



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

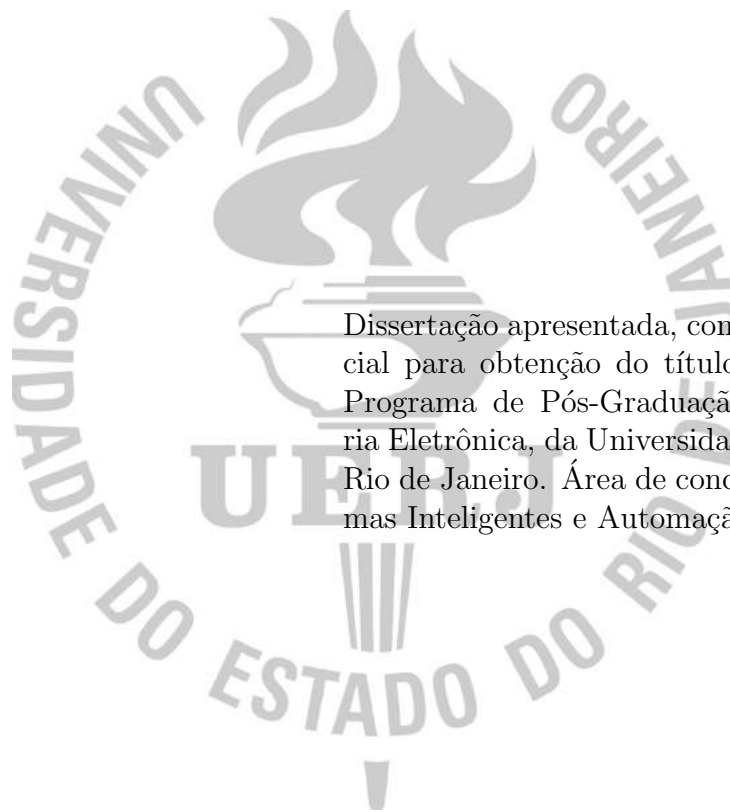
Luigi Maciel Ribeiro

Otimização da comunicação na alocação dinâmica
de tarefas em enxame de robôs

Rio de Janeiro
2018

Luigi Maciel Ribeiro

Otimização da comunicação na alocação dinâmica de tarefas em enxame de robôs



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof^a. Dr^a. Nadia Nedjah

Coorientadora: Prof^a. Dr^a. Luiza de Macedo Mourelle

Rio de Janeiro
2018

CATALOGAÇÃO NA FONTE
UERJ/REDE SIRIUS/CTC/B

M539 Ribeiro, Luigi Maciel

Otimização da comunicação na alocação dinâmica de tarefas em enxame de robôs / Luigi Maciel Ribeiro. – 2018.

115f.

Orientadora: Nadia Nedjah.

Coorientadora: Luiza de Macedo Mourelle.

Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica. 2. Robótica – Dissertações. 3. Algoritmos de computador – Dissertações. I. Nedjah, Nadia. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro. IV. Título.

CDU 007.52

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Luigi Maciel Ribeiro

Otimização da comunicação na alocação dinâmica de tarefas em enxame de robôs

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 30 de maio de 2018

Banca Examinadora:

Prof^a. Dr^a. Nadia Nedjah (Orientadora)
Faculdade de Engenharia, UERJ

Prof^a. Dr^a. Luiza de Macedo Mourelle (Coorientadora)
Faculdade de Engenharia, UERJ

Prof. Dr. Roberto Pinheiro Domingos
Instituto Politécnico do Rio de Janeiro, UERJ

Prof. Dr. Helon Vicente Hultmany Ayala
Pontifícia Universidade Católica do Rio de Janeiro, PUC-RIO

Rio de Janeiro
2018

O maior inimigo do conhecimento não é a ignorância,
é a ilusão do conhecimento.

Stephen Hawking

AGRADECIMENTOS

Gostaria de agradecer primeiramente à Deus, por todas as bênçãos e proteção que recebi durante minha caminhada até aqui e pela oportunidade de conviver com pessoas incríveis ao meu redor.

Agradecimentos especiais às minhas orientadoras, Professora Nadia Nedjah e Professora Luiza M. Mourelle. Obrigado por acreditarem no meu potencial e nunca perderem a fé nas minhas capacidades. Obrigado também pelas broncas e puxões de orelha nos momentos certos, os quais me ajudaram a concluir esta dissertação. E obrigado, principalmente, por me guiarem por todo esse caminho, cheio de incertezas e dificuldades.

Agradeço aos professores que tive até aqui que, com dedicação e carinho, compartilharam seus conhecimentos comigo e sem os quais eu não seria o profissional que sou hoje.

Agradeço aos meus amigos da faculdade, pois sozinho nada conquistaria, unidos tudo se tornou mais fácil.

Agradeço especialmente aos meus pais Edson Soligo Ribeiro e Maria Eugênia Maciel Ribeiro, pelo apoio incondicional durante toda a minha vida, fundamental para o meu sucesso até aqui. Agradeço por todo esse amor incomensurável, que moldou a minha personalidade e o meu caráter.

Agradeço a minha irmã Renata, por ser um dos meus poucos amigos verdadeiros. Obrigado por todos os sonhos compartilhados e as conversas sem sentido.

Agradeço ao Francis, meu filhotinho de quatro patas, por todo o seu amor puro e incondicional, que afasta qualquer resquício de tristeza dos meus dias.

Agradeço à minha esposa Shimene Reis Graça Ribeiro, por dar sentido a minha vida. Sem ela, este trabalho não teria sentido. Obrigado por me motivar a ser sempre mais, mais educado, mais humildade, mais bondoso, mais apaixonado.

RESUMO

RIBEIRO, Luigi Maciel *Otimização da comunicação na alocação dinâmica de tarefas em enxame de robôs*. 2018. 115f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

O interesse da comunidade acadêmica, em solucionar problemas de alta complexidade, vem crescendo nos últimos anos, intensificando a utilização da Inteligência de Enxame. A inteligência de Enxame possui inspiração biológica, proposta a partir da observação social de espécies auto-organizáveis, como formigas, abelhas, cupins, entre outros. A cooperação é a ideia central, pois possibilita a solução de problemas complexos com a realização coordenada de pequenas tarefas, que somadas levam a um objetivo comum. Essa coordenação só é possível com uma Alocação de Tarefas eficiente. A alocação deverá ser dinâmica, pois deve reagir às alterações do problema e do enxame, e também possuir uma solução distribuída e estocástica, respeitando o comportamento coletivo biológico que inspirou a Inteligência de Enxame. Otimização por enxame de partículas (*Particle swarm optimization - PSO*) é um algoritmo de otimização que atende à esses requisitos, gerenciando um grupo de partículas que navegam em um espaço de busca limitado, onde a posição atual da partícula é uma possível solução. A posição de cada partícula é adaptativa em função da sua própria experiência e da experiência das demais. Com inspiração no PSO, o algoritmo Alocação Dinâmica de Tarefas em Cluster (ADTC) foi proposto. O algoritmo ADTC realiza a alocação de tarefas para um grupo de robôs de forma totalmente distribuída, onde cada um dos robôs representará uma partícula e a sua posição no espaço de busca representará uma alocação factível. Baseado no PSO, o ADTC realiza uma busca orientada do espaço, utilizando o mesmo conceito de velocidade adaptativa. Porém, esse processo exige uma intensa troca de informações entre os robôs, podendo ser um empecilho para grandes enxames. Nesta dissertação é proposta a utilização da topologia de comunicação em cluster, capaz de otimizar processos de comunicação, viabilizando a alocação de tarefas de grandes enxames. Os resultados obtidos com a topologia em cluster são comparados com os obtidos com a topologia em malha completa, mostrando o impacto da otimização da comunicação no desempenho da alocação dinâmica de tarefas. Na média, os resultados apresentam uma otimização de 30%.

Palavras-chave: Alocação dinâmica de tarefas; Robótica de enxame; Computação distribuída; Inteligência de enxame; Topologia de comunicação; Cluster.

ABSTRACT

RIBEIRO, Luigi Maciel *Optimization of communication in the dynamic task allocation using swarm of robots*. 2018. 115f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

The interest of the academic community in solving highly complex problems has been growing in recent years, putting an emphasis on the use of Swarm Intelligence. The Swarm Intelligence has a biological inspiration, proposed from the social observation of self-organizing species such as ants, bees and termites. Cooperation is the central idea, it allows the solution of complex problems with the coordinated accomplishment of small tasks, which together lead to a common goal. This coordination is only possible with an efficient Task Allocation. The allocation should be dynamic, as it must react to the problem and swarm changes, and also to have a distributed and stochastic solution, respecting the biological collective behavior that inspired the Swarm Intelligence. Particle swarm optimization (PSO) is an optimization algorithm that meets these requirements by managing a set of particles that navigate in a limited search space where the particle's current position is a possible solution. The position of each particle is adaptive in function of its own experience and the experience of the others. Inspired by the PSO, the *Alocação Dinâmica de Tarefas em Cluster* (ADTC) algorithm was proposed. The algorithm ADTC assigns tasks to a group of robots in a totally distributed way, where each one will represent a particle and its position of the search space will represent a feasible allocation. Based on the PSO, the ADTC performs an oriented search of the space, using the same concept of adaptive speed. However, this process requires an intense exchange of information between robots, and can hinder the task allocation for large swarms. In this dissertation, we propose the use of the cluster communication topology, capable of optimizing the communication process between robots, making the allocation of tasks of large swarms. The results obtained using the cluster topology are compared to those obtained with the complete mesh topology, showing the impact of the communication optimization on the performance of the dynamic allocation of tasks. On average, the results show an optimization of 30%.

Keywords: Dynamic task allocation; Swarm robotics; Distributed computing; Swarm intelligence; communication topology; cluster.

LISTA DE FIGURAS

1	Tipos de controle	26
2	Busca competitiva	26
3	Busca cooperativa	27
4	Taxonomia dos algoritmos de ADT	29
5	Composição da nova velocidade	43
6	PSO aplicado na minimização do função Rastrigin	45
7	Função Rastrigin	45
8	Topologias de comunicação	47
9	Diferentes topologias em cluster com 12 robôs	48
10	Etapas da comunicação em cluster	48
11	Topologias de comunicação para um enxame de 12 robôs	50
12	Organização da comunicação em clusters de um enxame de 9 robôs	51
13	Fluxograma do algoritmo ADTC	52
14	Representação gráfica da função-objetivo com diferentes proporções	60
15	Enxame de robôs ELISA-3	61
16	Dispositivos embarcados no <i>hardware</i> do robô	62
17	Comunicação dos robôs via RF	65
18	Estrutura da mensagem enviada pelo robô para a estação-base	67
19	Estrutura da mensagem enviada pela estação-base para o robô	67
20	Tipos de mensagens enviadas pelo robô no ADTC	68
21	Formato do pacote das mensagens recebidas pela estação-base	69
22	Formato do pacote das mensagens enviadas pela estação-base	70
23	Mensagens otimizadas	70
24	Dinâmica da formação dos pacotes	71
25	Topologias de comunicação em cluster com 24 robôs divididos em diferentes números de clusters	76
26	Número de mensagens enviadas por ciclo para alocação de 24 robôs com diferentes números de clusters	77
27	Curvas do número médio de mensagens enviadas por ciclo para alocação de 5 tarefas em um enxame de 24 robôs	78
28	Topologias de comunicação em cluster com 24 robôs com números de informantes diferentes	79
29	Tempo de convergência para alocação de 24 robôs variando o número de informantes	79

30	Número de mensagens enviadas para alocação de 24 robôs variando o número de informantes	80
31	Topologias de comunicação utilizadas na implementação do ADTC	81
32	Tempo de convergência obtido pelo algoritmo ADTC	82
33	Total de mensagens recebidas obtido pelo algoritmo ADTC	82
34	Total de mensagens enviadas obtido pelo algoritmo ADTC	83
35	Comparação do tempo de convergência obtidos pelos algoritmos ADTC e ADTG	84
36	Comparação do número de mensagens recebidas pelos algoritmos ADTC e ADTG	85
37	Comparação do número de mensagens enviadas pelos algoritmos ADTC e ADTG	86

LISTA DE TABELAS

1	Especificações técnicas do robô ELISA-3	63
2	Distribuição da proporção-objetivo entre as tarefas	75
3	Número de alocações factíveis para os ensaios realizados	75
4	Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 2$	99
5	Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 3$	100
6	Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 4$	101
7	Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 6$	102
8	Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 8$	103
9	Resultados para a análise do número de informantes com 17% de informantes	104
10	Resultados para a análise do número de informantes com 38% de informantes	105
11	Resultados para a análise do número de informantes com 50% de informantes	106
12	Resultados obtidos pelo ADTC para $\rho = 8$ e $\gamma = 2$	107
13	Resultados obtidos pelo ADTC para $\rho = 12$ e $\gamma = 4$	108
14	Resultados obtidos pelo ADTC para $\rho = 16$ e $\gamma = 4$	109
15	Resultados obtidos pelo ADTC para $\rho = 20$ e $\gamma = 5$	110
16	Resultados obtidos pelo ADTC para $\rho = 24$ e $\gamma = 6$	111
17	Resultados obtidos pelo ADTC para $\rho = 28$ e $\gamma = 7$	112
18	Resultados obtidos pelo ADTC para $\rho = 32$ e $\gamma = 8$	113
19	Resultados obtidos pelo ADTC para $\rho = 40$ e $\gamma = 10$	114
20	Resultados obtidos pelo ADTC para $\rho = 50$ e $\gamma = 10$	115

LISTA DE ALGORITMOS

1	Algoritmo PSO	43
2	Algoritmo ADTC	53
3	Inicialização	53
4	Atualizar melhor alocação no cluster	55
5	Identificação do robô <i>Gbest</i>	56
6	Atualizar melhor alocação global	57
7	Critério de parada	58

LISTA DE SIGLAS

A	Alocação de tarefas para o enxame
C	Conjunto dos contadores do número de robôs alocados a cada tarefa
F_c	Número de mensagens enviadas em cluster por ciclo
F_m	Número de mensagens enviadas em malha completa por ciclo
I	Conjunto dos identificadores dos robôs do enxame
M	Matriz de Comunicação
P	Proporção-objetivo
Q	Conjunto de alocações factíveis
Q_v	Conjunto de alocações válidas
T	Conjunto de identificadores das tarefas a serem alocadas aos robôs
V	Vetor com as qualidades das melhores alocações dos robôs $Pbests$
V^*	Vetor com as qualidades das melhores alocações dos clusters $Cbests$
Y	Número de Combinações Possíveis
ρ	Número de robôs
τ	Número de tarefas para alocar
γ	Número de cluster
σ	Número de informantes por cluster
<i>id</i>	Identificador do robô
<i>Pbest</i>	Melhor solução do robô
<i>Cbest</i>	Melhor solução no clusters
<i>Gbest</i>	Melhor solução do enxame
ADT	Alocação Dinâmica de Tarefas
ADTC	Algoritmo Alocação Dinâmica de Tarefas em Cluster

ADTG	Algoritmo Alocação Dinâmica de Tarefas com Abordagem Global
BF	Byte de confirmação
BV	Byte de validade
CTRL	Byte de controle
EEPROM	Electrically-Erasable Programmable Read-Only Memory
IR	Infrared
ISM	Industrial, Scientific and Medical
LED	Light-emitting diode
RAM	Random Access Memory
RF	Rádio frequência
RGB	Red-Green-Blue
PSO	Particle Swarm Optimization
SPI	Serial peripheral interface
USB	Universal Serial Bus

SUMÁRIO

INTRODUÇÃO	15
1 ALOCAÇÃO DINÂMICA DE TAREFAS	21
1.1 Definição Formal	21
1.2 Espaço de Busca	23
1.3 Estratégias de Alocação Dinâmica de Tarefas	24
1.3.1 <u>Decisão Determinística ou Estocástica.</u>	25
1.3.2 <u>Controle Centralizado ou Distribuído</u>	25
1.3.3 <u>Busca Competitiva ou Cooperativa</u>	26
1.4 Considerações Finais do Capítulo.	27
2 TRABALHOS RELACIONADOS	29
2.1 Algoritmos Bioinspirados.	31
2.2 Algoritmos Baseados nas Leis de Mercado	34
2.3 Algoritmos Orientados à Comunicação	35
2.4 Algoritmos Comportamentais	36
2.5 Algoritmos com Outras Orientações	38
2.6 Considerações Finais do Capítulo.	39
3 ALOCAÇÃO DINÂMICA DE TAREFAS EM CLUSTER	41
3.1 Otimização Baseada em Enxame de Partículas	42
3.1.1 <u>Visão Geral.</u>	42
3.1.2 <u>Topologias de Comunicação.</u>	45
3.1.3 <u>Cluster</u>	47
3.2 Algoritmo Proposto	51
3.2.1 <u>Visão Geral.</u>	51
3.2.2 <u>Etapas do Algoritmo.</u>	53
3.2.2.1 Inicialização	53
3.2.2.2 Ajuste.	54
3.2.2.3 Identificação	56
3.2.2.4 Atualização.	57
3.2.2.5 Critério de Parada	58
3.2.3 <u>Função Objetivo</u>	59
3.3 Considerações Finais do Capítulo.	59
4 IMPLEMENTAÇÃO DO ALGORITMO	61
4.1 O Robô ELISA-3.	61
4.1.1 <u>Arquitetura do Robô.</u>	62
4.1.2 <u>A Arquitetura de Comunicação.</u>	65

4.1.3	O Protocolo de Comunicação	66
4.2	Metodologia da Implementação	67
4.3	Considerações Finais do Capítulo.	72
5	ANÁLISE DOS RESULTADOS	74
5.1	Metodologia de Avaliação	74
5.2	Impacto do Tamanho do Cluster	76
5.3	Impacto do Número de Informantes	78
5.4	Desempenho do ADTC	79
5.5	Comparação entre o ADTC e o ADTG.	81
5.6	Considerações Finais do Capítulo.	87
6	CONCLUSÕES E TRABALHOS FUTUROS	88
6.1	Conclusões	88
6.2	Trabalhos Futuros.	90
	REFERÊNCIAS	92
	APÊNDICE A – RESULTADOS NUMÉRICOS OBTIDOS	97

INTRODUÇÃO

Observando o ambiente que nos cerca podemos encontrar exemplos de inteligência coletiva, como o peixe *Thunnus thynnus* (TEO et al., 2007), conhecido popularmente como atum, e as formigas *Eciton* (FRANKS, 1986). Vivendo numa estrutura social altamente organizada que tem como objetivo a proteção, cada peixe do cardume possui percepção parcial do ambiente, mas que lhe possibilita navegar de maneira organizada formando estruturas que afugentam predadores. Já as formigas *Eciton*, possuem habilidade de organização extremamente complexa, sendo capazes de criar estruturas com elas próprias, como a construção de uma ponte para atravessar determinado obstáculo. Contudo, o exemplo mais significativo, e que mostra o quão poderosa pode ser a inteligência coletiva, é o organismo humano. Constituído por uma infinidade de células de diferentes tipos, cada uma realiza pequenas e simples tarefas, com sincronismo e organização perfeitos, realizando em conjunto todas as nossas funções motoras e intelectuais. A partir desse ponto começamos a vislumbrar a imensidão, ainda, inexplorada dos fundamentos da inteligência de um enxame.

Por esses motivos a comunidade acadêmica, há algum tempo, tem mostrado um grande interesse em pesquisas sobre a inteligência de enxame. A inteligência de enxame é um comportamento cooperativo de estruturas descentralizadas, auto-organizadas e que podem ser naturais ou artificiais. O conceito foi apresentado por Gerardo Beni e Jing Wang em 1993, em um trabalho sobre sistemas robóticos celulares (BENI; WANG, 1993). A inteligência de enxame consiste em um grupo de agentes que interagem entre si e com o ambiente. Impõe aos agentes regras simples que definem como eles devem se comportar, porém, não existe uma estrutura centralizadora que dite o comportamento dos agentes. A interação entre os agentes possibilita o surgimento de um comportamento global inteligente. Sistemas naturais são ótimas inspirações para a inteligência de enxame, como por exemplo, o comportamento de forrageamento e locomoção de bando de pássaros, cardume

de peixes ou colônia de insetos (REYNOLDS, 1987), sistema imunológico (KEPHART et al., 1994) e reprodução bacteriana (DAS et al., 2009).

A inteligência de enxame aplicado a uma plataforma física (robôs) vem sendo chamada, em trabalhos publicados, como robótica de enxame (TAN; ZHENG, 2013; BRAMBILLA et al., 2013). Composto por múltiplos robôs autônomos que interagem entre si, a robótica de enxame é capaz de solucionar problemas complexos, através da cooperação entre robôs que ao executar, cada um, tarefas simples os levam a um objetivo comum. Essa inteligência pode incluir abordagens metodológicas, procedurais, funcionais, busca algorítmica ou aprendizagem por reforço. Possui uma inteligência coletiva que emerge espontaneamente da interação entre os robôs e o ambiente. Pesquisas utilizando a robótica de enxame vêm crescendo na comunidade acadêmica, assim como pesquisas voltadas ao desenvolvimento de novos robôs móveis com maior autonomia e novas funcionalidades (CHRISTENSEN; SCHULTZ; STOY, 2013).

A robótica de enxame possui três características desejáveis para a solução de problemas com alta complexidade: flexibilidade; escalabilidade e robustez. A flexibilidade é o grau de facilidade em se adaptar a novas situações, um sistema flexível se adapta rapidamente a novos problemas mantendo o mesmo nível de desempenho e de organização. Define-se como escalabilidade a capacidade de um determinado sistema de expandir ou comprimir, suportando a variação do número de integrantes no sistema sem a perda de eficiência. A robustez é definida a partir do quanto um determinado sistema é capaz de continuar operante mediante aos erros ou anomalias, mantendo a sua capacidade de busca.

A inteligência de enxame é capaz de solucionar problemas de alta complexidade, pois possibilita a decomposição do problema em tarefas simples. Pesquisas que abordam a inteligência de enxame apresentam-se organizadas em campos de pesquisas, como apresentado em (PARKER, 1994) e (MOHAN; PONNAMBALAM, 2009), que dentre esses campos de pesquisa destaca-se em função do tema abordado: a Alocação de Tarefas (CHAND; CARNEGIE, 2012).

A alocação de tarefas é a maneira como as tarefas são atribuídas, escolhidas, coordenadas e subdivididas, processo fundamental para a utilização da inteligência de enxame objetivando solucionar problemas de alta complexidade. Uma alocação de tarefas eficiente

garante a organização e sincronização da execução de todas as tarefas, que são classificadas como um sub-objetivo necessário para a conclusão do objetivo global.

A comunidade acadêmica tem apresentado inúmeras propostas de algoritmo de alocação de tarefas sob a influência de diferentes inspirações e abordagens como: baseadas nas leis do mercado (LUO; CHAKRABORTY; SYCARA, 2015), comportamentais (TANG; PARKER, 2005), bio-inspiradas (WANG; GU; LI, 2012) e orientadas à comunicação (CARABALLO et al., 2017). Existem diferentes arquiteturas de coordenação da alocação de tarefas de enxame de robôs apresentadas em trabalhos na última década, possuindo diferentes características como organização do grupo, topologia de comunicação e a composição do grupo. Uma classificação mais genérica para o problema de alocação seria a classificação do robô como ST (*single-task robots*) ou MT (*multi-task robots*), capacidade de executar uma ou mais tarefas simultaneamente; a classificação das tarefas como SR (*single-task robots*) ou MR (*multi-robot tasks*), define que cada tarefa requer um ou mais robôs para ser executada; e, a classificação da alocação como IA (*instantaneous assignment*) ou TA (*time-extended assignment*), define que as informações são limitadas e permitem apenas alocações instantâneas ou as informações são completas e permitem um planejamento futuro (GERKEY; MATARIĆ, 2004).

De forma direta, pensa-se em uma solução com abordagem centralizada para o problema de alocação, onde uma entidade central detém o poder de monitoramento do ambiente e controle do enxame. Porém uma abordagem distribuída, onde a solução deve ser encontrada de forma cooperativa por todos os robôs do enxame, aproxima-se do comportamento social de comunidades biológicas. A vida social vem demonstrando ser de grande vantagem, apresentando bons resultados para redução de risco de predação, proteção ambiental, vantagens de acasalamento e aquisição de alimentos. A forma mais avançada de vida social presente na natureza é o eusocialidade, caracterizada pela sobreposição de geração, o cuidado cooperativo entre os indivíduos do enxame e a divisão do trabalho. Os maiores exemplos presentes na natureza de sociedades eusociais, e que tem demonstrado grande eficiência ecológica e evolutiva, são os insetos (formigas, abelhas, vespas, cupins, tripés e afídeos) (KREBS; DAVIES, 1987). Por esse motivo, a alocação de tarefas pode ser melhor resolvida com uma abordagem distribuída, inspirada em resultados biológicos que possuem comprovada eficiência. Seguindo esse mesmo raciocínio a

alocação de tarefas precisa ser um processo dinâmico, continuamente ajustado em função de alterações do ambiente, do enxame e/ou do problema, assegurando uma solução generalizada e resistente a falhas.

A inteligência de enxame fundamenta diferentes técnicas de otimização. O algoritmo proposto nesta dissertação possui inspiração na Otimização Baseada em Enxame de Partículas (*Particle Swarm Optimization* - PSO), proposto em (EBERHART; KENNEDY, 1995). O PSO é um algoritmo estocástico baseado em inteligência de enxame, utilizado normalmente em funções não deriváveis, não lineares e/ou com elevado número de dimensões. Possui inspiração biológica, onde em (REYNOLDS, 1987) é emulado o padrão de voo dos pássaros em busca de alimento e abrigo, e posteriormente, em 1995, esse padrão é adaptado como processo de busca. O PSO coordena um grupo de partículas, onde a posição no espaço de busca de cada partícula é uma potencial solução para o problema. A velocidade da partícula é adaptativa em função de resultados passados, ajustada em função da experiência da própria partícula e das demais do enxame. A cada iteração, a partícula se locomove no espaço de busca tendo sua posição alterada pela velocidade atual. O conhecimento das experiências das demais partículas do enxame só é possível através de um processo de comunicação. Dependendo do tamanho de enxame, o processo de comunicação pode ser custoso e demorado, por esse motivo deve ser implementado de forma otimizada. Além de uma estrutura de mensagem, que possui aproveitamento máximo de sua carga útil, o tipo de topologia de comunicação poderá também impactar no resultado final.

Esta dissertação tem como objetivo otimizar os processos de comunicação na alocação dinâmica de tarefas em um enxame de robôs, com intuito de melhorar os resultados obtidos pelo algoritmo Alocação Dinâmica de Tarefas com Abordagem Global (ADTG) apresentados em (MENDONÇA, 2014). O ADTG realiza a alocação abordando o problema como uma otimização. Inspirado no PSO, no qual cada robô representa uma partícula que possui uma posição correspondente a uma alocação factível, a cada iteração do algoritmo as alocações são avaliadas em função de sua aptidão em atender a proporção-objetivo. Esta abordagem possibilita uma solução totalmente distribuída, permitindo a solução de problemas de alta complexidade, mesmo utilizando robôs limitados. As alocações são adaptativas, desta forma o enxame é continuamente atualizado mediante trocas de in-

formações entre os robôs. Contudo, este tipo de abordagem possui um grande fluxo de informação entre robôs, o que pode acarretar perda de desempenho ou até a inviabilidade de implementação. No ADTG é utilizada a topologia de comunicação em malha completa, em que todos os robôs se comunicam com os demais, gerando crescimento exponencial do fluxo de dados em função do número de robôs. O objetivo é implementar outra topologia de comunicação com o intuito de reduzir o fluxo de informação entre os robôs do enxame, permitindo aumento de desempenho. O algoritmo Alocação Dinâmica de Tarefas em Cluster (ADTC) possui mesma abordagem e metodologia utilizada pelo ADTG, porém, ao invés da comunicação ocorrer através de uma topologia em malha completa, o ATDC utiliza a topologia de comunicação em cluster.

O conceito inicial que levou ao termo cluster foi desenvolvido pela *International Business Machines* (IBM), na década de 1960, como uma maneira de interligar grandes *mainframes* visando obter uma solução viável de paralelismo. A topologia de comunicação em cluster é uma adaptação da topologia em malha completa, desenvolvida com o objetivo de diminuir o fluxo de informações entre os elementos integrantes do grupo. Este último é subdividido em clusters, grupos menores de elementos, onde a comunicação é permitida apenas dentro dos clusters e a troca de informações entre clusters é realizada por pontos de acesso previamente determinados.

Esta dissertação é estruturada em 6 capítulos como descritos a seguir: o Capítulo 1 apresenta a definição formal do problema de alocação dinâmica de tarefas e também uma avaliação da complexidade da Alocação Dinâmica de Tarefas (ADT).

O Capítulo 2 analisa os principais trabalhos publicados nos últimos anos sobre alocação de tarefas em sistemas baseados em robótica de enxame e as diferentes metodologias e técnicas utilizadas para a sua implementação.

O Capítulo 3 apresenta as características do algoritmo de otimização PSO e como será a adaptação da topologia de comunicação em cluster ao propor um algoritmo estocástico para alocação dinâmica de tarefas para um enxame de robôs. Apresentando suas etapas de implementação e características estruturais, demonstra de que forma a otimização da comunicação será realizada.

O Capítulo 4 demonstra como é realizada a implementação do algoritmo proposto na plataforma física escolhida (ELISA-3), descrevendo detalhadamente as estruturas do

hardware, como os dispositivos embarcados disponíveis e suas funcionalidades. Apresenta também como é realizada a implementação do processo de comunicação e as estruturas de mensagens utilizadas.

O Capítulo 5 apresenta uma análise dos resultados obtidos nos ensaios com diferentes arranjos de enxame, avaliando o número ideal de clusters e de informantes. Também avalia a capacidade e o desempenho do algoritmo proposto, analisando o tempo de convergência e o fluxo de informação. Ao final, são comparados os resultados obtidos com os apresentados em (MENDONÇA, 2014).

Por último, no Capítulo 6, são apresentadas as considerações finais da dissertação com relação aos resultados obtidos, os aspectos observados durante os ensaios e são propostas melhorias para o algoritmo e considerações para trabalhos futuros.

Capítulo 1

ALOCAÇÃO DINÂMICA DE TAREFAS

A inteligência de enxame introduz, através da cooperação entre indivíduos, a possibilidade de decomposição de uma tarefa complexa em uma série de tarefas mais simples. Proposta por Gerardo Beni and Jing Wang em 1989 (BENI; WANG, 1989), pela observação de comunidades que realizam atividades coletivas, como as formigas ou outras comunidades que possuem comportamentos biológicos auto-organizados. Um enxame de robôs é constituído por um determinado número de robôs, onde cada um deles deve ser independente e autônomo, possuindo percepção parcial do ambiente e do problema e capacidade de atuação e processamento limitados.

A Alocação Dinâmica de Tarefas é o processo que gerencia e organiza um enxame de robôs com a finalidade de executar tarefas que os levarão à um único objetivo. Este processo consiste em determinar, de uma forma organizada, qual tarefa deve ser realizada por cada um dos robôs do enxame. Também é necessário que a alocação seja dinâmica, pois deve reagir constantemente as variações do ambiente e do enxame.

Este capítulo é estruturado da seguinte forma: a Seção 1.1 define o problema da alocação dinâmica de tarefas; na Seção 1.2 é apresentado o espaço de busca; a Seção 1.3 apresenta diferentes tipos de estratégias para alocação e na Seção 1.4 são realizadas as considerações finais para o capítulo.

1.1 Definição Formal

Para existir cooperação entre os robôs há a necessidade de uma alocação de tarefas eficiente. O processo de Alocação Dinâmica de Tarefas (ADT) é reativo as variações do ambi-

ente e do enxame. Um enxame possui a capacidade de solucionar problemas complexos, permitindo a realização de trabalhos paralelos e a simplificação do problema. Buscando uma representação formal do problema de ADT, seja o conjunto de identificadores das tarefas $\mathbb{T} = \{t_1, t_2, \dots, t_\tau\}$ a serem alocadas aos indivíduos do enxame, composta por τ tarefas válidas.

A alocação de tarefas só é necessária após o processo de decomposição do problema em uma série de pequenas tarefas, essa decomposição gera um vetor de proporções $\mathbb{P} = \{p_1, p_2, \dots, p_\tau\}$ que é definida por um conjunto de números naturais positivos $p_j \in \mathbb{N}^*$, que representa o número de robôs necessários para realizar a tarefa t_j , onde o somatório é igual a 1, conforme apresenta a Equação 1:

$$\sum_{j=1}^{\tau} p_j = 1. \quad (1)$$

Em um enxame composto por ρ robôs, a partir da proporção \mathbb{P} desejada, podemos calcular o número de robôs alocados para cada tarefa, representado por um conjunto de contadores $\mathbb{C} = \{c_1, c_2, \dots, c_\tau\}$, tal que cada contador é o produto da proporção vezes o número de indivíduos $c_j = p_j \times \rho$, onde o somatório dos contadores deve ser ρ , como demonstra a Equação 2:

$$\sum_{j=1}^{\tau} c_j = \rho. \quad (2)$$

Em um enxame, cada robô possui um identificador id , único e exclusivo. O conjunto de identificadores é representado por $\mathbb{I} = \{id_1, id_2, \dots, id_\rho\}$. A alocação da tarefa dos indivíduos do enxame é representada por $\mathbb{A} = \{a_1, a_2, \dots, a_\rho\}$, onde a_j identifica a tarefa a ser executada pelo indivíduo id_j . A partir da alocação \mathbb{A} podemos calcular a quantidade de robôs alocados em cada uma das tarefas representado pelo conjunto de contadores $\mathbb{C}_{\mathbb{A}} = \{c_1, c_2, \dots, c_\rho\}$, como mostra a Equação 3:

$$c_j = \mathbb{C}_{\mathbb{A}}[t_j] = \sum_{r=1}^{\rho} \theta(a_r, t_j), \quad (3)$$

onde a função θ é definida conforme apresentado na Equação 4:

$$\theta(a, t) = \begin{cases} 1 & \text{se } a = t; \\ 0 & \text{senão.} \end{cases} \quad (4)$$

A solução do problema de alocação consiste em encontrar uma alocação $\mathbb{A}^* = \{a_1^*, a_2^*, \dots, a_\rho^*\}$ que atenda a proporção desejada conforme apresenta a Equação 5:

$$\forall t_j \in \mathbb{T} \text{ e } \forall p_j \in \mathbb{P}, \mathbb{C}_{\mathbb{A}^*}[t_j] = p_j \times \rho. \quad (5)$$

Para encontrar uma solução utilizando um algoritmo de otimização é necessário uma função objetivo, que deverá calcular o erro introduzido pela alocação encontrada \mathbb{A} em relação a alocação desejada \mathbb{A}^* . A função objetivo irá fornecer um valor de qualidade para alocação encontrada conforme mostra a Equação 6:

$$f(\mathbb{A}) = \frac{\sum_{j=1}^{\tau} |\mathbb{C}[j] - \mathbb{C}_{\mathbb{A}}[j]|}{\tau}, \quad (6)$$

onde τ é o número de tarefas válidas, $\mathbb{C}[j]$ é a quantidade desejada de robôs alocados para tarefa j de acordo com a proporção-objetivo \mathbb{P} e $\mathbb{C}_{\mathbb{A}}[j]$ é o número de robôs alocados na tarefa j de acordo com a alocação encontrada \mathbb{A} .

Com o objetivo de exemplificar a definição formal do problema de ADT, idealiza-se um enxame de 10 robôs que precisa executar 2 tarefas, cortar e carregar, sendo necessário 30% do robôs cortando e 70% dos robôs carregando. Nesse caso, tem-se a seguinte definição formal do ADT: $\tau = 2$; $\mathbb{T} = \{\text{cortar}, \text{carregar}\}$; $\mathbb{P} = \{0.3, 0.7\}$; $\rho = 10$; $\mathbb{C} = \{3, 7\}$; $\mathbb{I} = \{3341, 3358, 3365, 3259, 3247, 3456, 3221, 3288, 3398, 3365\}$ e $\mathbb{A}^* = \{1, 1, 0, 1, 1, 0, 1, 1, 1, 0\}$.

1.2 Espaço de Busca

Ao analisar o espaço de busca pode-se observar a complexidade do problema de alocação de tarefas. Deseja-se encontrar a alocação \mathbb{A}^* , que atenda a proporção-objetivo \mathbb{P} , que pertence a um conjunto que é chamado de alocações factíveis \mathbb{Q} , esse conjunto representa todas as possíveis soluções para um determinado problema de ADT. O número de alocações factíveis \mathbb{Q} depende do número de robôs no enxame ρ e do número de tarefas τ a serem alocadas, como demonstra a Equação 7:

$$\mathbb{Q} = \tau^\rho. \quad (7)$$

O problema de ADT apresenta uma complexidade computacional alta, são problemas polinomiais, não determinísticos e que possuem grande espaço de busca. O tamanho

do espaço de busca de um problema de ADT está em função do número de robôs ρ e do número de tarefas τ , que representam, respectivamente, o número de dimensões e o intervalo de busca. Para exemplificar em um enxame de 25 robôs com 5 tarefas a serem alocadas, terá mais 298 quatrilhões de alocações factíveis.

Dentro de todas as alocações factíveis existe uma quantidade de alocações válidas, que são quaisquer alocações factíveis que satisfazem os requisitos do problema. Uma mesma proporção-objetivo \mathbb{P} pode ser atendida por diferentes alocações, pois quaisquer uns dos robôs possui a capacidade de realizar qualquer uma das tarefas.

As alocações válidas, representadas por \mathbb{Q}_v , dependem do número de tarefas τ , do número de robôs ρ e da proporção-objetivo \mathbb{P} . Baseado na definição de combinação simples (GRAHAM, 1995), o cálculo de combinações possíveis $\mathbb{Y} = \{y_1, y_2, \dots, y_\tau\}$ é demonstrado pela Equação 8:

$$y_j = \frac{\rho!}{c_j! \times (\rho - c_j)!}, \quad (8)$$

onde c_j é o número ideal de robôs alocados na tarefa j e y_j é o número de combinações possíveis da tarefas j . O número de alocações válidas \mathbb{Q}_v é o somatório das combinações, conforme apresentado pela Equação 9:

$$\mathbb{Q}_v = \sum_{j=1}^{\tau} y_j. \quad (9)$$

Utilizando o mesmo exemplo, para um enxame com 25 robôs, 5 tarefas e uma proporção $\mathbb{P} = \{\frac{1}{25}, \frac{9}{25}, \frac{5}{25}, \frac{2}{25}, \frac{8}{25}\}$, terá mais 3 milhões de alocações válidas.

1.3 Estratégias de Alocação Dinâmica de Tarefas

A definição da metodologia empregada no processo de ADT é de extrema importância para o alcance dos resultados esperados, e essa definição se baseia nos tipos de estratégias escolhidas de acordo com o método de decisão, com o tipo de controle e o método de busca. Tendo em vista o método de decisão, o processo de ADT pode assumir um comportamento de decisão determinístico ou estocástico, conforme descrito na Seção 1.3.1. Quanto ao tipo de controle utilizado, a solução para o problema pode ser encontrada de uma forma centralizada ou distribuída, conforme apresentado na Seção 1.3.2. Por fim, a metodologia de busca pode ser de uma forma cooperativa ou competitiva, conforme descreve a Seção 1.3.3.

1.3.1 Decisão Determinística ou Estocástica

Um processo determinístico é onde uma ação gera um único resultado. Esse tipo de decisão é baseado em processos não probabilísticos, o que permite a previsão do comportamento durante o processo de ADT. Na maioria das vezes, um processo determinístico é modelado analiticamente. Na ADT essa modelagem é definida pela configuração inicial do problema, tais como a proporção-objetivo \mathbb{P} , número de tarefas τ e número de robôs ρ . Uma determinada configuração inicial sempre gerará uma mesma solução final. Uma estratégia determinística, normalmente, é mais simples e de fácil implementação.

A estratégia estocástica é baseada em eventos probabilísticos, onde os estados futuros são indeterminados e desconhecidos. Ao incluir elementos aleatórios torna-se possível a obtenção de diferentes soluções para a ADT, introduzindo, dessa maneira, uma capacidade exploratória do espaço de busca, gerando soluções inesperadas, podendo obter resultados, até o momento, desconhecidos. É um complemento do processo determinístico, ao invés de possuir um único caminho para evoluir, uma decisão estocástica possui uma indeterminação: mesmo com condições iniciais conhecidas, existem várias direções nas quais a busca pode seguir.

1.3.2 Controle Centralizado ou Distribuído

O controle de um enxame de robôs pode ser realizado através de uma rede de comunicação centralizada ou distribuída, definida em (IOCCHI; NARDI; SALERNO, 2001) e (STEELE; THOMAS, 2007). A estratégia centralizada, como mostra a Figura 1(a), consiste em centralizar toda a informação do enxame em um agente central único. O agente central é responsável por reunir toda a informação do enxame e munido dessa informação poderá determinar a alocação adequada para todos os robôs do enxame. Essa estratégia possui um baixo fluxo de informação, porém, possui uma forte dependência do agente centralizador, o que gera uma fragilidade, pois o mau funcionamento do agente central inviabiliza totalmente a ADT.

A estratégia distribuída não possui um agente centralizador, como mostra a Figura 1(b), onde a solução é determinada coletivamente por todos os robôs do enxame. Esse tipo de controle possui maior tolerância a falhas, pois não existe grande dependência de qualquer um dos robôs do enxame. Essa estratégia é mais adaptável às variações

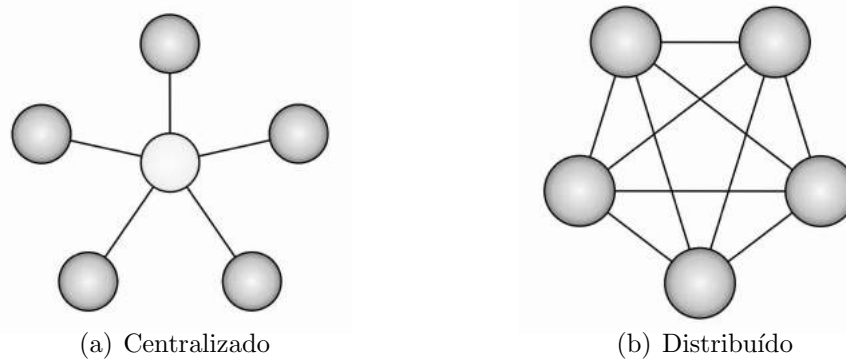


Figura 1: Tipos de controle

do ambiente e/ou do próprio enxame, no entanto, possui um maior fluxo de informação, que cresce exponencialmente em função do número de robôs. Outra desvantagem desse tipo de estratégia é a necessidade de uma organização mais complexa no processo de troca de informação entre os robôs, que deve ser sincronizada e ordenada, tornando sua implementação mais difícil.

1.3.3 Busca Competitiva ou Cooperativa

Ambos os tipos de busca tem como objetivo explorar a possibilidade do paralelismo que existe ao utilizar um enxame de robôs. O enxame de robôs possui um objetivo comum a todos, a busca por uma solução satisfatória para o problema pode ser competitiva ou cooperativa.

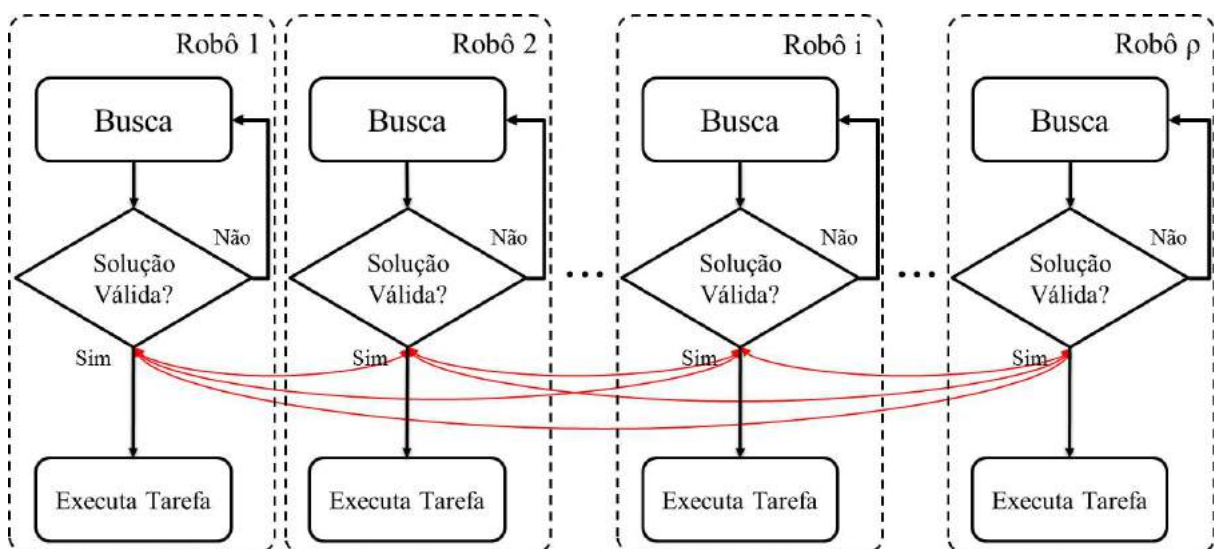


Figura 2: Busca competitiva

Na busca competitiva cada robô busca de forma individual por uma solução para a ADT, como mostra a Figura 2. Neste tipo de busca só existe comunicação entre os robôs para transmitir a informação de que a alocação foi encontrada e qual é essa alocação. O primeiro robô que encontrar uma alocação válida informa aos demais robôs, comunicando a alocação encontrada. Possui como vantagem o baixo fluxo de comunicação, no entanto exige alta capacidade de processamento de cada robô, o que tornar inviável a implementação desse tipo de busca dependendo da complexidade do problema.

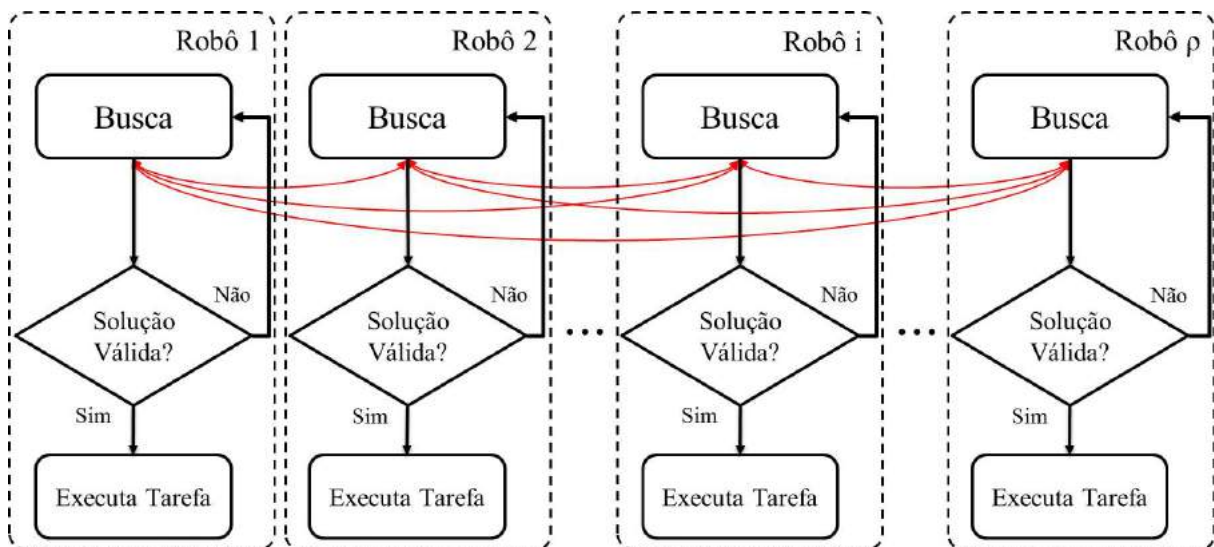


Figura 3: Busca cooperativa

Na busca cooperativa o enxame de robôs encontra uma solução para a ADT de forma coletiva, como mostra a Figura 3. Esse tipo de busca não exige uma alta capacidade de processamento, pois permite a desfragmentação do problema em subproblemas menos complexos, permitindo a solução de problemas de alta complexidade. No entanto, a busca cooperativa possui uma alta taxa de comunicação, podendo até inviabilizar sua utilização. Esse tipo de busca necessita maior refinamento nos processos de comunicação para garantir um bom desempenho global.

1.4 Considerações Finais do Capítulo

Neste capítulo definiu-se o problema da alocação dinâmica de tarefas para um enxame de robôs. Apresentou-se uma modelagem para viabilizar o tratamento desse tipo de problema como um problema de otimização, idealizando a função objetivo. Demonstrou-se as diferentes abordagens para a ADT, analisando as diferenças entre uma decisão

determinística ou estocástica, entre um controle centralizado ou distribuído e entre uma busca competitiva ou cooperativa.

Inúmeras pesquisas a respeito da alocação dinâmica de tarefas foram realizadas nos últimos anos. Buscando realizar a alocação de tarefas de uma forma mais eficiente foram publicadas diversas estratégias. No capítulo seguinte são apresentados os trabalhos recentes que são relacionados com algoritmos para a alocação de tarefas em enxame de robôs.

Capítulo 2

TRABALHOS RELACIONADOS

ESTE capítulo apresenta alguns trabalhos sobre alocação dinâmica de tarefas (ADT). Os trabalhos expostos a seguir apresentam diferentes abordagens para o problema de ADT, porém sempre em sistemas com multi-robôs e em sua maioria de forma distribuída. Neste capítulo é demonstrada a diversidade e a flexibilidade do uso da ADT em diferentes ambientes e contextos.

Este capítulo é organizado em seções, onde em cada uma são agrupados trabalhos com a mesma taxonomia. Existe uma grande quantidade de soluções para ADT na comunidade acadêmica e é possível classificá-los em função da taxonomia apresentada em (ZHANG; LIU, 2008), onde os algoritmos são classificados como bioinspirados, baseados nas leis do mercado, orientados à comunicação e comportamentais. Na Figura 4 é ilustrada a taxonomia para alguns dos principais algoritmos utilizados no processo de alocação dinâmica de tarefas.

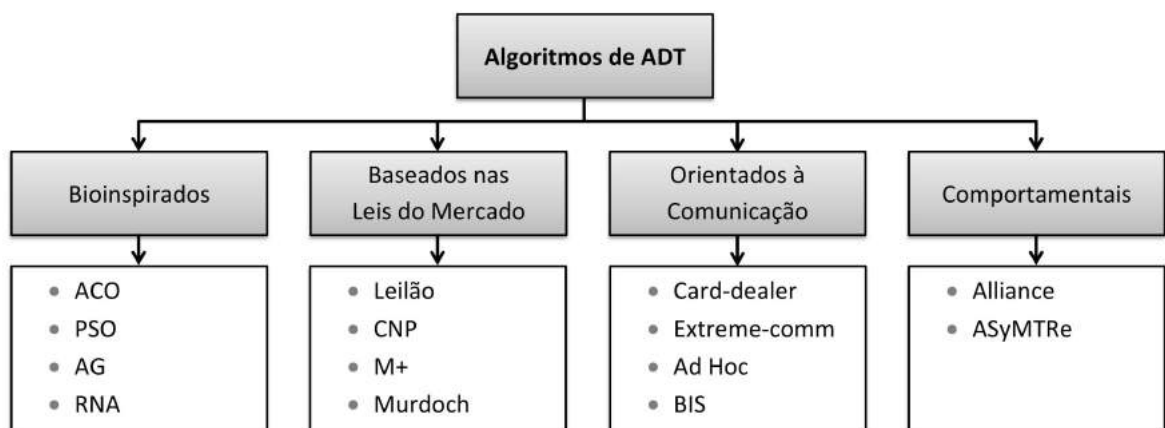


Figura 4: Taxonomia dos algoritmos de ADT

Algoritmos bioinspirados são largamente pesquisados e investigados na comunidade acadêmica. A biologia é uma vasta fonte de inspiração para pesquisas voltadas ao desenvolvimento de inteligências autônomas situadas em ambientes dinâmicos. Existem diferentes técnicas de otimização bioinspiradas, como algoritmo genético, sistemas imunológicos artificiais e inteligência coletiva (GAO, 2012). Em relação ao problema de ADT, a inteligência coletiva tem sido grande fonte de inspiração, pois possui características que auxiliam a alocação de tarefas como a capacidade de auto-organização e flexibilidade às variações do ambiente.

Os algoritmos baseados nas leis do mercado possuem como inspirações conceitos básicos da economia, como os de oferta e demanda, maximização de lucros, redução de custos e sistemas de leilão. Estratégias baseadas nas leis do mercado idealizam algum tipo de moeda de troca, que será utilizado pelos robôs e pelas tarefas, como forma de pagamento, troca ou lances. Esta moeda, na maioria das vezes, representa os objetivos a serem otimizados, como tempo, distância ou energia.

Algoritmos orientados a comunicação possuem uma estratégia baseada na troca de informação entre os robôs do enxame. Como cada um deles possui conhecimento parcial do problema, a troca de informação permite que o robô tenha um conhecimento mais abrangente podendo fazer escolhas mais assertivas. Estes algoritmos buscam a otimização da estrutura de comunicação, para que a troca de informação seja mais rápida e menos custosa. Na maioria das implementações desse algoritmo a busca da solução é centralizada, onde o problema é tratado de forma completa. O compartilhamento da informação possibilita uma solução centralizada, pois todos os robôs conhecem o tamanho do enxame e suas características. Com o conhecimento completo do enxame é possível a alocação direta e dinâmica das tarefas.

E por último, os algoritmos comportamentais possuem uma estratégia baseada em motivações, como recompensas, reforços ou punições e geralmente são aplicados em um grupo de robôs heterogêneos. O problema é dividido em grupos comportamentais, que em relação à ADT são as diferentes tarefas que precisam ser alocadas. Cada robô possui uma motivação que irá acionar um determinado grupo comportamental, este por sua vez irá definir a tarefa a ser realizada. Após o comportamento motivacional de todo o enxame, avalia-se as necessidades de cada grupo e a alocação é realizada em função das

necessidades e da capacidade de cada robô em realizar uma determinada tarefa, visando sempre atender as necessidades do problema mas também respeitar as características de cada robô.

Este capítulo é estruturado da seguinte forma: a Seção 2.1 apresenta alguns dos algoritmos bioinspirados, mais especificamente, algoritmos inspirados nos comportamentos coletivos de inteligência de enxames; a Seção 2.2 apresenta algoritmos inspirados nas Leis de Mercado; a Seção 2.3 apresenta alguns dos algoritmos orientados à Comunicação; a Seção 2.4 introduz alguns algoritmos comportamentais; a Seção 2.5 apresenta alguns algoritmos com outras orientações; e a Seção 2.6 apresenta as considerações finais do capítulo.

2.1 Algoritmos Bioinspirados

Em (WANG; GU; LI, 2012), é proposto uma nova solução, utilizando Otimização por Colônia de Formigas (*Ant colony optimization - ACO*) (SALAMI, 2009), para o problema de alocação de tarefas em sistemas multi-robôs (*multi-robot task allocation - MRTA*). Em um ambiente com N tarefas e M robôs, onde cada um deles é capaz de executar apenas uma tarefa. Cada robô possui uma classificação de desempenho em relação a cada tarefa, assim se um deles é incapaz de realizar uma determinada atividade sua classificação será zero. O objetivo é atribuir aos robôs as tarefas de modo a maximizar a eficiência, levando em consideração as características de cada um. O processo de alocação é realizado tarefa por tarefa, se uma das tarefas exigir apenas um robô, a alocação é realizada diretamente alocando o mais apto, caso necessite mais de um, utiliza-se o ACO para realizar a alocação. O ACO selecionará os robôs mais adequados, onde cada um será representado por uma formiga; as tarefas são representadas pelos nós; e a distância entre a formiga e o nó será a eficiência do robô ao executar a tarefa.

Este trabalho demonstra a eficiência do ACO comparando-o com outros três algoritmos, M+ (BOTELHO; ALAMI, 1999), ALLIANCE (PARKER, 1998) e BLE (WERGER; MATARIĆ, 2000). Este apresenta um método para solução do problema de alocação dinâmica de tarefas eficiente, porém de forma centralizada, criando-se um agente centralizador, implícito, que aloca os robôs, tarefa por tarefa, utilizando uma estratégia previamente definida.

Em (ZHANG et al., 2007) é estabelecido uma hierarquia para a atribuição das tarefas inspirado em comunidades de insetos (rainha, soldado e operário), que identifica os robôs especialistas. Estes por sua vez são alocados utilizando um método de reforço de aprendizado e os demais robôs são alocados utilizando o ACO. Na hierarquia superior o sistema de aprendizagem com auto reforço é utilizado para identificar os robôs especialistas em cada tarefa, a alocação é realizada por um modelo probabilista inspirado no comportamento social de insetos (BONABEAU et al., 1997). Na hierarquia inferior utiliza-se o conceito de feromônio artificial como ideia-chave da estratégia de alocação. Os robôs adicionam feromônio em tarefas que necessitam de um número maior de robôs para ser executadas. Assim, tarefas que exigem um número maior de robôs vão, na média, possuir uma quantidade de feromônio maior, o que irá lhe proporcionar uma maior probabilidade de ser escolhida. Cada robô possui uma lista de candidatas onde se encontram as tarefas preferidas, nas quais ele não estará restrito a ser alocado, entretanto dará preferência a elas.

O modelo é testado em uma plataforma simulada, em (JONES; MATARIC, 2003) CCFT e descrito com uma variação do forrageamento tradicional, que consiste em um ambiente com vários objetos diferentes que devem ser coletados, onde a dificuldade é desconhecida pelos robôs. Cada objeto possui um peso diferente que determinará o número de robôs necessários para coletá-lo. O modelo propõe uma segregação hierárquica do enxame, o que limita o número de possíveis soluções e sobrecarrega um subgrupo de robôs.

Em (JEVTIC et al., 2012) propõem um algoritmo de abelhas distribuídas (DBA) para alocação de tarefas em um enxame de robôs. O método é desenvolvido em um ambiente multi-robôs homogêneos e distribuídos que aborda um problema ST-MR-IA (*single-task robots, multirobot tasks and instantaneous assignment*). A alocação de tarefas consiste em atribuir a cada robô um destino presente em um arena 2D, sendo cada um deles representados por uma abelha que é inicializada em um ponto aleatório da arena e cada tarefa é representada por um destino. A escolha do destino é feita individualmente por cada abelha utilizando um sistema probabilístico do tipo roleta, onde a probabilidade de cada destino está em função do custo e da qualidade. O custo é calculado como sendo a distância euclidiana entre a abelha e o alvo e a qualidade é um valor escalar que representa

a prioridade ou a complexidade de uma tarefa, por exemplo, o número de robôs necessários naquela tarefa. Cada abelha pode escolher qualquer um dos destinos, porém os destinos mais próximos e com maior prioridade possuem maior probabilidade de serem escolhidos. A alocação dos robôs depende da distribuição inicial das abelhas, como essa distribuição é aleatória, o custo também será aleatório, por isso a qualidade do alvo é utilizada para controle da alocação dos robôs. O custo é responsável por introduzir uma dinâmica na alocação, onde é possível encontrar diferentes soluções para uma mesma alocação.

O autor comprova o funcionamento e a estabilidade desses métodos em uma plataforma simulada, onde se observa a dificuldade de implementação do método em uma plataforma física, de maneira distribuída onde cada robô seria uma abelha. Esta dificuldade existe por dois motivos: o primeiro é que não existe um protocolo que organize algum tipo de comunicação entre as abelhas, dificultando uma cooperação entre robôs; o segundo, é que a distribuição espacial influencia a escolha das tarefas, com isso em uma plataforma física o cálculo de uma posição no espaço é muito complexo.

Em (NEDJAH; MENDONÇA; MOURELLE, 2015) é apresentado um algoritmo distribuído e estocástico para alocação de tarefas em um enxame de robôs homogêneos. A estratégia apresentada pelos autores é baseada no PSO (*Particle Swarm Optimization*), que possui inspiração no comportamento social e cooperativo demonstrado por várias espécies ao locomover-se em bandos ou ao buscar alimento. O algoritmo guia-se na experiência pessoal (Pbest), na experiência geral (Gbest) e na velocidade atual das partículas para decidir as posições seguintes no espaço de busca. A abordagem, ADTG (Alocação Dinâmica de Tarefas com abordagem Global), propõem uma solução estocástica para a alocação utilizando o algoritmo de otimização por enxame de partículas (PSO), onde nesse contexto cada robô representa uma partícula do PSO e de uma forma coletiva encontra uma solução. O ADTG por ser uma estratégia distribuída de forma que a responsabilidade de cada robô seja simples, torna esse método robusto e escalável. Independente do tamanho do problema, essa estratégia permite sua divisão em sub-tarefas simples para que um grupo de robôs possa encontrar uma solução de forma cooperativa.

Essa abordagem possui alta taxa de comunicação, o que impacta muito o desempenho. Nesta é utilizada uma topologia de comunicação em malha completa, onde todos os robôs precisam se comunicar com todos os outros, tornando o crescimento do fluxo

de informação exponencial em função do tamanho do enxame. Nesta dissertação é apresentada a mesma estratégia de solução para a ADT, porém é utilizado a topologia de comunicação em cluster com o objetivo de diminuir o fluxo de informação.

2.2 Algoritmos Baseados nas Leis de Mercado

Em (LUO; CHAKRABORTY; SYCARA, 2015) é apresentada uma solução distribuída para alocação de tarefas em sistema multi-robôs, onde cada tarefa possui um tempo necessário de execução. Cada robô possui uma determinada limitação, que está em função da carga da bateria, o que introduz um limite superior sobre a quantidade de tempo que possui para executar tarefas. Cada tarefa necessita de um determinado tempo de execução e cada robô pode receber uma recompensa diferente, pois quanto maior o tempo gasto, maior a recompensa. O objetivo é maximizar o pagamento total recebido pelos robôs, respeitando, porém, os prazos exigidos pelas tarefas e as limitações das baterias dos robôs. A alocação é realizada por um sistema de leilão (BERTSEKAS, 1988) que tenta combinar as restrições das tarefas e dos robôs, onde cada um é um agente econômico trabalhando em prol dos seus interesses. Cada robô procura dentro de um grupo de tarefas possíveis para ele, atender a que ofereça o maior pagamento. Esse objetivo pode acarretar conflitos caso haja o interesse de diferentes robôs em uma mesma tarefa, por isso é estabelecido um sistema de licitações, onde cada um dá um lance para cada uma das tarefas que atenda as restrições. É atribuído a cada uma das tarefas os robôs que deram os melhores lances.

Pode-se observar que os robôs com menores limitações, ou seja, maiores cargas na bateria, possuem prioridade na alocação, pois podem dar lances maiores. E os com maiores limitações serão os últimos alocados. Isso caracteriza um processo de alocação elitista, onde a cada etapa se aloca o robô mais apto.

Em (SARIEL-TALAY; BALCH; ERDOGAN, 2009) é proposta uma abordagem para o problema de alocação de tarefas em um ambiente de múltiplos caixeiros viajantes. O MTRP (*multiple traveling robot problem*) é uma solução para um problema de busca, onde o objetivo é encontrar as melhores rotas para que os robôs cumpram os objetivos estabelecidos. A solução do problema são rotas que levam os robôs de um ponto inicial ao seu destino final. O problema de alocação está em função da escolha dos destinos para cada um dos robôs. A qualidade da solução depende do tamanho das rotas e a de alocação

eficiente dos destinos, que está em função das posições dos robôs. A alocação é dinâmica e se adapta as condições do ambiente e dos robôs, tendo como objetivo minimizar as rotas e alocar coerentemente cada robô no destino mais adequado. Existe uma forte conexão entre a geração das rotas e a alocação dos destinos. Inicialmente cada robô calcula a distância euclidiana entre ele os destinos próximos não visitados, em seguida seleciona o mais adequado para ele e anuncia sua intenção de executar essa tarefa.

A estratégia de alocação é baseada na CNP (*Contract Net Protocol*) (SMITH, 1980), responsável em determinar os robôs mais adequados para executar cada uma das tarefas. O CNP é um sistema de contratação, utilizado por empresas, para regular o intercâmbio de bens e serviços. É um sistema centralizador, que organiza licitações para reunir todas as intenções dos colaboradores ligados a rede, com intuito de comparar e encontrar o mais apto para uma determinada tarefa. A alocação não é encontrada de uma forma coletiva, não existe uma cooperação de todos os robôs na busca de uma solução.

2.3 Algoritmos Orientados à Comunicação

Em (CARABALLO et al., 2017) é proposta uma nova estratégia com o uso do BIS (*block-information-sharing*), compartilhamento de informações em blocos. BIS é uma estratégia distribuída para alocação dinâmica de tarefas em um enxame de robôs, que possui os princípios de compartilhamento e divisão com o objetivo de alcançar a melhor alocação em função das capacidades dos robôs. A estratégia baseia-se na comunicação, como os recursos de comunicação são escassos, cria-se uma estrutura organizada, onde cada robô só pode se comunicar com seus vizinhos. Os robôs são subdivididos em blocos e os blocos formam um grafo de comunicação. A estratégia de alocação de tarefas BIS é a organização dos blocos e do grafo, criando diferentes tipos de estruturas de blocos na busca de um melhor resultado.

A avaliação de uma determinada estrutura escolhida pelo BIS está em função da complexidade das tarefas e a capacidades dos robôs em realizá-las. A estratégia BIS é baseada no compartilhamento de informação, ou seja, trata o problema de alocação de forma centralizada, deixando de explorar uma grande vantagem da inteligência de enxame.

Em (LEE; ZAHEER; KIM, 2014) é apresentada uma abordagem descentralizada para a solução do problema de ADT, considerando as limitações da comunicação e dos recursos

dos robôs. A abordagem consiste em dois algoritmos, um para geração dos custos (RCG) e outro para alocação de tarefas baseado em uma rede ad hoc (ANTA). O algoritmo RCG permite que cada robô calcule o custo credível ao executar uma tarefa, considerando os recursos gastos. O algoritmo ANTA gera uma rede do tipo árvore com abrangência mínima entre os robôs, utilizada para identificar o com menor custo para executar uma determinada tarefa. Qualquer robô que deseje alocar uma tarefa se torna agente raiz e envia uma solicitação para aquela tarefa. Os robôs com um determinado número de pulos de distância máxima em relação ao agente raiz constroem uma rede em árvore com abrangência mínima. Ao construir a rede em árvore, cria-se uma estrutura organizada para a comunicação, onde cada robô irá se comunicar apenas com seus pais e filhos. Através de uma comunicação multihop ad hoc (JÜTTNER; MAGI, 2005) é transmitido o identificador e o custo do robô com o menor custo, após certo tempo o ID do vencedor, enviado nas mensagens, converge para um único robô e a tarefa é alocada a este.

Essa abordagem apresenta uma topologia de comunicação eficiente que demonstra bons resultados, pois aperfeiçoa a troca de informação entre os robôs, diminuindo drasticamente o fluxo de informação em comparação com a topologia em malha completa. Contudo este tipo de abordagem possui uma grande fragilidade, que é a dependência de todos os colaboradores envolvidos na rede em árvore para que a alocação seja feita de forma correta.

2.4 Algoritmos Comportamentais

Em (TANG; PARKER, 2005) é apresentado o algoritmo ASyMTRe (*Automated Synthesis of Multi-robot Task solutions through software Reconfiguration*) que introduz uma inovação ao propor uma metodologia para sintetizar automaticamente a distribuição de tarefas para equipes multi-robôs, em contraste com abordagens anteriores que exigem uma predefinição manual. A estratégia do ASyMTRe é baseada no mapeamento do ambiente e na percepção e movimentação do robôs. É possível dividir a arquitetura comportamental em: sensoriamento, percepção, movimentação e comunicação. A comunicação é o comportamento que mais contribui para se chegar a solução, pois realiza a troca de informações com os demais comportamentos garantindo a interação entre os robôs, ampliando a percepção individual do problema. A ideia básica do ASyMTRe é ser um algoritmo ganancioso que

lida primeiro com os robôs com menos recursos, onde a cada um é atribuído no início uma tarefa de acordo com sua capacidade de detecção. Os robôs menos capazes tem prioridade, pois possui maior probabilidade de fracasso. Caso este robô não tenha os atributos necessários para concluir a tarefa, o algoritmo encontrará outro para auxiliá-lo, garantindo dessa forma o sucesso em todas as tarefas alocadas. Os robôs com maiores recursos formam um banco de reservas, pois podem futuramente ser úteis de diferentes formas.

O ASyMTRe demonstra, a partir dos resultados alcançados, que é uma poderosa ferramenta para organização e alocação de tarefa em sistema multi-robôs. Porém as soluções são encontradas de maneira centralizada e com dependência direta de uma intensiva comunicação.

Em (PARKER, 1998) é apresentado outro algoritmo comportamental, ALLIENCE, que facilita o controle cooperativo do processo de alocação de tarefas em sistemas multi-robôs. Este algoritmo permite que, individualmente, cada robô possa escolher uma tarefa apropriada durante a busca da solução, porém a escolha deve ser baseada nos requisitos da alocação desejada, nas atividades dos outros robôs, nas condições atuais do ambiente e nos estados internos do próprio robô. O ALLIENCE, é uma arquitetura distribuída baseada no comportamento individual de cada robô, incorpora motivações modeladas matematicamente, como a impaciência e a aquiescência, com o objetivo de estimular ações adaptativas. O robô não é capaz de identificar a ação dos demais, ele possui a capacidade para aprender as ações dos companheiros de equipe através de uma comunicação explícita, onde cada um envia para os outros a tarefa que está executando. O ALLIENCE tem forte dependência da comunicação, pois cada robô precisa, para realizar uma ação, ter a percepção de suas ações, das ações dos outros e do ambiente. Os robôs são programados para trabalhar em cooperação com os demais, e não em concorrência direta, mas mesmo assim conflitos podem ocorrer, pois estão em um ambiente compartilhado. Nenhum sistema centralizado é utilizado com o objetivo de implementar um sistema cooperativo puramente distribuído que proporciona um grau maior de robustez, pois os agentes são totalmente autônomos, capazes de realizar ações úteis mesmo com falhas de outros robôs.

Este algoritmo demonstra bons resultados e comprova ser uma boa ferramenta para o problema de ADT, no entanto, o algoritmo necessita de um conhecimento completo de

todo o ambiente e de todos os colaboradores. A obtenção desses dados em uma plataforma física é custosa e complexa e está sujeita a muitos ruídos e perdas, o que pode acarretar falhas.

2.5 Algoritmos com Outras Orientações

Nesta seção são apresentados algoritmos que mesclam diferentes estratégias citadas anteriormente e outras não citadas. Em (MENDONÇA; NEDJAH; MOURELLE, 2016) apresenta uma solução determinista, o ADTL (Alocação Dinâmica de Tarefas com abordagem Local), que propõe uma solução determinística para a alocação, onde cada robô do enxame realiza ajuste em sua tarefa mediante avaliações das demais que são executadas pelos outros robôs do enxame. O algoritmo é organizado em três etapas: atualização, ajuste e execução. A etapa de atualização consiste em receber dos demais robôs a tarefa escolhida por eles. Munido dessa informação é possível calcular a alocação atual do enxame e executar as avaliações, onde irá determinar se uma tarefa específica está com excesso de robôs, em equilíbrio ou com falta. Após a atualização executa-se a etapa de ajuste, que a partir dos resultados das avaliações, os robôs selecionam uma nova tarefa que melhor satisfaça as necessidades do enxame. Finaliza-se o ciclo executando a tarefa selecionada no ajuste e repete-se o ciclo até encontrar a alocação desejada.

O ADTL demonstra ser uma solução eficiente para o problema de ADT, porém utiliza de uma estratégia determinista, que limita suas possíveis soluções. Estratégias estocásticas possuem maior robustez, pois permitem diferentes soluções para o mesmo problema ou diferentes caminhos para um mesmo destino.

Em (BERMAN et al., 2009) apresenta uma solução escalável para alocação dinâmica de tarefas em um sistema multi-robôs homogêneos, implementando-se uma estratégia descentralizada que não requer nenhuma comunicação entre os robôs, baseando-se em uma abstração contínua do enxame de robôs, que é obtido através de uma modelagem de uma fração da população. Este define o problema de alocação de tarefas como a seleção de um robô em função de taxas de entrada e saída de uma determinada tarefa. Essas taxas são utilizadas para determinar as probabilidades que definem as políticas de controle estocástico e individual de todos os robôs do enxame, produzindo dessa forma o comportamento coletivo desejado. O algoritmo calcula as taxas em função da disposição

inicial do enxame e das restrições de comutação entre tarefas, com o objetivo de alcançar uma rápida redistribuição.

Esta estratégia demonstra bons resultados em testes com até 250 robôs, porém criar uma abstração do enxame que seja próxima da realidade necessita de uma modelagem complexa. Em um ambiente simulado, o cálculo do modelo matemático é custoso, contudo é até possível com algumas restrições forçadas e em um ambiente muito limitado. No entanto em uma plataforma física um modelo de um enxame e de um ambiente englobam muitas variáveis, tornando quase impossível uma abstração eficiente.

Em (KIM; KO, 2012) é proposto um algoritmo distribuído para alocação de tarefas de um grupo de robôs que possuem restrições energéticas e operam em um ambiente dinâmico e desconhecido. O algoritmo busca maximizar a taxa de conclusão das tarefas, minimizando o gasto dos recursos. Essa abordagem tem inspiração no bem estar geral, com objetivo de prolongar a vida útil da equipe, maximizando a economia de energia, equilibrando o consumo de recursos entre os robôs. O processo de alocação de tarefas utiliza um mecanismo baseado no mercado que consiste em um sistema de leilão. Nessa abordagem, cada robô verifica sua elegibilidade, calcula o custo para executar a tarefa e envia uma licitação. A tarefa é leiloada e alocada ao robô com menor custo, porém se existir alguma restrição de recursos, as taxas de conclusão também devem ser consideradas para a alocação.

Essa estratégia possui a necessidade de concentrar toda a informação do enxame, e a partir dessa visão geral conseguir tomar a melhor decisão. Isso cria um gestor centralizador, tornando essa estratégia menos robusta e escalável.

2.6 Considerações Finais do Capítulo

Neste capítulo foram apresentados os trabalhos recentes relacionados com o problema de ADT. Esses algoritmos comprovam a relevância dessa pesquisa, demonstrando o grande potencial desse segmento acadêmico. Cada algoritmo possui abordagens distintas com suas particularidades, pontos fortes e fracos, o que comprova ser uma área rica e ainda com muitas possibilidades.

No próximo capítulo é apresentada uma abordagem estocástica, distribuída e cooperativa para a ADT, com o objetivo de otimizar o processo de comunicação. Neste

é proposto um algoritmo, com inspiração no PSO, para realizar a alocação dinâmica de tarefas em um enxame de robôs homogêneos. A solução do ADT é encontrada de forma totalmente distribuída, com base na inspiração do PSO cada robô representa uma partícula, onde dessa forma todo o enxame encontrará cooperativamente a solução do ADT, utilizando para isso uma comunicação otimizada.

Capítulo 3

ALOCAÇÃO DINÂMICA DE TAREFAS EM CLUSTER

NESTE capítulo propõe-se uma solução para o problema de ADT, utilizando uma decisão estocástica, um controle distribuído, uma busca cooperativa e uma topologia de comunicação em cluster, que será chamado de ADTC (Alocação Dinâmica de Tarefas em Cluster). A decisão estocástica permite solucionar o problema de ADT com flexibilidade nas soluções, pois é possível o encontro de diferentes soluções válidas. O controle distribuído juntamente com a busca cooperativa permite a exploração do paralelismo, que é uma poderosa metodologia, pois viabiliza a solução de problemas complexos e minimiza a possibilidade de falhas, como demonstrado no capítulo anterior. O algoritmo de otimização baseado em enxame de partículas (PSO) é utilizado como inspiração para a minimização da função objetivo idealizada.

O objetivo desse trabalho, conforme explicado na introdução, é a otimização da comunicação, com o intuito de melhorar os resultados obtidos no trabalho apresentado em (MENDONÇA, 2014). A topologia de comunicação em malha completa é utilizada no trabalho (MENDONÇA, 2014). Esse tipo de topologia gera um crescimento exponencial do fluxo de informação em função do número de robôs, o que pode ser um problema para grandes enxames.

A comunicação está presente em todos os processos de interação entre dois ou mais indivíduos com um objetivo comum. A forma que é codificada e utilizada impacta no desempenho e nos resultados obtidos. A topologia de comunicação em cluster foi escolhida porque possui grande capacidade de otimização da comunicação, além de ser a mais semelhante à malha completa.

Desta forma, este capítulo é estruturado da seguinte forma: a Seção 3.1 apresenta uma introdução à otimização por enxame de partículas; a Seção 3.2 descreve a dinâmica de execução do algoritmo de otimização ADTC e detalha as suas diferentes etapas; e a Seção 3.3 apresenta as considerações finais do capítulo.

3.1 Otimização Baseada em Enxame de Partículas

O PSO é um algoritmo de otimização estocástico baseado na inteligência coletiva proposto em (EBERHART; KENNEDY, 1995). O PSO é um algoritmo distribuído para otimizar funções objetivo, utilizado normalmente em funções não deriváveis, não lineares ou com elevado número de dimensões. Teve como inspiração inicial o padrão de voo dos pássaros em busca de abrigo e alimento, demonstrado no trabalho (REYNOLDS, 1987). Posteriormente observou-se a possibilidade de adaptar o comportamento social dos pássaros para um processo de busca, os pássaros viraram partículas e a primeira versão do PSO é criada em 1995.

O algoritmo PSO é um algoritmo estocástico que gerencia um grupo de partículas, que de forma cooperativa, buscam uma solução em um espaço de busca predeterminado. Cada partícula é iniciada com uma posição inicial aleatória e uma velocidade inicial aleatória, onde cada uma das partículas é uma solução em potencial. A cada interação a partícula se locomove no espaço de busca, adaptando-se em função de sua própria experiência e da experiência do grupo.

3.1.1 Visão Geral

Nessa seção é apresentada uma visão geral do PSO, como mostra o Algoritmo 1. Primeiramente é realizada a inicialização, que consiste em distribuir todas as partículas em posições aleatórias no espaço de busca. Após a alocação é calculada a qualidade de cada partícula em relação ao sua posição no espaço de busca, quanto mais próximo de uma solução ótima melhor é sua qualidade. A próxima etapa é comparar o resultado obtido com a melhor qualidade encontrada em seu passado. Essa melhor qualidade é chamada de *Pbest*, que resumindo consiste na melhor posição alcançada até o presente momento. A cada ciclo a qualidade *Pbest* é atualizada sempre que uma nova posição obtiver uma melhor qualidade. A etapa seguinte é a identificação da partícula *Gbest*, que consiste

na partícula com a melhor posição global. Encontra-se a partícula *Gbest* comparando-se todas as qualidades *Pbests*, a partícula que possuir a melhor qualidade *Pbest* é declarada a partícula *Gbest*.

Algoritmo 1 Algoritmo PSO

inicialização gera posições e velocidades aleatórias;

Repita

Atualizar *Pbest*; //melhor posição da partícula

Identificar a partícula *Gbest*; //melhor posição do enxame

Para cada partícula *i* **Faça**

Para cada dimensão *j* **Faça**

Atualizar a velocidade da partícula *i* na dimensão *j* (v_{ij}) utilizando a Equação 10;

Controlar velocidade;

$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$;

Confinar a partícula

Fim Para

Fim Para

Até Critério de parada;

Após o término da atualização da qualidade *Pbest* e da identificação da partícula *Gbest*, inicia-se a última etapa, na qual será calculada a nova velocidade e posição de todas as partículas. A nova velocidade é influenciada pela velocidade atual, pela experiência individual da partícula e pela experiência coletiva do enxame, como mostra a Figura 5. A velocidade é composta por três elementos: a inércia, o cognitivo e o social. A inércia é o fator que considera a velocidade atual da partícula, prevenindo mudanças drásticas de direção. O cognitivo é o fator que insere uma tendência para direcionar a partícula para sua melhor posição até agora. Por último, o social é o fator que insere uma tendência para direcionar a partícula para a melhor posição do enxame.

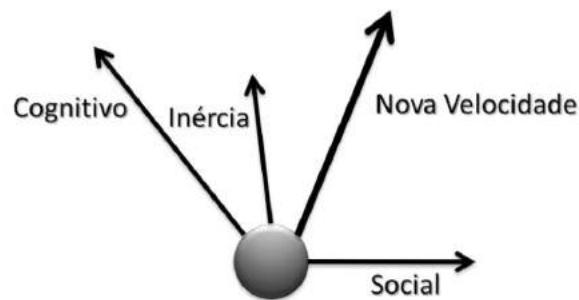


Figura 5: Composição da nova velocidade

A velocidade é a soma desses três componentes - a inércia, o cognitivo e o social - como mostra a Equação 10:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(y_{ij} - x_{ij}(t)) + c_2r_{2j}(t)(\hat{y}_j - x_{ij}(t)), \quad (10)$$

onde o $v_{ij}(t+1)$ é a velocidade atualizada da partícula i na dimensão j . As constantes w , c_1 e c_2 são denominadas de coeficientes de inércia, cognitivo e social; o r_{1j} e o r_{2j} são valores aleatórios dinâmicos gerados a cada ciclo, com distribuição uniforme no intervalo $[0, 1]$; o y_{ij} é a coordenada da qualidade $Pbest$ atingida pela partícula i na dimensão j ; o \hat{y}_j é a coordenada do robô $Gbest$ na dimensão j ; o $x_{ij}(t)$ e o $v_{ij}(t)$ são a posição e velocidade atuais da partícula i na dimensão j .

As constantes w , c_1 e c_2 são pesos para os fatores que compõem a velocidade e possuem um grande impacto nos resultados. Esses coeficientes podem ser determinados de uma forma empírica com experimentação, mas pode-se fazer uma análise sobre a influência de cada fator no comportamento da partícula. O fator da inércia leva em consideração a velocidade atual no cálculo da próxima velocidade, aumentando a estabilidade do movimento da partícula, o que ajuda a ultrapassar barreiras, como máximos locais. O fator cognitivo restringe a navegação da partícula, impondo que ela navegue em regiões próximas, fazendo com o que a partícula realize uma busca mais minuciosa do espaço. Por fim, o fator social aglomera o enxame em uma determinada região, limitando a exploração do ambiente e aumentando a chance de estagnação, porém é o fator que mais impõem uma característica de evolução ao enxame.

Após a atualização da velocidade é realizado o controle da velocidade da partícula, com o objetivo de evitar que a partícula dê grandes saltos no espaço de busca. Depois do controle da velocidade, atualiza-se a posição da partícula. Nesse momento é realizado o confinamento para garantir que a partícula esteja dentro dos limites do espaço de busca. A Figura 6 apresenta o comportamento do PSO aplicado na minimização do função Rastrigin, apresentada na Figura 7¹. Pode-se observa que existe uma convergência da maioria das partículas para a região de mínimo da função. Nesse exemplo é utilizado um enxame de 10 partículas.

¹Figuras retiradas da página https://en.wikipedia.org/wiki/Rastrigin_function

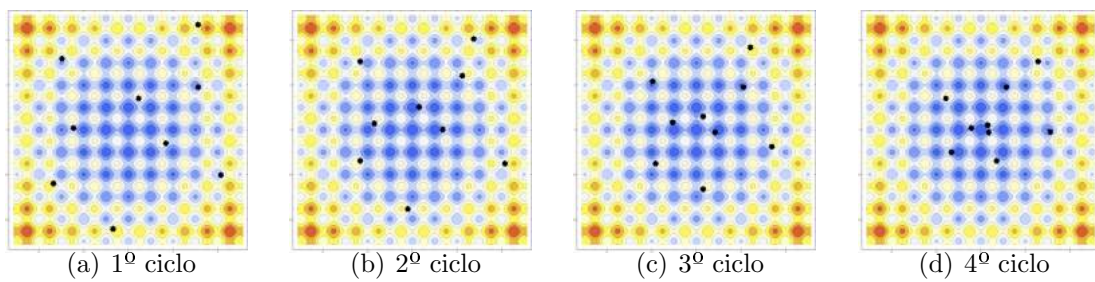


Figura 6: PSO aplicado na minimização do função Rastrigin

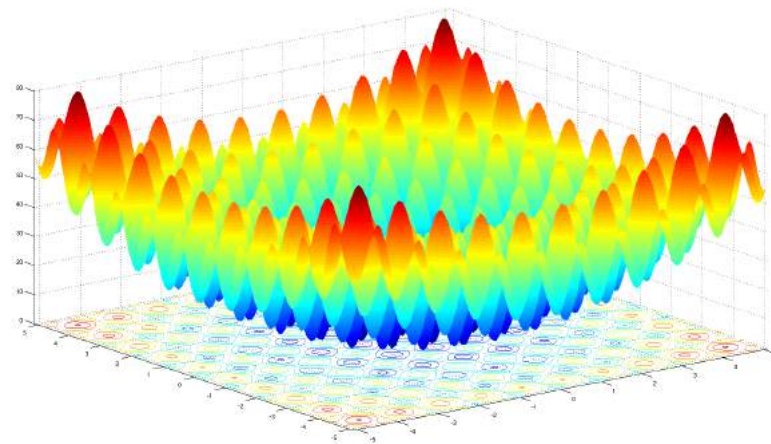


Figura 7: Função Rastrigin

3.1.2 Topologias de Comunicação

A comunicação tem como objetivo trazer harmonia a uma comunidade, evitar desperdício de material, trabalho e tempo. É clara sua importância nas diferentes esferas sociais, gerando, desta forma, grande impacto no resultado final, em função de sua intensidade e complexidade empregada. Pequenos erros de interpretação podem gerar grandes prejuízos.

A robótica de enxame consiste em um grupo de robôs autônomos, não sincronizados e com inteligência limitada que cooperam para alcançarem tarefas globais (BENI; WANG, 1993). A cooperação entre os robôs exige algum tipo de comunicação. A maneira como ela é utilizada pode causar grande impacto do desempenho do projeto.

É possível classificar dois extremos em relação ao fluxo de comunicação, que são classificados como estratégia centralizada ou descentralizada. Estratégias totalmente centralizadas são soluções sem nenhum tipo de comunicação, onde o robô irá operar de uma forma totalmente individual, limitando a comunicação apenas a sinalização de uma possível solução para os demais membros do grupo. Já as estratégias totalmente descen-

tralizadas, são implementações que exigem uma completa cooperação entre os robôs, onde cada um é uma pequena parte de uma estrutura que toma decisões como um todo para encontrar uma solução. Este tipo de estratégia possui um grande fluxo de informação entre os robôs.

Entre os extremos podem-se classificar alguns tipos de topologia de comunicação, uma estrutura que conecta cada um dos indivíduos de um grupo. Existem diferentes formas de organizar essa interligação, onde cada tipo de estrutura possui características que vão impactar o desempenho do projeto. Podemos citar cinco tipos de estruturas mais usuais (GROTH; SKANDIER, 2006; SOSINSKY, 2009):

- **Topologia em anel:** os robôs são conectados em série formando um circuito fechado (anel), como mostra a Figura 8 (a). A informação flui de forma unidirecional, passando de robô à robô até o destino. Possui baixo fluxo de informação, pois cada robô possui apenas duas conexões, porém com grandes rotas, do remetente ao destinatário, que acarretam atrasos na transmissão. A cada robô inserido no enxame, há um aumento no tempo de transmissão. A confiabilidade está inversamente proporcional ao número de robôs no grupo, de forma que quanto maior o grupo, menor é a confiança de que a mensagem chegará ao destino corretamente.

- **Topologia em malha:** permite que um robô conecte-se a qualquer número de robôs, podendo até se conectar a todos os demais do grupo, como mostra a Figura 8 (b). Ao transmitir uma mensagem de um robô a outro por diferentes caminhos, esta topologia torna-se bastante tolerante a falhas. A densidade do fluxo de informação está diretamente proporcional ao número de conexões entre os robôs.

- **Topologia em estrela:** é designado um robô central pelo qual toda a informação terá que passar obrigatoriamente, sendo ele responsável por conectar todos demais robôs do enxame. Essa topologia é caracterizada por um elemento central que gerencia o fluxo de informação, ligado ponto-a-ponto a cada robô do enxame, como mostra a Figura 8 (c). Toda informação de um robô a outro, passa primeiramente pelo robô central, que encarrega-se de enviar a informação somente para o destino. O tempo de transmissão está em função da capacidade do robô central em receber e transmitir a informação. E a confiabilidade dependente da qualidade do robô central.

- **Topologia em cluster:** é uma adaptação do topologia em malha, dividindo o

enxame em subgrupos menores, com a intenção de diminuir o tempo de transmissão. Em cada subgrupo é implementada a topologia em malha independente dos outros subgrupos, que são conectados por pontos de acessos, como mostra a Figura 8 (d). Essa topologia não possui regras que limitam o número de clusters, quantos robôs devem conter cada subgrupo ou número de pontos de acesso. É uma topologia flexível, que se adapta a diferentes aplicações.

- A **Topologia em árvore**: possui uma estrutura formada por ramos, onde a mensagem é transmitida do pai para os filhos, com mostra o Figura 8 (e), possuindo baixa densidade de fluxo de informação, pois cada robô possui poucas conexões. Estruturas em árvore possuem grande flexibilidade, possibilitando modificar ou remover um ramo sem causar perturbação no restante do enxame. A implementação é complexa, pelo fato dos ramos poderem apresentar características diferentes um dos outros, podendo causar diferentes velocidade de transmissão e, conseqüentemente, problemas de dessincronização.

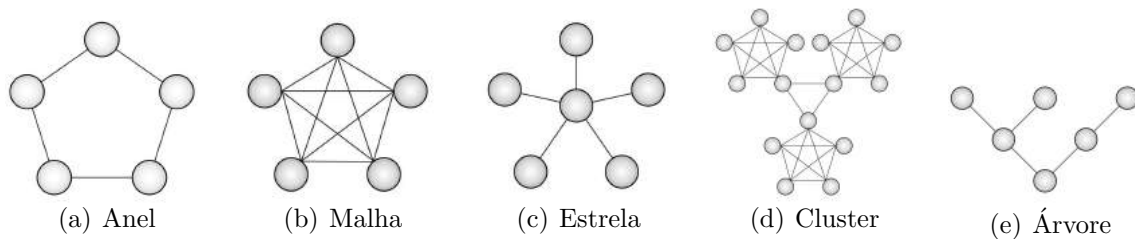


Figura 8: Topologias de comunicação

3.1.3 Cluster

A topologia de comunicação em cluster é formada por subgrupos gerados pela divisão do enxame, formando os clusters. A comunicação é limitada pelos clusters, ou seja, um robô não pode comunicar-se com outro robô que não seja do seu cluster. A comunicação entre os cluster é realizada apenas pelos informantes, que são agentes especiais responsáveis por compartilhar as informações adquiridas dentro dos clusters. A Figura 9 apresenta topologias de comunicação que exemplificam as diferentes estruturas permitidas na topologia em cluster, onde os nós brancos representam os agentes informantes.

O processo de comunicação é dividido em três etapas: a primeira etapa é semelhante à comunicação em malha, na qual a informação é compartilhada dentro de cada cluster, como ilustra a Figura 10(a), onde nessa etapa é identificado o robô que possui

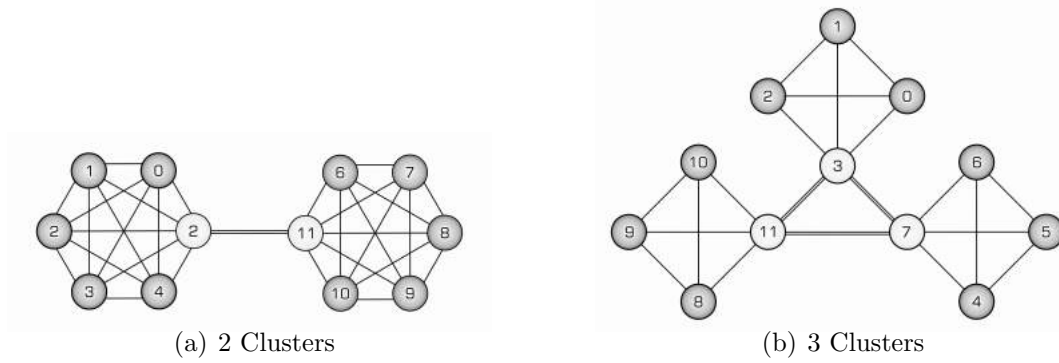


Figura 9: Diferentes topologias em cluster com 12 robôs

a melhor solução no cluster, o robô C_{best} ; a segunda etapa é uma comunicação especial realizada somente entre os informantes, que vão enviar e receber as melhores soluções encontradas em cada cluster e determinar, nesse momento, o robô que possui a melhor solução de todo o enxame o robô G_{best} , demonstrado pela Figura 10(b); e a última etapa, é a comunicação que permite ao informante enviar aos membros do seu cluster a melhor alocação $A_{G_{best}}$, como apresenta a Figura 10(c).

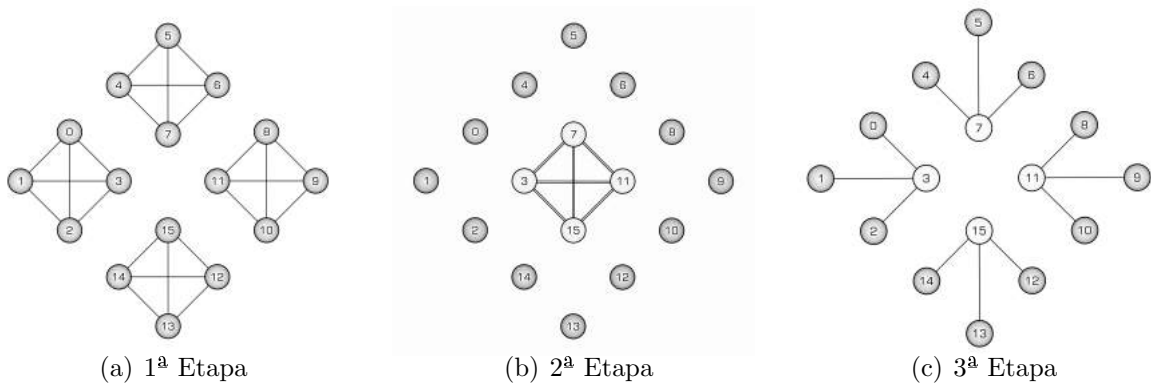


Figura 10: Etapas da comunicação em cluster

Nessa implementação a topologia de comunicação em cluster possui alguns restrições: todos os clusters possuem o mesmo número de robôs; tanto a comunicação dentro dos clusters como também a comunicação entre os informantes é em malha completa. Com o objetivo de mensurar a capacidade de redução do fluxo de informação, é proposta uma fórmula matemática para a topologia de comunicação em cluster com as restrições acima citadas. Essa fórmula demonstra o comportamento do número de mensagens enviadas por ciclo em função da variação do número de robôs ρ , do número de clusters γ e do número de informantes por cluster σ . A função \mathbb{F}_c é o número total de mensagens enviadas, que

consiste no somatório do número de mensagens utilizadas nas 3 etapas apresentadas na Figura 11, como mostra a Equação 11:

$$\mathbb{F}_c = F_1 + F_2 + F_3 \quad (11)$$

onde os termos F_1 , F_2 e F_3 representam respectivamente, ao número de mensagens utilizadas na 1ª, 2ª e 3ª etapas. O termo F_1 está em função do número de robôs por cluster $\rho_c = \frac{\rho}{\gamma}$ e do número de clusters γ , como apresenta a Equação 12:

$$F_1 = \gamma(\rho_c(\rho_c - 1)). \quad (12)$$

O termo F_2 está em função do número de informantes por cluster σ e pelo número de cluster γ , como demonstrado pela Equação 13:

$$F_2 = \gamma\sigma(\gamma\sigma - 1). \quad (13)$$

Por último, o termo F_3 está em função do número de robôs por cluster ρ_c , do número de clusters γ e do número de informantes por cluster σ , porém apenas os informantes enviam mensagens para os demais robôs do cluster, como mostra Equação 14:

$$F_3 = \gamma\sigma(\rho_c - 1). \quad (14)$$

Realizando as devidas substituições na Equação 11 e considerando $\sigma = 1$, \mathbb{F}_c fica apenas em função do número de robôs ρ e do número de clusters γ , como apresenta a Equação 15:

$$\mathbb{F}_c = \gamma^2 + \frac{\rho^2}{\gamma} - 2\gamma. \quad (15)$$

Com o objetivo de exemplificar o impacto do tipo de topologia no desempenho de um projeto, consideram-se um enxame com 12 robôs e 2 tarefa a serem alocadas, com uma proporção-objetivo $\mathbb{P} = \{\frac{4}{12}, \frac{8}{12}\}$. São considerados dois tipos de topologias, sendo a primeira de malha completa, onde todos os robôs comunicam-se com todos os demais e a segunda é em cluster, no qual o enxame é dividido em dois clusters, como mostra a Figura 11.

A função \mathbb{F}_m é o número de mensagens enviadas por ciclo na topologia em malha completa. O cálculo do número de mensagens enviadas é generalizado para qualquer tamanho de enxame que utilize a topologia em malha completa, como mostra a Equação 16:

$$\mathbb{F}_m = \rho(\rho - 1). \quad (16)$$

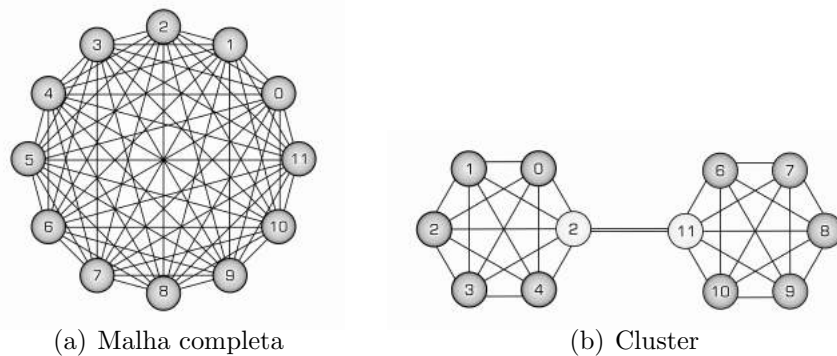


Figura 11: Topologias de comunicação para um enxame de 12 robôs

O problema consiste em encontrar o robô que possui a melhor alocação do enxame, o robô G_{best} . Para que os robôs possam identificar o robô G_{best} , devem conhecer as qualidades de todos os demais robôs do enxame $f(\mathbb{A})$. Na topologia em malha completa, todos os robôs enviam sua própria qualidade para todos os outros. Em um enxame de 12 robôs, cada um deverá enviar 11 mensagens, totalizando 132 mensagens ao todo. Na topologia em clusters, como já mencionado, são necessárias três etapas de comunicação: a primeira etapa terá um total de 30 mensagens em cada cluster, onde cada cluster possui 6 robôs e cada robô envia 5 mensagens; a segunda, 2 mensagens (uma por cada informante); e a terceira, 5 mensagens em cada cluster, enviadas pelo informante. Com um total de mensagens igual a 72.

Nesse exemplo houve uma redução do número de mensagens em 45,4%. Esses dados são baseados em uma simplificação, pois é desconsiderado na Equação 15 as mensagens de controle utilizadas pelo algoritmo. No entanto, já demonstra que a modificação da topologia de comunicação pode otimizar a comunicação. Na Seção 4.2 é apresentado a Equação 22 e a Equação 23 que definem de forma mais completa o número de mensagens enviadas.

A topologia de comunicação em cluster possui etapas de comunicação com características bem definidas. Cada etapa possui uma organização dos robôs e uma estrutura de comunicação que deve ser respeitada por todos os robôs de enxame. A matriz de comunicação \mathbb{M} estabelecerá a organização dos robôs e a estrutura da topologia de comunicação, definindo em qual cluster cada robô é integrante, quantos clusters existem, quantos robôs há em cada cluster e quais robôs são informantes. A Figura 12 apresenta uma Matriz de Comunicação e sua topologia de comunicação em cluster equivalente.

	R0	R1	R2	R3	R4	R5	R6	R7	R8
R0		1	1						
R1	1		1						
R2	1	1				9			9
R3					2	2			
R4				2		2			
R5			9	2	2				9
R6								3	3
R7							3		3
R8			9			9	3	3	

(a) Matriz

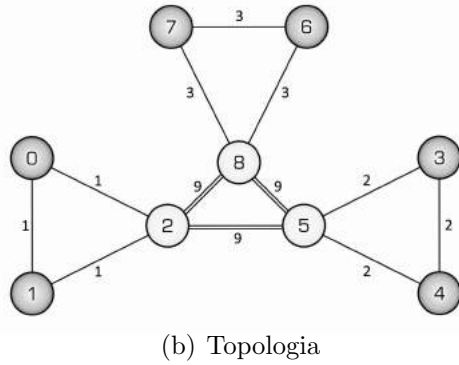


Figura 12: Organização da comunicação em clusters de um enxame de 9 robôs

3.2 Algoritmo Proposto

Com o objetivo de otimizar os resultados apresentados em (MENDONÇA, 2014), o algoritmo para Alocação Dinâmica de Tarefas em Cluster (ADTC) foi elaborado. O ADTC realiza a alocação de tarefas de forma dinâmica, distribuída, cooperativa e estocástica. O algoritmo é inspirado pelo PSO e tem como objetivo minimizar a função objetivo do ADT, apresentado detalhadamente no Capítulo 1.

3.2.1 Visão Geral

O algoritmo ADTC tem como objetivo alocar τ tarefas previamente definidas à um grupo de ρ robôs, satisfazendo uma proporção-objetivo \mathbb{P} . Este utiliza como inspiração a técnica de otimização PSO, porém de forma totalmente distribuída, onde cada partícula do PSO é representada fisicamente por um robô.

O algoritmo é dividido em cinco etapas: inicialização, ajuste, identificação, atualização e critério de parada. A Figura 13 apresenta o fluxograma com a dinâmica do algoritmo ADTC.

Na primeira etapa o robô é inicializado, gerando-se todas as variáveis necessárias para a execução do ADTC. A segunda etapa é a de ajuste, que tem como objetivo principal a identificação do robô C_{best} e atualização de sua alocação $\mathbb{A}_{C_{best}}$. A terceira etapa é a atualização, tem como objetivo identificar a alocação que possui a melhor qualidade do enxame G_{best} e compartilhar, a mesma, entre todos os informantes. A quarta etapa é a atualização, nessa etapa os informantes enviam para os demais robôs do cluster a alocação da melhor solução global $\mathbb{A}_{G_{best}}$.

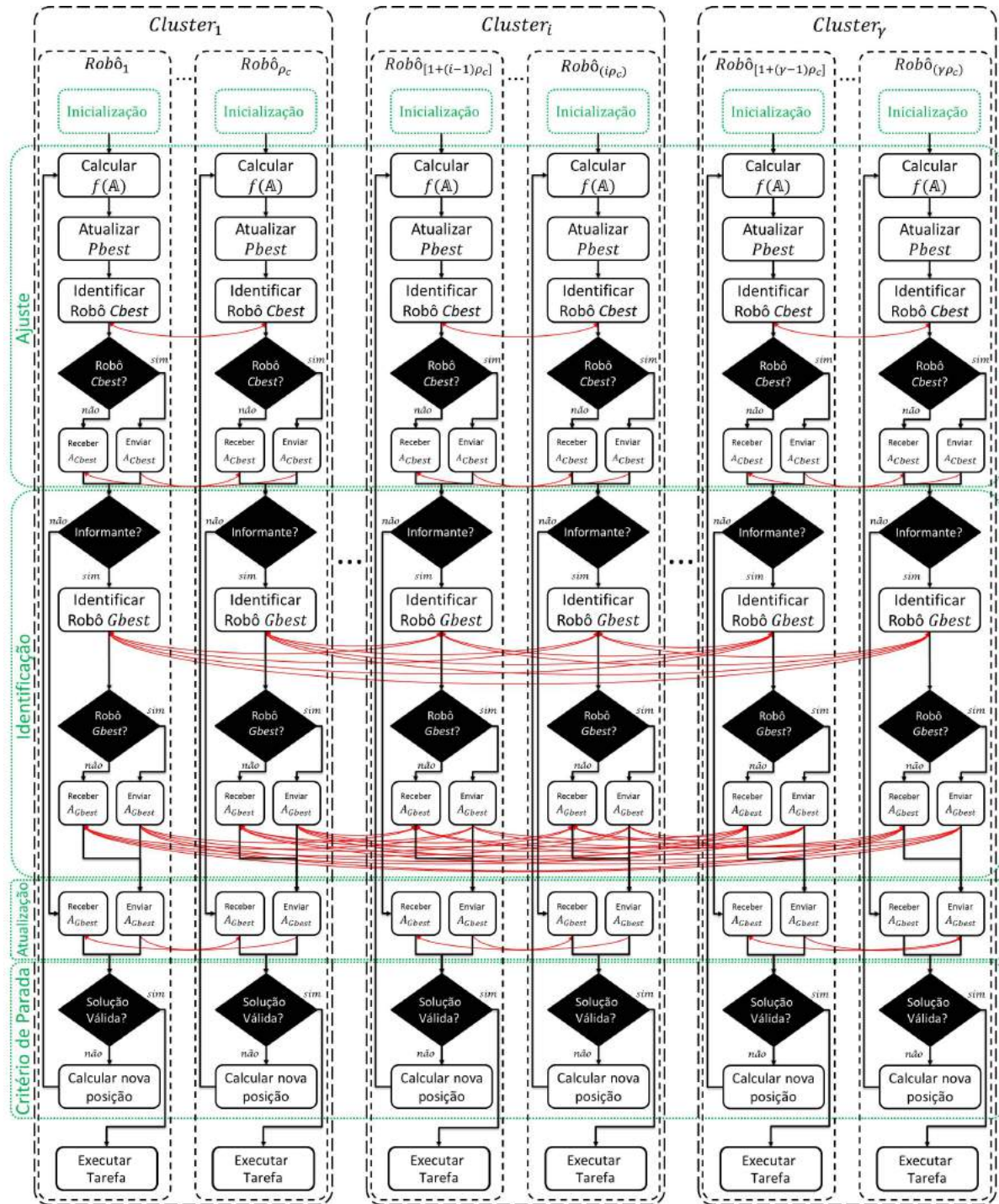


Figura 13: Fluxograma do algoritmo ADTC

A última etapa é o critério de parada, que define quando o ADTC finaliza a busca por uma solução. Esse processo irá avaliar se a alocação encontrada ($A_{G_{best}}$) é satisfatória, ou seja, atente as necessidades da proporção-objetivo. Caso seja satisfatória todo o enxame é autorizado a executar as respectivas tarefas. Caso a solução não seja satisfatória uma nova posição é calculada e um novo ciclo é inicializado, onde todo o

enxame é encaminhado para a etapa de ajuste. Na Seção 3.2.2 é apresentado com maiores detalhes todas as etapas mencionadas.

3.2.2 Etapas do Algoritmo

O Algoritmo 2 demonstra como é realizado a conexão entre as etapas e como é feita a dinâmica entre elas. Sua dinâmica possibilita que os robôs repitam as etapas inúmeras vezes com o objetivo de encontrar uma solução adequada para o problema de ADT.

Algoritmo 2 Algoritmo ADTC

- 1: Inicialização
 - 2: **Repita**
 - 3: Ajuste;
 - 4: **Se** for informante **Então**
 - 5: Identificação;
 - 6: **Fim Se**
 - 7: Atualização;
 - 8: Critério de parada;
 - 9: **Até** encontrar uma solução válida;
 - 10: Executar tarefa
-

3.2.2.1 Inicialização

Na etapa de inicialização é realizada a configuração inicial do ADTC, no qual possui como parâmetro de entrada: o número de robôs ρ , o número de tarefas τ ; a proporção-objetivo \mathbb{P} e a matriz de comunicação \mathbb{M} . Estes são utilizados para gerar as seguintes variáveis: a alocação atual do robô \mathbb{A}_i ; a alocação da melhor qualidade do robô \mathbb{A}_{Pbest_i} ; a alocação da melhor alocação do clusters \mathbb{A}_{Cbest_i} ; a alocação da melhor alocação do enxame \mathbb{A}_{Gbest} ; o contador de tarefas ideal \mathbb{C} ; o contador de tarefas atual $\mathbb{C}_{\mathbb{A}_i}$ e a velocidade atual v_i . Como demonstrado no Algoritmo 3.

Algoritmo 3 Inicialização

Entrada ρ, τ, \mathbb{P} e \mathbb{M} ;

Saída $\mathbb{A}_i, \mathbb{A}_{Pbest_i}, \mathbb{A}_{Cbest_i}, \mathbb{A}_{Gbest}, \mathbb{C}, \mathbb{C}_{\mathbb{A}_i}$ e v_i ;

- 1: Gera randomicamente o \mathbb{A}_i e v_i ;
 - 2: Calcula o \mathbb{C} e o $\mathbb{C}_{\mathbb{A}_i}$;
 - 3: $\mathbb{A}_{Pbest_i} := \mathbb{A}_i$;
 - 4: $\mathbb{A}_{Cbest_i} := \mathbb{A}_i$;
 - 5: $\mathbb{A}_{Gbest} := \mathbb{A}_i$;
-

A alocação \mathbb{A}_i representa a posição no espaço de busca do robô i . A alocação $\mathbb{A}_i = \{a_1, a_2, \dots, a_\rho\}$ é um vetor de ρ dimensões, onde cada posição do vetor representa a alocação de um determinado robô. A tarefa a ser executada pelo robô₂, por exemplo, está representada na alocação \mathbb{A}_i na dimensão 2 a_2 . Cada dimensão da alocação \mathbb{A}_i possui um intervalo de busca formado por inteiros positivos em um intervalo $[1, \dots, \tau]$, que representa todas as possíveis tarefas a serem alocadas. Note que nesse momento todas as melhores alocações (\mathbb{A}_{Pbest_i} , \mathbb{A}_{Cbest_i} e \mathbb{A}_{Gbest}) são inicializadas como a alocação atual. Os contadores \mathbb{C} e $\mathbb{C}_{\mathbb{A}_i}$ representam o número de robôs alocados em cada tarefa. O contador de tarefas ideal \mathbb{C} depende da proporção-objetivo \mathbb{P} e do número de robôs ρ , tem como finalidade definir qual é o número ideal de robôs em cada tarefa, onde é utilizado pelo função objetivo para determinar a qualidade de uma alocação. O contador de tarefas atual $\mathbb{C}_{\mathbb{A}_i}$ representa o número atual de robôs alocados em cada tarefa, também é utilizado para definir a qualidade da alocação atual. Por fim, a velocidade atual v_i é um vetor com ρ dimensão que é utilizada para o cálculo de uma nova alocação.

3.2.2.2 Ajuste

Na etapa de ajuste é realizada a atualização da melhor alocação no cluster $\mathbb{A}_{Cbest_\gamma}$, conforme demonstrado pelo Algoritmo 4. A alocação \mathbb{A}_{Pbest_i} é a melhor alocação já alcançada pelo robô i , para sua atualização é necessário o cálculo da qualidade da alocação atual $f(\mathbb{A}_i)$, caso a qualidade atual seja melhor, a qualidade $Pbest$ é substituído pela qualidade da alocação atual e a alocação \mathbb{A}_{Pbest_i} é substituída pela alocação atual \mathbb{A}_i .

Depois da atualização da qualidade $Pbest$, realiza-se a localização do robô $Cbest$. Nesse momento inicia-se o primeiro processo de comunicação entre os robôs, onde é definido como o robô $Cbest$ aquele que possui a melhor qualidade $Pbest$ dentro do cluster. A etapa de ajuste é independente em relação aos clusters, ou seja, ele pode ser realizado em paralelo em todos os clusters. Não existe nesse momento comunicação ou dependência entre os clusters.

Para a localização do robô $Cbest$ é realizada uma troca de informação entre os robôs, onde cada robô envia para os demais robôs do seu cluster o seu identificador id_i e a sua qualidade $Pbest_i$. Todos os robôs do cluster recebem as mesmas informações dos demais, dessa forma ao finalizar o processo de comunicação são capazes de identificar o robô $Cbest$. O processo de comunicação consiste no robô i , que é integrante do cluster γ , enviar

Algoritmo 4 Atualizar melhor alocação no cluster**Entrada** $\rho, \mathbb{A}_i, \mathbb{A}_{Pbest_i}$ e \mathbb{M} ;**Saída** $id_{Cbest}, Cbest$ e \mathbb{A}_{Cbest} ;

- 1: **Se** $f(\mathbb{A}_i) \leq Pbest_i$ **Então**
- 2: $Pbest_i := f(\mathbb{A}_i)$;
- 3: $\mathbb{A}_{Pbest_i} := \mathbb{A}_i$;
- 4: **Fim Se**
- 5: $msg \leftarrow \langle id_i, Pbest_i \rangle$;
- 6: **Repita**
- 7: **Enviar** msg aos demais robôs do cluster;
- 8: **Receber** msg dos demais robôs do cluster;
- 9: $\mathbb{V}[i] := Pbest_i$;
- 10: **Até** receber e enviar de/para todos os robôs
- 11: **identificar** robô $Cbest$ (id_{Cbest} e $Cbest$);
- 12: $msg1 \leftarrow \langle id_i, \mathbb{A}_i \rangle$;
- 13: $msg2 \leftarrow \langle id_i \rangle$;
- 14: **Repita**
- 15: **Se** $id_i = id_{Cbest}$ **Então**
- 16: **Enviar** $msg1$ aos demais robôs do cluster;
- 17: **Receber** $msg2$ dos demais robôs do cluster;
- 18: **Fim Se**
- 19: **Se** $id_i \neq id_{Cbest}$ **Então**
- 20: **Receber** $msg1$ do robô id_{Cbest} ;
- 21: **Enviar** $msg2$ ao robô id_{Cbest} ;
- 22: $\mathbb{A}_{Cbest} := \mathbb{A}_i$; (substitui a alocação \mathbb{A}_{Cbest} pela alocação recebida)
- 23: **Fim Se**
- 24: **Até** que todos tenham recebido

mensagens com o seu id_i e a sua qualidade $Pbest_i$, para todos os demais integrantes do cluster. De forma dinâmica, mensagens são trocadas entre os robôs: ao mesmo tempo em que o robô i está enviando mensagens também está recebendo mensagens semelhantes. À medida que recebe as mensagens dos demais robôs, o vetor $\mathbb{V} = \{Pbest_1, Pbest_2, \dots, Pbest_\rho\}$ é preenchido e ao concluir o processo, o vetor estará completo. Todos os robôs do cluster, nesse momento, possuem o mesmo vetor \mathbb{V} preenchido com as mesmas informações.

Com o vetor \mathbb{V} atualizado inicia-se a identificação do robô $Cbest$. Nesse ponto existem dois processos a ser executados ao mesmo tempo. O primeiro é o envio da alocação do robô $Cbest$, que deve ser realizado pelo robô $Cbest$; e o segundo, o recebimento da alocação pelos robôs não classificados como $Cbest$. O algoritmo realiza a classificação dos robôs do cluster, para que o robô $Cbest$ envie a alocação e os demais recebam.

3.2.2.3 Identificação

Na etapa de identificação é realizada a localização do robô $Gbest$, que é o robô que possui a melhor alocação global. No processo anterior, foi realizada uma busca pelo robô $Cbest$, limitando a comunicação dentro dos clusters. Nesse ponto todos os robôs já identificaram o robô $Cbest$ em seus clusters e atualizaram a melhor alocação no cluster \mathbb{A}_{Cbest} . Para a identificação do robô $Gbest$ é necessário comparar todas as melhores alocações encontradas nos clusters. Esta etapa é somente realizada pelos robôs classificados como informantes, que são os pontos de acesso dos clusters.

Dentre os robôs do cluster são selecionados os que são os informantes. São eles que possuem a responsabilidade de trocar informações entre os clusters, com o objetivo de localizar o robô $Gbest$. Para que ocorra a identificação, o vetor $\mathbb{V}^* = \{Cbest_1, Cbest_2, \dots, Cbest_\gamma\}$ deve ser preenchido com os $Cbests$ atualizados, que foram encontrados nos clusters na etapa anterior. Toda a dinâmica da etapa de identificação é apresentado no Algoritmo 5.

Algoritmo 5 Identificação do robô $Gbest$

Entrada $\rho, Cbest, \mathbb{A}_{Cbest}$ e \mathbb{M} ;

Saída $Gbest$ e \mathbb{A}_{Gbest} ;

- 1: $msg3 \leftarrow \langle id_i, Gbest \rangle$;
 - 2: **Repita**
 - 3: **Enviar** $msg3$ aos demais robôs informantes;
 - 4: **Receber** $msg3$ dos demais robôs informantes;
 - 5: $\mathbb{V}^*[i] := Cbest_i$;
 - 6: **Até** receber e enviar de/para todos os informantes
 - 7: **identificar** robô $Gbest$ (id_{Gbest} e $Gbest$);
 - 8: $msg4 \leftarrow \langle id_i, \mathbb{A}_{Cbest} \rangle$;
 - 9: $msg5 \leftarrow \langle id_i \rangle$;
 - 10: **Repita**
 - 11: **Se** $id_i = id_{Gbest}$ **Então**
 - 12: **Enviar** $msg4$ aos demais robôs informantes;
 - 13: **Receber** $msg5$ dos demais robôs informantes;
 - 14: **Fim Se**
 - 15: **Se** $id_i \neq id_{Gbest}$ **Então**
 - 16: **Enviar** $msg5$ ao robô id_{Gbest} ;
 - 17: **Receber** $msg4$ do robô id_{Gbest} ;
 - 18: $\mathbb{A}_{Gbest} := \mathbb{A}_{Cbest}$; (substitui a alocação \mathbb{A}_{Gbest} pela alocação recebida)
 - 19: **Fim Se**
 - 20: **Até** que todos os informantes tenham recebido
-

Os robôs informantes trocam mensagens dinamicamente, ao mesmo tempo em que enviam também recebem mensagens semelhantes. Esse processo é muito similar ao

utilizado para a identificação do robô C_{best} . À medida que recebem as mensagens dos demais informantes, o vetor \mathbb{V}^* é preenchido com os C_{best} s encontrados nos clusters. Ao final do processo, o vetor \mathbb{V}^* deve estar totalmente preenchido. Todos os informantes possuem, nesse momento, um vetor \mathbb{V}^* igual, com as mesmas informações. O robô G_{best} é aquele que possui a alocação com a melhor qualidade, semelhante a etapa de ajuste. Nesse momento os informantes são segregados em dois grupos, um contendo o robô G_{best} e o outro com os demais. Após a segregação, o robô G_{best} envia a melhor alocação do seu cluster $\mathbb{A}_{C_{best}}$ aos demais informantes.

3.2.2.4 Atualização

Na etapa de atualização, conforme apresenta o Algoritmo 6, os informantes enviam mensagens para os demais robôs do seu cluster contendo o identificador $id_{G_{best}}$ e a alocação $\mathbb{A}_{G_{best}}$. Essas mensagens contêm a identificação e a alocação do robô G_{best} . Os robôs ao receberem a mensagem do informante substituem a alocação $\mathbb{A}_{G_{best}}$ antiga pela alocação $\mathbb{A}_{G_{best}}$ atualizada.

Algoritmo 6 Atualizar melhor alocação global

Entrada ρ , G_{best} , $\mathbb{A}_{G_{best}}$ e \mathbb{M} ;

Saída G_{best} e $\mathbb{A}_{G_{best}}$;

- 1: **Repita**
 - 2: $msg6 \leftarrow \langle id_i, \mathbb{A}_{G_{best}} \rangle$;
 - 3: $msg7 \leftarrow \langle id_i \rangle$;
 - 4: **Se** for informante **Então**
 - 5: **Enviar** $msg6$ aos demais robôs do cluster;
 - 6: **Receber** $msg7$ dos demais robôs do cluster;
 - 7: **Senão**
 - 8: **Enviar** $msg7$ ao robô informante;
 - 9: **Receber** $msg6$ do robô informante;
 - 10: $\mathbb{A}_{G_{best}} := \mathbb{A}_{G_{best}}$;
 - 11: **Fim Se**
 - 12: **Até** que todos tenham recebido a atualização
-

Durante toda a etapa de identificação os robôs que não são informantes permanecem em um loop de espera, aguardando a atualização do informante. Após a conclusão da etapa de atualização, todo o enxame é autorizado à prosseguir para a etapa de Critério de Parada.

3.2.2.5 Critério de Parada

A etapa de critério de parada determina se o objetivo foi alcançado e em que momento deve ser finalizado o ADTC. De forma que, caso a alocação \mathbb{A}_{Gbest} satisfaça a proporção-objetivo \mathbb{P} , todos os robôs são autorizados a entrarem em um estado de *loop*, onde vão executar as tarefas nas quais foram alocados. Porém, caso a alocação \mathbb{A}_{Gbest} não satisfaça a proporção-objetivo \mathbb{P} , ou seja, uma qualidade diferente de zero, todos os robôs são encaminhados para o cálculo de uma nova posição no espaço de busca.

Algoritmo 7 Critério de parada

Entrada $\rho, \tau, \mathbb{A}_i, \mathbb{A}_{Pbest}, \mathbb{A}_{Gbest}$, e \mathbb{M} ;

Saída \mathbb{A}_i ;

```

1: Se  $f(\mathbb{A}_{Gbest}) = 0$  Então
2:   Repita
3:     executar tarefa alocada ( $\mathbb{A}_{Gbest}[i]$ );
4:   Até ser reiniciado
5: Senão
6:   Para  $k := 1 \rightarrow \rho$  Faça
7:      $r_1 := \text{GeradorAleatorio}[0 \dots 1]$ ;
8:      $r_2 := \text{GeradorAleatorio}[0 \dots 1]$ ;
9:      $\mathbb{A}_{cognitiva} := c_1 \times r_1 \times (\mathbb{A}_{Pbest_i}[k] - \mathbb{A}_i[k])$ ;
10:     $\mathbb{A}_{social} := c_2 \times r_2 \times (\mathbb{A}_{Gbest}[k] - \mathbb{A}_i[k])$ ;
11:     $\mathbb{A}_{inrcia} := w \times \text{velocidade}[k]$ ;
12:     $\text{velocidade}[k] := \mathbb{A}_{inrcia} + \mathbb{A}_{cognitiva} + \mathbb{A}_{social}$ ;
13:    Se  $\{\text{velocidade}[k] - \lfloor \text{velocidade}[k] \rfloor\} \leq 0.5$  Então
14:       $\text{velocidade}[k] := \lfloor \text{velocidade}[k] \rfloor$ ;
15:    Senão
16:       $\text{velocidade}[k] := \lceil \text{velocidade}[k] \rceil$ ;
17:    Fim Se
18:     $\mathbb{A}_i[k] := \mathbb{A}_i[k] + \text{velocidade}[k]$ ;
19:    Se  $\mathbb{A}_i[k] < t_1$  Então
20:       $\mathbb{A}_i[k] := t_1$ ;
21:    Senão
22:      Se  $\mathbb{A}_i[k] > t_\tau$  Então
23:         $\mathbb{A}_i[k] := t_\tau$ ;
24:      Fim Se
25:    Fim Se
26:  Fim Para
27: Fim Se

```

O Algoritmo 7 demonstra a dinâmica da etapa de critério de parada. Primeiro é avaliada a qualidade da melhor alocação global $f(\mathbb{A}_{Gbest})$, caso ele seja zero, significa que a locação é válida. Nessa situação os robôs entram em um loop de execução da tarefa

para qual foram alocados. Se não for zero, uma nova posição no espaço de busca deve ser calculada. Após o cálculo da nova posição, um novo ciclo do ADTC é iniciado, no qual o robô i é encaminhado novamente a etapa de ajuste.

3.2.3 Função Objetivo

A função objetivo, conforme define a Equação 17:

$$f(\mathbb{A}) = \frac{\sum_{k=1}^{\tau} |\mathbb{C}[k] - \mathbb{C}_{\mathbb{A}i}[k]|}{\tau}, \quad (17)$$

realiza uma média do erro entre \mathbb{C} que representa a quantidade de robôs desejados em cada tarefa e $\mathbb{C}_{\mathbb{A}}$ que representa a quantidade de robôs alocados em cada uma das tarefas. Trata-se de um problema de minimização, onde o objetivo da busca é encontrar o valor zero de qualidade, onde isso somente ocorrerá quando \mathbb{C} for igual a $\mathbb{C}_{\mathbb{A}}$.

Com o intuito de exemplificar, considere um enxame com 20 robôs e duas tarefas $\mathbb{T} = \{0, 1\}$. A Figura 14 mostra uma representação gráfica da qualidade para quatro proporções-objetivo, $\mathbb{P}_1 = \{\frac{5}{20}, \frac{15}{20}\}$, $\mathbb{P}_2 = \{\frac{8}{20}, \frac{12}{20}\}$, $\mathbb{P}_3 = \{\frac{10}{20}, \frac{10}{20}\}$ e $\mathbb{P}_4 = \{\frac{16}{20}, \frac{4}{20}\}$. Essas proporções estão relacionadas aos contadores de tarefas $\mathbb{C}_1 = \{5, 15\}$, $\mathbb{C}_2 = \{8, 12\}$, $\mathbb{C}_3 = \{10, 10\}$ e $\mathbb{C}_4 = \{16, 4\}$. Os gráficos mostram uma convergência para o valor de mínimo da função, onde a função $f(\mathbb{A})$ apresenta valor nulo quando a alocação \mathbb{A} satisfaz a proporção-objetivo \mathbb{P} .

Os gráficos apresentados na Figura 14 são uma representação gráfica da função $f(\mathbb{A})$, porém é uma representação indireta, pois representam a quantidade de robôs em cada tarefa. Uma determinada proporção possui inúmeras soluções possíveis. Por exemplo, as alocações $A_1 = \{0, 2, 2, 1, 0, 0\}$, $A_2 = \{0, 0, 0, 1, 2, 2\}$ e $A_3 = \{2, 2, 0, 1, 0, 0\}$, são todas soluções satisfatórias para a proporção $\mathbb{P} = \{\frac{3}{6}, \frac{1}{6}, \frac{2}{6}\}$, pois $\mathbb{C}_{A_1} = \mathbb{C}_{A_2} = \mathbb{C}_{A_3}$. Como apresentado na Seção 1.2, o problema de ADT possui um número de alocação factível e um número de alocações válidas.

3.3 Considerações Finais do Capítulo

Este capítulo apresentou o algoritmo para alocação dinâmica de tarefas em cluster ADTC que é inspirado pelo PSO, e tem como objetivo minimizar a função objetivo da ADT de um enxame de robôs homogêneos, de forma estocástica e descentralizada. O ADTC

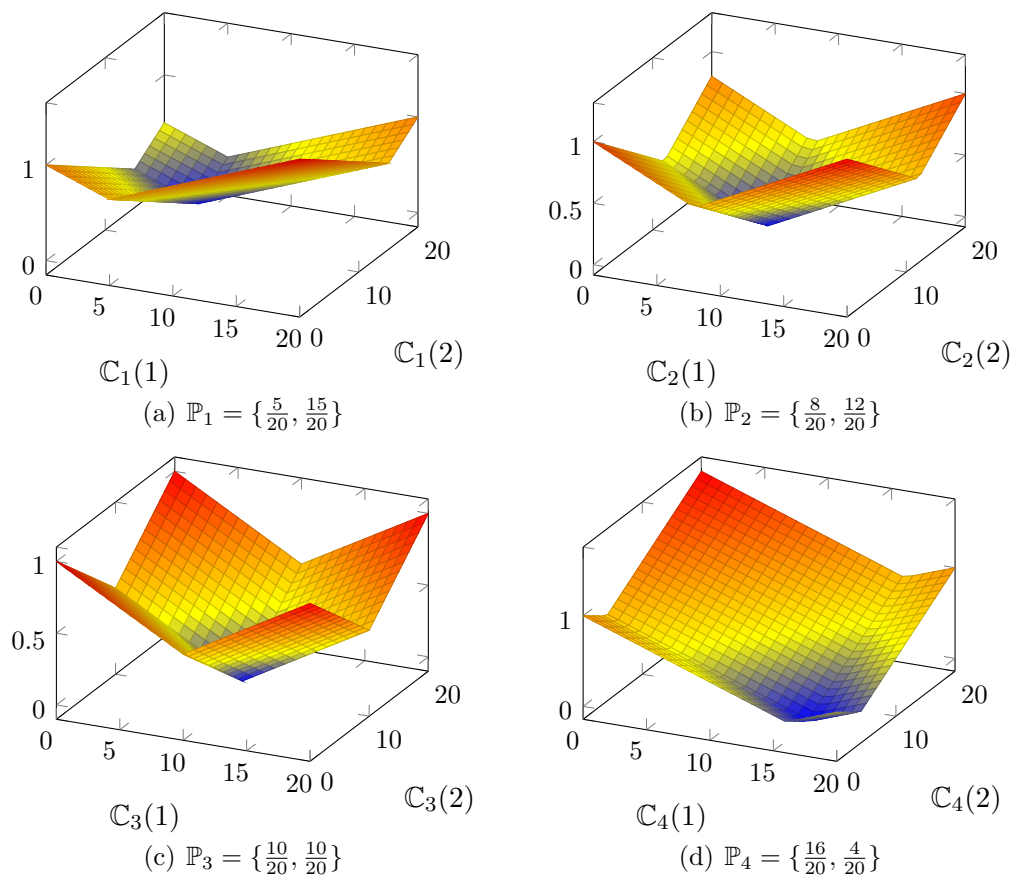


Figura 14: Representação gráfica da função-objetivo com diferentes proporções

procura em um espaço de busca limitado, uma solução para a ADT, onde a que apresentar uma qualidade nula, é considerada como a alocação de tarefas do enxame. No próximo capítulo é apresentada a implementação do ADTC na plataforma ELISA-3.

Capítulo 4

IMPLEMENTAÇÃO DO ALGORITMO

NESTE capítulo é apresentada de forma detalhada a implementação do algoritmo ADTC descrito no Capítulo 3. Escolheu-se uma plataforma física para a implementação e testes do ADTC, o robô ELISA-3.

Na Seção 4.1 apresenta-se a arquitetura física e de comunicação do robô ELISA-3; na Seção 4.2, a implementação, em detalhes, do ADTC e os processos de comunicação utilizados; e na Seção 4.3, as considerações finais do capítulo.

4.1 O Robô ELISA-3

A implementação do algoritmo ADTC necessita de um robô que seja autônomo, programável e que possua uma estrutura de comunicação. O ELISA-3 possui essas características, porém o que o definiu como o robô ideal foi sua estrutura compacta. Essa característica permite a implementação de grandes grupos de robôs. A Figura 15(b) mostra um enxame de robôs ELISA-3. Na Seção 4.1.1 são apresentados maiores detalhes sobre sua estrutura.



(a) Robô ELISA-3



(b) Enxame de robôs

Figura 15: Enxame de robôs ELISA-3

A estrutura de comunicação do ELISA-3 é de modo *wireless*, o que viabiliza a implementação em grandes enxames. No *firmware* existe uma estrutura de comunicação, já programada de fábrica, que auxilia as trocas de mensagens entre os robôs. A estrutura de comunicação será apresentada na Seção 4.1.2 e o protocolo utilizado na comunicação na Seção 4.1.3.

4.1.1 Arquitetura do Robô

O robô ELISA-3, desenvolvido pela empresa Gctronic, é programável e possui uma estrutura física compacta, com 5 centímetros de diâmetro e 3 centímetros de altura, como mostra a Figura 15(a). O mesmo possui uma série de dispositivos embarcados que o possibilita comunicar-se com outros robôs ELISA-3 ou com o usuário, se locomover, etc.

O *hardware* do robô possui diversos dispositivos embarcados, como apresentado na Figura 16¹. O ELISA-3 é um robô microcontrolado, que possui um microcontrolador ATmega 2560 de 8 MHz (ATMEL, 2018) com uma memória RAM de 8 KB. Possui outras duas memórias: uma flash de 256 KB, onde é programado o algoritmo desenvolvido; e uma tipo EEPROM de 4 KB, onde são armazenadas as instruções específicas para o funcionamento, por exemplo, do identificador do robô. A Tabela 1² exibe suas especificações detalhadas.

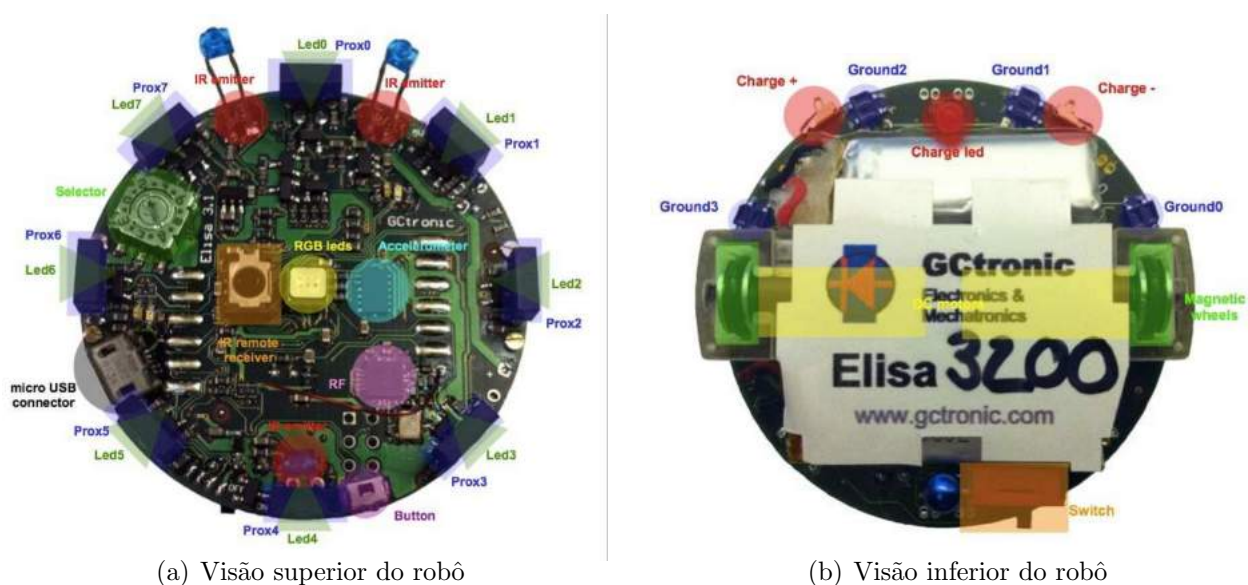


Figura 16: Dispositivos embarcados no *hardware* do robô

¹Figuras retiradas da página <http://www.gctronic.com/doc/index.php/Elisa-3>

²Informações retiradas da página <http://www.gctronic.com/doc/index.php/Elisa-3>

Tabela 1: Especificações técnicas do robô ELISA-3

Característica	Informação técnica
Tamanho, peso	50 mm de diâmetro, 30 mm de altura, 39 g
Bateria, autonomia	Bateria LiPo recarregável (2 x 130 mAh, 3,7 V). Cerca de 3 horas de autonomia. Tempo de recarga de cerca de 1h e 30.
Processador	Atmel ATmega2560 @ 8MHz (8 MIPS); Microcontrolador de 8 bits
Memória	RAM: 8 KB; Flash: 256 KB; EEPROM: 4 KB
Motores	2 motores de corrente contínua com redutor de 25: 1; velocidade controlada com backEMF
Rodas magnéticas	Força de aderência de cerca de 1 N (100 g) dependendo do material de superfície e pintura Diâmetro das rodas = 9 mm Distância entre rodas = 40,8 mm
Rapidez	Max: 60 cm / s
Estrutura mecânica	PCB, porta-motores, plástico branco superior para difundir a luz
Sensores IR	8 sensores infravermelhos medem a luz ambiente e a proximidade de objetos até 6 cm; cada sensor está a 45 ° um do outro 4 sensores de terra detectando a extremidade da superfície viável (colocada na parte frontal do robô)
Emissores IR	3 emissores IR (2 na frente, 1 na parte de trás do robô)
Acelerômetro	Acelerômetro 3D ao longo do eixo X, Y e Z
LEDs	1 LED RGB no centro do robô; 8 LEDs verdes ao redor do robô
Interruptor / selector	Interruptor rotativo de 16 posições
Comunicação	Porta serial padrão (até 38kbps) Sem fio: RF 2,4 GHz; o rendimento depende do número de robôs: por exemplo. 250Hz para 4 robôs, 10Hz para 100 robôs; até 10 m
Controle remoto	Receptor infravermelho para comandos de controle remoto padrão
Barramento	Conectores opcionais: 2 x UART, I2C, 2 x PWM, bateria, terra, tensão analógica e digital
Programação	Programação C / C ++ com o compilador AVR-GCC

O ELISA-3 possui dois motores de passo que permite sua locomoção. Os motores são acoplados a pequenas rodas e a um ímã, possibilitando sua locomoção em qualquer direção e em estruturas metálicas, podendo até mesmo se locomover em vertical ou de ponta a cabeça. O *firmware* possui já programado de fábrica um conjunto de funções para o gerenciamento e utilização dos motores de passo. Ele possui também um acelerômetro para

auxiliar sua locomoção. O acelerômetro (FREESCALE, 2009) informa ao microcontrolador mudanças de inclinação, indispensável para subir uma estrutura vertical, por exemplo.

Outro dispositivo utilizado para o auxílio da locomoção do ELISA-3 são os sensores de infravermelho, localizados próximos aos LEDs verdes. Totalizando 8 sensores IR (SEMICONDUCTOR, 2012), estes são utilizados para detecção de obstáculos durante a locomoção do robô, que ocorre através do recebimento de um feixe IR emitido que é refletido pelo obstáculo. Dessa forma, os sensores de infravermelho possuem duas estruturas, um emissor de IR e um receptor. O receptor ao detectar uma variação, causada pelo sinal IR refletido, indica ao microcontrolador a presença de um obstáculo próximo. Com essa informação, o microcontrolador altera o sentido de movimentação para evitar colisão com o obstáculo. Além dos 8 sensores infravermelho para detecção de obstáculo, o ELISA-3 possui 4 sensores, também de infravermelho (FAIRCHILD, 2011), na parte inferior do robô, responsáveis por monitorar alterações na superfície, informando ao microcontrolador as variações de níveis ou de cor. Esse tipo de informação possibilita que o microcontrolador execute rotinas como: evitar quedas de superfície parando o deslocamento nas extremidades de uma mesa e também orientar o robô para que se desloque seguindo uma linha de uma cor diferente na superfície.

A comunicação com o usuário é realizada através de informações luminosas. Ele possui um LED RGB central que é capaz de emitir qualquer cor do espectro visível, combinando diferentes intensidades de vermelho, verde e azul. Também possui 8 LEDs laterais na cor verde, onde cada um possui um identificador, como por exemplo LED1. Ambos possuem uma função específica utilizada para seu controle. Outro dispositivo que permite a interação com o usuário é um emissor/receptor de sinais IR, na parte superior do robô, que através de um controle remoto o usuário pode controlar o ELISA-3.

O *upload* do programa para o robô é realizado através de um conector micro USB. O algoritmo pode ser desenvolvido em uma programação nas linguagens C/C++, com uma plataforma de compilação AVR-GCC, compatível com a plataforma Arduino IDE (ARDUINO, 2018) ou Atmel Studio 7.

E por último, o dispositivo mais importante para aplicação nesse trabalho, o módulo de comunicação transmissor/receptor. O módulo de comunicação (NORDIC, 2018) do tipo *wireless* de baixo consumo de energia opera em Rádio Frequência (RF) de 2.4

GHz, designado para operar na faixa de rádio ISM (*Industrial, Scientific and Medical band basis*). Na próxima seção será apresentada detalhadamente a estrutura de comunicação implementada.

4.1.2 A Arquitetura de Comunicação

A comunicação é fundamentalmente importante em processos cooperativos, como já apresentado no Capítulo 3. A comunicação utilizada nesse trabalho é a troca de mensagens entre os robôs com o intuito de compartilhar informações pertinentes para a busca de uma solução para o ADT. A estrutura de comunicação embarcada no ELISA-3 é do tipo RF, que possui um *firmware* de comunicação já programado. A comunicação é realizada através de uma estação-base de rádio (nRF24L01+) conectada a um computador, através de um cabo USB, que transfere os dados para os robôs em *wireless*. A Figura 17³ apresenta uma ilustração da estrutura de comunicação.

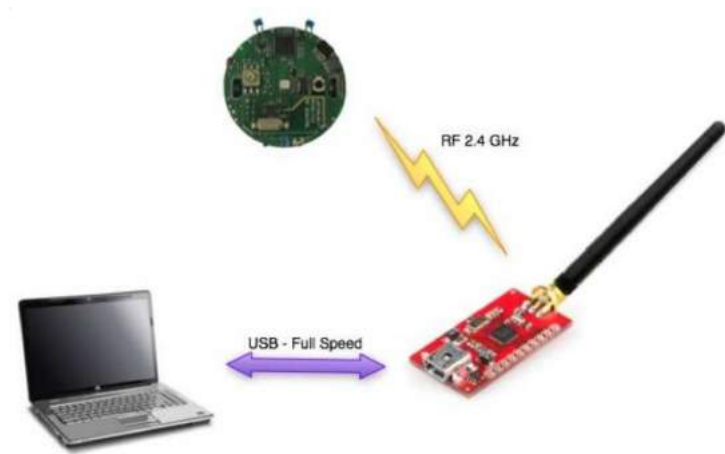


Figura 17: Comunicação dos robôs via RF

O módulo de comunicação utiliza uma comunicação SPI (*Serial Peripheral Interface*) para enviar os dados recebidos nas mensagens para o microcontrolador. O módulo também é responsável por encapsular os dados para serem enviados para a estação-base, que devem estar em uma estrutura contendo as informações necessárias para o envio. A estação-base interpreta as mensagens recebidas, podendo em seguida enviá-las aos seus destinos. As mensagens enviadas pela estação-base são recebidas pelos módulos de comunicação dos robôs, que são transmitidas para os microcontroladores.

³Figuras retiradas da página <http://www.gctronic.com/doc/index.php/Elisa-3>

4.1.3 O Protocolo de Comunicação

Cada robô integrante do enxame possui um identificador *id* (*RFid*), único e armazenado na memória EEPROM, programada de fábrica. Durante o processo de comunicação a EEPROM é acessada para fornecer o *id*. A estação-base identifica todas as mensagens com um endereço de destino que deve corresponder ao endereço encontrado na EEPROM.

Ao receber uma mensagem, o robô irá verificar se o seu endereço de destino corresponde ao seu *id*, confirmando que ele é o destinatário da mensagem. Se ele não for o destinatário, a mensagem é descartada, caso ele seja o destinatário a mensagem é armazenada e disponibilizada para o microcontrolador.

A estação-base possui um buffer que armazena as mensagens recebidas, que posteriormente são encaminhadas ao computador para que este possa interpretá-las. O computador realiza um monitoramento constante da estação-base, controlando o envio e o recebimento das mensagens. O computador verifica o status do buffer uma vez a cada milissegundo, o que limita a velocidade de comunicação. A estrutura de comunicação, juntamente com o protocolo utilizado, permite o envio e recebimento de mensagens de 64 *bytes*, que corresponde a mensagens de ou para 4 robôs. Assim, a estação-base tem a capacidade de enviar e receber mensagens para quatro robôs simultaneamente, o que suaviza a limitação de uma mensagem a cada milissegundo. A separação da mensagem em quatro pacotes individuais é realizada pela estação-base que envia separadamente cada mensagem ao seu destino. O processo inverso também é realizado pela estação-base, que recebe quatro mensagens de 16 *bytes* e forma um único pacote de 64 *bytes*, que será transmitido para o computador.

A estrutura da mensagem de 16 *bytes* enviada pelo robô à estação-base é apresentada na Figura 18. A mensagem é formada por quatro blocos: o primeiro com 1 byte; o segundo, com dois; o terceiro, com 12; e o último com 1, totalizando os 16 *bytes* da mensagem. O primeiro bloco é o *byte de validade* (BV), ele tem como finalidade validar a autenticidade da mensagem enviada pelo robô. Esse byte será utilizado pela estação para verificar a veracidade da mensagem. O próximo bloco contém o identificador *RFid* do robô emissor da mensagem, O terceiro bloco é a *carga útil* da mensagem, que pode conter quaisquer dados que precisem ser transmitidos. E o último bloco classifica a mensagem em relação ao seu tipo.



Figura 18: Estrutura da mensagem enviada pelo robô para a estação-base

A outra estrutura de mensagem é a enviada da estação para o robô, que também possui 4 blocos como mostra a Figura 19. Ele possui semelhanças com a estrutura anterior, igualmente possuindo 4 blocos que totaliza 16 *bytes*. O primeiro bloco é o *bytes de validade* com 1 *byte*; o segundo, é a *carga útil* com 12 *byte*; o terceiro, é o tipo de mensagem com 1 *byte*; e o quarto, é o endereço de destino da mensagem *RFid*, com 2 *bytes*. A próxima seção apresenta a implementação do ADTC nos robôs ELISA-3.



Figura 19: Estrutura da mensagem enviada pela estação-base para o robô

4.2 Metodologia da Implementação

A implementação do ADTC no robô ELISA-3 seguiu todos os processos e etapas especificados no capítulo anterior, sendo necessária a utilização de alguns dos dispositivos disponibilizados no robô. Além dos básicos de programação e processamento - como as memórias, o processador e porta micro USB - também são utilizados outros dispositivos para a comunicação com o meio externo. Um dos dispositivos de comunicação é o LED RGB, utilizado como uma comunicação visual que informa qual a etapa ou qual a tarefa o robô *i* está realizando. Os 8 LED's verdes de baixa intensidade também são utilizados como dispositivos de comunicação e sua função é informar qual tipo de mensagem o robô está enviando. O dispositivo mais importante para a implementação do ADTC é o módulo de comunicação por RF, responsável por trocar mensagens entre os robôs ELISA-3, fundamental para o desenvolvimento do PSO nesta implementação.

O processo de comunicação no ADTC possui três tipos distintos de mensagens, onde cada uma delas possui um objetivo e uma estrutura diferente. O primeiro tipo de mensagem tem como finalidade informar o valor de qualidade (P_{best} ou C_{best}), sendo utilizado na comunicação nos clusters, onde os robôs divulgam os respectivos valores da

qualidade P_{best} com o objetivo de encontrar o robô C_{best} . É também utilizado pelos informantes com o objetivo de encontrar o robô G_{best} .

Após a identificação do robô C_{best} nos clusters ou a identificação do robô G_{best} entre os informantes, esses robôs precisam enviar as alocações correspondentes ($A_{C_{best}}$ ou $A_{G_{best}}$) para os demais robôs como detalhado na Seção 3.2.2. O segundo tipo de mensagem possui essa finalidade, o envio de alocações. O terceiro tipo de mensagem é utilizado para o confirmação do recebimento da alocação enviada pelo robôs C_{best} e G_{best} , controlado pelo *byte de confirmação* (CF). Ao todo são utilizados três tipos de mensagens, como mostra a Figura 20, ilustrando as respectivas configurações da carga útil em cada um dos tipos de mensagem.

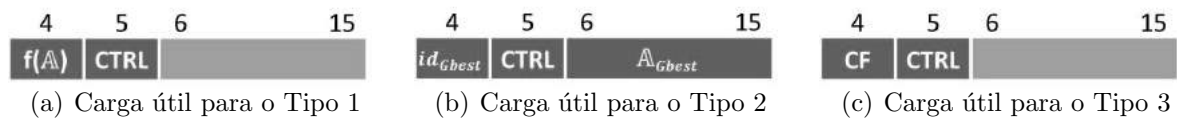


Figura 20: Tipos de mensagens enviadas pelo robô no ADTC

O *byte de controle* (CTRL) é utilizado para informar em que situação o processo de comunicação está naquele momento. Por exemplo, na mensagem tipo 1 juntamente com o valor da aptidão é enviado o *byte de controle*, que tem como finalidade informar aos demais robôs em qual situação o processo de comunicação está. O *byte de controle* pode informar três situações diferentes; quando no estado 0, significa que ainda não recebeu a qualidade de todos os robôs; quando no estado 1, significa que recebeu a qualidade de todos, sinalizando o início da conclusão do processo de comunicação; e quando em estado 2, significa que além de ter recebido a qualidade de todos, também recebeu a confirmação de recebimento de todos os robôs e que por isso o processo pode ser concluído. Resumindo, o processo de comunicação só será encerrado quando todos os robôs enviarem o *byte de controle* no estado 2. O *byte de controle* também é utilizado nas mensagens tipo 2 e 3 com a mesma finalidade.

A Figura 20 (a) apresenta a estrutura da carga útil da mensagem tipo 1, utilizada para troca de valores de qualidade entre os robôs. Dos 12 *bytes* de carga útil, são utilizados apenas 2 *bytes*, que corresponde ao valor de qualidade e o CTRL. A Figura 20 (b) apresenta a estrutura da carga útil da mensagem tipo 2, que tem como finalidade o envio da alocação. Esse tipo de mensagem utiliza os 12 *bytes* da carga útil, sendo que o primeiro *byte* é o

$id_{G_{best}}$; o segundo, o CTRL; e os 10 *bytes* restantes, a alocação. Por último, a mensagem tipo 3, utilizada para confirmar o recebimento da alocação, usa apenas os dois primeiros *bytes* (CF e CTRL), como demonstrado na Figura 20 (c).

A comunicação entre os robôs é intermediada pela estação-base, como apresentado na Seção 4.1.2 deste Capítulo. A estação-base tem a capacidade de intermediar quatro mensagens por pacote. As mensagens são recebidas pela estação e armazenadas em um *buffer* específico, de forma sequencial, formando o pacote de 64 *bytes*, como mostra a Figura 21. No momento em que a estação detecta a existência de um novo pacote no *buffer*, este é enviado ao computador que irá interpretar o conteúdo das mensagens.

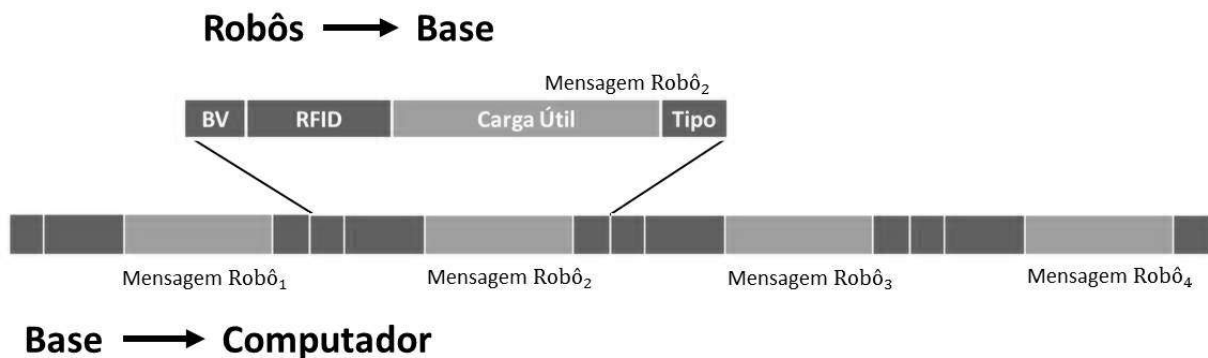


Figura 21: Formato do pacote das mensagens recebidas pela estação-base

O computador detecta o tipo de mensagem, armazena as cargas úteis e suas identificações $RFID_i$. Após a detecção e o armazenamento, é construído o pacote de resposta que será enviado da estação para os robôs, como mostra a Figura 22. Este pacote é constituído de quatro mensagens destinadas a quatro robôs distintos, que é encaminhado para a base que o envia sequencialmente para os quatro robôs.

Com o objetivo de otimizar a comunicação usufruindo ao máximo da carga útil das mensagens, realiza-se uma modificação nas mensagens de tipo 1 e 3, enviadas pela base para os robôs. A carga útil de uma mensagem é de 12 *bytes*, por isso é possível enviar em uma mesma mensagem a identificação id , a qualidade $f(\mathbb{A})$ e o CTRL de quatro robôs, em que essas três informações ocupam apenas 3 *bytes*. A mensagem tipo 3 é semelhante, porém, substituindo a qualidade pelo *bytes de confirmação* que também ocupa os mesmos 3 *bytes*. Figura 23 apresenta a estrutura das mensagens tipo 1 e 3 otimizadas. Para que isso seja possível, o identificador RFID precisou ser codificado, de modo a ocupar apenas

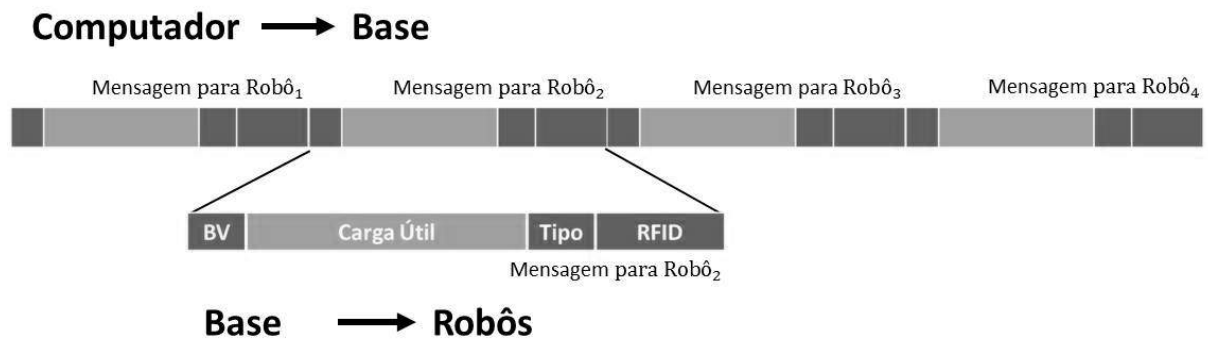
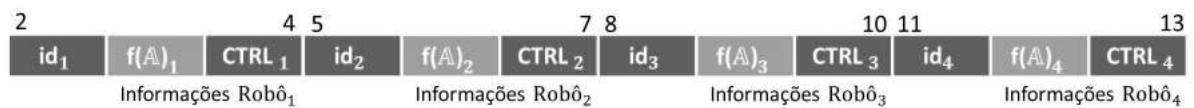
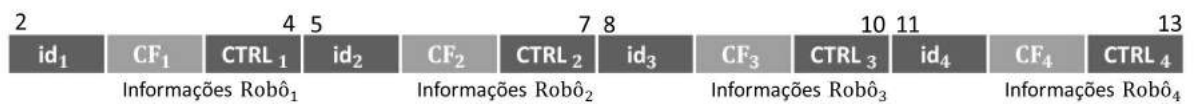


Figura 22: Formato do pacote das mensagens enviadas pela estação-base

1 byte.



(a) Carga útil para o Tipo 1 Otimizado



(b) Carga útil para o Tipo 3 Otimizado

Figura 23: Mensagens otimizadas

A Figura 24 apresenta a dinâmica de formação dos pacotes que ocorre durante a comunicação dos robôs com a base e da base com os robôs. No exemplo apresentado ocorre simultaneamente o envio da alocação pelo robô G_{best} e o envio da confirmação de recebimento dos demais robôs. A formação dos pacotes que são enviados da base para os robôs estão relacionados com as mensagens enviadas pelos robôs para base. A finalidade da estação-base é apenas intermediar a comunicação entre os robôs. Uma determinada mensagem só será enviada para o robô i se algum dos robôs enviar para a base uma mensagem destinada para o robô i .

Com o objetivo de formular uma definição mais completa para o número de mensagens enviadas por ciclo na topologia de comunicação em cluster e na topologia de comunicação em malha completa, os conceitos e processos apresentados neste capítulo e no Capítulo 3 complementaram a Equação 15 e a Equação 16 tornando-se funções mais completas para o cálculo do número de mensagens enviadas por ciclos. Dessa forma, a função \mathbb{F}_c^* representa o número de mensagens enviadas a cada ciclo do ADTC, semelhante

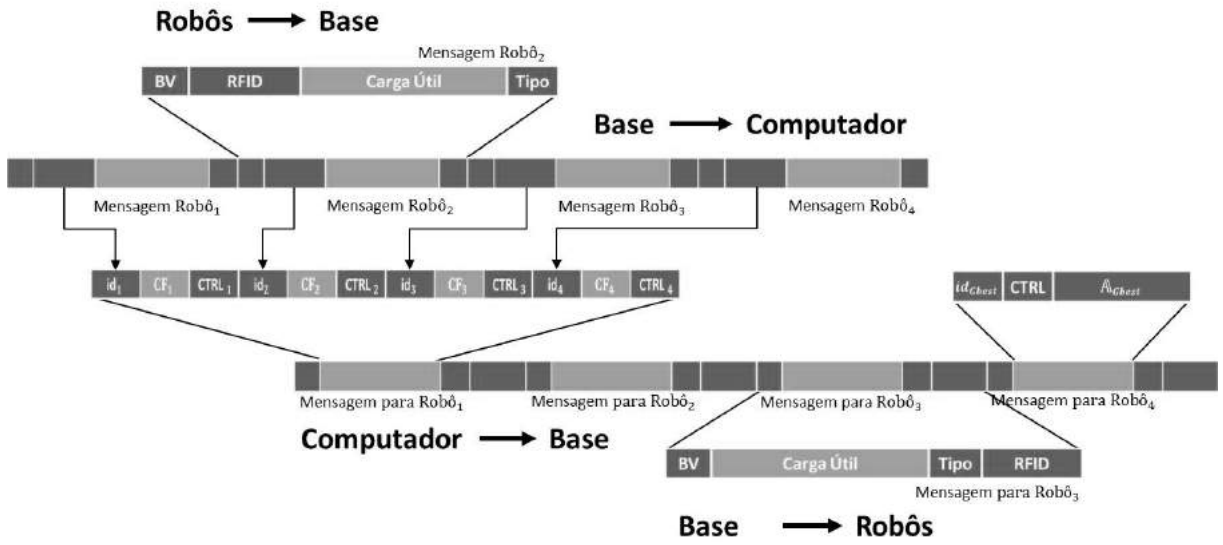


Figura 24: Dinâmica da formação dos pacotes

a Equação 12, sendo também formado pelo somatório do número de mensagens utilizadas nas etapas de ajuste, identificação e atualização como mostra a Equação 18:

$$\mathbb{R}^* = F_1^* + F_2^* + F_3^* \quad (18)$$

onde o termo F_1^* representa o número de mensagens utilizadas na etapa de ajuste, no qual para a identificação do robô C_{best} cada robô envia 3 mensagens (1 mensagem com a qualidade e 2 mensagens de controle) para os demais robôs do cluster. Após a identificação do robô C_{best} , o mesmo envia 1 mensagem para os demais robôs do cluster com sua alocação e os demais robôs enviam 1 mensagem de controle confirmando o recebimento da alocação. O robô C_{best} ao receber a confirmação do recebimento da alocação de todos os robôs do cluster, envia 1 mensagem de controle sinalizando a conclusão desta etapa, conforme mostra a Equação 19:

$$F_1^* = 3\gamma(\rho_c(\rho_c - 1)) + 3\gamma(\rho_c - 1). \quad (19)$$

O termo F_2^* corresponde ao número de mensagens utilizadas na etapa de identificação, onde para identificar o robô G_{best} cada informante envia 3 mensagens (1 mensagem com a qualidade e 2 mensagens de controle) para os demais informantes. Após a identificação do robô G_{best} , o mesmo envia 1 mensagem para os demais informantes como sua alocação e os demais informantes enviam 1 mensagem de controle confirmando o recebimento da alocação. O robô G_{best} ao receber a confirmação do recebimento da alocação de to-

dos os informantes envia 1 mensagem de controle sinalizando a conclusão da etapa de identificação, conforme demonstrado pela Equação 20:

$$F_2^* = 3\gamma\sigma(\gamma\sigma - 1) + 3(\gamma\sigma - 1) \quad (20)$$

E por fim, o termo F_3^* representa o número de mensagens utilizadas na etapa atualização, onde o informante envia 1 mensagem para os demais robôs do seu cluster com a alocação do robô G_{best} e os demais robôs enviam 1 mensagens de confirmação para o informante do seu cluster. O informante ao receber a confirmação do recebimento da alocação de todos os robôs do cluster envia 1 mensagem de controle sinalizando a conclusão da etapa, como mostra a Equação 21:

$$F_3^* = 3\gamma\sigma(\rho_c - 1). \quad (21)$$

Realizando as devidas substituição na Equação 18 e considerando $\sigma = 1$, a formula \mathbb{F}_c^* fica apenas em função no número de robôs ρ e o número de clusters γ , conforme apresenta a Equação 22:

$$\mathbb{F}_c^* = 3\gamma^2 + 3\frac{\rho^2}{\gamma} - 6\gamma - 3\rho - 3. \quad (22)$$

Da mesma forma, a função \mathbb{F}_m^* demonstra o cálculo do número de mensagens enviadas por ciclo na topologia em malha completa, sendo inseridos os conceitos e processos apresentados neste capítulo e no Capítulo 3 complementaram a Equação 16. A função \mathbb{F}_m^* corresponde ao número de mensagens utilizadas para identificar o robô G_{best} e atualizar a melhor alocação do enxame $\mathbb{A}_{G_{best}}$. Para identificar o robô G_{best} , cada robô envia 3 mensagens (1 mensagem com a qualidade e 2 mensagens de controle) para os demais robôs. Após a identificação do robô G_{best} , o mesmo envia 1 mensagem para os demais robôs como sua alocação e os demais enviam 1 mensagem de controle confirmando o recebimento da alocação. O robô G_{best} ao receber a confirmação do recebimento da alocação de todos os robôs envia 1 mensagem de controle sinalizando a conclusão da comunicação, conforme demonstrado pela Equação 23:

$$\mathbb{F}_m^* = 3\rho(\rho - 1) + 3(\rho - 1). \quad (23)$$

4.3 Considerações Finais do Capítulo

Neste capítulo foram apresentadas as características do robô ELISA-3 e os aspectos de implementação do algoritmo ADTC, explicando seus diversos dispositivos embarcados,

com ênfase no módulo de comunicação RF, utilizado para troca de informações entre os robôs através do intermédio de uma estação-base, gerenciada por um computador.

A implementação do ADTC é realizada através de um protocolo de comunicação otimizado que permite, simultaneamente, a troca de informação de quatro mensagens. O protocolo implementado também permite a comunicação com diferentes tipos de mensagens paralelamente. No próximo capítulo são apresentados os resultados e as análises dos ensaios realizados com robôs reais.

Capítulo 5

ANÁLISE DOS RESULTADOS

NESTE capítulo será apresentada uma análise sobre os resultados obtidos pela implementação do ADTC nos robôs ELISA-3. Essa análise tem como objetivo avaliar a eficácia do algoritmo ao resolver o problema de ADT. Também são analisadas as características e o desempenho da topologia de comunicação em clusters.

O algoritmo é implementado em diferentes estruturas e arranjos de enxame, variando o número total de robôs e o número de tarefas, tornando a análise mais diversificada e os resultados obtidos mais robustos. Os resultados apresentados preocupam-se em serem estatisticamente válidos, por isso, cada um dos ensaios é repetido 10 vezes. Como objetivo de apresentar o ADTC um vídeo foi postado no *link* <https://www.youtube.com/user/temente2>, onde demonstra a aplicação do ADTC para alocação dinâmica de tarefas em um enxame de robôs utilizando uma topologia de comunicação em cluster.

Este capítulo é dividido em 5 seções: na Seção 5.1 será apresentada a metodologia de avaliação utilizada nos ensaios; na Seção 5.2, uma análise sobre o número de clusters; na Seção 5.3, uma análise sobre o número de informantes; na Seção 5.4, os resultados do algoritmo ADTC; na Seção 5.5, a comparação do desempenho do algoritmo ADTC e ADTG; e na Seção 5.6, as considerações finais desse capítulo.

5.1 Metodologia de Avaliação

A metodologia utilizada nos ensaios avalia os resultados obtidos com diferentes estruturas de enxame que são definidas pelo número de robôs ρ e pelo número de tarefas τ . Cada um dos ensaios tem como objetivo alocar tarefas, dinamicamente, até encontrar uma nova alocação \mathbb{A}^* que atenda aos requisitos da proporção-objetivo \mathbb{P} . Em todos os ensaios o

enxame é inicializado em estado zero, o que significa alocar todos os robôs na tarefa 0. O enxame iniciará em todas os ensaios com a mesma alocação inicial $\mathbb{A}_0 = \{0, 0, \dots, 0\}$.

As proposições-objetivo utilizadas nos ensaios possuem o número de tarefas que varia de 2 à 5 tarefas. A Tabela 2 apresenta a distribuição padrão da proporção-objetivo que será utilizada em todos os ensaios. As estruturas dos enxames utilizadas variam de 8 à 50 robôs. Os diferentes arranjos de robôs e tarefas possuem níveis de complexidade diferentes. Como apresentada no Capítulo 1, a complexidade do problema de alocação de tarefas em um enxame homogêneo está relacionada com o par (ρ, τ) , onde cada par possuirá um número de alocações factíveis \mathbb{Q} . A Tabela 3 apresenta o número de alocações factíveis \mathbb{Q} para cada um dos ensaios realizados.

Tabela 2: Distribuição da proporção-objetivo entre as tarefas

# Tarefas	Proporção				
	Tarefa 0	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4
2	60%	40%	–	–	–
3	20%	30%	50%	–	–
4	10%	15%	30%	45%	–
5	5%	10%	20%	30%	35%

Tabela 3: Número de alocações factíveis para os ensaios realizados

# \mathbb{Q}	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$
$\rho = 8$	256	6561	65536	390625
$\rho = 12$	4096	531441	16777216	244140625
$\rho = 16$	65536	43046721	4294967296	$1,52588 \times 10^{11}$
$\rho = 20$	1048576	3486784401	$1,09951 \times 10^{12}$	$9,53674 \times 10^{13}$
$\rho = 24$	16777216	$2,8243 \times 10^{11}$	$2,81475 \times 10^{14}$	$5,96046 \times 10^{16}$
$\rho = 28$	268435456	$2,2877 \times 10^{13}$	$7,2057 \times 10^{16}$	$3,7253 \times 10^{19}$
$\rho = 32$	4294967296	$1,8530 \times 10^{15}$	$1,8447 \times 10^{19}$	$2,3283 \times 10^{22}$
$\rho = 40$	$1,0995 \times 10^{12}$	$1,2158 \times 10^{19}$	$1,2089 \times 10^{24}$	$9,0949 \times 10^{27}$
$\rho = 50$	$1,1259 \times 10^{15}$	$7,1790 \times 10^{23}$	$1,2676 \times 10^{30}$	$8,8818 \times 10^{34}$

A análise preocupa-se com a obtenção de resultados estatisticamente válidos, por esse motivo cada um dos ensaios realizados foram executados 10 vezes. Os resultados finais a serem analisados são uma média dos 10 ensaios. São analisados nos ensaios três fatores. O primeiro é o tempo de convergência, o tempo que o algoritmo implementado leva para encontrar uma alocação \mathbb{A}^* que satisfaça a proporção-objetivo. Esse resultados são mensurados em milissegundo (ms), que é o tempo entre a inicialização dos robôs e a

finalização da alocação, considerando que o algoritmo alcançou o seu objetivo. O segundo e o terceiro fator que serão analisados estão relacionados ao aspecto da comunicação. São analisados o número de mensagens enviadas pelos robôs à base (Mensagens Enviadas) e o número de mensagens enviadas da base para os robôs (Mensagens Recebidas).

5.2 Impacto do Tamanho do Cluster

Essa bateria de ensaios tem como objetivo avaliar o tamanho ideal do cluster. Sendo que ao avaliar os resultados desses ensaios, será possível concluir qual é o tamanho ideal de cluster para um melhor desempenho do ADTC.

Como o tamanho do cluster é o objeto de estudo, todas as demais variáveis permanecem constantes, tais como número de robôs, proporção-objetivo, número de tarefas e parâmetros iniciais. Como mostra a Figura 25, os ensaios são divididos em cinco partes. Com um enxame de 24 robôs, a primeira parte dividirá o enxame em dois clusters Figura 25(a), a segunda em três clusters Figura 25(b), a terceira em quatro clusters Figura 25(c), a quarta em seis clusters Figura 25(d) e a quinta em oito clusters Figura 25(e).

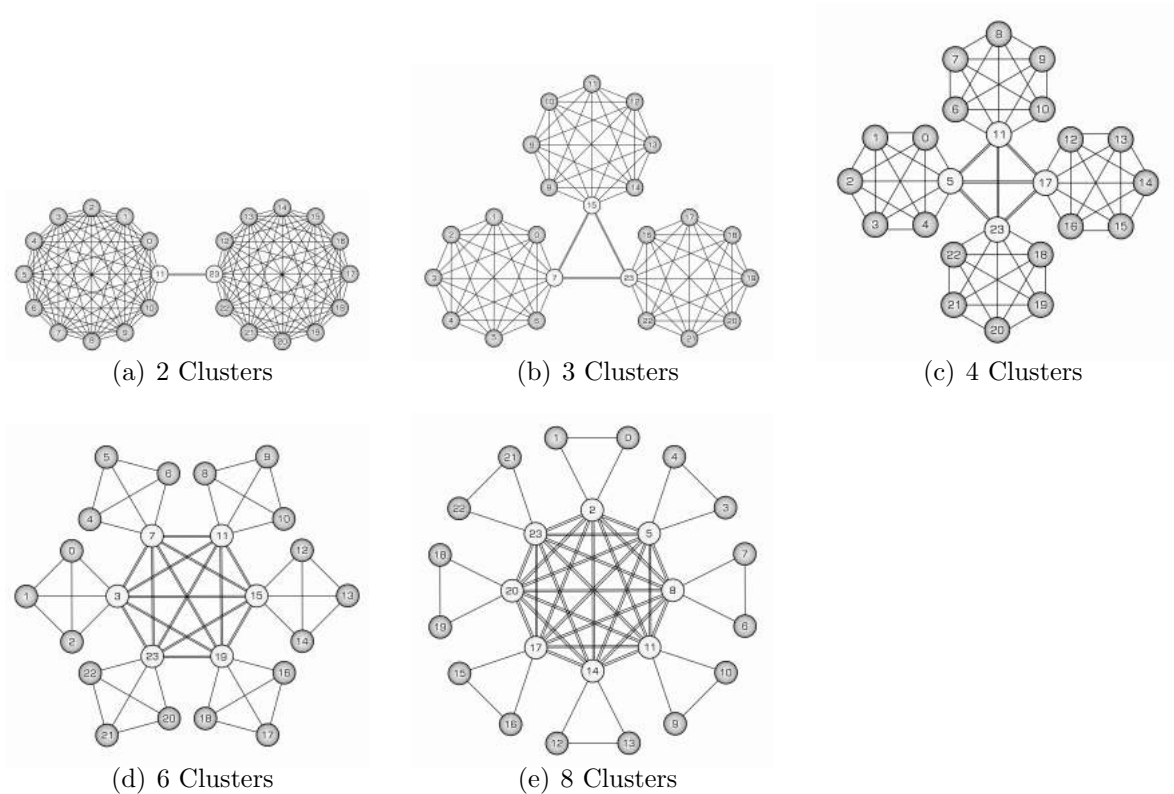


Figura 25: Topologias de comunicação em cluster com 24 robôs divididos em diferentes números de clusters

A Figura 26 apresenta o número de mensagens enviadas por ciclo. Os ensaios possuem uma topologia de comunicação que varia de 3, 4, 6, 8 e 12 robôs por clusters, em relação a topologia de comunicação em cluster. A Figura 27 apresenta duas curvas de tendência do número de mensagens enviadas com o algoritmo ADTC, que utiliza a topologia de comunicação em cluster. A curva vermelha representa os resultados práticos, que é a média do número de mensagens enviadas por ciclo de 10 amostras, para a alocação de 5 tarefas em um enxame de 24 robôs, e a curva azul os resultados teóricos conforme a Equação 22. Apresenta também duas retas que correspondem ao número de mensagens enviadas com o algoritmo ADTG, que utiliza a topologia de comunicação em malha completa. A reta verde representa o resultado prático, que é a média do número de mensagens enviadas por ciclo de 10 amostras, para a alocação de 5 tarefas em um enxame de 24 robôs, e a reta preto, o resultado teórico conforme a Equação 23.

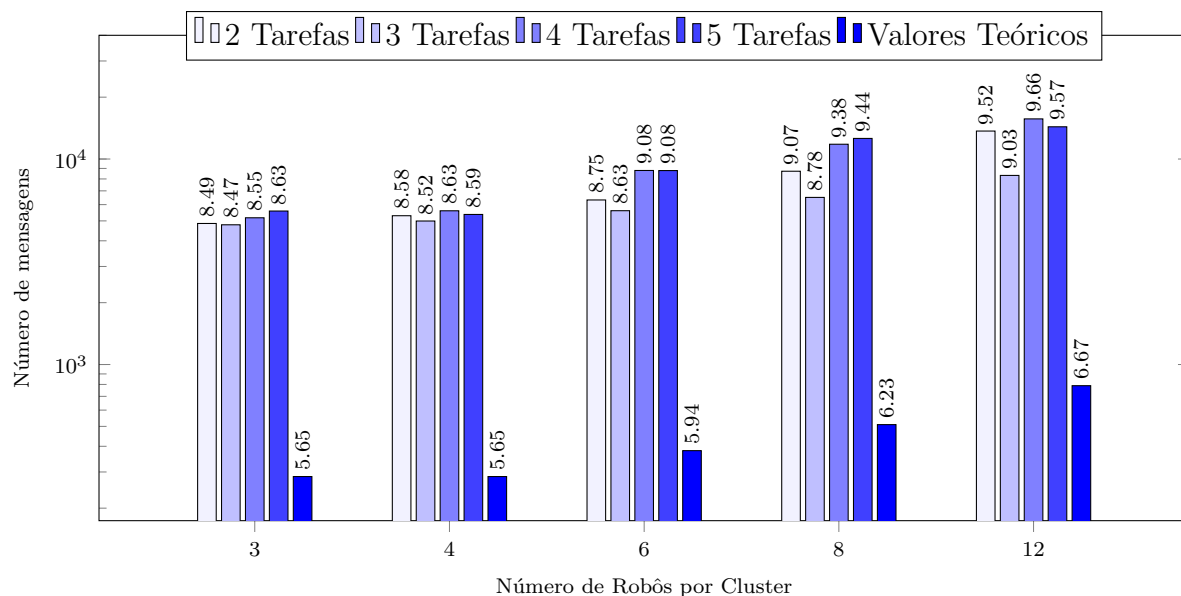


Figura 26: Número de mensagens enviadas por ciclo para alocação de 24 robôs com diferentes números de clusters

Os resultados apresentados nas Figuras 26 e 27 comprovam a validade da Equação 22. As curvas apresentam um comportamento similar, convergindo para o mesmo ponto de mínimo. A Equação 22 será utilizada para definir o tamanho ideal dos clusters em ensaios futuros, com o objetivo de minimizar o fluxo de informação entre os robôs.

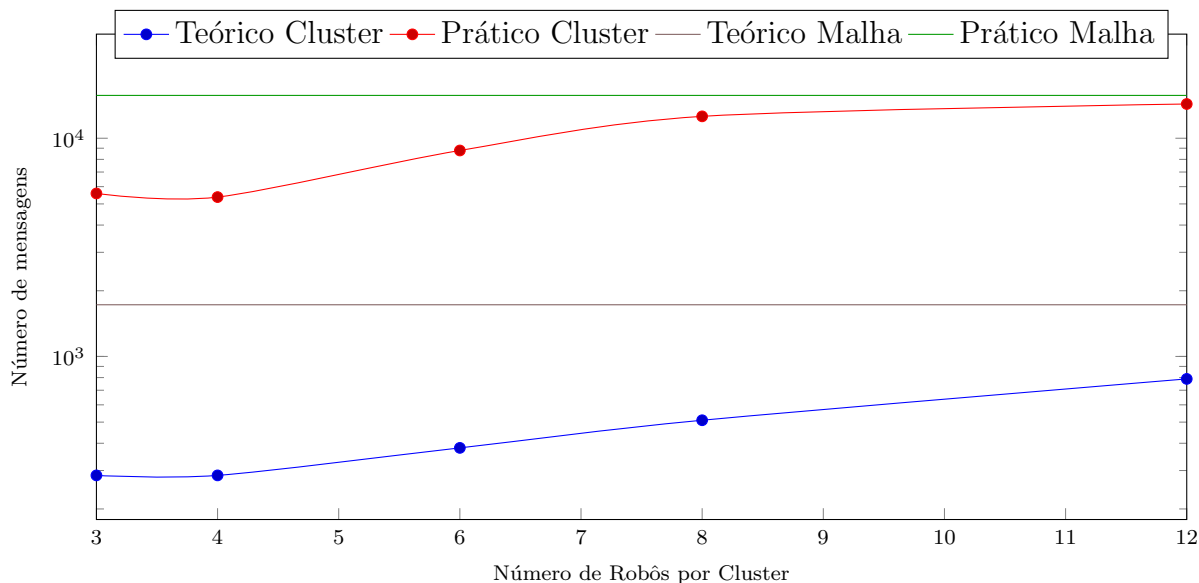


Figura 27: Curvas do número médio de mensagens enviadas por ciclo para alocação de 5 tarefas em um enxame de 24 robôs

5.3 Impacto do Número de Informantes

Esses ensaios têm como objetivo avaliar o impacto no desempenho do ADTC ao variar o número de informantes por cluster. Semelhante aos ensaios de número de cluster, o número de informantes é o objeto de estudo, por isso as demais variáveis devem permanecer constantes durante todos os ensaios.

Os ensaios possuem três etapas como mostra a Figuras 28. Em um enxame de 24 robôs dividido em 4 clusters, a primeira etapa será com um informante por cluster, totalizando 17% de informantes Figura 28(a), a segunda com dois informantes por cluster, totalizando 38% de informantes Figura 28(b) e a terceira com três informantes por cluster, totalizando 50% de informantes Figura 28(c).

A Figura 29 apresenta a média do tempo de convergência de 10 amostras, para a alocação do enxame de 24 robôs com uma variação de 2 à 5 tarefas. Os ensaios possuem uma topologia de comunicação que varia de 17%, 38% e 50% de informantes. A Figura 30 apresenta a média do total de mensagens enviadas de 10 amostras para a alocação de 5 tarefas em um enxame de 24 robôs.

Os resultados apresentados nas Figuras 29 e 30 demonstram que a variação do número de informantes não causa grandes variações no desempenho do ADTC, tanto no

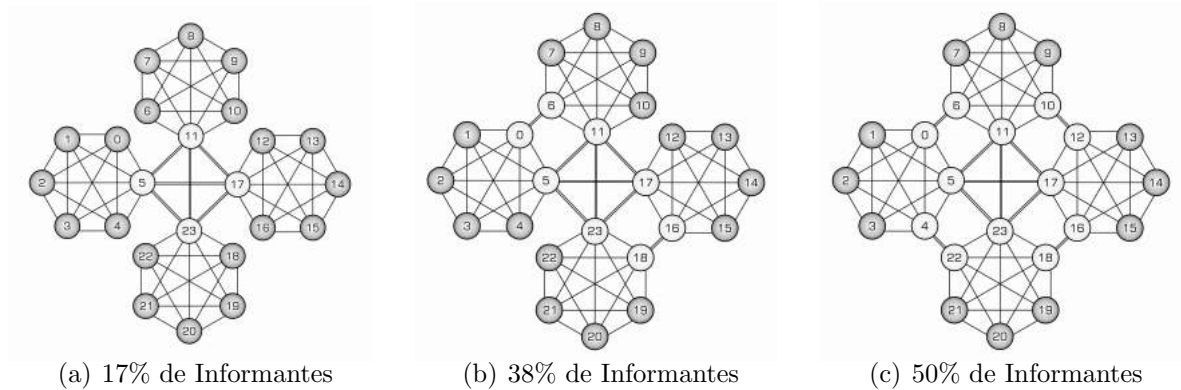


Figura 28: Topologias de comunicação em cluster com 24 robôs com números de informantes diferentes

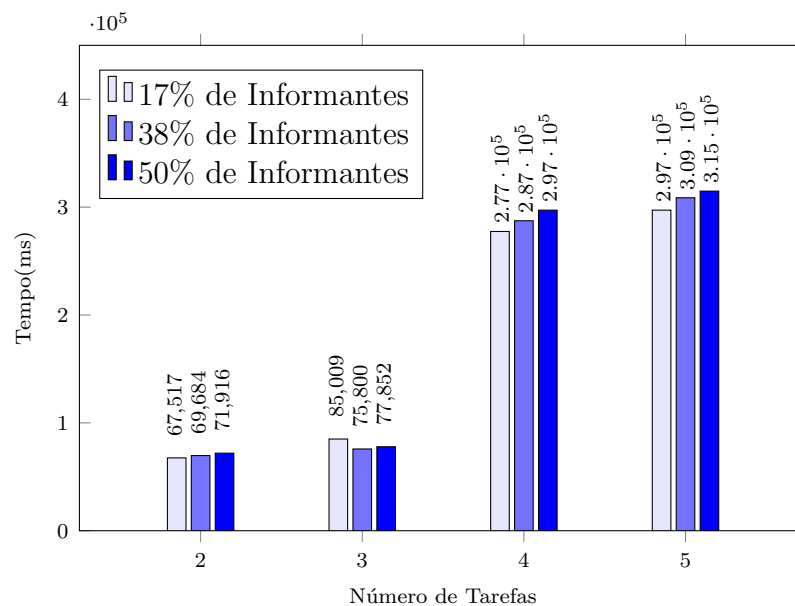


Figura 29: Tempo de convergência para alocação de 24 robôs variando o número de informantes

tempo de convergência como no número de mensagens enviadas. Existe, na média, uma pequena piora nos resultados, demonstrando que a melhor topologia é a com o menor número de informantes possível. Em ensaios futuros é implementado apenas 1 informante por cluster.

5.4 Desempenho do ADTC

Esse grupo de ensaios tem como objetivo avaliar o funcionamento e a capacidade do algoritmo desenvolvido, ADTC. Os grupos de robôs utilizados nessas avaliações, possuem

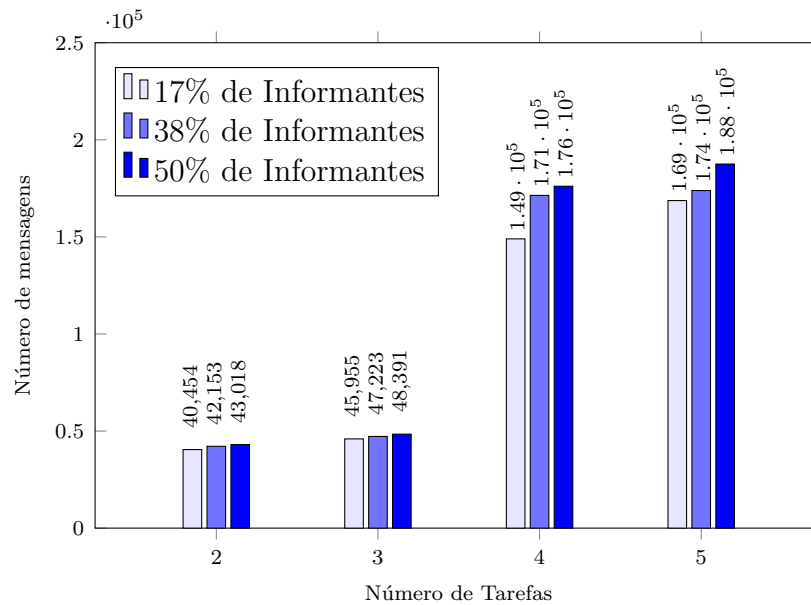


Figura 30: Número de mensagens enviadas para alocação de 24 robôs variando o número de informantes

um número de cluster e de informantes otimizados em função dos resultados dos ensaios anteriores, com o intuito de alcançar o melhor resultado possível.

Os ensaios utilizam topologias de comunicação em cluster, onde o número de clusters está em função do número total de robôs. Como mostra a Figura 31, são realizados ensaios com 9 estruturas diferentes. O objetivo deles é avaliar o funcionamento e capacidade do ADTC, por isso o ADTC é executado em diferentes níveis de complexidade.

A Figura 32 apresenta a média do tempo de convergência, em milissegundos, de 10 amostras para exames de robôs que variam de 8 à 50 robôs, e para uma alocação que varia de 2 à 5 tarefas. A Figura 33 apresenta a média do número total de mensagens recebidas e a Figura 34, a média do número total de mensagens enviadas.

Os resultados apresentados nessa Seção demonstram que o desempenho (tempo de convergência) e o fluxo de informação (número de mensagens enviadas e recebidas) possuem, na média, um comportamento similar. O que caracteriza uma relação direta do processo de comunicação com o resultado final de uma implementação deste tipo. O ADTC obteve êxito em todos os ensaios realizados, no entanto, como observado nos relatos apresentados, os ensaios com 50 robôs alcançam a convergência somente com 2 e 3 tarefas. Esse comportamento ocorreu devido ao fato de que a plataforma escolhida (ELISA-3) possui algumas limitações físicas, que neste caso estão em função da capacidade

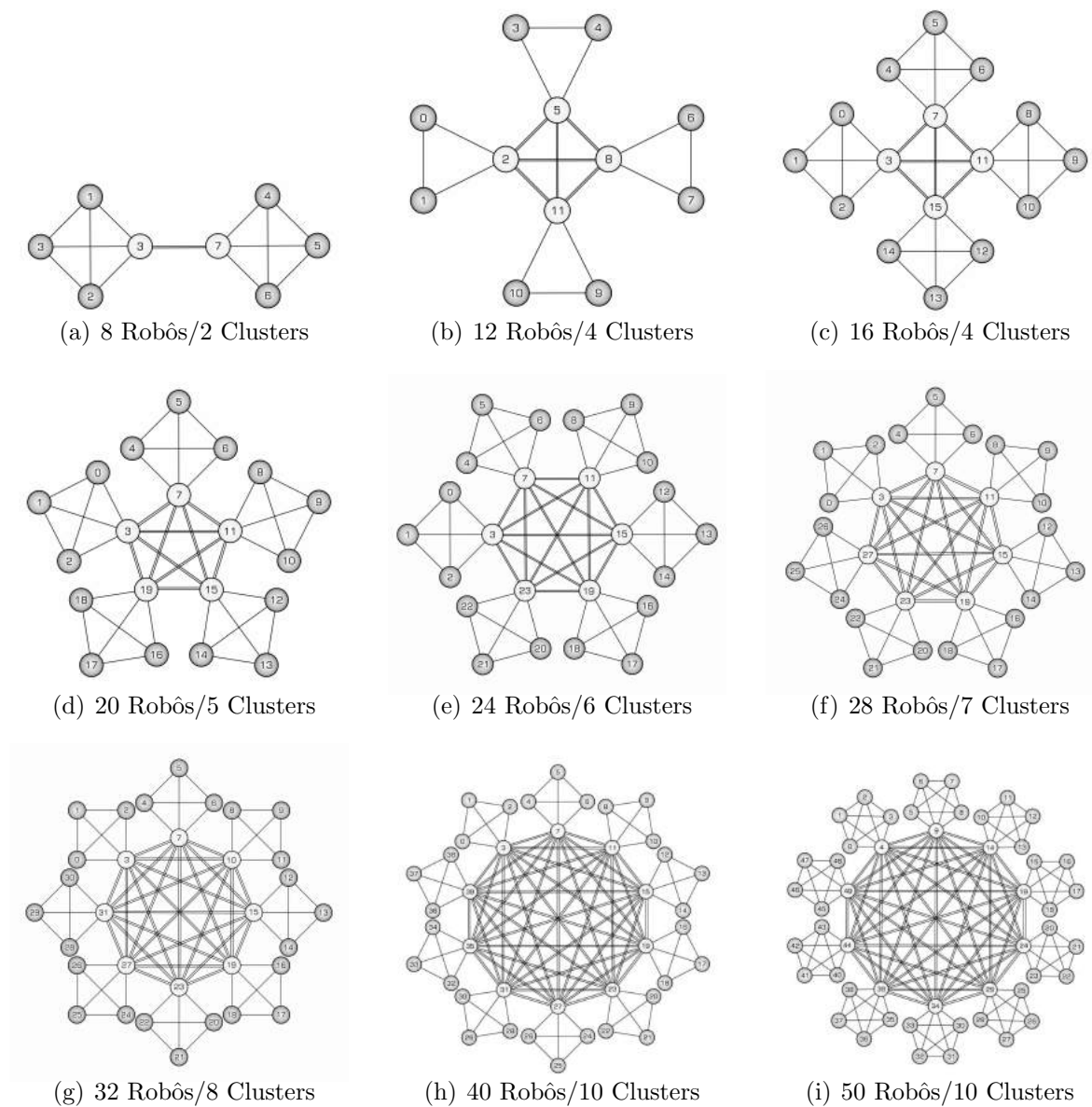


Figura 31: Topologias de comunicação utilizadas na implementação do ADTC

de carga da bateria, que limita o tempo útil de funcionamento. No ensaio com 50 robôs para a alocação de 4 tarefas, houve o esgotamento da carga antes da convergência da solução.

5.5 Comparação entre o ADTC e o ADTG

Os resultados apresentados são uma comparação entre os resultados obtidos no trabalho (MENDONÇA, 2014) com os resultados obtidos nesta dissertação. O ADTG é um algoritmo utilizado para buscar uma solução para o ADT utilizando o PSO de forma distribuída

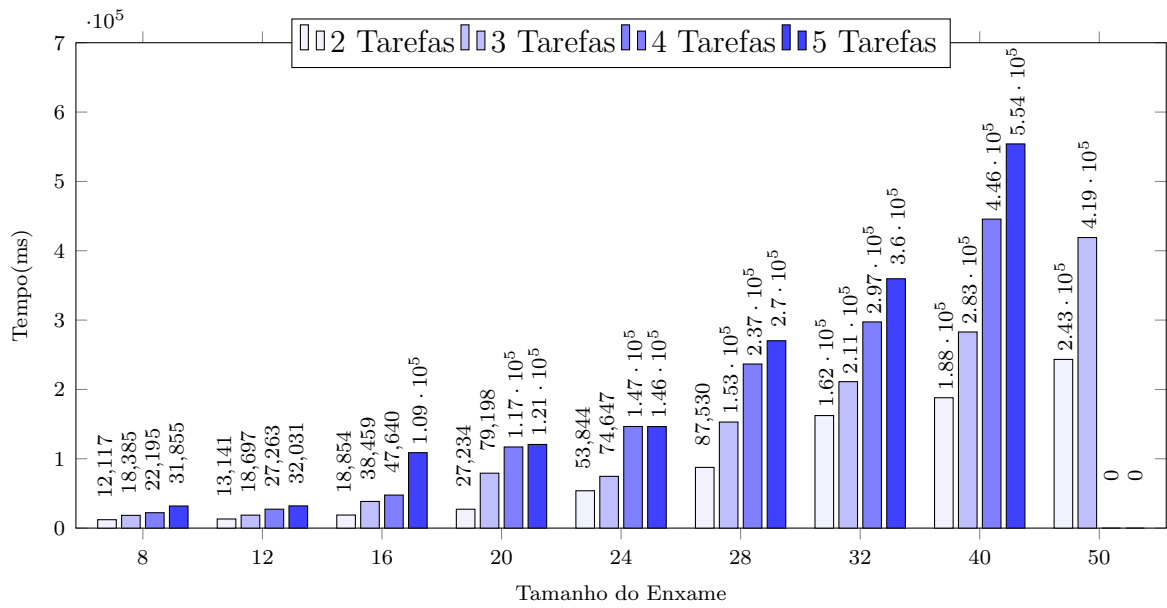


Figura 32: Tempo de convergência obtido pelo algoritmo ADTC

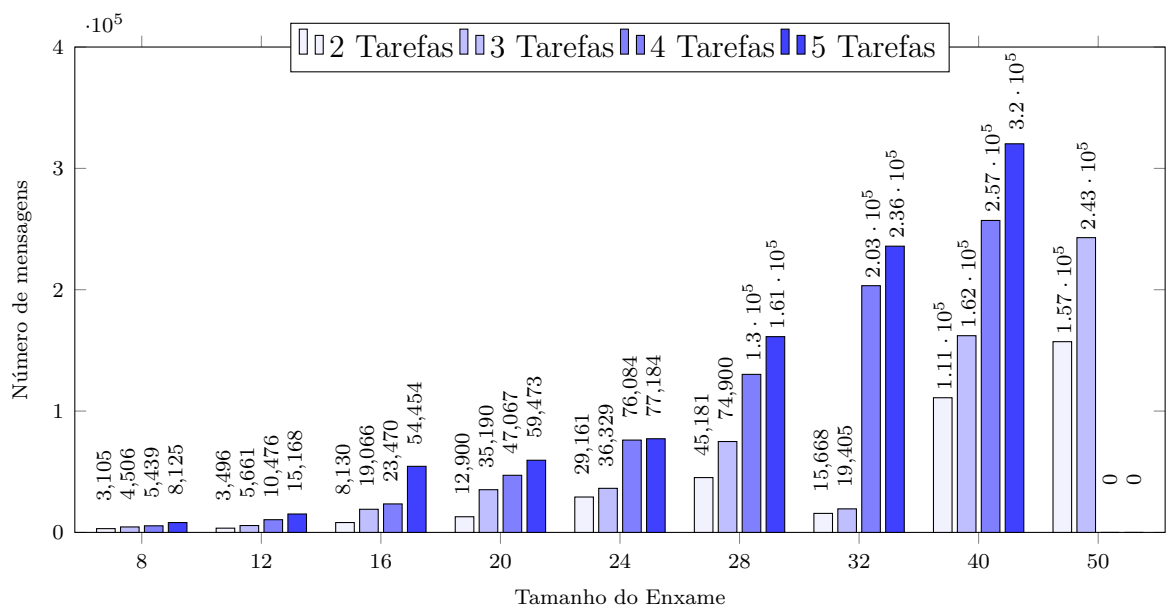


Figura 33: Total de mensagens recebidas obtido pelo algoritmo ADTC

com uma topologia de comunicação em malha completa. O ADTC é um algoritmo similar ao ADTG, porém ao invés de utilizar uma topologia em malha completa, utiliza uma topologia em cluster. O ADTC tem como objetivo otimizar os resultados. O conceito inicial do ADTC consiste que na medida em que é reduzido o fluxo de informação entre os robôs, o desempenho do algoritmo melhora, ou seja, é reduzido o tempo de convergência.

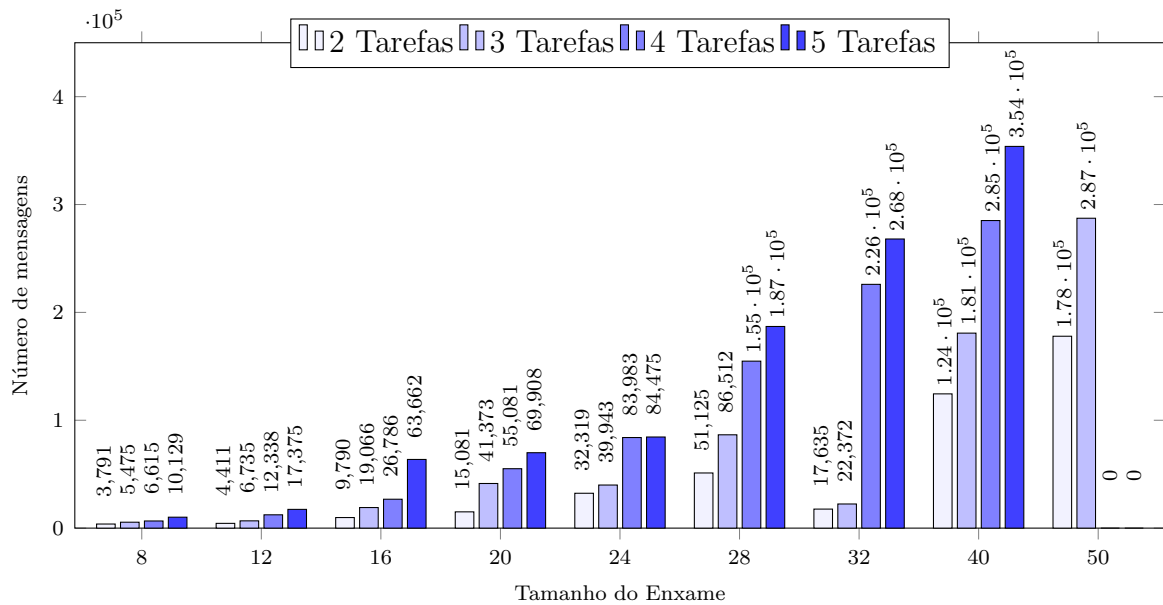


Figura 34: Total de mensagens enviadas obtido pelo algoritmo ADTC

A Figura 35 apresenta as médias do tempo de convergência dos algoritmos ADTC e ADTG, em milissegundos, de 10 amostras para enxames de robôs que variam de 8 à 24 robôs, e para uma alocação que varia de 2 à 5 tarefas. Os resultados em azul referem-se ao desempenho do ADTC e os resultados em vermelho, ao desempenho do ADTG.

O ADTC obteve, na média, uma redução do tempo de convergência de 30,03% em relação ao ADTG. Os ensaios com 8 robôs obtiveram, na média, uma redução de 16,92%. Os ensaios com 12 robôs obtiveram, na média, uma redução de 40,11%. Os ensaios com 16 robôs obtiveram, na média, uma redução de 63,55%. Os ensaios com 20 robôs obtiveram, na média, uma aumento de 10,38%. E os ensaios com 24 robôs obtiveram, na média, uma redução de 39,93%. O melhor desempenho do ADTC em função do ATDG foi com 16 robôs e 4 tarefas, no qual obteve uma redução do tempo de convergência de 77,39%. Já o pior desempenho do ADTC em função do ATDG foi com 20 robôs e 4 tarefas, no qual obteve um aumento do tempo de convergência de 24,73%.

Os ensaios com 8 e 20 robôs obtiveram os piores desempenhos do ADTC. Em relação aos ensaios com 8 robôs, os resultados mostram que a implementação do ADTC é apenas justificável com o aumento da complexidade do problema. E em relação aos ensaios com 20 robôs, o resultado apresentado com o ADTG foge ao comportamento padrão, se observado os resultados dos outros ensaios. O ADTG tem um comportamento no qual

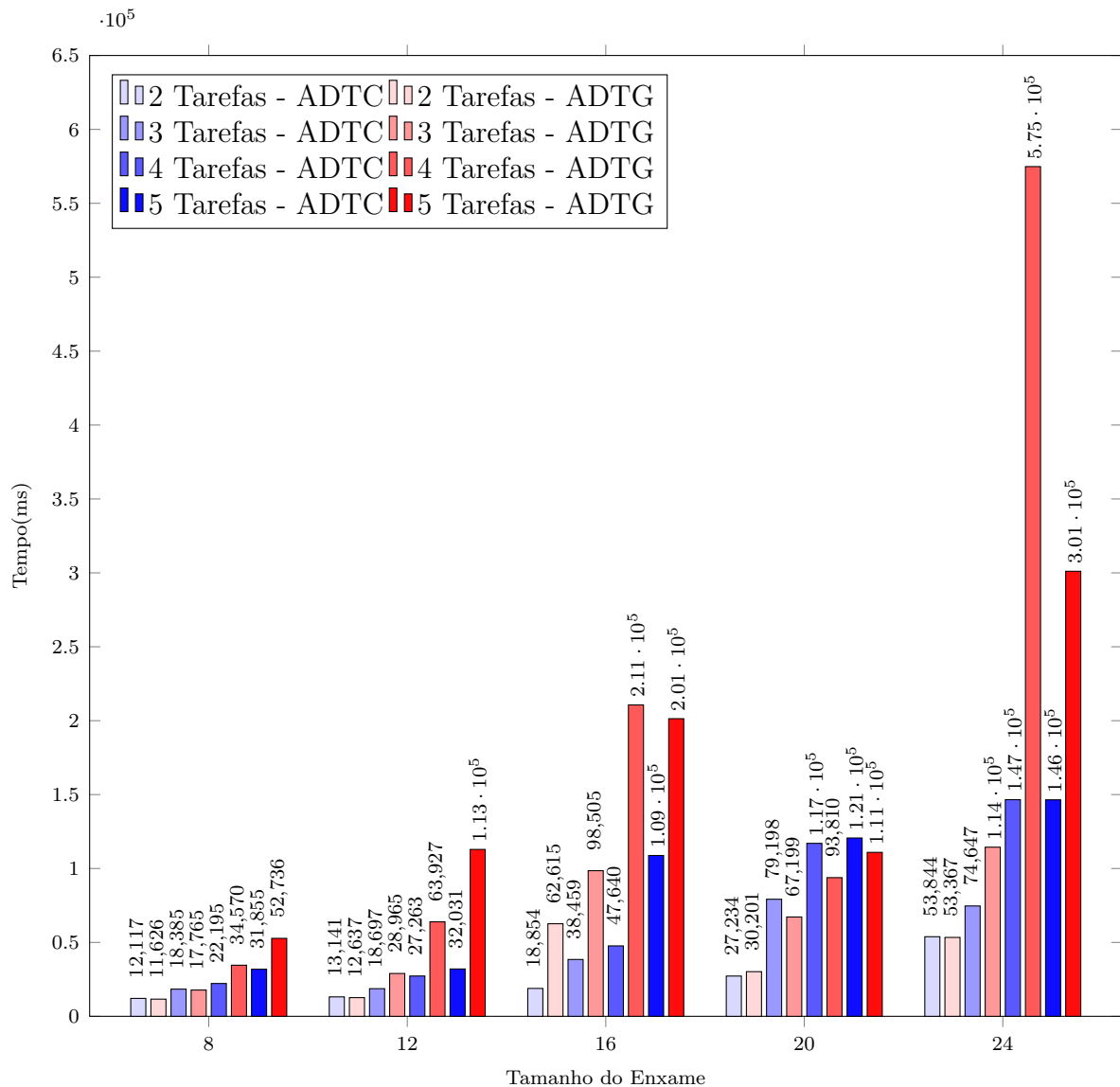


Figura 35: Comparação do tempo de convergência obtidos pelos algoritmos ADTC e ADTG

o tempo de convergência cresce exponencialmente em função do número de robôs, sendo o único grupo de ensaios que foge a esse padrão são os ensaios com 20 robôs. O ADTC também possui esse comportamento de crescimento do tempo de convergência, porém menos acentuado em função do crescimento do enxame.

A Figura 36 apresenta as médias do número de mensagens recebidas pelos algoritmos ADTC e ADTG, de 10 amostras para enxames de robôs que variam de 8 à 24 robôs, e para uma alocação que varia de 2 à 5 tarefas. Os resultados em azul referem-se ao desempenho do ADTC e os resultados em vermelho, ao desempenho do ADTG.

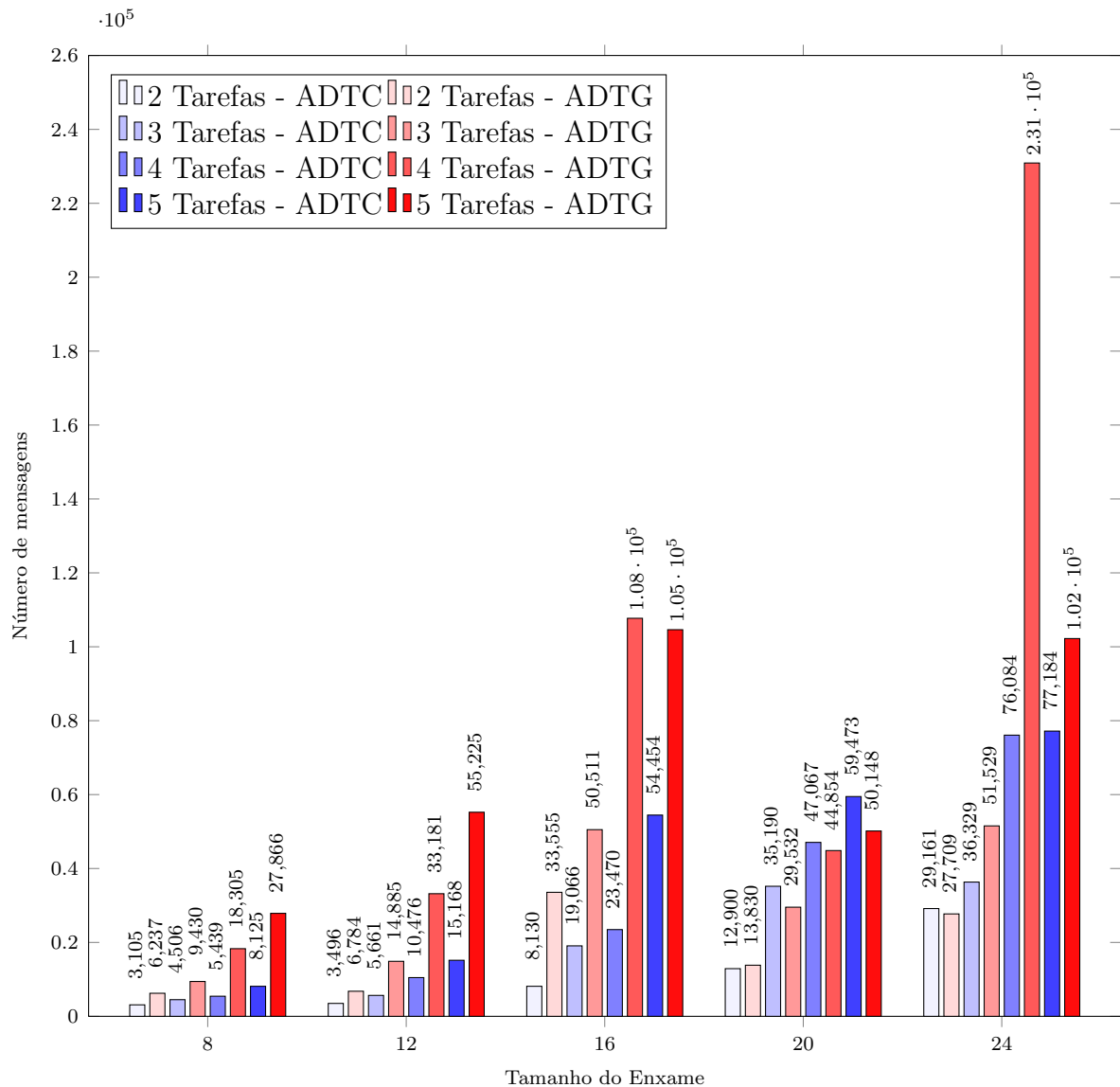


Figura 36: Comparação do número de mensagens recebidas pelos algoritmos ADTC e ADTG

O ADTC obteve, na média, uma redução do número de mensagens recebidas de 41,92% em relação ao ADTG. Os ensaios com 8 robôs obtiveram, na média, uma redução de 60,89%. Os ensaios com 12 robôs obtiveram, na média, uma redução de 62,43%. Os ensaios com 16 robôs obtiveram, na média, uma redução de 66,05%. Os ensaios com 20 robôs obtiveram, na média, um aumento de 9,07%. E os ensaios com 24 robôs obtiveram, na média, uma redução de 39,93%. O melhor desempenho do ADTC em função do ADTG foi com 16 robôs e 4 tarefas, no qual obteve uma redução do número de mensagens recebidas de 78,21%. Já o pior desempenho do ADTC em função do ADTG foi com 20

robôs e 3 tarefas, no qual obteve um aumento do número de mensagens recebidas de 19,16%.

A Figura 37 apresenta as médias do número de mensagens enviadas pelos algoritmos ADTC e ADTG, de 10 amostras para enxames de robôs que variam de 8 à 24 robôs, e para uma alocação que varia de 2 à 5 tarefas. Os resultados em azul referem-se ao desempenho do ADTC e os resultados em vermelho, ao desempenho do ADTG.

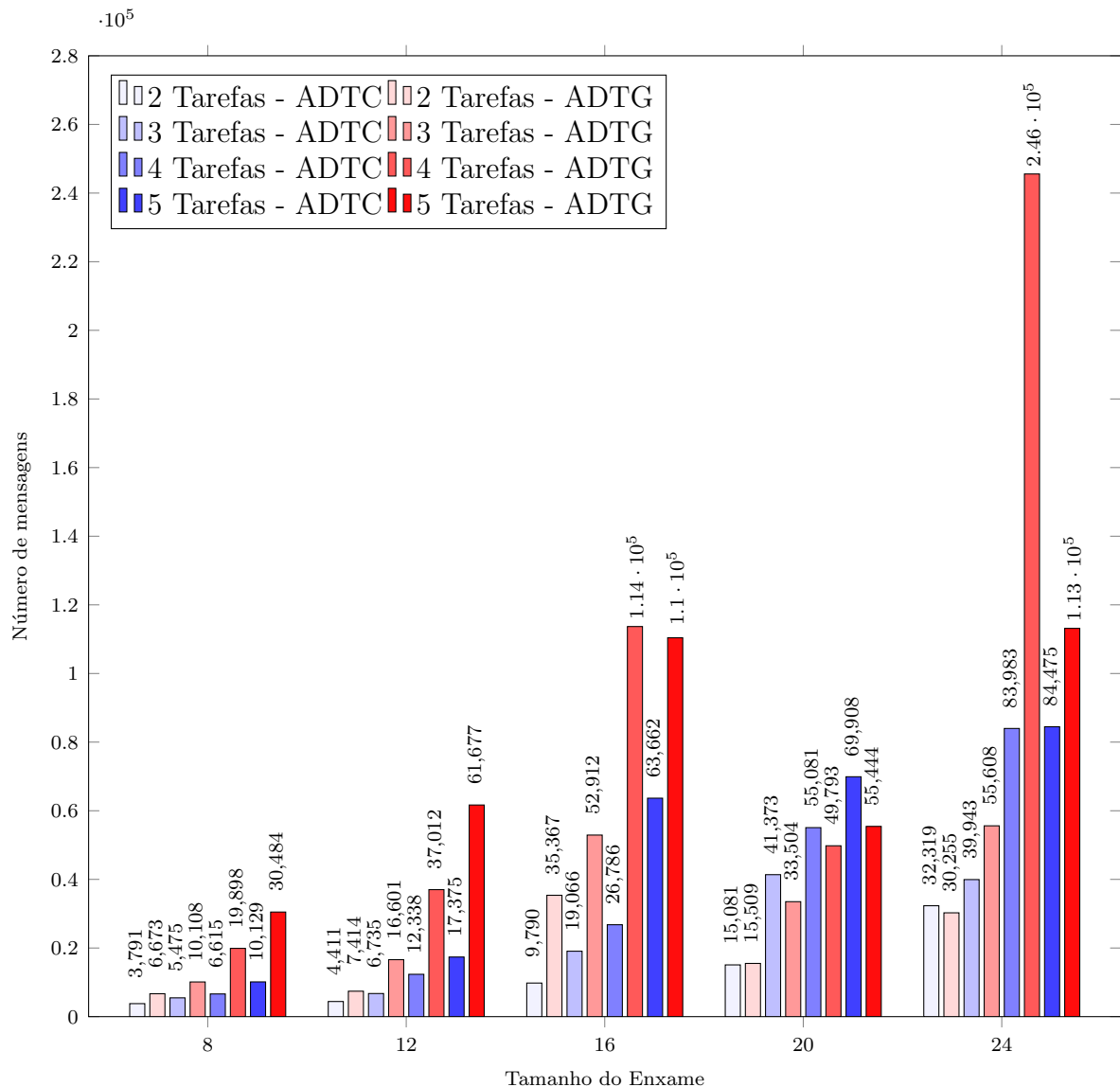


Figura 37: Comparação do número de mensagens enviadas pelos algoritmos ADTC e ADTG

O ADTC obteve, na média, uma redução do número de mensagens enviadas de 38,26% em relação ao ADTG. Os ensaios com 8 robôs obtiveram, na média, uma redução

de 55,64%. Os ensaios com 12 robôs obtiveram, na média, uma redução de 59,61%. Os ensaios com 16 robôs obtiveram, na média, uma redução de 62,29%. Os ensaios com 20 robôs obtiveram, na média, uma aumento de 14,36%. E os ensaios com 24 robôs obtiveram, na média, uma redução de 28,12%. O melhor desempenho do ADTC em função do ATDG foi com 16 robôs e 4 tarefas, no qual obteve uma redução do tempo de convergência de 76,44%. Já o pior desempenho do ADTC em função do ATDG foi com 20 robôs e 4 tarefas, no qual obteve um aumento do tempo de convergência de 26,08%.

Os resultados apresentados em relação à otimização do número de mensagens enviadas e recebidas possui um comportamento similar ao da otimização em relação ao tempo de convergência, confirmando a tese de que a otimização da comunicação aumenta o desempenho do algoritmo. Porém observando os ensaios com 8 robôs, é possível concluir que a topologia em cluster possui um custo operacional de execução de cada ciclo superior ao de malha completo, o que causa um impacto no tempo de convergência. Isto explica porque o resultado apresentado em relação ao número de mensagens enviadas e recebidas é três vezes maior que a otimização do tempo de convergência nos ensaios realizados com 8 robôs.

5.6 Considerações Finais do Capítulo

Neste capítulo foram apresentados os resultados obtidos na implementação do algoritmo ADTC, analisando seu desempenho com diferentes níveis de complexidade. Foi realizada também uma análise sobre as características estruturais da topologia em cluster, como o tamanho ideal de cluster e o número ideal de informantes, além de uma comparação entre o algoritmo ADTC e o ATDG. Esta análise comprovou que a otimização do processo de comunicação pode trazer melhorias ao desempenho do projeto. O ADTC obteve, na média, um desempenho melhor nesta aplicação.

O próximo capítulo encerrará esta dissertação. Este apresenta as principais conclusões, seus aspectos de maior relevância e algumas diretrizes para projetos futuros.

Capítulo 6

CONCLUSÕES E TRABALHOS FUTUROS

A OTIMIZAÇÃO da comunicação na alocação dinâmica de tarefas em enxames de robôs é fundamental para a alocação de grandes enxames. Um dos empecilhos para utilização da inteligência de enxame para a solução de problemas complexos é o fluxo de informação necessário para alocar tarefas em grandes grupos de robôs. A troca de informação é essencial para a realização coordenada das tarefas, pois sem uma alocação eficiente não é possível a solução de um problema complexo. Desta forma, torna-se necessária a implementação de uma comunicação otimizada e inteligente.

6.1 Conclusões

Esta dissertação teve como objetivo otimizar os processos de comunicação na Alocação Dinâmica de Tarefas em um enxame de robôs, com o intuito de melhorar os resultados do algoritmo ADTG, apresentado em (MENDONÇA, 2014). Para tanto, foi proposto o algoritmo de otimização ADTC inspirado no PSO, no qual cada robô representa uma partícula e sua posição no espaço de busca representa uma alocação factível. Sua posição é alterada por uma velocidade que é adaptativa, atualizando-se mediante sua própria experiência e nas dos demais robôs do enxame. O ADTC possui a mesma metodologia do ADTG, modificando-se, porém, o processo de comunicação. Para que a velocidade possa ser adaptativa, existe a necessidade de troca de informações entre os robôs do enxame e o fluxo de informação entre os robôs cresce exponencialmente em função do número de robôs, como no processo de comunicação utilizado pelo ADTG. Por isso o ADTC utiliza, em vez da topologia em malha completa empregada pelo ADTG, a topologia de comunicação em

cluster, com o objetivo de otimizar o processo de comunicação, viabilizando a alocação de grandes grupos de robôs.

A metodologia desta dissertação, com o intuito de alcançar o objetivo acima citado, é estruturada em 5 capítulos. A definição formal do problema de alocação dinâmica de tarefas foi apresentada no Capítulo 1, que aborda também sua complexidade, que está em função da variação do número de robôs e de tarefas a serem alocadas.

Trabalhos relacionados com alocação de tarefas em sistemas com multi-robôs foram apresentados no Capítulo 2, avaliando as características e abordagens utilizadas por cada um dos algoritmos apresentados, bem como suas vantagens e desvantagens.

Nesta dissertação foi proposto um algoritmo para otimização da comunicação na alocação de tarefas, o ADTC, apresentado no Capítulo 3. Este algoritmo bio-inspirado, utiliza como referência o PSO e metodologias semelhantes ao algoritmo ADTG. ADTC possui um processo de comunicação que possibilita a utilização da topologia em cluster.

Escolheu-se como plataforma de implementação os robôs ELISA-3 que possuem características que viabilizam sua utilização nessa aplicação, como seu tamanho reduzido e dispositivo de comunicação em RF. No Capítulo 4 foi apresentado detalhadamente a estrutura dos robôs ELISA-3, seus dispositivos embarcados e suas funcionalidades, além dos processos realizados em cada etapa para a implementação do ADTC nos robôs.

A análise dos resultados obtidos foi apresentada no Capítulo 5 e demonstrou que o desempenho da topologia em cluster - mediante a variação em sua estrutura, como o número de clusters e o número de informantes - valida a equação para o cálculo do número ideal de cluster, elaborada nesta dissertação.

Apresentou-se também os resultados acerca do desempenho do ADTC. A primeira parte dos ensaios teve como objetivo avaliar a capacidade do ADTC implementado no robô ELISA-3. O ADTC nesta etapa demonstrou eficiência em todos os ensaios realizados, excetuando-se os em que haviam 50 robôs. Devido as limitações físicas do robô, não houve convergência para aplicação de 4 e 5 tarefas. A segunda etapa desses ensaios teve como objetivo avaliar a otimização em relação ao ADTG.

O ADTC obteve, na media, redução do tempo de convergência de 30,03%, sendo seu melhor desempenho uma redução do tempo de convergência de 77,39% (para alocação de 4 tarefas em um enxame de 16 robôs) e seu pior desempenho um aumento do tempo de

convergência de 24,73% (para alocação de 4 tarefas em um enxame de 20 robôs). Obteve uma redução, na média, do número de mensagens recebidas de 41,92%, sendo seu melhor desempenho uma redução do número de mensagens recebidas de 78,21% (para alocação de 4 tarefas em um enxame de 16 robôs) e o pior desempenho um aumento do número de mensagens recebidas de 19,16% (para alocação de 4 tarefas em um enxame de 20 robôs). Obteve, na média, uma redução do número de mensagens enviadas de 38,26%, sendo seu melhor desempenho uma redução do número de mensagens enviadas de 76,44% (para alocação de 4 tarefas em um enxame de 16 robôs) e seu pior desempenho um aumento do número de mensagens enviadas de 26,08% (para alocação de 4 tarefas em um enxame de 20 robôs). A topologia de comunicação em clusters demonstra sua capacidade de otimização do processo de comunicação, mostrando melhores resultados que a topologia de comunicação em malha completa. Os resultados comprovam que a escolha da topologia de comunicação, bem como outros parâmetros do processo de comunicação, podem otimizar ou limitar o desempenho de um determinado projeto.

6.2 Trabalhos Futuros

Sugere-se nesta seção algumas propostas que podem trazer melhorias ao processo de alocação dinâmica de tarefas em enxames de robôs, assim como novos ensaios com o objetivo de aprofundar a análise sobre a otimização do processo de comunicação.

Uma limitação na implementação do ADTC é a capacidade operacional do robô ELISA-3. Seria de grande contribuição para esta pesquisa a implementação do ADTC em outros robôs que possuam características estruturais diferentes, como um tempo de funcionamento maior e/ou um processo de comunicação que permita maior paralelismo na troca de mensagens. Também seria de grande contribuição a implementação do ADTC em robôs com diferentes características, expandindo a aplicação do ADTC para enxames heterogêneos.

Poderia ser implementado outros tipos de algoritmos de otimização, como o ACO (*Ant Colony Optimization*) (DORIGO, 2007) ou o BFOA (*Bacterial Foraging Optimization Algorithm*) (DAS et al., 2009). Diferentes algoritmos de otimização podem estimular novos comportamentos ou características até o momento não observadas, assim como alcançar melhores resultados. Também seria de grande valia a implantação do PSO combinado

a outros algoritmos de otimização, podendo, nesse contexto, utilizar-se de algoritmos de outras categorias, fora do grupo de inteligência de enxame, como Sistemas Fuzzy, Algoritmos Genéticos ou Rede Neural Artificial.

Por fim, voltando ao tema central desta dissertação, a otimização da comunicação, poderia ser realizada uma nova proposta na qual seriam utilizadas outras topologias de comunicação, como em anel ou em árvore. Tanto a topologia em anel como em árvore trariam grande redução do fluxo de informação entre os robôs, porém, ambas necessitariam de uma estrutura complexa para o processo de comunicação.

REFERÊNCIAS

- ARDUINO. *Software Arduino 1.0.1*. USA, 2018. Disponível em: <<http://arduino.cc>>.
- ATMEL. *ATmega2560*. USA, 2018. Disponível em: <<http://www.atmel.com>>.
- BENI, G.; WANG, J. Swarm intelligence in cellular robotic systems, proceed. nato advanced workshop on robots and biological systems, tuscan, italy, june 26-30. NY: NATO, 1989.
- BENI, G.; WANG, J. Swarm intelligence in cellular robotic systems. In: *Robots and Biological Systems: Towards a New Bionics?* Heidelberg, Berlin, Springer, 1993. p. 703–712.
- BERMAN, S. et al. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, IEEE, v. 25, n. 4, p. 927–937, 2009.
- BERTSEKAS, D. P. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, Springer, v. 14, n. 1, p. 105–123, 1988.
- BONABEAU, E. et al. Adaptive task allocation inspired by a model of division of labor in social insects. In: *Biocomputing and Emergent Computation: Proceedings of BCEC97*, 1997. p. 36–45.
- BOTELHO, S. C.; ALAMI, R. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: IEEE. *International Conference on Robotics and Automation, 1999*. Detroit, Michigan, 1999. v. 2, p. 1234–1239.
- BRAMBILLA, M. et al. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, Springer, v. 7, n. 1, p. 1–41, 2013.

-
- CARABALLO, L. et al. The block-information-sharing strategy for task allocation: a case study for structure assembly with aerial robots. *European Journal of Operational Research*, Elsevier, 2017.
- CHAND, P.; CARNEGIE, D. A. Development of a reduced human user input task allocation method for multiple robots. *Robotics and Autonomous Systems*, Elsevier, v. 60, n. 10, p. 1231–1244, 2012.
- CHRISTENSEN, D. J.; SCHULTZ, U. P.; STOY, K. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robotics and Autonomous Systems*, Elsevier, v. 61, n. 9, p. 1021–1035, 2013.
- DAS, S. et al. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: *Foundations of Computational Intelligence Volume 3*. Heidelberg, Berlin, Springer, 2009. p. 23–55.
- DORIGO, M. Ant colony optimization. *Scholarpedia*, v. 2, n. 3, p. 1461, 2007.
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: *IEEE. Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. Nagoya, Japan, 1995. p. 39–43.
- FAIRCHILD. *QRE1113 - Miniature Reflective Object Sensor*. USA, 2011. Disponível em: <<http://www.fairchildsemi.com>>.
- FRANKS, N. R. Teams in social insects: group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, Springer, v. 18, n. 6, p. 425–429, 1986.
- FREESCALE. *MMA7455L: Three Axis Low-g Digital Output Accelerometer*. USA, 2009. Disponível em: <<http://www.freescale.com>>.
- GAO, S. *Bio-Inspired Computational Algorithms and Their Applications*. ed. InTech, 2012.
- 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, Utah, v. 23, n. 9, p. 939–954, 2004.

-
- GRAHAM, R. L. *Handbook of combinatorics*. North Holland, Elsevier, 1995.
- GROTH, D.; SKANDIER, T. *Network+ Study Guide: Exam N10-003*. John Wiley & Sons, 2006.
- IOCCHI, L.; NARDI, D.; SALERNO, M. Reactivity and deliberation: a survey on multi-robot systems. In: *Balancing reactivity and social deliberation in multi-agent systems*. Berlin, Alemanha: Springer, 2001. p. 9–32.
- JEVTIC, A. et al. Distributed bees algorithm for task allocation in swarm of robots. *IEEE Systems Journal*, IEEE, v. 6, n. 2, p. 296–304, 2012.
- JONES, C.; MATARIC, M. J. Adaptive division of labor in large-scale minimalist multi-robot systems. In: IEEE. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on Las Vegas, NV, USA*, 2003. v. 2, p. 1969–1974.
- JÜTTNER, A.; MAGI, Á. Tree based broadcast in ad hoc networks. *Mobile Networks and Applications*, Springer-Verlag New York, Inc., v. 10, n. 5, p. 753–762, 2005.
- KEPHART, J. O. et al. A biologically inspired immune system for computers. In: *Artificial Life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*. MIT Press, Cambridge, Massachusetts, 1994. p. 130–139.
- KIM, A. J.; KO, E. Do social media marketing activities enhance customer equity? an empirical study of luxury fashion brand. *Journal of Business Research*, Elsevier, v. 65, n. 10, p. 1480–1486, 2012.
- KREBS, J. R.; DAVIES, N. B. *An introduction to behavioural ecology*. Wiley-Blackwell, Sinauer Associates, 1987.
- LEE, D.-H.; ZAHEER, S. A.; KIM, J.-H. Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems. *IEEE Transactions on Industrial Electronics*, IEEE, v. 61, n. 12, p. 6871–6881, 2014.
- LUO, L.; CHAKRABORTY, N.; SYCARA, K. Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 12, n. 3, p. 876–888, 2015.

- MENDONÇA, R. M. de. *Algoritmos distribuídos para alocação dinâmica de tarefas em enxame de robôs*. 112 p. Dissertação (Mestrado) — UERJ, 2014.
- MENDONÇA, R. M. de; NEDJAH, N.; MOURELLE, L. de M. Efficient distributed algorithm of dynamic task assignment for swarm robotics. *Neurocomputing*, Elsevier, v. 172, p. 345–355, 2016.
- MOHAN, Y.; PONNAMBALAM, S. An extensive review of research in swarm robotics. In: IEEE. *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on Coimbatore, India*. 2009. p. 140–145.
- NEDJAH, N.; MENDONÇA, R. M. de; MOURELLE, L. de M. Pso-based distributed algorithm for dynamic task allocation in a robotic swarm. *Procedia Computer Science*, Elsevier, v. 51, p. 326–335, 2015.
- NORDIC. *nRF24L01+ - Ultra low power 2.4GHz RF Transceiver*. USA, 2018. Disponível em: <<http://www.nordicsemi.com>>.
- PARKER, L. E. *Heterogeneous multi-robot cooperation*. Tese (Doutorado) — Massachusetts Institute of Technology, Massachusetts, 1994.
- PARKER, L. E. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE transactions on robotics and automation*, IEEE, v. 14, n. 2, p. 220–240, 1998.
- REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. In: ACM. *Computer Graphics. SIGGRAPH*. USA, 1987. v. 21, n. 4, p. 25–34.
- SALAMI, N. M. A. Ant colony optimization algorithm. *UbiCC Journal*, v. 4, n. 3, p. 823–826, 2009.
- SARIEL-TALAY, S.; BALCH, T. R.; ERDOGAN, N. Multiple traveling robot problem: A solution based on dynamic task selection and robust execution. *IEEE/ASME Transactions On Mechatronics*, IEEE, v. 14, n. 2, p. 198–206, 2009.
- SEMICONDUCTOR, V. *TCRT1000: Reflective Optical Sensor with Transistor Output*. USA, 2012. Disponível em: <<http://www.vishay.com>>.

- 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.
- SOSINSKY, B. *Networking bible*. John Wiley & Sons, 2009.
- STEELE, F.; THOMAS, G. Directed stigmergy-based control for multi-robot systems. In: IEEE. *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on Arlington, VA, USA*. 2007. p. 223–230.
- TAN, Y.; ZHENG, Z.-y. Research advance in swarm robotics. *Defence Technology*, Elsevier, v. 9, n. 1, p. 18–39, 2013.
- 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 Barcelona, Spain 2005*. p. 1501–1508.
- TEO, S. L. et al. Annual migrations, diving behavior, and thermal biology of atlantic bluefin tuna, *thunnus thynnus*, on their gulf of mexico breeding grounds. *Marine Biology*, Springer, v. 151, n. 1, p. 1–18, 2007.
- WANG, J.; GU, Y.; LI, X. Multi-robot task allocation based on ant colony algorithm. *Journal of Computers*, v. 7, n. 9, p. 2160–2167, 2012.
- WERGER, B. B.; MATARIĆ, M. J. Broadcast of local eligibility for multi-target observation. In: *Distributed autonomous robotic systems 4*. Tokyo, Springer, 2000. p. 347–356.
- ZHANG, D. et al. Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robotics and Autonomous Systems*, Elsevier, v. 55, n. 7, p. 572–588, 2007.
- ZHANG, Y.; LIU, S.-H. Survey of multi-robot task allocation. *CAAI Transactions on Intelligent Systems*, v. 3, n. 2, p. 115–120, 2008.

APÊNDICE A – RESULTADOS NUMÉRICOS OBTIDOS

Neste apêndice são apresentados os resultados numéricos obtidos nos ensaios realizados nesta dissertação. Os ensaios são os resultados obtidos pelo algoritmo ADTC apresentado no Capítulo 3. Os ensaios apresentam os resultados referentes ao tempo de convergência em milissegundos (ms) e o número de mensagens recebidas (Msg. Rec.) e ao número de mensagens enviadas (Msg. Env.). Todos os ensaios possuem 10 amostras e a respectiva média.

A.1 Resultados obtidos na análise do tamanho dos clusters

Nesta seção é apresentado os resultados referentes a análise do tamanho dos clusters. A Tabela 4 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 2 clusters $\gamma = 2$. A Tabela 5 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 3 clusters $\gamma = 3$. A Tabela 6 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 4 clusters $\gamma = 4$. A Tabela 7 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 6 clusters $\gamma = 6$. E a Tabela 8 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 8 clusters $\gamma = 8$.

A.2 Resultados obtidos na análise do número de informantes

Nesta seção são apresentados os resultados referentes a análise do número de informantes. Os ensaios foram realizados em um enxame de 24 robôs $\rho = 24$ divididos em 4 clusters $\gamma = 4$. A Tabela 9 apresenta os resultados de um enxame com 17% de informantes. A Tabela 10 apresenta os resultados de um enxame com 38% de informantes. A Tabela 11 apresenta os resultados de um enxame com 50% de informantes.

A.3 Resultados obtidos no ADTC

Nesta seção são apresentados os resultados para o algoritmo ADTG. A Tabela 12 apresenta os resultados de um enxame de 8 robôs $\rho = 8$ divididos em 2 clusters $\gamma = 2$. A Tabela 13 apresenta os resultados de um enxame de 12 robôs $\rho = 12$ divididos em 4 clusters $\gamma = 4$. A Tabela 14 apresenta os resultados de um enxame de 16 robôs $\rho = 16$ divididos em 4 clusters $\gamma = 4$. A Tabela 15 apresenta os resultados de um enxame de 20 robôs $\rho = 20$ divididos em 5 clusters $\gamma = 5$. A Tabela 16 apresenta os resultados de um enxame de 24 robôs $\rho = 24$ divididos em 6 clusters $\gamma = 6$. A Tabela 17 apresenta os resultados de um enxame de 28 robôs $\rho = 28$ divididos em 7 clusters $\gamma = 7$. A Tabela 18 apresenta os resultados de um enxame de 32 robôs $\rho = 32$ divididos em 8 clusters $\gamma = 8$. A Tabela 19 apresenta os resultados de um enxame de 40 robôs $\rho = 40$ divididos em 10 clusters $\gamma = 10$. E a Tabela 20 apresenta os resultados de um enxame de 50 robôs $\rho = 50$ divididos em 10 clusters $\gamma = 10$.

Tabela 4: Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 2$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	128994	127378	144499	161454	113681	131988	115262	166180	136325	87776	131353,7
MSG. REC	57099	59053	71834	86489	55656	75112	67869	98199	65118	46236	68266,5
MSG. ENV.	70509	68495	97257	97849	66996	102741	76430	111980	78513	50148	82091,8
INT.	5	5	6	8	6	7	5	8	6	4	6
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	144578	109225	267110	198580	139745	120113	217308	193167	105242	153721	164878,9
MSG. REC	814485	42127	116764	93244	83652	42718	100567	74485	48851	95121	151201,4
MSG. ENV.	102554	51235	123046	105405	89105	54244	111135	82878	55259	115586	89044,7
INT.	10	6	15	12	11	11	13	12	6	11	10,7
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	274792	398066	757452	892100	545404	669039	482265	752325	349048	262983	538347,4
MSG. REC	144968	196726	416309	468678	293320	273014	223816	392322	186412	157116	275268,1
MSG. ENV.	175124	214172	468710	505335	356976	330008	285000	471177	228247	195109	322985,8
INT.	16	17	24	27	16	26	23	24	17	16	20,6
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	239296	816550	208842	611114	389929	754486	680777	507801	241562	589240	503959,7
MSG. REC	141855	528185	124795	259559	192424	441248	276038	189246	142356	310407	260611,3
MSG. ENV.	165814	631249	143261	312854	27012	494058	316142	243672	166221	355662	285594,5
INT.	15	28	15	26	17	24	26	16	15	17	19,9

Tabela 5: Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 3$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	82330	70078	92139	111348	87992	79081	128822	73342	102775	85014	91292,1
MSG. REC	51415	37899	40496	61778	48459	38099	73834	40923	49884	45221	48800,8
MSG. ENV.	57901	41445	48965	67951	62267	43907	79986	46204	65273	53049	56694,8
INT.	6	5	7	8	6	6	8	5	8	6	6,5
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	141843	111984	103270	75328	102481	119980	81583	77644	189440	135818	113937,1
MSG. REC	70640	68567	57655	30091	61368	48055	36730	31410	81086	69838	55544
MSG. ENV.	79852	73641	71218	35489	70052	53127	39471	37410	82581	75091	61793,2
INT.	11	11	9	7	9	11	7	7	12	11	9,5
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	284333	385812	249320	461406	175322	470668	583212	177285	386788	557562	373170,8
MSG. REC	149035	183456	132347	195010	101697	268586	294979	106594	203694	325471	196086,9
MSG. ENV.	162252	235537	158505	229172	118248	300455	336555	120775	239581	341443	224252,3
INT.	18	20	15	21	18	21	22	13	20	22	19
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	486269	167073	420886	165032	259953	575856	632984	394267	535096	317376	395479,2
MSG. REC	209119	102291	220147	101325	124145	367430	397131	190852	306422	131421	215028,3
MSG. ENV.	239502	118398	246987	115149	16371	393294	450892	215761	331583	164643	229258
INT.	21	12	20	12	21	22	23	16	20	15	18,2

Tabela 6: Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 4$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	56301	56893	72541	57174	88843	58316	77325	70394	53147	70394	66132,8
MSG. REC	32301	28646	39863	33387	47635	29345	40378	35632	30314	35632	35313,3
MSG. ENV.	36840	32524	44000	37356	53324	34975	45913	42943	33726	42943	40454,4
INT.	6	6	6	6	8	6	8	6	5	7	6,4
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	79457	78235	82314	55124	77231	88652	125457	81985	128452	53181	85008,8
MSG. REC	34325	46124	49531	24325	44237	52326	52314	45124	57321	22436	42806,3
MSG. ENV.	36894	49457	52147	26314	49094	56234	56178	48312	60314	24612	45955,6
INT.	7	7	9	7	8	6	11	9	11	7	8,2
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	358286	292282	353462	123115	196092	288379	357412	199456	331457	274318	277425,9
MSG. REC	171073	146765	209205	71063	96152	144452	211345	97314	201945	140479	148979,3
MSG. ENV.	192332	163568	233719	79984	108168	161389	235478	109314	227618	159721	167129,1
INT.	21	20	21	10	18	20	21	18	21	20	19
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	356731	199964	294709	292282	452132	198360	355467	133659	277653	411326	297228,3
MSG. REC	211326	99316	148312	146765	256214	97349	173468	75214	143498	262450	161391,2
MSG. ENV.	236845	11137	166321	163568	273268	115946	195946	81654	163456	278932	168707,3
INT.	20	19	20	20	21	20	20	11	20	21	19,2

Tabela 7: Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 6$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	58327	50560	51612	58718	52930	53214	50560	52014	51473	59036	53844,4
MSG. REC	25738	30080	30481	27374	30765	30972	30080	30028	30124	25971	29161,3
MSG. ENV.	28444	33084	33572	30444	34016	34541	33084	33716	33347	28943	32319,1
INT.	7	5	6	7	6	6	5	6	6	7	6,1
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	78218	74583	104926	53181	75395	54123	77849	74035	75417	78746	74647,3
MSG. REC	32615	44529	41483	22436	44869	22769	32328	44186	45136	32941	36329,2
MSG. ENV.	35772	48876	45652	24612	49336	25365	35624	48438	49416	36345	39943,6
INT.	8	8	11	7	8	6	8	8	8	8	8
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	135624	132974	221162	80059	191958	130352	83458	81325	190325	218235	146547,2
MSG. REC	81236	79734	132646	32513	80467	77458	34025	32984	79452	130325	76084
MSG. ENV.	89854	87628	146524	35784	88636	86045	38457	36325	87125	143458	83983,6
INT.	12	12	21	12	21	12	10	10	19	21	15
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	70256	79754	130254	189254	225470	72451	193652	229251	195214	79021	146457,7
MSG. REC	42512	47124	77123	78542	136524	42985	81785	137452	81254	46541	77184,2
MSG. ENV.	46457	52369	85684	85624	149857	47145	89214	148124	88521	51762	84475,7
INT.	10	12	15	19	21	10	19	21	19	12	15,8

Tabela 8: Resultados para a análise do número de cluster com $\rho = 24$ e $\gamma = 8$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	52365	51326	56214	51965	57326	58845	55124	51326	59965	53985	54844,1
MSG. REC	30046	30514	25023	30021	24521	25961	25326	29145	26345	31325	27822,7
MSG. ENV.	33845	33214	28444	33452	28265	28736	27154	33215	29562	34985	31087,2
INT.	7	6	6	6	6	7	7	6	7	6	6,4
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	102365	77326	75124	72645	55478	101326	70452	104926	75731	74325	80969,8
MSG. REC	40412	31345	45326	43985	29856	40745	41652	41483	30985	42847	38863,6
MSG. ENV.	44956	35314	49458	45847	30452	43895	45834	45652	32974	46745	42112,7
INT.	11	8	8	8	7	11	8	11	8	8	8,8
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	80945	81945	128945	190532	191114	205557	130315	191451	122654	181622	150508
MSG. REC	31654	33470	80795	80066	81895	128451	81947	80325	71942	68552	73909,7
MSG. ENV.	34978	36345	88674	87418	89143	140997	88860	88004	83449	74891	81275,9
INT.	10	10	12	20	21	21	12	19	12	20	15,7
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	70455	220650	132699	78633	190447	122645	201659	177659	70566	215449	148086,2
MSG. REC	40632	132644	78471	45473	81005	71985	128446	76632	42146	125119	82255,3
MSG. ENV.	45441	144889	86987	50061	87496	81469	139556	82557	50036	130488	89898
INT.	10	21	15	11	20	15	21	17	10	21	16,1

Tabela 9: Resultados para a análise do número de informantes com 17% de informantes

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	56301	56893	72541	57174	88843	58316	77325	70394	53147	70394	66132,8
MSG. REC	32301	28646	39863	33387	47635	29345	40378	35632	30314	35632	35313,3
MSG. ENV.	36840	32524	44000	37356	53324	34975	45913	42943	33726	42943	40454,4
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	79457	78235	82314	55124	77231	88652	125457	81985	128452	53181	85008,8
MSG. REC	34325	46124	49531	24325	44237	52326	52314	45124	57321	22436	42806,3
MSG. ENV.	36894	49457	52147	26314	49094	56234	56178	48312	60314	24612	45955,6
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	358286	292282	353462	123115	196092	288379	357412	199456	331457	274318	277425,9
MSG. REC	171073	146765	209205	71063	96152	144452	211345	97314	201945	140479	148979,3
MSG. ENV.	192332	163568	233719	79984	108168	161389	235478	109314	227618	159721	167129,1
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	356731	199964	294709	292282	452132	198360	355467	133659	277653	411326	297228,3
MSG. REC	211326	99316	148312	146765	256214	97349	173468	75214	143498	262450	161391,2
MSG. ENV.	236845	11137	166321	163568	273268	115946	195946	81654	163456	278932	168707,3

Tabela 10: Resultados para a análise do número de informantes com 38% de informantes

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	59124	55804	59679	89731	87412	72505	58030	80418	59415	74717	69683,5
MSG. REC	34245	30920	33916	48587	43843	36487	29505	40983	30325	41856	37066,7
MSG. ENV.	38248	34400	38313	53963	52176	43587	32849	46831	35412	45760	42153,9
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	57384	84371	90425	13412	83827	78157	54743	85325	79877	130475	75799,6
MSG. REC	25273	51858	53529	58324	36075	45121	22438	45521	47738	53098	43897,5
MSG. ENV.	27392	54545	57639	61438	38738	49683	25742	49412	50347	57301	47223,7
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	128421	295588	298743	348029	198052	208232	364143	281175	379783	371452	287361,8
MSG. REC	73742	147341	151237	205983	98075	98676	219428	143429	179626	214359	153189,6
MSG. ENV.	83235	166230	165389	232170	109466	111281	243328	162436	200025	240198	171375,8
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	369685	372070	140208	305142	431069	291358	203563	202378	303550	467956	308697,9
MSG. REC	176070	226118	76718	152635	270323	146746	101500	99577	155727	262875	166828,9
MSG. ENV.	199864	247976	82797	170110	286742	166517	11315	118679	172973	281739	173871,2

Tabela 11: Resultados para a análise do número de informantes com 50% de informantes

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	56223	74245	63039	93300	61785	76510	60410	83314	89160	61166	71915,2
MSG. REC	31032	37216	31296	50132	36128	42944	30095	41008	44808	35314	37997,3
MSG. ENV.	33845	44502	36403	55997	39701	46675	33703	46874	53116	39365	43018,1
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	14016	56987	59966	77654	83152	133606	85347	87156	87320	93318	77852,2
MSG. REC	60715	22887	26663	45061	48788	54479	46165	53155	36954	55884	45075,1
MSG. ENV.	63834	26411	28433	50214	51706	58447	49332	56291	39419	59830	48391,7
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	212396	363047	293266	380366	205975	379073	390417	305349	309199	132916	297200,4
MSG. REC	100452	212314	145580	219932	100036	223816	183937	151325	157287	77281	157196,0
MSG. ENV.	114063	240214	165035	245001	110998	249654	206026	171014	172170	86731	176090,6
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	140412	211080	294271	310835	471235	209328	362001	315212	448312	385837	314852,3
MSG. REC	79012	101070	147332	158842	2614235	102124	174215	158130	275729	232450	404313,9
MSG. ENV.	85436	120577	167849	171235	290191	116324	200245	175384	290756	257151	187514,8

Tabela 12: Resultados obtidos pelo ADTC para $\rho = 8$ e $\gamma = 2$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	14414	11481	11154	11404	12324	11124	11457	12569	13698	11548	12117,3
MSG. REC	3504	2808	2916	3016	3268	2652	3235	3214	3185	3254	3105,2
MSG. ENV.	4528	3488	3592	3624	3972	3487	3478	3972	4012	3763	3791,6
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	25960	11886	16566	18711	34357	11412	18984	11023	16781	18176	18385,6
MSG. REC	6497	3134	4090	4219	8136	3365	4362	2896	4348	4019	4506,6
MSG. ENV.	8064	3820	4960	5136	9964	3852	5457	3458	5179	4867	5475,7
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	15777	44502	11616	39266	12187	15985	13281	15204	39573	14562	22195,3
MSG. REC	4092	10526	2943	9066	2979	4362	3367	3946	9213	3897	5439,1
MSG. ENV.	5044	12840	3516	11136	3588	5438	3937	4732	11403	4521	6615,5
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	32307	12839	39866	42432	31746	13332	42042	12105	41137	50747	31855,3
MSG. REC	8093	2882	10376	10970	8249	2168	11124	3315	10817	13259	8125,3
MSG. ENV.	10148	3516	12872	13904	10364	2688	13788	3984	13448	16580	10129,2

Tabela 13: Resultados obtidos pelo ADTC para $\rho = 12$ e $\gamma = 4$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	15421	14215	13251	12145	13658	11235	11987	15478	13012	11012	13141,4
MSG. REC	3624	3452	3254	3256	3874	2458	3695	4569	3325	3458	3496,5
MSG. ENV.	4754	4251	3973	4685	4512	3845	3984	5789	4548	3769	4411
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	24652	19843	24913	14539	12495	18352	12785	24913	19741	14741	18697,4
MSG. REC	6436	5309	5885	3802	5338	8420	3742	4885	8845	3952	5661,4
MSG. ENV.	7832	6196	7680	4484	6920	10457	4325	5680	9412	4365	6735,1
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	29765	28863	21012	31028	21606	31425	32136	29036	27395	20369	27263,5
MSG. REC	10230	9845	8547	12325	7846	14521	13845	10469	9438	7698	10476,4
MSG. ENV.	11412	11240	9659	15369	8932	17439	15739	13035	11358	9197	12338
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	30125	44928	38048	21184	36566	37174	24398	29047	36831	22011	32031,2
MSG. REC	15452	22499	18654	10862	17382	20037	5221	12411	20037	9127	15168,2
MSG. ENV.	16236	25348	21512	11764	20156	23036	6096	15916	21996	11696	17375,6

Tabela 14: Resultados obtidos pelo ADTC para $\rho = 16$ e $\gamma = 4$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	19562	18174	19781	19188	18128	18895	19012	18210	19234	18352	18853,6
MSG. REC	8726	8360	8249	7100	7676	7978	8302	8389	8102	8420	8130,2
MSG. ENV.	10144	10048	10008	8652	9260	9546	9789	10124	9875	10457	9790,3
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	28557	40232	39358	43134	40700	40123	43457	28785	39012	41236	38459,4
MSG. REC	12871	22337	17086	21352	21441	21245	21532	13562	16874	22365	19066,5
MSG. ENV.	14888	26160	19980	24980	24972	24546	25236	15478	19735	26025	22200
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	27624	28995	97854	39603	74215	28325	73875	39985	27135	38795	47640,6
MSG. REC	15382	15639	42659	21514	33215	15765	32683	21973	14856	21023	23470,9
MSG. ENV.	17944	18404	49760	25140	35421	18329	35179	25573	17547	24569	26786,6
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	195139	72422	120247	131498	63404	119458	130784	64739	71783	118456	108793
MSG. REC	86694	35092	67416	58607	35986	66378	57681	36726	34538	65423	54454,1
MSG. ENV.	99880	40956	79076	68764	41564	78369	67649	42168	39784	78419	63662,9

Tabela 15: Resultados obtidos pelo ADTC para $\rho = 20$ e $\gamma = 5$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	27933	25343	31846	25562	26374	27412	26784	24758	31145	25179	27233,6
MSG. REC	10653	13323	17325	13799	10135	10356	10476	13245	16457	13234	12900,3
MSG. ENV.	12384	15728	18973	16448	12120	12045	12345	15896	18685	16186	15081
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	92955	61817	42866	98788	119421	97654	42012	43574	98317	94578	79198,2
MSG. REC	36674	24563	23657	38998	67347	37124	23249	23986	38452	37854	35190,4
MSG. ENV.	43024	28916	27580	45544	78372	44621	27378	28134	45038	45124	41373,1
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	94519	145885	171492	60113	157147	60113	145885	145885	94519	94519	117007,7
MSG. REC	52142	59424	23761	24343	63827	24343	59424	59424	52142	52142	47097,2
MSG. ENV.	60724	69448	28012	28716	74852	28716	69448	69448	60724	60724	55081,2
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	140330	162576	191918	55225	72511	141654	56472	164235	144689	76745	120635,5
MSG. REC	65704	75649	107339	30344	35098	66124	31457	77458	68439	37124	59473,6
MSG. ENV.	77588	89440	125572	35600	40936	78746	35987	91325	79638	44253	69908,5

Tabela 16: Resultados obtidos pelo ADTC para $\rho = 24$ e $\gamma = 6$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	58327	50560	51612	58718	52930	53214	50560	52014	51473	59036	53844,4
MSG. REC	25738	30080	30481	27374	30765	30972	30080	30028	30124	25971	29161,3
MSG. ENV.	28444	33084	33572	30444	34016	34541	33084	33716	33347	28943	32319,1
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	78218	74583	104926	53181	75395	54123	77849	74035	75417	78746	74647,3
MSG. REC	32615	44529	41483	22436	44869	22769	32328	44186	45136	32941	36329,2
MSG. ENV.	35772	48876	45652	24612	49336	25365	35624	48438	49416	36345	39943,6
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	135624	132974	221162	80059	191958	130352	83458	81325	190325	218235	146547,2
MSG. REC	81236	79734	132646	32513	80467	77458	34025	32984	79452	130325	76084
MSG. ENV.	89854	87628	146524	35784	88636	86045	38457	36325	87125	143458	83983,6
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	70256	79754	130254	189254	225470	72451	193652	229251	195214	79021	146457,7
MSG. REC	42512	47124	77123	78542	136524	42985	81785	137452	81254	46541	77184,2
MSG. ENV.	46457	52369	85684	85624	149857	47145	89214	148124	88521	51762	84475,7

Tabela 17: Resultados obtidos pelo ADTC para $\rho = 28$ e $\gamma = 7$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	98313	88767	80481	79362	93196	92417	91147	76607	84572	90435	87529,7
MSG. REC	43541	49081	45154	43525	38948	51030	50675	43297	35453	51113	45181,7
MSG. ENV.	49444	55889	50780	49900	44319	57611	57278	48624	40005	57401	51125,1
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	136155	97242	157066	191249	129742	127238	171171	180565	180793	158281	152950,2
MSG. REC	73105	50664	82391	80466	69662	50672	89913	95370	71779	84986	74900,8
MSG. ENV.	83827	59308	94974	93298	79879	58382	104540	110066	83397	97450	86512,1
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	143399	349654	102653	341865	251250	249565	99194	356583	316600	155390	236615,3
MSG. REC	87364	207061	49146	168882	125434	137776	48660	196857	187487	93745	130241,2
MSG. ENV.	104292	245678	58478	200106	142625	167010	58078	238628	222453	110627	154797,5
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	347149	444090	201762	219516	321802	120929	190122	145303	359229	352234	270213,6
MSG. REC	216398	230929	128127	140776	246912	60943	120735	75030	188782	204691	161332,3
MSG. ENV.	251462	267983	148083	164588	285820	71020	139540	87705	210229	243007	186943,7

Tabela 18: Resultados obtidos pelo ADTC para $\rho = 32$ e $\gamma = 8$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	177895	230609	137153	125209	141344	228664	134331	139554	181167	127147	162307,3
MSG. REC	17508	22242	13974	13109	12919	20314	11954	11677	18664	14325	15668,6
MSG. ENV.	19692	24844	15732	14700	14640	22954	14325	13448	20846	15178	17635,9
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	249185	166713	175083	159982	204268	349227	165709	195773	211768	234259	211196,7
MSG. REC	24784	13252	18739	13891	19991	32651	13172	16999	17902	22674	19405,5
MSG. ENV.	28377	15583	20273	16073	22799	37497	15489	19669	21905	26064	22372,9
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	358636	154484	186363	379091	417042	381498	162334	405662	333215	195332	297365,7
MSG. REC	247710	88389	132427	321192	240292	324338	90648	236214	214558	137445	203321,3
MSG. ENV.	276100	99104	148508	358351	267468	360124	101325	252314	244326	152369	225998,9
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	272658	181517	466446	368140	290873	406243	483049	154953	423279	549072	359623,0
MSG. REC	184060	96292	285328	241560	194438	224725	264408	84224	341866	441951	235885,2
MSG. ENV.	210756	109899	331755	274914	220089	245095	300507	96423	387577	503461	268047,6

Tabela 19: Resultados obtidos pelo ADTC para $\rho = 40$ e $\gamma = 10$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	241189	146362	244615	180923	192338	163331	156855	251381	132808	170260	188006,2
MSG. REC	117000	81146	146785	108712	81860	113732	115956	150931	98720	94956	110979,8
MSG. ENV.	132634	91259	166829	119990	93010	127613	126713	173682	107849	105015	124459,4
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	281151	391658	287538	215802	211658	283027	391805	283947	208326	273645	282855,7
MSG. REC	117266	186193	196215	117252	115236	193137	230538	193765	110326	161139	162106,7
MSG. ENV.	131880	208919	217918	130519	128659	214499	262581	215196	121547	176040	180775,8
$\tau = 4$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	328446	287538	409918	447353	461440	568855	405409	403678	412931	730429	445599,7
MSG. REC	174886	196215	274132	263222	266290	265024	271117	269959	168787	421189	257082,1
MSG. ENV.	192687	217918	301347	299808	287946	294337	298032	296759	187884	474835	285155,3
$\tau = 5$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	287538	451032	609762	447353	804697	1067349	605230	379833	287538	599748	554008
MSG. REC	196215	235356	351608	263222	367402	603159	342283	254013	196215	393060	320253,3
MSG. ENV.	217918	256666	396392	299808	403875	673043	366343	279230	217918	427671	353886,4

Tabela 20: Resultados obtidos pelo ADTC para $\rho = 50$ e $\gamma = 10$

$\tau = 2$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	212862	186049	388709	338632	192336	330458	196779	188659	207665	190620	243276,9
MSG. REC	151246	141117	250129	207349	109457	202459	111325	142314	148326	108159	157188,1
MSG. ENV.	171456	155756	286256	238092	122301	235378	126489	157119	165214	120900	177896,1
$\tau = 3$	Ensaio1	Ensaio2	Ensaio3	Ensaio4	Ensaio5	Ensaio6	Ensaio7	Ensaio8	Ensaio9	Ensaio10	Média
TEMPO (ms)	505925	389107	203230	745832	263411	524657	204941	598652	217593	536976	419032,4
MSG. REC	278806	214552	138734	412548	169328	294780	139136	359124	111117	310983	242910,8
MSG. ENV.	337929	263295	161633	485326	199086	355487	162145	401235	131107	375672	287291,5