

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Rafael José Fonseca de Sá

Otimização de um Sistema de Patrulhamento por Múltiplos Robôs Utilizando Algoritmo Genético

Juiz de Fora
2016

Rafael José Fonseca de Sá

Otimização de um Sistema de Patrulhamento por Múltiplos Robôs Utilizando Algoritmo Genético

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: André Luís Marques Marcato, D.Sc.
Coorientador: Ivo Chaves da Silva Junior, D.Sc.

**Juiz de Fora
2016**

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Fonseca de Sá, Rafael José.

Otimização de um Sistema de Patrulhamento Por Múltiplos Robôs Utilizando Algoritmo Genético / Rafael José Fonseca de Sá. - 2016.

72 p. : il.

Orientador: André Luís Marques Marcato

Coorientador: Ivo Chaves da Silva Junior

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós Graduação em Engenharia Elétrica, 2016.

1. Sistema de Patrulhamento. 2. Otimização. 3. Algoritmo Genético. 4. Multi-robô. I. Marcato , André Luís Marques, orient. II. Silva Junior, Ivo Chaves da, coorient. III. Título.

Rafael José Fonseca de Sá

Otimização de um Sistema de Patrulhamento por Múltiplos Robôs Utilizando Algoritmo Genético

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Eletrônicos, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Professor André Luís Marques Marcato, D.Sc.– Orientador
Universidade Federal de Juiz de Fora

Professor Ivo Chaves da Silva Junior, D.Sc.– Coorientador
Universidade Federal de Juiz de Fora

Professor Fernando Luiz Cyrino Oliveira, D.Sc.
Pontifícia Universidade Católica do Rio de Janeiro

Professor Leonardo Willer de Oliveira, D.Sc.
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Primeiramente, a Deus pelas oportunidades a mim dadas e por colocar ótimas pessoas em minha vida.

Aos meus pais, José Arimateia e Neiry de Fatima, pelo amor incondicional e pela oportunidade de realizar este sonho. Amo muito vocês.

À minha namorada, noiva e enfim esposa, Marilana, pelo companheirismo e principalmente pelo amor oferecido. Agradeço a Deus por ter você em minha vida.

Às minhas irmãs, Danielle e Ana Cristina; aos meus sobrinhos, Caio e Amanda, e seu pai, Vecson Caio, pelo apoio, carinho e amizade.

Aos meus avós, Maria Neuza, Sebastião, Maria Darc (in memoriam) e Alberto, pelo amor e apoio em todos os momentos. Amo vocês.

À minha madrinha, Fatima, pelo carinho e orações.

A todos meus familiares, pelos incentivos e pela torcida para que este sonho se concretizasse.

Aos meus amigos, pela amizade, apoio e palavras de incentivo e carinho. Agradeço a Deus por ter colocado vocês em meu caminho.

Ao Acampamento Juvenil de Juiz de Fora. Muito obrigado pela oração, apoio, carinho e principalmente por estarem sempre ao meu lado em todos os momentos.

A todos os amigos do GRIn e do GOHB, que contribuíram de forma efetiva para o desenvolvimento deste trabalho.

À banca examinadora, pelo tempo dedicado à correção do trabalho e pelas sugestões.

Aos meus orientadores, pelo empenho, motivação, paciência e credibilidade, muito obrigado por tudo.

"Peçam, e lhes será dado; busquem, e encontrarão; batam, e a porta lhes será aberta. Pois tudo o que pede, recebe; o que busca, encontra; e aquele que bate, a porta será aberta. ”

(MATEUS 7:7,8)

RESUMO

Com a evolução da tecnologia, estão aumentando as aplicabilidades dos robôs em nosso meio. Em alguns casos, a utilização de sistemas com múltiplos robôs autônomos trabalhando em cooperação se torna uma ótima alternativa. Há várias pesquisas em andamento na área de robótica com o intuito de aprimorar estas tarefas. Entre estas pesquisas estão os sistemas de patrulhamento. Neste trabalho, o sistema de patrulhamento utilizando múltiplos robôs é implementado considerando a série de chegada de alertas nas estações de monitoramento e o robô pode andar somente em uma única direção. Devido ao número de estações que podem entrar em alerta e ao número de robôs, o controle desse sistema se torna complexo. Como a finalidade de um sistema de patrulhamento é atender possíveis alertas de invasores, é imprescindível que haja uma resposta rápida do controlador responsável para que um robô logo seja encaminhado com o propósito de atender a esse alerta. No caso de sistemas com múltiplos robôs, é necessário que haja uma coordenação do controlador para que os robôs possam atender o máximo de alertas possíveis em um menor instante de tempo. Para resolver esse problema, foi utilizado um controlador composto por uma técnica inteligente de otimização bioinspirada chamada de “Algoritmo Genético” (AG). Este controlador centraliza todas as decisões de controle dos robôs, sendo responsável por orientá-los em relação aos movimentos e captação de informação. As decisões são tomadas com o intuito de maximizar a recompensa do sistema. Esta recompensa é composta pelo ganho de informação do sistema e por uma penalização gerada pela demora em atender aos alertas ativados. Foram feitas simulações com a intenção de verificar a eficácia desse controlador, comparando-o com um controlador utilizando heurísticas pré-definidas. Essas simulações comprovaram a eficiência do controlador via Algoritmo Genético. Devido ao fato do controlador via AG analisar o sistema como um todo enquanto que o controlador heurístico analisa apenas o estágio atual, foi possível observar que a distribuição dos robôs no mapa permitia um atendimento mais ágil às estações com alerta ativados, assim como uma maior aquisição de informações do local. Outro fato importante foi em relação à complexidade do sistema. Foi notado que quanto maior a complexidade do sistema, ou seja, quanto maior o número de robôs e de estações, melhor era a eficiência do controlador via Algoritmo Genético em relação ao controlador heurístico.

Palavras-chave: Sistema de Patrulhamento, Otimização, Algoritmo Genético, Multi-robô

ABSTRACT

New technologies have been considerable advances, and consequently, thus allows the robot appearance as an integral part of our daily lives. In recent years, the design of cooperative multi-robot systems has become a highly active research area within robotics. Cooperative multi-robot systems (MRS) have received significant attention by the robotics community for the past two decades, because their successful deployment have unquestionable social and economical relevance in many application domain. There are several advantages of using multi-robot systems in different application and task. The development and conception of patrolling methods using multi-robot systems is a scientific area which has a growing interest. This work, the patrol system using multiple robots is implemented considering the series of arrival of alerts in the monitoring stations known and the robot was limited to move in one direction. Due to the large number of stations that can assume alert condition and due to the large number of robots, the system control becomes extremely complex. Patrol systems are usually designed for surveillance. An efficient controller permits a patrol in a way that maximizes their chances of detecting an adversary trying to penetrate through the patrol path. The obvious advantage of multi-robot exploration is its concurrency, which can greatly reduce the time needed for the mission. Coordination among multiple robots is necessary to achieve efficiency in robotic explorations. When working in groups, robots need to coordinate their activities. However, a Genetic Algorithm approach was implemented to carryout an optimized control action provided from the controller. In fact the controller determines the robot's behavior. The decision strategies are implemented in order to maximize the system response. The present work deals with a computational study of controller based on Genetic Algorithm and it comparison with another controller based pre-defined heuristics. The simulation results show the efficiency of the proposed controller based on Genetic Algorithm, when compared with the controller based on heuristics. The right decisions from the controller based on Genetic Algorithm allowed a better distribution of the robots on the map leading to fast service stations with active alert, as well as increased acquisition of location information. Another important fact was regarding the complexity of the system. Also, as a result, it was noticed an excellent efficiency of the controller based on Genetic Algorithm when the existence of the large number of robots and stations.

Keywords: Multi-Robot Patrolling, Optimization, Genetic Algorithm, Multi-Robot Systems.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo do Sistema de Patrulhamento.	12
Figura 2 – Relação entre o Ganho de Informação em Relação ao Tempo de Permanência do Robô em uma Estação.	16
Figura 3 – Fluxograma do Sistema de Patrulhamento.....	18
Figura 4 – O Sistema de Patrulhamento é Tratado Como Uma Caixa Preta Pelo AG.....	20
Figura 5 – Fluxograma do AG Utilizado Para Otimizar o Sistema de Patrulhamento Utilizando Múltiplos Robôs.....	22
Figura 6 – Composição de Um Indivíduo para a População.	23
Figura 7 – Exemplo da Técnica da Roleta.....	25
Figura 8 – Processo de Crossover.	26
Figura 9 – Exemplo de Mutação em Um Indivíduo da População.....	27
Figura 10 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.....	30
Figura 11 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 25$	31
Figura 12 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 50$	32
Figura 13 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 75$	33
Figura 14 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dois Robôs e Oito Estações.	34
Figura 15 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.....	35
Figura 16 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 25$	36
Figura 17 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 50$	37
Figura 18 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 75$	38
Figura 19 – Resultado Final da Evolução da Recompensa e da Posição Final dos Cinco Robôs e Vinte Estações.....	39
Figura 20 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.....	40

Figura 21 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 25$	41
Figura 22– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 50$	42
Figura 23 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 75$	43
Figura 24 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dez Robôs e Trinta Estações.....	44
Figura 25 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG Utilizando o Máximo de 200 Gerações e 200 de População.....	45
Figura 26 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG Utilizando o Máximo de 2000 Gerações e 2000 de População.....	46
Figura 27 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 25$	47
Figura 28– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 50$	48
Figura 29– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 75$	48
Figura 30 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dois Robôs e Oito Estações.	49

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
FOB	Função Objetivo
IBGE	Instituto Brasileiro de Geografia e Estatística
VANT	Veículo Aéreo Não Tripulado

LISTA DE SÍMBOLOS

N	Número total de estações
M	Número de estações de alerta
K	Estágio
J	Número de robôs
$x_j(k)$	Posição do robô j , no estágio k
$d_j(k)$	Tempo de permanência do robô j em uma estação, no estágio k
I	Número da estação de alerta
$Y_i(k)$	Chegada de alerta na estação i , no estágio k
α	Taxa de alerta por unidade de tempo
P_j	Probabilidade de uma estação receber um alerta
$u_j(k)$	Variável de controle para o robô j , no estágio k
$x_j(k + 1)$	Posição do robô j , no estágio futuro
$d_j(k + 1)$	Tempo de permanência do robô j em uma estação, no estágio futuro
$A_i(k)$	Status de alerta em cada estação i , no estágio k
$A_i(k + 1)$	Status de alerta em cada estação i , no estágio futuro
X_i	Posição da estação i
$\delta(i, j)$	Delta de Kronecker
D	Tempo máximo em que o robô pode permanecer em uma estação
$R^u(k)$	Recompensa, no estágio k
$I(d_j)$	Ganho de informação em função do tempo de permanência do robô em uma estação
β	Constante de ganho da recompensa
μ	Micro

SUMÁRIO

	LISTA DE ILUSTRAÇÕES	iv
	LISTA DE ABREVIATURAS E SIGLAS	vi
	LISTA DE SÍMBOLOS	vii
	SUMÁRIO	viii
1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DO TRABALHO	3
1.3	REVISÃO BIBLIOGRÁFICA	4
1.3.1	Otimização.....	7
1.3.2	Algoritmo Genético.....	9
1.4	ORGANIZAÇÃO DO TRABALHO	10
2	FORMULAÇÃO DO PROBLEMA.....	12
3	SOLUÇÃO DO PROBLEMA	19
3.1	CONTROLE UTILIZANDO HEURÍSTICA PRÉ-DEFINIDA	19
3.2	CONTROLE VIA ALGORITMO GENÉTICO (AG)	20
3.2.1	Declaração das Variáveis Para Inicializar o AG	22
3.2.2	Avaliação da Função Aptidão.....	24
3.2.3	Seleção dos Indivíduos.....	24
3.2.4	Crossover	25
3.2.5	Mutação	26
3.2.6	Seleção da Nova População.....	27
3.2.7	Critério de Parada	28
4	RESULTADOS	29
4.1	PRIMEIRO CASO	30

4.2	SEGUNDO CASO	34
4.3	TERCEIRO CASO	39
4.4	QUARTO CASO	45
4.5	CONCLUSÃO DO CAPÍTULO	50
5	CONCLUSÃO E TRABALHOS FUTUROS.....	52
5.1	SUGESTÕES PARA TRABALHOS FUTUROS.....	53
	REFERÊNCIAS.....	55

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Com o desenvolvimento tecnológico, cada dia é mais comum o uso de veículos autônomos para realizar determinadas tarefas cotidianas. E tal fato só é possível devido ao avanço nos estudos de áreas como eletrônica, computação, controle e automação. Esse avanço se deve em parte ao interesse de governos e empresas pelo grande potencial de aplicações em áreas civis e militares (GHADIRY et al., 2016).

A utilização de veículos autônomos em algumas tarefas pode significar um avanço expressivo em termos de eficiência, redução de custos e precisão na execução. Para exemplificar tais fatos, há várias aplicações em que veículos autônomos estão sendo utilizados com esse intuito (TEIXEIRA, 2015).

Entre os vários exemplos estão os robôs de limpeza. Com a correria do dia a dia, sobra pouco tempo para cuidar da casa. Para resolver esse problema, há no mercado robôs aspiradores de pó compactos, que prometem de maneira autônoma efetuar a limpeza da casa ou escritório.

Outro exemplo é a utilização dos robôs para transporte de carga, utilizados em algumas indústrias. Além de facilitar o transporte das cargas dentro da indústria, todo o processo ocorre de forma autônoma e alguns modelos permitem uma precisão de $1mm$ (um milímetro) de erro. Com seus diversos sensores, esses robôs são capazes de detectar a aproximação de pessoas evitando possíveis colisões, garantindo a segurança dos trabalhadores a sua volta.

A empresa *Amazon* anunciou em 2013 que estava testando a realização de entregas de encomendas através de drones. Com isso, esperava reduzir o tempo de entrega de seus produtos para apenas 30 minutos (AMAZON PRIME AIR, 2016). Neste ano, a *Amazon* fez uma parceria com o governo do Reino Unido para começar a testar as entregas utilizando os drones (LISA VAAS, 2016).

Há, ainda, os robôs garçons. Em 2013, um restaurante na China foi inaugurado com essa inovação. Estes robôs autônomos tinham a capacidade de cozinhar, entregar pedidos e levar menus para os clientes (DA AGÊNCIA EFE, 2016).

Além das aplicações supracitadas, há outras várias nas quais a robótica móvel pode ser aplicada, tais como: monitoramento de incêndios (MARJOVI et al., 2009), operações de

patrulhamento (ELMALIACH et al., 2008; GHADIRY et al., 2016), reconhecimento de terrenos (CLARK; FIERRO, 2007), entre outros.

Em alguns exemplos dos casos citados anteriormente, o uso de múltiplos robôs trabalhando de maneira cooperativa pôde apresentar vantagens para a conclusão da missão de maneira mais eficiente. Tal rendimento pode reduzir consideravelmente o tempo de realização dessas missões.

O presente trabalho foi inspirado no artigo *UAV perimeter patrol operations optimization using efficient dynamic programming* (KRISHNAMOORTHY et al., 2011). O referido trabalho apresenta um sistema de patrulhamento que utiliza múltiplos robôs. Nesse sistema, foi levantado um modelo matemático no qual os robôs só podem se mover de uma estação à outra e somente podem parar dentro das estações. Outra restrição do modelo é que o robô não pode voltar para uma estação anterior, sendo definido que o mesmo somente pode andar para a que está à sua frente. Essas restrições são impostas para que nenhuma estação fique muito tempo sem ser atendida por um robô.

Como o objetivo deste trabalho é otimizar o sistema, é necessário avaliar a eficiência com que é feito o patrulhamento. Para tal, foi definida uma função objetivo (FOB) que engloba duas informações essenciais a um sistema de patrulhamento, que são:

- Aquisição de informações para que um operador possa definir se aquele alerta foi causado por algum intruso ou apenas um erro do equipamento;
- Reduzir o tempo para atender um alerta ativado em qualquer estação.

Para que esse sistema de patrulhamento possa ter um ponto de máxima eficiência, ou seja, para que consiga maximizar a aquisição de informações e minimizar o tempo de atendimento a uma estação com alerta ativado, é necessário que os robôs trabalhem de maneira cooperativa. Isso possibilita que os robôs atendam de forma rápida as estações e que cada robô capture o máximo de informação das mesmas.

Com este mesmo modelo matemático, o presente trabalho apresenta como solução para otimizar este sistema um controlador centralizado que será responsável por coordenar os robôs de maneira cooperativa. Esse controlador terá seus parâmetros definidos através de um algoritmo heurístico de busca bioinspirado, o “Algoritmo Genético” (AG).

1.2 OBJETIVOS DO TRABALHO

Considere como exemplo um *shopping*, onde agentes de segurança efetuem o patrulhamento à noite com o intuito de evitar furtos. Neste *shopping* há um conjunto de várias lojas, tais como: lojas de roupas, *bomboniere*, cinema, lojas de alimentação e de equipamentos eletrônicos e até joalherias. No caso, um ladrão poderia se sentir mais atraído a efetuar furtos nas lojas de eletrônico ou nas joalherias, devido ao valor agregado a esses produtos, do que em uma *bomboniere*, por exemplo.

Sendo assim, é importante que os agentes de segurança efetuem o monitoramento desses estabelecimentos com uma frequência superior às das lojas que possuem menor atratividade para sofrer furtos. Contudo, não se pode descartar a possibilidade de que essas lojas menos atrativas também sofram furtos. Para resolver esse problema, é necessário que os agentes monitorem todos os estabelecimentos, sem exceção. Porém, a frequência de visitas aos diversos tipos de lojas deve ser diferenciado. Portanto, para a elaboração deste trabalho, foram pesquisados alguns modelos de patrulhamento que contemplassem essa questão.

Outra informação importante a ser tratada pelo sistema de patrulhamento é o retorno de informações para a central. É importante que o robô forneça informações completas a respeito do local monitorado. Estas informações podem ser a presença ou não de um ladrão no local ou até mesmo a informação se houve ou não alterações na cena (loja) desde a última visita do robô. Caso tenha ocorrido alguma mudança, esta pode ter sido ocasionada por um intruso. Portanto, é importante que se maximize essa aquisição de informações adquiridas pelo robô de patrulhamento.

Além disso, há trabalhos como Agmon et al. (2008) em que se afirma que um sistema de patrulhamento com múltiplos robôs trabalhando em cooperação é mais eficiente do que um sistema de patrulhamento utilizando um único robô. Em contrapartida, quanto maior for o número de lojas que se deseja monitorar e/ou maior for o número de robôs trabalhando em cooperação, maior será a complexidade do problema de otimização.

Partindo desse pressuposto, um dos objetivos deste trabalho é investigar a capacidade de uma metaheurística de encontrar uma solução de qualidade para um sistema de patrulhamento com múltiplos robôs. Esse método tem que ser capaz de encontrar uma solução de qualidade, independentemente da complexidade do problema proposto, ou seja, deve encontrar uma boa solução independentemente do número de pontos que se deseja monitorar e do número de robôs que terão de trabalhar em cooperação.

Para solucionar esse problema, foi escolhido um algoritmo de otimização bioinspirado denominado “Algoritmo Genético” (AG). Este algoritmo será responsável por efetuar todo o controle do sistema. Entre as tarefas de responsabilidade do controle do sistema estão as tarefas de autorizar o robô a avançar para próxima estação ou permanecer na mesma e a aquisição ou não de informações dos locais que é de interesse monitorar.

Em resumo, o trabalho apresentado tem como objetivos:

- Efetuar a escolha de um sistema de patrulhamento com múltiplos robôs para monitorar uma área que possua múltiplos pontos de monitoramento;
- Minimizar o tempo em que um robô demora a atender o ponto que se deseja monitorar;
- Maximizar o ganho de informações que um robô envia para a central.

1.3 REVISÃO BIBLIOGRÁFICA

A cada dia aumenta mais a preocupação com a segurança. Em paralelo, as tecnologias têm evoluído constantemente e, com isso, a aplicação de robôs com o intuito de auxiliar a segurança, principalmente nos casos em que as missões são repetitivas ou perigosas, tem ganhado destaque (PORTUGAL et al., 2013).

Nesse contexto, há diversos estudos com foco na utilização de veículos autônomos para a aplicação em tarefas como monitoramento, vigilância e mapeamento, por exemplo. Nesses tipos de sistemas, a utilização de multi-robôs trabalhando de maneira cooperativa tende a aumentar a eficiência na execução das tarefas (TEIXEIRA, 2015).

Seguindo essa linha de pesquisa, uma das principais aplicações de sistemas multi-robôs se dá em sistemas de patrulhamento (AGMON et al., 2012; AGMON et al., 2008; HWANG et al., 2009; KRISHNAMOORTHY et al., 2011; PORTUGAL et al., 2013; YU et al., 2015). Nesses sistemas, os times de robôs necessitam visitar repetitivamente algumas áreas com a finalidade de verificar alguma mudança de estado.

Sistemas de patrulhamento por meio de multi-robôs têm sido estudados utilizando diferentes abordagens a fim de melhorar a qualidade da tarefa. Alguns estudos têm como foco aumentar a frequência de visitas em cada parte da área. Em outros casos, a qualidade está atrelada a lidar com intrusos (AGMON et al., 2008). Outra questão é sobre o controle ser centralizado ou não (KARTAL et al., 2015).

Em relação ao tipo de controle a utilizar, pode-se destacar o trabalho do Agmon et al. (2012), em que foram implementados três tipos de controle com diferentes níveis de coordenação na prática. Um controle não possuía nenhuma coordenação, o outro com um nível mediano de coordenação e por fim, um controle totalmente cooperativo. Esse algoritmo foi avaliado pela média de visitação aos pontos que foram designados para serem patrulhados. Através desse método de avaliação, foi identificado que quando os robôs trabalharam de forma desordenada, houve uma maior frequência de visitação aos pontos de patrulhamento. Porém, uma das justificativas utilizada pelo autor para esse resultado controverso em relação à teoria, é que o sistema totalmente cooperativo teve vários problemas de comunicação, fazendo com que sua resposta fosse comprometida. Como o próprio autor diz, “seu modelo de sistema totalmente cooperativo na teoria é o ideal, porém não é prático na prática”. (AGMON et al. 2012)

No trabalho de Portugal et al. (2013), é proposta uma estratégia de patrulhamento cooperativo em que ocorre uma aprendizagem do sistema multi-robô a fim de resolver o problema de patrulhamento de forma distribuída. Nas tomadas de decisões, o robô utiliza decisão Bayesiana para “pensar” sobre seus movimentos, a fim de patrulhar efetivamente um ambiente e ao mesmo tempo coordenar os seus comportamentos. Neste trabalho, o sistema foi otimizado para cobrir o mapa de maneira eficiente.

Em Ghadirly et al. (2016), é proposto um modelo de patrulhamento semelhante ao problema do caixeiro viajante com janela de tempo. Nesse modelo, cada cliente possui um tempo para completar o serviço e uma janela de tempo. A restrição desse sistema é que cada cliente deve ser visitado antes de sua devida hora, caso contrário, o passeio é dito ser inviável. No caso deste trabalho, o controle foi avaliado através da minimização do tempo total de viagem.

Segundo Portugal et al. (2013) e Ghadirly et al. (2016), não há um sensor de presença com o intuito de avisar o sistema de patrulhamento que há a presença de um possível intruso. Tal dispositivo pode agir de modo cooperativo com o sistema, alertando os robôs para verificar o mais rápido possível se está realmente ocorrendo uma invasão naquele local. Este trabalho visa a comunicação entre o sistema de patrulhamento e os sensores, com a intenção de aumentar a eficiência do mesmo.

Em Hwang et al. (2009), é proposto um método de patrulhamento baseado em leilão cooperativo para resolver o problema. Nesta empreitada, a equipe de patrulhamento se organizou de tal forma que cada ponto de patrulha pode ser visitado por um membro da equipe exatamente uma vez. E o desempenho do sistema era medido através de três fatores:

da quantidade de trabalho que o sistema multi-robô tinha feito, do equilíbrio de carga de trabalho entre os robôs e do tempo de espera para os pontos de patrulha.

Esse sistema proposto por Hwang et al. (2009) tem como ponto negativo a previsibilidade. Caso um intruso perceba que um robô já passou por um determinado ponto, o intruso saberá o momento para invadir.

No trabalho de Asghar e Smith (2016), foi discutida uma estratégia de monitoramento para minimizar a latência do caminho. Há pontos específicos no sistema que possuem maior peso de recompensa, fazendo com que os robôs frequentem esses pontos com maior frequência. Porém, diferente dos anteriores, os autores desse trabalho pensaram em intrusos astutos, que seriam capazes de aprender como funciona o sistema de patrulhamento e utilizar essas informações para invadir. Como solução para esse problema, os robôs não poderiam seguir caminhos determinísticos.

O trabalho de Agmon et al. (2008) também apresenta um sistema de patrulhamento com a possibilidade de um intruso tentando invadir a área monitorada. Como consequência, o sistema de patrulha utilizado nos robôs é não-determinístico. Para tal, é utilizado um controlador com características probabilísticas. Esse esquema permite reduzir a probabilidade de entrada ilícita, mesmo sob a ameaça de um adversário sagaz, que conheça o esquema de patrulha.

O problema de patrulhamento escolhido foi baseado no trabalho Krishnamoorthy et al. (2011) devido à forma como é calculada a eficiência do sistema de patrulhamento e ao desafio de otimizá-lo, devido a sua complexidade. Além de ser um sistema de múltiplos robôs, esse sistema de patrulhamento ainda engloba o ganho de informação que o robô pode obter de um lugar onde esteja patrulhando (YU et al., 2015), além de englobar no algoritmo de otimização, através de sua função objetivo, a minimização do tempo de resposta para casos em que possa ocorrer um alerta como em Ghadiry et al. (2016).

O sistema de patrulhamento apresentado neste trabalho é baseado em um sistema de vigilância, em que a equipe de robôs ganha uma recompensa sempre que atende uma estação com um alerta ativado. Ademais, a mesma técnica utilizada neste trabalho pode ser utilizada em outros sistemas de patrulhamento, como em Hwang et al. (2009) e Lauri e Koukam (2014), ou em outras tarefas, como em uma equipe de robôs de serviços ou um sistema de entrega de mercadorias, com um nível considerável de eficiência.

1.3.1 Otimização

Pensando em otimizar um sistema no qual não seja possível conhecer todo o modelo matemático, cria-se uma restrição à utilização dos métodos clássicos de otimização. Outro fato que restringe a utilização desses métodos clássicos é o esforço computacional. Dependendo da complexidade, o problema pode se tornar inviável de ser resolvido através da utilização dos métodos clássicos de otimização. Como solução, para tratar dos problemas que possuem uma maior complexidade, são utilizados métodos baseados na inteligência computacional (REID; MALAN, 2015).

Os animais utilizam determinados comportamentos para resolver diferentes problemas na natureza. Tais métodos vão desde artifícios para conseguir alimento ou se proteger de predadores, até a reprodução da espécie. Observando o comportamento dos animais na resolução dos problemas, foram desenvolvidos métodos computacionais, também conhecidos como técnicas inteligentes, com a intenção de replicar o comportamento dos animais para resolver problemas matemáticos considerados intratáveis. Em sua maioria, esses métodos bioinspirados são baseados em características dos animais presentes na natureza, tais como a inteligência coletiva, a evolução das espécies, as redes neurais, o sistema imunológico, entre outros (CASAR; STEFEK, 2015). Essas técnicas possuem, como principal vantagem, encontrar a solução de problemas com tempo de resolução considerado inaceitável ou problemas multimodais. Alguns exemplos podem ser encontrados na literatura, tais como:

- Algoritmo Baseado em Cardumes de Peixes (CIOARGA et al., 2008);
- Algoritmo Baseado em Colônia de Abelhas (ANDRADE, 2012);
- Algoritmo Baseado em Colônia de formigas (MOTA et al., 2014);
- Algoritmo Baseado em Cardumes de Peixes (BASTOS-FILHO et al., 2014);
- Algoritmo Baseado na Ecolocalização de Morcegos (CORDEIRO et al., 2012) e (YANG, 2010);
- Algoritmo Colônia de Vaga-lumes (YANG, 2010);
- Filtro de Partículas (BASTOS-FILHO et al., 2014; FURTADO et al., 2015);
- Sistema Imunológico Artificial (VARGAS, 2005);
- Redes Neurais Artificiais (SILVA et al., 2010);
- Algoritmo Genético (BASTOS-FILHO et al., 2014; YANG, 2010).

Em relação à aplicação das técnicas inteligentes, é importante destacar que esses algoritmos têm como ponto negativo o fato de nem sempre conseguirem encontrar o ponto

ótimo global do sistema, ficando muitas vezes presos a pontos ótimos locais (WU; PANG, 2010). Tem-se que destacar que mesmo nos casos onde a solução encontrada esteja localizada em um ponto ótimo local, esta resposta possui alta probabilidade de ser considerada boa se comparada a uma solução adquirida sem a aplicação de uma técnica inteligente.

Há uma vasta variedade de técnicas de otimização que estão constantemente sendo desenvolvidas e aprimoradas. Porém, ainda não existe um algoritmo de otimização ideal que possibilite a sua utilização universal e, com isso, sempre conseguir encontrar as soluções ótimas globais (CASAR; STEFEK, 2015). Por isso que a aplicação de técnicas inteligentes para tentar solucionar novos problemas é um tema tão discutido na atualidade.

O problema de patrulhamento tratado neste trabalho possui um alto grau de complexidade, principalmente se aumentar o número de robôs ou de áreas a serem patrulhadas. Nos casos em que a complexidade do problema é elevada, o uso de técnicas inteligentes se torna uma ferramenta eficaz. Há na literatura trabalhos que utilizam algumas técnicas inteligentes para resolver problemas de patrulhamento, como em Lauri e Koukam (2014) e Marino et al. (2009).

Como exemplo, em Marino et al. (2009), é aplicada uma técnica chamada de *Fuzzy*. Nesse trabalho, o algoritmo tem como função controlar um sistema de patrulhamento de fronteiras multi-robô descentralizado. O algoritmo é projetado para recorrer a um quadro de controle comportamental e é organizado em uma estrutura hierárquica. Cada ação de controle é obtida pela combinação adequada entre vários comportamentos elementares, através do quadro de controle comportamental. Um sistema de inferência *Fuzzy* é projetado para selecionar a ação adequada de acordo com informações do sensor.

Em Lauri e Koukam (2014), foi elaborado um problema de patrulhamento em que os múltiplos robôs podiam se mover em velocidades diferentes e os nós (ponto de interesse) podiam receber prioridades distintas. Neste trabalho, foram estudadas duas classes de patrulhamento. Como solução para uma dessas classes, foi apresentado um algoritmo bioinspirado híbrido baseado na combinação entre o Algoritmo Evolutivo e o Algoritmo *Ant-Colony*. Como resultado, o algoritmo bioinspirado híbrido teve uma resposta melhor que o controle baseado em heurística. Outra vantagem apresentada pelo algoritmo híbrido foi em relação a não apresentar problemas independentemente das restrições ou regras de velocidade e prioridades impostas, algo que gerou problemas no controle heurístico.

Para o caso do problema de otimização do sistema de patrulhamento utilizando múltiplos robôs proposto neste trabalho, foi pensado no Algoritmo Genético devido às suas características binária e combinatória. O Algoritmo Genético é um algoritmo bastante

discernido no meio acadêmico por ser um algoritmo poderoso baseado no mecanismo de busca estocástica semelhante à evolução natural. Por esse motivo, o Algoritmo Genético tem sido utilizado com sucesso em diversas aplicações, incluindo problemas em que os métodos numéricos não conseguem encontrar uma solução ou simplesmente não podem ser aplicados. Outra vantagem do AG é por ser uma ferramenta de busca automática de soluções para determinada função objetivo (BARRERA; FLORES, 2007).

Pode ser considerada uma desvantagem do Algoritmo Genético a característica de ser um método não determinístico, ou seja, pode encontrar soluções ótimas diferentes, em simulações diferentes, para a mesma função objetivo. Porém, tal fato mostra que o algoritmo genético tem uma forte capacidade de adaptação (SHI; CUI, 2010).

Um cuidado necessário ao utilizar esse método é em relação à sua convergência. Em algumas situações, a convergência para uma solução ótima é excessivamente lenta. Essa limitação pode impedir que esse algoritmo seja utilizado para uma serie de aplicações (RENDERS; FLASSE, 1996).

1.3.2 Algoritmo Genético

O Algoritmo Genético (AG) é uma técnica inteligente muito disseminada no meio acadêmico. O AG surgiu na década de 60 quando John H. Holland propôs um método de otimização baseado em uma analogia com o processo de seleção natural, o darwinismo, e a genética evolutiva (RENDERS; FLASSE, 1996).

Quando se utiliza o Algoritmo Genético, tem-se o objetivo de maximizar ou minimizar uma função objetivo. Para conseguir cumprir com esse objetivo, o AG mantém uma população de indivíduos ao longo de todo o espaço de busca. A cada estágio desse algoritmo, é dito que se criou uma nova geração de indivíduos e, como consequência, uma nova população é criada.

Cada indivíduo da população possui um código genético e como consequência um valor de aptidão, que é adquirido através da FOB. Na maioria das vezes, essa função objetivo é formada por uma que retorna um único valor (BARRERA; FLORES, 2007).

De acordo com o processo de seleção natural, os indivíduos mais aptos têm maior probabilidade de gerar descendentes e assim levar sua prole (genética) para a futura geração. Com base no processo de seleção natural, para o AG gerar a nova população, primeiro os indivíduos são avaliados através da função objetivo. Após essa avaliação são selecionados os indivíduos para o processo de reprodução e mutação.

Depois dos processos de reprodução e mutação, também chamados de operadores genéticos, é obtida a nova população. Uma característica do AG é que a população sempre é mantida com o mesmo número de indivíduos de uma geração para a outra, independentemente da técnica que se utiliza para selecionar esses indivíduos.

A nova população é formada por indivíduos que se ajustaram melhor à função objetivo. Como há sucessivas iterações ao longo do tempo, a população tende a se tornar ideal de acordo com a função objetivo.

O AG modela computacionalmente os processos de evolução natural, tais como: nascimento, seleção, reprodução, cruzamento e mutação de indivíduos, com o intuito de construir uma ferramenta para resolução de diversos problemas em várias áreas de conhecimento (SHI; CUI, 2010).

Abaixo é apresentado um pseudocódigo de um AG genérico.

Algoritmo 1: Algoritmo Genético

1. Declara_variáveis
 2. Inicializa_população
 3. Avaliação da Função_aptidão
 4. Enquanto o critério de convergência estabelecido não for atendido, faça
 5. Seleciona_pais
 6. Crossover
 7. Mutação
 8. Nova_população
 9. Avaliação da Função_aptidão
 10. Fim enquanto
 11. Retorna_melhor_solução
-

O funcionamento de cada função dentro do AG será explicado de forma mais detalhada na seção 3.2 no capítulo 3.

1.4 ORGANIZAÇÃO DO TRABALHO

A dissertação está organizada da seguinte maneira:

- Capítulo 2: apresenta a formulação do problema, bem como o modelo matemático do sistema de patrulhamento. Neste capítulo, é explicado todo o sistema de patrulhamento que será otimizado, desde como é feito o movimento do robô até o cálculo da recompensa por completar a missão;
- Capítulo 3: apresenta a solução para o problema proposto neste trabalho, tal como um outro modelo genérico para comparação com o método proposto;
- Capítulo 4: são apresentados os resultados obtidos através de simulações. O resultado do sistema de patrulhamento utilizando AG é comparado com outro controle genérico com o intuito de verificar a diferença de ambos no comportamento dos robôs no sistema assim como a eficiência. Neste capítulo, serão apresentados quatro casos com complexidades diferentes;
- Capítulo 5: finalizando o trabalho, são apresentadas as conclusões obtidas através da análise dos resultados das simulações e sugestões para trabalhos futuros.

2 FORMULAÇÃO DO PROBLEMA

O modelo matemático utilizado neste trabalho é o mesmo apresentado por Krishnamoorthy et al. (2011). Como citado anteriormente, esse modelo foi escolhido por ser um sistema de patrulhamento utilizando múltiplos robôs e ainda possuir uma função objetivo que engloba duas informações importantes para um sistema de patrulhamento, tais como as informações captadas pelos robôs e o tempo gasto para atender a um sinal de alerta.

O sistema de patrulhamento utilizando múltiplos robôs proposto por Krishnamoorthy et al. (2011) é composto por um número N de estações. Estas são divididas em estações de passagem e estações de alerta. As estações de passagem são utilizadas apenas como caminho obrigatório que o robô precisa seguir, ou seja, não recebem alerta. Por outro lado, as estações de alerta são os pontos de interesse onde o robô tem de monitorar. O número de estações de alerta é representado pela variável m . A variável $x_j(k)$ representa a posição do robô j no perímetro de estações no estágio k . Já $d_j(k)$ é o tempo de permanência do robô em qualquer uma das estações no estágio k .

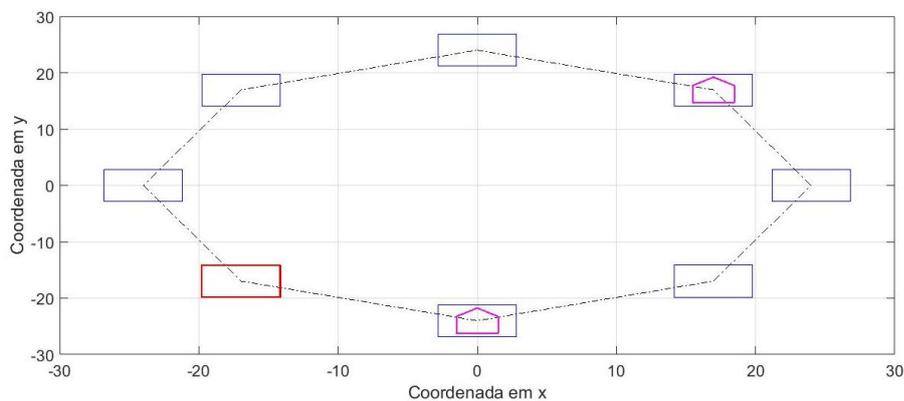


Figura 1 – Modelo do Sistema de Patrulhamento.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 1 mostra o modelo do sistema de patrulhamento composto por oito estações ($N = 8$), representadas pelas cores azul e vermelha, e dois robôs ($j = 2$), representados pela cor rosa. Sempre que uma estação estiver com um alerta ativado, ela ficará na cor vermelha para se diferenciar das outras estações.

As estações de alerta ($i = 1, 2, \dots, m$) possuem uma regra probabilística para entrar em alerta. A variável $Y_i(k)$ é binária e representa a chegada de alerta na estação i . Esta chegada nas estações ocorre de forma independente uma da outra, com uma taxa de alerta α por

unidade de tempo. A probabilidade de uma estação receber um alerta é definida através da equação (2.1).

$$p_j = p^{(m-n_j)} * (1 - p)^{n_j} \quad (2.1)$$

Onde:

$$p = e^{-\alpha} \quad (2.2)$$

$$n_j = \sum_{i=1}^m Y_i(k) \quad (2.3)$$

A equação (2.2) representa uma parte da probabilidade de uma estação entrar em alerta ou não. A variável α que representa a taxa para que cada estação entre em alerta pode ser diferente para cada estação. Quanto maior for o valor do α , menor será o p . No caso da equação (2.3), é somado o número de estações que recebeu alerta no estágio k . Essa variável n_j influencia para que não chegue alerta durante todos os estágios.

Por fim, a equação (2.1) refere-se à probabilidade da estação i receber alerta. Quanto menor for o p , menor será a probabilidade (p_j) de uma estação entrar em alerta. Assim como em relação a n_j , onde quando maior for n_j , menor será a probabilidade (p_j) de uma estação entrar em alerta.

Após o cálculo de p_j , efetuado pela equação (2.1), será sorteado um número entre 0 e 1 para cada estação i que será comparado com o valor de p_j . Quando o valor de p_j for maior que esse número sorteado, a estação i receberá um alerta no próximo estágio ($Y_i(k + 1) = 1$). Caso o número sorteado seja menor que p_j , a estação i não receberá um alerta no próximo estágio ($Y_i(k + 1) = 0$).

As equações (2.4) e (2.5) mostram, respectivamente, como é calculado a posição futura ($x_j(k + 1)$) e o tempo no futuro que cada robô permanecerá na estação ($d_j(k + 1)$). Essas variáveis dependem da decisão de controle do robô. Esta decisão é indicada por uma variável binária $u_j(k)$.

$$x_j(k + 1) = (x_j(k) + 1 - u_j(k)) \text{ mod } N \quad (2.4)$$

$$d_j(k + 1) = (d_j(k) + 1) * u_j(k) \quad (2.5)$$

Analisando as equações (2.4) e (2.5), pode-se notar que se o valor da variável de controle for igual a zero ($u_j(k) = 0$), no estágio futuro o robô irá para a próxima estação e o tempo de permanência do mesmo será zero ($d_j(k + 1) = 0$). No caso em que a variável de controle for igual a "1" ($u_j(k) = 1$), o robô irá permanecer na mesma estação e seu tempo de permanência será incrementado ($d_j(k + 1) = d_j(k) + 1$).

Na equação (2.4), há na expressão o termo *mod*. Este termo é uma função do MATLAB®, que retorna o resto da divisão. Como exemplo, se $N = 8$ e o termo $x_j(k) + 1 - u_j(k)$ for igual a "6", o resultado será conforme a equação (2.6):

$$\begin{aligned} x_j(k + 1) &= (6) \text{ mod } 8 \rightarrow x_j(k + 1) = \text{resto} \left(\frac{6}{8} \right) \\ x_j(k + 1) &= 6 \end{aligned} \quad (2.6)$$

Como o resto da equação (2.6) foi 6, logo a posição do robô j no estágio futuro será 6, ($x_j(k + 1) = 6$). No exemplo onde $N = 8$ e o termo $x_j(k) + 1 - u_j(k)$ for igual a 8, a posição do robô será conforme a equação (2.7):

$$\begin{aligned} x_j(k + 1) &= (8) \text{ mod } 8 \rightarrow x_j(k + 1) = \text{resto} \left(\frac{8}{8} \right) \\ x_j(k + 1) &= 0 \end{aligned} \quad (2.7)$$

O status de alerta em cada estação é armazenado em outro vetor binário $A_i(k)$. A equação (2.8) mostra como é atualizado esses alertas para cada estação.

$$A_i(k + 1) = \prod_{j=1}^j \left(1 - \delta(x_j(k), X_i), u_j(k) \right) * \max\{A_i(k), Y_i(k)\} \quad (2.8)$$

Na equação acima, X_i representa a posição da estação i . Percebe-se na equação o impacto do *delta de Kronecker*, que é definido na equação (2.9).

$$\delta(i, j) = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (2.9)$$

Analisando a equação (2.8) e considerando o *delta de Kronecker* (2.9), observa-se que caso o robô esteja em uma estação com alerta ativado ($A_i(k) = 1$) e a variável de controle ativada ($u_j(k) = 1$), a estação terá seu alerta desativado ($A_i(k + 1) = 0$). Isso significa que o robô está atendendo ao alerta naquela estação e, portanto, não é mais necessário que o mesmo fique ativado para atrair outro robô. Outra situação que deve ser analisada é o caso em que ocorra a chegada de alerta ($Y_i(k) = 1$) e não possuir nenhum robô na estação com a variável de controle ativada. Para este caso, a estação de alerta assumirá o status de alerta ativado ($A_i(k + 1) = 1$).

Uma restrição do problema modelado está relacionada ao tempo de permanência do robô em uma estação (d_j), como mostrado na equação (2.10). Nesta equação, o parâmetro D se refere ao número máximo que um robô pode permanecer em uma estação. Este valor (d_j) influencia diretamente no ganho de recompensa do sistema, como será mostrado a frente.

$$d_j(k) \leq D \quad (2.10)$$

Já a equação (2.11) mostra a condição responsável por desativar a variável de controle (u_j).

$$d_j(k) = D \rightarrow u_j(k) = 0 \quad (2.11)$$

Um dos intuitos desse sistema de patrulhamento é maximizar o ganho de informação que os robôs conseguem transmitir para um provável operador através de filmagem, por exemplo. Os autores Krishnamoorthy et al. (2011), no qual este trabalho se baseia, disponibilizaram uma relação entre o ganho de informação que cada robô consegue obter em relação ao número de estágio em que o mesmo permanece em uma estação, onde através da utilização desses dados foi gerada uma curva como mostrado na Figura 2.

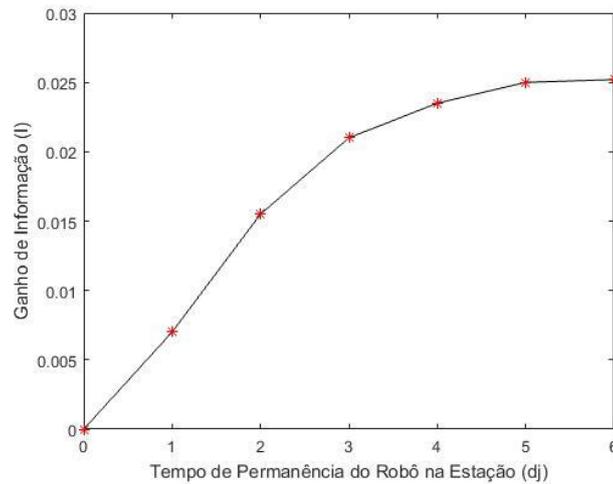


Figura 2 – Relação entre o Ganho de Informação em Relação ao Tempo de Permanência do Robô em uma Estação.

Fonte: KRISHNAMOORTHY et al., 2011.

Utilizando os dados da Figura 2, é possível calcular o ganho de informação para cada estágio. Este ganho varia de acordo com o tempo de permanência do robô em cada estação e é agregado na equação (2.12), referente ao cálculo da recompensa, pela variável $I(d_j)$.

Há duas formas de calcular a recompensa baseada na análise da posição dos robôs, como mostrado na equação (2.12). A equação de cima representa o caso em que todos os robôs estão em posições diferentes ($x_{j1} \neq x_{j2}$), ou seja, todos os robôs podem gerar recompensa para o sistema. A equação de baixo calcula a recompensa quando há dois robôs ou mais parados na mesma estação ($x_{j1} = x_{j2}$). Para o segundo caso, somente o robô que chegou primeiro na estação, ou seja, que possuir maior valor de d_j , irá gerar recompensa para o sistema.

$$R^u(k) = \begin{cases} \sum_{j=1}^j [I(d_j + 1) - I(d_j)] * u_j(k) - \beta * \sum_{i=1}^m A_i, x_{j1} \neq x_{j2} \\ [I(d_{\bar{j}} + 1) - I(d_{\bar{j}})] * u_{\bar{j}}(k) - \beta * \sum_{i=1}^m A_i, x_{j1} = x_{j2}, \bar{j} = \arg \max d_j \end{cases} \quad (2.12)$$

Para simplificar o entendimento da equação (2.12), pode-se dividi-la em dois termos. O primeiro é utilizado para atingir o objetivo de maximizar o ganho de informação do sistema, como mostrado na equação (2.13). Nesse termo, quanto maior for a diferença de ganho de informação, mostrado na Figura 2, maior será o ganho de recompensa do sistema.

$$\sum_{j=1}^j [I(d_j + 1) - I(d_j)] * u_j(k) \quad (2.13)$$

O outro termo da equação (2.12), apresentado na equação (2.14), é referente ao outro objetivo do trabalho, que é minimizar o tempo em que o sistema de robôs demorará a atender a um alerta ativo. Quando tiver algum alerta ativado no sistema, esse termo receberá um valor diferente de “0” (zero) e como consequência haverá decréscimo na recompensa do sistema. Na situação em os alertas estão todos desativados, esse termo será zerado fazendo com que não ocorra decréscimo na recompensa. Ainda é possível notar o parâmetro β , que é uma constante de multiplicação, com o intuito de equilibrar o valor do ganho de recompensa e o valor da punição pelo tempo que o robô demora a atender aos alertas ativados. Caso esses termos da equação não sejam equilibrados, pode ocorrer duas situações, ou o ganho da recompensa será muito superior em relação a essa punição e assim o sistema de otimização deixaria de minimizar o tempo em que o robô demora a atender ao alerta, ou o contrário. Para este trabalho, foi definido através de testes o valor do parâmetro $\beta = 0,001$.

$$-\beta * \sum_{i=1}^m A_i \quad (2.14)$$

O algoritmo do modelo matemático do sistema de patrulhamento é descrito no pseudocódigo a seguir:

Algoritmo 1: Modelo Matemático do Sistema de Patrulhamento com Múltiplos Robôs

1. Inicializa_variáveis
 2. Calcula_chegada_de_alerta Y_i
 3. Para $k=1$ até $k=\text{iterações_do_robô}$, faça
 4. Calcula_posição_futura_robôs $x_j(k+1)$
 5. Calcula_tempo_futuro_permanência_robô $d_j(k+1)$
 6. Define_estações_com_alerta_futuro $A_i(k+1)$
 7. Calcula_recompensa $R_u(k)$
 8. Fim
 9. Retorna_Recompensa $R_u(\text{iterações_do_robô})$
-

O funcionamento do sistema de patrulhamento também pode ser visto no fluxograma, conforme a Figura 3.

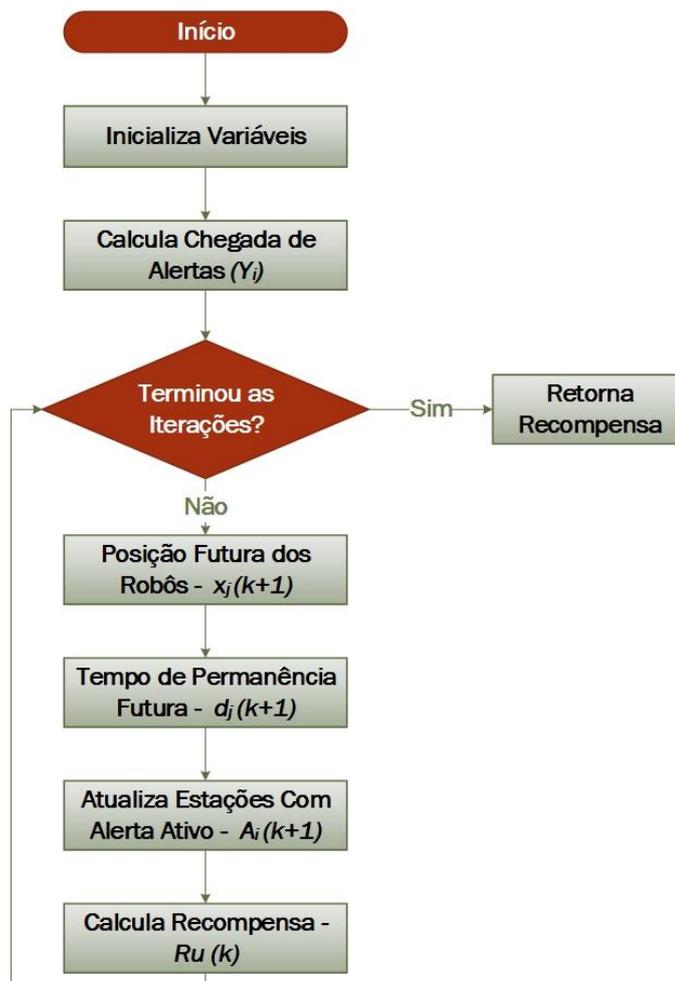


Figura 3 – Fluxograma do Sistema de Patrulhamento.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

3 SOLUÇÃO DO PROBLEMA

Neste trabalho foram implementados dois tipos de controle para otimizar o sistema de patrulhamento proposto.

O primeiro controle foi baseado na implementação de uma heurística, na qual o problema possui regras de decisão pré-estabelecidas. O segundo, no Algoritmo Genético (AG).

3.1 CONTROLE UTILIZANDO HEURÍSTICA PRÉ-DEFINIDA

Para a implementação dessa estratégia de controle, foi adotado que o valor de D é limitado a um valor fixo e foram adotados dois modelos.

No controle 1, a única informação levada em consideração é o valor do D . Nesse caso, sempre que o valor de d_j se igualar ao valor de D , a variável de controle retornará a zero ($u_j = 0$). Porém, para esse caso, acontece a situação em que dois ou mais robôs andem sempre juntos, prejudicando o acúmulo de recompensa.

Com o intuito de resolver os problemas encontrados, foi criada outra versão de controle heurístico. Nesse controle 2, foi incrementado duas heurísticas em relação ao controle 1, com a intenção de aumentar a recompensa. Uma das heurísticas acrescentadas é em relação a vários robôs estarem caminhando sempre juntos. Como solução adotada, foi implementado a heurística em que sempre que um robô estiver parado em uma estação, o outro seguirá para a próxima em busca de uma maior recompensa. A outra heurística acrescentada foi sobre ter ou não alguma estação em alerta. Para o caso em que não haja outra estação com alerta ativado, o robô poderá permanecer na estação atendendo ao alerta durante um período de estágios maior que D , já para o caso de surgir um alerta o robô permanecerá na estação durante um período de no mínimo D estágios, a fim de obter o máximo de recompensa possível.

Através da análise da Figura 2, pode-se notar algumas possibilidades de valores de D . Foram testados os dois casos citados acima para os valores de D em que há um ganho de recompensa, ou seja, para os valores entre 1 e 6.

Após a comparação dos resultados desses vários testes realizados, foi utilizado o que retornou a maior recompensa para efetuar a comparação com o AG. Para as situações em que mais de um controle obteve o mesmo valor que a maior recompensa acumulada, é utilizado

apenas um desses controles para comparação com o AG. Visto que o responsável por medir a eficiência do controle é o ganho de recompensa e que esses controles obtiveram a mesmo valor, logo, conclui-se que a eficiência de ambos se equiparam, não havendo necessidade de comparação.

3.2 CONTROLE VIA ALGORITMO GENÉTICO (AG)

O controle baseado no AG foi implementado de modo a otimizar a eficiência do sistema de patrulhamento com múltiplos robôs. Essa eficiência é medida através da recompensa. Por conta disto, o AG trata o problema como um todo, ou seja, não se preocupa com a recompensa em cada estágio, e sim com a recompensa final.

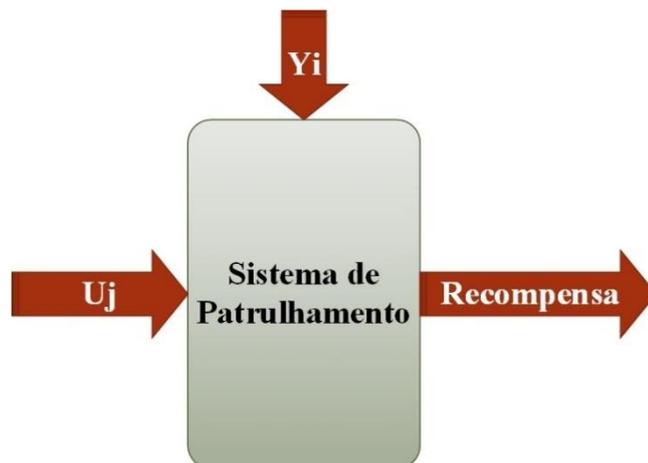


Figura 4 – O Sistema de Patrulhamento é Tratado Como Uma Caixa Preta Pelo AG.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 4 apresenta como o sistema de patrulhamento é visto pelo AG. Tal fato diminui a complexidade da resolução do problema e faz com que o único objetivo do sistema de controle seja as variáveis do mesmo, a interferência externa e a função objetivo do problema.

No caso do problema de patrulhamento, a variável de controle é o $u_j(k)$, sendo a mesma binária. Como dito no Capítulo 2, essa variável é a responsável por controlar o movimento dos robôs e, como consequência, pelo acúmulo ou não de recompensa.

A interferência externa desse problema é na chegada ou não de alertas (Y_i) a uma determinada estação. Essa variável que descreve a chegada de alertas é gerada de maneira probabilística, como mostrado no Capítulo 2, e interfere em todo sistema de patrulhamento,

inclusive penalizando a recompensa. É de extrema importância que o AG possa conseguir minimizar a penalização gerada pela variável externa. Para isso, é necessário que sempre que ocorrer a chegada de um alerta, haja um robô próximo a essa estação para atender ao alerta o mais breve possível.

A função objetivo (FOB) é a recompensa. Como dito, esse dado é responsável por quantificar o quão eficiente o sistema de patrulhamento está sendo perante as entradas do sistema, as variáveis de controle e as variáveis externas.

Como o intuito é maximizar a recompensa final, é importante que os robôs atendam aos alertas ativos rapidamente, para que não sejam penalizados ou que sejam penalizados o mínimo possível. Para isso, os robôs devem estar bem posicionados no mapa. Tal fato mostra que a estratégia de controle pode ser complexa, fazendo com que possa haver casos em que a recompensa do controle via heurística possa ser maior que a recompensa do AG nos primeiros estágios.

Outro fato que influencia na maximização da recompensa é o tempo em que o robô permanece adquirindo informações em uma estação com alerta ativado, como pode ser visto na Figura 2. Esse tempo é de extrema importância, pois se o robô ficar pouco tempo na estação com alerta ativado, o mesmo terá adquirido poucas informações e como consequência terá uma recompensa pequena. Outra possibilidade é caso o robô permanecer muito tempo na estação com o alerta ativado, irá adquirir uma recompensa mais alta. Porém, a partir de certo número de estágios em que esse robô permanece nessa estação, as informações começarão a ser redundantes, fazendo com que o ganho de recompensa por estágio seja pequeno ou até mesmo nulo.

Outro fato que acontece somente no caso do controle utilizando AG é o fato de o robô poder parar em uma estação com alerta desativado. Tal possibilidade não pode gerar recompensa e nem punição. Isso ocorre apenas para o AG tentar posicionar melhor os robôs no mapa.

Outra característica do AG utilizado é em relação a como é tratada a sua FOB. Como o sistema proposto por este trabalho é um problema de maximização de recompensa e o algoritmo utilizado foi o *Toolbox* de AG do próprio *software* MATLAB®, que por sua vez é um algoritmo de minimização, houve a necessidade de adaptar a recompensa do sistema de patrulhamento. Sendo assim, nas análises dos resultados adquiridos através do AG, aparecerá a recompensa negativa.

Na Figura 5 a seguir, é apresentado o fluxograma referente ao controle via Algoritmo Genético (AG) implementado para otimizar o sistema de patrulhamento utilizando múltiplos

robôs. Na frente de cada função há o número do tópico em que estará a explicação sobre a função.

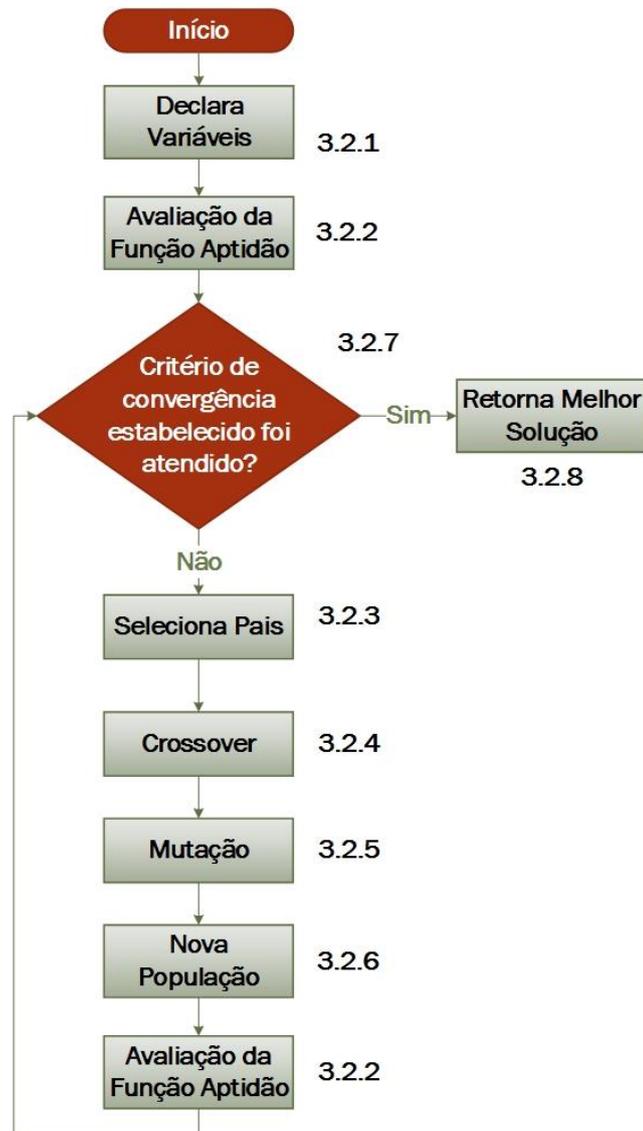


Figura 5 – Fluxograma do AG Utilizado Para Otimizar o Sistema de Patrulhamento Utilizando Múltiplos Robôs.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

3.2.1 Declaração das Variáveis Para Inicializar o AG

Inicialmente, são declaradas as variáveis para inicializar o algoritmo. Estas variáveis são: número de indivíduos, número de cromossomos (robôs), número de genes (estágios), taxa de mutação e dados para o critério de parada.

As escolhas dos valores dessas variáveis são de extrema importância. Esses dados influenciarão diretamente na eficiência do algoritmo, podendo inclusive ocasionar a não convergência do algoritmo para uma solução satisfatória.

Nessa etapa do algoritmo também é inicializada a população inicial. Para solucionar esse problema de patrulhamento, definiu-se uma população de 2000 indivíduos para o AG. O número de indivíduos e de gerações foram definidos a partir de testes. Como esse algoritmo possui dados de entrada aleatórios (Y_i) e outros que podem ser alterados pelo usuário, como número de robôs (j) e número de estações (N), é difícil estabelecer quais valores são os ideais para esses dados. Outra questão que foi levada em consideração é o tempo para o algoritmo encontrar a resposta.

Para o sistema de patrulhamento, a população inicial foi gerada de maneira aleatória para que o AG encontrasse a solução ótima independente de se ter ou não algum dado anterior a sua utilização.

As características dos indivíduos alteram-se de acordo com o problema, pois o número de cromossomos depende do número de robôs (j) utilizados no sistema de patrulha. Já o número de genes depende do número de estágios do sistema de patrulhamento (k).

A Figura 6 apresenta o exemplo de um indivíduo gerado por esse algoritmo. Para o caso foi utilizado 2 robôs ($j = 2$) e 4 estágios para o sistema de patrulhamento ($k = 4$). O indivíduo em verde (solução) é composto por dois cromossomos em azul, sendo que cada cromossomo corresponde a um robô do sistema. Por sua vez, o robô 1 é formado por quatro genes em amarelo, e o cromossomo 2 é formado por outros quatro genes em vermelho, ou seja, cada cromossomo é formado pelo número de genes correspondente ao número de estágios. Portanto, esse indivíduo é composto por dois cromossomos e oito genes.

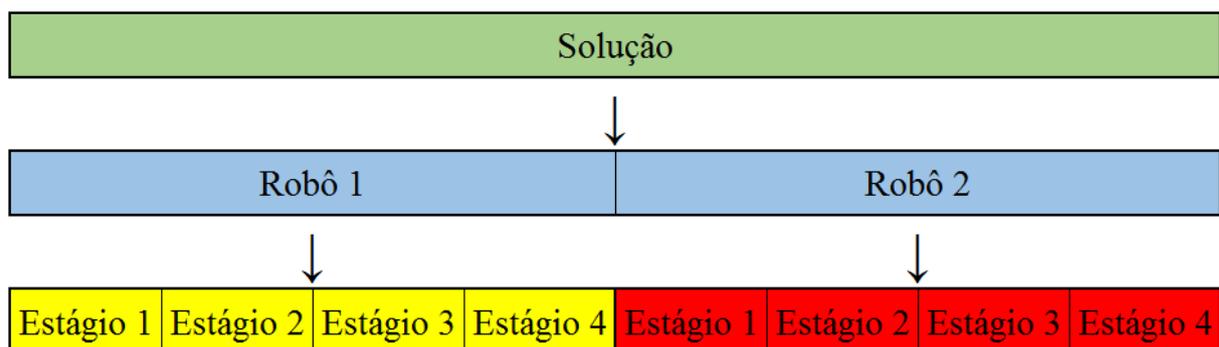


Figura 6 – Composição de Um Indivíduo para a População.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

3.2.2 Avaliação da Função Aptidão

A função aptidão é a responsável por avaliar a qualidade do indivíduo. Essa função é a única parte do AG ligado ao problema de patrulhamento em si. A avaliação da qualidade da resposta de cada indivíduo é feita através da função objetivo (FOB).

Portanto, a função objetivo deve levar em consideração as restrições e características do problema. Caso contrário, poderá causar uma avaliação errada e, como consequência, ocasionar o descarte de soluções que são boas, assim como reproduzir soluções que são ruins para o problema.

Para o sistema de patrulhamento utilizando múltiplos robôs foi utilizado como função aptidão a recompensa do sistema ($Ru(k)$).

3.2.3 Seleção dos Indivíduos

O processo de reprodução depende da avaliação dos indivíduos. Nesta etapa do algoritmo são selecionados os pais para efetuar a reprodução (geração de novos descendentes) seguindo o critério de seleção natural, que determina que pais melhores possuem a probabilidade de gerar mais descendentes.

Uma observação importante em relação à seleção dos indivíduos para a reprodução, é que os indivíduos que não possuem uma boa avaliação também podem ser selecionados para a etapa da reprodução.

Há algumas técnicas que podem ser utilizadas para a seleção dos pais, tais como roleta e torneio. No caso deste trabalho foi utilizada a roleta para a seleção dos indivíduos.

A técnica da roleta funciona da seguinte forma: é criada uma roleta na qual cada indivíduo recebe um pedaço da roleta proporcional à sua avaliação feita pela função aptidão. A Figura 7 abaixo apresenta um exemplo de uma roleta em que cada cor corresponde a um indivíduo. A proporção de espaço que cada indivíduo ocupa na roleta depende da qualidade da função aptidão. No caso deste trabalho, quanto maior a recompensa (Ru), maior a avaliação da função aptidão. Portanto, os indivíduos que possuísem uma recompensa maior, receberiam um espaço maior na roleta. Como no exemplo da figura, a melhor solução ficou com o espaço em roxo. Já a pior solução ficaria com um espaço menor, que foi representado pela cor azul.

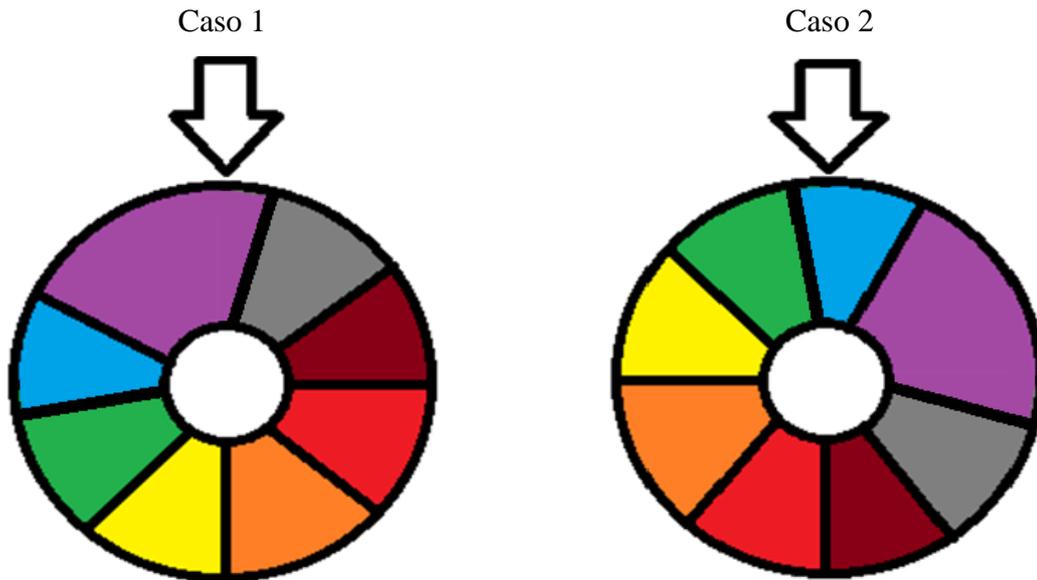


Figura 7 – Exemplo da Técnica da Roleta.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A soma de todos os pedaços deve ser de 100%. Depois de pronta, a roleta é girada. Como exemplo, foi apresentado na Figura 7 dois casos de giros da roleta, sendo que no caso 1 o indivíduo sorteado corresponde a cor roxa e, no caso 2 o indivíduo sorteado corresponde a cor azul. Como consequência, esses dois indivíduos serão utilizados como pais para gerar descendentes na próxima etapa.

Esse processo de seleção foi escolhido por permitir que tanto indivíduos com uma boa solução, como aqueles que possuem uma solução ruim, tenham a chance de serem selecionados para gerar descendentes, tendo em vista que os detentores de uma melhor solução têm maior probabilidade de serem escolhidos como pai.

3.2.4 Crossover

Após a seleção dos pais, é feito o cruzamento entre os mesmos. Esse procedimento é chamado de *crossover*. Esta função é responsável por gerar a nova população a partir dos indivíduos selecionados para serem pais.

Há algumas variedades de cruzamento entre os indivíduos, como por exemplo: o cruzamento de um ponto, o *crossover* de dois pontos, o *crossover* uniforme e o *crossover* com mais de dois pais. Para este trabalho, foi adotado o processo de cruzamento feito Ponto a Ponto. Esse processo se dá através do cruzamento realizado gene a gene.

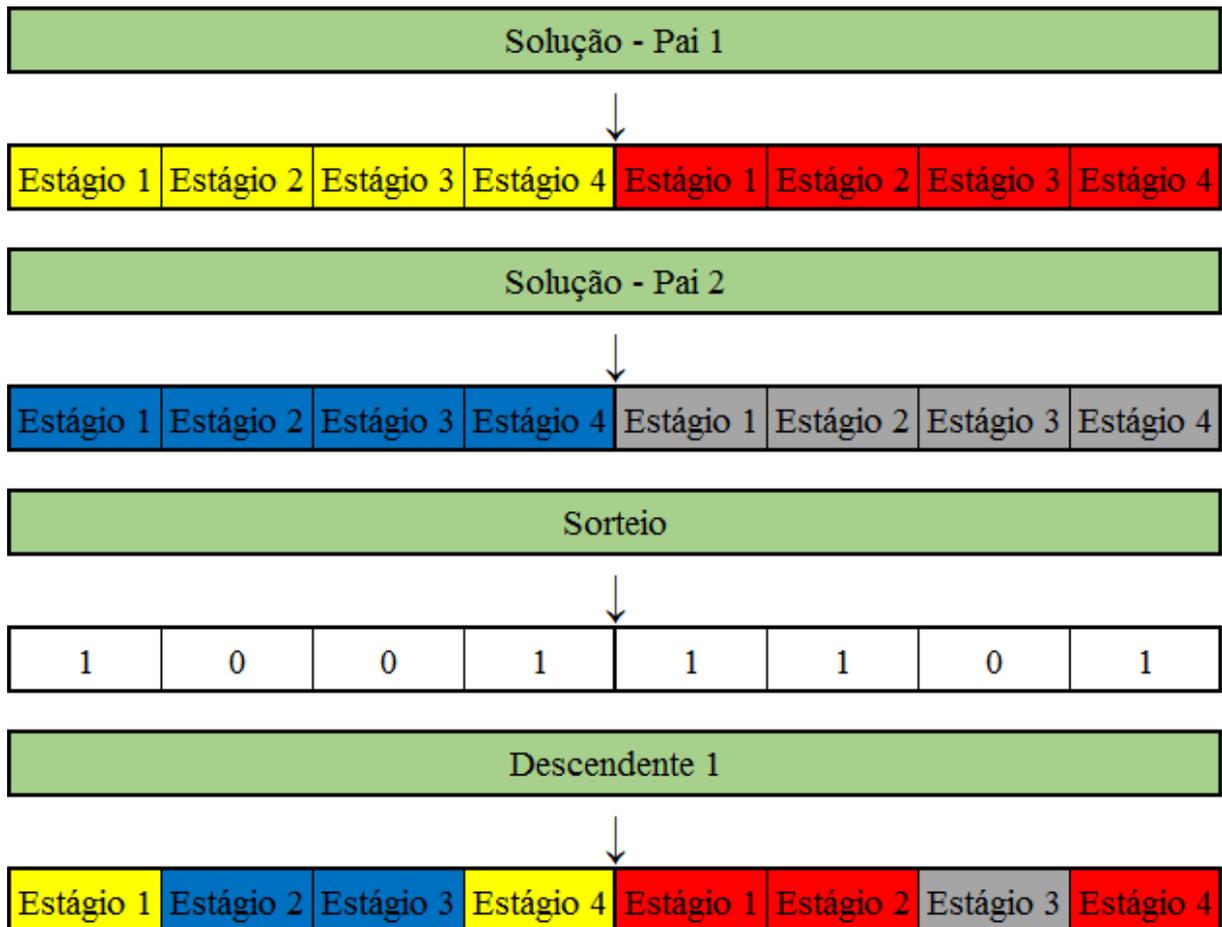


Figura 8 – Processo de *Crossover*.
 Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 8 mostra como é feito o processo de cruzamento Ponto a Ponto. Após selecionar os pais para a etapa de *crossover*, é sorteado um vetor com as mesmas características dos indivíduos quanto ao número de genes dos indivíduos (vetor sorteio). O novo indivíduo (descendente) é gerado dependendo do dado que está na posição relativa àquele gene. Por exemplo, no caso do estágio 1 do primeiro robô, no vetor sorteio há o número 1, logo o descendente terá o gene correspondente do pai 1. Já para o caso do estágio 2 do robô 1, o vetor sorteio possui naquela mesma posição um valor de 0, logo o gene do descendente naquela posição será do pai 2.

3.2.5 Mutação

O operador genético de mutação dos indivíduos é aplicado para aumentar a diversificação entre os descendentes. A mutação é importante para que o algoritmo não fique preso a ótimos locais. Porém, tem que haver cuidado na escolha desta taxa, pois se for alta, o

algoritmo se tornará aleatório. A mutação foi feita de maneira uniforme utilizando uma taxa de 1%.

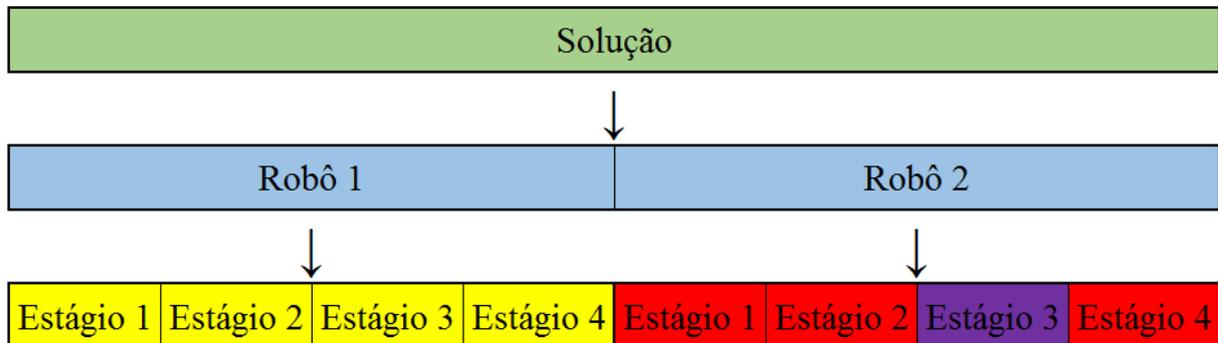


Figura 9 – Exemplo de Mutação em Um Indivíduo da População.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Para o algoritmo utilizado era sorteado para todos os genes do indivíduo um número entre 0 e 1 de forma aleatória. Este número sorteado era comparado com a taxa de mutação e, quando possuía um valor menor que a taxa, esse gene receberia o valor binário contrário ao que possuía anteriormente. Como mostrado na Figura 9, o gene relativo ao estágio 3 do robô 2 teve um valor sorteado menor que a taxa de mutação, por isso foi substituído por outro gene. Nesse caso, é dito que esse gene sofreu mutação.

3.2.6 Seleção da Nova População

A nova população é formada após a mutação. Esta nova população sempre deve possuir o mesmo número de indivíduos da população anterior. A nova população será formada apenas pelos descendentes da geração anterior ou pode ser utilizado algum método de formação. Dois exemplos desses métodos são o elitismo e o *steady state*.

Para a formação da nova população foi utilizado o elitismo. Portanto, após ter todos os descendentes formados, a pior solução é substituída pela melhor solução da geração anterior. Com a utilização do elitismo há a garantia de que a melhor solução do problema nunca é perdida ou descartada.

3.2.7 Critério de Parada

O critério de parada deve ser escolhido com cuidado, pois pode gerar uma demora excessiva para interromper o algoritmo, e assim causar um esforço computacional desnecessário, ou pode interromper a busca antes que uma boa solução seja encontrada.

Os critérios de parada estão em constante discussão, pois são considerados um dos principais problemas dos métodos heurísticos de otimização existentes atualmente. Esse fato se dá devido à impossibilidade de saber qual é a melhor solução para um determinado problema, logo não há como saber se aquela solução ótima encontrada é global ou não (WU; PANG, 2010).

Por conseguinte, o que deve prevalecer na escolha do critério de parada é o bom senso. É necessário ter em mente três informações relativas à solução para tomar esse tipo de decisão. São elas:

- Precisão;
- Confiabilidade;
- Esforço computacional.

A partir dessas informações, é possível ter uma maior clareza para tomar essa decisão. Dependendo da complexidade do problema, pode ser impossível chegar a uma solução ótima global com um pequeno esforço computacional e alto grau de precisão e confiabilidade.

Para o controle proposto, foram utilizados dois critérios de parada. Um dos critérios é baseado no número máximo de gerações. É importante ressaltar que o critério de parada por número de gerações não garante que a resposta é de qualidade e nem que o esforço computacional seja reduzido. A vantagem em relação à utilização desse critério é devido à simplicidade de ser implementado. Neste trabalho foi definido que o número máximo de gerações seria de duas mil.

O outro critério adotado é em relação à estagnação das soluções encontradas. Caso não houvesse mudanças significativas nas soluções durante algumas gerações, o algoritmo era interrompido e retornava a melhor solução. A tolerância adotada em relação a essas mudanças foi de 1μ (um micro).

No caso do AG proposto neste trabalho, como foram adotados dois critérios de parada, a seleção de qual critério irá interromper a busca por novas soluções será analisada a cada iteração, sendo que o critério cuja condição for atingida primeiro, será o responsável por parar o algoritmo.

4 RESULTADOS

Para comprovar a eficiência do AG utilizado, foram efetuados quatro testes com diferentes números de robôs (j), de estações (N) e de estações de alerta (m), aumentando gradativamente a complexidade do problema. Esses dados foram escolhidos de forma empírica, sendo que a única relação que deve ser respeitada é que o número de estações de alerta obrigatoriamente tem que ser igual ou menor ao número de estações ($m \leq N$).

Neste trabalho é considerado que o sistema de patrulhamento utilizando múltiplos robôs proposto seja utilizado para efetuar a segurança de um *shopping*, como apresentado na seção 1.2, em que foi descrito como os agentes de segurança deveriam agir para inibir possíveis furtos nas lojas desse *shopping*.

Para isso, foi definido que cada estação representa uma loja. Nos casos das lojas que estão desocupadas, foi definido que estas não são sujeitas a furtos e, como consequência, foram determinadas como estações de passagens. Por outro lado, as que estão ocupadas e, por isso, visadas por ladrões, são definidas como estações de alerta.

Outra consideração importante é em relação aos diferentes tipos de estabelecimentos. O fato de haver lojas comercializando diferentes produtos tornam as que vendem produtos valiosos mais atrativas para intrusos, sendo tal fato considerado no modelo apresentado, onde foi fixado que o parâmetro α (parâmetro responsável por definir a probabilidade das estações terem seu alerta ativado) fosse distinto para diferentes estabelecimentos desse *shopping*.

Em Asghar e Smith (2016), foi descrito uma importante consideração a respeito do comportamento de possíveis ladrões. É dito que se o sistema de patrulhamento for previsível, um intruso pode estudar como funciona o sistema e cometer o furto quando não houver patrulhamento naquele local.

Para evitar esse tipo de problema, é essencial que o controle não seja previsível, ou seja, que os ciclos de funcionamento do sistema de patrulhamento não sejam iguais durante todo o tempo. Um exemplo de como tornar o sistema não previsível, é fazer com que os robôs possam parar em uma estação mesmo que não haja alerta ativo. Outro exemplo é fazer com que os robôs não fiquem parados em uma estação por um período fixo. No caso do controle utilizando AG, essas duas características estão presentes.

O controle via heurística foi feito com o intuito de conseguir ganhar o máximo possível de recompensa considerando apenas cada estágio. Por outro lado, o controle utilizando o AG possui como característica maximizar o ganho de recompensa ao final de todo o processo. Por conta disso, o AG se torna uma solução interessante para esse tipo de

problema, uma vez que tenta maximizar a recompensa como um todo, posicionando os robôs nas estações de modo a prever a chegada de alertas e, desse modo, atendendo às solicitações de alertas ativados rapidamente, de modo a evitar sofrer punição por essa demora, e captar o máximo de informação.

4.1 PRIMEIRO CASO

Para o primeiro caso, foram adotadas oito estações ($N = 8$), sendo quatro as estações de alerta ($m = 4$), apenas dois robôs ($j = 2$) e um número de cem estágios ($iterações_do_robô = 100$). As estações das posições 0, 2, 4 e 5, foram definidas como estações de alerta.

Através da Figura 10 abaixo, pode-se observar que na medida em que as gerações do AG vão avançando, as soluções vão melhorando. Isso ocorre porque, de acordo com o método de seleção natural, as soluções melhores tendem a produzir mais descendentes. Outra observação é em relação à melhor solução encontrada na milésima sexagésima terceira (1063ª) geração. Pode-se notar que a melhor solução sempre é mantida para a geração seguinte devido ao uso do elitismo. Tal fato faz com que o AG retorne à melhor solução encontrada, independentemente da geração que a mesma foi encontrada.

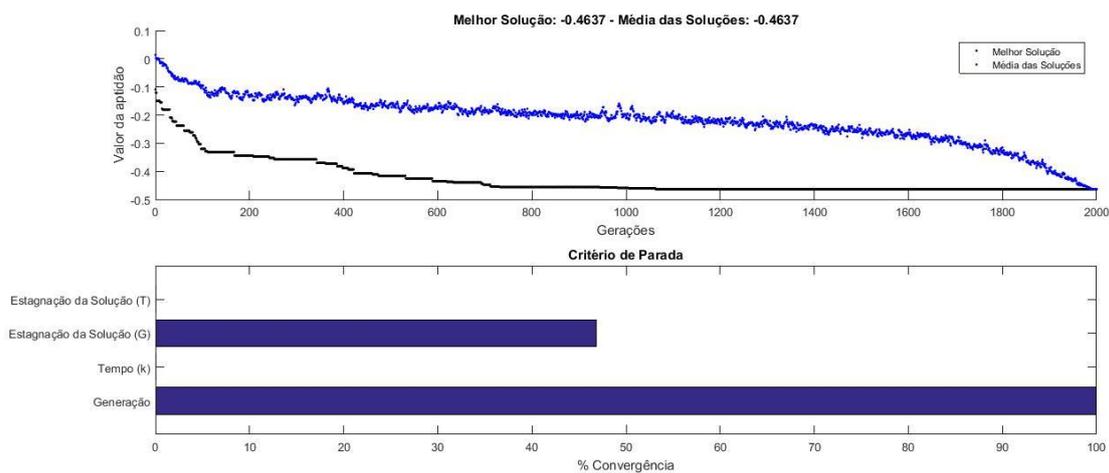


Figura 10 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

O processo de evolução da recompensa e da posição dos robôs é mostrado da Figura 11 à Figura 14. A comparação com o controle baseado no AG é feita com a segunda versão do controle heurístico em que o valor de $D = 3$. Para os dois casos, o sistema de patrulhamento

iniciou com as mesmas condições, ou seja, um robô iniciou na estação 4 e o outro a partir da estação 5.

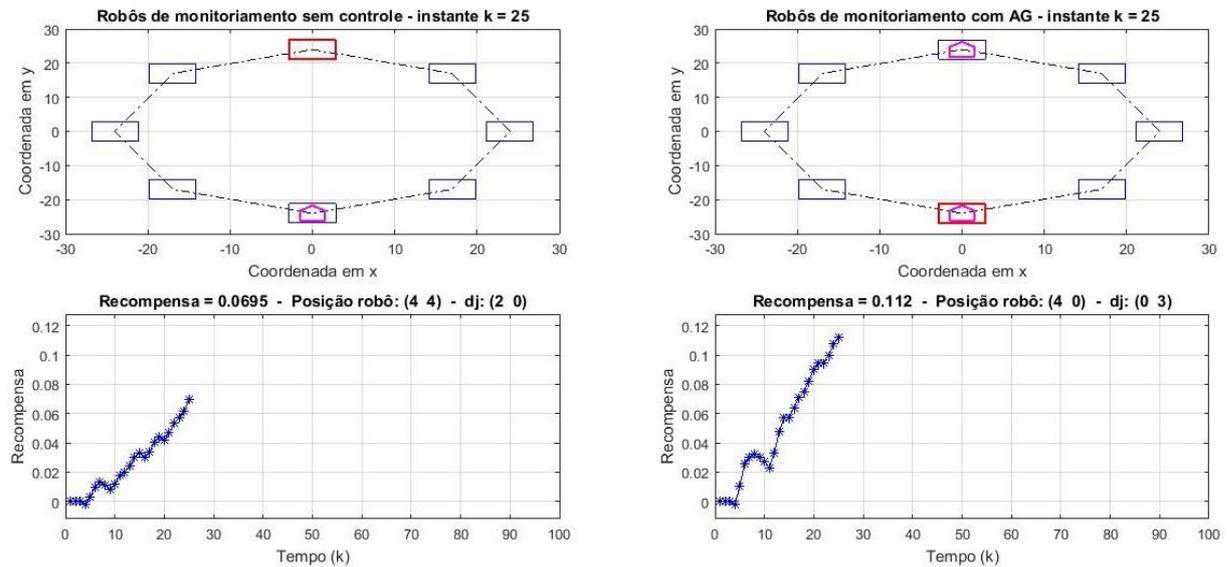


Figura 11 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 25$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 11 mostra que os dois sistemas estão agindo de forma diferente, a começar pela posição dos robôs e das estações com alertas ativados. Enquanto os robôs no sistema de patrulhamento utilizando AG estão em posições diferentes e distribuídos de modo a atender aos alertas de forma mais ágil, os do sistema de patrulhamento com o controle via heurística estão na mesma estação. O posicionamento espalhado dos robôs no mapa no sistema utilizando o controle baseado em AG exemplifica o fato já citado que diz que o controle utilizando AG é menos susceptível a sofrer penalização por demorar a atender uma estação com alerta ativo. Nota-se que no estágio $k = 25$ há um alerta ativado em ambos os sistemas. Porém, no caso do sistema de patrulhamento utilizando o AG, o robô se encontra na estação para atender a esse alerta rapidamente. O que não ocorre no caso do controle utilizando heurística, pois ambos os robôs estão distantes da estação com alerta ativo.

Por conta dessas questões citadas anteriormente, no estágio $k = 25$ a recompensa do sistema de patrulhamento controlado pelo AG é 61,15% superior a heurística.

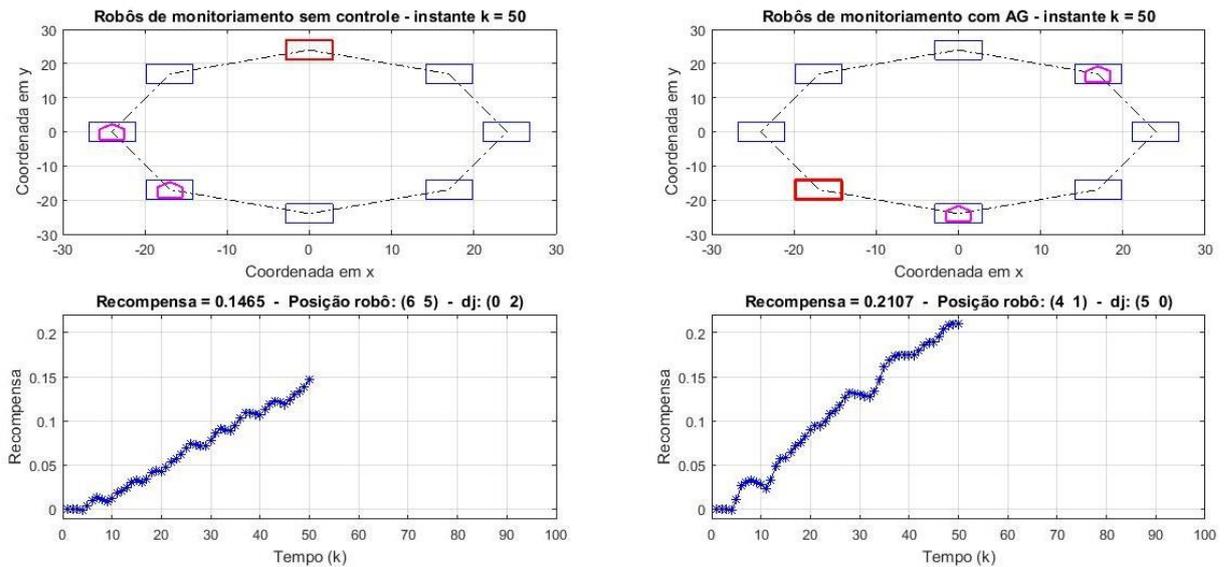


Figura 12 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 50$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Com o passar do tempo, a superioridade do AG se mantém. Porém, como pode ser visto na Figura 12, essa superioridade é reduzida para 43,82% no estágio $k = 50$. Novamente, os robôs do sistema de patrulhamento utilizando AG estão mais distantes um do outro e, como consequência, menos susceptíveis a sofrer penalização por demorar a atender uma estação com alerta ativado, o que não significa que nunca irá ocorrer essa penalização. Inclusive, a Figura 12 exemplifica um caso em que há um alerta ativado e os dois sistemas de patrulhamento demoram a chegar à estação para atender a esse alerta.

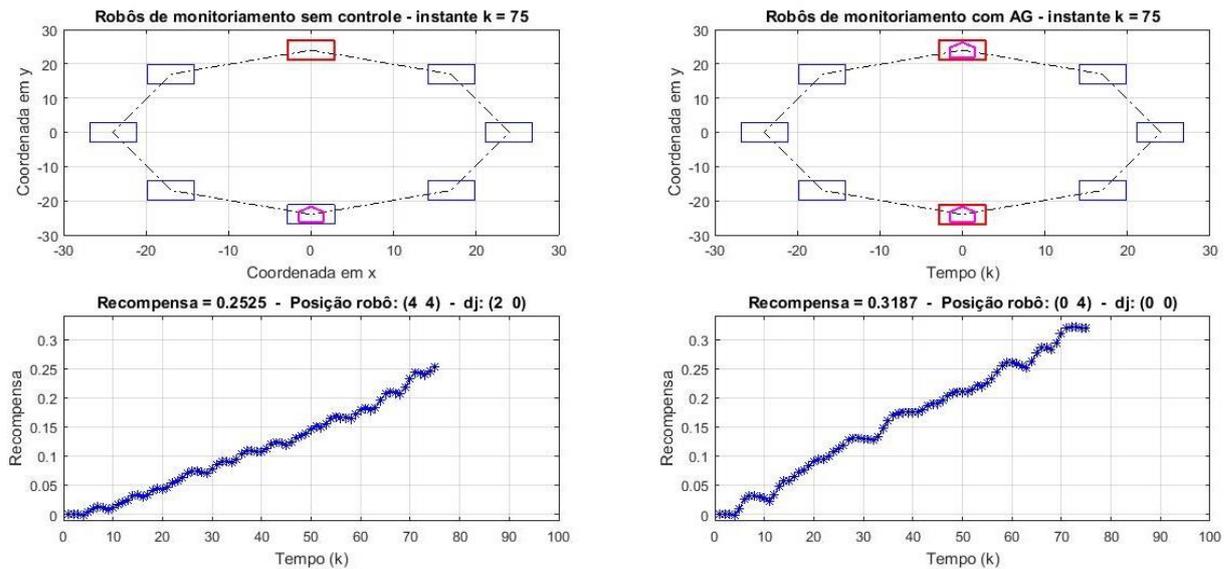


Figura 13 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 75$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 13 segue o raciocínio das figuras anteriores e mostra a comparação entre o progresso da estratégia dos controles. Como aconteceu entre os estágios $k = 25$ (Figura 11) e $k = 50$ (Figura 12), o sistema de patrulhamento utilizando o controle heurístico continuou a reduzir a diferença da recompensa em relação ao AG. Sendo que, no estágio $k = 75$, a recompensa do AG é 26,22% maior que a recompensa do controle heurístico.

Como aconteceu nos estágios anteriores, os robôs no sistema de patrulhamento utilizando o AG continuam melhores distribuídos no mapa. Comprovando essa vantagem, há o fato no estágio $k = 75$, em que os dois robôs chegaram para atender as duas estações que estão com alertas ativados. Por outro lado, no caso do sistema utilizando o controle heurístico, esse atendimento a uma das estações irá demorar.

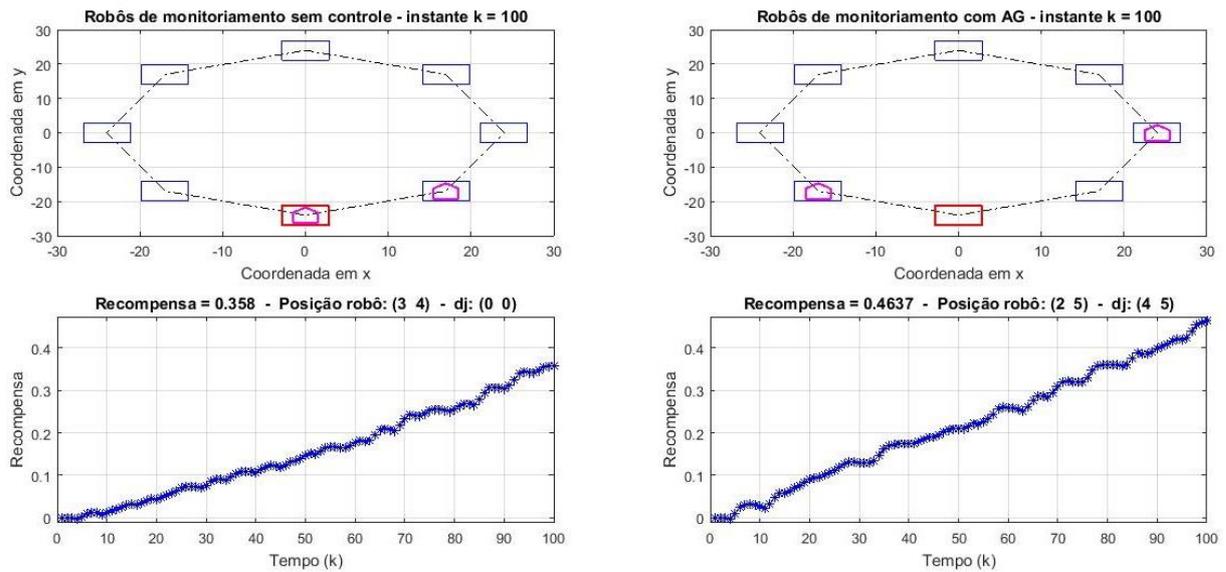


Figura 14 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dois Robôs e Oito Estações.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Por fim, na Figura 14 pode-se notar uma evolução superior do AG em relação ao controle heurístico, fazendo com que ao final dos estágios a recompensa do patrulhamento utilizando AG fosse superior em 29,53%.

Outra informação importante é em relação a atender todas as estações. Nos dois casos, ao final dos estágios, uma estação ainda estava com alerta ativado. Porém, no caso do sistema de patrulhamento utilizando o controle heurístico, o robô tinha chegado para atender ao alerta, o que iria demorar alguns estágios para acontecer com o sistema controlado pelo AG.

Durante o processo de estágios, foi gerado um vídeo no qual aparece o avanço da recompensa e a posição do robô para cada estágio. Este vídeo pode ser acessado através do link abaixo:

https://www.dropbox.com/s/64zaygoot1b10m1/caminho_roboto_int_100_est_8_roboto_2.avi?dl=0

4.2 SEGUNDO CASO

Com o intuito de aumentar a complexidade do problema, foi adotado um número maior de estações ($N = 20$), sendo nove as estações de alerta ($m = 9$). Nesse caso, foram escolhidos cinco robôs ($j = 5$) e cem estágios ($iterações_do_robô = 100$). Para este teste,

foram adotadas como estações de alerta, as localizadas nas posições 0, 2, 4, 6, 10, 11, 12, 15 e 18.

A Figura 15 abaixo apresenta o avanço da recompensa em relação ao avanço das gerações do AG. Pode-se notar que no começo a recompensa avança rapidamente, mas depois de algumas gerações esse avanço ocorre de maneira lenta até estabilizar na milésima sexcentésima quinquagésima primeira (1651^a) geração, quando não foi possível o AG encontrar outra solução melhor.

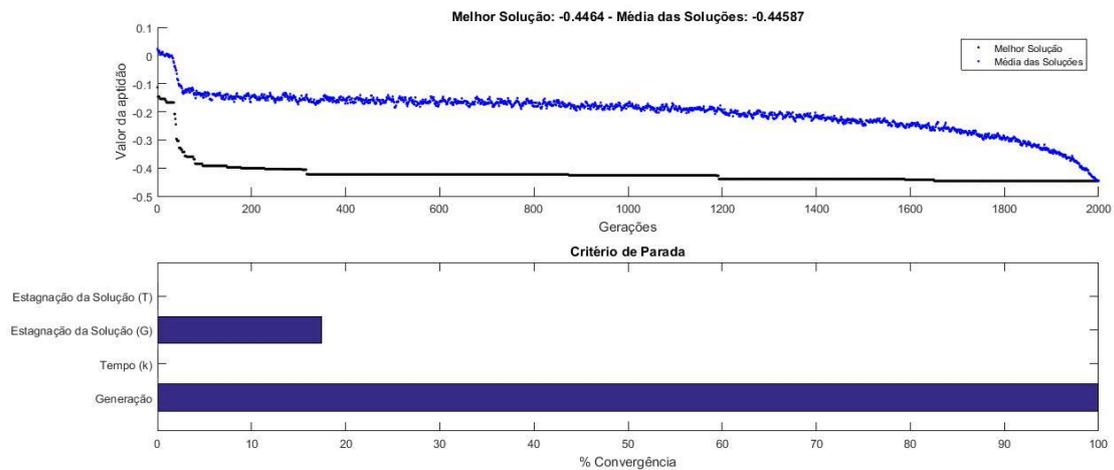


Figura 15 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Como no primeiro caso, da Figura 16 à Figura 19 será mostrada a evolução da recompensa e da posição dos robôs durante os estágios de evolução do sistema. As posições iniciais dos robôs são: 18, 7, 12, 12 e 2. Nesse caso, dois robôs estão saindo inicialmente da mesma estação.

Para ser comparado com o controle baseado no AG, foi selecionado o primeiro controle heurístico para a situação em que o número máximo de tempo em que o robô poderá permanecer em uma estação, será de três estágios ($D = 3$).

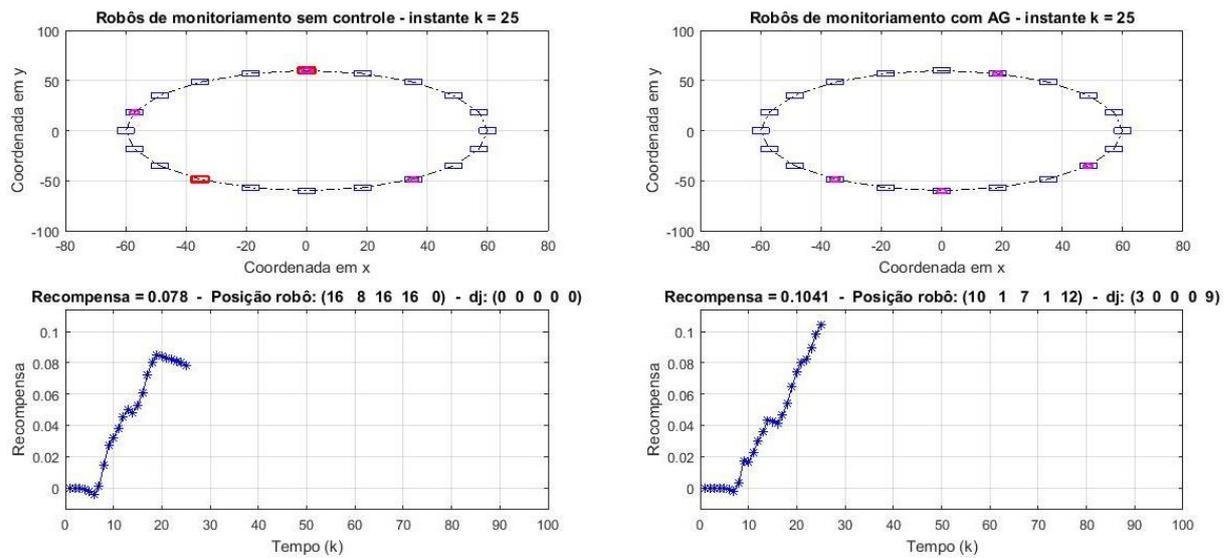


Figura 16 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 25$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Analisando a Figura 16, é possível observar que a recompensa utilizando AG supera o controle baseado em heurística já no estágio $k = 25$. Em valores percentuais, essa superioridade é de 33,46%. Também é possível observar que os robôs no AG atenderam a todas as requisições, enquanto que o controle heurístico ainda possuía duas estações para serem atendidas. Como ocorreu essa demora no atendimento a essas estações, a recompensa do controle heurístico começou a sofrer um decréscimo. Este decréscimo ocorre com o intuito de otimizar o controle para que os robôs, além de captarem o máximo de informação de cada estação em alerta, ainda atendam às estações que entrarem em alerta o mais breve possível.

Como o AG trata o problema de maneira global (analisando todos os estágios), o controle baseado no AG procura posicionar os robôs de modo a antecipar a chegada do robô à estação que entrará em alerta, a fim de que quando isso ocorrer se tenha um robô próximo a esta estação para atendê-la.

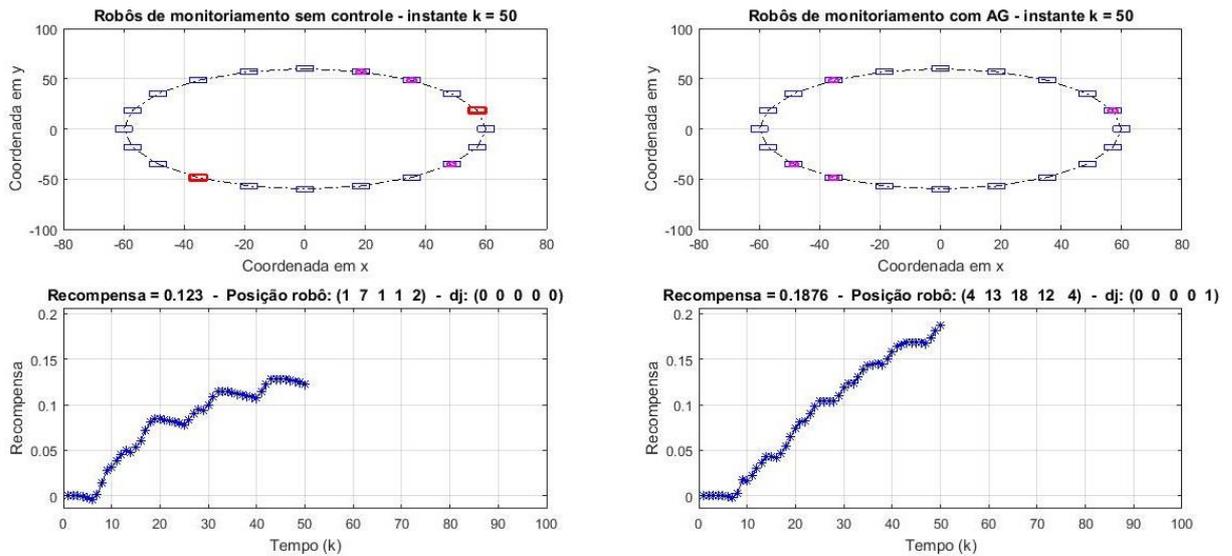


Figura 17 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 50$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 17 ilustra a diferença entre as recompensas, em que o AG é superior à heurística. No estágio $k = 50$, a recompensa já é superior a 52,5% em relação ao controle heurístico. Como justificativa à superioridade do controle AG, está o fato do controle heurístico continuar sendo mais penalizado, devido à demora em atender as estações que estão com alerta ativado, algo que ocorre com menor frequência no caso do AG.

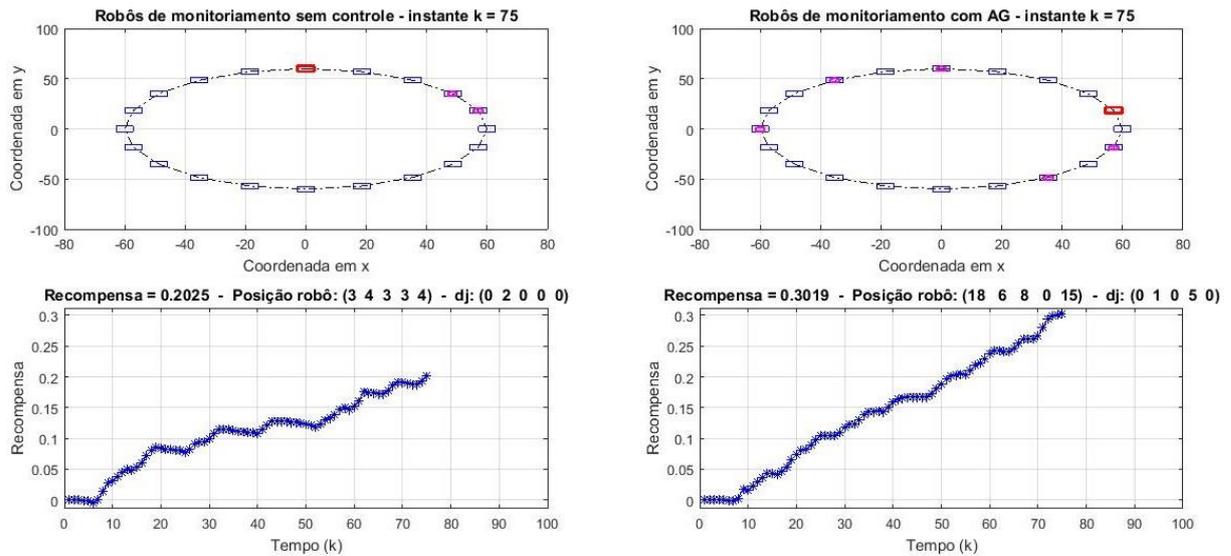


Figura 18 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Vinte Estações no Estágio $k = 75$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Analisando a Figura 18, nota-se no estágio $k = 75$ que os robôs estão posicionados de maneiras bem diferentes. Enquanto no AG todos os cinco robôs estão em estações diferentes e bem distribuídos no mapa, no controle heurístico estão concentrados em apenas duas estações que se encontram lado a lado.

Esse fato contribui de forma significativa para que a recompensa do controle AG seja superior ao controle heurístico em 49,09%.

Outra questão importante nessa análise é em relação à posição da estação que está como alerta ativado. No controle baseado AG, há um robô próximo a essa estação, ou seja, será atendida em pouco tempo. No caso do controle heurístico, como os robôs estão distantes da estação com alerta ativado, haverá uma demora significativa para atender ao alerta, gerando assim uma penalização durante esse período.

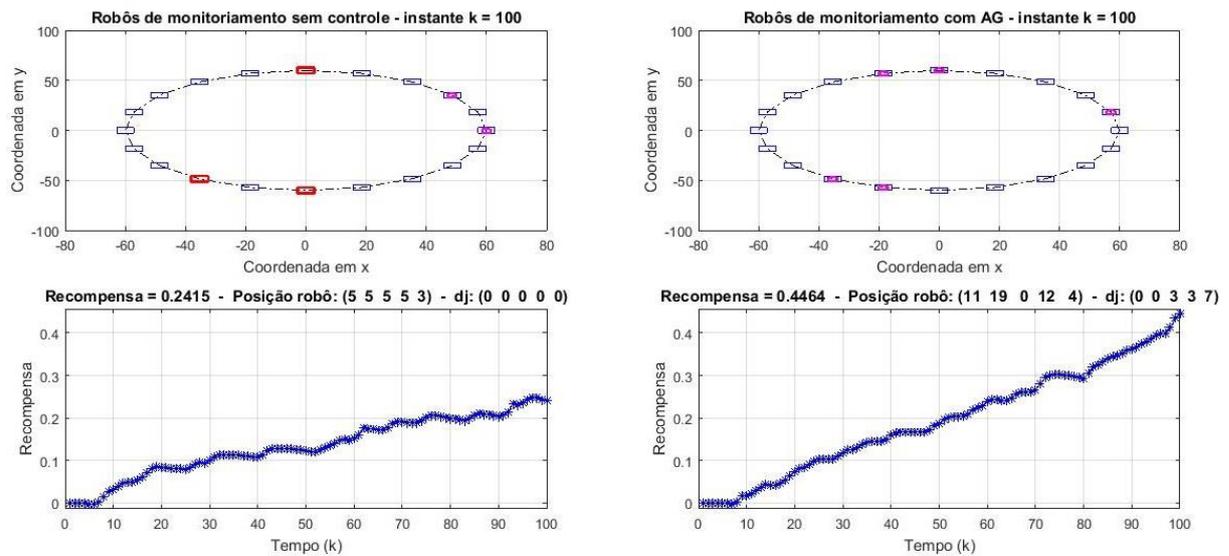


Figura 19 – Resultado Final da Evolução da Recompensa e da Posição Final dos Cinco Robôs e Vinte Estações.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Ao final deste processo, a Figura 19 mostra a diferença de 84,84% do AG em relação à heurística. Outro dado interessante nesta figura é o fato de o controle AG possuir as estações com alertas desativados, enquanto o sistema controlado via heurística possui no mesmo estágio três estações com alerta ativados. Tais fatos confirmam a maior eficiência do AG em relação ao controle heurístico, desde os primeiros estágios de tempo.

Da mesma forma que no caso anterior, foi gerado um vídeo para mostrar a evolução das posições dos robôs, bem como a recompensa durante o processo de estágios. Através do vídeo, é possível acompanhar essa evolução para cada estágio de tempo. Este vídeo pode ser acessado através do link abaixo:

https://www.dropbox.com/s/y23pgqabjhmljjg/caminho_roboto_int_100_est_20_roboto_5.avi?dl=0

4.3 TERCEIRO CASO

Para o terceiro caso, foi utilizado um número de trinta estações ($N = 30$), sendo doze as estações de alerta ($m = 12$). Para este caso, foram escolhidos dez robôs ($j = 10$) e cem estágios ($iterações_do_robô = 100$).

Foram adotadas como estações de alerta as estações nas posições 0, 2, 4, 6, 10, 11, 12, 13, 15, 18, 21 e 25.

Ao analisar a Figura 20 abaixo, pode-se notar o avanço da recompensa em relação às gerações. Como aconteceu nos casos anteriores, nas gerações iniciais há um grande avanço na seleção de soluções melhores. Contudo, depois de um tempo, a probabilidade de encontrar uma solução melhor que a já existente é menor. Pode-se notar que a melhor solução foi encontrada na milésima quadragésima vigésima sétima (1427^a) geração.

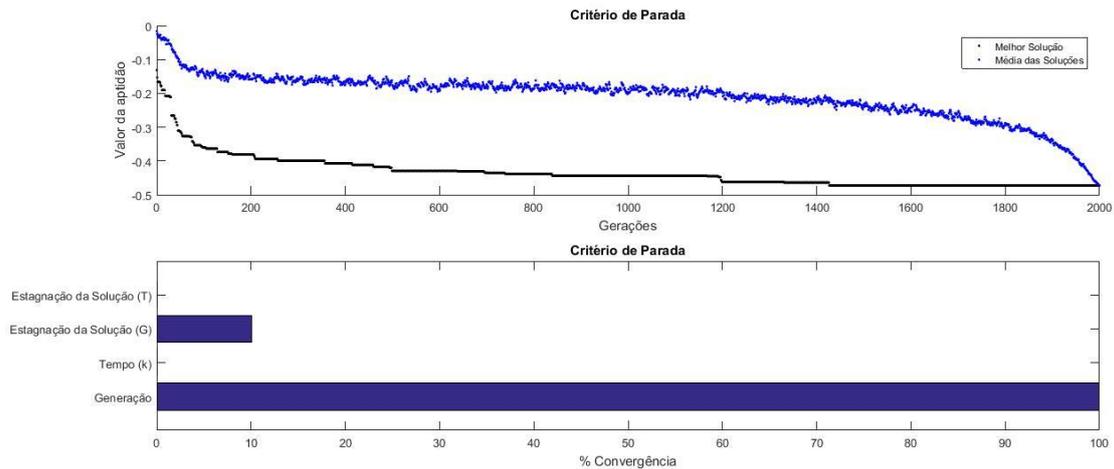


Figura 20 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Seguindo os modelos dos casos anteriores, da Figura 21 à Figura 24, é mostrada a evolução da recompensa e da posição dos robôs. Com o intuito de comparar o controle AG com um modelo do controle heurístico, foram feitos testes utilizando as diversas possibilidades de controle heurístico, em que se observou que para as mesmas condições, o controle heurístico 2 com $D = 3$ obteve a maior recompensa. As posições iniciais dos robôs são: 25, 6, 6, 10, 25, 11, 15, 22, 10 e 24.

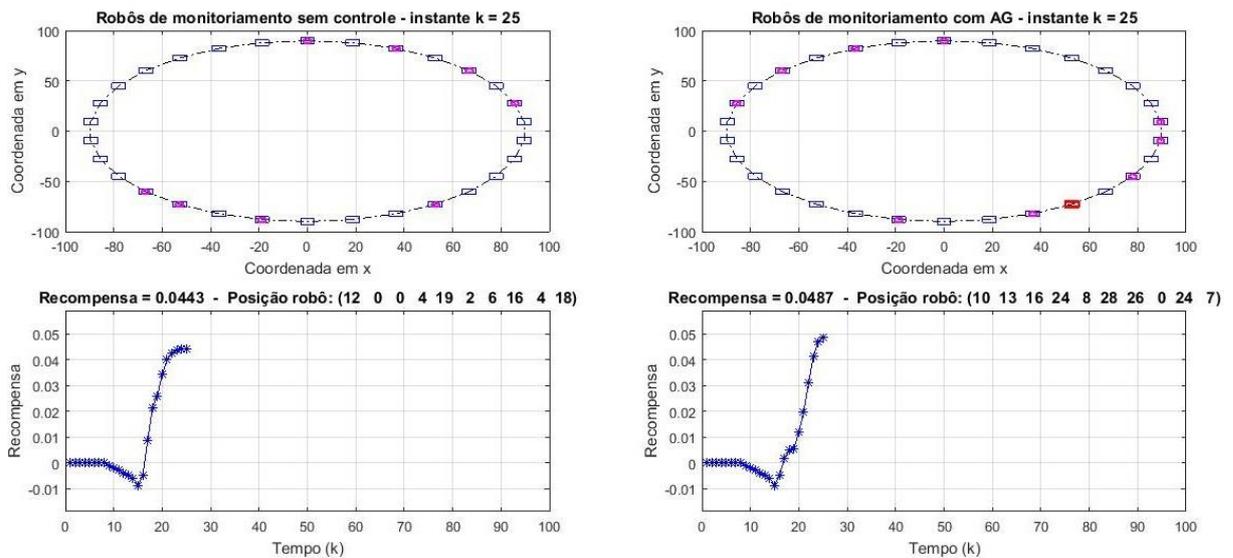


Figura 21 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 25$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Na Figura 21, é possível observar em ambos os controles que as recompensas nos estágios iniciais estão estabilizadas em zero. Isso ocorre porque durante esse período não ocorreu a chegada de nenhum alerta. No estágio $k = 8$, há a aparição do primeiro alerta, o que faz com que ocorra um decaimento da recompensa até o estágio $k = 16$. Neste estágio, os robôs começam a atender as estações de alerta.

Pode-se verificar que no estágio $k = 25$, há uma superioridade de 9,93% na recompensa do sistema de patrulhamento utilizando o AG. Porém, há uma estação com alerta ativado no sistema de controle com AG que causará uma penalização na recompensa do sistema nos próximos estágios, algo que não ocorrerá no caso do controle heurístico.

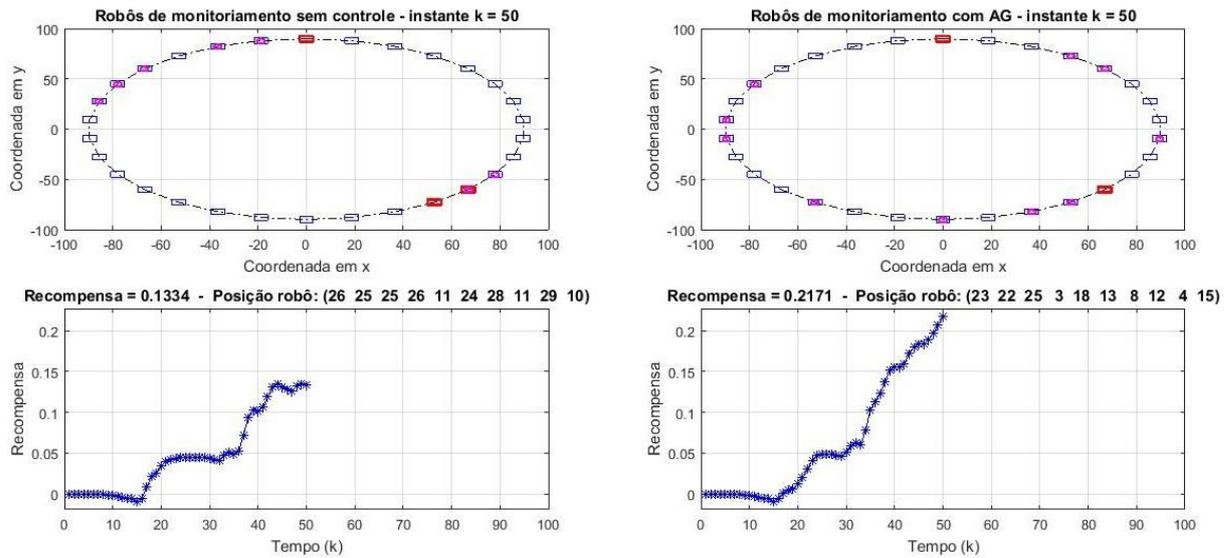


Figura 22– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 50$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Como observado anteriormente, a partir do estágio $k = 25$ ocorre um decréscimo da recompensa do AG, enquanto que o controle heurístico mantém a recompensa quase inalterada, como pode ser visto na Figura 22. Porém, a partir do estágio de tempo $k = 30$, a recompensa do controle AG volta a aumentar diferentemente do controle heurístico que demora mais alguns estágios. A partir do estágio de tempo $k = 34$, devido à diferença das posições dos robôs, a superioridade do AG volta a ampliar a diferença em relação ao controle heurístico, fazendo com que no estágio $k = 50$, haja uma diferença de 62,74% entre as recompensas.

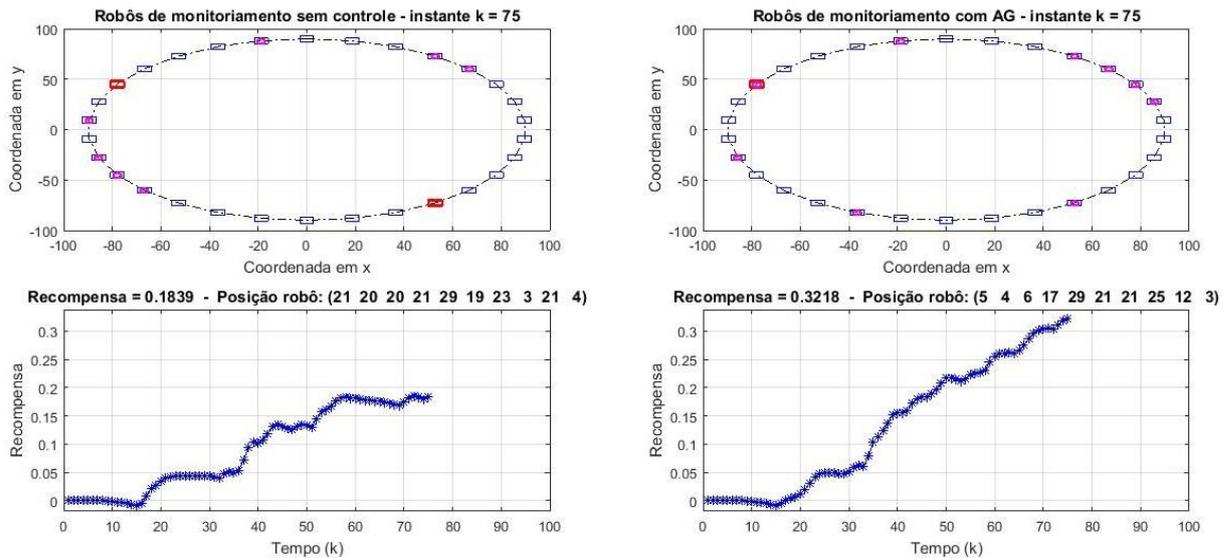


Figura 23 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Trinta Estações no Estágio $k = 75$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Na Figura 23, é possível observar que o avanço da recompensa ocorre de maneira distinta entre as duas estratégias, enquanto no sistema de patrulhamento utilizando AG a recompensa cresce de forma quase linear, o controle heurístico possui uma evolução pequena no mesmo período de tempo, fazendo com que no estágio $k = 75$ a diferença de recompensa seja de 74,99%.

Outro ponto a ser observado é em relação ao número de estações com alertas ativados. Enquanto no sistema utilizando AG há apenas uma estação com alerta ativo, no sistema com o controle heurístico há duas. Isso indica que a recompensa do controle heurístico irá sofrer uma penalização duas vezes maior que a recompensa do sistema utilizando AG.

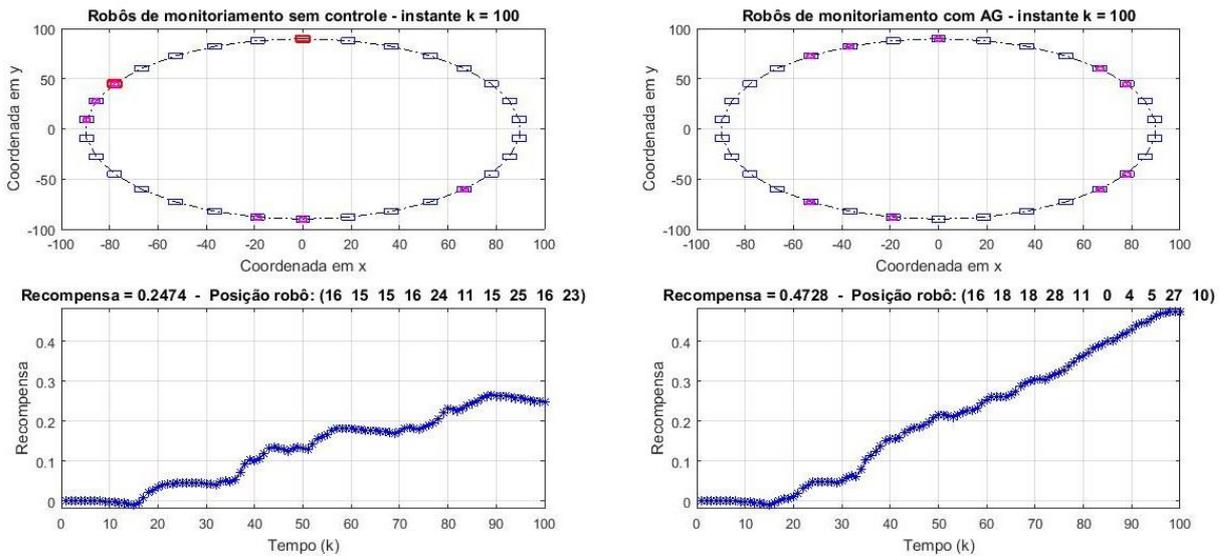


Figura 24 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dez Robôs e Trinta Estações.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Analisando a Figura 24, é nítida a diferença entre a evolução da recompensa entre o controle contendo o algoritmo heurístico e o controle utilizando AG, assim como a diferença em que mostra como ocorreu a evolução da recompensa em ambos os casos. Ao final dos estágios, a recompensa do sistema de patrulhamento utilizando AG foi 91,1% superior em relação à recompensa do controle heurístico.

Outro fato a ser observado é em relação às estações com alerta ativo ao final do processo dos estágios. No caso do sistema de controle utilizando o AG, todas as estações estão com alerta desativados. Já o sistema de patrulhamento utilizando heurística possui duas estações com pedidos de alerta ativados, o que significa que o sistema utilizando o controle heurístico iria sofrer decréscimo na sua recompensa por conta da demora a atender essas solicitações de alerta.

Para que esse sistema de patrulhamento possa ser analisado em cada estágio, está sendo disponibilizado abaixo um link que contém o vídeo mostrando a evolução do sistema de patrulhamento utilizando múltiplos robôs controlados pelo AG e pela heurística. Eis o link:

https://www.dropbox.com/s/1uszbdnzs5ch60/caminho_roboto_int_100_est_30_roboto_10.avi?dl=0

4.4 QUARTO CASO

Neste último caso, a exemplo do primeiro, foram adotadas oito estações ($N = 8$), sendo quatro as estações de alerta ($m = 4$), apenas dois robôs ($j = 2$) e cem estágios ($iterações_do_robô = 100$). Foram fixadas como estações de alerta as estações nas posições 0, 2, 4 e 5.

Diferente do que ocorreu nos casos anteriores, desta vez o controle utilizando o Algoritmo Genético não será comparado com o controle via heurística. Esta comparação será feita entre dois controles utilizando o AG. O que irá diferenciar ambos é o número de gerações e o número da população utilizada.

A Figura 25 a seguir mostra a evolução da recompensa quando utilizados duzentos indivíduos na população e o número máximo de duzentas gerações. Pode-se notar que até por volta da octogésima (80^a) geração houve um avanço mais acentuado da recompensa. Após essa geração, a evolução ocorreu de forma mais lenta. Sendo que a última evolução da recompensa aconteceu na centésima quinquagésima sexta (156^a) geração.

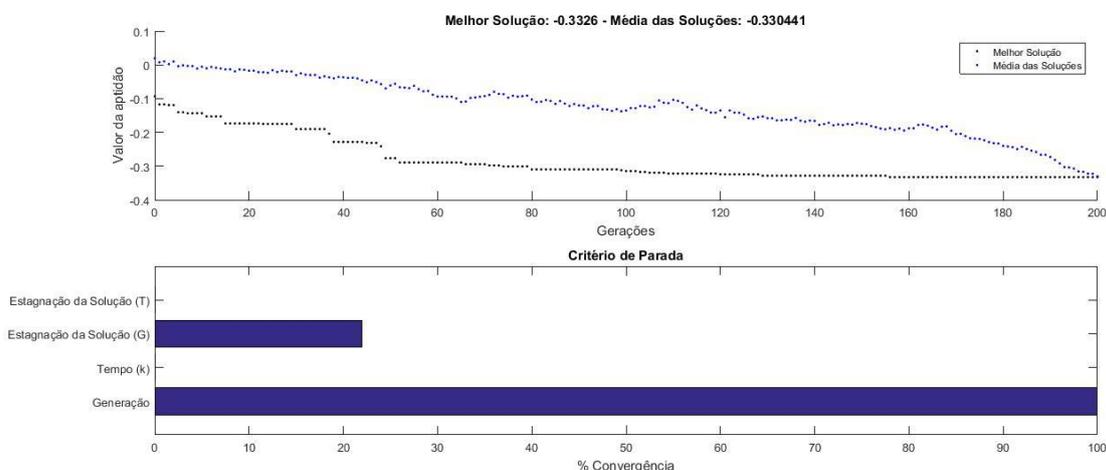


Figura 25 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG Utilizando o Máximo de 200 Gerações e 200 de População.
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 26 abaixo mostra a evolução da recompensa quando utilizando dois mil indivíduos na população e o número máximo de duas mil gerações. Neste caso, a exemplo do que ocorreu na Figura 25, há um avanço da recompensa de maneira mais rápida no começo do algoritmo, porém, como o número da população é maior, esse avanço foi mais acentuado até a ducentésima vigésima sexta (226^a) geração. A partir desse ponto, a recompensa continuou avançando de maneira um pouco mais lenta até a noningentésima nonagésima primeira (991^a)

geração. Até atingir as duas mil gerações, só houve mais um avanço na recompensa, que ocorreu na milésima quingentésima sexagésima sexta (1566^a) geração. Tal fato pode ter ocorrido devido a uma mutação que fez com que a solução que estava estagnada durante quinhentas e setenta e cinco (575) gerações, em um ponto local, conseguisse convergir para uma solução melhor.

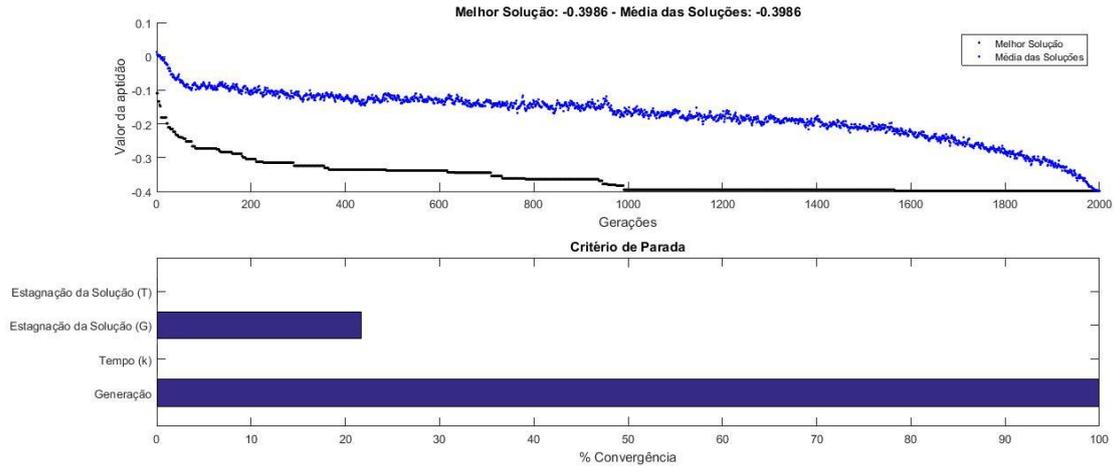


Figura 26 – Evolução da Recompensa em Virtude do Avanço das Gerações do AG Utilizando o Máximo de 2000 Gerações e 2000 de População
Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Da Figura 27 à Figura 30, são mostrados como acontece a evolução da recompensa e da posição dos robôs. Para os dois casos forma utilizados as mesmas condições iniciais, ou seja, um robô iniciou na estação 3 e o outro na estação 4.

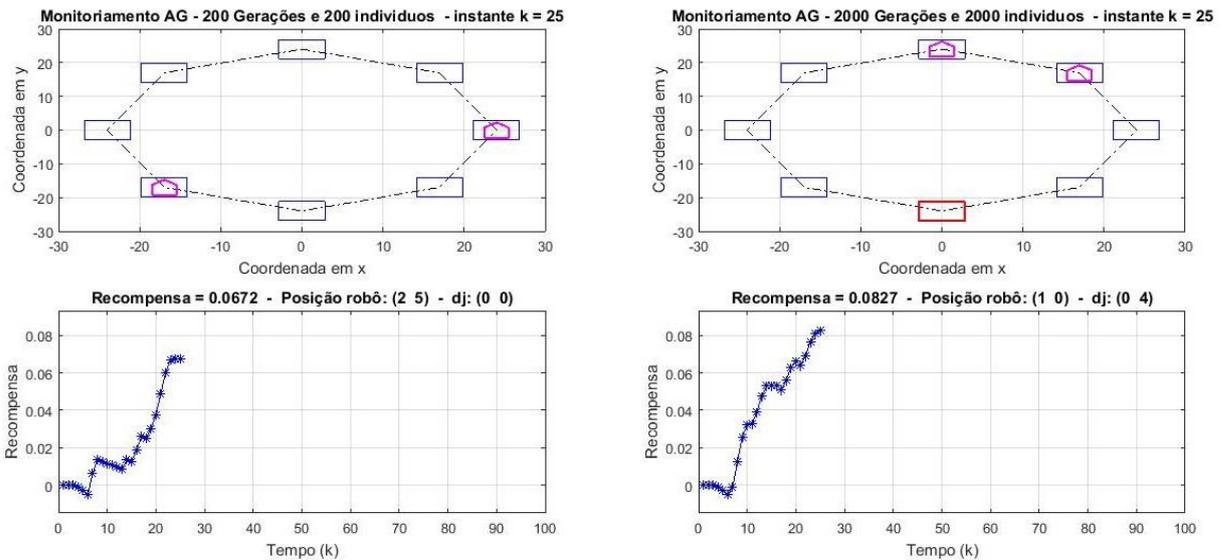


Figura 27 – Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 25$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Analisando a Figura 27, percebe-se que nos primeiros estágios há uma diferença entre o comportamento dos dois controles. No estágio em que $k = 25$, observa-se que os robôs estão em posições diferentes, sendo que o controle com 200 indivíduos estava melhor distribuído no mapa, enquanto que o controle com 2000 indivíduos estava com os robôs próximos. Outro fator favorável ao controle com 200 indivíduos é o fato de possuir todas as estações com alerta desativados neste estágio, ou seja, esse controle já atendeu a todos os alertas ativados. Algo que não ocorreu com a outra estratégia de controle (2000 indivíduos), o que gerou sua penalização. Como ponto desfavorável ao controle com 200 indivíduos é justamente a sua recompensa ser menor que a recompensa do controle AG com 2000 indivíduos. Por conta do posicionamento e do tempo que permaneceu em cada estação, os robôs do controle com 2000 indivíduos conseguiram captar uma recompensa 23,07% superior ao controle com 200 indivíduos.

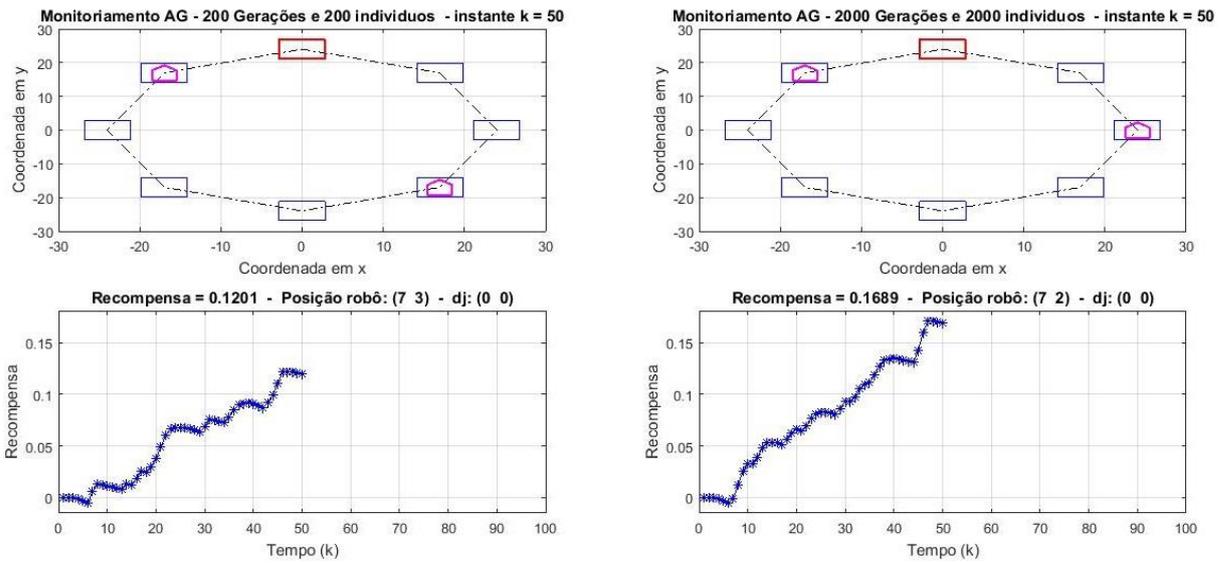


Figura 28– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 50$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

No estágio $k = 50$, como mostra a Figura 28, pode-se notar semelhanças na distribuição dos robôs no mapa e em possuir uma estação com alerta ativado. O que difere os dois casos é justamente a recompensa, em que o controle com 2000 indivíduos aumentou a superioridade em 40,63% com relação ao controle com 200 indivíduos.

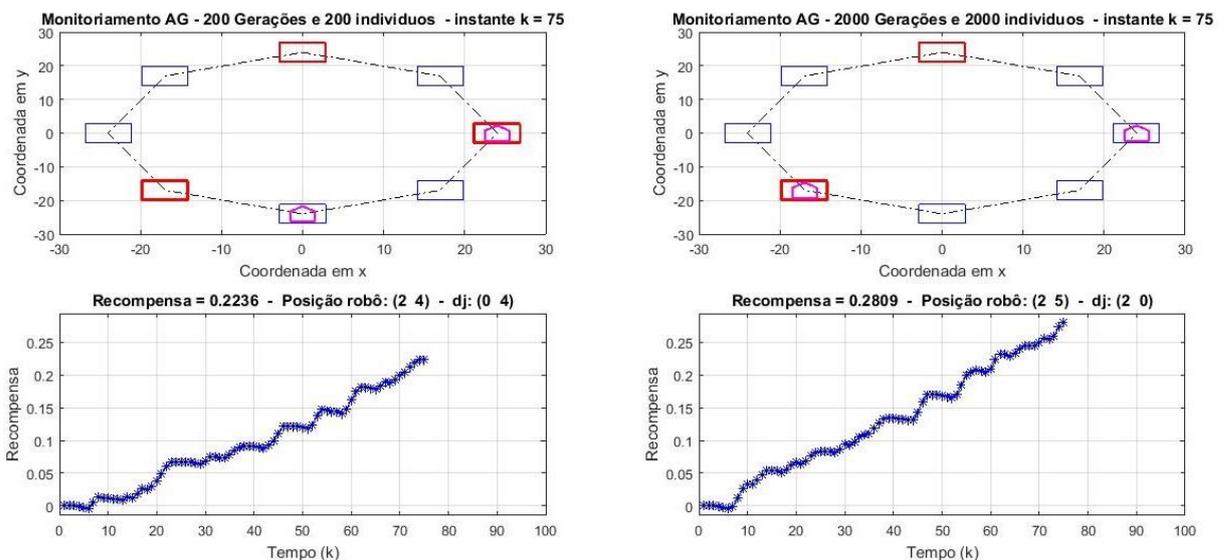


Figura 29– Evolução da Recompensa e da Posição dos Robôs em Função do Tempo para Oito Estações no Estágio $k = 75$.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

A Figura 29 continua a mostrar o progresso das duas estratégias de controle, porém agora no estágio $k = 75$. Neste estágio, os dois controles continuam a manter os dois robôs em posições parecidas. Outra questão a se observar é o fato que nesse estágio houve uma redução de 25,63% na superioridade do controle com 2000 indivíduos em relação ao controle com 200 indivíduos.

Já o tempo que se demora a atender a uma solicitação, inverteu-se. Enquanto que na Figura 27 ($k = 25$) o controle com 2000 indivíduos foi o que demorou a atender a uma solicitação, na Figura 28 ($k = 50$) os dois controles estavam iguais nesse quesito. A Figura 29 mostra essa inversão, pois no estágio $k = 75$, o controle que demora a atender ao alerta é o controle com 200 indivíduos.

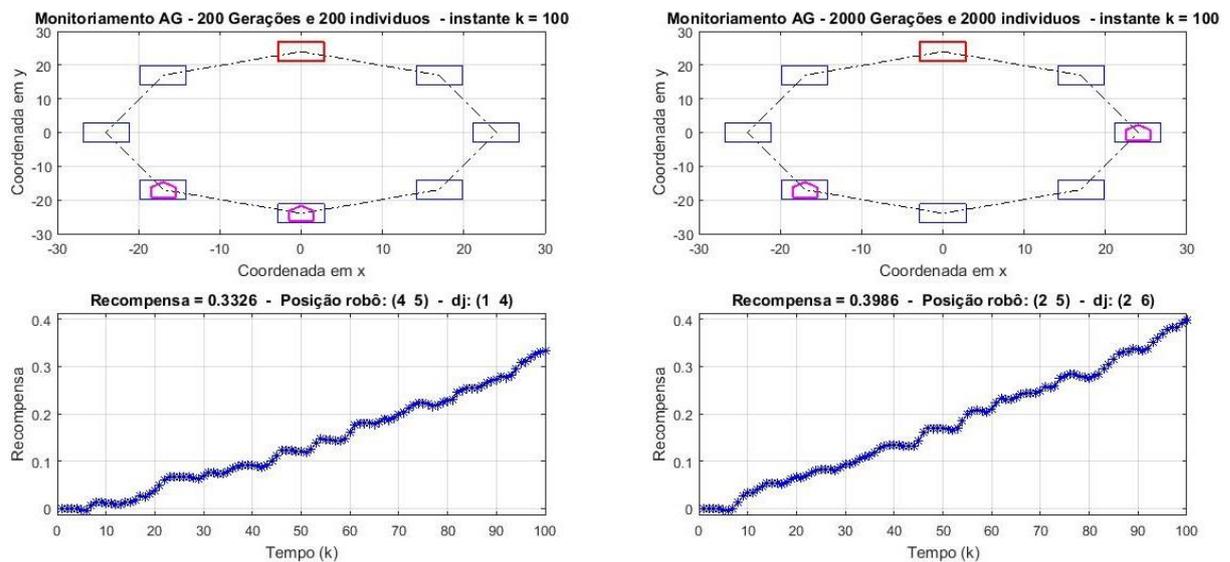


Figura 30 – Resultado Final da Evolução da Recompensa e da Posição Final dos Dois Robôs e Oito Estações.

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Finalizando a análise desta simulação, a Figura 30 mostra o comportamento do sistema de patrulhamento para o estágio $k = 100$. Nesta figura, pode-se notar que para ambos os sistemas há uma estação com alerta ativo.

A distribuição dos robôs se difere, enquanto o controle utilizando 200 indivíduos está com os robôs próximos, o controle com 2000 indivíduos possui os robôs mais espalhados no mapa.

Da mesma maneira que aconteceu nos estágios anteriores, para $k = 100$, a recompensa do controle com 2000 indivíduos foi superior ao controle com 200 indivíduos. Ao

final dos estágios, a recompensa para o controle com 2000 indivíduos foi 19,84% superior a recompensa do controle com 200 indivíduos.

O fato de dois controles utilizando o mesmo Algoritmo Genético, apenas com dois parâmetros distintos (população e número máximo de gerações), ter dado soluções diferentes é explicado pela teoria do AG. Por este algoritmo ser heurístico e combinatório, não há como garantir que será encontrado um ótimo global independentemente do número de indivíduos ou do número de gerações. Porém, com o aumento do número de gerações, a probabilidade de se aproximar da solução ótima global da função objetivo torna-se maior.

Como ocorreu nos casos anteriores, também foi gerado um vídeo no qual aparece o avanço da recompensa e a posição do robô para cada estágio. Este vídeo pode ser acessado através do link abaixo:

https://www.dropbox.com/s/8fecrgz802qrkiz/caminho_robo_dois_controles_AG.avi?dl=0

4.5 CONCLUSÃO DO CAPÍTULO

As simulações foram realizadas no *software* MATLAB®, em um sistema operacional Windows 7 Ultimate - 64 bits. Utilizou-se um computador com as seguintes configurações: processador Intel® Core™ i5-2430M, 2,40 GHz e 6,00 GB de memória RAM.

Com o intuito de facilitar a comparação entre os casos mostrados anteriormente, foi elaborado uma tabela com os dados e resultados dos casos apresentados, como pode ser visto na Tabela 1 a seguir.

Na Tabela 1, pode-se observar a evolução da recompensa em relação à complexidade do problema. Quanto mais complexo foi o problema, maior foi a vantagem do AG em relação ao controle baseado em heurística.

Tabela 1 – Resumo dos Resultados Encontrados.

	Caso 1		Caso 2		Caso 3	
Algoritmo	Heurístico	AG	Heurístico	AG	Heurístico	AG
Estágio	0,0695	0,112	0,078	0,1041	0,0443	0,0487
$k = 25$	AG – 61,15%		AG – 33,46%		AG – 9,93%	
Estágio	0,1465	0,2107	0,123	0,1876	0,1334	0,2171
$k = 50$	AG – 43,82%		AG – 52,5%		AG – 62,74%	
Estágio	0,2525	0,3187	0,2025	0,3019	0,1839	0,3218
$k = 75$	AG – 26,22%		AG – 49,09%		AG – 74,99%	
Estágio	0,358	0,4637	0,2415	0,4464	0,2474	0,4728
$k = 100$	AG – 29,53%		AG – 84,84%		AG – 91,1%	
Número de robôs	$j = 2$		$j = 5$		$j = 10$	
Número de estações	$N = 8$		$N = 20$		$N = 30$	
Número de estações de alerta	$m = 4$		$m = 9$		$m = 12$	
Tempo de máquina AG (seg)	885,763		1641,376		3191,648	
Tempo de máquina Total (seg)	931,848		1693,052		3250,750	

Fonte: ELABORADO PELO PRÓPRIO AUTOR.

Outro dado importante a ser observado é que em todos os estágios analisados, e em todos os casos, sempre o algoritmo baseado no AG teve uma superioridade na recompensa.

Como ponto negativo em relação ao controle utilizando AG, está o tempo necessário para encontrar a solução. Quanto maior era a complexidade do problema, maior era o tempo de máquina necessário para rodar o AG.

5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como principal objetivo otimizar um sistema de patrulhamento por múltiplos robôs. A fim de viabilizar uma maior eficiência, esse sistema foi composto por multi-robôs atuando de forma cooperativa.

Para ser possível otimizar um sistema de patrulhamento, foi efetuado um estudo no qual se definiu qual modelo seria utilizado neste trabalho. No capítulo 2, foi levantado o modelo matemático escolhido para ser otimizado, assim como descrito o seu funcionamento. Nesse capítulo, também foi apresentada a função objetivo do problema em forma de recompensa do sistema. A equação que gerava a recompensa do sistema foi composta de duas partes. A primeira parte da equação da recompensa representa o ganho do sistema em função da quantidade de informação adquirida pelo robô. A segunda parte da equação que calcula a recompensa consiste em uma punição pela demora a atender um alerta ativado.

Após as definições da escolha do sistema de patrulhamento e da técnica que seria utilizada para otimizar este sistema, foram apresentadas duas opções de controladores, no capítulo 3. A primeira opção foi um controlador baseado em heurística, ou seja, com regras de decisão previamente definidas. A segunda opção foi de um controlador baseado no Algoritmo Genético, o qual tinha como objetivo maximizar a recompensa do sistema como um todo.

Finalizando o trabalho, foram realizadas simulações de quatro casos, sendo os três primeiros casos tratando o problema com complexidades diferentes e comparando as duas estratégias de controle. No quarto e último caso, foi apresentada uma simulação comparando dois controles utilizando o AG, porém com variáveis de entradas diferentes.

Através da análise dos resultados dos três primeiros casos, pode-se concluir que a escolha pela utilização do AG para otimizar o desempenho do sistema de patrulhamento é vantajosa. O algoritmo conseguiu obter resultados melhores que o outro controle em todos os casos simulados, independentemente do número de estações (N), de estações de alerta (m) ou de robôs (j).

Outro ponto importante a ser abordado é a eficiência do controlador AG em relação ao controle heurístico em casos de maior complexidade. Foi observado que quanto maior a complexidade do problema, melhor foi o desempenho do AG em relação à utilização da heurística. Pode-se notar que dado o fato desse controlador considerar o horizonte de tempo à frente, foi possível observar que a distribuição dos robôs no mapa permitia um atendimento mais ágil nas estações com alerta ativado, assim como uma maior aquisição de informações do local.

O quarto caso teve como objetivo avaliar parâmetros do AG (tamanho da população e número de gerações). A intenção neste caso foi comprovar que, com o aumento do número das gerações e da população, a probabilidade de se aproximar da solução ótima global da função objetivo se torna maior. Tal fato ocorre devido às características não determinísticas desse algoritmo.

Portanto, os principais objetivos foram cumpridos conseguindo assim um sistema de patrulhamento com múltiplos robôs que fosse eficiente independentemente da quantidade de robôs ou de estações a serem monitoradas.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

A continuação natural deste trabalho possui 2 vertentes. A primeira seria a implementação deste algoritmo em sistemas reais na prática e a outra seria o aperfeiçoamento do algoritmo.

No caminho da implementação deste sistema na prática, há vários pontos a serem verificados. Para que seja possível fazer a integração do controlador com o robô, é necessário efetuar um estudo sobre a duração da constante de tempo k . Outro ponto é efetuar a implementação de um sistema de aquisição de informações através de sensores e câmeras.

Outra opção de trabalho futuro seguindo a ideia de implementação deste sistema de patrulhamento na prática é efetuá-lo utilizando veículo aéreo não tripulado (VANT), ou veículos com bases terrestres, ou até mesmo um sistema de patrulha misto, utilizando veículos aéreos e terrestres autônomos.

Para efetuar o aperfeiçoamento deste algoritmo, há vários estudos que podem ser feitos. Uma possibilidade é realizar um sistema de processamento de imagens com intuito de comparar a imagem atual obtida pelo robô com uma imagem anterior da estação. Com esse processamento, o robô já seria capaz de detectar se houve alteração na estação ou não.

Outro ponto que é interessante de ser implementado é a aplicação de outras técnicas inteligentes a esse sistema com o intuito de conseguir um resultado melhor.

Neste trabalho, foi utilizado um sistema determinístico, em que o controlador tinha informações prévias de quando uma estação iria entrar em alerta. A principal melhoria é a implementação de um controlador baseado em otimização estocástica.

Outra opção é investigar outros sistemas de aquisição de recompensa e generalizar esse método para que possa ser utilizado em outras aplicações diferente da aplicação em segurança.

REFERÊNCIAS

AGMON, N., FOK, C. L., EMALIAH, Y., STONE, P., JULIEN, C., & VISHWANATH, S.: **On coordination in practical multi-robot patrol**. In: Robotics and Automation (ICRA) 2012 IEEE International Conference on, pp. 650-656. IEEE, Saint Paul, MN (2012).

AGMON, N., KRAUS, S., & KAMINKA, G. A.: **Multi-robot perimeter patrol in adversarial settings**. In: Robotics and Automation (ICRA) 2008 IEEE International Conference on, pp. 2339-2345. IEEE, Pasadena, CA (2008).

AMAZON PRIME AIR. In: Amazon. Amazon Prime Air. Disponível em: https://www.amazon.com/b?ref=tsm_1_yt_s_amzn_mx3eqp&node=8037720011. Acesso em: 29 ago. 2016.

ANDRADE, Luiz Augusto Canito Gallego de.: **Heurística baseada em colônia artificial de abelhas para o problema de localização de bases, alocação e realocação de ambulâncias**. Dissertação (Mestrado em Engenharia de Sistemas Logísticos) - Escola Politécnica, University of São Paulo, São Paulo, 2012. doi:10.11606/D.3.2012.tde-17062013-144228. (2012). Acesso em: 28 ago. 2016.

ASGHAR, A. B., & SMITH, S. L.: **Stochastic Patrolling in Adversarial Settings**. In: 2016 American Automatic Control Council (AACC), pp. 6435-6440. IEEE, (2016).

BARRERA, J., & FLORES, J. J.: **Search of Initial Conditions for Dynamic Systems using Intelligent Optimization Methods**. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007), pp. 348-353. IEEE, (2007).

BASTOS-FILHO, C. J., SILVA, W. A., & LIRA, L. R.: **Comparing meta-heuristics for AdaBoost training applied to platelets detection**. In: IEEE Latin America Transactions, v. 12, n. 5, pp. 942-950. IEEE, (2014).

CASAR, J. & STEFEK, A.: **Utilization of optimization methods cooperation for the improvement of optimization process**. In: Military Technologies (ICMT), 2015 International Conference on, pp. 1-7. IEEE. (2015).

CIOARGA, R., PANUS, B., OANCEA, C., MICEA, M. V., CRETU, V., & PETRIU, E.: **Fish shoal inspired movement in robotic collectives**. In: *Robotic and Sensors Environments*, 2008. ROSE 2008. International Workshop on, pp. 7-12. IEEE, (2008).

CLARK, J. & FIERRO, R.: **Mobile robotic sensors for perimeter detection and tracking**. *ISA Transactions*, vol. 46, pp. 3–13, (2007).

CORDEIRO, J. A., PARPINELLI, R. S., & LOPES, H. S.: **Análise de Sensibilidade dos Parâmetros do Bat Algorithm e Comparação de Desempenho**. (2012). Acesso em: 28 ago. 2016.

DA AGÊNCIA EFE. In: Portal G1. **China inaugura restaurante com 20 robôs como garçons e cozinheiros**. Disponível em: <http://g1.globo.com/mundo/noticia/2013/01/restaurante-com-20-robos-como-garcons-e-cozinheiros-e-inaugurado-na-china.html>. Acesso em: 29 ago. 2016.

ELMALIACH, Y., SHILONI, A., & KAMINKA, G. A.: **A realistic model of frequency-based multi-robot polyline patrolling**. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems – v. 1*, pp. 63-70. (2008).

FURTADO, L., NETO, W. A., MENEZES, A., DA SILVA, I. C., OLIVI, L. R., & MARCATO, A. L. M.: **Bat Search Algorithm Aplicado na Localização De Robôs Moveis**. In: *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, pp. 1248-1253. Natal-RN, (2015).

GHADIRY, W., HABIBI, J., AGHDAM, A. G., & ZHANG, Y.: **Time-efficient trajectory optimization in patrolling problems with non-prespecified depots and robots**. In: *24th Mediterranean Conference on Control and Automation (MED)*. pp. 1047-1052. IEEE (2016).

HWANG, K. S., LIN, J. L., & HUANG, H. L.: **Cooperative patrol planning of multi-robot systems by a competitive auction system**. In: *ICCAS-SICE*, pp. 4359-4363. IEEE, Fukuoka (2009).

KARTAL, B., GODOY, J., KARAMOUZAS, I., & GUY, S. J.: **Stochastic tree search with useful cycles for patrolling problems**. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1289-1294. IEEE, Seattle, WA (2015).

KRISHNAMOORTHY, K., PACHTER, M., CHANDLER, D. & DARBHA, S.: **UAV perimeter patrol operations optimization using efficient dynamic programming**. In: American Control Conference, pp. 462-467. IEEE, San Francisco, CA (2011).

LAURI, F., & KOUKAM, A.: **Hybrid ACO/EA algorithms applied to the multi-agent patrolling problem**. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 250-257. IEEE, Beijing (2014).

LISA VAAS. In: Naked Security. **Amazon para começar a testar as entregas de drones**. Disponível em: <https://nakedsecurity.sophos.com/pt/2016/07/27/amazon-to-start-testing-drone-deliveries/>. Acesso em: 29 ago. 2016.

MARINO, A., CACCAVALE, F., PARKER, L. E., & ANTONELLI, G.: **Fuzzy behavioral control for multi-robot border patrol**. In: Control and Automation, 2009. MED'09. 17th Mediterranean Conference on, pp. 246-251. IEEE, Thessaloniki (2009).

MARJOVI, A., NUNES, J. G., MARQUES, L., & DE ALMEIDA, A.: **Multi-robot exploration and fire searching**. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1929-1934. IEEE, (2009).

MOTA, J. F. B., SILVA JUNIOR, I. C., WILLER, L., DE SÁ, R. J. F.: **Otimização da RCB de Projetos de Eficiência Energética do Tipo Baixa Renda via Ant Colony**. In: 2014 IEEE/IAS International Conference on Industry Applications, IEEE/IAS INDUSCON 2014, Juiz de Fora (2014).

PORTUGAL, D., COUCEIRO, M. S., & ROCHA, R. P.: **Applying bayesian learning to multi-robot patrol**. In: 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1-6. IEEE, Linkoping (2013).

REID, S. G., & MALAN, K. M.: **Constraint Handling Methods for Portfolio Optimization Using Particle Swarm Optimization**. In: Computational Intelligence, 2015 IEEE Symposium Series on, pp. 1766-1773. IEEE, (2015).

RENDERS, J. M., & FLASSE, S. P.: **Hybrid methods using genetic algorithms for global optimization**. In: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), v. 26, n. 2, pp. 243-258, (1996).

SHI, P., & CUI, Y.: **Dynamic path planning for mobile robot based on genetic algorithm in unknown environment**. In: Chinese control and decision conference, pp. 4325-4329. IEEE, Xuzhou (2010).

SILVA, I. N. D., SPATTI, H. D., & FLAUZINO, R. A.: **Redes Neurais Artificiais, Curso prático. Para engenharia e ciências aplicadas**. São Paulo: Artliber Editora Ltda (2010).

TEIXEIRA, A. M.: **Coordenação Ótima de Múltiplos Robôs de Serviço em Tarefas Persistentes**. Tese de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica – Universidade Federal de Juiz de Fora (PPEE, UFJF). Juiz de Fora – MG (2015).

VARGAS, P. A.: **Sistemas Computacionais Bio-Inspirados: Síntese e Aplicação em Inteligência Computacional e Homeostase Artificial**. Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação – Universidade Estadual de Campinas (FEEC, Unicamp). Campinas – SP (2005).

WU, Z. & PANG, X.: **A new technique for global optimization methods**. In: Green Circuits and Systems (ICGCS), 2010 International Conference on, pp. 398-403. IEEE, (2010).

YANG, XIN-SHE.: **Nature-Inspired Metaheuristic Algorithms**. Second Edition. University of Cambridge, United Kingdom: Luniver press (2010).

YU, J., KARAMAN, S., & RUS, D.: **Persistent monitoring of events with stochastic arrivals at multiple stations**. IEEE Transactions on Robotics, v. 31, n. 3, pp. 521-535, (2015).