à respectiva arquitetura.

projeto fazer dos robôs egoístas e preguiçosos”, segundo Parker (1994). Os robôs são considerados egoístas por fazer apenas o que “querem”² e o que “acreditam” ser de seu próprio interesse, como determinado pela sua motivação e pelo ambiente (PARKER, 1994, p. 31). É importante ressaltar que o julgamento do robô quanto aos seus interesses próprios é feito a partir de uma visão local, própria do robô que não tem uma visão onisciente. A qualidade de preguiçoso se dá pelo fato de que, mesmo que o robô deseje executar determinada tarefa, caso outro robô possa fazê-lo por ele, o segundo não será impedido (PARKER, 1994, p. 31). Essa característica, apesar de aparentemente danosa, traz robustez à arquitetura. Além de “prevenir o gasto desnecessário de energia devido à repetição do esforço” (PARKER, 1994, p. 32).

Os robôs nessa arquitetura são projetados sob uma abordagem baseada em comportamento. A Figura 3.4a exemplifica a organização típica de uma arquitetura desse tipo. Os algoritmos de controle dos robôs podem ser divididos em dois níveis de abstração: comportamentos de baixo-nível e de alto-nível. Os comportamentos de baixo-nível são aqueles considerados para a “sobrevivência primitiva” do robô, como definido por Parker (1995). Comportamentos de baixo-nível podem ser exemplificados como *desviar_de_obstáculos*, *carregar_bateria*, etc. Os comportamentos de alto-nível estão relacionados à realização efetiva de tarefas, tais como *explorar_ambiente* ou *construir_mapa_do_ambiente* (PARKER, 1994). Os comportamentos de alto nível, adequadamente agrupados, formam uma função de realização de tarefas de alto nível (*high-level task-achieving functions*), que são chamados também de conjuntos de comportamentos (*behavior sets*). O conjunto de comportamentos é o conjunto de ações necessárias para a execução de uma tarefa. Essas ações podem ser descritas por módulos, que interagem entre si dentro de uma camada, gerando o comportamento desejado para execução de uma tarefa. A mesma interação possível entre camadas, de supressão e inibição, também pode ocorrer dentro da camada, como mostra a Figura 3.4b.

Assim, a arquitetura tem como premissa a definição de cada tarefa como um conjunto de comportamentos. O robô que detém o conjunto de comportamentos relacionado à execução de determinada tarefa está apto para executá-la. Os robôs membros da equipe podem ter diferentes conjuntos de comportamentos para a mesma tarefa ou, ainda, podem ter diferentes conjuntos de comportamento significando aptidões para execução de tarefas diferentes (PARKER, 1994).

Os conjuntos de comportamento recebem os sinais de entrada dos sensores e controlam as saídas para os atuadores simultaneamente, de certa forma (PARKER, 1994). Os comportamentos de alto-nível podem *suprimir* ou *inibir* a saída dos comportamentos de baixo-nível quando for necessário. Inibir um comportamento significa excluir sua influência na saída de controle. Enquanto suprimir significa, além de excluir a saída do comportamento de baixo-nível, substituí-la por um valor desejado. Essa substituição é realizada pelo comportamento de alto-nível responsável por esse controle. Por exemplo, o robô conta com um comportamento de baixo-nível de desvio de obstáculo, entretanto, um comportamento de alto-nível necessita empurrar uma caixa. Nessa situação, os comportamentos são conflitantes e, por isso, o compor-

² A autora destaca o uso de palavras que expressam intenção ou cognição, mas é apenas uma forma de simplificar a discussão e não significa que os robôs realmente possuem tal organização interna.

tamento de baixo-nível deve ser inibido.

A seleção adaptativa de ações e, conseqüentemente, de alocação de tarefas da arquitetura tem como base o mecanismo do comportamento motivacional (*motivational behavior*) (PARKER, 1994). Esse comportamento está ligado diretamente a um conjunto de comportamentos do robô e monitora um conjunto de entradas (leituras de sensores, mensagens de comunicação entre os robôs, sinais de realimentação de inibição vindos de outros comportamentos motivacionais e motivações internas ao robô, chamadas *impaciência* e *aquiescência*) para ativar ou não a saída de seu respectivo conjunto de comportamentos. “A saída de um comportamento motivacional em um determinado instante é o nível de ativação do conjunto de comportamentos a ele correspondente, representado por um número real positivo” (PARKER, 1994, p. 36). A ativação de um determinado conjunto de comportamentos levará à respectiva saída de controle aos atuadores do robô. A Figura 3.5 exemplifica uma estrutura da arquitetura, composta pelos comportamentos motivacionais, os conjuntos de comportamentos, as camadas de comportamentos de baixo-nível e suas possíveis interações de supressão e inibição.

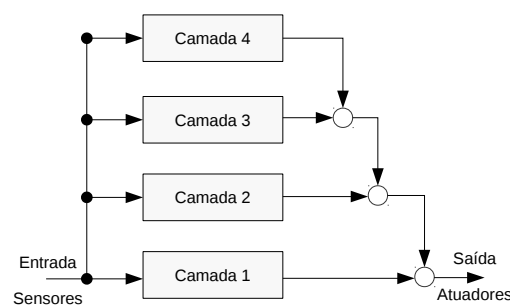
Antes de discutir o modelo formal da arquitetura, Parker (1994) estabeleceu algumas premissas para a mesma:

- a. Os robôs da equipe podem detectar sua influência no ambiente causada por suas próprias ações, com uma probabilidade maior que 0.
- b. O robô r_i pode identificar capacidades redundantes às suas próprias em outros robôs da equipe e o efeito das ações desses outros robôs, com uma probabilidade maior que 0. Essa identificação pode ocorrer por qualquer meio disponível, incluindo comunicação por difusão explícita.
- c. Os robôs partilham de uma linguagem comum.
- d. Robôs de uma equipe não mentem ou são inimigos intencionalmente.
- e. Não há um mapeamento global do ambiente disponível em dispositivo de armazenamento centralizado.
- f. O meio de comunicação não tem garantia de disponibilidade, mensagens podem ser perdidas.
- g. Os robôs não possuem sensores ou atuadores perfeitos.
- h. Se um robô falha, não necessariamente comunica aos demais robôs da equipe seu estado de falha atual.

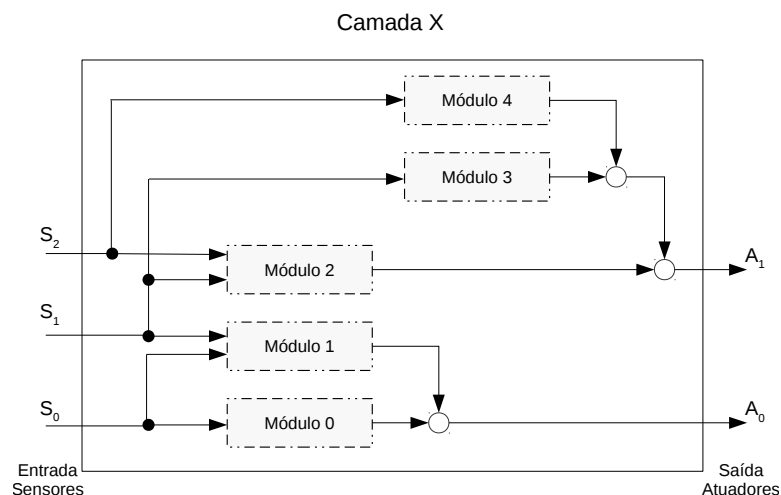
O modelo formal da arquitetura define o conjunto $R = \{r_0, r_1, r_2, \dots, r_n\}$ de $n + 1$ robôs e o conjunto de m tarefas $T = \{task_1, task_2, task_3, \dots, task_m\}$. Além disso, cada robô é dotado de conjuntos de comportamentos ou funções de realização de tarefas de alto nível $A_i = \{a_{i1}, a_{i2}, \dots\}$. Como os robôs podem executar a mesma tarefa de maneiras diferentes, é estabelecido o conjunto de funções de ativação $\{h_1(a_{1k}), h_2(a_{2k}), \dots, h_n(a_{nk})\}$, onde $h_i(a_{ik})$ retorna a

tarefa em T a qual o robô r_i está executando quando o conjunto de comportamentos a_{ik} é ativado (PARKER, 1998). O funcionamento da arquitetura depende de seis fatores principais: o limiar de ativação, responsável por definir o nível de motivação que ativa um determinado conjunto de comportamentos, e cinco entradas principais. Esses fatores serão apresentados a seguir.

Limiar de Ativação θ Esse parâmetro, identificado pela letra grega θ , identifica o nível de motivação que faz com que o comportamento motivacional ative determinado conjunto de comportamentos. Segundo Parker (1994), mesmo que diferentes níveis de limiar possam ser utilizados para diferentes conjuntos de comportamentos (o grau variável de impaciência e de aquiescência dos robôs, por exemplo) apenas um limite basta ser utilizado.



(a) As camadas correspondem às capacidades do robô. Camadas mais baixas, como a Camada 1, são responsáveis por comportamentos fundamentais e básicos. As camadas superiores tratam de comportamentos mais sofisticados, relacionados à execução de tarefas específicas. Os círculos indicam a supressão ou inibição das saídas, durante a interação entre as camadas.



(b) Internamente, as camadas podem apresentar a mesma característica de supressão e inibição na interação de seus módulos básicos. Essa interação é a responsável por gerar os comportamentos desejados.

Figura 3.4 – A estrutura típica de camadas de uma arquitetura baseada em comportamento e a interação entre camadas e módulos responsáveis por gerar os comportamentos desejados. Fonte: Adaptado de Parker (1994).

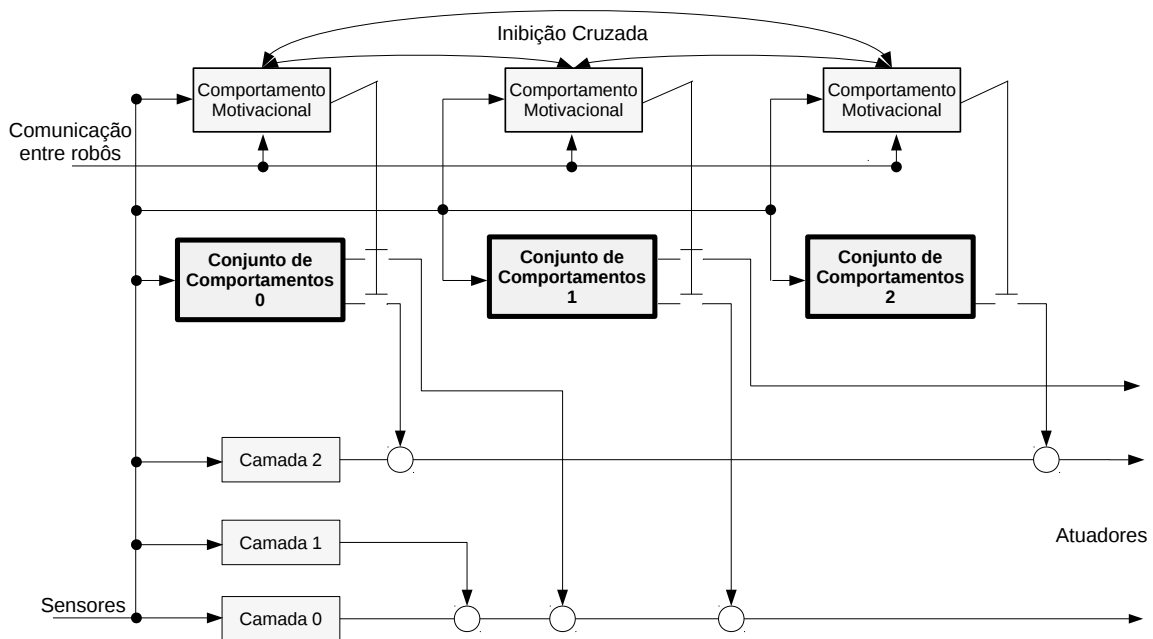


Figura 3.5 – Arquitetura ALLIANCE. Essa estrutura é desenvolvida para cada robô que compõe a equipe, cada um com suas características e capacidades. Fonte: Adaptado de Parker (1994).

Realimentação sensorial - *sensory_feedback* Essa entrada define quando um conjunto de comportamentos é aplicável ou não, de acordo com as informações dos sensores físicos do robô, principalmente, no tempo t . Entretanto, algumas variáveis não podem ser medidas, como os estados da missão, por exemplo. Esses estados, comunicados à equipe por mensagens de difusão ou atualizados por um robô, são armazenados em memória, sendo considerados sensores *virtuais* (PARKER, 1994).

A entrada para o comportamento a_{ij} , no tempo t , pode ser representada de acordo com a Equação 3.2:

$$sensory_feedback_{ij}(t) = \begin{cases} 1, & \text{se a realimentação sensorial do robô } r_i \text{ indica no} \\ & \text{tempo } t \text{ que o conjunto de comportamentos } a_{ij} \text{ é} \\ & \text{aplicável} \\ 0, & \text{caso contrário} \end{cases} \quad (3.2)$$

Confirmação de comunicação - *comm_received* A comunicação entre os robôs por meio de mensagens por difusão é um mecanismo chave para a determinação dos estados dos companheiros de equipe (PARKER, 1994). Através desse mecanismo, os comportamentos motivacionais de um robô atuam na inibição dos conjuntos de comportamentos, conforme as mensagens recebidas. Ainda, a comunicação depende de parâmetros temporais, tornando o mecanismo mais robusto.

Dois parâmetros de tempo são utilizados: ρ_i e τ_i . O parâmetro ρ_i informa a taxa com que

o robô r_i envia sua atividade corrente. O segundo parâmetro, τ_i , determina o máximo período de tempo permitido por r_i a um companheiro de equipe deixar de enviar sua atividade. A partir desse período de tempo, r_i pode julgar que o companheiro parou de funcionar. Esse parâmetro aumenta o nível de tolerância à falhas da arquitetura (PARKER, 1994). Esse parâmetro avalia constantemente o estado dos demais membros da equipe, em relação às tarefas alocadas. A Equação 3.3 define os valores de acordo com as mensagens recebidas:

$$comm_received(i, k, j, t_1, t_2) = \begin{cases} 1, & \text{se o robô } r_i \text{ recebeu mensagem do robô } r_k \\ & \text{relacionada a tarefa } j \text{ no período de tempo} \\ & (t_1, t_2), \text{ onde } t_1 < t_2 \\ 0, & \text{caso contrário} \end{cases} \quad (3.3)$$

Supressão de atividade - *activity_suppression* Quando um conjunto de comportamentos é ativado, simultaneamente, ele passa a inibir os demais conjuntos daquele robô. Nesse momento, o robô efetivamente escolheu sua ação (PARKER, 1994), ou seja, alocou uma tarefa. Com isso, o primeiro comportamento motivacional continua monitorando as demais entradas, verificando a necessidade da manutenção da ativação do comportamento. Essa supressão dos demais conjuntos de comportamentos pode ser traduzido pela Equação 3.4, que determina que o conjunto de comportamentos a_{ij} seja suprimido, no tempo t no robô r_i , se algum outro conjunto de comportamentos a_{ik} está atualmente ativo no robô r_i no tempo t :

$$activity_suppression_{ij}(t) = \begin{cases} 0, & \text{se outro conjunto de comportamentos } a_{ik} \\ & \text{está ativo, } k \neq j, \text{ no robô } r_i \text{ no tempo } t \\ 1, & \text{caso contrário} \end{cases} \quad (3.4)$$

Impaciência - *impatience* O grau de impaciência do robô é uma grandeza que varia com o tempo para cada conjunto de comportamentos, a medida que as tarefas relacionadas aos conjuntos de comportamentos ainda não foram concluídas. Essa característica “reflete o fato de que outros robôs podem falhar” (PARKER, 1994, p. 37). A impaciência envolve três parâmetros básicos: $\phi_{ij}(k, t)$, $\delta_slow_{ij}(t)$ e $\delta_fast_{ij}(t)$. O parâmetro $\phi_{ij}(k, t)$ define o período de tempo o qual o robô r_i está disposto a permitir que as mensagens do robô r_k influenciem seu cálculo de motivação. Em outras palavras, este é o tempo máximo que o robô r_k tem para executar a tarefa $h_i(a_{ij})$.

Os parâmetros $\delta_slow_{ij}(t)$ e $\delta_fast_{ij}(t)$ são as taxas de variação da impaciência do robô enquanto a tarefa é executada e enquanto a mesma não é executada, respectivamente. Quando uma tarefa não está alocada, a impaciência de um robô evolui mais rapidamente, de forma a alocar a tarefa, para que a mesma seja concluída. A Equação 3.5 define, de forma básica, quando cada uma das taxas de variação deve ser utilizada para o comportamento a_{ij} :

$$impatience_{ij}(t) = \begin{cases} \delta_slow_{ij}(t), & \text{se } (comm_received(i, k, j, t - \tau_i, t) = 1) \wedge \\ & (comm_received(i, k, j, 0, \phi_{ij}(k, t)) = 1) \\ \delta_fast_{ij}(t), & \text{caso contrário} \end{cases} \quad (3.5)$$

A Equação 3.5 define que, se o robô r_i recebe mensagem referente ao comportamento a_{ij} nas últimas τ_i unidades de tempo, sua impaciência irá evoluir com uma taxa lenta, pois há um robô executando a tarefa relativa àquele comportamento. Entretanto, se esse período não for respeitado, configura-se a não execução da tarefa dada por $h_i(a_{ij})$, e, assim, a evolução da impaciência será mais rápida, de forma a atingir o limiar de ativação mais rapidamente.

Restauração da Impaciência - *impatience_reset* Se a ativação de um conjunto de comportamentos é comunicado por outro robô, a motivação para esse conjunto deve ser zerada na primeira vez em que a comunicação for recebida (PARKER, 1994). Essa influência limitada tem o objetivo de evitar que robôs mais lentos ou que apresentam problemas retardem a conclusão da missão. A Equação 3.6 apresenta a definição da condição para restauração da impaciência do comportamento a_{ij} , que depende do recebimento da mensagem no último intervalo de tempo e apenas nesse último intervalo de tempo. Ou seja, se anteriormente a mesma mensagem já havia sido recebida, a impaciência não será restaurada.

$$impatience_reset_{ij}(t) = \begin{cases} 0, & \text{se } \exists k.(comm_received(i, k, j, t - \delta t) = 1) \\ & \wedge (comm_received(i, k, j, 0, t - \delta t) = 0), \\ & \text{onde } \delta t = \text{tempo em que a última comunicação} \\ & \text{foi conferida} \\ 1, & \text{caso contrário} \end{cases} \quad (3.6)$$

Aquiescência - *acquiescence* A Aquiescência indica a capacidade de reconhecimento do robô que ele próprio pode falhar (PARKER, 1994, p. 37). Ainda segundo Parker (1998), essa característica faz com que o robô adapte suas ações às suas próprias falhas e, dessa forma, que um comportamento possa ser abandonado se uma tarefa não está sendo executada de forma satisfatória. Dois parâmetros efetivam a aquiescências aos robôs: $\psi_{ij}(t)$ and $\lambda_{ij}(t)$. O parâmetro $\psi_{ij}(t)$ indica o tempo em que o robô r_i mantém o comportamento a_{ij} ativo, antes de permitir que o robô r_k o ative. Ou seja, esse parâmetro define o tempo máximo de execução de uma tarefa. Já o parâmetro $\lambda_{ij}(t)$ define o período de tempo em que r_i manterá o comportamento a_{ij} ativo, antes de desistir deste e partir para outro comportamento. O parâmetro $\psi_{ij}(t)$ trata da anuência de r_i para que outro robô execute a tarefa e $\lambda_{ij}(t)$ trata da aceitação de r_i de que ele não é capaz de cumprir aquela tarefa de forma satisfatória. A Equação 3.7 assinala as situações em que o robô desativa a execução do comportamento a_{ij} :

$$acquiescence_{ij}(t) = \begin{cases} 0, & \text{se o conjunto de comportamentos } a_{ij} \text{ do robô } r_i \\ & \text{está ativo por mais } \psi_{ij}(t) \text{ unidades de tempo } \wedge \\ & \exists x.(comm_received(i, x, j, t - \tau_i, t) = 1) \\ & \vee \\ & \text{se o conjunto de comportamentos } a_{ij} \text{ do robô } r_i \\ & \text{está ativo por mais } \lambda_{ij}(t) \text{ unidades de tempo} \\ 1, & \text{caso contrário} \end{cases} \quad (3.7)$$

Com as entradas definidas, a motivação do comportamento a_{ij} é calculada de acordo com a Equação 3.8:

$$\begin{aligned}
 m_{ij}(0) &= 0, \\
 m_{ij}(t) &= [m_{ij}(t-1) + \textit{impatience}_{ij}(t)] \\
 &\quad \cdot (\textit{sensory_feedback}_{ij})(t) \\
 &\quad \cdot (\textit{activity_suppression}_{ij})(t) \\
 &\quad \cdot (\textit{impatience_reset}_{ij})(t) \\
 &\quad \cdot (\textit{acquiescence}_{ij})(t)
 \end{aligned} \tag{3.8}$$

Inicialmente, a motivação é nula. Com o passar do tempo, há um aumento de acordo com o nível de impaciência do robô, até que a motivação atinja o limiar de ativação θ . E, então, o conjunto de comportamentos é ativado e a tarefa $h_i(a_{ij})$ alocada. Isso ocorre a menos que uma, ou mais, dessas situações ocorram (PARKER, 1994): (1) a realimentação sensorial indica que o comportamento não é adequado; (2) outro conjunto de comportamentos é ativado; (3) outro robô assume a tarefa $h_i(a_{ij})$ pela primeira vez; e (4) o robô consente em deixar de executar a tarefa. Se uma das situações citadas anteriormente ocorre, a motivação retorna ao nível zero.

Assim, as tarefas são alocadas de acordo com as ativações dos conjuntos de comportamentos realizadas por cada um dos robôs da equipe. As motivações internas (impaciência e aquiescência) e as respostas do ambiente (informações dos sensores e comunicação difusa entre os robôs) são as informações necessárias à adaptação das ações, imprimindo, assim, um comportamento cooperativo, mesmo que não haja nenhum tipo de comunicação explícita entre os robôs para a execução de tarefas.

Para avaliar o funcionamento da arquitetura, Parker (1994, p. 44) propõe um conjunto de definições de forma a verificar se o robô é confiável, se o time de robôs é confiável, se as tarefas que compõem a missão são minimamente cobertas, ou seja, pelo menos um robô é capaz de executá-las, entre outras. Essa análise engloba, também, resultados de implementações da arquitetura em equipes de robôs reais e em simulações. Neste trabalho, não será abordada tal análise com maior profundidade.

3.4.1.1.1 O problema de eficiência do ALLIANCE

A arquitetura ALLIANCE por si só faz com que o projetista tenha grande responsabilidade no projeto e na definição dos valores dos parâmetros discutidos na seção anterior. Parker (1994) propõe uma versão estendida da arquitetura que permite o aprendizado, chamada L-ALLIANCE (*Learning ALLIANCE*). Nessa extensão, o time de robôs pode aprender sobre o desempenho dos demais membros, assim, melhorando a eficiência da cooperação da equipe de robôs na execução de uma missão.

A versão L-ALLIANCE aborda vários problemas não considerados na arquitetura original, como certificar de que os melhores robôs para cada tarefa sejam alocados para as tarefas que compõem a missão, como minimizar o tempo inativo de um robô, ou como selecionar um

método para a execução da tarefa, visto que mais de um pode levar à conclusão da mesma (PARKER, 1994). Além disso, as motivações para a implementação da melhoria incluem: (a) isentar o robotista do ajuste dos parâmetros, (b) aumentar a eficiência de desempenho da missão, (c) facilitar a interação de grupos customizados de robôs heterogêneos e (d) permitir que o grupo avalie seu desempenho e o adapte de forma eficiente durante a missão (PARKER, 1994; PARKER, 1998).

As premissas consideradas por Parker (1994) para o L-ALLIANCE são:

- a. O desempenho médio de um robô na execução de uma determinada tarefa, em ensaios recentes, é um indicador razoável de seu desempenho no futuro.
- b. Se o robô r_i está monitorando o ambiente para avaliar o desempenho do robô r_k e as condições do ambiente mudam, então as mudanças são atribuídas ao robô r_k .

A primeira afirmação tem ligação direta à capacidade de aprendizado do robô, tanto de seu desempenho, quanto de seus colegas de equipe. A segunda premissa considera que apenas os robôs modificam o ambiente e, por isso, é possível medir e creditar uma determinada mudança à um determinado robô que se esteja observando. Essas premissas levam ao aprendizado do desempenho do grupo, como um todo. Entretanto, o interesse não está no algoritmo de aprendizado, mas sim no problema de otimização observado por Parker (1994) para a alocação das tarefas, o Problema de Eficiência do ALLIANCE (*ALLIANCE Efficiency Problem - AEP*).

Seja o modelo formal da arquitetura definido na Seção 3.4.1.1, Parker (1994) define uma nova função métrica de avaliação da “qualidade” da ação a_{ij} , $q(a_{ij})$, medida a partir de uma métrica pré-definida. Robôs que ainda não têm conhecimento de suas habilidades ou das de seus colegas de equipe não têm acesso às funções de qualidade $q(a_{ij})$. Dessa forma, um aspecto importante do aprendizado dos robôs é o conhecimento da qualidade para que a escolha seja feita de uma maneira satisfatória.

Ainda, seja definido o conjunto $U_i = \{a_{ij} | \text{o robô } r_i \text{ irá executar a tarefa } h_i(a_{ij}) \text{ durante a missão corrente}\}$. Esse conjunto define as tarefas as quais o robô elege executar durante a missão.

Uma consideração importante é que robôs diferentes são diferentes (PARKER, 1994). Isso quer dizer que, como comportamentos diferentes podem levar à execução das mesmas tarefas, não é possível esperar que dois robôs diferentes possam ter a mesma medida de qualidade para a mesma tarefa, independente da métrica utilizada. Com isso, o AEP pode ser enunciado:

Problema de Eficiência do ALLIANCE - AEP

Para cada robô r_i : Dado T , A_i e $h_i(a_{ik})$, determinar o conjunto de ações U_i tal que:

- $\forall i. U_i \subseteq A_i$
- $\forall j. \exists i. \exists k. ((task_j = h_i(a_{ik})) \wedge (a_{ik} \in U_i))$

e que seja minimizada uma das seguintes funções, dependendo da métrica utilizada:

- $\max_i(\sum_{a_{ik} \in U_i} q_{tempo}(a_{ik}))$
- $\sum_i \sum_{a_{ik} \in U_i} q_{energia}(a_{ik})$

A primeira condição do problema define que o conjunto de ações eleitas pelos robôs para execução U_i é um subconjunto do conjunto de comportamentos A_i . A segunda, que para toda tarefa $task_j$ há um comportamento equivalente a_{ik} , componente do subconjunto U_i de ações eleitas. Essas condições juntas asseguram que cada tarefa da missão será atribuída a um robô que pode realmente executá-la (PARKER, 1994, p.83). As funções a serem minimizadas, dependendo da métrica utilizada, asseguram a busca por uma resposta ótima. No caso da métrica ser o tempo de execução de tarefas, o tempo total da missão será o tempo que o último robô levou para terminar sua última tarefa. Com essa métrica, o paralelismo e a redundância podem ser utilizados como forma de minimizar o tempo total gasto na missão. Entretanto, quando a métrica utilizada é a energia, esses princípios não se aplicam, visto que os robôs utilizarão a energia de qualquer maneira. Neste caso, o foco é minimizar a quantidade de energia utilizada durante a missão.

Parker (1994) ressalta que o problema deve ser resolvido por um grupo distribuído de robôs, não por um único decisor centralizador. Para a autora, tal decisor elimina vantagens importantes da arquitetura como robustez, tolerância à falhas e flexibilidade. A decisão deve ser coletiva, mesmo que sem comunicação explícita entre os membros da equipe.

O problema, de acordo com Parker (1994), é classificado como um problema $\mathcal{N}(\mathcal{P})$ -difícil. Gerkey e Mataric (2004) identificam o AEP como um variante da classe dos problemas de escalonamento. Como o problema trata das tarefas elencadas pelos robôs para a execução durante a missão, deve-se dispor uma programação de execução de tarefas ao longo do tempo. Dessa forma, esse problema se enquadra como um exemplo de taxonomia ST-SR-TA (GERKEY; MATARIC, 2004).

A extensão L-ALLIANCE foi concebida para amenizar o impacto do AEP na arquitetura, de modo que os robôs pudessem aprender suas utilidades e seus algoritmos para escalonamento a partir da experiência (GERKEY; MATARIC, 2004). Sabendo-se da complexidade do problema, Gerkey e Mataric (2004) afirmam que, quando treinada para um domínio particular, o sistema tem a capacidade de atingir soluções aproximadas à solução ótima, mesmo que isso não seja garantido. Além disso, Parker (1994) reconhece que não se pode esperar respostas ótimas do sistema em um período de tempo razoável, focando na heurística que apresentou respostas aproximadas na prática.

3.4.2 Arquiteturas Baseadas em Mercado

Os princípios da economia de mercado, a qual governa por anos a interação entre as sociedades, podem também ser aplicados à coordenação de SMR (DIAS et al., 2006). Segundo

Kalra et al. (2005), nessa economia virtual, os robôs são os negociantes, as tarefas e recursos são equivalentes às mercadorias a serem negociadas e a recompensa dada aos robôs pela execução das tarefas é como a moeda. Nessa abordagem, os robôs agem de maneira individualista negociando tarefas e recursos visando o aumento de seus próprios ganhos. Simultaneamente, a busca pelo aumento dos ganhos individuais leva ao aumento da eficiência da equipe (DIAS et al., 2006).

Arquiteturas baseadas em mercado envolvem, normalmente, uma negociação explícita para elevar a utilidade global do sistema, ou seja, otimizar a função objetivo (LIU, 2014). Em termos de organização, esse tipo de abordagem pode ser vista como uma categoria híbrida entre as classificações centralizada e descentralizada. Em algumas circunstâncias, a arquitetura centraliza a distribuição de tarefas, focando em subequipes. A abordagem pode, de modo oportunista, adaptar-se às condições dinâmicas da missão, de forma a produzir respostas mais centralizadas ou mais descentralizadas (KALRA et al., 2005; LIU, 2014). Para Kalra et al. (2005) os mecanismos de mercado podem distribuir significativamente o planejamento e a execução entre os membros da equipe e, assim, mantendo os benefícios de organizações descentralizadas, incluindo robustez, flexibilidade e velocidade.

Entretanto, essa abordagem não está livre de desvantagens. Dias et al. (2006) afirmam que os maiores problemas de abordagens baseadas em mercado são a falta de garantia de performance e a falta de formalização na definição apropriada das funções de custo e de recompensa (utilidade) que atendam aos requisitos dos domínios de aplicação.

Há várias arquiteturas baseadas em mercado na literatura, como a MURDOCH (GERKEY; MATARIĆ, 2002), a M+ (BOTELHO; ALAMI, 1999), e a *Dynamic role assignment* (CHAIMOWICZ; CAMPOS; KUMAR, 2002), além de várias outras baseadas nas mesmas premissas. Nessas arquiteturas, após o anúncio de uma nova tarefa disponível, os robôs interessados (aqueles que podem executar tal tarefa), avaliam as métricas apropriadas e calculam uma oferta pela execução da tarefa. Esse valor tem o nome de *valor de aptidão* em (GERKEY; MATARIĆ, 2002), ou *custo* em (BOTELHO; ALAMI, 1999), ou *utilidade* da tarefa em (CHAIMOWICZ; CAMPOS; KUMAR, 2002). Os interessados enviam mensagens por difusão com suas ofertas e o robô mediador, chamado *leiloeiro* em (GERKEY; MATARIĆ, 2002) ou *líder de atribuição* em (CHAIMOWICZ; CAMPOS; KUMAR, 2002), avalia as ofertas para atribuir a tarefa.

3.4.2.1 MURDOCH

A arquitetura, proposta por Gerkey e Mataric (2002), é uma variação do protocolo *Contract Net*, utilizando leilões simples para alocação das tarefas. O protocolo de comunicação *Contract Net* (SMITH, 1980), tido como um dos primeiros exemplos de coordenação multiagente baseado em mercado (KALRA et al., 2005), foi concebido para tratar a comunicação entre nós de um sistema solucionador distribuído. O CNP (*Contract Net Protocol*) não se preocupa com a arquitetura física do sistema, considerando que se trata de uma rede de nós fracamente

acoplada e assíncrona. O protocolo define o que deve ser comunicado entre os nós e não como isso ocorre (SMITH, 1980). Os nós se relacionam como uma rede de contratos onde a execução de uma tarefa é negociada com um contrato firmado entre dois nós. Os nós na rede podem assumir dois papéis, o administrador e o contratante. O nó administrador monitora a execução da tarefa, processando os resultados dessa execução. O nó contratante é o real executor da tarefa. São definidos também no CNP parâmetros como a especificação da elegibilidade, critérios que tornam o nó apto a submeter uma oferta, a abstração da tarefa, uma breve descrição da mesma, a especificação da oferta e o tempo de validade do recebimento de ofertas (SMITH, 1980).

O problema tratado pela arquitetura foi definido como um *problema de alocação dinâmica de recursos* (GERKEY; MATARIĆ, 2002). Segundo Gerkey e Matarić (2002), “dado um conjunto de recursos e uma sequência de tarefas, o objetivo é atribuir recursos às tarefas de maneira eficiente”. O domínio caracterizado tem as seguintes considerações:

- a. O sistema é composto por robôs fisicamente materializados.
- b. Os robôs são heterogêneos, com diferentes capacidades.
- c. Os robôs podem se comunicar, mas mensagens podem ser perdidas.
- d. Os robôs são honestos e cooperativos.
- e. Os robôs, ou parte deles, podem falhar a qualquer momento.
- f. Um robô pode não estar ciente de sua falha ou falha parcial.
- g. Os robôs não tem habilidade específica para apenas uma tarefa, tem múltiplas capacidades.
- h. Não há um modelo prévio que descreva a sequência na qual as tarefas são geradas.
- i. Se um robô supervisiona uma tarefa, ele pode determinar o progresso e a conclusão dessa tarefa.

Na arquitetura, como não há um modelo que descreva o comportamento futuro dos requisitos de alocação de tarefas, o objetivo é alocar as tarefas já conhecidas da forma mais eficiente possível, minimizando o uso de recursos, o tempo de execução de tarefas e a sobrecarga de comunicação (GERKEY; MATARIĆ, 2002). Gerkey e Matarić (2002) ressaltam que esses critérios não são ortogonais entre si, o que significa uma elevada interação. Neste caso, a minimização de um critério afeta sensivelmente a minimização de outro, entretanto, a otimização global permanece a mesma.

O modelo empregado na arquitetura para a resolução do problema de alocação de tarefas baseia-se em três eixos principais: comunicação anônima, estrutura hierárquica de tarefas e o modelo de leilões (GERKEY; MATARIĆ, 2002; GERKEY, 2003). Além das vantagens do método de comunicação ser via mensagens por difusão, a MURDOCH apresenta uma vantajosa comunicação anônima, frequentemente chamada *subject-based addressing*, ou, em tradução livre, endereçamento baseado no assunto, ou *publish/subscribe model*, modelo editor/assinante,

(GERKEY; MATARIĆ, 2002; PARKER, 2008). Esse tipo de comunicação é centrada nos recursos disponíveis em cada robô, os quais assinam apenas os tópicos que correspondem aos recursos que os mesmos dispõem.

A estrutura hierárquica de tarefas proposta na arquitetura torna flexível a geração das mesmas, segundo Gerkey e Mataric (2002). Como o foco principal da arquitetura não é a decomposição das tarefas, ou seja, agir como um planejador na criação de árvores de tarefas a partir de uma missão (um conjunto de tarefas), é delegado ao projetista a definição da árvore de tarefas e como as mesmas estão conectadas, na forma aceita pela MURDOCH. Essa descrição hierárquica das tarefas as conecta na forma pais-filhas, de modo que a tarefa pai é responsável pela alocação e monitoramento da tarefa filho (GERKEY; MATARIĆ, 2002).

Várias áreas de estudo já se dedicaram à investigação do uso de leilões para tomada de decisão, incluindo Teoria dos Jogos, Pesquisa Operacional e Sistemas Multiagente, assim como vários algoritmos já foram desenvolvidos e analisados para manter e decidir os resultados de leilões (GERKEY; MATARIĆ, 2002). Segundo Gerkey (2003), “como resultado, alguns sistemas de leilões são conhecidos por produzir soluções ótimas para problemas de alocação de recursos”. As características que justificam a utilização desse mecanismo de decisão na arquitetura são a escalabilidade do sistema de leilões, a possibilidade de uma significativa categorização dos dados e do controle e eficiência na comunicação e na computação (GERKEY; MATARIĆ, 2002). É admitido para a arquitetura que “de alguma forma, o leiloeiro centraliza o sistema. Entretanto, devido à utilização da estrutura hierárquica de tarefas e como qualquer membro da equipe pode se tornar o leiloeiro de uma determinada tarefa, a MURDOCH é essencialmente um mecanismo distribuído” (GERKEY; MATARIĆ, 2002).

Protocolo de leilões da arquitetura MURDOCH

- 1. Anúncio da tarefa** O anúncio da tarefa pode ser realizado por um membro da equipe em nome de um usuário (o projetista do sistema, por exemplo), de um alarme ou de uma tarefa (da estrutura hierárquica de tarefas pais e filhas). Ao anunciante da tarefa é usado o termo leiloeiro. A partir desse momento, o agente leiloeiro é o responsável pelo monitoramento da tarefa. A mensagem de anúncio publicada contém detalhes da tarefa, como os recursos necessários para sua execução, duração e tópico no qual será realizada a negociação (via comunicação anônima). Essa mensagem é destinada apenas aos tópicos que representam os recursos necessários à execução da mesma, de forma que apenas os robôs que possuam tais recursos recebam a mensagem.
- 2. Avaliação da métrica** Este é o processo decisório fundamental da arquitetura, já que mais de um robô é capaz de executar a mesma tarefa. O objetivo é alocar cada tarefa ao robô mais indicado para cada uma delas. Para isso, a mensagem de anúncio inclui as métricas as quais os robôs candidatos devem se basear para calcular seu grau de aptidão (*fitness score*). Esse valor é a utilidade calculada em cada robô, para as métricas apropriadas para cada tarefa. Por exemplo, se a tarefa inclui um deslocamento, a métrica envolvida

pode ser a distância atual do robô da posição alvo. Além disso, “é importante ressaltar que sendo as métricas funções dos dados dos sensores de cada robô, esse cálculo pode não representar fielmente a utilidade dos robôs, resultando em alocações não-ótimas da tarefa” (GERKEY, 2003).

3. Submissão da oferta Após a avaliação das métricas apropriadas à tarefa, cada robô publica sua oferta (seu valor de utilidade para aquela tarefa) no tópico referente à negociação da mesma.

4. Fechamento do prazo de submissão das ofertas O leiloeiro aguarda um determinado período de tempo para o recebimento das mensagens de oferta. Após esse período, as ofertas são processadas, determinando o vencedor do leilão (o robô que irá executar a tarefa), além de noticiar o resultado aos demais participantes, que voltam a aguardar por novas tarefas. Ao ganhador é dado um contrato de execução da tarefa de tempo limitado.

5. Monitoramento do progresso e renovação do contrato Enquanto o ganhador executa a tarefa, o leiloeiro monitora seu progresso, avaliando o quão satisfatório é o mesmo. Caso o progresso seja suficiente, o leiloeiro renova o contrato periodicamente por meio de mensagem específica, para qual recebe uma mensagem de confirmação do ganhador. Caso o progresso seja insatisfatório, a tarefa pode ser realocada, retornando ao passo 1. Ainda, caso o leiloeiro não receba a mensagem de confirmação após a renovação de um contrato, este assume que o executor falhou e a tarefa pode ser realocada. Esse procedimento, aliado ao contrato de tempo determinado para execução das tarefas, dá à arquitetura capacidade de tolerância à falhas (GERKEY; MATARIĆ, 2002).

A tarefa alocada de acordo com o protocolo de leilões será designada ao robô disponível no momento que apresente a maior oferta, mostrando ter a maior aptidão para a aquela execução. Dessa forma, a arquitetura trabalha como um planejador de tarefas guloso e instantâneo (GERKEY; MATARIĆ, 2002), mostrado no Algoritmo 3.1.

Algoritmo 3.1 (Alocação de tarefas na arquitetura MURDOCH).

1. Quando uma nova tarefa é anunciada, a atribua ao robô disponível de maior aptidão.

Devido às características do Algoritmo 3.1 comuns aos algoritmos gulosos, a arquitetura pode apresentar soluções finais não-ótimas devido a ordem de apresentação não-ótima das tarefas, isto é, apesar de existir recursos suficientes para execução das tarefas, a ordem em que são apresentadas levam à exploração não-ótima destes recursos, de forma que nem todas as tarefas são executadas (GERKEY; MATARIĆ, 2002). Apesar disso, Gerkey e Matarić (2004) afirmam que, “sem um modelo das tarefas que serão apresentadas e sem a opção de reatribuir tarefas aos robôs que já foram designados, é impossível construir um sistema alocador de tarefas melhor que a MURDOCH”. Essa comparação é feita em relação à performance de algoritmos gulosos de atribuição *online*, como a MURDOCH, comparado com soluções ótimas *offline*, as quais são formuladas após a apresentação de todas as tarefas. Ainda segundo Gerkey e Matarić (2004),

arquitecturas que permitem a realocação de tarefas entre os robôs “tem desempenho potencialmente melhor que a MURDOCH”.

A arquitectura foi validada experimentalmente em dois domínios diferentes: (a) para a alocação de tarefas fracamente acopladas e (b) para a tarefa cooperativa de empurrar uma caixa.

O experimento (a) consistiu em um grupo de robôs, contando com diferentes recursos (como câmeras, medidores de distância a *laser*, sensores de toque e uma câmara com visão aérea da área de trabalho dos robôs) realizando quatro diferentes tarefas SR (*single-robot tasks*): rastrear objeto, ficar de guarda, limpeza e monitorar objeto. Cada tarefa necessita de determinados recursos, de forma que apenas os robôs que contenham os recursos necessários serão avisados sobre cada tarefa. De forma aleatória, tarefas com durações aleatórias foram apresentadas ao grupo de robôs e monitoradas por um agente autônomo de tarefas, rodando em um computador separado do grupo de robôs. Como resultado, sempre que uma tarefa era anunciada e os recursos necessários estavam disponíveis, esta era executada. Sendo as tarefas anunciadas de forma aleatória, ocorreram situações em que todos os robôs capazes estavam ocupados, então houveram tarefas não alocadas por falta de recursos. Além disso, foi verificada a capacidade de escalabilidade da arquitectura em relação a comunicação entre os robôs, dada a largura de banda média utilizada na comunicação ser muito menor que a largura de banda disponível em redes sem fio atuais.

O experimento (b) consiste de uma tarefa fortemente acoplada, em que o sistema deve mostrar a capacidade de coordenar uma tarefa que necessita de mais de um robô para ser concluída. A tarefa de empurrar uma caixa necessita de três robôs, dois impulsores e um observador, que analisa o transcorrer da tarefa, avaliando a posição atual e a posição final e, assim, coordenando a movimentação dos impulsores. Além disso, a arquitectura também é validada em relação à tolerância a falhas, no caso da tarefa cooperativa, a partir das medições de trajetória da caixa, tempo de execução e taxa de sucesso da tarefa. As referências (GERKEY; MATARIĆ, 2002; GERKEY, 2003) apresentam os experimentos e seus resultados de forma mais detalhada.

3.4.2.2 Protocolo M+

Proposto por Botelho e Alami (1999), o protocolo M+³, assim como a MURDOCH, é adaptado de acordo com as premissas do CNP e baseia-se na “combinação de planejamento local e negociação para a alocação de tarefas e a reação cooperativa à emergências”. O protocolo assume que em um sistema complexo, composto por vários robôs com seus próprios sensores e atuadores, a alocação das tarefas pode ser realizada por um planejador. O planejamento realizado na arquitectura tem dois níveis, o local, planejamento realizado pelo próprio robô, e o global, planejamento que funde cada planejamento local a fim de produzir um plano que “contém ações coordenadas e cooperativas que asseguram sua execução apropriada e servem como base para a negociação” (BOTELHO; ALAMI, 1999).

A fusão dos planos de cada robô é realizada de acordo com o chamado Paradigma da

³ Botelho e Alami (2000) chamam o protocolo M+ também de M+NTA (M+ *Negotiation for Task Allocation*).

Fusão de Planos (*Plan-merging Paradigm*) (ALAMI et al., 1998). Esse paradigma tem o objetivo de resolver conflitos de utilização de recursos no aspecto temporal e quanto à eventos de início e fim de tarefa. A camada de ação (*action layer*), que é responsável pela manipulação das ações e o planejamento e execução de ações de navegação cooperativas inclui, também, esse módulo de resolução de conflitos.

Cada robô dispõe de um planejador na camada de tarefas (*task layer*), que é responsável pela negociação proposta pelo protocolo M+. Duas entidades compõem a camada: o planejador de tarefas e o supervisor de tarefas (BOTELHO; ALAMI, 1999). O planejador de tarefas gera planos de execução de acordo com as solicitações do supervisor de tarefas. A partir desses planos, o protocolo M+ calcula os custos referentes às execuções, que serão a base para a negociação entre os robôs. Acima da camada de tarefas, há a camada de missão (*mission layer*). Essa camada é responsável por gerar a lista de tarefas que serão enviadas aos robôs.

As tarefas são apresentadas aos robôs por meio de uma lista parcialmente ordenada e todos os robôs recebem a mesma missão. Por exemplo, $T_1 \prec T_5$ significa que a tarefa T_1 precede tarefa T_5 e deve ser realizada primeiro. Os robôs devem produzir seus planos de acordo com as restrições impostas pelo ordenamento parcial.

Três atividades são realizadas pelo protocolo M+: alocação de tarefas, reação cooperativa à emergências e execução de tarefas. A alocação de tarefas no protocolo refina as tarefas, a partir da negociação, e efetua a alocação das mesmas. A reação cooperativa é ativada quando, por alguma falha, um robô pede ajuda ao grupo. Essa atividade atualiza o estado do ambiente, administra a troca de informação entre os robôs e controla o replanejamento e os pedidos de ajuda (BOTELHO; ALAMI, 1999). A última atividade, execução de tarefas, é responsável pelo controle da execução de tarefas e a sincronização entre as tarefas do robô e suas ações.

Para o alocador de tarefas, “o critério de escolha será o custo dos planos elaborados pelos diferentes robôs, dependendo de suas capacidades e situações” (BOTELHO; ALAMI, 1999). A atividade pode ser melhor visualizada na Figura 3.6. Segundo Botelho e Alami (1999), são possíveis cinco estados: planejamento, avaliação, candidato, melhor candidato e inativo.

Quando há uma atualização no conjunto de tarefas disponíveis para execução, ou executáveis, o robô entra no estado de *avaliação* (8). Nesse estado, ele está apto a iniciar uma negociação. Para isso, ele invoca o planejador (1) para que seja elaborado um plano para as tarefas executáveis e o custo de cada plano seja calculado (2). Com essa informação, o robô compara seus custos com os custos oferecidos pelos demais robôs e, então, seleciona a tarefa de menor custo a qual está apto a executar e a qual seu custo é o mais baixo, comparado com as demais ofertas. Caso essa tarefa não exista, o robô entra no estado *inativo* (7). Com a tarefa selecionada, o robô entra no estado *candidato* (3). Neste caso, há duas possibilidades: (a) se nenhum outro robô R_j seleciona a tarefa, ou seja, nenhuma oferta é anunciada, o robô R_i envia uma mensagem por difusão anunciando sua “primeira oferta”; após esse anúncio, R_i é o responsável por todas as mensagens relacionadas àquela tarefa e entra no estado *melhor candidato* (5); e (b) se a tarefa já foi anunciada por outro robô R_j , o robô R_i ainda pode anunciar uma oferta melhor, até que R_j inicie a execução da tarefa; R_i envia sua oferta e a resposta pode ser

positiva (R_i está agora no estado *melhor candidato* para esta tarefa) ou negativa, outro robô deu um melhor lance; no segundo caso, o robô R_i volta para o estado de *avaliação* (4) para escolher nova tarefa.

A partir do estado *melhor candidato* o robô pode iniciar a execução da tarefa ou, ainda, receber uma oferta melhor de outro robô para a execução da mesma. Em todas as situações, o robô sai desse estado, entretanto, na segunda, ele retorna ao estado de *avaliação* (6). Outra possibilidade é o robô abandonar a tarefa por uma falha ou por uma reação cooperativa, que também o desloca para o estado de *avaliação*.

A reação cooperativa à emergências é a reação do protocolo M+ à eventuais falhas. Quando um robô falha, primeiramente, ele busca replanejar a execução da tarefa e, no caso de o novo planejamento não ser possível, ele envia um *pedido de ajuda* aos demais membros da equipe de robôs (BOTELHO; ALAMI, 1999). O pedido de ajuda deve conter as informações necessárias para que os demais robôs possam calcular seu custo em atender tal pedido. Para isso, o robô que requer ajuda envia informações de atualização do seu ambiente que são relevantes à execução da tarefa, por exemplo, informando o estado dos recursos necessários à execução da tarefa.

A elaboração do novo planejamento pelos robôs que possivelmente atenderão ao pedido de ajuda é realizado de forma a incluir a tarefa solicitada em seu planejamento prévio. Esse novo planejamento gera um novo custo. Ao robô requerente, o custo é apresentado e o robô com menor custo é aceito para atender ao pedido de ajuda. Após o aceite do novo robô para a execução da tarefa, o robô falho a abandona. Portanto, a partir de sua falha, o robô fica apenas responsável por atribuir a tarefa à outro membro capaz de executá-la, ou seja, responsável por todas as mensagens relacionadas à determinada tarefa.

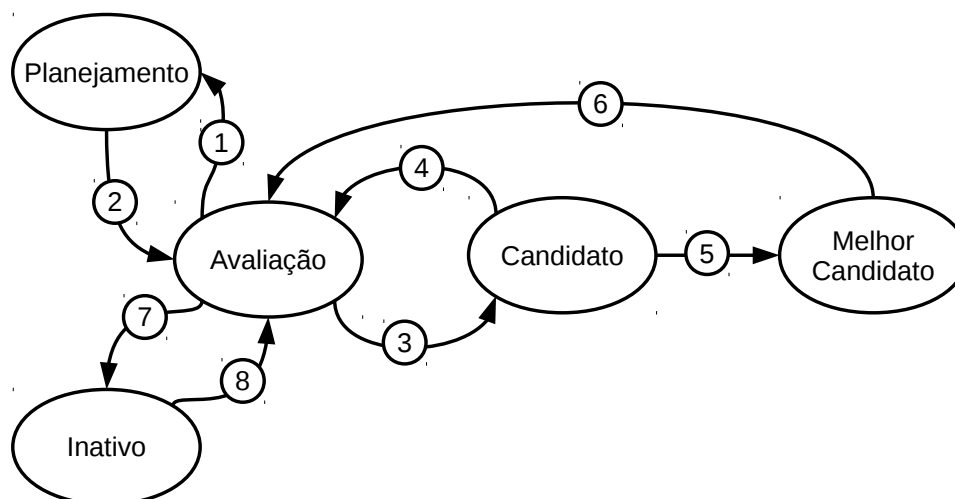


Figura 3.6 – Diagramas de estado do protocolo M+ para a alocação de tarefas. Fonte: adaptado de Botelho e Alami (1999).

A atividade de execução das tarefas é encarregada de controlar tal execução, sincronizando as ações e as tarefas dos robôs. A sincronização das tarefas se dá pelo envio de duas

mensagens por difusão, uma mensagem de início de execução de tarefa e uma mensagem de término de execução de tarefa. A mensagem de início possibilita aos demais robôs a atualização do seu quadro de tarefas executáveis. Já as mensagens de término permitem o início da execução de tarefas sucessoras, de acordo com o ordenamento da missão. A sincronização de ações ocorre quando há uma reação de cooperação. Assim, os robôs devem sincronizar as mensagens enviadas acerca da tarefa que será reatribuída.

A coerência na execução do protocolo é obtido de acordo com as premissas definidas por (BOTELHO; ALAMI, 1999):

- Como o tráfego de mensagens é controlado, apenas um robô emitirá a *primeira oferta*.
- O robô que emite a *primeira oferta* é responsável por todas as mensagens relativas à tarefa. Dessa forma, as informações relativas à tarefa permanecem coerentes durante a negociação, dispensando o uso de controle de mensagens.
- Cada robô mantém sua própria coerência, por exemplo, executando apenas uma atividade de planejamento por vez.

O protocolo foi implementado em um grupo de três robôs manipuladores para tarefas de transporte de cargas em (BOTELHO; ALAMI, 1999). A comunicação entre os robôs se dá via protocolo TCP/IP. Um total de dez tarefas em um ordenamento parcial foi apresentado aos robôs. Experimentalmente, foram verificadas as características de alocação de tarefas via negociação e a reação cooperativa entre robôs, com a reatribuição de tarefa após falha no primeiro robô executor. Os resultados detalhados da aplicação são mostrados em (BOTELHO; ALAMI, 1999).

3.4.2.2.1 M+CTA

O protocolo M+ foi expandido para uma arquitetura genérica para cooperação multirrobo em dois trabalhos posteriores: Botelho e Alami (2000) e Botelho e Alami (2001).

A arquitetura M+CTA apresentada em (BOTELHO; ALAMI, 2001) (M+ *Cooperative Task Allocation*) é um complemento à arquitetura M+NTA, introduzindo o conceito de *mecanismo* para generalizar a descrição de recursos. Mecanismos são estruturas de dados que definem a utilização de um determinado dispositivo. Em (BOTELHO; ALAMI, 2001) três formas de mecanismos são apresentadas: estados iniciais e finais conhecidos, ações sequenciais e as *regras sociais*. As ações sequenciais são as ações que um robô deve tomar para sair de um estado inicial conhecido e atingir um estado final conhecido. As regras sociais são as limitações impostas ao uso de mecanismos.

A arquitetura, segundo Botelho e Alami (2000), preocupa-se com a cooperação para o aprimoramento dos planos de cada robô de forma a gerar um aumento na eficiência global do sistema. Para isso, são consideradas três características (BOTELHO; ALAMI, 2000; BOTELHO; ALAMI, 2001):

Ação oportunista de realocação Durante a execução de sua tarefa, o robô pode, de forma oportunista, detectar que será benéfico para o desempenho global se ele puder executar outra tarefa, a qual a execução foi previamente planejada por outro robô.

Supressão de redundâncias Com a combinação dos planos de cada robô, mesmo que vários robôs tenham planejado ações que atingem o mesmo estado no ambiente, ou seja, a execução das mesmas tarefas, essa característica fornece capacidades de inferência que permite aos robôs a escolha de quem irá executar a tarefa, evitando execuções redundantes.

Ações aditivas/incrementais Ações aditivas permitem aos robôs a percepção de que uma ação, originalmente planejada para um robô, pode ser realizada por vários robôs e essa ação conjunta beneficia o desempenho global.

O processo de realização da tarefa (*task achievement process*) envolve três atividades básicas: (a) o planejamento local que produz um planejamento *mono-robô* combinável (*mergeable* (BOTELHO; ALAMI, 2000)), (b) a negociação que adapta o planejamento ao contexto multirrobo e (c) a execução efetiva do planejamento (BOTELHO; ALAMI, 2000). Apesar de serem etapas complementares, as mesmas podem ser executadas em paralelo, visto que têm horizontes temporais diferentes. Por exemplo, o planejamento local é totalmente interno ao robô, enquanto a negociação e a própria execução do planejamento ocorre em períodos específicos. A atividade de negociação de planejamentos permite aos robôs a coordenação dos planos individuais de forma a cooperar e evitar o conflito no uso de recursos a fim de melhorar o desempenho global do sistema (BOTELHO; ALAMI, 2000).

A contribuição da arquitetura é a introdução do conceito de mecanismos, utilizados para o aprimoramento dos planos gerados nos robôs de forma cooperativa. Segundo Botelho e Alami (2000), “essa atividade cooperativa baseia-se na satisfação comum de um conjunto de limitações expressas nos termos de [...] *regras sociais*⁴, as quais foram incorporadas a fim de que possam ser produzidos planos facilmente combináveis”. As três regras sociais definidas em (BOTELHO; ALAMI, 2000; BOTELHO; ALAMI, 2001) são:

tempo Essa regra define uma quantidade de tempo s a qual um estado é mantido *verdadeiro*.

Após esse intervalo de tempo, um estado sucessor é proposto para que a regra não seja violada. Por exemplo, o estado pode ser um determinado recurso, como um motor, que é ligado. Esse estado é *verdadeiro* por s unidades de tempo. A fim de que a regra não seja violada, após o tempo s , um novo estado é proposto: o recurso motor é desligado.

quantidade Define a quantidade de robôs que pode utilizar determinado recurso. Além disso, é proposta uma ação, caso o número de robôs exceda o número máximo, para que a regra não seja violada.

⁴ O termo utilizado pelos autores em (BOTELHO; ALAMI, 2000) não tem o mesmo significado nem os mesmos fundamentos que as *regras de escolha social*, apresentadas no capítulo 2, tratando apenas de limitações que auxiliam nas atividades de planejamento e negociação

conclusão Define o objetivo final de cada ação do robô. Para que a ação possa ser dada como concluída, a regra deve ser atendida. Além disso, a regra garante um estado final conhecido, “permitindo ao planejador prever o estado final de uma atribuição” e, consequentemente, o estado inicial da atribuição subsequente (BOTELHO; ALAMI, 2001).

A cada regra social é associado um valor escalar chamado de *nível de obrigatoriedade*. Esse valor escalar é levado em consideração no momento do planejamento da execução das tarefas. Segundo Botelho e Alami (2000), o parâmetro faz a distinção entre regras sociais que devem ser sempre respeitadas para a geração de planos combináveis daquelas que podem ser adiadas, planejadas mas não necessariamente executadas.

A elaboração do plano considera os estados finais propostos em cada regra social como objetivos obrigatórios. Entretanto, dependendo dos níveis de obrigatoriedade de cada regra, os estados finais podem ser considerados como objetivos atuais do robô ou como objetivos que o robô buscará satisfazer, porém, em outras etapas de planejamento (BOTELHO; ALAMI, 2000). No segundo caso, o nível de obrigatoriedade das regras sociais é baixo.

A arquitetura foi implementada em uma simulação de transporte de objetos e limpeza de leitos em um ambiente hospitalar, as referências (BOTELHO; ALAMI, 2000; BOTELHO; ALAMI, 2001) apresentam o experimento detalhadamente.

3.4.3 Outras Arquiteturas Estudadas

Na seção 3.4.4 e no capítulo 5 o enfoque será dado nas arquiteturas ALLIANCE, MURDOCH e M+CTA, entretanto, outras arquiteturas foram estudadas ao longo do trabalho de pesquisa.

Broadcast for Local Eligibility - BLE Essa arquitetura proposta por Werger e Matarić (2000) aplica o princípio de comportamentos arbitrados pelas portas (*port-arbitrated behaviors* - PAB) para o controle de um SMR.

O PAB, inicialmente proposto para a interação entre os comportamentos do mesmo robô, utiliza as chamadas *portas* para o registro de dados, que podem ser desde simples números inteiros até estruturas complexas de dados. As portas de entrada e saída de cada comportamento são conectadas entre si, formando caminhos unidirecionais de trocas de dados. A transmissão dos dados, entretanto, pode ser modificada dependendo dos tipos de conexões entre as portas. Essas conexões podem ser inibidoras, supressivas ou imperiosas, transmitindo ou impedindo a transmissão de mensagens através das conexões dependendo da configuração das mesmas.

A arquitetura BLE utiliza uma extensão da linguagem C, a AYLLU, proposta por Werger (2000) explorando os princípios do PAB para desenvolvimento do controle distribuído para grupos de robôs móveis. Na AYLLU os comportamentos são conectados, e arbitrados, tanto localmente, ou seja, no próprio robô, quanto globalmente por mensagens de difusão enviadas via redes IP. Além disso, a proposta inclui novos tipos de portas ao paradigma PAB, de forma a

tratar melhor os dados recebidos e permitir a escalabilidade do sistema.

O mecanismo de alocação de tarefas da arquitetura BLE compara um valor local de elegibilidade, ou seja, a capacidade para a execução de determinada tarefa pelo próprio robô, com a melhor elegibilidade determinada por um comportamento similar em outro robô (WERGER; MATARIĆ, 2000). Quando um robô tem a melhor elegibilidade de determinado comportamento para a execução de determinada tarefa, ele inibe os comportamentos semelhantes nos demais robôs da equipe, reivindicando, assim, a tarefa. Se o robô que reivindicou a tarefa falha, a mesma encontra-se livre para ser reivindicada por outro membro do time, após a avaliação das elegibilidades.

É importante ressaltar que todo o processo de comparação de elegibilidades ocorre em cada robô, de forma distribuída. Não há um mecanismo central para tal comparação. Por meio das mensagens de difusão, os robôs têm as informações necessárias para determinar qual tarefa será reivindicada e quais comportamentos devem ser inibidos, devido à reivindicações de outros robôs.

MONAD Segundo Vu et al. (2003), MONAD é uma arquitetura flexível orientada à programação e à execução da equipe de agentes, integrando controle hierárquico baseado em comportamentos, mecanismos de coordenação multiagente e serviços de alocação de tarefas. A arquitetura congrega um ambiente *offline* de programação e projeto da equipe de agentes com um mecanismo de coordenação *online* que pode executar diferentes modelos de controle da equipe de agentes automaticamente em uma estrutura baseada em comportamento. O objetivo de tal organização é permitir uma rápida implementação de diferentes arquiteturas de sistemas multiagentes, além da possibilidade de flexibilidade na construção de novas equipes (MOSTEO, 2010).

O desenvolvimento da arquitetura leva em consideração *scripts* e códigos *offline* determinados pelo projetista. Os *scripts* são arquivos escritos em formato específico da MONAD, e são responsáveis pela descrição da equipe de agentes e pela programação da mesma. Os códigos são responsáveis pela mediação e pelo comportamento da execução. Esses arquivos são as entradas do mecanismo de sincronização de coordenação da arquitetura, que é capaz de agir como um sistema de coordenação distribuída para toda a equipe. Cada agente executa uma cópia idêntica desse mecanismo.

O *script* da programação da equipe especifica a hierarquia dos comportamentos e a interação entre eles. Para os comportamentos de baixo nível definidos na ALLIANCE há comportamentos similares que são aqui chamados de comportamentos filhos. Esses comportamentos possuem sub-objetivos, dos quais os comportamentos internos (equivalentes aos comportamentos de alto-nível da ALLIANCE) dependem para completar seus próprios objetivos.

Um método de mediação ou mediador é definido como um algoritmo o qual um grupo de agentes utiliza quando é necessária uma transição entre comportamentos internos (VU et al., 2003). Essa transição entre os comportamentos internos é responsável pela alocação de tarefas.

Entretanto, a arquitetura apresenta fragilidade quanto à falhas de comunicação (VU

et al., 2003), o que representa uma carência de capacidades de tolerância à erros (MOSTEO, 2010).

Automated Synthesis of Multi-robot Task solutions through software Reconfiguration - ASyMTRe Essa arquitetura contrasta com as arquiteturas vistas anteriormente dada a abordagem proposta. Tipicamente, as arquiteturas para problemas MRTA necessitam de uma predefinição manual do projetista de como a equipe de robôs irá executar a missão e cada tarefa que a compõe. A árvore de tarefas descrevendo a decomposição das tarefas para a equipe de robôs em sub-tarefas é definida previamente pelo projetista. Ou seja, o controle precisa apenas definir quem executará as tarefas. A ASyMTRe, segundo Tang e Parker (2005), automatiza tanto quem irá executar quanto como o irá executar, sem que seja definido pelo projetista antecipadamente. Segundo Zhang (2012), a arquitetura foi desenvolvida para tratar de tarefas multirrobo fortemente acopladas e foi a primeira arquitetura a possibilitar a capacidade autônoma de compartilhar dados nas camadas computacional e de sensoriamento.

Para os autores, aplicações que envolvem robôs heterogêneos em SMR têm dois problemas principais: (a) determinar como serão divididos os recursos sensoriais e de percepção entre os membros heterogêneos da equipe e (b) determinar os comportamentos da equipe apropriados para a conclusão da tarefa quando essa definição depende da coleção de robôs disponíveis, de acordo com suas características (TANG; PARKER, 2005).

O formalismo da arquitetura baseia-se em tipos de informações, as quais serão compartilhadas entre os robôs da equipe, de forma a maximizar a utilidade do grupo de robôs para uma determinada tarefa. Além disso, são modelados blocos básicos que representam as capacidades individuais dos robôs. Esses blocos são divididos em sensoriamento do ambiente, esquema perceptivo, esquema motor e esquema de comunicação.

O sensoriamento do ambiente tem como entrada a saída de um sensor físico e tem apenas uma saída, mesmo que essa saída represente um conjunto de dados. Os esquemas perceptivos recebem informações do sensoriamento do ambiente, podendo ter várias entradas, e tem uma única saída com um conjunto de características, como distância e ângulo. Outras entradas possíveis no esquema perceptivo são as entradas provenientes de esquemas de comunicação. Esses esquemas são responsáveis pelo compartilhamento das informações na arquitetura. As saídas dos esquemas perceptivo e de comunicação são entradas para o esquema motor, responsável pelo acionamento dos atuadores do robô, no qual as saídas têm informações do tipo *velocidade*, por exemplo. As conexões entre os esquemas seguem as regras estabelecidas em (TANG; PARKER, 2005) e podem ser feitas quando as informações de saída são do mesmo tipo das informações de entrada do bloco subsequente.

Para a formação de coalizões, a arquitetura busca todas as formas possíveis para conectar esquemas diferentes de forma a ativar determinado esquema motor nos robôs para a conclusão de uma tarefa (ZHANG, 2012). E a ativação de cada esquema está relacionado com um grau de aptidão da solução relativa ao robô. Esse valor é chamado de utilidade e é denotado pela Equação 3.9 (TANG; PARKER, 2005), onde $\mu(i)$ é a utilidade do robô R_i para j -ésimo

sistema sensorial ou computacional que o mesmo necessita utilizar. O termo P_j significa a probabilidade de sucesso e o termo C_j o custo de sensoriamento de dado sensor, ambos para o j -ésimo sistema sensorial ou computacional utilizado. O peso w combina a probabilidade e o custo.

$$\mu(i) = \sum_j (w \times 1/C_j + (1 - w) \times P_j) \quad (3.9)$$

A solução da arquitetura visa organizar a equipe em subequipes de forma que cada uma delas contribua com o objetivo final. Além disso, a utilidade para cada robô em cada subequipe deve ser maximizada para cada tarefa.

Zhang (2012) afirma que, mesmo que a arquitetura aumente a flexibilidade para SMR, ainda há problemas a serem tratados. Um dos principais problemas ocorre quando o sensoriamento não consegue recuperar a informação em uma determinada situação. Por exemplo, quando um robô que não possui capacidade de localização e um que a possui estão na mesma localização, entretanto, o primeiro está fora do campo de visão, do alcance, do segundo (ZHANG, 2012). Nesta situação, a coalizão formada é inviável, pois a tarefa pode não ser cumprida ou ter seu custo elevado.

Dynamic role assignment A arquitetura *Dynamic Role Assignment*, desenvolvida por Chaimowicz, Campos e Kumar (2002), é uma arquitetura baseada em mercado em que os robôs utilizam funções de utilidade para a atribuição de papéis na execução da missão. Os papéis exercidos pelos robôs são, na verdade, as tarefas a serem executadas. Por isso, a atribuição de papéis é a própria alocação de tarefas. A arquitetura é comparada com MURDOCH, M+ e leilões de única oferta em termos de complexidade de comunicação. Nessa arquitetura, em que é utilizado um variante do CNP, um robô líder envia pedidos solicitando assistência até que um número suficiente de robôs se voluntarie para uma tarefa cooperativa. A arquitetura utiliza um autômato híbrido para modelar os papéis e as transições possíveis entre eles.

Com o chamado do robô líder, os robôs podem ter a preferência na execução de tarefas alteradas, por exemplo, abandonar a execução de uma tarefa devido a mensagem do robô líder. Entretanto, apenas farão a transição se a utilidade da nova tarefa for alta o suficiente para compensar o custo envolvido na troca (GAGE, 2004). Segundo Chaimowicz, Campos e Kumar (2002), a definição das funções ótimas de utilidade é um problema de difícil resolução.

TraderBots Proposta inicialmente por Stentz e Dias (1999) e detalhada por Dias (2004), a arquitetura tem o objetivo de unir o melhor das abordagens centralizadas e descentralizadas. Segundo Dias (2004), a arquitetura é inerentemente distribuída, mas, pode ser centralizada em subgrupos para aumentar sua eficiência. Essa arquitetura também tem por base o CNP.

Na arquitetura, os robôs são modelados como agentes egoístas e o time de robôs é modelado como uma economia. O objetivo da equipe é completar as tarefas minimizando o custo global (DIAS, 2004). São determinadas funções de custo e de retribuição, as quais calculam o custo de cada esquema possível de execução das tarefas e transformam os resultados possíveis

da execução das tarefas em valores de retribuição, respectivamente. Os robôs utilizam essas funções para negociar a execução de uma tarefa.

Segundo [Dias \(2004\)](#), a arquitetura trata algumas das características desejáveis para um sistema de coordenação multirrobo de sucesso, como a robustez, a capacidade de lidar com condições dinâmicas, a capacidade de lidar com o surgimento de novas tarefas, flexibilidade, aprendizado, entre outras. Entretanto, “a abordagem não é uma solução ótima para todos os domínios de aplicação multirrobo. Podem haver dificuldades na determinação apropriada de funções de custo e retribuição para representar alguns problemas com acurácia” ([DIAS, 2004](#)).

Information Quality-ASyMTRe A arquitetura IQ-ASyMTRe é uma extensão da ASyMTRe, apresentada previamente, proposta por [Zhang \(2012\)](#). Essa extensão tem o objetivo de tratar alguns dos problemas da arquitetura original, como, por exemplo, a formação de coalizões inviáveis. Além disso, um novo tratamento às informações é utilizado na conversão dos tipos de informações.

A alocação de tarefas utiliza uma abordagem baseada em mercado. Um leiloeiro anuncia as tarefas aos robôs, que calculam e submetem as ofertas para as coalizões possíveis. O leiloeiro, então, determina as ofertas ganhadoras e atribui as tarefas à essas coalizões.

Para tratar do problema de formação de coalizões inviáveis, a arquitetura trabalha com a recuperação de informações. Caso nenhuma coalizão possa recuperar as informações de uma tarefa, então essa tarefa é não executável. Há, também, a definição de coalizão local, que trata do robô que, diretamente, fornece informações a outro robô para a ativação dos esquemas de sensoriamento ou de atuadores ([ZHANG, 2012](#)). Essas informações locais dão origem à uma potencial coalizão-solução local. A partir das coalizões encontradas, o algoritmo escolhe a coalizão com o menor custo de execução, alocando, assim, a tarefa.

3.4.4 Comparação

Uma breve comparação será feita para as arquiteturas que são o foco deste trabalho: ALLIANCE, MURDOCH e protocolo M+. Alguns pontos de comparação foram levantados: o processo decisório para a alocação de tarefas, métodos de tolerância a falhas, comunicação utilizada, descrição das tarefas, conhecimento do ambiente por parte dos robôs e classificação de acordo com a Taxonomia de [Gerkey e Mataric \(2004\)](#).

A alocação de tarefas em todas as três arquiteturas é realizada de forma descentralizada. Na ALLIANCE, é definida com base nas motivações internas dos robôs e na comunicação implícita entre eles. Se o grau de motivação atinge um determinado valor de limiar, um comportamento é ativado e, então, uma tarefa começa a ser executada. Quando um robô transmite uma mensagem informando o início da execução de determinada tarefa, os demais são satisfeitos, e a motivação interna para esta determinada tarefa passa a evoluir de uma forma mais lenta. Essa evolução é importante para que, caso o primeiro robô falhe, os demais assumam a execução da tarefa após um intervalo de tempo apropriado. Ainda, nenhum tipo de negociação

é realizado para definir o robô que executará a tarefa. Sendo os robôs heterogêneos, cada um terá uma evolução própria das motivações internas. E caso a extensão L-ALLIANCE tenha sido utilizada, os robôs têm conhecimento das capacidades dos demais robôs do grupo e, portanto, adaptam suas motivações de acordo com essas capacidades individuais. Dessa forma, o robô com melhor desempenho em determinada tarefa terá mais chance de executá-la. A realocação de tarefas já atribuídas é possível na arquitetura, dependendo apenas da evolução da motivação interna de cada robô, especificamente, a impaciência e a aquiescência.

A negociação está presente nas outras arquiteturas discutidas, porém, com métodos diferentes. A MURDOCH apresenta um mecanismo de alocação em que os leilões são baseados em apenas uma rodada de ofertas, com a primeira oferta de cada robô envolvido (GERKEY; MATARIĆ, 2002). Não há negociação posterior, mesmo se o ambiente dinâmico apresentar alguma variação. O contrato de execução da tarefa é fechado com o robô que apresentou a melhor oferta no período da rodada de ofertas. O monitoramento da execução da tarefa é constante. Se um progresso suficiente é demonstrado, o contrato é renovado por mais um intervalo de tempo determinado e o robô contratado envia uma mensagem de reconhecimento da renovação. Isto ocorre até a tarefa ser finalizada. Entretanto, a arquitetura não permite a realocação de tarefas, dado, por exemplo, pela disponibilidade de um robô com melhor capacidade para execução de uma tarefa já atribuída. A mudança no executor ocorrerá apenas se o contrato não for cumprido, por exemplo, devido a uma falha.

O protocolo M+ apresenta um mecanismo de negociação que permite mais de uma rodada de ofertas, gerando uma maior flexibilidade na alocação em relação às mudanças de estado do sistema. Como todos os robôs recebem a mesma missão, seu planejamento envolve todas as tarefas da mesma. Entretanto, de acordo com a alocação de tarefas dos demais robôs, um novo planejamento será executado. Com mudança no planejamento individual, a negociação de uma tarefa poderá ocorrer em mais de uma rodada. Isso ocorre, pois há um intervalo de tempo entre a atribuição da tarefa a um robô (estado de *melhor candidato* na Figura 3.6) e a execução efetiva da tarefa. A realocação de tarefas é possibilitada devido à solicitação de ajuda em situações de emergência, o que leva a um novo planejamento por parte dos robôs que se disponibilizam a atender ao pedido. Entretanto, essa situação não pode ser entendida como a execução da tarefa por dois ou mais robôs, e sim como um mecanismo de realocação. Após a negociação para a execução da tarefa, o robô que a estava executando previamente a abandona, buscando executar outra tarefa. Não é um mecanismo de formação de coalizão.

Os mecanismos de alocação de todas as arquiteturas em questão utilizam um valor escalar calculado, a utilidade, como referência para a atribuição. Em relação à utilidade calculada, a arquitetura ALLIANCE trata esse problema usando uma espécie de utilidade variável. Esse valor é a motivação interna dos robôs, especialmente o grau de impaciência e de aquiescência, que variam inerentemente com o tempo (PARKER, 1998; PARKER, 1995; BASTOS; RIBEIRO; SOUZA, 2008). Em contraste, a oferta apresentada com valor de aptidão da arquitetura MURDOCH depende das métricas utilizadas pelo robô, como somas simples ou ponderadas de variáveis, tais como medições de sensores e estados do robô, mas nem sempre levando o tempo em

consideração. Seguindo a mesma linha, o protocolo M+ apresenta para cada tarefa um custo calculado durante o planejamento das ações.

As arquiteturas têm métodos diferentes para prevenir falhas dos membros da equipe de robôs. A ALLIANCE, denominada pela própria autora como uma arquitetura totalmente distribuída e tolerante à falhas (PARKER, 1998), utiliza a motivação interna (aquiescência) para evitar que um robô fique travado em uma determinada tarefa e, paralelamente, a impaciência de outro robô para que uma tarefa seja executada no menor tempo possível. Além disso, falhas parciais também são prevenidas. Um robô que tenha um de seus recursos em falha, terá sua motivação reduzida a comportamentos que necessitem do recurso. Os parâmetros de um comportamento que não pode ser habilitado podem ser definidos pelo programador, ou aprendidos pelos robôs, através da L-ALLIANCE.

O principal mecanismo para prevenção de falhas na MURDOCH é a renovação do contrato de execução da tarefa. Com o monitoramento periódico, uma falha no robô executor e/ou a falha no envio da mensagem de reconhecimento de renovação de contrato gera um novo leilão, pois o leiloeiro entende que a tarefa pode ser redesignada. Do mesmo modo, um novo leilão pode ocorrer se parâmetros de execução da tarefa não estiverem em níveis satisfatórios. Dessa forma, com o término do contrato sem renovação, o robô executor irá parar de executar a tarefa e outros robôs podem apresentar suas ofertas para iniciar a execução da mesma.

No protocolo M+, caso um robô falhe na execução de uma tarefa, ele recorrerá à camada de planejamento para encontrar um novo conjunto de ações para que a tarefa seja executada. Se isso não for possível, há um pedido de ajuda ao grupo, quando uma mensagem é transmitida ao grupo com os atributos pertinentes daquela tarefa, para que os demais consigam calcular o custo em ajudar o determinado robô. A falha total de um robô, e a consequente não execução de uma tarefa, gera a falta de atualização das tarefas executáveis e, portanto, a tarefa poderá entrar no planejamento dos demais robôs para o retorno da execução.

A comunicação, parâmetro importante para todas as arquiteturas, ocorre por meio de mensagens de difusão (*broadcast*). Entretanto, apresentam métodos distintos. A ALLIANCE conta com dois parâmetros de transmissão: ρ_i , que determina a taxa de transmissão de mensagens, e τ_i , que determina o período de tempo o qual um robô pode ficar sem receber mensagens de outro robô da equipe antes de decidir que este último parou de funcionar. O segundo parâmetro incrementa o nível de tolerância à falha. As mensagens recebidas de outros robôs da equipe afetam a motivação do robô através da Equação 3.3, o que pode alterar seu comportamento ou inibir um comportamento. Nas premissas da arquitetura a falha na comunicação é prevista, pois o meio de comunicação pode estar indisponível. Além disso, a forma de comunicação da arquitetura permite formas de comunicação mais sofisticadas, como o reconhecimento das ações entre os robôs, de forma a não ser necessária a troca de dados.

A comunicação na MURDOCH é anônima e baseada no conceito *editor/assinante*. Com isso, os tópicos publicados são assinados apenas pelos robôs que dispõem dos recursos necessários a cada tópico, recebendo informações sobre o mesmo. Dessa forma, os editores publicam uma mensagem de anúncio de tarefa apenas nos tópicos com recursos relacionados àquela ta-

refa. Os membros assinantes desses tópicos respondem à essa mensagem, sem a comunicação entre os demais assinantes. Perdas de mensagens e perdas de conexão dos robôs por estarem fora do alcance de comunicação são previstos e permitidos.

Enquanto na MURDOCH as mensagens são centradas nos recursos, o protocolo M+ baseia suas mensagens nas tarefas, sendo transmitida na mensagem o custo para execução de determinada tarefa (durante uma negociação). No caso de um pedido de ajuda, o robô requisitante envia os atributos relevantes daquela tarefa, mostrando aos demais o estado atual de seu ambiente, para que os demais possam calcular o custo das novas ações a serem executadas.

Um aspecto importante para o controle da arquitetura é a forma de apresentação das tarefas, ou da missão, aos robôs. A ALLIANCE tem as tarefas descritas através do repertório de comportamentos de alto-nível programados nos robôs. Para que haja a ordenação das tarefas, a motivação dos robôs evolui de acordo com os parâmetros pré-estabelecidos e impede de tarefas posteriores serem realizadas antes de tarefas que causem um aumento maior na motivação do robô, significando prioridade na execução. De alguma forma, a missão é apresentada a todos os robôs simultaneamente. Entretanto, dadas as especificidades de cada robô, dois comportamentos de descrição diferente podem executar a mesma tarefa. Ainda considerando características diferentes nos robôs, alguns podem não estar aptos a executar determinadas tarefas, por falta de recursos, por exemplo.

O protocolo M+ apresenta as tarefas da missão parcialmente ordenadas a todos os robôs da equipe, simultaneamente. O ordenamento parcial é considerado no planejamento para a definição de tarefas predecessoras e sucessoras de forma a estabelecer as tarefas disponíveis para execução. A arquitetura MURDOCH é classificada por [Gerkey e Matarić \(2004\)](#) como uma arquitetura de alocação de tarefas *online*. Ou seja, não são definidas todas as tarefa da missão de forma simultânea. As tarefas são incluídas na arquitetura por um sistema dedicado ou pelo programador. Assim que a tarefa é apresentada, ela é atribuída para o robô mais capaz, de acordo com o Algoritmo 3.1.

Como visto, as arquiteturas têm seu foco na alocação das tarefas e não na descrição ou desmembramento das mesmas. Essa etapa deve ser realizada por outra ferramenta de software dedicada ou pelo próprio projetista.

Sobre o conhecimento do ambiente, nenhuma das arquiteturas dispõe de um conhecimento centralizado e completo do ambiente e nenhum robô tem tal conhecimento prévio. Na ALLIANCE, o mapeamento do ambiente pode ser definido como um comportamento de alto-nível, por exemplo. Entretanto, os robôs têm conhecimento de seus companheiros de equipe. Na MURDOCH, o conhecimento do ambiente depende do recurso de cada robô, impactando na elegibilidade para as tarefas que envolvem recursos determinados. Os robôs não têm conhecimento dos demais robôs nem de seus recursos e não levam isso em consideração nas negociações dos leilões. O conhecimento do ambiente no Protocolo M+ é individual, ou seja, cada robô tem seu “mundo”. Os robôs são conscientes dos demais robôs da equipe e utilizam as informações trocadas para atualizar suas informações do ambiente.

O último ponto a ser discutido, é a classificação de cada arquitetura segundo a taxono-

mia para problemas MRTA proposto em (GERKEY; MATARIĆ, 2004). As arquiteturas discutidas têm tarefas que necessitam de um único robô (SR - *single-robot task*) para execução. O experimento (b), apresentado na seção 3.4.2.1, introduz uma missão fortemente acoplada, necessitando a cooperação entre múltiplos robôs. Entretanto, entendendo-se cada papel como uma tarefa separada (*observador, impulsor da esquerda e impulsor da direita*), pode-se concluir que a divisão de tarefas da missão leva à tarefas de único robô. Os robôs utilizados nas simulações e nas implementações físicas das arquiteturas podem executar apenas uma tarefa por vez. Dessa forma, os robôs são de tarefa única (ST - *single-task robots*). Já o tipo de atribuição varia de acordo com a arquitetura.

A alocação instantânea de tarefas (IA) é uma característica da arquitetura MURDOCH, que implementa um algoritmo de alocação *online* (GERKEY; MATARIĆ, 2002; LIU, 2014). A ALLIANCE atende tanto à atribuições instantâneas quanto à atribuições escalonadas no tempo, ou pré-agendadas (GERKEY; MATARIĆ, 2004; PARKER, 2008). Segundo Gerkey e Matarić (2004), o caso mais simples de alocação instantânea (IA) a arquitetura aloca as tarefas de forma iterativa de forma gulosa até que a missão seja finalizada. Ou seja, a partir das informações do ambiente e da troca de mensagens entre os robôs, as tarefas são alocadas aos robôs cujas motivações atinjam os determinados limiares de ativação do comportamento. Do ponto de vista do planejamento das tarefas ao longo do tempo, a alocação escalonada no tempo (TA), o Problema da Eficiência da ALLIANCE é um variante do problema MRTA (GERKEY; MATARIĆ, 2004), apresentado na seção 3.4.1.1. Nesse caso, as tarefas devem ser alocadas de forma a minimizar a função objetivo, ou o tempo de execução ou a energia gasta pelos robôs. Finalmente, o Protocolo M+ também trata de alocações IA e TA. Nesse caso, os robôs conhecem previamente as tarefas da missão em um ordenamento parcial. Dessa forma, para a alocação IA, os robôs realizam negociações iterativas para definir a atribuição das tarefas (GERKEY; MATARIĆ, 2004). Entretanto, os robôs podem negociar uma tarefa antecipadamente. Os módulos de alocação de tarefas, reação cooperativa à emergências e execução de tarefas podem ser executados paralelamente. Dessa forma, dá a característica de alocação TA para a arquitetura (BOTELHO; ALAMI, 1999). Em resumo, a arquitetura MURDOCH ataca problemas ST-SR-IA e as arquiteturas ALLIANCE e Protocolo M+ problemas ST-SR-IA e ST-SR-TA.

3.5 Discussão

Para condensar a classificação exposta neste capítulo, a Tabela 4 classifica as arquiteturas aqui apresentadas de acordo com a taxonomia de sistemas multirrobo, apresentada por Farinelli, Iocchi e Nardi (2004) e a Tabela 5 identifica a taxonomia das arquiteturas citadas neste trabalho de acordo com a taxonomia de problemas MRTA, segundo Gerkey e Matarić (2004).

De acordo com Tabela 4, todas as arquiteturas apresentadas têm a cooperação intencional, o que significa que nenhuma das arquiteturas trabalha com enxame de robôs. Além disso, nenhuma das arquiteturas têm sua organização centralizada, reafirmando as vantagens de sis-

temas distribuídos. O tipo de comunicação mais frequente, por difusão, também demonstra a busca pela robustez das arquiteturas, uma vez que a comunicação pode apresentar falhas.

Outro traço importante é o número de robôs que compõe as equipes. Esses valores foram retirados dos experimentos e simulações realizadas nos trabalhos citados. É importante notar que conforme a equipe aumenta, as soluções tornam-se mais complexas. Além disso, com o aumento no número de robôs no ambiente, a interferência entre os robôs também aumenta, o que pode gerar uma menor eficiência.

A Tabela 5 mostra a predominância de tarefas que necessitam um único robô para sua execução, dentre as arquiteturas estudadas. Essa consideração simplifica o controle de alocação de tarefas. As arquiteturas que tratam de tarefas multirrobo têm seu foco na formação de coalizões para a execução destas. Importante ressaltar que na arquitetura *Dynamic Role Assignment*, apesar de tratar de tarefas que necessitam de um número igual ou maior que dois robôs para a execução de cada tarefa, pode-se simplificar considerando que cada papel equivale a uma tarefa (CHAIMOWICZ; CAMPOS; KUMAR, 2002). Assim, cada robô que executa seu papel cumpre uma tarefa individualizada.

As colunas da Tabela 5 que classificam quanto ao tipo de robô mostram a complexidade de se desenvolver um robô com a capacidade de execução de mais de uma tarefa simultaneamente. As arquiteturas apresentadas trabalham com robôs que executam uma tarefa por vez, mesmo que suas capacidades o permitam executar várias tarefas diferentes. Além disso, o tipo de alocação predominante também mostra como o planejamento pode aumentar a complexidade do sistema. A alocação de tarefas escalonada no tempo demanda um sistema planejador centralizado, que tenha conhecimento global do problema. Entretanto, essa característica conflita com a maior robustez do sistema distribuído. Característica a qual as arquiteturas apresentadas buscam. As arquiteturas que apresentam tal característica (ALLIANCE e M+) não fazem tal alocação dado um planejamento global, como discutido na seção 3.4.4.

Tabela 4 – Classificação das arquiteturas de acordo com a taxonomia de sistemas multirrobo.

Arquiteturas	Dimensões						
	Conhecimento	Coordenação	Organização	Comunicação	Composição da Equipe	Arquitetura do Sistema	Tamanho da Equipe
Alliance	Intencional	Fracamente Acoplada	Completamente Distribuído	Explícita - Broadcast	Heterogênea	Reativo	>10 robôs
Murdoch	Intencional	Fracamente Acoplada	Distribuído	Explícita - publisher/ subscriber	Heterogênea	Híbrido	>10 robôs
M+	Intencional	Fracamente Acoplada	Distribuído	Explícita - broadcast	Heterogênea	Híbrido	>5 robôs
BLE	Intencional	Fracamente Acoplada	Completamente Distribuído	Explícita - broadcast	Heterogênea	Reativo	>5 robôs
Monad	Intencional	Fracamente ou Fortemente Acoplada	Distribuído	Implícita	Não definido	Deliberativo	Não definido
ASyMTRe	Intencional	Fortemente Acoplada	Distribuído	Explícita - broadcast	Heterogênea	Híbrido	>10 robôs
Dynamic Role Assignment	Intencional	Fortemente Acoplada	Hierárquico	Explícita - broadcast	Heterogênea	Híbrido	>20 robôs
TraderBots	Intencional	Fortemente Acoplada	Distribuído	Explícita - broadcast	Heterogênea	Híbrido	>10 robôs
IQ-ASyMTRe	Intencional	Fortemente Acoplada	Distribuído	Explícita - broadcast	Heterogênea	Híbrido	>10 robôs

Fonte: Próprio autor.

Tabela 5 – Classificação das arquiteturas de acordo com a taxonomia para problemas de alocação de tarefas em sistemas multirrobo.

Arquiteturas	Classificações					
	Tipo de Robô		Tipo de Tarefa		Tipo de Alocação	
	<i>ST</i>	<i>MT</i>	<i>SR</i>	<i>MR</i>	<i>IA</i>	<i>TA</i>
Alliance	X		X		X	X
Murdoch	X		X		X	
M+	X		X		X	X
BLE	X		X		X	
Monad	X		X		X	
ASyMTRe	X			X	X	
Dynamic Role Assignment	X		X		X	
TraderBots	X		X		X	
IQ-ASyMTRe	X			X	X	

Fonte: Próprio autor.

4 A visão arroviana do problema MRTA

“A Teoria da Escolha Social começa com algumas condições minimamente razoáveis que relacionam os resultados às comunicações (ou interações)” (ARROW; SEN; SUZUMURA, 2011). As condições buscam tornar a comunicação factível. E se assume-se que a escolha social depende do bem-estar individual, então o que é comunicado é a preferência individual, embora o resultado coletivo deva ser uma função das comunicações (ARROW; SEN; SUZUMURA, 2011).

Este capítulo visa apresentar a estrutura formal do problema de alocação de tarefas em sistemas multirrobo sob a visão arroviana, apresentando as condições necessárias para evitar a impossibilidade da decisão em conjunto do grupo de robôs. Além disso, serão tratados os principais problemas classificados pela taxonomia de Gerkey e Matarić (2004), no mesmo domínio. E, ainda, serão discutidos os mecanismos de alocação baseados em regras de escolha social para esses casos, observadas as condições atendidas e aquelas que devem ser relaxadas.

4.1 Bem-estar social em SMR

Antes de analisar o Bem-estar social para SMR, é necessário entender o conceito bruto de bem-estar (*welfare*) e bem-estar social (*social welfare*). O termo bem-estar engloba vários aspectos e significados.

De acordo com Greve (2008), geralmente, é relacionado ao critério monetário, embora, algumas vezes, seja necessário incluir outros aspectos como a felicidade. A palavra *welfare* deriva de *well*, em seu sentido comum, ou seja, de maneira satisfatória ou boa, e *fare* como um fornecimento de alimento (GREVE, 2008).

E de acordo com Greve (2008), “bem-estar pode ser relacionado tanto com o individual quanto com o coletivo, envolvendo necessidades materiais e imateriais”. Seu significado tem forte dependência com o domínio de referência. Por exemplo, o termo pode ser relacionado desde aplicações de justiça social até a aspectos biológicos.

Como uma abordagem sociológica do bem-estar, Allardt (1975 apud GREVE, 2008) sugere que o bem-estar está relacionado com ter, amar e ser. Então, bem-estar é, essencialmente, sobre realizar necessidades básicas dos indivíduos e famílias. Entretanto, as duas principais áreas de estudo são as ciências Sociais e Econômicas. A primeira analisa o Estado de Bem-estar, a intervenção do Estado na sociedade. A segunda estuda “os possíveis efeitos de várias políticas econômicas no bem-estar da sociedade” (NATH, 1973). Além disso, preocupa-se com a forma de medição do bem-estar e, conseqüentemente, com sua maximização.

Na Economia do Bem-estar, o bem-estar é comumente relacionado à utilidade. A comparação entre bem-estar e a ideia de utilidade de Benthan no século XVIII é clara, segundo Greve (2008). Ademais, bem-estar (e bem-estar social) tem sido associado com a percepção individual de utilidade. Assim, o bem-estar social resulta da soma do bem-estar (utilidade) dos indivíduos da sociedade. Em outras palavras, “a felicidade total de uma sociedade é frequentemente entendida como a agregação das felicidades individuais” (GREVE, 2008).

O termo utilidade traz a ideia de tomada de decisão, de escolha. Um indivíduo escolhe uma alternativa (dentre as possíveis) de acordo com a maior utilidade por ele calculada. Esse valor segue parâmetros individuais. Para Praag e Frijters (1999), atribuir valores de utilidade às alternativas possíveis é um método eficiente de descrever um problema de escolha.

A medição da utilidade ainda é um assunto nebuloso. Vários métodos cardinais foram apresentados por economistas, como citado por Praag e Frijters (1999). Mas, do lado oposto, muitos são os críticos. Vide a conclusão de Vilfredo Pareto sobre o ordenamento entre dois bens, citado na seção 2.1. Da mesma forma, a medição e o cálculo da utilidade em sistemas robóticos enfrenta desafios, ainda mais quando os valores individuais devem ser comparados.

Então, o conceito de bem-estar social apresentado aqui tem o objetivo de agregar preferências individuais em uma tomada de decisão coletiva. Entretanto, essas decisões devem elevar o nível de bem-estar social. Ou, pelo menos, mantê-la no mesmo nível. A fim de atingir o bem-estar social, o processo de tomada de decisão deve ter uma estrutura correta para este fim. Por esse motivo, funções de bem-estar social ou de escolha social são indicadores para tais mecanismos de agregação de preferências.

4.2 Modelo Formal da Escolha Social para problemas MRTA

Dentre outras definições, a palavra sociedade significa: “agrupamento de pessoas que vivem em estado gregário e em cooperação mútua; grupo de indivíduos que se submetem a um regulamento com o objetivo de exercer uma atividade comum ou defender interesses de todos” (DILGUERIAN; WEISZFLOG, 1998). O termo sociedade é apropriado, então, para denominar um grupo de robôs programados para exprimir comportamento cooperativo na execução de tarefas de uma missão. Assim, seja $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ o conjunto de m robôs, móveis ou não, que compõe uma sociedade, submetendo-se a uma mesma arquitetura de controle de alocação de tarefas.

As tarefas consideradas neste modelo seguem a Definição 3.1. Adicionalmente, serão consideradas apenas tarefas fracamente acopladas, cuja interdependência na execução é nula. As tarefas são independentes. Essa consideração simplifica o modelo. Considere $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ o conjunto de n tarefas independentes possivelmente ponderadas, sendo necessário um ou mais robôs para a execução de cada tarefa.

Como visto na seção 2.1, a medição da utilidade como uma quantidade sempre foi um problema. E conforme a seção 3.3, a estimativa cardinal da utilidade em SMR para tomada de decisão apresenta seus problemas, principalmente, incertezas de medição e do ambiente. Então,

a estimativa não negativa da taxa de eficiência do desempenho do robô para cada tarefa, a utilidade, será substituída por uma análise ordinal de preferências. Essa análise será representada pelo ordenamento das preferências dos robôs em relação à execução das tarefas ou pelo ordenamento de preferências dos robôs sobre os robôs que executarão as tarefas. O ordenamento de preferências segue a comparação binária entre duas alternativas. As alternativas possíveis são consideradas em dois domínios diferentes, o das tarefas ou o dos robôs. As comparações par a par formarão um ordenamento completo e transitivo.

O ordenamento de preferência traz vantagens para a análise da decisão coletiva no domínio da alocação de tarefas em SMR. Segundo [Fleurbaey e Maniquet \(2011\)](#), o “*ordinalismo* segue do fato de que qualquer função de utilidade pode ser substituída por uma transformada crescente dela mesma, de modo que apenas a classificação do grupo é importante”. O princípio deixa claro que a intensidade não mais será levada em consideração, mas, sim, a ordem de preferência. Em um SMR, cada robô continuará com sua função de utilidade interna, para o cálculo da utilidade (ou custo, ou aptidão) para cada tarefa, entretanto, esse valor não mais será comparado entre os agentes, seguindo o princípio da *Incomparabilidade*: considerando que as transformadas das funções de utilidade são tantas quantos os agentes envolvidos, ou seja, cada agente pode ter sua própria função de utilidade, não há um significado comum que pode ser dado aos níveis de utilidade ou às diferenças de utilidades ([FLEURBAEY; MANIQUET, 2011](#)). Dessa forma, sob os princípios do ordinalismo e da incomparabilidade, não se perderá a generalidade definindo a preferência da sociedade diretamente na forma das preferências dos robôs do conjunto \mathcal{R} .

Por outro lado, a utilização de valores escalares de utilidade permitem uma busca clara pela maximização de uma função objetivo. Para o caso ordinal, [Arrow \(1950\)](#) enuncia a dificuldade de se obter uma função objetivo clara para a maximização:

se adotarmos a identificação tradicional da racionalidade com a maximização de alguma característica, então o problema de alcançar um máximo social derivado de desejos individuais é precisamente o problema que tem sido central no campo da economia do bem-estar. Entretanto, a busca por uma definição clara do bem-estar social ótimo tem sido assolado pelas dificuldades da comparação interpessoal. A ênfase, como é sabido, mudou para uma definição mais fraca de otimização, a saber, a determinação de todos os estados sociais, de modo que nenhum indivíduo pode melhorar sem piorar a situação de outro.

Dessa forma, atingir o bem-estar social no problema de alocação de tarefas será a obtenção de um ordenamento de preferências tal que respeite os Axiomas 2.1 e 2.2, pois, segundo [Arrow \(1950\)](#), “como em qualquer tipo de comportamento descrito pela maximização, a mensurabilidade do bem-estar social não precisa ser considerada; tudo que importa é a existência de um ordenamento social que satisfaça os Axiomas 2.1 e 2.2”.

Nesse trabalho as funções objetivo na qual se busca a maximização são chamadas *função de bem-estar social Arroviana* e a função de Escolha Social. Cada caso de aplicação será detalhado na seção 4.2.1. Isso significa que, além da necessidade de satisfazer os Axiomas 2.1

e 2.2, a função de bem-estar social deve satisfazer um conjunto de condições similares às condições estabelecidas na seção 2.3 para o Teorema da Impossibilidade de Arrow. E as funções de escolha social devem satisfazer condições paralelas às condições de Arrow, entretanto para o caso da escolha social. Essas condições serão traduzidas para o domínio da alocação de tarefas em SMR e apresentadas na seção 4.2.2.

Essa função de bem-estar social Arroviana pode ser considerada utilitarista mesmo que não realize uma soma explícita das utilidades individuais. Isso pode ser considerado pelo fato do mecanismo levar em consideração as preferências individuais de cada robô e agregar as preferências de acordo com as mesmas, sem levar em consideração, por exemplo, o robô ou tarefa de menor utilidade. Do ponto de vista igualitarista, a busca seria pela maximização da menor utilidade.

A notação de preferências apresentada na seção 2.1.1 será, também, utilizada para representar as preferências da sociedade de robôs. Dessa forma, para o robô r_i , com $i = (1, \dots, m)$, se a execução da tarefa τ_1 é tão preferível quanto à execução da tarefa τ_2 , então $\tau_1 \succeq_i \tau_2$. Analogamente, se a execução de τ_1 é estritamente preferível à execução de τ_2 , logo, $\tau_1 \succ_i \tau_2$. E se o agente r_i é indiferente em relação às tarefas, $\tau_1 \sim_i \tau_2$.

De outro ponto de vista, o robô r_i pode expressar sua preferência acerca do robô que executará determinada tarefa. Assim, para representar a preferência do robô r_i à execução da tarefa τ_k , com $k = (1, \dots, n)$: se a escolha do robô r_1 para a execução é tão boa quanto ao robô r_2 , então $r_1 \succeq_i^{\tau_k} r_2$; se r_1 é estritamente preferível à r_2 , então $r_1 \succ_i^{\tau_k} r_2$; e se r_1 é indiferente em relação a r_2 , então $r_1 \sim_i^{\tau_k} r_2$. Ainda considerando todos as preferências transitivas, os ordenamentos de preferências serão considerados não-estritos. Dessa forma, será denotado por $\succeq_{r_i} = \{\tau_1 \succeq \tau_2 \succeq \dots \succeq \tau_n\}$ o ordenamento de preferências do robô r_i . O ordenamento de preferências do robô r_i sobre os robôs para a execução de determinada tarefa τ_k é denotado por $\succeq_{r_i}^{\tau_k} = \{r_1 \succeq r_2 \succeq \dots \succeq r_m\}$.

O conjunto das preferências individuais racionais possíveis não-estritas será denotado por R . O conjunto de ordenamentos de preferências, aqui também chamado de perfil de preferências, será denotado por: (i) $[\succeq]_{\mathcal{R}} \in R^n$ quando o conjunto representa o ordenamento de preferências individuais dos robôs em relação à execução das tarefas; (ii) $[\succeq]_{\mathcal{R}}^{\mathcal{T}} \in R^{m \times n}$ quando o conjunto representa o ordenamento de preferências individuais dos robôs sobre os robôs do conjunto \mathcal{R} para a execução das tarefas do conjunto \mathcal{T} . A escolha social será feita sobre $[\succeq]_{\mathcal{R}}$ ou $[\succeq]_{\mathcal{R}}^{\mathcal{T}}$.

4.2.1 Funções de escolha e bem-estar social

Para o problema de alocação de tarefas em sistemas multirrobô, duas situações de escolha podem ser identificadas: (i) a escolha sobre as tarefas alternativas e (ii) a escolha sobre os robôs alternativos. Funções de bem-estar social têm como resultado um ordenamento total das alternativas apresentadas à sociedade. Esse ordenamento é um mapeamento das escolhas sociais de acordo com as preferências individuais. Ou seja, o mapeamento realizado por uma

função de bem-estar social gera um planejamento total da alocação de tarefas. Funções de escolha social selecionam a alternativa melhor ranqueada, de classificação mais elevada. Dessa forma, as funções de escolha social no domínio multirrobô definirão a tarefa melhor classificada para execução ou o robô melhor classificado para execução de uma determinada tarefa. Em tempo, essas funções serão divididas para os dois casos de ordenamento de preferências vistos, o ordenamento de tarefas e o ordenamento de robôs, para melhor entendimento.

Definição 4.1 (Função de escolha social multirrobô sobre o conjunto \mathcal{T}). Uma função de escolha social multirrobô sobre o conjunto de tarefas \mathcal{T} (FES-MR \mathcal{T}) é um processo ou uma regra que, para cada ordenamento de preferências individual sobre as tarefas, $\{\succeq_{r_1}, \dots, \succeq_{r_m}\}$, resulta em apenas uma escolha, a melhor colocada: $C([\succeq]_{\mathcal{R}}) : R^m \mapsto \mathcal{T}$.

Uma FES-MR \mathcal{T} sobre o conjunto de tarefas \mathcal{T} deve agregar as preferências dos membros da sociedade de robôs sobre as tarefas da missão.

Definição 4.2 (Função de escolha social multirrobô sobre os conjuntos \mathcal{R} e \mathcal{T}). Uma função de escolha social multirrobô (FES-M $\mathcal{R}\mathcal{T}$) sobre o conjunto de robôs \mathcal{R} e o conjunto de tarefas \mathcal{T} é um processo ou uma regra que, para cada ordenamento de preferências sobre a execução das tarefas, $\{\succeq_{r_1}^{\tau_k}, \dots, \succeq_{r_m}^{\tau_k}\}$, resulta em apenas uma escolha, o robô melhor colocado para a execução da tarefa τ_k : $C(\tau_k) : R^m \mapsto \mathcal{R}$, para cada tarefa do conjunto \mathcal{T} e com $k = \{1, 2, \dots, n\}$.

Já a FES-M $\mathcal{R}\mathcal{T}$ apresentada na Definição 4.2 é um mecanismo de alocação em que o robô melhor ranqueado será escolhido para execução da tarefa. Dessa forma, a agregação de preferências ocorre mediante o conhecimento das capacidades próprias do robô, além do conhecimento das capacidades dos demais membros da equipe. A função mapeia o melhor robô para a execução de cada tarefa, segundo a preferência social. Dessa forma, os agentes devem ter suas preferências de acordo com os candidatos para execução daquela tarefa. Assim, essa função de escolha pode ser entendida como uma votação em que as tarefas são alocadas para os robôs melhores candidatos em relação à sua execução.

Funções de bem-estar social têm como resultado um ordenamento total das alternativas apresentadas à sociedade. Esse ordenamento é um mapeamento das escolhas sociais de acordo com as preferências individuais. Uma função de bem-estar social sobre o conjunto de tarefas \mathcal{T} agrega as preferências individuais dos robôs sobre as tarefas, gerando um ordenamento completo das tarefas a serem executadas.

Definição 4.3 (Função de bem-estar social multirrobô sobre o conjunto \mathcal{T}). Uma função de bem-estar social multirrobô sobre o conjunto de tarefas \mathcal{T} (FBS-MR \mathcal{T}) significa um processo ou uma regra a qual, para cada relação individual de preferências sobre as tarefas alternativas $\succeq_{r_1}, \dots, \succeq_{r_m}$, sendo $\succeq_{r_i} = (\tau_1 \succeq \dots \succeq \tau_k)$ e $k = (1, \dots, n)$, resulta em um ordenamento social correspondente às alternativas, $[\succeq]_{\mathcal{R}} = R^m \mapsto \mathcal{T}$.

Caso o mecanismo de alocação para essa função de bem-estar social tiver como resultado um ordenamento completo da execução das tarefas, a mesma pode ser entendida como a formação de uma coalização, em que os robôs da sociedade decidirão a ordem das tarefas as quais todos devem executar de forma cooperativa. Esse caso será discutido quando o problema envolve tarefas que requerem múltiplos robôs para sua execução.

Ainda sobre a FBS-MRT, há mecanismos de alocação que, apesar de atingirem o resultado de ordenamento social, o fazem de maneira iterativa, considerando as preferências dos robôs em uma ordem pré-definida. Tais mecanismos serão discutidos na seção 4.3.

Além do ordenamento das tarefas alternativas, a situação em que a preferência será revelada acerca dos robôs alternativos para a execução das tarefas também é possível. Para tal, a agregação de preferências ocorre mediante o conhecimento das capacidades próprias do robô, além do conhecimento das capacidades dos demais membros da equipe. A função da Definição 4.4 mapeia o melhor robô para a execução de cada tarefa, segundo a preferência social.

Definição 4.4 (Função de bem-estar social multirrobô sobre os conjuntos \mathcal{R} e \mathcal{T}). Uma função de bem-estar social multirrobô sobre o conjunto de robôs \mathcal{R} e o conjunto de tarefas \mathcal{T} (FBS-MRT) significa um processo ou uma regra a qual, para cada relação individual de preferências sobre os robôs para a execução das tarefas $\succeq_{r_i}^{\mathcal{T}}, \dots, \succeq_{r_m}^{\mathcal{T}}$, sendo $\succeq_{r_i}(\mathcal{R}, \mathcal{T}) = (r_i, \tau_1 \succeq \dots \succeq r_m, \tau_1; \dots; r_i, \tau_k \succeq \dots \succeq r_m, \tau_k)$ e $k = (1, \dots, n)$, resulta em um ordenamento social correspondente às alternativas, $[\succeq]_{\mathcal{R}}^{\mathcal{T}} = R^{m \times n} \mapsto \mathcal{R}$.

De forma análoga ao observado para a função da Definição 4.3, um mecanismo iterativo também é aplicável à implementação da função de bem-estar social. Nessa situação será iterada cada tarefa e definida a melhor opção, ou seja, o robô de maior preferência para a execução da mesma. A partir dessa observação, é possível afirmar que uma FES é uma instância de uma FBS para o problema MRTA, pois ao fim das iterações de uma FES (em que a cada iteração é dado um resultado), o mapeamento social completo é feito. Para o caso da alocação de tarefas, esse mapeamento é a própria alocação de tarefas apropriada dado os perfis de preferências individuais dos robôs da sociedade.

As condições de existência das funções de bem-estar social definidas são definidas na próxima seção. As condições são baseadas nas condições definidas por Arrow (1951) e por Gibbard (1973) e Satterthwaite (1975) para a existência de uma função de bem-estar e de escolha social, respectivamente, que agregue as preferências individuais de forma democrática e à prova de manipulações.

4.2.2 As condições arrovianas para o problema MRTA

A fim de considerar a axiomatização proposta por Arrow (1951), são propostos os axiomas e as condições para a sociedade de robôs. As condições apresentadas a seguir são consideradas na situação em que o número m de robôs membros da sociedade é pelo menos três, ou seja, $m \geq 3$. Além disso, é assumido que o número n de tarefas apresentadas aos indivíduos é

de pelo menos três, ou seja, $n \geq 3$. Tais considerações são importantes para que as definições impostas no teorema ainda sejam válidas. A diferença é que o número de indivíduos da sociedade, originalmente, é no mínimo dois. O número de alternativas mínimas permanece o mesmo, três. Como os robôs, que são membros da sociedade, também podem ser alternativas, o número mínimo de robôs é fixado em três.

Por ter relação estreita com o Teorema de Gibbard-Satterthwaite, as condições aqui descritas também serão aplicadas às funções de escolha social, adaptando-se apenas para o tipo de resultado característico dessa função. Assim, por simplicidade, tanto as funções de escolha social quanto as funções de bem-estar social estarão no domínio arroviano, tendo como única diferença o resultado, ou seja, uma escolha vencedora ou uma relação de preferência social ganhadora, respectivamente.

O axioma da Completude 4.1 define que os robôs da sociedade devem ordenar as tarefas disponíveis, ou os robôs para execução de determinada tarefa, de acordo com sua preferência (utilidade) individual, sendo permitidos empates. Ressalta-se que o cálculo da utilidade interna realizado pelos robôs será considerado para esse ordenamento. Entretanto, os valores internos não serão comparados com os valores dos demais membros da equipe. Além disso, as decisões devem ser racionais. Desse modo, o ordenamento de preferências dos indivíduos deve ser transitivo, seguindo o Axioma 4.2 da Transitividade para problemas MRTA.

Axioma 4.1 (Completude em problemas MRTA). Para o ordenamento de preferências sobre tarefas, $\forall \tau_1, \tau_2 \in \mathcal{T}$, $\tau_1 \succeq \tau_2$ ou $\tau_2 \succ \tau_1$ ou $\tau_1 \sim \tau_2$. Para o ordenamento de preferências sobre robôs, $\forall r_1, r_2 \in \mathcal{R}$, $r_1 \succeq^{\tau_k} r_2$ ou $r_2 \succ^{\tau_k} r_1$ ou $r_1 \sim^{\tau_k} r_2$.

Axioma 4.2 (Transitividade em problemas MRTA). Para o ordenamento de preferências sobre tarefas, sendo $\tau_1, \tau_2, \tau_3 \in \mathcal{T}$, se $\tau_1 \succeq \tau_2$ e $\tau_2 \succeq \tau_3$, então $\tau_1 \succeq \tau_3$. Para o ordenamento de preferências sobre robôs, sendo $r_1, r_2, r_3 \in \mathcal{R}$, se $r_1 \succeq^{\tau_k} r_2$ e $r_2 \succeq^{\tau_k} r_3$, então $r_1 \succeq^{\tau_k} r_3$.

A condição de Domínio Irrestrito na teoria de Arrow (1951) impõe que, a partir dos ordenamentos de preferências individuais sem restrições, um ordenamento social que agrega as preferências individuais é resultante. No caso de problemas MRTA, essa condição estabelece que a relação de preferências entre as tarefas, ou os robôs que executarão as tarefas, podem ser ordenadas de forma irrestrita, de acordo com a preferência do indivíduo. Esse ordenamento é um ordenamento fraco, uma vez que não exclui relações de indiferença. Essa observação é importante em dois casos: (i) quando o robô não é capaz de executar duas ou mais tarefas e (ii) quando o robô não tem o conhecimento para julgar a preferência de execução de uma tarefa entre dois ou mais robôs.

No caso (i), se duas ou mais tarefas não podem ser executadas pelo robô, seu grau de preferência quanto a execução das mesmas deve ser o mais baixo, além da execução entre as mesmas ser indiferente. Já no caso (ii), o ordenamento das preferências do robô r_i sobre dois outros robôs dos quais não se tem referência sobre a execução de determinada tarefa, além de ter um grau de preferência baixo, deve ser indiferente.

Condição 4.1 (Domínio Irrestrito em problemas MRTA). Independente dos ordenamentos de preferências individuais realizados sobre as tarefas ou sobre os robôs, há um ordenamento social $[\succeq]_{\mathcal{R}}$ ou $[\succeq]_{\mathcal{R}}^T$.

A condição 2.3 que estabelece que o julgamento das preferências deve ser par a par, independentemente de uma terceira alternativa, pode ter duas visões diferentes na aplicação à SMR. A diferenciação depende do conjunto de alternativas que será efetivamente ordenado: tarefas ou robôs para execução das tarefas.

A Condição 4.2 faz referência ao ordenamento de tarefas. Assim, a preferência entre a execução de duas tarefas depende apenas na relação binária entre essas mesmas duas tarefas e independente da informação de preferência de uma terceira alternativa. Esse ordenamento também é independente da informação de utilidade da alternativa irrelevante.

Condição 4.2 (Independência de Tarefas Irrelevantes). Sejam $\succeq_{r_1}, \dots, \succeq_{r_n}$ e $\succeq'_{r_1}, \dots, \succeq'_{r_n}$ dois conjuntos de relações de ordenamento individuais e $C(R)$ e $C'(R)$ as funções de escolha social correspondentes. Se, para cada indivíduo r_i e todo τ_1 e τ_2 em determinado ambiente R , $\tau_1 \succeq_{r_i} \tau_2$ se e somente se $\tau_1 \succeq'_{r_i} \tau_2$, então, $C(R)$ e $C'(R)$ são iguais.

A Condição 4.3 é a versão da Condição 2.3 sobre a preferência em relação aos robôs que executarão determinada tarefa. Essa condição, como será discutido na Seção 4.3, encaixa-se na situação em que os robôs têm prévio conhecimento das capacidades dos demais membros da equipe, além do conhecimento prévio da tarefa ou da missão. Dessa forma, a alocação de tarefas pode ser executada de acordo com as preferências de cada robô para cada tarefa. A maximização do bem-estar social ocorre quando as preferências individuais são consideradas, sob os Axiomas 4.1 e 4.2, e a tarefa é alocada ao robô que conta com a maior preferência da sociedade para tal execução.

Condição 4.3 (Independência de Robôs Irrelevantes). Sejam $\succeq_{r_1}, \dots, \succeq_{r_n}$ e $\succeq'_{r_1}, \dots, \succeq'_{r_n}$ dois conjuntos de relações de ordenamento individuais e $C(R)$ e $C'(R)$ as funções de escolha social correspondentes. Se, para cada indivíduo r_i e todo r_1 e r_2 em determinado ambiente R , $r_1 \succeq_{r_i} r_2$ se e somente se $r_1 \succeq'_{r_i} r_2$, então, $C(R)$ e $C'(R)$ são iguais.

Juntamente com os Axiomas 4.1 e 4.2, a maximização do bem-estar será atingido com o Princípio de Pareto, ou Princípio da Unanimidade. Antes de definir a Condição da Unanimidade para problemas MRTA, serão redefinidos os conceitos de Pareto-dominância e Otimalidade de Pareto para o domínio SMR.

Definição 4.5 (Pareto-dominante). O perfil de preferências $[\succeq]_{\mathcal{R}} \in R^n$ (ou $[\succeq]_{\mathcal{R}}^T \in R^{m \times n}$) é Pareto-dominante em relação ao perfil de preferências $[\succeq]'_{\mathcal{R}}$ (ou $[\succeq]'_{\mathcal{R}}^T$) se para todo $r_i \in \mathcal{R}$, $u_{r_i}([\succeq]_{\mathcal{R}}) \geq u_{r_i}([\succeq]'_{\mathcal{R}})$ e existe algum indivíduo $r_j \in \mathcal{R}$ para o qual $u_{r_j}([\succeq]_{\mathcal{R}}) > u_{r_j}([\succeq]'_{\mathcal{R}})$ (ou $u_{r_i}([\succeq]_{\mathcal{R}}^T) \geq u_{r_i}([\succeq]'_{\mathcal{R}}^T)$ e existe algum indivíduo $r_j \in \mathcal{R}$ para o qual $u_{r_j}([\succeq]_{\mathcal{R}}^T) > u_{r_j}([\succeq]'_{\mathcal{R}}^T)$).

A Definição 4.5 determina que um perfil de preferências é dominante se não existe outro perfil de preferências que é capaz de melhorar o bem-estar dos indivíduos da equipe de robôs sem que prejudique algum dos membros da equipe ¹. Se esse estado é atingido, então, entende-se que a sociedade está na condição Ótima de Pareto. De acordo com a Definição 4.6, dado um perfil de preferências sobre as tarefas do conjunto \mathcal{T} que é dominante sobre todos os outros perfis de preferência da sociedade, então a condição de eficiência de Pareto foi atingida. De maneira análoga, se o perfil de preferências sobre os robôs do conjunto \mathcal{R} em relação às tarefas do conjunto \mathcal{T} é dominante, então a condição de eficiência de Pareto foi atingida para esta alocação de tarefas.

Definição 4.6 (Otimalidade de Pareto). O perfil de preferências $[\succeq]_{\mathcal{R}}$ (ou $[\succeq]_{\mathcal{R}}^{\mathcal{T}}$) é Pareto-ótimo, ou Pareto-eficiente, se não existe outro perfil de preferências $[\succeq]'_{\mathcal{R}} \in R^n$ (ou $[\succeq]'_{\mathcal{R}} \in R^{m \times n}$) que domina $[\succeq]_{\mathcal{R}}$ (ou $[\succeq]_{\mathcal{R}}^{\mathcal{T}}$).

Com isso, a condição da Unanimidade para a alocação de tarefas em sistemas multirrobô pode ser enunciada como:

Condição 4.4 (Princípio de Pareto ou da Unanimidade em problemas MRTA). A função de bem-estar social no domínio multirrobô é Pareto-eficiente se para qualquer $\tau_1, \tau_2 \in \mathcal{T}$, $\forall r_i \in \mathcal{R}$ $\tau_1 \succ_{r_i} \tau_2$ implica em $\tau_1 \succ_{\mathcal{R}} \tau_2$ ou se para qualquer $r_1, r_2 \in \mathcal{R}$, $\forall r_i \in \mathcal{R}$ e $\forall \tau_k \in \mathcal{T}$, $r_1 \succ_{r_i}^{\tau_k} r_2$ implica em $r_1 \succ_{\mathcal{R}}^{\tau_k} r_2$.

Para enunciar a última condição da estrutura de análise arroviana, é apresentada a Definição 4.7 de Dominância da função de bem-estar social. A Dominância ocorre sempre que as preferências de um indivíduo sobrepõem as demais, dominando, assim, o perfil de preferência social.

Definição 4.7 (Dominância em problemas MRTA). Uma função de bem-estar social é dita dominante, ou ditatorial, se existe um indivíduo r_i tal que, para toda tarefa τ_1 e τ_2 , $\tau_1 \succ_{r_i} \tau_2$ implica em $\tau_1 \succ_{\mathcal{R}} \tau_2$, ou para todo robô r_1 e r_2 , $r_1 \succ_{r_i}^{\tau_k} r_2$ implica em $r_1 \succ_{\mathcal{R}}^{\tau_k} r_2$, independentemente dos ordenamentos $\succeq_{r_1}^{\tau_k}, \dots, \succeq_{r_n}^{\tau_k}$ dos demais indivíduos além de r_i . Onde $\succ_{\mathcal{R}}$ é a relação estrita de preferência social correspondente à $\succeq_{r_1}, \dots, \succeq_{r_n}$ e $\succ_{\mathcal{R}}^{\tau_k}$ é a relação estrita de preferência social correspondente à $\succeq_{r_1}^{\tau_k}, \dots, \succeq_{r_n}^{\tau_k}$.

A característica da Definição 4.7 não é benéfica para a agregação de preferências individuais buscando a maximização do bem-estar social. Por isso, a Condição 4.5 estabelece que a função de bem-estar social não deve permitir que um indivíduo governe a alocação de tarefas com base em suas preferências individuais.

Condição 4.5 (Não-ditadura em problemas MRTA). A função de bem-estar social não deve ser ditatorial ou dominante.

¹ Definição adaptada de Shoham e Leyton-Brown (2008)

Com as definições preliminares e das condições para existência da escolha social, o problema da alocação de tarefas em sistemas multirrobô apresentado na seção 3.3 é revisto do ponto de vista Arroviano, conforme a Descrição 4.8.

Definição 4.8 (O problema Arroviano de alocação de tarefas em SMR). Dada uma sociedade $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ de m robôs, cada um capaz de executar uma ou mais tarefas simultaneamente, e um conjunto $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ de n tarefas possivelmente ponderadas, sendo necessário um ou mais robôs para a execução de cada tarefa. Dado, além disso, para cada robô r_i um ordenamento fraco de preferências \succeq_{r_i} para a execução das tarefas (se o robô r_i é incapaz de executar determinada tarefa, então esta é a tarefa de menor preferência) ou um ordenamento fraco de preferências $\succeq_{r_i}^{\mathcal{T}}$ sobre conjunto de robôs \mathcal{R} para a execução das tarefas do conjunto \mathcal{T} (se o robô r_j é incapaz de executar a tarefa τ_k , então o robô r_i irá atribuir a menor preferência de execução da tarefa τ_k para o robô r_j). Então, o objetivo é atribuir robôs às tarefas de modo a maximizar o bem-estar social, ou seja, satisfazer os Axiomas 4.1 e 4.2 e as condições 4.1, 4.2 ou 4.3, 4.4, 4.5, tendo um ordenamento de preferências social $[\succeq]_{\mathcal{R}}$ ou $[\succeq]_{\mathcal{R}}^{\mathcal{T}}$ que agrega as preferências individuais.

Em resumo, o problema arroviano de alocação de tarefas em SMR (*problema MRTA-arroviano*) é uma tripla $\langle \mathcal{R}, \mathcal{T}, [\succeq]_{\mathcal{R}} \rangle$ ou $\langle \mathcal{R}, \mathcal{T}, [\succeq]_{\mathcal{R}}^{\mathcal{T}} \rangle$.

Entretanto, como já discutido e demonstrado no capítulo 2, a função de escolha social ou de bem-estar social que atenda aos Axiomas 4.1 e 4.2 e às condições 4.1, 4.2 ou 4.3, 4.4 é ditatorial, não satisfazendo a condição 4.5. Dessa forma, na seção 4.3 e no capítulo 5, respectivamente, serão analisadas as condições arrovianas para os problemas que envolvem o tipo ST de robôs, delineados na Taxonomia de Gerkey e Mataric (2004), e as arquiteturas ALLIANCE, MURDOCH e M+. A restrição aos problemas ST deve-se, primeiramente, à características das arquiteturas práticas propostas na literatura, conforme observado na Tabela 5, assim como à maior complexidade da análise de problemas MT (o que será brevemente abordado na Seção 4.3.5).

4.2.3 Preferências de longo e curto prazo

A característica dinâmica de um sistema multirrobô leva à definição de dois tipos de ordenamentos de preferências possíveis. A diferença entre as classificações é o tempo.

A alocação de tarefas leva o sistema à alterações constantes. Uma tarefa disponível para execução é alocada, outra tarefa que estava sendo executada é abandonada pelo robô. Essas alterações no ambiente geram alterações nas preferências dos robôs, uma vez que os robôs são programados para executar a missão, mesmo que resulte em uma solução quase ótima.

Segundo Klaus e Bochet (2013), o domínio de preferências arroviano é um domínio irrestrito de preferências estritas. De forma mais geral, o domínio de preferências arroviano será considerado o conjunto de preferências logicamente possíveis que satisfaçam as condições impostas por Arrow (1951).

A fim de tornar a análise das arquiteturas de alocação de tarefas mais clara, serão definidos dois conjuntos de preferências contidos no domínio arrowiano. Esses conjuntos têm uma relação com o tempo. Essa relação é estabelecida principalmente por conta da característica dinâmica do sistema.

As preferências iniciais do robô serão chamadas *preferências de longo-prazo*. Isso se deve ao fato de, no início, as preferências do robô dependerem apenas de suas motivações internas (ALLIANCE), primeiras ofertas recebidas (MURDOCH) ou planejamentos individuais (M+) sobre as tarefas apresentadas. Em contrapartida, as *preferências de curto-prazo* são aquelas em que há a interação entre os robôs do sistema, a conclusão de tarefas que diminui o número de alternativas possíveis, ou o abandono de tarefas que torna a aumentar o número de alternativas possíveis. As preferências de curto-prazo são mais voláteis no sentido de variação no número de alternativas.

Entretanto, as características das preferências arrowianas deve se manter em ambos os conjuntos de preferências. Isso quer dizer que as preferências de curto-prazo, mesmo com toda volatilidade, devem manter a transitividade das preferências e a independência de alternativas irrelevantes.

A manutenção da transitividade se deve à necessidade de manter as decisões racionais e não-cíclicas. A independência de alternativas irrelevantes é importante, pois, com a variação nas alternativas, o ordenamento par a par não deve ser alterado. Assim, com o passar do tempo de execução do sistema, o ordenamento de preferências individual de longo-prazo não deve ter a ordem inicial alterada, mesmo que as alternativas sejam modificadas.

Por exemplo, sendo as preferências de longo-prazo de um robô r_i para três tarefas alternativas $\succeq_{r_i} = \{\tau_2 \succeq \tau_1 \succeq \tau_3\}$. Após a alocação e execução da tarefa τ_1 por outro robô r_j (ou até mesmo pelo próprio r_i), as preferências de curto-prazo de r_i devem permanecer com o ordenamento $\succeq'_{r_i} = \{\tau_2 \succeq \tau_3\}$. A alteração do ordenamento para $\succeq'_{r_i} = \{\tau_3 \succeq \tau_2\}$ não satisfaz as condições já discutidas. A mesma análise é válida quando as alternativas a serem ordenadas são os robôs para a execução de determinada tarefa.

É desejável que as preferências, tanto de longo-prazo quanto de curto-prazo, sejam independentes da (i) da ordem em que as alternativas são apresentadas e (ii) da ordem em que as alternativas são retiradas do conjunto de alternativas possíveis. Essa característica é conhecida como decisão de trajetória independente (*path independent*). Esse é o conceito de que o “passado não importa” no ordenamento de preferências. Arrow (1951) afirma que a condição de transitividade garante tal independência. Os trabalhos de Bandyopadhyay (1988) e Bandyopadhyay (1990) investigam tal relação, corroborando o que foi enunciado por Arrow (1951) para a comparação binária.

4.3 Análise arroviana da Taxonomia de problemas MRTA

A análise dos problemas de alocação de tarefas definidos na taxonomia de [Gerkey e Mataric \(2004\)](#) será feito de forma individual, sob cada característica das classificações dos eixos robô, tarefa e tipo de alocação. Uma análise inicial foi proposta por [Reis e Bastos \(2017\)](#), a qual é agora expandida. Algumas premissas são consideradas para a referida análise:

- i. Os robôs falam a verdade e não são invejosos.
- ii. O número de tarefas é maior ou igual o número de robôs, ou seja, $n \geq m$.
- iii. Mesmo que a missão não seja conhecida previamente, os robôs têm conhecimento dos recursos e capacidades necessárias para a execução das tarefas.
- iv. Todos os ordenamentos são racionais, ou seja, completos e transitivos.
- v. Os robôs devem ordenar (estritamente) todas as alternativas (tarefas ou robôs) disponíveis.
- vi. Se um robô não tem capacidade de executar determinada tarefa, então esta deve ter a menor preferência em seu ordenamento de preferências individual.
- vii. Se o robô r_i tem conhecimento sobre o desempenho do robô r_j acerca da tarefa τ_k , então seu ordenamento de preferências deve resultar na preferência da execução pelo robô de maior desempenho.
- viii. Todas as tarefas da missão são conhecidas antecipadamente ao processo de alocação de tarefas.
- ix. Os robôs buscam maximizar o bem-estar social, não apenas o bem-estar individual.
- x. Os robôs têm a capacidade de calcular sua utilidade individual para cada tarefa e, a partir desse valor, estabelecer seu ordenamento de preferências.
- xi. Os robôs não dispõem de estratégias ou incentivo externo para para mentir sobre suas intenções ou funções de utilidade e, conseqüentemente, sobre seu ordenamento de preferências individual. Assim como estão inseridos em um ambiente cooperativo, não competitivo.

Além disso, as condições a seguir são consideradas comuns à todas as situações colocadas analisadas ([REIS; BASTOS, 2017](#)).

Domínio Irrestrito O ordenamento das alternativas (tarefas ou robôs) é irrestrito. Dada a consideração da independência entre as tarefas, o ordenamento de preferências sobre essas alternativas pode ter qualquer forma transitiva e completa possível. Sobre os robôs, os membros da equipe podem ordenar suas preferências sobre os robôs para a execução de uma determinada tarefa, também, de forma transitiva e completa.

Não-ditadura Não existe robô r_i cujas preferências são dominantes em relação as preferências dos demais robôs da equipe.

Independência de (Tarefas ou Robôs) Irrelevantes Assim como Yuan (2015) destaca a condição para a estrutura de Arrow, no domínio MRTA ela imprime mais uma restrição à manipulação do resultado pelos indivíduos com a mudança no ordenamento de preferências. Tanto tarefas quanto robôs são entendidos como totalmente independentes.

4.3.1 ST-SR-IA

Segundo Gerkey (2003), este é o problema mais simples de alocação de tarefas. Diferentes abordagens são apresentadas para a solução deste tipo de alocação, como Heurísticas Gulosas, Programação Linear e Otimização Combinatorial (GERKEY, 2003; CHOUDHURY, 2009). Gerkey (2003) divide tais problemas em duas classes de alocação de tarefas: *offline* e *online*. A primeira classe engloba as abordagens em que a premissa *vii* (página 98) é atendida e a alocação de tarefas é realizada de forma centralizada (usando programação linear, por exemplo) ou distribuída (algoritmo de leilões). Uma variação da abordagem *offline* é o algoritmo guloso iterativo. Nele, as tarefas são alocadas aos agentes com maior utilidade de forma iterativa, até que não haja mais tarefas disponíveis para alocação. Dessa forma, o algoritmo busca a maximização da utilidade global por meio da maximização das utilidades individuais. Nesse ponto, há uma comparação entre as utilidades individuais.

A classe de alocação *online* engloba os problemas em que novas tarefas podem ser adicionadas a qualquer momento, ou seja, a missão completa não é conhecida pelos robôs no início da execução. Tais abordagens podem permitir ou não a realocação de tarefas entre os robôs dado o surgimento de uma nova tarefa. Quando se é permitido, o problema é reduzido a uma instância do ST-SR-IA iterativo e pode chegar a uma solução ótima, segundo Gerkey (2003).

A partir de uma visão arroviana do problema, o nível de bem-estar social da sociedade de robôs aumenta conforme as tarefas alocadas a cada indivíduo respeitam suas maiores preferências. Assim, na análise que se segue, serão considerados primeiramente os problemas que atendem a premissa *vii* (página 98). No caso de problemas de alocação *online*, a comparação de preferências sobre o conjunto \mathcal{T} e, conseqüentemente, o ordenamento de preferências fica prejudicado, pois o conjunto de alternativas não é conhecido inicialmente. Com isso, o problema se torna dependente da trajetória, ou seja, a solução do problema dependerá muito da ordem com que as tarefas são inseridas na missão e apresentadas aos robôs.

Para a análise social do problema ST-SR-IA, duas situações em relação às alternativas serão consideradas. Primeiramente, quando as alternativas são as tarefas da missão (ST-SR-IA- \mathcal{T}). E como demarcado anteriormente, são todas conhecidas previamente. A segunda situação considerada será a de robôs como alternativa (ST-SR-IA- \mathcal{R}). Neste caso, é considerado que os robôs da equipe sabem da existência dos demais robôs, além de ter conhecimento de suas capacidades e desempenho nas tarefas, como ocorre na arquitetura ALLIANCE com a aplicação da extensão ALLIANCE (PARKER, 1994). Essa segunda interpretação tem como objetivo englo-

bar sob a visão arroviana os problemas de alocação *online*. Ou seja, o conjunto de alternativas passa a ser o conjunto \mathcal{R} e o ordenamento de preferências será realizado para cada nova tarefa incluída.

Em ambos os casos, como será feita a alocação de uma tarefa para um robô e de forma instantânea, não tendo qualquer alocação futura, serão definidas Funções de Escolha Social e de Bem-estar Social Multirrobô para o problema ST-SR-IA. Isso se deve ao fato de se ter, a cada alocação, apenas uma alternativa vencedora, seja ela a tarefa eleita ou o robô eleito. Embora seja necessário apenas um resultado, como as tarefas da missão são conhecidas, o ordenamento completo das preferências individuais é realizado, bem como o ordenamento dos robôs alternativos.

ST-SR-IA- \mathcal{T}

No problema ST-SR-IA- \mathcal{T} , a condição 4.2 é considerada. Assim, o ordenamento das tarefas deve ser realizado par a par, sendo tais comparações independentes entre si. Além disso, o Princípio de Pareto (Condição 4.4) significa que a escolha social deve seguir o princípio de otimização, de forma que a solução é Pareto-ótima se o bem-estar do robô r_i aumenta sem que isso ocasione a diminuição do bem-estar nos demais membros do grupo.

Uma FBS-MR \mathcal{T} é, então, aplicável ao problema uma vez que a preferência social deva ser mapeada a partir da preferências individuais dos robôs da equipe. Entretanto, como para cada robô (ST) uma única tarefa (SR) deva ser a vencedora (IA) até que o robô esteja novamente disponível para executar outra tarefa, o mecanismo de alocação deve respeitar características. Além disso, pode-se discutir a relativa simplificação em se aplicar um mecanismo de alocação centralizado, seja em um dos robôs ou seja em um dispositivo central. Como o objetivo do presente estudo é apresentar uma metodologia para uma aplicação real em um SMR, as vantagens dos sistemas distribuídos serão sempre levadas em consideração. Dessa forma, será exemplificado um mecanismo de alocação que atende às condições impostas para a existência da FBS-MR \mathcal{T} .

Gerkey e Matarić (2004) propõem uma variação do algoritmo guloso para a alocação iterativa de tarefas. Esse algoritmo pode ser adaptado para o caso de ordenamento de preferências. Originalmente, a avaliação é feita com base na maior utilidade do par robô-tarefa ($r_i; \tau_k$). Assim, a tarefa é alocada para o robô com maior utilidade. Para o caso no domínio da Escola Social, a atribuição ocorrerá iterativamente conforme a preferência do robô r_i , como mostrado no Algoritmo 4.1.

Algoritmo 4.1 (Algoritmo Guloso Iterativo para Alocação de Tarefas).

1. Selecione um robô r_i ainda disponível e o atribua a tarefa de sua maior preferência.
2. Remova a tarefa do rol de alternativas disponíveis.
3. Repita os passos 1. e 2. até que todos os robôs tenham sido atribuídos à alguma tarefa.

O Algoritmo 4.1 apresenta um forma muito próxima ao algoritmo chamado ditadura em série simples (*simple serial dictatorship*). Nesse algoritmo, para um ordenamento pré-fixado de agentes, o agente ranqueado no topo escolhe a alternativa de sua maior preferência, o agente ranqueado em segundo lugar escolhe sua alternativa de maior preferência, dentre as alternativas restantes, e assim por diante (ABDULKADIROĞLU; SÖNMEZ, 1998). Mecanismos dessa natureza são bastante aplicados aos problemas de alocação de objetos indivisíveis, como visto na seção 2.7. Esses mecanismos apresentam soluções Pareto-ótimos, segundo Brandt et al. (2016). Bade (2014) afirma que o mecanismo é Pareto-ótimo, pois mapeia em cada perfil de preferências uma alocação Pareto-ótima nesse perfil, e é à prova de estratégia, pois não há perfil de preferência ou agente tal que a comunicação de uma falsa preferência beneficie esse agente. Entretanto, a ordem com que os agentes são processados influenciam o resultado final (mecanismo dependente da trajetória).

Abdulkadiroğlu e Sönmez (1998) propõem o mecanismo ditadura aleatória em série (*Random Serial Dictatorship* - RSD), de forma a contornar o problema de ordenamento de escolha dos agentes. Basicamente, o mecanismo estabelece uma distribuição de probabilidades uniforme com a qual cada agente é ordenado e, assim, faz sua escolha de acordo com suas preferências individuais.

Bogomolnaia e Moulin (2001) ressaltam a simplicidade de se implementar um mecanismo ordinal em comparação com um cardinal: “um mecanismo ordinal permite aos participantes formular apenas essa parte [preferências sobre objetos concretos] de suas preferências e não requer que pensem sobre a escolha sobre as loterias”. A dificuldade em se estabelecer a preferência sobre as loterias é a representação dos resultados por funções de utilidade sob incerteza no modelo vN-M, que segundo Bogomolnaia e Moulin (2001) já se provou inadequado.

Considere $\Pi^{\mathcal{R}}$ o conjunto de permutações simples possíveis sobre o conjunto de robôs \mathcal{R} , ou seja, um conjunto de $P(m, m) = m!$ permutações, sendo m o número de robôs da sociedade onde a permutação deve envolver todos os robôs. Considere, ainda, $\pi \in \Pi^{\mathcal{R}}$ uma permutação dos robôs membros da sociedade, sendo $\pi = \{\pi(1), \dots, \pi(m)\}$ e $\pi(1)$ o primeiro robô da permutação, e assim sucessivamente até $\pi(m)$, o m -ésimo robô da sociedade. Uma permutação π será chamada de política. O Algoritmo 4.1 pode, então, ser adaptado para um algoritmo de ditadura aleatória em série, $RSD(\mathcal{R}, \mathcal{T}, [\succeq], \pi)$, como mostrado no Algoritmo 4.2.

Algoritmo 4.2 (Ditadura Aleatória em Série para Alocação de Tarefas).

1. Determine a política π de probabilidade uniforme $\frac{1}{m}$ com que os robôs serão escolhidos.
2. Selecione um robô r_i de acordo com a política π e o atribua a tarefa de sua maior preferência.
3. Remova o robô da lista de robôs disponíveis e a tarefa do rol de alternativas disponíveis.
4. Repita os passos 2. e 3. até que todos os robôs tenham sido atribuídos à alguma tarefa.
5. Se $n > m$, quando robô r_i terminar a execução de sua tarefa, retorne r_i para a lista de robôs disponíveis.

6. Repita os passos 4. e 5. até que não haja mais tarefas disponíveis.

O Algoritmo 4.2 atende às condições de existência de uma FBS-MR \mathcal{T} com uma pequena relaxação no critério da Não-ditadura, com a consideração da Eficiência *ex post* de Pareto e considerando-o à prova de estratégias.

Um mecanismo de ditadura em série é considerado por Kato e Ohseto (2002) um mecanismo “fazedor de ditadores”, no sentido em que o próprio mecanismo elege ditadores (agentes cuja preferência individual será a única a ser levada em consideração) em série. No Algoritmo 4.2, o primeiro robô escolhido é o ditador para os demais, que são afetados por ele em termos das alternativas (tarefas) disponíveis, mas não podem afetá-lo de volta. Entretanto, ele não é um ditador no sentido mais amplo da alocação de tarefas, pois ele não impõe nenhuma escolha aos demais agentes, em consonância com o que afirmam Satterthwaite e Sonnenschein (1981). Assim como, o segundo robô escolhido é o ditador para os demais, e assim sucessivamente. Bade (2014) ainda afirma que quaisquer dois agentes que submetem a mesma preferência encaram a mesma loteria sobre as alternativas, isto é, tem a mesma probabilidade de escolherem a alternativa de sua maior preferência. “Então, a ditadura aleatória em série trata os iguais de forma igual” (BADE, 2014), dando ao mecanismo uma característica de justiça.

Satterthwaite e Sonnenschein (1981), considerando a dificuldade em atender ao critério da não-ditadura, introduzem o conceito de não-imperiosidade (*non-bossiness*). Um mecanismo é não-imperioso se não há agente capaz de alterar a escolha de outro agente sem que sua própria escolha seja alterada (BADE, 2014).

Como já pontuado acima, o mecanismo é à prova de estratégias e atende ao critério de Pareto. Reforçando a consideração de que o mecanismo é à prova de estratégias, a premissa *i* (página 98) estabelece que os robôs falam a verdade e não são invejosos, logo, não há possibilidade de manipulação das preferências individuais buscando um maior bem-estar individual.

A consideração de eficiência *ex post* se deve ao fato do mecanismo ser eficiente após a apresentação das preferências pelos agentes. Segundo Aziz e Mestre (2014), essa característica faz com que o mecanismo aleatorize sobre o conjunto de soluções Pareto-ótimas alternativas. Ou seja, uma solução é Pareto-ótima quando não é possível encontrar outro conjunto solução de modo que um agente aumente seu bem-estar sem que outro agente tenha seu bem-estar diminuído, ou, formalmente, na Definição 4.9.

Definição 4.9 (Mecanismo *ex post* Pareto-eficiente). Um mecanismo é *ex post* Pareto-eficiente se, $\forall \succeq_{r_i} \in [\succeq]$, não existe política π tal que $\pi \succeq_{r_i} \pi^*$ para todo $r_i \in \mathcal{R}$ e $\pi \succ_{r_j} \pi^*$ para algum $r_j \in \mathcal{R}$.

ST-SR-IA- \mathcal{R}

O problema discutido anteriormente está centrado nas tarefas como alternativa, levando em consideração que a missão completa é conhecida no início da execução. Entretanto, há problemas em que as tarefas são inseridas uma a uma, o que gera uma alocação *online*. Gerkey

(2003) propõe o Algoritmo 3.1 para a solução desse problema. Quando uma nova tarefa é inserida, a mesma é ofertada aos robôs disponíveis e alocada para o robô com maior utilidade em relação à tarefa. Como não há modelo ou conhecimento da próxima tarefa a ser inserida, não é possível fazer um ordenamento de preferências das tarefas da missão, a comparação de preferências em relação às tarefas é prejudicada. Assim, considerando a premissa vi (página 98) e que os robôs têm conhecimento do desempenho de seus companheiros de equipe, pode-se propor uma solução em que os robôs são as alternativas a serem escolhidas.

O problema anterior foi tratado como um problema de atribuição (*assignment problem*). Com a mudança no conjunto de alternativas, o problema ST-SR-IA- \mathcal{R} pode ser relacionado com um problema de votação. Ainda assim, o algoritmo RSD pode ser adaptado para a solução do problema.

A metodologia aplicada a seguir é adaptada de Aziz e Mestre (2014) para o domínio MRTA. Ainda, por simplificação, será considerado um mecanismo que não permite realocação de tarefas. Ou seja, após atribuída uma tarefa ao robô r_i , o mesmo não a ofertará novamente dada a inserção de uma nova tarefa. A realocação é prevista apenas para casos de falha ou execução insatisfatória, o que não será coberto no algoritmo genérico apresentado a seguir, pois necessita de outras estruturas definidas, como meio de comunicação, etc.

A estrutura apresentada por Aziz e Mestre (2014) tem formalização semelhante à definida na seção 2.2.1, apresentando um conjunto de membros da sociedade neste trabalho denotado por \mathcal{R} , um conjunto de alternativas X , e um conjunto de perfis de preferências $[\succeq]$. Entretanto, para o problema em questão, o conjunto de alternativas é o próprio conjunto \mathcal{R} . A cada nova tarefa τ_k inserida no sistema, os robôs atualizarão seu ordenamento de preferências para os candidatos ainda disponíveis em \mathcal{R} .

Considere o subconjunto $B \subseteq \mathcal{R}$ e o ordenamento de preferências \succeq_{r^i} . Seja, ainda, a alternativa de menor preferência (*less preference*) $r_{lp}^{r_i}$, ou seja, o robô classificado pelo robô r_i como o de menor preferência para a execução da tarefa τ_k . Então, o operador mostrado na Equação 4.1 executa a eliminação da alternativa de menor preferência de acordo com o ordenamento de preferências do robô r_i . Ou, por outro lado, separa o conjunto de alternativas de maior preferência do robô r_i .

$$\max_{\succeq_{r_i}^{\tau_k}}(B) = \{B \setminus \{r_{lp}^{r_i}\} : r_{lp}^{r_i} \in B\} \quad (4.1)$$

Considere $\Pi^{\mathcal{R}}$ o conjunto de políticas possíveis de \mathcal{R} , como mostrado anteriormente. O operador visto acima é aplicado, então, à *ditadura aleatória em série Prio*($\mathcal{R}, \tau_k, [\succeq], \pi$) no que diz respeito ao conjunto de robôs \mathcal{R} , à tarefa τ_k , ao perfil de preferências $[\succeq]$ e ao ordenamento π , como mostrado na Equação 4.2.

Basicamente, para a execução da tarefa τ_k , o robô $\pi(1)$ irá retirar das alternativas possíveis seu robô de menor preferência. Assim, $\max_{\succeq_{\pi(1)}^{\tau_k}}(B) = \{\mathcal{R} \setminus \{r_{lp}^{\pi(1)}\}\}$.

O robô $\pi(2)$, a partir de seu ordenamento individual de alternativas $\succeq_{\pi(2)}^{\tau_k} \setminus \{r_{lp}^{\pi(1)}\}$ terá sua alternativa de menor preferência $r_{lp}^{\pi(2)}$ excluída.

Nesse ponto, $B = \mathcal{R} \setminus \{r_{lp}^{\pi(1)}; r_{lp}^{\pi(2)}\}$. Isso ocorre sucessivamente até o robô $\pi(m)$ excluir o candidato de menor preferência entre as duas únicas opções possíveis.

$$Prio(\mathcal{R}, \tau_k, [\succeq], \pi) = \max_{\succeq_{\pi(m)}}^{\tau_k} (\max_{\succeq_{\pi(m-1)}}^{\tau_k} (\cdots (\max_{\succeq_{\pi(1)}}^{\tau_k} (\mathcal{R})) \cdots)) \quad (4.2)$$

O Algoritmo 3.1 pode, então, ser modificado para um algoritmo de alocação *online* baseado no mecanismo RSD, dando origem ao Algoritmo 4.3. A característica de aleatoriedade é dada pela loteria de permutações π sobre o conjunto \mathcal{R} , genericamente, $|\mathcal{R}|!$, com uma distribuição de probabilidade uniforme $\frac{1}{|\mathcal{R}|}$.

Algoritmo 4.3 (Ditadura Aleatória em Série para Alocação Online de Tarefas).

1. Com uma nova tarefa τ_k inserida na missão, determine a permutação π com que os robôs serão avaliados, baseada no conjunto \mathcal{R} e na cardinalidade $|\mathcal{R}|$.
2. Aplique a Equação 4.2 para a permutação do passo 1.
3. Atribua ao robô eleito a tarefa τ_k .
4. Remova o robô eleito do conjunto B de robôs disponíveis.

O mecanismo, que será considerado uma FBS-M \mathcal{RT} , identifica iterativamente (FES-M \mathcal{RT}) o robô com o maior nível de preferência para cada tarefa considerando a preferência do grupo. Como entradas, a função recebe o ordenamento de preferências de cada robô para cada tarefa, $\succeq_{r_i}^{\tau_k}$, com $i = (1, 2, \dots, m)$. A saída mapeia o ordenamento de preferências resultando na escolha da alternativa melhor ranqueada para cada tarefa, de acordo com as preferências individuais.

Assim como no problema anterior, a condição de não-ditadura é relaxada para o uso do algoritmo de ditadura aleatória em série. O princípio de Pareto é atendido, uma vez que as sucessivas eliminações de alternativas de menor preferência tornam a próxima escolha Pareto-ótima. Ou seja, o próximo robô a fazer a escolha não pode melhorar seu bem-estar sem que altere o bem estar do robô anteriormente selecionado. Além disso, fica clara a definição mais simples da condição de unanimidade. Se todos os robôs ranquearem o robô r_j como a maior preferência para a execução da tarefa τ_l , então o par $(\tau_l; r_j)$ será a escolha social, independentemente da ordem em que os robôs forem selecionados no algoritmo de ditadura em série. Ainda assim, é importante ressaltar a característica de eficiência *ex post* do algoritmo. As demais condições levantadas e atendidas para o caso anterior são também atendidas aqui por associação.

Os casos vistos anteriormente tratam a alocação instantânea de tarefas. O Algoritmo 4.2 pode ser modificado para a situação de alocação de tarefas escalonadas ao longo do tempo, atendendo aos problemas TA.

4.3.2 ST-SR-TA

Esta classe de problemas é similar a classe anterior, entretanto, com o escalonamento de tarefas. Segundo Gerkey (2003), esse problema reflete em conhecer as utilidades (preferências) futuras dos robôs. Além disso, exige um conhecimento completo das tarefas da missão ou um modelo de como as tarefas serão incluídas futuramente. Essas considerações refletem um número n de tarefas maior que o número m de robôs, ou seja, $n > m$.

Por se tratar de um problema que exige planejamento futuro, a função aplicada aqui será FBS-MRT, porém, como saída apresentando a possibilidade de um subconjunto $\mathcal{B}_{r_i} \subseteq \mathcal{T}$ de tarefas alocadas a cada robô r_i . Esse subconjunto de tarefas será o planejamento para execução de cada robô, seguindo seu ordenamento de preferências individual e, ainda, considerando as preferências de longo prazo. E, após a alocação de tarefas, $\mathcal{B}_{r_1} \cap \mathcal{B}_{r_2} = \emptyset$.

Brandt et al. (2016) chama essa situação de sequências de escolha (*picking sequences*). Os objetos ou recursos são alocados aos agentes envolvidos de forma incremental numa sequência pré-determinada. O autor ainda destaca que esse mecanismo tem duas propriedades muito atrativas: “é muito simples de implementar e explicar e, ainda, livra a autoridade central da carga de provocar as preferências dos agentes” (BRANDT et al., 2016). Basicamente, os agentes escolhem sequencialmente a alternativa de sua maior preferência que ainda não foi escolhida. A questão é como determinar a sequência de forma a atender as condições para a escolha social apresentadas neste trabalho.

Para o caso em que cada agente escolherá mais de um objeto, há a possibilidade de que o ordenamento de preferências seja feito por alternativa ou por conjuntos de alternativas (*bundles*). Neste trabalho, será considerado o caso mais simples: o ordenamento de alternativas. Isso leva em consideração dois pontos: (i) a natureza combinatória do problema e (ii) simplificação quanto ao perfil de preferências e, conseqüentemente, quanto ao mecanismo de escolha.

Para (i), Bouveret, Endriss e Lang (2010) destacam que o conjunto de *bundles* cresce exponencialmente com o aumento de alternativas. Por exemplo, com um conjunto de vinte alternativas, o agente pode classificar mais de um milhão de *bundles*, considerando um domínio irrestrito para os ordenamentos.

Quanto à (ii), considera-se, por simplicidade, que se o agente apresenta o ordenamento de preferências $(\tau_1 \succeq \tau_2 \succeq \tau_3 \succeq \tau_4)$, indiretamente, é estabelecida o ordenamento de preferências quando aos *bundles* $(\tau_1, \tau_2) \succeq (\tau_3, \tau_4)$, sem perda de generalização.

Aziz, Walsh e Xia (2015) destacam que “o conhecido mecanismo ditadura em série é englobado no conjunto de mecanismos sequenciais”. Por esse motivo, o Algoritmo 4.2 pode ser modificado para que a alocação escalonada no tempo ocorra, originando o Algoritmo 4.4. A discussão será feita acerca da política π ou sequência em que os robôs serão ordenados para realizarem suas escolhas. A política π é equivalente à permutação $\pi \in \Pi^{\mathcal{R}}$ vista na seção anterior, entretanto, com repetições. Sendo o número de tarefas n e o número de robôs m , é possível a repetição do robô, dada a alocação das tarefas escalonada no tempo. Por isso, são possíveis m^n permutações, onde n define o número de posições a serem permutadas.

Seguindo o exposto anteriormente sobre a relaxação da condição da não-ditadura, ainda será levado em consideração que o ordenamento de robôs para a escolha das tarefas será dado de modo aleatório. Isso traz justiça ao mecanismo. Entretanto, considerando que o número de tarefas n é maior que o número de robôs m , não necessariamente duas vezes maior, uma nova rodada de atribuição de tarefas deve ser efetuada, tratando-se de um problema TA.

Na literatura, existem proposições de sequências de agentes para o problema. Considere dois agentes, 1 e 2. A sequência alternada estabelece uma política do tipo $\pi = 121212\dots$, para o número n de alternativas possíveis. Já uma sequência balanceada, estabelece políticas do tipo $\pi = 122121122\dots$. Ainda, é possível que a política siga uma ordem aleatória. A discussão, nesse ponto, gira em torno da justiça e eficiência de cada sequência.

Kalinowski, Narodytska e Walsh (2013) provam em seu trabalho que uma política alternada maximiza o bem-estar social utilitarista para qualquer número de alternativas. Isso quer dizer que, dada uma função de utilidade linear, como a contagem de Borda, é possível avaliar que a atribuição de alternativas a partir de uma sequência alternada dos membros da sociedade maximiza o bem-estar social em relação às outras abordagens. Por esse motivo, o Algoritmo 4.4 estabelece, primeiramente, um ordenamento aleatório dos robôs, dando a mesma probabilidade a todos os robôs de exercer sua preferência antes dos demais, entretanto, para a alocação dos itens restantes, a mesma política π será utilizada, resultando em um mecanismo $RSDTA(\mathcal{R}, \mathcal{T}, [\succeq], \pi)$.

Algoritmo 4.4 (Ditadura Aleatória em Série - Tarefas Escalonadas no Tempo).

1. Determine a política π com que os robôs serão escolhidos, seguindo uma permutação m^n com distribuição uniforme de probabilidade $\frac{1}{m}$.
2. Atribua ao robô $\pi(i)$, com $i = (1, 2, \dots, m)$, sua tarefa de maior preferência ainda disponível no conjunto de tarefas.
3. Passe para o próximo robô de π e remova a tarefa do rol de alternativas disponíveis.
4. Repita os passos 2. e 3. até que todos os robôs tenham sido atribuídos à alguma tarefa.
5. Com $n > m$, repita os passos 2., 3. e 4., para mesma política determinada no passo 1.
6. Repita o passo 5. até que não haja mais tarefas disponíveis.

Sobre a eficiência de Pareto do mecanismo, seja um exemplo adaptado de Brams, Edelman e Fishburn (2003). Considere um conjunto de três tarefas, τ_1 , τ_2 e τ_3 , e dois robôs, r_1 e r_2 , com os ordenamentos de preferências

$$\succeq_{r_1} = \{\tau_1 \succeq \tau_2 \succeq \tau_3\},$$

$$\succeq_{r_2} = \{\tau_1 \succeq \tau_3 \succeq \tau_2\}.$$

Considere todas as permutações possíveis de atribuição de tarefas. Dentre elas existem quatro divisões em que cada robô é atribuído com sua primeira ou segunda tarefa de maior

preferência:

$$\mathbf{A} = (\{\tau_1\}, \{\tau_2, \tau_3\}),$$

$$\mathbf{B} = (\{\tau_2, \tau_3\}, \{\tau_1\}),$$

$$\mathbf{C} = (\{\tau_1, \tau_2\}, \{\tau_3\}),$$

$$\mathbf{D} = (\{\tau_2\}, \{\tau_1, \tau_3\}).$$

Sendo a divisão na forma $\mathbf{A} = (\mathcal{B}_{r_1}, \mathcal{B}_{r_2})$. [Brams, Edelman e Fishburn \(2003\)](#) destacam que outras divisões possíveis, nessa situação, são Pareto-dominadas ou muito desiguais. Por exemplo, a divisão $(\{\tau_1, \tau_2, \tau_3\}, \{\emptyset\})$ é Pareto-ótima, pois não há formas de que r_2 aumente seu bem-estar sem que diminua o de r_1 , porém, atribui todas as tarefas para apenas um robô.

As divisões \mathbf{C} e \mathbf{D} são Pareto-ótimas ou Pareto-eficientes. Não há formas de aumentar o bem-estar de qualquer um dos agentes sem que se prejudique o outro. As divisões \mathbf{A} e \mathbf{B} não apresentam tal característica por, segundo [Brams, Edelman e Fishburn \(2003\)](#), apresentarem uma dependência da preferência sobre um conjunto de preferência, por exemplo, $(\{\tau_1\} \succeq \{\tau_2, \tau_3\})$ ou $(\{\tau_2, \tau_3\} \succeq \{\tau_1\})$. Como já ressaltado anteriormente, neste trabalho serão consideradas apenas preferências sobre as alternativas, e não sobre *bundles* de alternativas.

Ainda para o exemplo, pode-se observar que apenas as divisões \mathbf{C} e \mathbf{D} respeitam as políticas $\pi' = \{r_1, r_2, r_1\}$ e $\pi'' = \{r_2, r_1, r_2\}$ geradas pelo Algoritmo 4.4. Assim, pode-se afirmar que, para $n = 3$ e $m = 2$, o mecanismo é Pareto-ótimo. [Kalinowski, Narodytska e Walsh \(2013\)](#) provam que o mecanismo é Pareto-ótimo para qualquer valor de n , com $m = 2$.

Proposição 1. O mecanismo de ditadura aleatória em série aplicado de forma sequencial ao domínio MRTA é *ex post* Pareto-ótimo para o caso do número de tarefas n maior que o número de robôs m .

Prova. Considere as premissas de que (i) os robôs falam a verdade e não são invejosos, (iv) todos os ordenamentos são racionais, (v) (página 98) os robôs devem ordenar todas as tarefas alternativas disponíveis e (ix) os robôs buscam maximizar o bem-estar social, não apenas o bem-estar individual. Pela Definição 4.9, o mecanismo $RSD(\mathcal{R}, \mathcal{T}, [\succeq], \pi)$ é *ex post* Pareto-eficiente. Considere a extensão desse mecanismo para $n > m$, o mecanismo $RSDTA(\mathcal{R}, \mathcal{T}, [\succeq], \pi)$. Então, o mecanismo para alocação de tarefas escalonadas no tempo é *ex post* Pareto-eficiente. \square

Para o problema com alocação instantânea, foi apresentada uma estrutura similar a uma votação para a atribuição de tarefas *online*. No caso da atribuição escalonada no tempo, é necessário o conhecimento da missão completa, ou pelo menos do modelo com que as tarefas serão inseridas no futuro. Por esse motivo, a análise e proposição de solução se dará apenas para o conjunto \mathcal{T} como alternativas. Ainda assim, é possível estabelecer um mecanismo baseado na eleição dos robôs segundo a preferência da sociedade.

4.3.3 ST-MR-IA

Esse problema requer o esforço combinado de vários robôs para a execução de uma mesma tarefa. Segundo [Gerkey \(2003\)](#), este caso deve considerar a utilidade combinada dos robôs, mas não necessariamente a soma dessas utilidades. No caso do problema arroviano, as preferências dos robôs devem ser agregadas, de forma a resultar em uma coalizão. [Gerkey e Mataric \(2004\)](#) ainda reforçam que este é um problema sensivelmente mais difícil que os apresentados anteriormente, restritos à tarefas de um único robô.

Esse problema é equivalente ao Problema de Particionamento e, apesar de na literatura constarem heurísticas para sua solução, a aplicação em problemas MRTA não é derivada automaticamente de tais soluções, com alguns trabalhos propostos como a arquitetura ASYMTRÉ de [Tang e Parker \(2005\)](#) e a RACHNA de [Vig e Adams \(2006\)](#).

Do ponto de vista da escolha social, o caso mais simples do problema ocorre quando o número de robôs necessários para a execução de cada tarefa é igual à m , ou seja, o mesmo número de robôs membros da sociedade. Nesse caso, a agregação de preferências deve resultar de uma FBS-MR \mathcal{T} que elege uma única tarefa por vez por meio de uma FES-MR \mathcal{T} , ou seja, a partir das preferências individuais, a sociedade de m robôs decidirá qual tarefa $\tau \in \mathcal{T}$ será executada. Consequentemente, a cada término de execução, uma nova escolha será realizada.

Entretanto, em termos práticos, a situação exposta apresenta alguns problemas. Com essa estrutura, a arquitetura de MRTA teria um nível de robustez muito baixo, uma vez que todas as tarefas precisam de todos os robôs. Com a falha, mesmo que parcial, de um robô, toda a missão será condenada. Ainda, uma das principais vantagens de um sistema multirrobô não é empregada, a divisão de tarefas.

Outra estrutura possível é um alocador centralizado, que detém toda a informação do ambiente, inclusive as preferências individuais de cada robô da sociedade sobre a execução de tarefas e sobre as coalizões possíveis. Esse problema é combinatorial, visto que a alocação ótima seria uma solução combinatorial a partir das informações completas disponibilizadas. E o número de estados possíveis cresce exponencialmente com o aumento de tarefas e/ou de robôs no sistema. Entretanto, esse tipo de solução também não atende ao problema MRTA em termos práticos. A característica centralizada não é vantajosa do ponto de vista de SMR com robôs móveis e tal mecanismo necessita uma elevada capacidade computacional.

Uma abordagem mais genérica é necessária. A característica MR do problema gera a necessidade de se formar coalizões entre os robôs e alocar tarefas à essas coalizões. De acordo com a estrutura proposta neste trabalho, tanto a formação da coalizão quanto a alocação de tarefas precisa levar em consideração as preferências individuais. Os dois processos podem ser entendidos como processos distintos, não sendo executados por apenas um mecanismo. A seguir, serão propostas algumas soluções, sendo discutidas suas características sob o aspecto da escolha social.

O problema Tarefas/Robôs

O nome dado à seção faz paralelo com o problema Hospital/Residentes apresentado por Gale e Shapley (1962 apud BRANDT et al., 2016). Nesse problema, tanto hospitais têm preferência sobre os residentes, quanto os residentes têm preferências sobre os hospitais. O problema transportado para o domínio MRTA será abreviado para $\mathcal{T}\mathcal{R}$.

Seguindo a formalização proposta por Brandt et al. (2016), um problema $\mathcal{T}\mathcal{R}$ envolve um conjunto de n tarefas, \mathcal{T} , e um conjunto m robôs, \mathcal{R} . Entretanto, cada tarefa $\tau_k \in \mathcal{T}$ possui um número inteiro e positivo $p_{\tau_k} \in \mathcal{P}$ de *postos*, ou seja, o número de robôs necessários para a execução da mesma. Assim, nessa condição, uma informação a mais é necessária sobre as tarefas.

Como dito, no problema proposto por Gale e Shapley (1962 apud BRANDT et al., 2016), os hospitais tem preferências sobre os residentes, gerando um conjunto de pares hospital-residente aceitáveis. Aqui, não será considerada a preferência das tarefas em relação aos robôs, considerando que a própria natureza da tarefa e seus requisitos farão um ordenamento de preferência implícito. Por exemplo, se a execução de uma tarefa precisa de um recurso específico do robô, como um sensor *laserscan*, então, os robôs que não dispõem, desse recurso terão a menor preferência. Ainda, se dois recursos diferentes podem atender à mesma tarefa, mas com qualidades diferentes, essa diferença afetará o ordenamento de preferências respectivo. Ainda, a estrutura do problema proposta por Gale e Shapley (1962 apud BRANDT et al., 2016) permite listas incompletas de preferências, de forma que uma preferência não classificada é tida como inaceitável. Uma das premissas das análises no domínio MRTA é que os robôs realizam ordenamentos racionais, por isso, completos e transitivos, premissa (iv) (página 98).

Gale e Shapley (1962 apud BRANDT et al., 2016) propuseram um algoritmo para a resolução do problema, chamado Algoritmo de Gale-Shapley orientado ao residente (*resident-oriented Gale-Shapley algorithm* - RGS). Esse algoritmo estabelece a combinação entre as preferências dos residentes e as preferências dos hospitais. De modo que, primeiramente, os residentes solicitam o hospital de sua maior preferência. Os residentes que estiverem melhores colocados nas preferências do hospital, dentro do número de postos de trabalho desse hospital, são alocados à esse hospital. Os demais são rejeitados. Posteriormente, os residentes rejeitados solicitam o hospital com a segunda maior preferência, e assim sucessivamente, até todos os residentes estarem alocados aos hospitais. É importante notar que, durante a execução do algoritmo, o hospital pode rejeitar residentes que foram previamente aceitos, devido ao seu ordenamento de preferências próprio. E, ainda, ao final da execução do algoritmo, podem haver residentes que não serão alocados em nenhum hospital, pois ou não estão na lista de preferências de nenhum hospital, ou estão classificados na lista de preferência em posição além do número de postos possível.

Uma proposição para o domínio MRTA é mostrado no Algoritmo 4.5, chamado de Algoritmo de Gale-Shapley Orientado ao robô - *rGS*. Como poderá ser observado, o algoritmo usará uma instância do Algoritmo 4.2 para determinar o ordenamento com que os robôs revela-

rão suas preferências sobre as tarefas. Como não é considerado um ordenamento de preferências da tarefa, não serão permitidas realocações de robôs nas coalizões formadas.

Algoritmo 4.5 (Algoritmo de Gale-Shapley Orientado ao Robô - *rGS*).

1. Determine a política π com que os robôs serão escolhidos, com distribuição uniforme de probabilidade $\frac{1}{m}$.
2. Atribua ao robô $\pi(i)$, com $i = (1, 2, \dots, m)$, sua tarefa τ_j de maior preferência ainda disponível no conjunto de tarefas e desconte uma unidade de p_{τ_j} .
3. Remova do conjunto de tarefas disponíveis, as tarefas τ_l com $p_{\tau_l} = 0$.
4. Passe para o próximo robô de π .
5. Repita os passos de 2. a 4. até que todos os robôs tenham sido atribuídos à alguma tarefa.
6. Para as tarefas com $p_{\tau} = 0$, a coalizão está formada e a tarefa pode ser executada.
7. Para as tarefas com $p_{\tau} \geq 1$, mova a tarefa para um conjunto de tarefas sem coalizão e repita os passos 2 e 3. para o conjunto de tarefas disponíveis e para os robôs ainda disponíveis em π .
8. Atualize o conjunto de tarefas disponíveis com as tarefas que ainda não formaram coalizões e repita os passo 5. e 6.
9. Repita o passo 8. até que todas os robôs sejam membros de coalizões.
10. Quando pelo menos dois robôs estiverem disponíveis, repita o passo 8.

O algoritmo descrito depende da formação da coalizão para que a tarefa seja executada. De acordo com o passo 6., se uma tarefa não gera uma coalizão, então ela será transferida para o conjunto de tarefas sem coalizão \mathcal{T}_{NC} e, posteriormente, voltará ao conjunto de tarefas disponíveis.

O mecanismo será considerado uma FBS- $\mathcal{MR}\mathcal{T}$, identificando iterativamente uma coalizão com o maior nível de preferência para cada tarefa, considerando as preferências individuais. Como entradas, a função recebe o ordenamento de preferências de cada robô para cada tarefa, $\succ_{r_i}^{\tau_k}$, com $i = (1, 2, \dots, m)$ e a informação p_{τ} de cada tarefa. A saída mapeia as coalizões de p_{τ} robôs necessária para execução de cada tarefa, sendo que os robôs eleitos para cada coalizão têm a tarefa relacionada como a de maior preferência dentre as tarefas disponíveis.

Como visto anteriormente, a condição de não-ditadura é relaxada para o uso do algoritmo de ditadura aleatória em série para a determinação da ordem com que os robôs irão manifestar sua preferência.

Quanto à eficiência *ex post* de Pareto, após a formação de uma coalizão, a troca de qualquer robô do grupo só ocorrerá com a diminuição do bem-estar do robô substituído. Seja a seguinte configuração: $\mathcal{T} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$, $\mathcal{P} = \{p_{\tau_1}, p_{\tau_2}, p_{\tau_3}, p_{\tau_4}\} = \{2, 1, 3, 2\}$ e $\mathcal{R} = \{r_1, r_2, r_3\}$.

Considere os seguintes ordenamentos de preferências:

$$\succ_{r_1} = \{\tau_1 \succ \tau_4 \succ \tau_3 \succ \tau_2\},$$

$$\succ_{r_2} = \{\tau_1 \succ \tau_2 \succ \tau_3 \succ \tau_4\},$$

$$\succ_{r_3} = \{\tau_1 \succ \tau_3 \succ \tau_4 \succ \tau_2\}.$$

Para a política $\pi = \{r_2, r_3, r_1\}$, após o passo 5. do Algoritmo 4.5, tem-se

$$rGS = \{(r_1, \tau_4), (r_2, \tau_1), (r_3, \tau_1)\},$$

$$\mathcal{P} = \{0, 1, 3, 1\},$$

$$\mathcal{T}_{NC} = \{\tau_4\}.$$

Com isso, τ_1 tem a coalizão $\{r_2, r_3\}$ formada e pode ser executada, como $p_{\tau_1} = 0$. Quando o robô r_1 manifestou sua preferência, a tarefa τ_1 já não estava disponível, portanto, sua segunda maior preferência foi manifestada. Entretanto, τ_4 necessita de uma coalizão de dois robôs, não sendo alocada para r_1 e movida para o conjunto de tarefas sem coalizão \mathcal{T}_{NC} . Com apenas uma rodada de alocação do algoritmo é possível perceber que para r_1 ter seu nível de bem-estar elevado, seria necessário diminuir o bem-estar de r_2 ou r_3 .

Após o passo 8. e considerando que a tarefa τ_1 ainda não foi finalizada, à r_1 é atribuída a tarefa τ_2 . Assim, dentre as soluções possíveis nesse estágio, ainda é válida a afirmação sobre o nível de bem-estar de r_1 , r_2 e r_3 . Portanto, o mecanismo proposto, mesmo sendo uma extensão do algoritmo RSD já discutido, continua sendo *ex post* Pareto-eficiente. As demais condições levantadas e atendidas para o caso anterior são também atendidas aqui por associação.

Gale e Shapley (1962 apud BRANDT et al., 2016) propõem um algoritmo semelhante ao RGS, porém, orientado aos hospitais, chamado HGS. Caso as tarefas tenham a descrição das preferências sobre os robôs da sociedade, então um algoritmo semelhante pode ser proposto para essa situação.

Aumento do conjunto de tarefas possíveis

Em um problema de atribuição, quando o número de objetos é diferente do número de agentes e se deseja simplificar o mecanismo de atribuição, são criados objetos ou agentes nulos, de forma que o número de objetos e de agentes seja o mesmo. Assim, caso um agente receba um objeto nulo, na verdade não recebeu nenhum objeto. Da mesma forma, se um objeto foi alocado à um agente nulo, ele não foi alocado à nenhum agente.

A proposta a seguir leva em consideração esse conceito para expandir o conjunto de tarefas possíveis, criando cópias das tarefas originais. A criação das cópias depende diretamente do número p_τ de robôs necessários para a execução de cada tarefa. A formação da coalizão, então, dependerá da escolha da mesma tarefa τ_k por p_{τ_k} robôs.

Assim, o Algoritmo 4.6, outra variação do Algoritmo 4.2, pode ser proposto.

Algoritmo 4.6 (Ditadura Aleatória em Série - Formação de Coalizões por Afinidade).

1. Dado o conjunto \mathcal{P} , copie cada tarefa τ_k p_{τ_k} vezes, criando o conjunto \mathcal{T}' .
2. Determine a política π de probabilidade uniforme $\frac{1}{m}$ com que os robôs serão escolhidos.
3. Selecione um robô r_i ainda disponível e o atribua a tarefa de sua maior preferência.
4. Remova o robô da lista de robôs disponíveis e a tarefa do rol de alternativas disponíveis.
5. Repita os passos 3. e 4. até que todos os robôs tenham sido atribuídos à alguma tarefa.
6. Se $\mathcal{T}' \setminus \{\tau_k\}$, então comunique à coalizão o início da execução de τ_k .
7. Se $\tau_j \in \mathcal{T}'$, então mova a tarefa e suas cópias para o conjunto \mathcal{T}_{NC} e comunique aos robôs a não formação da coalizão.
8. Repita os passos de 4. a 7. até que não haja mais tarefas disponíveis.
9. Retorne as tarefas em \mathcal{T}_{NC} para o conjunto de tarefas possíveis e repita o passo 7. até que todas as tarefas sejam executadas.

O termo “afinidade” foi utilizado para nomear o algoritmo, pois as coalizões serão formadas pelos agentes com maior preferência na execução daquela tarefa, dadas as tarefas ainda disponíveis no conjunto \mathcal{T}' . Mesmo com a formação de coalizões, o algoritmo atende à todas as condições que o algoritmo RSD original atende, já que a decisão é dependente apenas do indivíduo.

Formação ditatorial sequencial de coalizões

Considerando-se as premissas (iii), (v) e (vii) (página 98), pode-se, ainda, propor uma variação a mais do Algoritmo 4.2 para a formação de coalizões. Dados os conjuntos \mathcal{T} , \mathcal{R} e \mathcal{P} , um robô r_i é capaz de ordenar, além das tarefas, a preferência sobre o conjunto $\mathcal{R} \setminus \{r_i\}$ de robôs para cada tarefa, $[\succeq]_{\mathcal{R} \setminus \{r_i\}}^{\mathcal{T}}$.

Algoritmo 4.7 (Ditadura Aleatória em Série - Formação Ditatorial de Coalizões).

1. Determine a política π com que os robôs serão escolhidos, com distribuição uniforme de probabilidade $\frac{1}{m}$.
2. Atribua ao robô $\pi(i)$, com $i = (1, 2, \dots, m)$, sua tarefa τ_j de maior preferência ainda disponível no conjunto de tarefas.
3. Dado o p_{τ_j} da tarefa alocada, forme a coalizão com os $(p_{\tau} - 1)$ robôs melhores colocados em $[\succeq]_{\mathcal{R} \setminus \{r_i\}}^{\mathcal{T}}$ e ainda disponíveis em \mathcal{R} .
4. Remova do conjunto de tarefas disponíveis τ_j e do conjunto de robôs disponíveis r_i e os $(p_{\tau} - 1)$ robôs melhores colocados em $[\succeq]_{\mathcal{R} \setminus \{r_i\}}^{\mathcal{T}}$.
5. Passe para o próximo robô de π .

6. Repita os passos de 2. a 5. até que todos os robôs tenham sido atribuídos à alguma coalizão.
7. Quando pelo menos dois robôs estiverem disponíveis, repita o passo 6.

Apesar de ter certo sentido, o Algoritmo 4.7 viola a condição de não-ditadura de forma mais drástica que o próprio algoritmo da ditadura em série, como já discutido. O robô ditador impõe a formação da coalizão aos robôs de sua maior preferência. Assim, independente da preferência individual desse robô escolhido, a coalizão é formada. É importante observar que para uma aplicação prática, há de se considerar aspectos que o Algoritmo 4.7 não considera, como a comunicação e suas possíveis falhas, a oferta para a composição da coalizão, a aceitação da composição da coalizão, etc. Esse algoritmo define um passo a passo cru de uma escolha unilateral dos membros da coalizão. Justamente por esse motivo, viola a condição de não-ditadura.

4.3.4 ST-MR-TA

“A classe de problemas ST-MR-TA inclui tanto a formação de coalizões quanto a programação [da execução de tarefas]” (GERKEY, 2003). A solução do problema deve levar em consideração a formação da coalizão. Caso as coalizões não sejam dadas e previamente definidas, segundo Gerkey (2003), todas devem ser consideradas para todas as programações de tarefas possíveis.

Por outro lado, se as coalizões para cada tarefa já são dadas, essa é uma instância do Problema de Agendamento de Multiprocessadores (*Multiprocessor scheduling problem*, o qual é \mathcal{NP} -difícil (GERKEY, 2003). Entretanto, Gerkey e Mataric (2004) afirmam ser possível ignorar a componente de escalonamento no tempo, aproximando o problema ao tipo ST-MR-IA. Tal consideração simplifica o problema. Um algoritmo guloso similar ao Algoritmo 4.1 pode ser utilizado, embora Gerkey e Mataric (2004) observem que a qualidade da aproximação é difícil de se determinar.

Como os robôs devem executar as tarefas em conjunto, uma função de bem-estar social no domínio multirrobô mapeia a ordem de execução das tarefas de acordo com as preferências individuais dos robôs. Considere o ordenamento de preferências individual \succeq_{r_i} do robô r_i .

De forma genérica, Reis e Bastos (2017) apresentam o ordenamento de preferências do robô r_m como uma comparação par a par sobre o conjunto \mathcal{T} de tarefas alternativas disponíveis para execução $\succeq_{r_m} = (\tau_1 \succeq \tau_2 \succeq \tau_3 \dots \succeq \tau_n)$, o conjunto $R = (\succeq_{r_1}, \succeq_{r_2}, \dots, \succeq_{r_m})$ de ordenamentos de preferências individuais, definindo a Função de bem-estar social Multirrobô como na Definição 4.4.

Para o caso mais simples, como citado na seção anterior, quando o número robôs necessários para a execução de cada tarefa é igual à m , ou seja, o mesmo número de robôs membros da sociedade, a FBS- \mathcal{RT} tem como entrada o perfil de preferências R e mapeia na saída o ordenamento completo das tarefas a serem executadas pela sociedade. Esse caso apresenta os mesmos problemas já discutidos.

Por esse motivo, o Algoritmo 4.5 será expandido para o caso de escalonamento de tarefas no tempo. Além da alocação de tarefas para as coalizões, também serão escalonadas no tempo as coalizões formadas para a execução das tarefas. Com isso, o robô r_i terá um *bundle* de tarefas alocadas de forma similar ao problema ST-MR-TA. Entretanto, a execução das tarefas dependerá de coalizões com diferentes robôs. Assim, é proposto o Algoritmo 4.8, que apenas aloca as tarefas e forma novas coalizões de forma iterativa até que não hajam tarefas a serem alocadas.

Algoritmo 4.8 (Algoritmo de Gale-Shapley Orientado ao Robô - rGS).

1. Determine a política π com que os robôs serão escolhidos, seguindo a permutação $\frac{1}{m}$.
2. Atribua ao robô $\pi(i)$, com $i = (1, 2, \dots, m)$, sua tarefa τ_j de maior preferência ainda disponível no conjunto de tarefas e desconte uma unidade de p_{τ_j} .
3. Remova do conjunto de tarefas disponíveis, as tarefas τ_l com $p_{\tau_l} = 0$.
4. Passe para o próximo robô de π .
5. Repita os passos de 2. a 4. até que todos os robôs tenham sido atribuídos à alguma tarefa.
6. Para as tarefas com $p_{\tau} = 0$, a coalizão está formada e a tarefa é retirada do conjunto de tarefas disponíveis.
7. Para as tarefas com $p_{\tau} \geq 1$, mova a tarefa para o conjunto de tarefas sem coalizão \mathcal{T}_{NC} e repita os passos 2 e 3. para o conjunto de tarefas disponíveis e para os robôs ainda disponíveis em π .
8. Atualize o conjunto de tarefas disponíveis com as tarefas que ainda não formaram coalizões e repita os passos 5. e 6.
9. Repita o passo 8. até que todas as tarefas tenham coalizões formadas.
10. Inicie a execução das tarefas.

Assim, esse algoritmo nada mais é que uma extensão do algoritmo RGS apresentado para o problema ST-SR-IA e, portanto, mantém todas as características do mesmo.

4.3.5 O problema MT: *multi task robots*

Essa classe de problemas envolve robôs multi-tarefas (*MT*) e permite a execução de, pelo menos, duas tarefas de forma simultânea pelo mesmo robô. Isso torna os problemas mais complexos, pois exige dos robôs uma capacidade computacional maior. Assim como, aumenta a necessidade de sensores e atuadores diferentes constantes em um mesmo indivíduo. Gerkey (2003) descreve que as soluções dos problemas MT-SR-IA e MT-SR-TA são equivalentes às soluções matemáticas descritas para os problemas ST-MR-IA e ST-MR-TA, respectivamente. O problema MT-MR-TA é modelado como “uma instância do problema de escalonamento de máquinas multipropósitos e tarefas que necessitam vários processadores” (GERKEY, 2003, p. 78).

Entretanto, a modelagem não engloba características e limitações inerentemente físicas da aplicação à SMR.

Korsah, Stentz e Dias (2013) afirmam desconhecer trabalhos em MRTA que englobem os problemas MT-SR-IA, MT-SR-TA e MT-MR-TA. Da mesma forma, a pesquisa para esta dissertação não encontrou proposição para esses problemas. Embora hajam modelagens matemáticas para alguns dos problemas, ainda faltam algoritmos de solução, de acordo com Gerkey e Mataric (2004) e aplicação à sociedades de robôs e ao problema MRTA.

Para o problema MT-MR-IA, existe a arquitetura COMUTAR, proposta por Shiroma e Campos (2009). Essa arquitetura baseia-se na determinação de ações do robô (semelhantes aos esquemas da arquitetura ASYMTRE) envolvidas na execução de uma tarefa e na determinação de recursos de partilha limitada. Esses recursos são periodicamente checados e atualizados pelas ações e impõem limitações às execuções das ações.

O protocolo de alocação de tarefas da arquitetura é o *leilão de único turno*. Assim, dada a oferta de uma tarefa por um leiloeiro (um robô, por exemplo), os demais robôs dão suas ofertas para a execução da tarefa. Essas ofertas são baseadas na formação de coalizões de ações internas de cada robô para a execução da tarefa ofertada. O leiloeiro, então, determina a coalizão de ações vencedora e o(s) robô(s) inicia(m) a execução da tarefa. A partir desse mapeamento e formação das coalizões de ações, é possível verificar se duas tarefas podem ser realizadas com o mesmo conjunto de ações ou se uma ação do robô r_i necessita de um recurso limitado do robô r_j . Dessa forma, para um mesmo robô, ou são alocadas duas tarefas para execução concorrente (MT), ou dois robôs executam a tarefa em conjunto (MR), dividindo os recursos limitados necessários, ou, ainda, cada robô executa uma tarefa separadamente. A característica IA deve-se ao período de tempo mínimo com que as tarefas são inseridas no ambiente. Ou seja, a missão completa não é conhecida antecipadamente, o que impede uma alocação escalonada no tempo.

Apesar do exemplo de implementação apresentado, problemas que envolvem robôs multi-tarefas ainda são bastante complexos. Do ponto de vista arroviano, proposta deste trabalho, muitas considerações devem ser feitas para o estabelecimento de um mecanismo de agregação de preferências. Primeiramente, um algoritmo centralizado poderia fornecer uma resposta ótima para os quatro problemas aqui envolvidos, sendo conhecidos o conjunto \mathcal{R} e as características e capacidades de cada robô $r_i \in \mathcal{R}$, o conjunto \mathcal{T} e as necessidades e características de cada tarefa $\tau_k \in \mathcal{T}$ e as preferências individuais dos robôs sobre \mathcal{T} , ou seja, $2^{\mathcal{T}}$ (e sobre as coalizões de robôs possíveis, $2^{\mathcal{R}}$, quando o problema necessitar). Entretanto, esse é um problema combinatorial com um elevado número de possibilidades, o que aumenta significativamente sua complexidade computacional. Além disso, abordagens centralizadas não são o foco deste trabalho, assim como não são o foco da pesquisa em SMR, principalmente no caso de robôs móveis.

Além disso, a aplicação de preferências ordinais nesse tipo de problema revela uma complicação em relação às Condições 4.2 e 4.3. Assim, a preferência individual entre duas alternativas (tarefas ou robôs) não depende apenas dessas duas alternativas. A possibilidade de alocação de mais de uma tarefa para o mesmo robô pode levar, por exemplo, ao seguinte orde-

namento de preferências: $\tau_1 \succeq \tau_2$, mas $\tau_2 \succeq \{\tau_1, \tau_3\}$ ou $\{\tau_2, \tau_3\} \succeq \tau_1$. Com isso, o ordenamento de preferências dos robôs deve ser feito sobre o *power set* do conjunto de tarefas \mathcal{T} . Ainda, é possível levantar a hipótese de que esse conjunto de preferências pode conter preferências intransitivas ou cíclicas.

Uma última observação pode ser feita em relação à preferência condicional inerente aos problemas MT-MR-IA e MT-MR-TA. Preferências condicionais apresentam características de dependência, por exemplo, entre as alternativas, em relação à reputação dos demais agentes, em relação à distribuição espacial dos demais agentes. Essa preferência condicional se deve à necessidade do ordenamento refletir a preferência individual acerca da execução da tarefa τ_k (ou de mais tarefas) em diferentes coalizões dos robôs de \mathcal{R} . As possíveis limitações causadas por preferências condicionais podem levar ou podem necessitar a restrição do domínio de ordenamentos, contrariando a Condição 4.1. Tal característica torna a busca por um mecanismo de alocação mais difícil.

Por esses motivos, esses problemas não serão abordados neste trabalho.

4.4 Discussão

Os algoritmos descritos nesse capítulo para os problemas não consideram certos aspectos práticos de uma arquitetura de alocação de tarefas em SMR. O objetivo é mostrar outra área e outras soluções possíveis para o problema MRTA. Alguns desses aspectos são, do ponto de vista do *hardware*, o meio de comunicação, o poder computacional, limitações de atuadores e sensores, etc, e, do ponto de vista do *software*, a tolerância a falhas, cálculo da utilidade e ordenação de preferências, etc. Apesar disso, é importante notar que as características se mantêm mesmo em um sistema mais elaborado, que leve os aspectos citados em conta. Uma dessas características é a não-imperiosidade no caso de sistemas que permitem a realocação de tarefas. A realocação pode ocorrer por motivos distintos, por exemplo, o abandono da tarefa por um robô que apresentou uma falha, ou pelo surgimento de uma tarefa de maior preferência. Entretanto, o critério ainda continua sendo válido, pois a alteração da alocação não ocorre em via única. Ou seja, um robô não se apodera da tarefa de outro para aumentar o seu bem-estar individual. Ainda, em arquiteturas que não permitem a realocação de tarefas, o critério da não-imperiosidade é um reforço ao critério da não-ditadura. Impedindo, assim, que as preferências de um indivíduo sejam impostas a qualquer outro.

Os algoritmos propostos para problemas com escalonamento de tarefas no tempo leva em consideração as preferências de longo prazo dos robôs, como definido na seção 4.2.3. Em uma aplicação real, com um ambiente dinâmico, as preferências estão suscetíveis à mudanças e as preferências de curto prazo podem não atender ao critério da transitividade, principalmente, quando houver algum tipo de interdependência entre as tarefas. Entretanto, pode ser inerente à arquitetura essa alteração das preferências, de modo a dar maior robustez e tolerância à falhas ao sistema, como poderá ser observado no Capítulo 5.

A consideração de que as tarefas independentes neste trabalho simplificam as proposições aqui feitas. Tratando-se de preferências ordinais, uma relação de maior dependência entre as alternativas geraria a necessidade de definir as preferências condicionais entre as mesmas. Quando as alternativas são as tarefas, para o cálculo de utilidade feito por cada robô, deveria ser incorporada uma função condicional, para avaliar a influência de um novo estado de uma tarefa sobre as demais. Por exemplo, se a tarefa τ_k é alocada, a preferência sobre as demais será alterada? Tratando-se de robôs como alternativa, além das informações das capacidades e desempenho já considerados, um acompanhamento dos estados atuais do robô é necessário para a avaliação constante do ordenamento.

Uma importante característica da uma arquitetura MRTA é a descentralização da alocação, quando não há um planejador ou um alocador central que toma todas as decisões. Os algoritmos aqui apresentados podem ser implementados tanto na forma centralizada quanto na forma descentralizada. De forma descentralizada, a maneira mais direta de se propor a alocação é através de um robô ou vários robôs, como nas arquiteturas baseadas em mercado, na figura do leiloeiro. Entretanto, mesmo sem uma negociação explícita é possível implementar os mesmos passos e princípios para a alocação de tarefas. Essa relação será feita com a arquitetura ALLIANCE no próximo capítulo. Uma ideia básica em sistemas descentralizados de divisão justa é cada agente começar com um objeto e ao longo da execução negociar ou trocar o objeto. Para a arquitetura ALLIANCE é basicamente isso que ocorre, pois as tarefas que o robô é capaz de executar já formam um *bundle* de tarefas. A motivação, aliada ao limiar de ativação, é a função que aloca a tarefa ao robô e as entradas que afetam a motivação de cada tarefa são a forma de “negociação” ou “troca” entre os robôs.

Outro ponto importante de se considerar é o número de robôs e de tarefas em SMR reais. Como visto no Capítulo 3, o número de robôs nas arquiteturas estudadas não ultrapassa uma dezena, de forma geral. Da mesma forma, sistemas situados, não simulados, geralmente não apresentam um número elevado de tarefas, até mesmo por conta do tempo de execução. Então os ordenamentos de preferências não envolvem um número elevado de alternativas, sejam elas tarefas ou robôs. Ainda assim, cabe uma futura verificação da escalabilidade dos algoritmos quando aplicados à SMR situados.

5 Análise arroviana das arquiteturas de MRTA

5.1 ALLIANCE

A fim de exemplificar as condições que serão discutidas, serão utilizados os dados de dois experimentos. O primeiro é o trabalho original de [Parker \(1998\)](#) em que serão analisados os gráficos de níveis motivação interna dos robôs da equipe para três diferentes situações. O segundo trata de simulações da implementação em ROS (*Robot Operational System*) baseada na arquitetura ALLIANCE apresentada por [Reis e Bastos \(2015\)](#) e [Reis e Bastos \(2016\)](#).

O experimento de [Parker \(1998\)](#) consiste em uma missão de localização e contenção do derramamento de lixo tóxico. Para essa missão são programados nos robôs cinco comportamentos:

Find-locs-methodical: Esse comportamento gera ações para localização do lixo tóxico de forma metódica. O robô usa seus recursos para localizar seu ponto inicial e, a partir da movimentação, mapear o ponto de vazamento para informar aos demais robôs da equipe.

Find-locs-wander: Caso o recurso para a localização metódica não esteja disponível ou falhe, há o comportamento para a localização ao acaso. Isso ocorre pois o robô se movimenta pela área do experimento de forma aleatória até encontrar o vazamento. O comportamento metódico tem prioridade sobre a localização ao acaso, entretanto, os dois comportamentos atendem à mesma tarefa: identificar o local do vazamento. Mesmo que sejam geradas ações diferentes para cada comportamento.

Move-Spill(right) e *Move-Spill(left)*: Após a comunicação do local do vazamento, os robôs iniciam a contenção do mesmo, com a retirada do lixo. Os comportamentos *Move-Spill* são responsáveis por mover o lixo para duas áreas designadas à direita ou à esquerda da área do experimento.

Report-Progress: De tempos em tempos o progresso da missão deve ser reportado ao comando. Um robô se encarrega de deslocar-se até determinada localização para enviar a mensagem.

Serão analisadas as preferências dos robôs para três situações: execução sem falhas, execução com falha parcial de um robô e execução com falha total de um robô.

O segundo experimento é uma arquitetura de alocação de tarefas baseada na ALLIANCE e implementada em ROS. Serão utilizados os dados de simulações para a estrutura apresentada por Reis e Bastos (2016). Essa implementação tem quatro comportamentos diferentes para a execução de quatro tarefas distintas. São elas:

vagar-lado-direito e vagar-lado-esquerdo: Esses são dois comportamentos de alto nível compostos por dois comportamentos de baixo-nível: “vagar” e “manter o lado da arena”. O primeiro é responsável pela movimentação do robô pelo ambiente, desviando de obstáculos, executando a exploração. O segundo mantém o robô movimentando-se apenas em uma determinada área do ambiente, definidas como lado direito ou lado esquerdo. A tarefa executada é a exploração do ambiente e sua conclusão se dá após uma determinada distância ter sido percorrida pelo robô.

Proteger-fronteira: Os comportamentos de baixo-nível “encontrar parede” e “seguir parede” formam o comportamento de alto-nível que executa a tarefa de patrulhar as bordas da arena delimitada. A tarefa é concluída quando o robô dá uma volta completa da arena, retornando ao ponto inicial de identificação da parede. Esse comportamento foi programado com um nível de prioridade maior em relação aos demais. Dessa forma, é o comportamento de maior preferência em todos os robôs. Essa preferência pode ser entendida como a preferência de longo-prazo, uma vez que independe da situação atual do sistema.

Reportar Esse comportamento é o único que depende da conclusão dos demais para ser ativado. Quando ativado, executa a tarefa de reportar a conclusão da missão. É formado pelos comportamentos de baixo-nível “ir para área de referência” e “publicar relatório”.

A seguir, serão analisadas as características da arquitetura ALLIANCE para os critérios estabelecidos na seção 4.2.2. O objetivo é usar a axiomática proposta para verificar as características da tomada de decisão coletiva.

Compleitude

Considere primeiramente o sistema não treinado. Do ponto de vista do robô, as tarefas são ordenadas de acordo com a motivação interna, sendo esta motivação alterada constantemente pelas respostas do ambiente e pela comunicação com os demais robôs da equipe. Entretanto, como o ordenamento ocorre de acordo com os níveis de motivação para cada tarefa, que pode ser nulo, o robô não evita a decisão, mesmo que esse ordenamento não seja realizado numa preferência de longo prazo. Assim, o Axioma da Compleitude será entendido como satisfeito.

Com o sistema treinado, ou seja, com a aplicação do L-ALLIANCE, os robôs têm conhecimento de suas próprias capacidades em relação às capacidades dos demais membros. Com isso, o ordenamento leva em consideração esse conhecimento. Basicamente, pode-se entender que um robô terá preferência maior para uma tarefa que ele tem certeza que tem desempenho melhor que os demais membros. Por outro lado, terá uma preferência menor por uma tarefa a

qual outro robô tem desempenho superior. Do mesmo modo, a decisão, ou o ordenamento, não é evitado, o robô tem apenas mais informações para realizá-lo. Assim, o axioma continua sendo satisfeito.

Seguindo a notação utilizada na seção 4.2.2, na arquitetura ALLIANCE as tarefas são as alternativas a serem ordenadas, na forma $\tau_k \succeq \dots \succeq \tau_n$. O ordenamento fraco será considerado por dois motivos. O primeiro é que o controle interno de cada robô não exclui a relação de indiferença entre tarefas, por exemplo, quando o robô não tem os recursos necessários para a execução de duas tarefas. Ou, ainda, quando duas tarefas têm a mesma evolução para seus níveis de motivação. E o segundo motivo é que os robôs são programados para cooperar, não mentem e não são inimigos intencionalmente. Por essas características, o ordenamento não será considerado estrito, mas sim fraco, de modo que um robô abre mão da execução de uma tarefa se há outro robô com melhor desempenho para a execução. Essa característica pode ser observada em duas situações diferentes. Quando os robôs ainda não alocaram a tarefa, a motivação de um robô mais apto para sua execução tem um nível de crescimento maior que o robô menos apto. Estando a tarefa alocada e o robô executor tem um crescimento no seu nível de aquiescência, quando determinado nível é atingido, a tarefa será realocada a um membro da equipe mais apto a sua execução naquele momento.

Entretanto, como a arquitetura é baseada em comportamento e os comportamentos ativados geram as ações necessárias para a execução da tarefa, é importante ressaltar que o ordenamento, por níveis de motivação internos, levará a um ordenamento de comportamentos de alto-nível. Não há um ordenamento explícito de tarefas. A diferenciação é importante pois robôs com comportamentos diferentes e, conseqüentemente, ações diferentes podem executar e concluir a mesma tarefa. A tarefa relacionada ao comportamento é definida pela função $h_i(a_{ik})$, que retorna a tarefa que é executada quando o comportamento a_{ik} é ativado. Com isso em mente, fica mais clara a impossibilidade de comparação de níveis de motivações internas.

Adicionalmente, como o sistema é inerentemente dinâmico, o ordenamento de preferências pode ser alterado a qualquer instante, dependendo do comportamento do sistema. As preferências de curto-prazo dependem do estado do ambiente, o que pode alterar o ordenamento de preferência de longo-prazo. Isso ocorre por conta do número de entradas que alteram a motivação interna do robô. Mesmo que, originalmente, as configurações de cada comportamento, como os limiares de ativação e funções de impaciência, levem a um pré-ordenamento dos comportamentos, durante a execução da missão, as preferências são sensíveis às modificações no ambiente e às ações dos demais robôs. Por isso, pode-se concluir que as preferências definidas como de curto prazo, para o ALLIANCE são preferências condicionais.

Transitividade

Considerando o ordenamento de preferências individual de acordo com as motivações internas, ou seja, com os valores escalares internos calculados pelo robô, não é possível que o ordenamento individual apresente comportamento cíclico. Essa característica se sustenta mesmo

com a variação dinâmica das preferências de curto-prazo, como observado no item anterior. Seja o primeiro exemplo, a Figura 5.1 mostra a evolução da motivação dos robôs no experimento de alocação de tarefas da arquitetura ALLIANCE, com uma execução sem falhas.

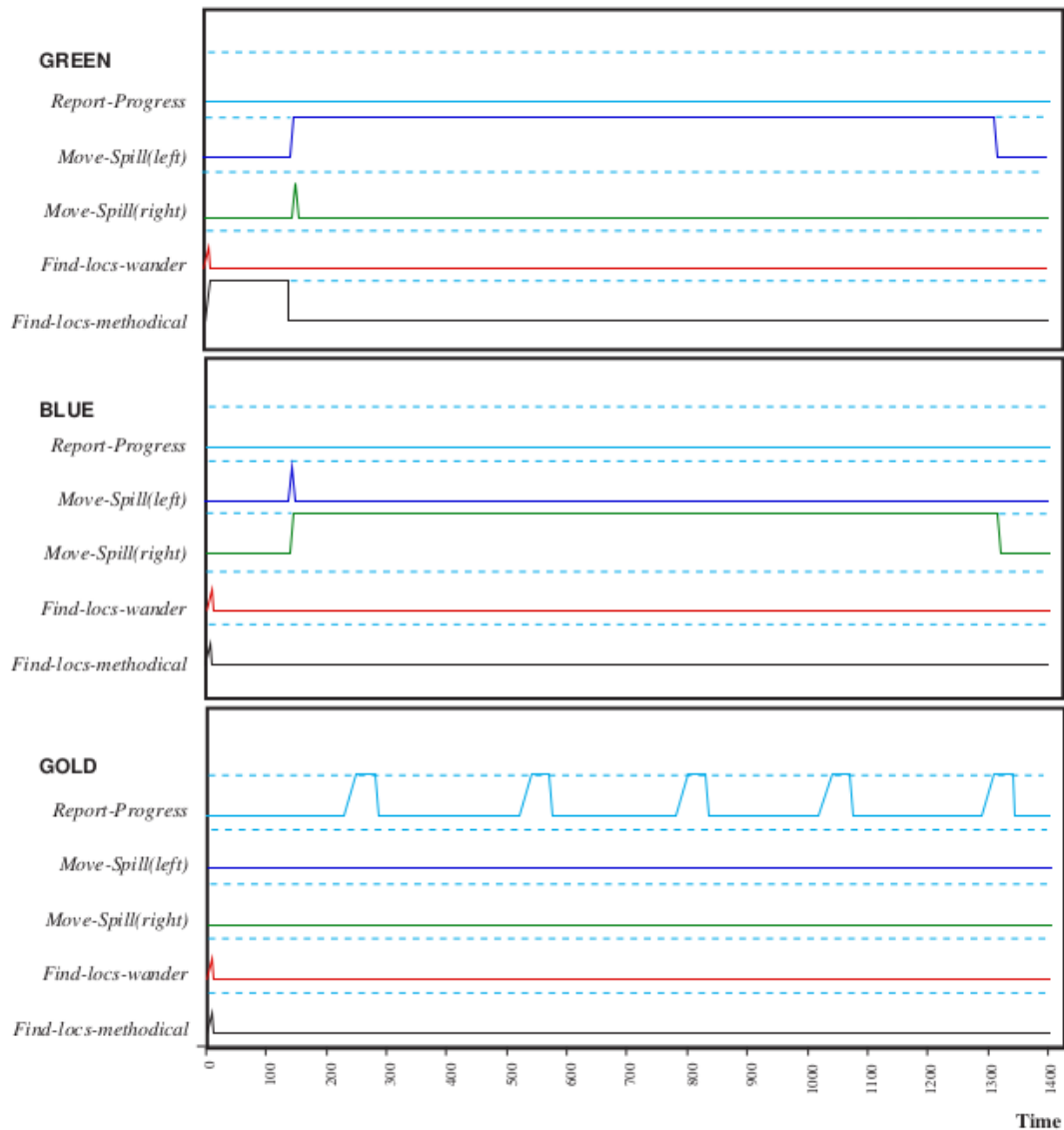


Figura 5.1 – Níveis de motivação interna dos robôs durante avaliação da arquitetura ALLIANCE em experimento de localização e contenção de lixo tóxico, apresentado em (PARKER, 1998). As linhas tracejadas indicam o limiar de ativação dos comportamentos e as linhas coloridas indicam a evolução do nível de motivação para cada comportamento ao longo do tempo. Fonte: Parker (1998).

Nos instantes iniciais da execução (entre 0 e 100 unidades de tempo), é possível perceber a preferência dos robôs para execução da tarefa “identificar local de vazamento”, o que significa ativar ou o comportamento *Find-locs-methodical* ou o comportamento *Find-locs-wander*. E ainda, os demais comportamentos são indiferentes, já que as tarefas de mover o lixo e reportar o progresso ainda não podem ser executadas. Esses comportamentos terão sua preferência alterada de acordo com a conclusão da tarefa de localização. Analisando a evolução das moti-

vações dos robôs *GREEN* e *BLUE* entre 100 e 200 unidades de tempo para os comportamentos *Move-Spill*, pode-se perceber que ambos têm uma preferência maior em relação à ativação do comportamento *Move-Spill(left)*, que é o primeiro comportamento a ser ativado. Com a ativação do comportamento pelo robô *GREEN*, o robô *BLUE* ativa sua alternativa na segunda colocação, nesta situação. Ainda nesse intervalo de tempo, é possível observar que o robô *GOLD* não apresenta preferência em relação à ativação dos comportamentos para a execução das tarefas de movimentação. Entretanto, quando a tarefa de reportar o progresso é uma alternativa disponível, o robô a tem como a alternativa melhor colocada.

No primeiro caso, todos os robôs estão em pleno funcionamento. Pode-se perceber a característica transitiva do ordenamento de preferências. Não há inversão das preferências de curto-prazo, mesmo com a variação das tarefas disponíveis para execução.

No segundo exemplo, cujos níveis de motivação são apresentados na Figura 5.2, há uma falha parcial na execução da tarefa de localização. A falha ocorre no período entre 0 e 200 unidades de tempo. O robô *GREEN* inicia a execução da tarefa, entretanto, não é capaz de concluí-la. Com a demora na conclusão, a motivação do comportamento *Find-locs-wander* do robô *BLUE* começa a aumentar, até que o comportamento é ativado e a tarefa é realocada do robô *GREEN* para o robô *BLUE*.

Neste caso, é possível perceber a mudança no ordenamento de preferências do robô *BLUE* através da diferença da evolução da motivação, causada pelo grau de impaciência do robô. Inicialmente, logo após 0 unidades de tempo, o robô *BLUE* apresenta um ordenamento de preferência como $Find-locs-methodical \succeq Find-locs-wander$, com uma diferença sutil na evolução da motivação. Entretanto, com a falha do robô *GREEN*, o ordenamento passa a ser $Find-locs-wander \succeq Find-locs-methodical$. É importante notar que, como a mesma tarefa é executada com a ativação de ambos os comportamentos, a preferência de execução da tarefa “encontrar localização” não é alterada. Embora haja a alteração no comportamento alternativo para a execução dessa tarefa.

Essa mudança do ordenamento de preferência de curto-prazo exprime um comportamento intransitivo devido à interdependência entre os dois comportamentos e a consideração do estado dos demais robôs da equipe. Dada uma mudança no ambiente, houve a mudança nas preferências individuais do robô.

O terceiro caso apresentado por Parker (1998) envolve a falha total de um robô. Os níveis de motivação são mostrados na Figura 5.3. Com a falha de um robô, os dois robôs remanescentes devem executar as três tarefas ainda não concluídas. Entretanto, uma das tarefas requer dos robôs reportar o progresso da missão, o que deve ser executado mesmo que as tarefas de movimentação do lixo ainda não tenham terminado.

Assim, as preferências de curto-prazo dos robôs alternam o ordenamento dos comportamentos, conseqüentemente, alternando as tarefas. As preferências de longo-prazo não são mantidas, exprimindo comportamento intransitivo, quando há uma preferência condicional que envolve as alternativas. Nesse caso, a prioridade na ação de reportar a missão.

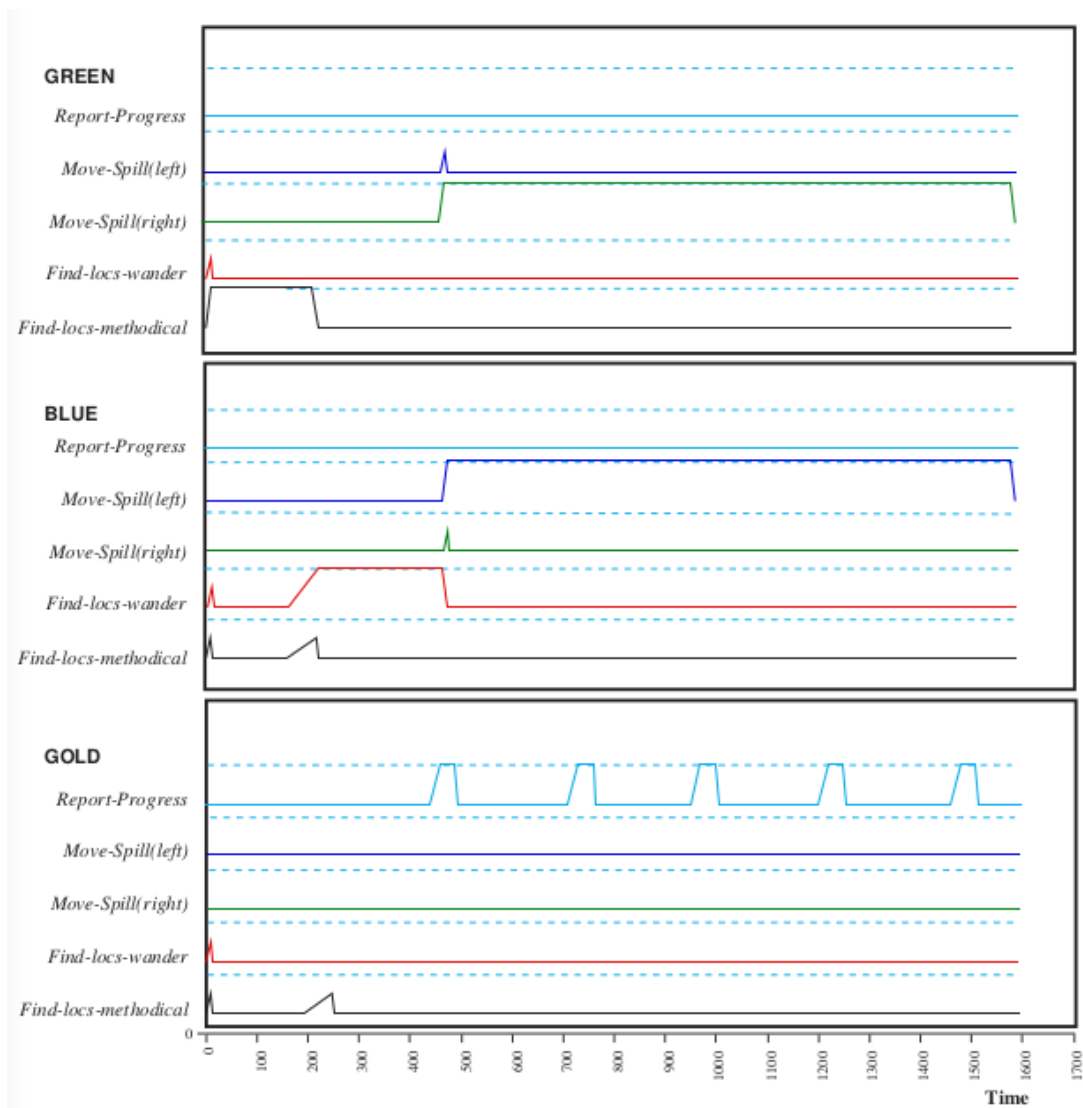


Figura 5.2 – Níveis de motivação interna dos robôs durante a mesma missão representada na Figura 5.1, porém, com uma falha do robô *GREEN* na localização do lixo. A falha, entretanto, não representa uma falha geral do robô, que ainda executa outra tarefa na execução da arquitetura de alocação. As linhas tracejadas indicam o limiar de ativação dos comportamentos e as linhas coloridas indicam a evolução do nível de motivação para cada comportamento ao longo do tempo. Fonte: Parker (1998).

Como forma de comparação, serão mostradas duas simulações da arquitetura baseada na ALLIANCE e implementada em ROS. A principal diferença entre a ALLIANCE e a arquitetura proposta por Reis e Bastos (2015) e Reis e Bastos (2016) é a simplificação do modelo da arquitetura. Sendo a principal simplificação a falta de um mecanismo de aprendizado como o L-ALLIANCE. Assim, as preferências dos robôs dependem dos parâmetros programados pelo desenvolvedor.

Os robôs não sabem das capacidades dos demais robôs da equipe de forma a comparar com suas próprias capacidades e modificar suas funções de motivação interna baseado nas respostas do grupo. Isso causa uma rigidez nas motivações internas e, conseqüentemente, em suas preferências.

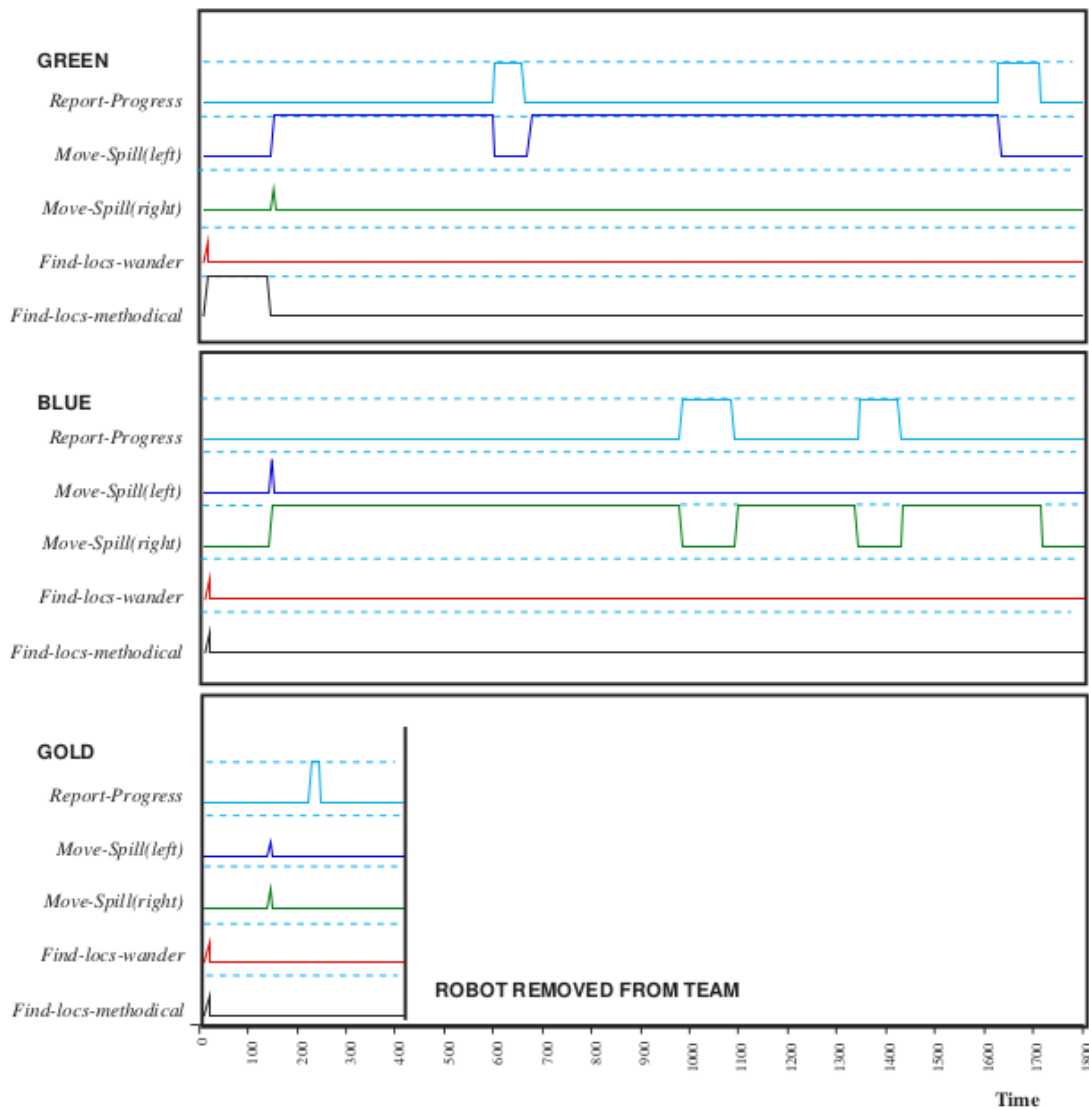


Figura 5.3 – Níveis de motivação interna dos robôs durante a mesma missão representada na Figura 5.1, porém, com uma falha total no robô *GOLD*. O robô foi retirado da equipe. As linhas tracejadas indicam o limiar de ativação dos comportamentos e as linhas coloridas indicam a evolução do nível de motivação para cada comportamento ao longo do tempo. Fonte: Parker (1998).

No algoritmo original de seleção de tarefas, o robô separa as tarefas em dois grupos: (i) aquelas em que sua performance é melhor do que a dos demais e (ii) as outras tarefas as quais o robô consegue executar. Ou seja, separa em duas categorias de preferências. Depois de separar em duas categorias, o robô seleciona tarefas da primeira categoria até que não haja mais tarefas disponíveis. Então, seleciona tarefas disponíveis na segunda categoria, até que não hajam mais tarefas disponíveis da missão.

A arquitetura baseada na ALLIANCE usa os níveis de impaciência e acquiescência como uma forma de exprimir a prioridade das tarefas. Assim, um nível de impaciência maior significa uma prioridade maior de execução daquela tarefa. Esse nível de prioridade dá também o ordenamento de preferências.

A Figura 5.4 apresenta as motivações internas de três robôs ($Robot_0$, $Robot_1$ e $Robot_2$) para a simulação da arquitetura baseada na ALLIANCE. Essa primeira simulação ocorre sem falhas dos robôs. Assim, é possível verificar as preferências dos mesmos quanto a execução das tarefas baseando-se na evolução da motivação. Como explicado anteriormente, o comportamento *Proteger-fronteira* tem a maior prioridade e, com isso, a maior preferência. Este é o comportamento que exprime a maior motivação no início da simulação. Após a alocação dessa tarefa pelo robô $Robot_0$, os demais alocam as tarefas restantes. Ao final da execução das tarefas, um dos robôs deve reportar o fim da missão e, assim, encerrar a missão. Os três robôs apresentam evolução na motivação, sendo alocada a tarefa para o primeiro robô que envia a mensagem informando a alocação, assim a motivação (preferência) dos demais vai a zero.

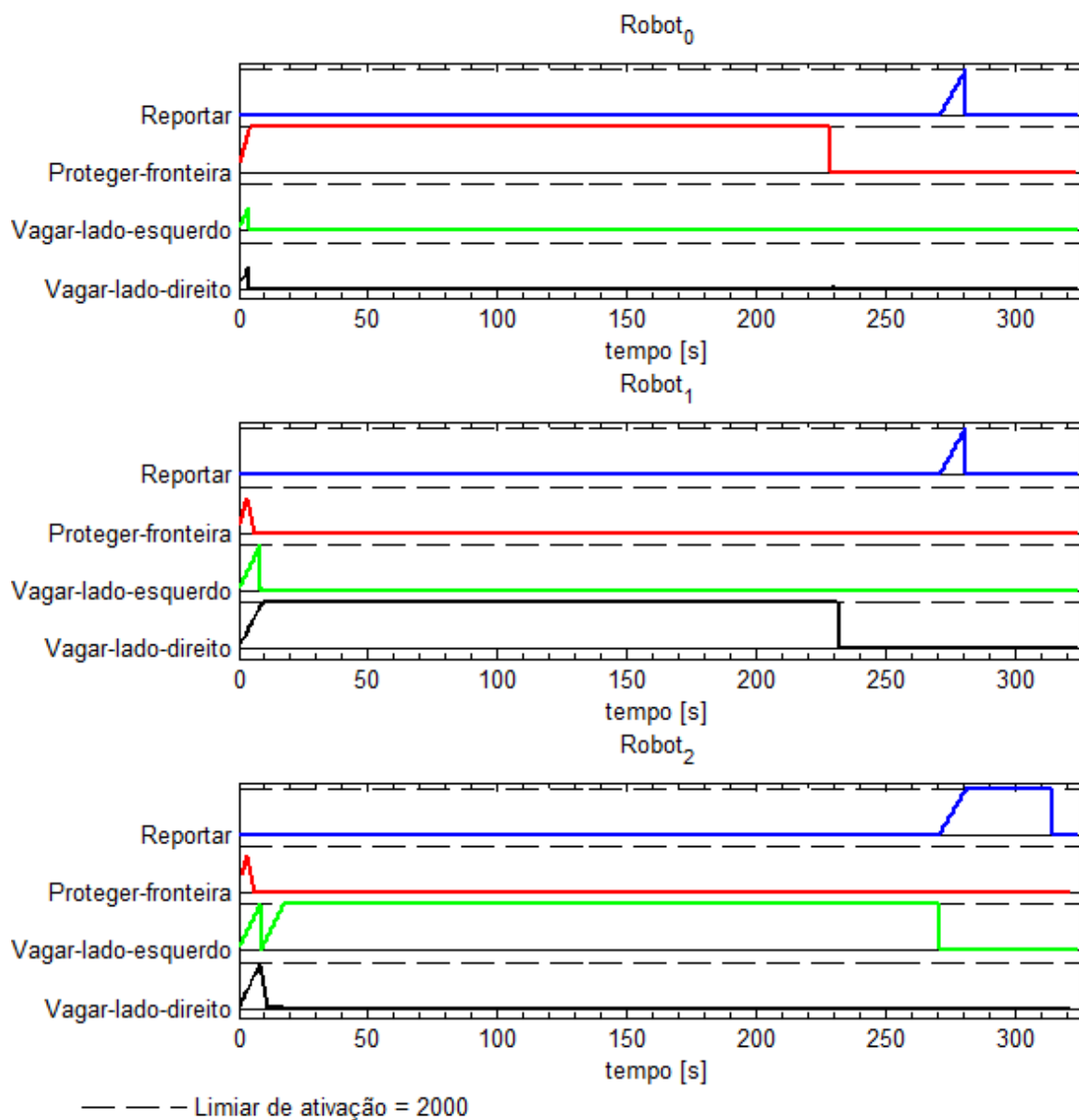


Figura 5.4 – Níveis de motivação interna dos robôs durante a simulação apresentada por Reis e Bastos (2016). A simulação ocorre sem falhas dos robôs. Fonte: Próprio autor.

A Figura 5.5 apresenta as motivações internas de três robôs ($Robot_0$, $Robot_1$ e $Robot_2$) para uma simulação da arquitetura baseada na ALLIANCE em que há uma falha parcial na execução de uma das tarefas. Como verificado na simulação da Figura 5.4, o comportamento *Proteger-fronteira* tem a evolução da motivação mais acentuada que os demais, sendo esta a primeira tarefa alocada. Entretanto, uma falha na execução da mesma (o robô $Robot_0$ travou em determinado ponto do trajeto) faz com que tarefas sejam realocadas.

Quando o robô $Robot_0$ trava durante a execução da tarefa, seu nível de aquiescência se eleva, assim como o nível de impaciência dos demais robôs. Isso leva à realocação da tarefa *Proteger-fronteira* ao robô $Robot_1$. Em consequência, a tarefa abandonada pelo robô $Robot_1$ é realocada ao $Robot_0$, de acordo com sua preferência e com as tarefas disponíveis para alocação naquele momento.

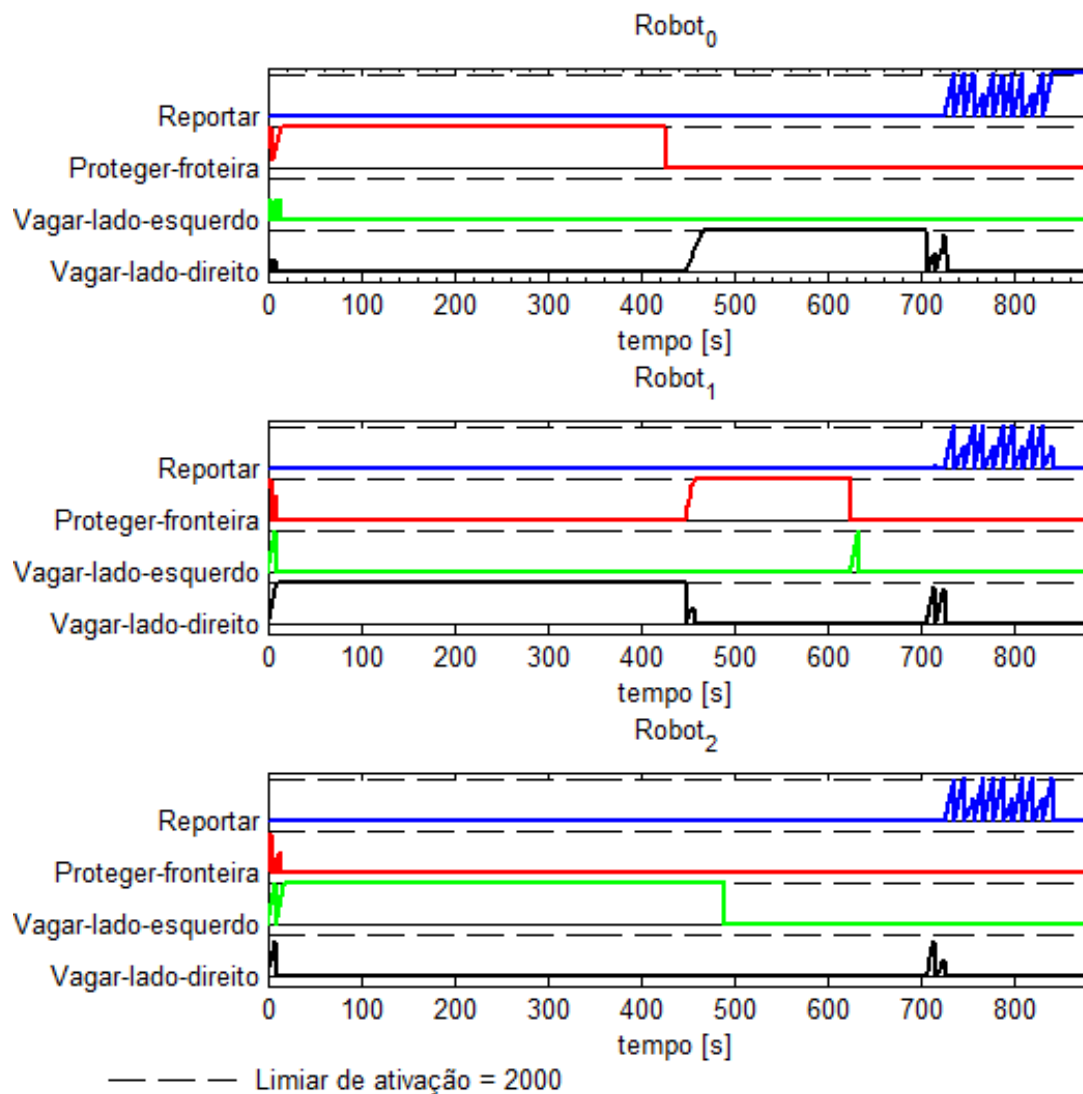


Figura 5.5 – Níveis de motivação interna dos robôs durante uma simulação apresentada por Reis e Bastos (2016) em que há uma falha parcial em um robô, o que causa uma mudança nas motivações internas (preferências individuais) e uma consequente realocação de tarefas. Fonte: Próprio autor.

Entretanto, com a definição da prioridade da execução de uma tarefa em relação as demais, as preferências de longo-prazo são mantidas. Os robôs preferem que a tarefa *Proteger-fronteira* seja executada, mesmo que outra tarefa seja abandonada. Dessa forma, a definição de prioridade, faz com que as preferências de longo-prazo sejam mantidas e faz com que a decisão coletiva seja transitiva, de modo que o bem estar social seja alcançado. Porém, como desvantagem da arquitetura baseada na ALLIANCE, os robôs não têm conhecimento das capacidades dos demais robôs, sendo suas funções de motivação interna engessadas. Isso faz com que o comportamento da arquitetura recaia muito mais na programação desenvolvida pelo desenvolvedor do que nas características próprias do robô, como a resposta dos sensores e a incerteza dos mesmos.

Nas simulações apresentadas, as motivações dos robôs tinham os mesmos valores, ou seja, as preferências de longo-prazo dos mesmos são iguais, de forma que a transitividade foi refletida na escolha social. Na arquitetura ALLIANCE, principalmente após o treinamento utilizando a extensão L-ALLIANCE, cada robô é um indivíduo único, que expressa suas próprias preferências. Assim, como visto nos experimentos apresentados, emerge o comportamento intransitivo na sociedade.

Domínio Irrestrito

Essa condição estabelece que um ordenamento de preferências social deve emergir de um conjunto logicamente possível de ordenamentos individuais. A arquitetura limita o conjunto de alternativas logicamente possíveis pelos próprios sinais de entrada da motivação dos comportamentos. Por exemplo, nas situações das Figuras 5.1, 5.2 e 5.3, a entrada *sensory_feedback* para os comportamentos *Move-Spill* and *Report* só habilita a evolução das motivações desses comportamentos quando o vazamento já foi encontrado, ou seja, quando a tarefa de encontrar o vazamento (seja pelo comportamento *Find-locs-methodical* ou *Find-locs-wander*) está concluída. Assim, não é um ordenamento lógico das alternativas se o robô prefere mover o lixo derramado antes mesmo de identificar a localização do mesmo.

De forma análoga, nas simulações das Figuras 5.4 e 5.5, o comportamento *Reportar* só pode ser ativado após o término das demais tarefas, limitando, assim, os ordenamentos de preferência logicamente possíveis. Esta implementação tem ainda uma outra forma de restrição, que é a prioridade dada a determinada tarefa, ou seja, as motivações (preferências) seguem determinado perfil, dado que um comportamento tem prioridade de ativação.

Essa é mais uma diferença notada entre as duas formas da arquitetura apresentadas. Com os níveis de motivação semelhantes, há, logicamente, a convergência dos ordenamentos de preferências na arquitetura baseada na ALLIANCE. E com isso, o domínio dos ordenamentos possíveis é ainda mais limitado.

Além disso, a arquitetura ALLIANCE apresenta decisões dependentes da trajetória (*path dependent*), ou seja, influenciadas pelas respostas do ambiente, como visto nas análises das motivações dos robôs. Ou seja, de acordo com que as tarefas são alocadas ou realocadas, as

preferências individuais são alteradas, embora a decisão social não seja alterada, já que o mecanismo de alocação (uma FBS) faz o mapeamento das alternativas ainda atendendo à outras condições.

Ainda, é possível perceber que o conhecimento das tarefas e das capacidades dos demais robôs faz com que cada robô tenha, mesmo que de forma implícita, uma limitação interna de preferências dos comportamentos, que segue a classificação de preferência de pico único. Ou seja, mesmo que ao longo da execução as externalidades afetem a preferência do robô, as preferências seguem um padrão de ordenamento, dada a própria característica da missão. De forma implícita, é uma forma de relaxar a condição.

Independência das Tarefas Irrelevantes

O ordenamento individual segue os valores de motivação interna de cada robô. As motivações internas são independentes. Como consequência, a evolução de determinado nível de motivação interno para um comportamento não afeta os demais. Desse ponto de vista, há independência das tarefas irrelevantes internamente.

Entretanto, a motivação de um comportamento a_{ik} do robô r_i é diretamente afetada pela ativação do comportamento a_{jk} de outro robô r_j . Isso resulta em uma interdependência entre funções $h_i(a_{ik})$ e $h_j(a_{jk})$, por exemplo. Essa ativação gera uma restauração na impaciência (*impatience_reset*), que leva a motivação a zero. A partir desse ponto de vista, a ativação de comportamentos, as alternativas de escolha de um robô, são dependentes dos estados dos demais membros da equipe.

Por exemplo, considere que a motivação para o comportamento a_{i1} é maior que a motivação para o comportamento a_{i2} para o robô r_i . Se o comportamento a_{j1} é ativado no robô r_j e r_i recebe a mensagem enviada confirmando a ativação, então a ordem entre os comportamentos a_{i1} e a_{i2} será invertida.

Independência dos Robôs Irrelevantes

Considerando a equipe treinada a partir da extensão L-ALLIANCE, as motivações dos robôs levam em consideração o conhecimento acerca das capacidades dos demais robôs da equipe. Sendo assim, o ordenamento individual dos comportamentos, ou tarefas, considera o desempenho dos demais robôs. Então, o robô terá uma evolução no nível de motivação que depende das capacidades conhecidas dos demais robôs.

Um robô terá maior preferência para a ativação de um comportamento o qual tem conhecimento de seu bom desempenho, enquanto terá menor preferência por um comportamento no qual possui desempenho inferior à algum outro robô membro da equipe. Porém, essa comparação pode ser entendida como uma comparação par a par das alternativas, no caso os comportamentos de alto-nível, e não há dependência de um terceiro robô nessa comparação.

O nível de motivação para todos os comportamentos válidos, aqueles que o robô tem recursos para a ativação, irá evoluir ao longo do tempo. Entretanto, a taxa de evolução da motivação traduz a preferência na ativação do comportamento e consequente execução da tarefa.

Princípio de Pareto

A característica dinâmica já ressaltada da arquitetura faz com que não haja apenas um conjunto de preferências Pareto-dominante, mas sim, um conjunto de conjuntos de preferências Pareto-dominante, que depende da evolução da execução das tarefas e das condições dos robôs (por exemplo, se houve falha de algum robô).

Seguindo este conceito, a arquitetura ALLIANCE faz escolhas Pareto-eficientes. Na Figura 5.1, por exemplo, a alocação de tarefas ocorre de forma que todos os robôs executam determinada tarefa, ativando o comportamento relacionado, e a missão seja cumprida. A ativação dos comportamentos nos robôs faz com que os demais busquem ativar outros comportamentos para a execução de outras tarefas. Nas Figuras 5.2 e 5.3, em que falhas ocorrem, é possível perceber que, ainda assim, a escolha para execução da tarefa é Pareto-eficiente, escolhendo o robô com maior motivação interna (preferência) para a execução da tarefa.

As decisões sociais tomadas visam sempre a maximização do bem-estar social. Uma outra forma de descrever uma alocação Pareto-dominante é quando não há outra alocação de tarefas possível a qual melhore a situação de um robô sem prejudicar a situação dos demais. A característica de aquiescência é essencial para tal comportamento, quando um robô abdica da execução de uma determinada tarefa quando não a está executando de forma satisfatória. Ou seja, o bem-estar do robô aumenta se a tarefa for executada, seja por ele, seja por outro membro do time. Com a realocação de tarefas nesses casos, busca-se uma alocação Pareto-dominante, de forma que o bem-estar de todos os robôs aumente.

No caso específico da Figura 5.3, quando um robô é retirado da missão, três tarefas devem ser desempenhadas pelos dois robôs restantes. Soma-se a isso o fato de uma das tarefas (*Report*) necessitar a execução antes do final das demais, de forma a reportar o progresso da missão. Assim, as preferências de curto-prazo são alteradas conforme o grau de impaciência desse comportamento cresce, fazendo, então, com que a motivação para esse comportamento evolua e o mesmo seja ativado. Nessa situação, a preferência do robô em continuar a executar uma tarefa já alocada muda em relação à execução dessa nova tarefa. Essa decisão também é Pareto-eficiente, uma vez que o bem-estar do indivíduo cujo comportamento *Report* é ativado aumenta sem que o bem-estar do outro membro da equipe diminua.

Assim, os robôs buscam sempre por uma alocação de tarefa Pareto-dominante (Definição 2.5) e, conseqüentemente, ótima. Entretanto, como o problema é \mathcal{NP} -Difícil, é conhecido que a solução ótima pode não ser atingida (PARKER, 1994).

Ditadura

Os níveis de motivações internas do robô para cada comportamento é afetado pelos demais robôs, sejam pelas mensagens trocadas, sejam pelas ações dos demais robôs no ambiente. Assim, mesmo a preferência individual leva em consideração a preferência coletiva, de modo que a preferência social não tem a característica ditatorial.

Mesmo que as tarefas sejam executadas de acordo com o ordenamento de preferências original de determinado membro, a troca de mensagens e a ativação de comportamentos gera mudanças no ordenamento individual. Portanto, o ordenamento social é resultado da interação entre as preferências individuais e, por isso, não pode ser ditatorial.

De outro ponto de vista, a arquitetura possui ditadores aleatórios em série, como será discutido sobre o mecanismo de alocação, a seguir. Isso faz com que o conceito forte de ditadura não se aplique. Mesmo que um robô faça suas escolhas levando em consideração apenas suas motivações internas e a ativação do comportamento de alto-nível correspondente à tarefa, ele não impõe nenhuma tarefa diretamente a outro robô. Ainda assim, esse comportamento pode ser entendido como o pior caso, como na arquitetura baseada na ALLIANCE, em que os robôs desconhecem a capacidade de seus companheiros.

Mecanismo de alocação

A alocação de tarefas na ALLIANCE é feita sem uma negociação explícita entre os robôs. Do ponto de vista de um robô específico, a partir da troca de mensagens por difusão, com as informações já apresentadas, os comportamentos nos demais membros da equipe é suprimido ou ativado. Como pode-se perceber nos exemplos das arquiteturas, tanto a original quando a proposta por [Reis e Bastos \(2016\)](#), a motivação de dois robôs pode evoluir de maneira muito similar, embora apenas um ative o comportamento e execute a tarefa. Como a inibição de um comportamento depende do envio e do recebimento de mensagens e considerando que o meio de comunicação não é perfeito, para o caso descrito anteriormente, a ativação do comportamento se dará, aproximadamente, de forma aleatória.

Ainda, considerando que as tarefas em um sistema situado levam um tempo para serem executadas e concluídas, pode-se relacionar o algoritmo de alocação da ALLIANCE com o Algoritmo 4.2 com uma política π alternada, ou seja, todos os robôs têm a mesma probabilidade de manifestar suas preferências melhores ranqueadas e o fazem de maneira alternada (dado o tempo de execução da tarefa).

A implementação na ALLIANCE tem, ainda, a característica (desejável) da descentralização total do ordenamento de preferências e do controle em cada robô das tarefas executadas e ainda disponíveis. Por analogia, cada robô tem suas próprias funções de escolha social e bem-estar social, que são atualizadas pelas alterações no ambiente e nos estados dos demais robôs. A função de escolha social é entendida como a escolha instantânea do robô (IA), ordenando as motivações e executando a tarefa que atinge o limiar de ativação. Já a função de bem-estar social trata o ordenamento completo dos comportamentos internos, conseqüentemente, tarefas,

de acordo com as preferências individuais.

Do ponto de vista da sociedade, ou seja, do sistema como um todo, o resultado final da missão tende a uma alocação de *bundles* de tarefas a cada um dos robôs. Portanto, a ALLIANCE não compara a motivação (utilidade) dos robôs para uma mesma tarefa de forma cardinal, atribui as tarefas de forma aleatória e sequencial, atribui as tarefas de modo Pareto-eficiente, buscando maximizar o bem-estar social, e à prova de estratégias, dada a independência no ordenamento dos comportamentos e à premissa de que os robôs não mentem. Por esse motivo, o mecanismo pode ser considerado como um mecanismo arroviano de escolha social.

Uma função de bem-estar social para a ALLIANCE é $\succ_W^{\text{ALLIANCE}}: [\succeq]_{\mathcal{R}} \mapsto U^{|\mathcal{R}|}$, que mapeia as preferências individuais dos robôs em um conjunto U de conjuntos de comportamentos que serão ativados em cada robô durante a missão, ou seja, as tarefas que serão alocada em cada robô.

5.1.1 O Problema de Eficiência do ALLIANCE sob a visão arroviana

O problema, apresentado na seção 3.4.1.1 apresenta as condições e as funções que devem ser minimizadas para a otimização da alocação de tarefas. A primeira condição do problema define que, para todos os robôs, um conjunto U_i de ações eleitas pelo robô r_i é um subconjunto do conjunto de comportamentos A_i . Sendo as ações escolhidas em U_i responsáveis pela alocação de tarefas para o robô r_i durante a missão, o conjunto será considerado como um ordenamento de preferências de r_i sob os comportamentos alternativos constantes em A_i . Ainda, como a mesma tarefa pode ser executada com um conjunto de ações diferentes para cada robô, as preferências das ações a serem executadas, consequentemente tarefas, são individuais e incomparáveis.

Deste ponto de vista, o ordenamento de preferências de cada robô poderá ter qualquer ordenamento logicamente possível de ações. Esse ordenamento possível deverá atender às limitações do próprio robô, como a falta de recursos, e as características das tarefas, como prioridade de execução ou dependência da conclusão de tarefa antecessora. Adicionalmente, tal ordenamento deve ser completo e transitivo.

O axioma da completude requer que seja feito um ordenamento de preferências, mesmo que U_i seja um subconjunto de A_i e o ordenamento não seja feito sob todas as alternativas possíveis. A transitividade estabelece que as ações escolhidas pelo robô sejam fruto de um ordenamento transitivo de preferências. Esse ordenamento deve levar em consideração os recursos e capacidades do robô. Não há preferência de execução por uma tarefa a qual não se pode executar. Ainda, deve-se considerar a relação entre as tarefas, prioridades de acordo com a missão e relação de antecedência e sucessão. As duas condições são necessárias à racionalidade do robô em relação à escolha das ações em U_i .

A segunda condição impõe, como já foi discutido anteriormente, que dois comportamentos diferentes a_{ij} e a_{ik} presentes em robôs diferentes podem ser ativados e executar a mesma tarefa $task_j$ se $task_j = h_i(a_{ij}) = h_i(a_{ik})$.

As funções objetivo de otimização, para as métricas tempo e energia, devem ser minimizadas buscando uma solução ótima. Para tal, os robôs têm a informação da qualidade $q(a_{ij})$ de si próprio e dos companheiros acerca de cada comportamento a_{ij} . Essa informação é obtida a partir do treinamento da equipe pelo L-ALLIANCE. Esse valor é um valor escalar determinado a partir das métricas relacionadas a cada comportamento. Quando a variável tempo deve ser minimizada, deseja-se minimizar o valor máximo do somatório o tempo gasto na execução dos comportamentos a_{ik} a cada robô r_i da equipe. Sendo a grandeza da função objetivo a energia, deseja-se minimizar a energia total gasta para a execução dos comportamentos a_{ik} para cada robô r_i da equipe.

Parker (1994) ressalta que a solução ideal da arquitetura envolve cada robô escolhendo individualmente suas ações de forma que um resultado global ótimo seja atingido. Essa consideração é feita para que não seja sacrificada a característica de sistema totalmente descentralizado da arquitetura com o uso de um tomador de decisões centralizado, mesmo que a solução global não seja ótima, ou não seja atingida em um tempo aceitável (PARKER, 1994).

Com isso em mente, a decisão social do grupo de robôs que maximiza o bem-estar social emergirá das decisões individuais dos robôs. As preferências individuais agregadas levarão a uma maximização do bem-estar social. Entretanto, é importante lembrar que mesmo as decisões individuais levam em consideração as capacidades dos demais robôs (L-ALLIANCE) e as mensagens enviadas pelos demais robôs, que trazem informações sobre seus estados atuais. Assim, apesar de não haver comunicação explícita ou qualquer tipo de negociação entre os robôs, as decisões individuais avaliam a situação do grupo.

A maximização do bem-estar social, então, se dará com o atendimento dos Axiomas 2.1 e 2.2, principalmente, e a partir da aplicação do princípio de Pareto, que é um critério de eficiência. Sob a visão arroviana, as funções objetivos com métricas tempo e energia devem ser maximizadas indiretamente, ou seja, não com base em valores escalares calculados, mas, sim, baseando-se nas preferências completas, transitivas e Pareto-dominantes dos robôs. Além disso, para que a função de bem-estar social exista, ela deve atender às demais condições da estrutura arroviana.

Considerando que o L-ALLIANCE treina a equipe para que as capacidades dos membros sejam conhecidas mutuamente de forma que a qualidade na execução das ações governe a motivação de ativação dos comportamentos e, conseqüentemente, na execução das tarefas, quando treinada para um domínio particular, a equipe dispõe de um conjunto de motivações Pareto-dominante. Ou seja, o conjunto de qualidades medidas para os comportamentos ativados torna as motivações dos robôs Pareto-dominante no sentido de que o ordenamento de preferências de execução da tarefa respeitará todos os membros do grupo aumentando o bem-estar do robô executor sem diminuir o bem-estar dos demais robôs da equipe. Os demais robôs têm as informações que tornam o primeiro o mais indicado para aquela execução.

5.2 MURDOCH

A arquitetura, em comparação com as outras duas aqui apresentadas, tem uma particularidade: a alocação *online* de tarefas, quando os robôs não conhecem a missão completa ou o modelo com o qual as tarefas serão inseridas. Caso o modelo fosse conhecido, seria possível estabelecer um ordenamento de preferências sobre o conjunto de possíveis tarefas. Assim, com a inserção de uma nova tarefa, o robô poderia manifestar sua preferência dependendo de seu ordenamento individual, ou aguardar até que uma tarefa de maior preferência fosse inserida no sistema. Como o conjunto de tarefas não é conhecido, então o ordenamento de preferências sobre as tarefas fica prejudicado.

Por esse motivo, a arquitetura não será analisada do ponto de vista arroviano para as tarefas, mas sim para os robôs como alternativas. Isso ocorre quando o robô responsável por alocar uma tarefa, o leiloeiro, a recebe e precisa negociá-la com os robôs interessados. Na arquitetura, o robô compara o grau de aptidão (utilidade) de cada robô para essa tarefa e determina o vencedor de acordo com o maior grau de aptidão comunicado.

Do ponto de vista arroviano, analisar a arquitetura modificando a comparação das aptidões (valores cardinais) por uma comparação de preferência do próprio leiloeiro parece não fazer sentido, pois as preferências individuais do robô leiloeiro serão impostas aos demais robôs. Mesmo que o leiloeiro não seja sempre o mesmo robô. Assim, o critério de não-ditadura seria logo violado. Ainda, outros critérios não podem ser analisados, por não caber qualquer comparação entre a arquitetura original e as condições da escolha social no domínio MRTA propostas nesse trabalho.

Por esse motivo, mais que uma análise, o que se segue é uma sugestão de modificações na arquitetura, considerando as premissas da seção 4.3, para que a arquitetura englobe um mecanismo de escolha social para a alocação de tarefas. Importante ressaltar que não se espera aqui propor modificações que melhorem a qualidade da solução. De certo, não se intenta piorar e, ainda, qualquer comparação prática (seja simulando ou em sistemas situados) não é o objetivo deste trabalho. Porém, é uma adaptação à visão arroviana. Para isso, o problema será considerado $ST-SR-IA-\mathcal{R}$, conforme descrito na seção 4.3.1. Com isso, as alternativas passam a ser os robôs, o mecanismo de alocação pode ser comparado a uma estrutura de votação e o leiloeiro irá recolher os “votos” (preferências) dos robôs e eleger o robô para a execução da tarefa. Em tempo, a discussão se encerra no mecanismo de alocação de tarefas e as vantagens, desvantagens e complicações da implementação prática não serão abordadas.

Completude e Transitividade

Para a adaptação proposta, as alternativas para escolha social são os robôs e dentre esses será escolhido um robô para executar a tarefa inserida no sistema. Considerando que a premissa (vii) é atendida, os robôs, além de estarem cientes da existência dos demais membros da equipe, têm conhecimento do desempenho dos demais robôs da equipe para as possíveis tarefas propostas. Com isso, o axioma trata da ordenação de preferências sobre o conjunto de robôs.

Para o atendimento ao axioma, o robô não pode evitar a decisão, comunicando ao leiloeiro qual é o seu ordenamento de preferências. Dessa forma, a cada nova tarefa inserida, um ordenamento de preferências deve ser realizado por todos os membros sobre os membros ainda disponíveis. O leiloeiro terá o papel de agregador de preferências, de acordo com o Algoritmo 4.3.

Para um ordenamento completo, os robôs devem individualmente comparar suas preferências entre os demais robôs levando em consideração seu desempenho. Uma função de utilidade pode ser utilizada para quantificar a preferência sobre um robô e tornar a comparação entre dois robôs mais simples. Entretanto, esse valor não será comparado com o calculado por outro robô. As preferências ordinais feitas par a par serão comparadas.

Por esse motivo, os ordenamentos devem ser transitivos. O cálculo efetuado pelo robô para ordenar as preferências individuais deve se traduzir em um ordenamento transitivo. Isso quer dizer que espera-se da função de utilidade uma característica linear. Esse cálculo está presente na arquitetura de Gerkey e Matarić (2002) e é baseado nas métricas da tarefa. Basicamente, a sugestão é que se expanda o cálculo para o cálculo da aptidão para todos os robôs e, assim, se estabeleça um ordenamento racional das preferências. Entretanto, é sabido que para isso, a premissa (vii) deva ser atendida.

A decisão será racional se os robôs não evitarem a decisão sobre os robôs ainda disponíveis e se o ordenamento de preferências for transitivo. Na arquitetura original, não há alternativas (tarefas) para se ordenar e, portanto, não há ordenamentos transitivos.

Domínio Irrestrito e Ditadura

Para a arquitetura original, o domínio das alternativas é singular. Há apenas uma tarefa por vez e os robôs não têm o conhecimento se haverá outra tarefa inserida no sistema.

Para o caso sugerido, a condição é atendida se, dadas todas as possibilidades de ordenamento dos robôs, ainda assim é possível emergir uma decisão social. Com o mecanismo proposto (Algoritmo 4.3), dado qualquer ordenamento logicamente possível dos membros da sociedade, haverá uma decisão social, devido, principalmente, à relaxação da condição da não-ditadura.

Utilizando uma ditadura aleatória de forma a eliminar as alternativas de menor preferência iterativamente, o critério do domínio irrestrito também é relaxado, já que, iterativamente, o conjunto de ordenamentos possíveis é reduzido, pois as tarefas serão alocadas e os robôs serão retirados do conjunto de alternativas possíveis. Dessa forma, ambas as condições são relaxadas para a aplicação do mecanismo.

Independência das Tarefas e dos Robôs Irrelevantes

Como tanto a arquitetura original quanto a proposta não são centradas nas tarefas como alternativas, mas sim nos robôs, a condição 4.3 não se aplica à análise nem à sugestão de adaptações. Entretanto, é importante notar que, a cada tarefa, as preferências dos robôs poderão ser

diferentes. Isso se deve às diferentes capacidades, recursos e níveis de desempenho dos robôs da sociedade.

Já a independência dos robôs irrelevantes é uma restrição aos ordenamentos de preferências individuais. A condição exige que a comparação par a par entre duas alternativas, nesse caso robôs, dependa apenas dessas alternativas. Com a alocação de tarefas, o conjunto de robôs disponíveis é alterado, pois o número de alternativas decresce. Assim, se o ordenamento não for independente, então a variação nos robôs disponíveis causará uma variação no ordenamento de preferências. Basicamente, o ordenamento de preferências deve seguir a mesma ordem para os agentes disponíveis se uma tarefa é ofertada no início da missão ou se é ofertada durante a missão.

Mecanismo de Alocação e o Princípio de Pareto

O mecanismo de alocação é o Algoritmo 4.3. A partir das preferências dos robôs, o leiloeiro agrega as preferências de forma aleatória, elegendo um robô para a execução da tarefa. Assim, a função de bem-estar social é $\succ_W^{\text{MURDOCH}}: [\succeq]_{\mathcal{R}} \mapsto \mathcal{R}$.

A eficiência *ex post* de Pareto-eficiência é mantida, como discutido na seção 4.3.1.

5.3 M+CTA

Completude

No protocolo M+, o robô tem conhecimento de todas as tarefas da missão e elabora seu planejamento individual com base em seu próprio conhecimento do ambiente e seus recursos. Dessa forma, o planejamento individual pode seguir qualquer ordenamento, sendo que o cálculo do custo (utilidade) para cada tarefa dependerá da situação atual do ambiente, das tarefas disponíveis para execução e quais as capacidades do robô. O planejamento do custo para cada tarefa gera um ordenamento individual, sendo as tarefas de menor custo mais preferíveis às tarefas de custo mais elevado. O robô não evita a decisão.

Transitividade

Os robôs recebem, simultaneamente, todas as tarefas da missão parcialmente ordenadas. A descrição utilizada ordena as tarefas explicitando qual tarefa deve ser executada antes de outra de forma binária, com a notação “ \prec ” (*a* “precede” *b*). Por exemplo, a partir do ordenamento parcial $T_0 \prec T_3$, $T_1 \prec T_3$, $T_0 \prec T_9$, $T_1 \prec T_9$, $T_3 \prec T_5$ e $T_4 \prec T_5$ (BOTELHO; ALAMI, 1999) do conjunto de tarefas T_0, \dots, T_9 , os robôs devem planejar suas ações de acordo com as tarefas predecessoras e sucessoras provenientes da interpretação do ordenamento. As tarefas não relacionadas tem a ordem de execução indiferente em relação às demais. Por outro lado, a tarefa T_3 é necessariamente precedida das tarefas T_0 e T_1 , que tem sua ordem de execução indiferente. Além da ordem de execução indiferente, as tarefas T_0 e T_1 não tem restrição quanto

aos robôs que as executarão. A tarefa T_3 só estará disponível para execução após a conclusão de suas tarefas precedentes. O mesmo raciocínio se aplica às tarefas T_5 e T_9 .

Esse ordenamento parcial se refere às características e prioridades das tarefas. Mesmo com um ordenamento parcial das tarefas, este é acíclico, ou seja, o ordenamento das tarefas é transitivo. Essa característica evita que planejamentos não racionais sobre a ordem de execução das tarefas sejam realizados. Tanto do ponto de vista do robô como do sistema, essa restrição no ordenamento de preferências resulta no atendimento à condição.

Domínio Irrestrito

A descrição da missão utilizando o ordenamento parcial limita o conjunto de ordenamentos viáveis no planejamento de cada robô. Uma camada centralizada, a camada de tarefas, envia a todos os robôs a mesma descrição da missão. Mesmo de forma parcial, essa descrição limita o conjunto de ordenamentos possíveis, além de impor certo ordenamento aos robôs.

Ainda, na arquitetura M+CTA, a apresentação das regras sociais também limita a possibilidade de ordenamento das tarefas. Dessa forma, a condição não é atendida.

Caso o ordenamento parcial imposto aos robôs seja modelado de forma a classificar os possíveis ordenamentos como ordenamentos de pico único, então essa restrição pode ser vista como uma relaxação da condição do Domínio Irrestrito. É possível, ainda, que a camada de tarefas modele os ordenamentos como Ordenamentos Fortes (Definição 2.7), em vez de ordenamentos parciais.

Independência das Tarefas Irrelevantes

Como a descrição da missão já traz os ordenamentos parciais, a execução das tarefas não consideradas nesse ordenamento tem a preferência indiferente. Porém, essa descrição torna a comparação de tarefas par a par muito dependente da relação de predecessora e sucessora. Assim, apesar de apresentar um ordenamento transitivo, há uma forte dependência entre as tarefas na arquitetura $M+$.

Independência dos Robôs Irrelevantes

Como o mecanismo de alocação envolve um leilão, existe uma comparação entre o custo (utilidade) calculado para a tarefa dentre os robôs que fazem a oferta. Dessa comparação, a alocação de tarefas poderá ter o resultado alterado, ou seja, o robô que primeiro manifestou sua preferência para a execução daquela tarefa não será o verdadeiro executor.

Do ponto de vista do leilão, o robô não tem sua preferência modificada dada uma oferta maior para a execução da tarefa. Entretanto, do ponto de vista do sistema, a utilização do Paradigma de Fusão de Planos altera, de forma centralizada, os planos individuais, buscando a resolução de conflitos no uso dos recursos. Isso faz com que as preferências individuais sejam alteradas um planejamento global. Caso o planejamento (preferências) do robô não seja compatível com o planejamento (preferência) global, então, um novo planejamento é necessário.

Princípio de Pareto

Quando um robô está no estado de Melhor-candidato de uma tarefa, essa é a solução Pareto-dominante para aquela tarefa. O robô comunica aos demais que aquela tarefa é sua tarefa de menor custo, portanto, maior preferência. Como o ambiente é dinâmico, no período de tempo entre entrar no estado Melhor-Candidato e iniciar efetivamente a execução da ação, outro robô pode se tornar o melhor candidato. Ou seja, outro robô pode manifestar sua preferência para a mesma tarefa. O algoritmo original da arquitetura compara os valores do custo para estabelecer um novo melhor candidato.

Como a agregação de preferências não permite a comparação de intensidade de preferência. Em ambos os casos, se há ou não troca do melhor candidato, a condição de otimalidade de Pareto é violada. Ambas as soluções admitem que para o aumento do bem-estar do robô ofertante, o bem-estar do robô melhor candidato deve diminuir. Portanto, usando apenas preferências ordinais, um mecanismo de desempate deve ser o responsável por encontrar a solução Pareto-dominante nesses casos. Quando não há oferta de outro robô, então, essa é a alocação Pareto-ótima da tarefa em questão.

Do ponto de vista do sistema, a solução final é a solução Pareto-ótima, pois após a negociação das tarefas entre os robôs (ainda usando o custo como desempate), a alocação é uma solução que maximiza o bem-estar social. Assim, a arquitetura falha no critério de Pareto na comparação local (entre robôs) do ordenamento de preferências.

Ditadura

O mecanismo de alocação não considera o planejamento de apenas um robô ou grupo de robôs para a alocação de tarefas. Com a negociação e o procedimento de solicitação de ajuda, em que a tarefa é realocada, o planejamento de todos os robôs influencia a alocação das tarefas. Por isso, não há um robô que imponha sua vontade sobre todos os demais. Entretanto, o sistema planejador centralizado faz o papel de ditador, à medida que o planejamento individual de um robô pode ser limitado ou modificado por essa camada.

Assim, apesar de ter a contribuição de todos os robôs, a camada de planejamento atua como uma impositora nos planejamentos individuais quando estes não convergem para o planejamento global.

Mecanismo de alocação

O mecanismo de alocação da arquitetura M+ é mais sofisticado que o leilão simples da arquitetura MURDOCH, pois o cálculo do custo leva em consideração um planejamento global. Vantagens e desvantagens práticas à parte, uma interpretação segundo a visão arroviana será proposta. Para essa análise, serão consideradas atendidos os axiomas da completude e da transitividade e a condição de Pareto. As demais condições serão consideradas relaxadas ou mesmo ignoradas.

Cada robô tem sua função de bem-estar social, cujas entradas são os ordenamentos parciais, o ordenamento de preferências (função de custo) e tem como saída um planejamento de execução das tarefas. Então, pode-se definir a função de bem-estar social local (do robô) como $\succ_{W_{R_i}}^{M^+}: \succeq_{R_i} \mapsto T_{R_i}$, sendo \succeq_{R_i} o conjunto de ordenamentos parciais do robô R_i e T_{R_i} o conjunto de tarefas ordenadas, o planejamento. Como já dito anteriormente, a função de escolha social é uma instância da função de bem-estar social, e a primeira é responsável pela oferta da tarefa de maior preferência, aquela que o robô elegeu para executar.

A função de bem-estar social global agrega as funções de bem-estar locais, ou seja, agrega os planejamentos (preferências) individuais. O objetivo da função é mapear as tarefas aos robôs de forma a minimizar o custo, que, conseqüentemente, aumente o bem-estar social. A FBS-MRT global ficaria $\succ_W^{M^+}: \succeq_R \mapsto T$, sendo \succeq_R o conjunto de planejamentos individuais.

De forma a sintetizar os conceitos discutidos nessa seção, as Tabelas 6, 7 e 8 apresentam os critérios e comentários para cada arquitetura, sendo a coluna da arquitetura MURDOCH preenchida de acordo com a adaptação da mesma.

Tabela 6 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte I

Critérios	Arquiteturas		
	ALLIANCE	MURDOCH	M+CTA
Axioma da Completude	A arquitetura atende ao axioma, pois o robô não evita a decisão e dispõe de comportamentos para a execução das tarefas, mesmo que comportamentos diferentes atendam à tarefas iguais. Quando uma tarefa não pode ser executada, então aquele comportamento terá uma motivação nula e entende-se que sua preferência é a menor.	Para o atendimento ao axioma, o robô não pode evitar a decisão, comunicando ao leiloeiro qual é o seu ordenamento de preferências. Para um ordenamento completo, os robôs devem individualmente comparar suas preferências entre os demais robôs levando em consideração seu desempenho.	A arquitetura atende ao axioma, pois o robô não evita a decisão. A partir do cálculo do custo de execução de cada tarefa, planejamento completo é proposto.
Axioma da Transitividade	A transitividade é afetada pela interdependência das alternativas, criando uma preferência condicional. Do ponto de vista da sociedade, a transitividade é mantida se considerado que a missão é finalizada, sendo as tarefas alocadas e executadas de forma a atender à ordem correta da missão. Entretanto, a transitividade individual não é sempre mantida.	O cálculo efetuado pelo robô para ordenar as preferências individuais deve se traduzir em um ordenamento transitivo. Isso quer dizer que espera-se da função de utilidade uma característica linear.	A condição é atendida devido à transitividade imposta pela camada de tarefas, resultando tanto em transitividade individual quanto da sociedade.
Domínio Irrestrito	A condição é relaxada por uma classificação implícita das preferências individuais que tende à classificação de preferências de pico único.	Para a arquitetura original, o domínio das alternativas é singular. A condição é atendida se, dadas todas as possibilidades de ordenamento dos robôs, ainda assim é possível emergir uma decisão social.	A condição não é atendida devido ao ordenamento parcial imposto pela camada de tarefas da arquitetura.

Tabela 7 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte II

Critérios	Arquiteturas		
	ALLIANCE	MURDOCH	M+CTA
Independência das Tarefas Irrelevantes	O robô ordena seus comportamentos internos independentemente. A ativação do comportamento em outro robô (a alocação de uma tarefa), retira a tarefa do conjunto de tarefas disponíveis. A falha de um robô, por exemplo, faz com que a tarefa retorne à esse conjunto. Portanto, a realocação considera as preferências de forma independente.	A cada tarefa, as preferências dos robôs poderão ser diferentes. Isso se deve às diferentes capacidades, recursos e níveis de desempenho dos robôs da sociedade.	Apesar de apresentar um ordenamento transitivo, há uma forte dependência entre as tarefas na arquitetura, devido, principalmente, à descrição da missão pela camada de tarefas conter ordenamentos parciais.
Independência dos Robôs Irrelevantes	Apesar de usar as informações dos demais robôs para calcular as motivações internas, as comparações par a par não são afetadas por um terceiro robô.	O ordenamento de preferências deve seguir a mesma ordem para os agentes disponíveis se uma tarefa é ofertada no início da missão ou se é ofertada durante a missão.	O planejador global altera o planejamento (preferências) individual do robô de acordo com a necessidade do plano (preferências) global.
Princípio de Pareto	A arquitetura é Pareto-eficiente, pois as alocações são sempre Pareto-dominantes, ou seja, não há outra alocação de tarefas possível a qual melhore a situação de um robô sem prejudicar a situação dos demais.	A eficiência <i>ex post</i> de Pareto-eficiência é mantida, como discutido na seção 4.3.1, por conta da utilização do Algoritmo 4.3.	Quando um robô está no estado de Melhor-candidato de uma tarefa, essa é a solução Pareto-dominante para aquela tarefa. Entretanto, caso outro robô faça uma oferta melhor e se torne o melhor candidato, o princípio é violado, mesmo que pontualmente. Do ponto de vista do sistema, a solução final é a solução Pareto-ótima, pois a alocação é uma solução que maximiza o bem-estar social.

Tabela 8 – Síntese dos critérios da visão Arroviana de problemas MRTA - Parte III

Critérios	Arquiteturas		
	ALLIANCE	MURDOCH	M+CTA
Ditadura	No pior caso, a arquitetura apresenta uma ditadura aleatória.	Utiliza uma ditadura aleatória de forma a eliminar as alternativas de menor preferência iterativamente.	O planejamento é coletivo, porém, a camada de planejamento pode impor determinadas restrições ao planejamento individual do robô.
Mecanismo de Alocação	Pode ser comparado ao RSD. Não compara a motivação (utilidade) dos robôs para uma mesma tarefa de forma cardinal, atribui as tarefas de forma aleatória e sequencial, atribui as tarefas de modo Pareto-eficiente, buscando maximizar o bem-estar social, e à prova de estratégias. A função de bem-estar social é $\gamma_W^{ALLIANCE}: [\gamma]_{\mathcal{R}} \mapsto U^{ \mathcal{R} }$.	O mecanismo de alocação é o Algoritmo 4.3. A partir das preferências dos robôs, o leiloeiro agrega as preferências de forma aleatória, elegendo um robô para a execução da tarefa. Assim, a função de bem-estar social é $\gamma_W^{MURDOCH}: [\gamma]_{\mathcal{R}} \mapsto \mathcal{R}$.	Há a comparação de utilidades cardinais (custo) para a alocação de tarefas. Com a comparação cardinal, muitos critérios da axiomática proposta não são atendidos e devem ser relaxados. Para o caso da comparação de preferências ordinais um mecanismo de desempate deve ser implementado. A função de bem-estar social global é $\gamma_W^{M+}: \gamma_R^{T_R} \mapsto T$.

6 Conclusão

6.1 Considerações Finais

Esta dissertação propôs uma estrutura arroviana, baseada na Teoria da Escolha Social, para a avaliação de mecanismos de alocação de tarefas em sistemas multirrobô. Além disso, é abordada a utilização de preferências ordinais individuais e mecanismos de agregação de preferências para a tomada de decisão social. Isso se deve à dificuldade/desafio da medição e da comparação de valores cardinais entre diferentes robôs.

Após uma breve apresentação da Teoria da Utilidade, o objetivo foi relacionar a medição cardinal da utilidade com o uso de preferências ordinais da Teoria da Escolha Social. O Teorema da Impossibilidade de Arrow, e por afinidade o Teorema de Gibbard-Satterthwaite, foram apresentados como balizadores para mecanismos de escolha social que, atendendo à determinadas condições razoáveis, quando analisadas individualmente, produzam decisões democráticas e Pareto-ótimas.

Uma revisão de SMR e do problema MRTA é apresentada no segundo capítulo. Uma classificação completa das arquiteturas e dos problemas de alocação de tarefas é exposta. As características dos termos são explanados e, ao final, é realizada a classificação das arquiteturas pesquisadas na literatura presentes no trabalho.

Assim como a escolha social pode ser aplicada à um conjunto de pessoas, assim também o é no caso de um conjunto de robôs. O Capítulo 4 apresenta a problemática da alocação de tarefas em sistemas multirrobô com uma visão arroviana. Para isso, são estabelecidas condições análogas às condições de Arrow para o domínio MRTA. Além disso, são discutidos mecanismos de escolha social para as classes de problemas MRTA que envolvem robôs capazes de realizar apenas uma tarefa (ST), de acordo com a taxonomia de problemas MRTA apresentada no Capítulo 3. Para as classes envolvendo robôs capazes de realizar múltiplas tarefas simultaneamente, são apresentadas as dificuldades e complexidades dos problemas e, por isso, justificado o não aprofundamento deste trabalho no tema.

A análise proposta permite avaliar soluções já propostas e implementadas sob a ótica arroviana. Como visto, considerações devem ser feitas e nem todas as arquiteturas são diretamente relacionadas com a estrutura proposta. Entretanto, novas soluções podem ser propostas.

O objetivo principal deste trabalho foi alcançado, e um paralelo entre a Teoria da Escolha Social e o problema de alocação de tarefas em sistemas multirrobô foi proposto. A estrutura proposta no Teorema da Impossibilidade de Arrow, e, por afinidade, no Teorema da Impossibilidade de Gibbard-Satterthwaite, foi relacionada com o problema MRTA.

Por fim, entende-se que o tema do trabalho é mais um assunto a ser desbravado para a aplicação em SMR e, mais especificamente, em problemas MRTA, podendo contribuir para o estudo teórico e prático de soluções para tal problema.

6.2 Trabalhos Futuros

Com a conclusão desta etapa, alguns trabalhos futuros podem ser propostos:

- Implementar os algoritmos de ditadura aleatória em série em um problema de alocação de tarefas e a comparação dos resultados com arquiteturas já consolidadas.
- Desenvolver uma arquitetura de alocação de tarefas baseada na Teoria da Escolha Social e baseada em preferências ordinais, avaliando os resultados quanto à qualidade e (possível) otimização da solução do problema MRTA.
- Comparar o RSD com outros mecanismos de escolha social, como o Serial Probabilístico (*Probabilistic Serial* - PS) de [Bogomolnaia e Moulin \(2001\)](#), para o problema de alocação de tarefas. O mecanismo PS dispõe de uma função de velocidade de deterioração, que é a velocidade com que o objeto é consumido. Essa função pode ser relacionada com o tempo de execução de uma tarefa. Essa proposição se justifica em estruturas em que as tarefas podem ser realocadas (por falhas ou por aumento no bem-estar social, por exemplo) e essa realocação não significa, necessariamente, que a tarefa volte à estaca zero, mas, sim, que o próximo robô retome de onde ela parou, a partir da função de velocidade de “deterioração” da tarefa.
- Definir uma metodologia de classificação das tarefas, em uma camada de planejamento da missão e decomposição de tarefas, por exemplo, de forma a possibilitar o ordenamento de preferências de pico-único.
- Estudar a aplicação de preferências dicotômicas ao problema MRTA, onde as tarefas também tem suas preferências, por exemplo, parâmetros pré-definidos e recursos necessários à execução, de forma a criar um ordenamento preferências sobre os robôs.
- Estudar a influência de tarefas dependentes ou altamente acopladas com a preferência dos agentes.
- Estudar sobre a utilização de listas de preferências incompletas e a influência dessa estrutura na eficiência das alocações.
- Para sistemas situados, verificar a escalabilidade dos algoritmos propostos na Seção 4.3.

APÊNDICE A – Prova do Teorema da Impossibilidade de Arrow

A demonstração apresentada neste trabalho é uma adaptação da apresentada em [Shoham e Leyton-Brown \(2008\)](#). Dividida em quatro passos, a demonstração assume que a FBS respeita a condição UD, é Pareto-ótima e satisfaz a condição IIA.

Prova (Shoham e Leyton-Brown (2008)). Assumindo que a função de bem-estar social W satisfaz UD, WP e IIA, demonstra-se que a mesma deve ser ditatorial.

Passo 1: *Se todo decisor expressa a preferência por uma alternativa x ou como a de maior preferência, ou como a de menor preferência, x deve ser ou como a de maior preferência, ou como a de menor preferência, de \succ_W também.*

Considerando um perfil de preferências arbitrário $[\succ]$, no qual cada decisor classifica $x \in X$ ou como a alternativa de maior preferência, ou como a de menor preferência, e considere, por contradição, que a afirmação anterior não seja verdadeira. Então, deve existir um par de alternativas distintas $x', x'' \in X$ para o qual $x' \succ_W x$ e $x \succ_W x''$, de forma que $\succ_W = x' \succ x \succ x''$.

Agora, assumindo que $[\succ]$ seja modificado de maneira que todos os decisores movam a alternativa x'' para uma posição logo acima da alternativa x' em sua classificação de preferências, deixando inalterada a relação entre os demais pares de alternativas, e seja chamado de $[\succ']$ (satisfazendo a condição UD). Sabe-se pela condição IIA que para $x' \succ_W x$ ou $x \succ_W x''$ mudar, a relação entre os pares x', x e/ou x, x'' deve mudar. Entretanto, uma vez que x ocupa uma posição extrema para todos os decisores, x'' pode ser movimentado para uma posição acima de x' sem modificar qualquer uma das relações entre os pares de alternativas.

Portanto, no perfil $[\succ']$ também é o caso em que $x' \succ_W x$ e $x \succ_W x''$. A partir desse fato e da característica transitiva da tomada de decisão, tem-se que $x' \succ_W x''$. Entretanto, todos os decisores em $[\succ']$ classificam x'' preferível à x' e a condição WP requer que $x'' \succ_W x'$. Há uma contradição.

Passo 2: *Existe algum decisor k^* que é extremamente principal (extremely pivotal) de forma que caso mude sua preferência em algum perfil, ele mude uma determinada alternativa x da última posição da classificação (ranking) social para a primeira posição.*

Considere um perfil de preferências $[\succ]$ no qual todos os decisores classificam x como a alternativa menos preferida, e no qual as preferências são arbitrárias. Pela condição WP, W deve, também, classificar x como a menos preferida.

Agora, os decisores de 1 a k modificam sucessivamente $[\succ]$ movendo x do final de suas classificações para o topo, preservando as demais posições relativas. Denote por k^* o primeiro decisor cuja mudança causou a mudança de x na classificação social \succ_W . Deve haver, claramente, um decisor como esse que, quando move x par ao topo da classificação, WP requer que x seja classificado no topo da preferência social.

Denotando $[\succ^1]$ como o perfil de preferências pouco antes de k^* mover x e $[\succ^2]$ o perfil de preferência logo após a movimentação de x , como mostrado na Figura A.1. Em $[\succ^1]$, x permanece na última classificação em \succ_W . Em $[\succ^2]$, x ocupa a posição no topo de \succ_W , e todos os decisores classificam x como a melhor (topo) ou a pior alternativa (última posição). Pelo argumento utilizado no Passo 1, em $[\succ^2]$, x deve ser classificado no topo de \succ_W .

Passo 3: O decisor k^* é um ditador sobre qualquer par x', x'' não envolvendo x .

Inicialmente, escolhe-se uma alternativa do par x', x'' . Sem perda de generalidade, considere x' escolhido. Um novo perfil de preferências $[\succ^3]$ será construído a partir de $[\succ^2]$. A primeira modificação a ser feita é mover x' para o topo da preferência de k^* , mantendo-se as demais inalteradas. Com isso, $x' \succ_{k^*} x \succ_{k^*} x''$. Após a modificação na classificação de k^* , a classificação relativa entre x' e x'' dos demais decisores, exceto k^* , é reorganizada de forma arbitrária, enquanto x permanece em sua posição extrema, no topo ou na última posição, de acordo com a Figura A.1.

Em $[\succ^1]$, tinha-se $x' \succ_W x$, já que x estava na última posição de \succ_W . Quando $[\succ^1]$ é comparada com $[\succ^3]$, a classificação relativa às alternativas x' e x são as mesmas para todos os agentes. Desse modo, de acordo com IIA, deve-se ter $x' \succ_W x$ em $[\succ^3]$ da mesma forma que em $[\succ^1]$. Em $[\succ^2]$, tinha-se $x \succ_E x''$, pois x estava no topo das preferências. A classificação relativa entre x e x'' é a mesma em $[\succ^2]$ e $[\succ^3]$. Portanto, em $[\succ^3]$, $x \succ_W x''$. Usando os dois fatos mencionados sobre $[\succ^3]$ e o princípio da transitividade, conclui-se que $x' \succ x''$.

Por fim, constrói-se um quarto perfil de preferências $[\succ^4]$ a partir de duas modificações de $[\succ^3]$. A primeira é, arbitrariamente, mudar a posição de x no ordenamento de cada agente decisor, enquanto mantêm-se as demais classificações relativas as mesmas. A segunda modificação inclui mover x' para uma posição arbitrária no ordenamento de preferência do decisor k^* , com a única limitação de que x' continue sendo preferida a x'' . Observa-se, na Figura A.1, que todos os agentes exceto k^* têm preferências totalmente arbitrárias em $[\succ^4]$, enquanto as preferências de k^* são arbitrárias exceto por $x' \succ_{k^*} x''$. Em $[\succ^3]$ e $[\succ^4]$, todos os agentes tem as mesmas preferências relativas entre x' e x'' , portanto, da mesma forma que $x' \succ_W x''$ em $[\succ^3]$ e satisfazendo a condição IIA, $x' \succ_W x''$ em $[\succ^4]$.

Assim, a preferência social foi determinada entre x' e x'' assumindo nada além de que $x' \succ_{k^*} x''$.

Passo 4: O decisor k^* é um ditador sobre todos os pares x', x .

Considere uma terceira alternativa x'' . Pelo argumento utilizado no Passo 2, há um decisor k^{**} que é *extremamente principal* para x'' . E pelo argumento do Passo 3, k^{**} é um ditador para qualquer par x, x' que não envolve a alternativa x'' .

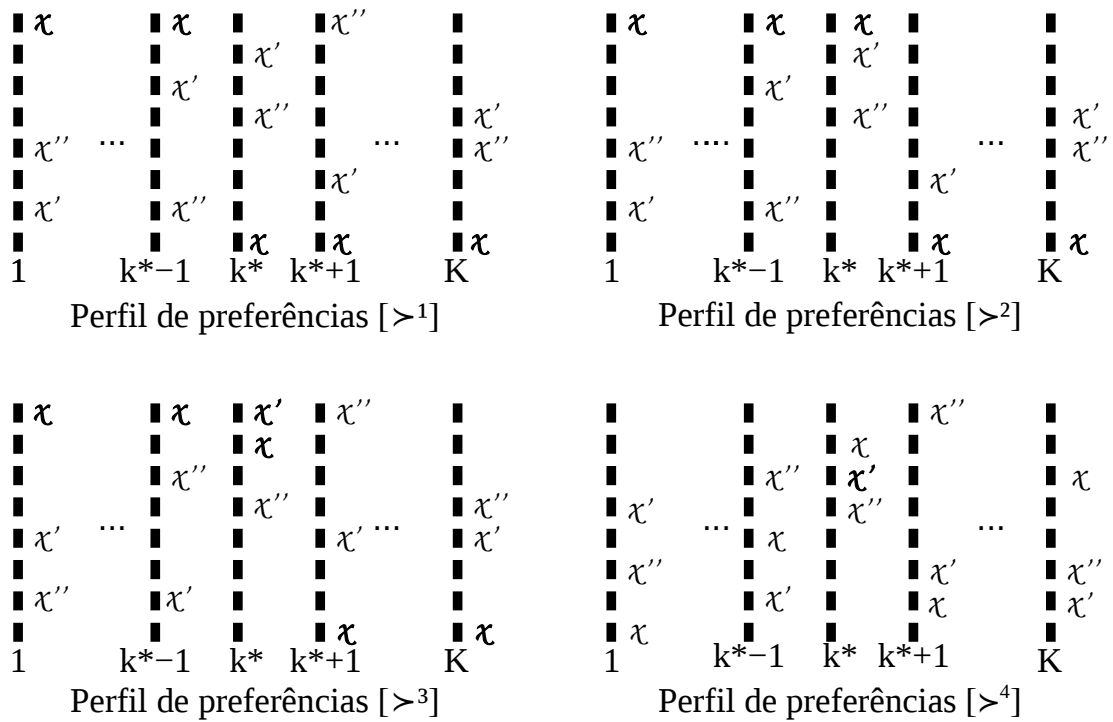


Figura A.1 – Os quatro perfis de preferências usados na prova do Teorema de Arrow. O ordenamento segue a linha pontilhada, sendo a posição superior da linha preferível à posição imediatamente abaixo. As alternativas em negrito são avaliadas durante os passos da prova, as demais exemplificam o que poderia ocorrer. Fonte: Adaptado de [Shoham e Leyton-Brown \(2008\)](#).

Observa-se que k^* é capaz de afetar a classificação de x, x' em W , como a mudança na classificação de x efetuada no Passo 2. Consequentemente, k^{**} e k^* são o mesmo decisor. Mostrou-se, assim, que o decisor k^* é um ditador sobre todos os pares de alternativas. \square

Referências

- ABDULKADIROĞLU, A.; SÖNMEZ, T. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, JSTOR, v. 66, n. 3, p. 689–701, 1998. [101](#)
- ALAMI, R. et al. Multi-robot cooperation in the martha project. *Robotics & Automation Magazine, IEEE*, IEEE, v. 5, n. 1, p. 36–47, 1998. [71](#)
- ALLARDT, E. To have, to love, to be—about welfare in the nordic countries. *Argos, Lund (Swedish)*, 1975. [87](#)
- ARROW, K. J. A difficulty in the concept of social welfare. *The Journal of Political Economy*, JSTOR, p. 328–346, 1950. [27](#), [89](#)
- ARROW, K. J. *Social choice and individual values*. New York: Yale university press, 1951. [18](#), [22](#), [27](#), [28](#), [29](#), [30](#), [31](#), [33](#), [34](#), [35](#), [40](#), [41](#), [43](#), [44](#), [56](#), [92](#), [93](#), [96](#), [97](#)
- ARROW, K. J.; SEN, A.; SUZUMURA, K. *Handbook of social choice and welfare*. Amsterdam: Elsevier, 2011. [18](#), [87](#)
- AZIZ, H. Computational social choice: some current and new directions. In: AAAI PRESS. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. New York, 2016. p. 4054–4057. [42](#)
- AZIZ, H.; MESTRE, J. Parametrized algorithms for random serial dictatorship. *Mathematical Social Sciences*, Elsevier, v. 72, p. 1–6, 2014. [102](#), [103](#)
- AZIZ, H.; WALSH, T.; XIA, L. Possible and necessary allocations via sequential mechanisms. In: AAAI PRESS. *Proceedings of the 24th International Conference on Artificial Intelligence*. Buenos Aires, 2015. p. 468–474. [105](#)
- BADE, S. Random serial dictatorship: the one and only. *Royal Holloway College, Unpublished mimeo*, 2014. [101](#), [102](#)
- BANDYOPADHYAY, T. Revealed preference theory, ordering and the axiom of sequential path independence. *The Review of Economic Studies*, Oxford University Press, v. 55, n. 2, p. 343–351, 1988. [97](#)
- BANDYOPADHYAY, T. Sequential path independence and social choice. *Social Choice and Welfare*, Springer, v. 7, n. 3, p. 209–220, 1990. [97](#)
- BARBERÀ, S.; HAMMOND, P.; SEIDL, C. *Handbook of Utility Theory: Volume 1 Principles*. Boston: Springer Science & Business Media, 2004. [21](#), [22](#)
- BARBERÀ, S.; HAMMOND, P.; SEIDL, C. *Handbook of Utility Theory: Volume 2 Extensions*. Boston: Springer Science & Business Media, 2004. [30](#)

- BASTOS, G. S.; RIBEIRO, C. H. C.; SOUZA, L. E. d. Variable utility in multi-robot task allocation systems. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2008 IEEE Latin American Robotic Symposium*. Salvador, 2008. p. 179–183. [51](#), [80](#)
- BELTRAME, B. *O debate de Amartya Sen com Kenneth Arrow e John Rawls e a Abordagem das Capacidades*. 77 p. Dissertação (Mestrado) — Mestrado em Economia Política, Pontifícia Universidade Católica de São Paulo, 2009. [20](#)
- BLACK, D. On the rationale of group decision-making. *Journal of political economy*, The University of Chicago Press, v. 56, n. 1, p. 23–34, 1948. [34](#), [35](#)
- BOGOMOLNAIA, A.; MOULIN, H. A new solution to the random assignment problem. *Journal of Economic Theory*, v. 100, p. 295–328, 2001. [101](#), [143](#)
- BORDA, J.-C. de. On elections by ballot. *Classics of social choice, eds. I. McLean, AB Urken, and F. Hewitt - 1995*, p. 83–89, 1781. [26](#)
- BOTELHO, S.; ALAMI, R. Robots that cooperatively enhance their plans. In: *In Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*. Knoxville: Springer, 2000. [19](#), [70](#), [73](#), [74](#), [75](#)
- BOTELHO, S.; ALAMI, R. Multi-robot cooperation through the common use of “mechanisms”. In: IEEE. *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. Maui, 2001. v. 1, p. 375–380. [73](#), [74](#), [75](#)
- BOTELHO, S. C.; ALAMI, R. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: IEEE. *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. Detroit, 1999. v. 2, p. 1234–1239. [17](#), [19](#), [66](#), [70](#), [71](#), [72](#), [73](#), [83](#), [135](#)
- BOUVERET, S.; ENDRISS, U.; LANG, J. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In: IOS PRESS. *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. Düsseldorf, 2010. p. 387–392. [105](#)
- BRAMS, S. J.; EDELMAN, P. H.; FISHBURN, P. C. Fair division of indivisible items. *Theory and Decision*, Springer, v. 55, n. 2, p. 147–180, 2003. [106](#), [107](#)
- BRANDT, F. et al. *Handbook of Computational Social Choice*. Nova York: Cambridge University Press, 2016. [17](#), [33](#), [42](#), [43](#), [44](#), [101](#), [105](#), [109](#), [111](#)
- CAMPBELL, D. E.; KELLY, J. S. t or 1-t. that is the trade-off. *Econometrica: Journal of the Econometric Society*, JSTOR, p. 1355–1365, 1993. [36](#)
- CAMPBELL, D. E.; KELLY, J. S. Impossibility theorems in the arroviaan framework. *Handbook of social choice and welfare*, Elsevier, v. 1, p. 35–94, 2002. [35](#), [36](#)
- CAO, Y. U. et al. Cooperative mobile robotics: Antecedents and directions. In: IEEE. *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots'*, *Proceedings. 1995 IEEE/RSJ International Conference on*. Pittsburgh, 1995. v. 1, p. 226–234. [17](#), [45](#), [46](#), [48](#), [49](#)
- CASTRO, L.; FARO, J. Introdução à teoria da escolha. *25º Colóquio Brasileiro de Matemática*, IMPA, 2005. Disponível em: <www.impa.br>. [24](#), [25](#), [27](#), [28](#), [32](#), [33](#)

- CATO, S. Independence of irrelevant alternatives revisited. *Theory and decision*, Springer, v. 76, n. 4, p. 511–527, 2014. [30](#)
- CHAIMOWICZ, L.; CAMPOS, M. F.; KUMAR, V. Dynamic role assignment for cooperative robots. In: IEEE. *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*. Washington D.C., 2002. v. 1, p. 293–298. [66](#), [78](#), [84](#)
- CHEVALEYRE, Y. et al. Issues in multiagent resource allocation. *Informatica (Slovenia)*, v. 30, n. 1, p. 3–31, 2006. [41](#), [43](#)
- CHEVALEYRE, Y. et al. A short introduction to computational social choice. In: SPRINGER-VERLAG. *Proceedings of the 33rd conference on Current Trends in Theory and Practice of Computer Science*. Harrachov, 2007. p. 51–69. [42](#)
- CHOUDHURY, B. B. *Task Allocation Strategies in Multi-Robot Environment*. 2009. 236 p. Tese (Doutorado) — Departamento de Engenharia Mecânica, National Institute of Technology Rourkela, Orissa, Índia, 2009. [17](#), [54](#), [99](#)
- CRAVEN, J. *Social choice: A framework for collective decisions and individual judgements*. Cambridge: Cambridge University Press, 1992. [30](#), [32](#), [33](#)
- CUSINATO, R. T. *Teoria da Decisão sob Incerteza e a Hipótese da Utilidade Esperada: Conceitos analíticos e paradoxos*. 181 p. Dissertação (Mestrado) — Mestrado em Economia, Pós-Graduação em Economia, Faculdade de Ciências Econômicas, Universidade Federal do Rio Grande do Sul, 2003. [21](#), [22](#), [23](#), [24](#), [25](#)
- DIAS, M. B. Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments. *Robotics Institute*, p. 153, 2004. [78](#), [79](#)
- DIAS, M. B. et al. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, IEEE, v. 94, n. 7, p. 1257–1270, 2006. [65](#), [66](#)
- DILGUERIAN, A. M.; WEISZFLOG, W. *Michaelis Moderno Dicionário da Língua Portuguesa*. São Paulo: Editora Melhoramentos LTDA, 1998. [88](#)
- DRYZEK, J. S.; LIST, C. Social choice theory and deliberative democracy: a response to aldred. *British Journal of Political Science*, Cambridge University Press, v. 34, n. 4, p. 752–758, 2004. [36](#)
- DUDEK, G. et al. A taxonomy for multi-agent robotics. *Autonomous Robots*, Springer, v. 3, n. 4, p. 375–397, 1996. [46](#)
- ENDRISS, U. Social choice theory as a foundation for multiagent systems. In: SPRINGER. *Multiagent System Technologies: 12th German Conference, MATES 2014, Proceedings*. Stuttgart, 2014. v. 8732, p. 1. [42](#)
- ENDRISS, U.; MAUDET, N. Welfare engineering in multiagent systems. In: SPRINGER. *International Workshop on Engineering Societies in the Agents World*. London, 2003. p. 93–106. [41](#)
- FALIK, D.; FRIEDGUT, E. An algebraic proof of a robust social choice impossibility theorem. In: IEEE. *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*. Palm Springs, 2011. p. 413–422. [33](#)

- FARINELLI, A.; IOCCHI, L.; NARDI, D. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 34, n. 5, p. 2015–2028, 2004. [19](#), [45](#), [46](#), [47](#), [48](#), [50](#), [51](#), [83](#)
- FLEURBAEY, M.; MANIQUET, F. *A theory of fairness and social welfare*. Cambridge: Cambridge University Press, 2011. [89](#)
- GAERTNER, W. *A primer in social choice theory: Revised edition*. Oxford: Oxford University Press, 2009. [25](#), [26](#), [27](#), [28](#), [30](#), [32](#), [33](#), [38](#)
- GAGE, A. Multi-robot task allocation using affect. 2004. [78](#)
- GALE, D.; SHAPLEY, L. S. College admissions and the stability of marriage. *The American Mathematical Monthly*, JSTOR, v. 69, n. 1, p. 9–15, 1962. [109](#), [111](#)
- GEANAKOPOLOS, J. Three brief proofs of arrow's impossibility theorem. *Economic Theory*, Springer, v. 26, n. 1, p. 211–215, 2005. [33](#)
- GERKEY, B.; MATARIĆ, M. J. Are (explicit) multi-robot coordination and multi-agent coordination really so different. In: *Proceedings of the AAAI spring symposium on bridging the multi-agent and multi-robotic research gap*. Stanford: [s.n.], 2004. p. 1–3. [18](#), [51](#), [55](#)
- GERKEY, B. P. *On multi-robot task allocation*. 2003. 117 p. Tese (Doutorado) — Pós-graduação em Ciência da Computação, University of Southern California, Los Angeles, 2003. [19](#), [55](#), [67](#), [68](#), [69](#), [70](#), [99](#), [103](#), [105](#), [108](#), [113](#), [114](#)
- GERKEY, B. P.; MATARIĆ, M. J. Sold!: Auction methods for multirobot coordination. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 18, n. 5, p. 758–768, 2002. [17](#), [19](#), [66](#), [67](#), [68](#), [69](#), [70](#), [80](#), [83](#), [134](#)
- GERKEY, B. P.; MATARIĆ, M. J. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, SAGE Publications, v. 23, n. 9, p. 939–954, 2004. [17](#), [19](#), [45](#), [47](#), [48](#), [51](#), [52](#), [53](#), [54](#), [65](#), [69](#), [79](#), [82](#), [83](#), [87](#), [96](#), [98](#), [100](#), [108](#), [113](#), [115](#)
- GIBBARD, A. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, JSTOR, p. 587–601, 1973. [38](#), [92](#)
- GREVE, B. What is welfare? *Central European Journal of Public Policy*, v. 2, n. 1, p. 50–73, 2008. [87](#), [88](#)
- HANSSON, S. O. Decision theory: A brief introduction. *Department of Philosophy and the History of Technology Royal Institute of Technology (KTH) Stockholm*, 2005. [20](#)
- HICKS, J. R. A revision of demand theory. 1956. [21](#)
- HUNT, E. K.; LAUTZENHEISER, M. *História do Pensamento Econômico: uma perspectiva crítica*. Tradução de José Ricardo Brandão Azevedo e Maria José Cyhlar Monteiro. Rio de Janeiro: Elsevier, 2005. [20](#), [21](#)
- HURWICZ, L. The design of mechanisms for resource allocation. *The American Economic Review*, JSTOR, v. 63, n. 2, p. 1–30, 1973. [43](#)

- IOCCHI, L.; NARDI, D.; SALERNO, M. Reactivity and deliberation: a survey on multi-robot systems. In: SPRINGER. *Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Berlin, 2000. p. 9–32. [19](#), [45](#), [46](#), [50](#), [51](#)
- JACOBS, J. F.; POEL, I. van de; OSSEWEIJER, P. Clarifying the debate on selection methods for engineering: Arrow’s impossibility theorem, design performances, and information basis. *Research in Engineering Design*, Springer, v. 25, n. 1, p. 3–10, 2014. [41](#)
- KALINOWSKI, T.; NARODYTSKA, N.; WALSH, T. A social welfare optimal sequential allocation procedure. In: AAAI PRESS. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. Beijing, 2013. p. 227–233. [106](#), [107](#)
- KALRA, N. et al. *Market-based multirobot coordination: A comprehensive survey and analysis*. Pittsburgh, 2005. [66](#)
- KATO, M.; OHSETO, S. Toward general impossibility theorems in pure exchange economies. *Social Choice and Welfare*, Springer, v. 19, n. 3, p. 659–664, 2002. [102](#)
- KIM, M.-H.; BAIK, H.; LEE, S. Resource welfare based task allocation for uav team with resource constraints. *Journal of Intelligent & Robotic Systems*, Springer, v. 77, n. 3-4, p. 611–627, 2015. [41](#)
- KIM, M.-H.; KIM, S.-P.; LEE, S. Social-welfare based task allocation for multi-robot systems with resource constraints. *Computers & Industrial Engineering*, Elsevier, v. 63, n. 4, p. 994–1002, 2012. [41](#)
- KLAUS, B.; BOCHET, O. The relation between monotonicity and strategy-proofness. *Social Choice and Welfare*, Springer, v. 40, n. 1, p. 41–63, 2013. [96](#)
- KLEINBERG, J. M. An impossibility theorem for clustering. In: NIPS. *Advances in neural information processing systems*. Vancouver, 2003. p. 463–470. [40](#)
- KORSAH, G. A.; STENTZ, A.; DIAS, M. B. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, SAGE Publications, v. 32, n. 12, p. 1495–1512, 2013. [45](#), [52](#), [53](#), [115](#)
- LIST, C. Social choice theory. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Winter 2013. [S.l.]: Metaphysics Research Lab, Stanford University, 2013. [27](#), [34](#), [35](#), [36](#), [37](#), [38](#)
- LIU, C. *Multi-Robot Task Allocation for Inspection Problems with Cooperative Tasks Using Hybrid Genetic Algorithms*. Tese (Doutorado) — Kassel Univ. Press, 2014. [66](#), [83](#)
- MASKIN, E.; SEN, A. *The Arrow Impossibility Theorem*. Nova York: Columbia University Press, 2014. (Kenneth J. Arrow Lecture series). Com Kenneth J. Arrow, Partha Dasgupta, Prasant K. Pattanaik e Joseph E. Stiglitz. [28](#), [29](#), [32](#), [33](#)
- MATARIĆ, M. J. Introdução à robótica. Ed Unesp, 2014. [48](#), [50](#), [51](#)
- MCLEAN, I. a. The borda and condorcet principles: three medieval applications. *Social Choice and Welfare*, Springer, v. 7, n. 2, p. 99–108, 1990. [25](#)
- MEYER, T.; GHOSE, A.; CHOPRA, S. Multi-agent context-based merging. In: CITeseer. *Proceedings of Common Sense*. [S.l.], 2001. [41](#)

- MORREAU, M. Arrow's theorem. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Winter 2014. [S.l.: s.n.], 2014. 28, 32, 33
- MOSTEO, A. R. *Multi-robot task allocation for service robotics: From unlimited to limited communication range*. Tese (Doutorado) — Citeseer, 2010. 76, 77
- MOULIN, H. *Axioms of cooperative decision making*. [S.l.]: Cambridge University Press, 1991. 27, 28
- NATH, S. K. *A perspective of welfare economics*. London: Macmillan Press LTD, 1973. 87
- NEUMANN, J. V.; MORGENSTERN, O. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press, 1953. 22
- NORVIG, P.; RUSSELL, S. *Inteligência Artificial, 3ª Ed*. Rio de Janeiro: Elsevier, 2013. 22, 23, 24
- PARKER, L. E. *Heterogeneous Multi-Robot Cooperation*. Tese (Doutorado) — Massachusetts Institute of Technology, 1994. 19, 57, 58, 59, 60, 61, 62, 63, 64, 65, 99, 129, 132
- PARKER, L. E. On the design of behavior-based multi-robot teams. *Advanced Robotics*, Taylor & Francis, v. 10, n. 6, p. 547–578, 1995. 49, 56, 57, 80
- PARKER, L. E. L-ALLIANCE: Task-oriented multi-robot learning in behavior-based systems. *Advanced Robotics*, Taylor & Francis, v. 11, n. 4, p. 305–322, 1996. 17, 19
- PARKER, L. E. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 14, n. 2, p. 220–240, 1998. 47, 48, 56, 59, 62, 64, 80, 81, 118, 121, 122, 123, 124
- PARKER, L. E. Multiple mobile robot systems. In: *Springer Handbook of Robotics*. [S.l.]: Springer, 2008. p. 921–941. 47, 49, 68, 83
- PENNOCK, D. M. et al. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In: *AAAI/IAAI*. [S.l.: s.n.], 2000. p. 729–734. 40
- PRAAG, B. M. V.; FRIJTERS, P. The measurement of welfare and well-being: The leyden approach. *Well-being: Foundations of hedonic psychology*, Russell Sage Foundation, p. 413, 1999. 88
- READ, D. Utility theory from jeremy bentham to daniel kahneman. Department of Operational Research, London School of Economics and Political Science, 2004. 21, 22
- REIS, W. P. N. dos; BASTOS, G. S. Multi-robot task allocation approach using ros. In: *IEEE. Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), 2015 12th Latin American*. [S.l.], 2015. p. 163–168. 118, 123
- REIS, W. P. N. dos; BASTOS, G. S. Implementing and simulating an alliance-based multi-robot task allocation architecture using ros. In: *SPRINGER INTERNATIONAL PUBLISHING. Latin American Robotics Symposium*. [S.l.], 2016. p. 210–227. 9, 118, 119, 123, 125, 126, 130
- REIS, W. P. N. dos; BASTOS, G. S. An arrowian view on the multi-robot task allocation problem. In: *IEEE. Advanced Robotics (ICAR), 2017 18th International Conference on*. [S.l.], 2017. p. 290–295. 98, 113

- RENY, P. J. Arrow's theorem and the gibbard-satterthwaite theorem: a unified approach. *Economics Letters*, Elsevier, v. 70, n. 1, p. 99–105, 2001. [38](#), [39](#)
- ROHRMÜLLER, F. *Action Selection in Cooperative Multi-Robot Systems*. 2011. 148 p. Tese (Doutorado) — Faculdade de Engenharia Elétrica e Tecnologia da Informação, Technischen Universität München, Munique, 2011. [54](#)
- ROOS, M.; ROTHE, J. Complexity of social welfare optimization in multiagent resource allocation. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. [S.l.], 2010. p. 641–648. [41](#)
- SAARI, D. G. From arrow's theorem to 'dark matter'. *British Journal of Political Science*, Cambridge Univ Press, v. 46, n. 01, p. 1–9, 2016. [27](#)
- SATTERTHWAITE, M. A. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of economic theory*, Elsevier, v. 10, n. 2, p. 187–217, 1975. [38](#), [92](#)
- SATTERTHWAITE, M. A.; SONNENSCHNEIN, H. Strategy-proof allocation mechanisms at differentiable points. *The Review of Economic Studies*, JSTOR, v. 48, n. 4, p. 587–597, 1981. [102](#)
- SEN, A. The impossibility of a paretian liberal. *Journal of political economy*, The University of Chicago Press, v. 78, n. 1, p. 152–157, 1970. [36](#)
- SHIROMA, P. M.; CAMPOS, M. F. Comutar: A framework for multi-robot coordination and task allocation. In: IEEE. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. [S.l.], 2009. p. 4817–4824. [115](#)
- SHOHAM, Y.; LEYTON-BROWN, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. [S.l.]: Cambridge University Press, 2008. [22](#), [23](#), [24](#), [25](#), [27](#), [28](#), [32](#), [33](#), [38](#), [47](#), [95](#), [144](#), [146](#)
- SIMON, H. A. A behavioral model of rational choice. *The quarterly journal of economics*, MIT Press, v. 69, n. 1, p. 99–118, 1955. [40](#)
- SMITH, R. G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, IEEE, n. 12, p. 1104–1113, 1980. [66](#), [67](#)
- STENTZ, A.; DIAS, M. B. *A free market architecture for coordinating multiple robots*. [S.l.], 1999. [78](#)
- STIRLING, W. C.; NOKLEBY, M. S. Satisficing coordination and social welfare for robotic societies. *International Journal of Social Robotics*, Springer, v. 1, n. 1, p. 53–69, 2009. [17](#), [40](#), [41](#)
- TANG, F.; PARKER, L. E. Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration. In: IEEE. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. [S.l.], 2005. p. 1501–1508. [77](#), [108](#)
- VIG, L.; ADAMS, J. A. Market-based multi-robot coalition formation. *Distributed Autonomous Robotic Systems 7*, Springer, p. 227–236, 2006. [108](#)

- VU, T. et al. *Monad: A flexible architecture for multi-agent control*. In: *ACM. Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. [S.l.], 2003. p. 449–456. [76](#), [77](#)
- WAGNER, S. A. L. *Novas e Velhas Abordagens da Teoria da Escolha e da Utilidade*. 179 p. Dissertação (Mestrado) — Mestrado em Economia Política, Pontifícia Universidade Católica de São Paulo, 2009. [20](#), [24](#)
- WERGER, B. B. *Ayllu: Distributed port-arbitrated behavior-based control*. In: *Distributed autonomous robotic systems 4*. [S.l.]: Springer, 2000. p. 25–34. [75](#)
- WERGER, B. B.; MATARIĆ, M. J. *Broadcast of local eligibility for multi-target observation*. In: *Distributed autonomous robotic systems 4*. [S.l.]: Springer, 2000. p. 347–356. [56](#), [75](#), [76](#)
- YAN, Z.; JOUANDEAU, N.; CHERIF, A. A. *A survey and analysis of multi-robot coordination*. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 10, n. 12, p. 399, 2013. [47](#)
- YUAN, Z. *A new explanation of kj arrow's impossibility theorem: On conditions of social welfare functions*. *Open Journal of Political Science*, Scientific Research Publishing, v. 5, n. 01, p. 26, 2015. [30](#), [33](#), [99](#)
- ZHANG, Y. *Coalition Formation and Execution in Multi-robot Tasks*. 173 p. Tese (Doutorado) — Universidade do Tennessee, 2012. [77](#), [78](#), [79](#)