

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**OTIMIZAÇÃO DE CONTROLE DE TRÁFEGO EM GRUPO
DE ELEVADORES COM ALGORITMOS BIOINSPIRADOS**

JUAN PABLO DIAGO RODRÍGUEZ

ORIENTADOR: Dr. GUILHERME CARIBÉ DE CARVALHO

COORIENTADOR: Dr. DANIEL MUÑOZ ARBOLEDA

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

PUBLICAÇÃO: ENM.DM - 088A/15

BRASÍLIA/DF: OUTUBRO– 2015

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

OTIMIZAÇÃO DE CONTROLE DE TRÁFEGO EM GRUPO
DE ELEVADORES COM ALGORITMOS BIOINSPIRADOS

JUAN PABLO DIAGO RODRÍGUEZ

DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM SISTEMAS MECATRÔNICOS.

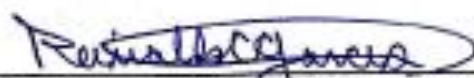
APROVADA POR:



Prof. Dr. Guilherme Caribé de Carvalho (ENM-UnB)
(Orientador)



Prof. Dr. Carlos Humberto Llanos Quintero (ENM-UnB)
(Examinador Interno)



Prof. Dr. Reinaldo Crispiniano Garcia (EPR-UnB)
(Examinador Externo)

BRASÍLIA/DF, 23 DE OUTUBRO DE 2015

FICHA CATALOGRÁFICA

DIAGO R., JUAN PABLO.

Otimização de controle de tráfego em grupo de elevadores com algoritmos bioinspirados [Distrito Federal] 2015.

xvii, 105p, 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2015). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia. Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. Sistemas de Elevadores

2. Enxame de Partículas

3. Simulador *Arena*

4. Sistemas de Controle

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DIAGO. J. P. (2015). Otimização de controle de tráfego em grupo de elevadores com algoritmos bioinspirados. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-088A/15, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 105p.

CESSÃO DE DIREITOS

AUTOR: Juan Pablo Diago Rodríguez.

TÍTULO: Otimização de controle de tráfego em grupo de elevadores com algoritmos bioinspirados.

GRAU: Mestre

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Juan Pablo Diago Rodríguez
SQN 409 Bloco Q Apto 205.
70.857-170 Brasília – DF – Brasil.

DEDICATÓRIA

Dedico este trabalho Aos meus Avós, meus pais e minha tia, que com seu amor e carinho me fortaleceram nos momentos mais difíceis, também por que me ensinaram a perseguir meus sonhos com dedicação e coragem. E minha avó Luz no céu. Obrigado a todos vocês. E a Lizeth Marcela Gómez por ter-me inspirado para fazer este Mestrado, teu amor me deu forças pra chegar até este momento, te amo.

AGRADECIMENTOS

Primeiramente quero agradecer a Deus pela sua proteção, pelas oportunidades e bênçãos recebidas, por ter colocado no meu caminho pessoas maravilhosas que tornaram isto possível.

Aos meus avós Carlos e Margoth, aos meus pais, Hugo Diago e Gladys Rodríguez, e à minha tia Cristina, obrigado pelo amor e apoio em cada passo da minha vida; aos meus irmãos, que sempre me animaram, e pelo seu carinho; à minha família, pelo seu apoio incondicional, e também palavras de incentivo em nos momentos mais difíceis.

À Lizeth Marcela, pelo carinho, por acreditar em mim e pela inspiração neste caminho para obter esta nova conquista, te amo.

Ao professor Guilherme e ao professor Daniel Muñoz, pela orientação e pelo apoio durante a pesquisa.

Ao Rodrigo Muñoz e à Paola Cajas pelo carinho e pelo apoio nos momentos críticos, vocês são minha família aqui. Ao meu amigo Alexander Narvaez e Diana Montilla, pela ajuda incondicional. Aos meus colegas e amigos pelos conselhos e também pelas alegrias e pela companhia nos momentos difíceis.

Aos meus professores do mestrado, ao prof. Carlos pela formação acadêmica, agradeço também a Prof. Suélia pela oportunidade no Lab. e ao professor Edson pela sua total disposição e apoio.

RESUMO

OTIMIZAÇÃO DE CONTROLE DE TRÁFEGO EM GRUPO DE ELEVADORES COM ALGORITMOS BIOINSPIRADOS

Autor: Juan Pablo Diago Rodríguez

Orientador: Dr. Guilherme Caribé De Carvalho

Coorientador: Dr. Daniel Muñoz Arboleda

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, 23 de Outubro de 2015

Este trabalho tem como objetivo apresentar a implementação de uma técnica de otimização bioinspirada como solução ao problema de controle de tráfego em sistemas de grupos de elevadores (EGCS). A técnica de controle usada é o algoritmo de otimização por inteligência de enxame (PSO - *swarm optimization particle*) de tipo binário. A ideia é que o algoritmo escolha o melhor elevador para um usuário que faz uma chamada de serviço em um sistema de controle destino (DCS – *destination control system*). Para a escolha do elevador o algoritmo tem uma função custo que considera as variáveis: (1) tempo de espera; (2) tempo de voo; (3) capacidade do elevador; (4) número de paradas alocadas; entre outras. A solução que apresente o melhor valor de aptidão será o elevador selecionado para atender a chamada. Como caso de estudo o controlador foi testado em um prédio do tipo comercial simulando o caso mais crítico (uma situação de tráfego *up-peak*). Finalmente, os resultados obtidos são apresentados e comparados com os resultados de outros autores.

Palavras chaves: EGCS controle de sistemas de grupos de elevadores, PSO otimização com enxames de partículas, DCS sistemas de controle destino, Função custo.

ABSTRACT

OPTIMIZATION FROM THE TRAFFIC IN ELEVATORS GROUP CONTROL WITH BIOINSPIRED ALGORITHM

Author: Juan Pablo Diago Rodríguez

Supervisor: Dr. Guilherme Caribé De Carvalho

Co-supervisor: Dr. Daniel Muñoz Arboleda

Mechatronic Systems Post-graduation Program

Brasília, October 23th2015

This work aims at presenting the implementation of a Bio-inspired optimization technique applied to Elevator Group Control Systems (EGCS) as a solution for traffic problem. The control technique used is a binary optimization algorithm based on swarm intelligence (PSO - particle swarm optimization). The main idea consist of choosing the best elevator for a user who calls for service in a destination control system (DCS). In order to choosing the elevator, the algorithm uses a cost function that considers the variables: (1) waiting time; (2) flight time; (3) elevator capacity; (4) number of allocated stops; among others. The solution that presents the best fitness value corresponds to the selected elevator that is going to answer the call. As case of study, the controller was tested in a commercial building, simulating the most critical case (up-peak traffic situation). Finally, the results obtained are summarized and compared with results obtained by other authors.

Keywords: EGCS elevator group control system, PSO Particle Swarm Optimization, DCS destination control system, Function cost.

SUMÁRIO

RESUMO.....	vi
ABSTRACT	vii
1. INTRODUÇÃO	1
1.1. DESCRIÇÃO DO PROBLEMA	2
1.2. JUSTIFICATIVA	3
1.3. OBJETIVOS.....	4
1.3.1. Objetivo Geral.....	4
1.3.2. Objetivos Específicos	4
1.4. CONTRIBUIÇÕES DO TRABALHO.....	5
1.5. ASPECTOS METODOLÓGICOS	5
1.6. ORGANIZAÇÃO DO TRABALHO.....	6
2. FUNDAMENTAÇÃO TEÓRICA	7
2.1. SISTEMAS DE ELEVADORES	7
2.1.1. Restrições dadas pelos usuários dos elevadores	7
2.1.2. Quantidade de elevadores ótima	10
2.1.3. Tipos de prédios	12
2.1.4. Padrões de tráfego	13
2.1.5. Outros tipos de padrões de tráfego	16
2.2. SOFTWARE ESCOLHIDO PARA A SIMULAÇÃO DOS SISTEMAS DE GRUPOS DE ELEVADORES: <i>ARENA</i> ®	16
2.3. CONTROLE DE TRÁFEGO EM SISTEMAS DE GRUPOS DE ELEVADORES.....	21
2.3.1. Sistemas de Controle Destino DCS.....	21
2.3.2. Controle de tráfego	23
3. ALGORITMOS DE OTIMIZAÇÃO BASEADOS EM INTELIGÊNCIA DE ENXAMES	29
3.1. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO)	30
3.1.1. Algoritmo PSO básico	30

3.2.	ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE ABELHAS (ABC)...	33
3.2.1.	Algoritmo ABC	35
3.3.	OTIMIZACAO POR VAGA LUMES (FA)	38
3.4.	OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS PSO BINÁRIO	40
3.5.	TÉCNICAS DE MELHORIA DE DESEMPENHO DOS ALGORITMOS BIOINSPIRADOS	42
3.5.1.	O método atrativo repulsivo	42
3.5.2.	O método de congregação passiva seletiva	42
3.5.3.	O método de aprendizado em oposição.....	43
4.	TESTES DE ESCALABILIDADE PARA A SELEÇÃO DO MELHOR ALGORITMO BIOINSPIRADO	44
4.1.	FUNÇÕES DE TESTE <i>BENCHMARK</i>	44
4.1.1.	Função esfera.....	44
4.1.2.	Função <i>Rosenbrock</i>	45
4.1.3.	Função <i>Rastrigin</i>	46
4.2.	RESULTADOS DOS TESTES DOS ALGORITMOS DE OTIMIZAÇÃO USANDO AS FUNÇÕES BENCHMARK	47
4.2.1.	Teste dos algoritmos PSO, ABC e FA na função esfera.....	48
4.2.2.	Teste de PSO, ABC e FA para a função <i>Rosenbrock</i>	50
4.2.3.	Teste de PSO, ABC e FA na função <i>Rastrigin</i>	53
5.	IMPLEMENTAÇÃO DO ALGORITMO PSO BINÁRIO PARA O CONTROLE DE SISTEMAS DE GRUPOS DE ELEVADORES	56
5.1.	CARACTERÍSTICAS DO PRÉDIO DE ESTUDO	60
5.2.	GERADOR DE CHAMADAS.....	61
5.3.	FUNÇÃO CUSTO PARA AVALIAÇÃO DO PSO BINÁRIO	64
5.4.	RESULTADOS EXPERIMENTAIS DO ALGORITMO DE CONTROLE COM O PSO DE TIPO BINARIO.....	75
5.5.	ANALISES DE RESULTADOS E COMPARAÇÃO COM OUTROS AUTORES.....	81
5.5.1	– Comparação com os resultados de Patiño [10]	82
5.5.2	– Comparação com os resultados de Siikonen [42]	84

6. DISCUSSÃO DOS RESULTADOS	88
7. CONCLUSÕES E TRABALHOS FUTUROS	91
7.1. CONCLUSÕES	91
7.2. SUGESTÕES PARA TRABALHOS FUTUROS	92
8. REFERENCIAS BIBLIOGRÁFICAS	93
APÉNDICE A	98
APÉNDICE B	101

LISTA DE TABELAS

TABELA 1. PADRÕES DOS TEMPOS DE ESPERA POR TIPO DE PRÉDIO [2].	9
TABELA 2. TEMPOS DE VOO PADRÕES POR TIPO DE PRÉDIO [10]	9
TABELA 3. PORCENTAGENS DE PADRÕES DE TRÁFEGO [13].	16
TABELA 4. BASE DE REGRAS DE UM CONTROLE NEBULOSO [10].	26
TABELA 5. PARÂMETROS DE CONFIGURAÇÃO DOS ALGORITMOS	47
TABELA 6. RESULTADOS DO ALGORITMO PSO NA FUNÇÃO ESFERA	48
TABELA 7. RESULTADOS DO ALGORITMO ABC NA FUNÇÃO ESFERA	48
TABELA 8 RESULTADOS DO ALGORITMO FA NA FUNÇÃO ESFERA	49
TABELA 9. ALGORITMO PSO COM A FUNÇÃO <i>ROSENBROCK</i>	50
TABELA 10. ALGORITMO ABC NA FUNÇÃO <i>ROSENBROCK</i>	51
TABELA 11. ALGORITMO FA NA FUNÇÃO <i>ROSENBROCK</i>	51
TABELA 12. ALGORITMO PSO NA FUNÇÃO <i>RASTRIGIN</i> .	53
TABELA 13 ALGORITMO ABC NA FUNÇÃO <i>RASTRIGIN</i>	53
TABELA 14 ALGORITMO FA COM A FUNÇÃO <i>RASTRIGIN</i> .	54
TABELA 15. CARACTERÍSTICAS DO PRÉDIO COMERCIAL CASO DE ESTUDO.	61
TABELA 16 CHAMADAS GERADAS DO TIPO <i>UP-PEAK</i>	64
TABELA 17. PARÂMETROS DO PSO BINÁRIO	72
TABELA 18. RESULTADOS TEMPOS COM DIFERENTES PORCENTAGENS DE POPULAÇÃO.	77
TABELA 19 COMPARAÇÃO DE RESULTADOS COM OUTROS AUTORES	85
TABELA 20 TESTES DE SELEÇÃO DE ELEVADOR PELO BPSO	98

LISTA DE FIGURAS

FIGURA 1. A) PESSOAS ESPERANDO POR UM ELEVADOR, B) GRUPO DE ELEVADORES EM VOO..	3
FIGURA 2. A) METODOLOGIA <i>TOP DOWN</i> , B) METODOLOGIA <i>BOTTOM UP</i> .	6
FIGURA 3. CURVA DE DISTÂNCIA, VELOCIDADE E ACELERAÇÃO [2].	8
FIGURA 4. A) EXEMPLO DE CAPACIDADE ULTRAPASSADA, B) ESPAÇO PADRÃO POR PESSOA [2].	9
FIGURA 5. PADRÕES DE TRÁFEGO EM UM PRÉDIO COMERCIAL [2].	13
FIGURA 6. INTENSIDADE DE TRÁFEGO <i>UP-PEAK</i> [2].	14
FIGURA 7. INTENSIDADE DE TRÁFEGO <i>DOWN-PEAK</i> [2].	15
FIGURA 8. CHAMADAS DE ACORDO COM OS PADRÕES DE TRÁFEGO	15
FIGURA 9. PRÉDIO COM DOIS MOMENTOS DE TRÁFEGO <i>UP-PEAK</i> .	16
FIGURA 10. SOFTWARE PROPRIETÁRIO <i>ARENA</i> (<i>ROCKWELL AUTOMATION</i>).	17
FIGURA 11. SIMULAÇÃO DE PROCESSOS COM <i>ARENA</i> . [14]	18
FIGURA 12. ENTIDADES, FILAS, PROCESSOS, E OUTROS. [14]	19
FIGURA 13 REPORTES DEPOIS DA SIMULAÇÃO [14]	20
FIGURA 14. <i>ARENA</i> E CONEXÃO COM OUTROS APLICATIVOS. [16]	20
FIGURA 15. MÉTODO CLÁSSICO DE CHAMADA DE SERVIÇO [10].	22
FIGURA 16. DCS: SISTEMAS DE CONTROLE DE DESTINO.	22
FIGURA 17. VARIÁVEIS COM GRAUS DE INCERTEZA	25
FIGURA 18. VARIÁVEIS EM CONJUNTOS NO <i>MATLAB</i> [10].	25
FIGURA 19. INTERAÇÃO DE VARIÁVEIS DE ENTRADA E A SAÍDA DE UM CONTROLE NEBULOSO.	26
FIGURA 20. NEURÔNIO ARTIFICIAL	27
FIGURA 21. REDE NEURAL ARTIFICIAL	27
FIGURA 22. MOVIMENTOS DAS PARTÍCULAS	31
FIGURA 23. SOMA VETORIAL PARA DEFINIR A NOVA POSIÇÃO DA PARTÍCULA.	32
FIGURA 24. PSEUDOCÓDIGO DO PSO [7].	33
FIGURA 25 EXEMPLO DE FONTES DE ALIMENTO E REPRESENTAÇÃO PELAS ABELHAS	35
FIGURA 26 PSEUDOCÓDIGO DO ALGORITMO ABC [25]	37
FIGURA 27. PSEUDOCÓDIGO DO ALGORITMO <i>FA</i> BÁSICO [28]	39
FIGURA 28. PSEUDOCÓDIGO DO ALGORITMO PSO BINÁRIO BÁSICO	41

FIGURA 29. FUNÇÃO ESFERA	45
FIGURA 30. FUNÇÃO <i>ROSENBROCK</i>	45
FIGURA 31. FUNÇÃO <i>RASTRIGIN</i>	46
FIGURA 32. GRÁFICO DE CONTORNO DA FUNÇÃO <i>RASTRIGIN</i>	46
FIGURA 33. EXEMPLOS DE ELEVADORES SELECIONADOS PARA ATENDER UMA CHAMADA.	56
FIGURA 34. ESQUEMA DO SIMULADOR.	57
FIGURA 35. ESQUEMA DE CONTROLE DO SIMULADOR.	58
FIGURA 36. ESQUEMA DO PSO BINÁRIO	59
FIGURA 37. PRÉDIO COMERCIAL DO ESTUDO DE CASO.	61
FIGURA 38. GERADOR DE CHAMADAS NO <i>ARENA</i>	62
FIGURA 39. GERAÇÃO DE CHAMADAS NÃO <i>UP-PEAK</i>	63
FIGURA 40. GRÁFICO DE CHAMADAS GERADAS PELO SIMULADOR (<i>UP-PEAK</i>) EM FUNÇÃO DO TEMPO DE SIMULAÇÃO.	63
FIGURA 41. VETORES DE ESTADO DOS ELEVADORES.	66
FIGURA 42. A) CENÁRIO DO EXEMPLO 1, B) CENÁRIO DO EXEMPLO 2	69
FIGURA 43. A) CENÁRIO DO EXEMPLO 3, B) CENÁRIO DO EXEMPLO 4	70
FIGURA 44. A) CENÁRIO DO EXEMPLO 5, B) CENÁRIO DO EXEMPLO 6.	71
FIGURA 45. BLOCO VBA DO BPSO.	73
FIGURA 46. PSEUDOCÓDIGO DO BPSO EM VBA.	73
FIGURA 47. SIMULAÇÃO DO MOVIMENTO DOS ELEVADORES.	74
FIGURA 48. SIMULAÇÃO DE USUÁRIOS NO ELEVADOR 1.	74
FIGURA 49. TEMPOS MÉDIOS DE ESPERA PELO ELEVADOR ESCOLHIDO.	78
FIGURA 50. TEMPO MÉDIO DE VOO.	79
FIGURA 51. TEMPOS MÉDIOS DO SISTEMA DE ELEVADORES	80
FIGURA 52. NÚMERO DE PARADAS MÉDIO DOS ELEVADORES.	81
FIGURA 53. COMPARAÇÃO DOS TEMPOS MÉDIOS DE ESPERA COM OS RESULTADOS DE PATINO [10],	82
FIGURA 54 COMPARAÇÃO DE TEMPOS DE DESTINO COM PATINO [10].	83
FIGURA 55. COMPARAÇÃO DE NUMERO DE PARADAS MÉDIO COM PATINO [10].	84
FIGURA 56. COMPARAÇÃO COM O TRABALHO	85
FIGURA 57. TEMPOS MÉDIOS DE ESPERA	87
FIGURA 58. TEMPOS MÉDIOS DE VOO	87

LISTA DE SIMBOLOS, NOMENCLATURAS E ABREVIACÕES.

DCS	- <i>Destination Control System</i>
<i>Down-peak</i>	-Tráfego de descida.
<i>Interfloor</i>	-Tráfego distribuído nos andares.
<i>Up-peak</i>	-Tráfego de subida
EGCS	-Sistema de Controle de Grupo de Elevadores
RTT	-Tempo de viagem de ida e volta.
L	-Número de elevadores
cc	-Capacidade do elevador
HC	-Capacidade de transporte
I	-Intervalo de tempo
NC	- <i>Control Near Car</i>
FSO	- <i>Fixed Sectoring Common Sector System</i>
FS4	- <i>Fixed Sectoring Priority Sector System</i>
AG	-Algoritmos Genéticos
AE	-Algoritmos Evolutivos
PE	-Programação Evolutiva
PG	-Programação Genética
PSO	- <i>Particle Swarm Optimization</i>
ABC	- <i>Artificial Bee Colony</i>
FA	- <i>Firefly Algorithm</i>
BPSO	- <i>Binary Particle Swarm Optimization</i>

1. INTRODUÇÃO

Desde o século passado tem havido um acelerado aumento da propriedade vertical (prédios de vários andares) como resultado da intenção do homem em otimizar o espaço nas cidades, além disso, devido a fatores econômicos e comerciais, tem se procurado fazer com que prédios governamentais, de negócios, de atendimento à saúde e residências fiquem próximos uns dos outros, e em uma razão midiática pela necessidade do homem de tentar preservar a natureza evitando desmatar as zonas verdes na expansão das urbes. O incremento destas propriedades gera o problema de transporte vertical [1], que consiste na necessidade de ter um meio adequado e eficaz que facilite o transporte nesse tipo de prédios.

Desse modo, essa necessidade de transporte, que a princípio foi solucionada pelo uso das escadas (incluindo as elétricas da atualidade), persistiu, na medida que essa alternativa foi se tornando rapidamente inadequada como solução quando os prédios apresentavam mais que cinco andares e dispunham de grande quantidade de pessoas se movimentando na edificação. As principais desvantagens do transporte vertical baseado em escadas são: tempos de deslocamento entre andares elevados, desconforto dos usuários, fadiga física das pessoas, probabilidade de acidentes, lesões físicas nas pernas, além de outros fatores próprios de cada pessoa tais como idade, uso de cadeira de rodas e limitações físicas. Entretanto, esse tipo de solução é importante e fundamental por ser o meio de transporte recomendado em situações de emergência, e em muitos casos seu uso é adequado para prédios de poucos andares (até quatro andares) [2].

Atualmente a solução mais eficiente para o transporte vertical se baseia no uso do elevador, entendendo que ao se aumentar o tamanho do prédio (andares, capacidade, população) pode ser necessário o uso de sistemas de grupos de elevadores (três ou mais elevadores em um prédio) [2] [3]. Esses sistemas envolvem diferentes tipos de tecnologias no campo da instrumentação, controle e comunicação para a supervisão e intercâmbio de informações referentes ao funcionamento dos elevadores. Tal conjunto de tecnologias pertence a área denominada de *automação predial* [2].

Uma vez que as soluções para o problema de transporte vertical ainda não estão definidas, surge uma oportunidade interessante de pesquisa neste campo, pensando em uma possível solução com o uso de novas técnicas de controle e automação. Dessa forma, este trabalho apresenta os resultados obtidos com as implementadas com algoritmos de otimização bioinspirados como possível solução para fazer o controle de sistemas de grupos de elevadores (EGCSs – *Elevator Group Control Systems*) [4].

1.1. DESCRIÇÃO DO PROBLEMA

Desde o ponto de vista computacional, o problema do transporte vertical com sistemas de grupos de elevadores pode ser classificado como NP - difícil (problemas com alto grau de complexidade computacional) e, portanto, o seu controle representa um desafio para as técnicas de controle modernas.

Em síntese o problema abordado neste trabalho é o controle de tráfego em sistemas de grupos de elevadores, que consiste no uso ótimo dos elevadores de forma a diminuir o consumo de energia e o tempo de espera, e aumentar o conforto dos usuários. O tempo de espera pode ser dividido em duas partes: (1) tempo de espera nos pavimentos (tempo desde a solicitação da chamada até um elevador abrir as portas no andar de origem); e (2) tempo de voo (tempo que demora o elevador para levar o usuário do andar da chamada de origem até o andar de destino). Outros parâmetros relevantes nos EGCSs são: a intensidade do fluxo de passageiros e a capacidade dos elevadores [5]. Logo, a importância de otimizar o funcionamento de cada elevador é evidente para garantir que o serviço seja prestado dentro de parâmetros adequados de tempo e conforto [2]. Na Figura 1, apresenta-se um exemplo de grupos de elevadores em prédios comerciais.

Neste trabalho, faz-se a implementação de algoritmos de otimização bioinspirados como solução para o problema descrito, visando otimizar o tempo de espera dos usuários através do uso eficiente dos elevadores. É importante ressaltar que a otimização simultânea do consumo de energia e do tempo de espera constitui um problema de otimização multi-objetivo que não será considerado neste trabalho [6].

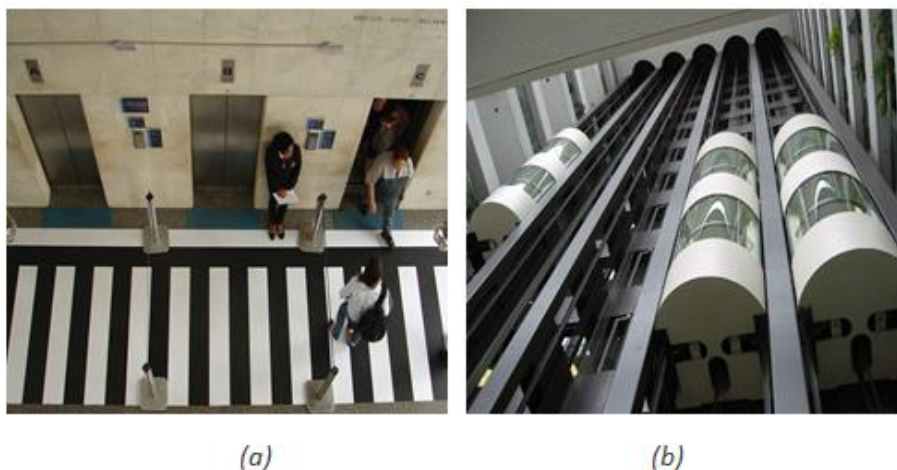


Figura 1. a) pessoas esperando por um elevador, b) grupo de elevadores em voo.

1.2. JUSTIFICATIVA

A proposta deste trabalho esta inserida dentro do conjunto de algoritmos de otimização bioinspirados para resolução de problemas de tipo discreto. Essa escolha se justifica pela natureza do problema dos EGCSs nos quais, de forma geral, a função custo é não linear e de tipo discreta (o número de elevadores do sistema é uma variável discreta), dependendo de parâmetros dinâmicos conforme o fluxo de passageiros. Algoritmos bioinspirados são uma solução viável para resolver problemas de otimização de alta complexidade computacional, como o descrito anteriormente. Contudo, existem poucos trabalhos na literatura científica que abordam o tema de controle de grupos de elevadores desde uma perspectiva bioinspirada.

Neste trabalho será usado o algoritmo de otimização por enxame de partículas (PSO) para solucionar o problema do controle de um grupo de elevadores. O PSO é uma técnica de otimização baseada na inteligência dos enxames, inspirada no comportamento social de cardumes de peixes e bandos de aves na procura por alimento [7]. Essa escolha se justifica dada a facilidade de entendimento e implementação do PSO, assim como pela sua capacidade para resolver problemas de otimização multimodais se comparado com outros algoritmos bioinspirados [7] [8]. Adicionalmente, há na literatura científica trabalhos prévios onde o PSO tem sido utilizado para resolver problemas de otimização discretos e binários [9].

Este trabalho faz parte da linha de pesquisa de “controle e automação” do programa de pós-graduação em engenharia mecatrônica, com foco na automação predial. É importante levar em consideração que, nesta área, há um trabalho prévio no qual o controle do grupo de elevadores foi implementado usando a lógica nebulosa [10]. Dentre as sugestões para trabalhos futuros dessa pesquisa, foi citado o uso de técnicas de otimização para a implementação do controle de sistemas de grupos de elevadores.

1.3. OBJETIVOS

1.3.1. Objetivo Geral

O principal objetivo deste trabalho é a implementação da técnica de otimização bioinspirada PSO (*Particle Swarm Optimization*) de tipo binária para o controle de tráfego de um sistema de grupos de elevadores, possibilitando minimizar o tempo de espera dos usuários.

1.3.2. Objetivos Específicos

Os objetivos específicos do trabalho são:

- Estudar as características do problema de controle de tráfego em sistemas de grupos de elevadores, seus parâmetros e limitações.
- Análise e teste de algoritmos de otimização bioinspirados para resolução de problemas tipo *benchmark*.
- Levantamento do estado da arte da aplicação de algoritmos bioinspirados de tipo binário para o controle de grupo de elevadores.
- Desenvolver uma função de custo baseada no tempo de espera que permita avaliar qual é o melhor elevador para atender a uma chamada.
- Apresentar a modelagem de um simulador de tráfego de grupos de elevadores baseado no uso do algoritmo PSO binário, usando tecnologias de automação industrial (*Arena, da Rockwell Automation*).
- Validar os diferentes resultados do sistema de controle projetado para a situação de tráfego mais crítica, permitindo estudar e comparar os resultados obtidos nas simulações.

1.4. CONTRIBUIÇÕES DO TRABALHO

No desenrolar do trabalho foi realizado um artigo para ser publicado no “*Congreso Interamericano de Computación Aplicada a la Industria De Procesos, CAIP 2015*” sobre a concepção da função custo que representa o problema dos sistemas de grupos de elevadores e o modelo de controle com o algoritmo PSO do tipo binário. Também se trabalha em outro artigo sobre a modelagem do sistema de controle dos sistemas de elevadores com o algoritmo de otimização *PSO do tipo binário* testado no tráfego *Up-peak* e implementado no software *Arena*, esse último, ainda não foi submetido.

1.5. ASPECTOS METODOLÓGICOS

Este trabalho de pesquisa foi desenvolvido em duas etapas: uma teórica e outra experimental. Na primeira etapa o trabalho de pesquisa se foca no marco teórico do tema, permitindo realizar um estudo da literatura existente sobre o problema do controle de elevadores e as técnicas usadas como possíveis soluções para ele. Na fase experimental, foi proposta uma abordagem do problema baseada inicialmente em: a) geração de chamadas dos usuários; b) construção da função de custo que representa o sistema de elevadores, a qual foi testada e refinada; c) funcionamento do controlador com o algoritmo PSO binário. Logo, foram feitos testes em um simulador desenvolvido no software *Arena*, obtendo-se assim os resultados que são usados para a conclusão da pesquisa.

Nas duas etapas foram empregadas as metodologias *top down* e *bottom up*, amplamente usadas no desenvolvimento na área de *software* e simulação, as quais foram escolhidas por sua capacidade de abordar e facilitar a compreensão do problema, assim como para a implementação da solução final. A metodologia *top down* foi usada na análise do problema dos sistemas dos elevadores, começando pelo geral e avançando pouco a pouco para os aspectos específicos, permitindo dividir de forma estruturada o conhecimento desse tema. Essa metodologia é usada basicamente na construção da fundamentação teórica [11], e como resultado da sua aplicação foi obtido um modelo para o desenvolvimento do simulador. A metodologia *bottom up*, foi utilizada de forma modular para a construção do simulador do sistema de controle de grupos de elevadores (simulador de chamadas, função custo, análises de cenários, controlador desenvolvido usando o PSO binário). Os módulos

desenvolvidos foram integrados para a realização dos testes que permitirão a obtenção dos resultados esperados da pesquisa. Na Figura 2 podem-se observar as características mais relevantes dessas duas metodologias. [11]

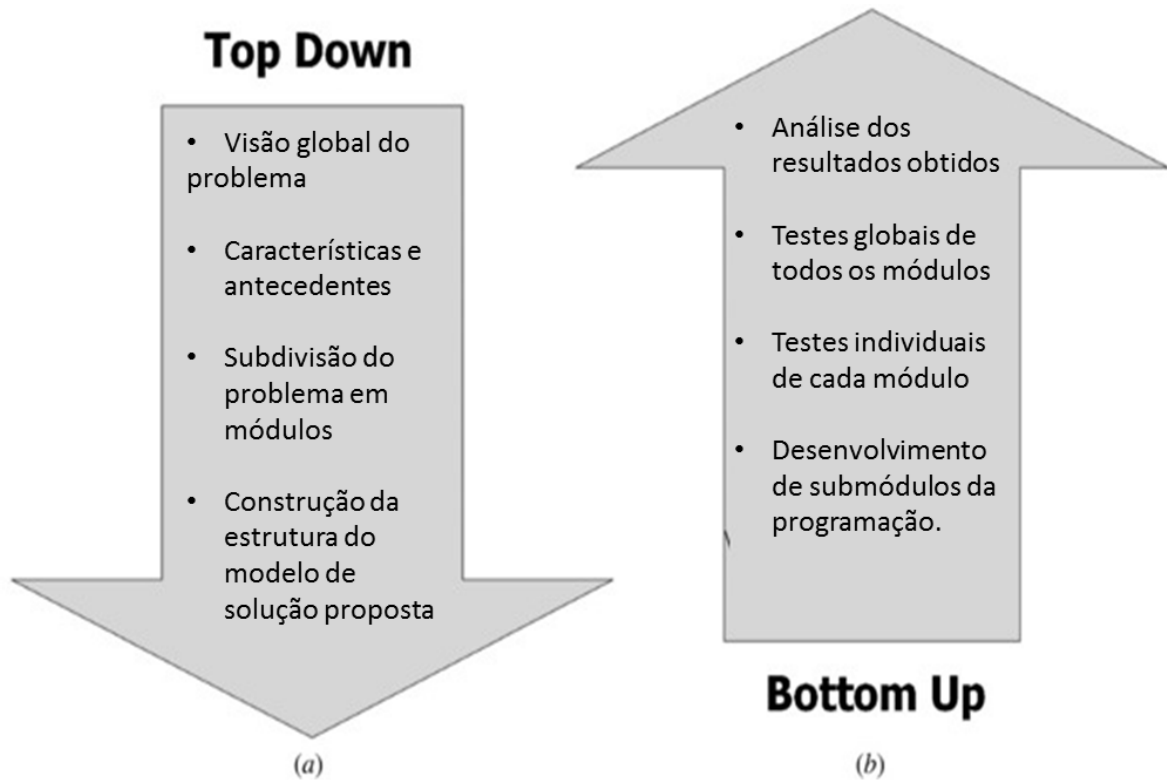


Figura 2. a) metodologia *top down*, b) metodologia *bottom up*.

1.6. ORGANIZAÇÃO DO TRABALHO

No Capítulo 2 apresenta a fundamentação teórica necessária para o desenvolvimento deste trabalho, abordando assuntos tais como: o funcionamento e as características do sistema de elevadores e dos algoritmos de otimização bioinspirados. No Capítulo 3, descrevem-se os algoritmos de otimização bioinspirados estudados nesta pesquisa. O Capítulo 4 apresenta a implementação do algoritmo PSO binário para o controle dos sistemas de grupos de elevadores e os resultados obtidos. O Capítulo 5 descreve a discussão dos resultados obtidos e uma proposta de conclusão do projeto de mestrado. Finalmente, no Capítulo 6 apresentam-se as conclusões do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica dos temas relacionados com o objetivo desta pesquisa. Dessa forma, introduzem-se os sistemas de grupos de elevadores, definindo as características mais relevantes, dando ênfase especial aos sistemas de elevadores de prédios comerciais (limitações, tempos de espera, padrões de tráfego, e etc.). Em seguida são feitas considerações relevantes descrevendo algumas das técnicas de controle de sistemas de elevadores apresentadas na literatura científica.

2.1. SISTEMAS DE ELEVADORES

Considera-se um sistema de grupo de elevadores quando em um prédio qualquer, há três ou mais elevadores como principal meio de transporte. As características desses sistemas e o seu funcionamento estão relacionados ao tipo de prédio onde são usados (residencial, comercial, hospital, *shopping centers*, entre outros). Cada tipo de prédio tem características e necessidades de transporte diferentes. Por exemplo, suponha-se um prédio do tipo residencial e outro do tipo comercial. No primeiro caso, a demanda do serviço dos elevadores pode ter uma intensidade menor que no segundo, dado que a demanda de transporte no prédio comercial é elevada, por causa da necessidade de uma quantidade maior de pessoas a serem transportadas.

Desse modo, a importância de definir as características dos prédios se foca em incrementar as possibilidades de criar um sistema de controle adequado para cumprir com as necessidades de transporte (tempo de espera adequado), elevando as possibilidades de gerar conforto para os usuários desses sistemas [2].

2.1.1. Restrições dadas pelos usuários dos elevadores

Os usuários dos elevadores fornecem restrições relacionadas à sensação de conforto e à satisfação do serviço recebido. Assim, podem-se definir dois grupos de restrições: (1) restrições de tipo fisiológicas e (2) restrições de tipo psicológicas. As primeiras são determinadas, na maioria dos casos, pelo desconforto que as pessoas sentem em seu corpo quando o elevador varia sua velocidade (aceleração e desaceleração). A possibilidade de

diminuir essa sensação de desconforto reside em limitar a variação da velocidade. Entretanto, a sua eliminação é um tema complexo, já que esta sensação depende de cada usuário e da sua tolerância diante desse fenômeno. Na Figura 3 a) pode-se observar a distância h percorrida pelo elevador em um tempo T , e na Figura 3 b) observam-se as curvas de aceleração máxima sugeridas na literatura para prédios com grupos de elevadores. É importante destacar que essas curvas não são aplicadas para elevadores de alta velocidade como os encontrados em arranha-céus.

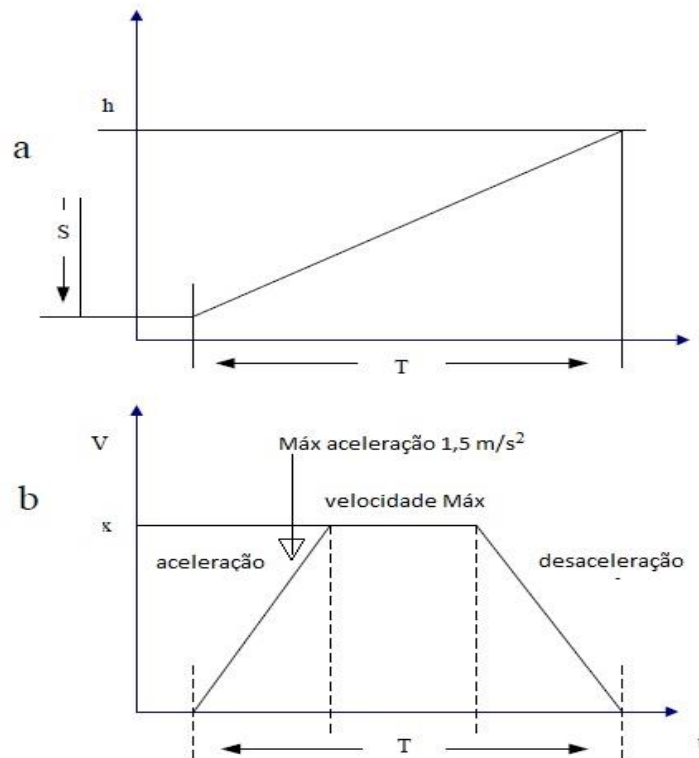


Figura 3. Curva de distância, velocidade e aceleração [2].

Outra restrição do tipo fisiológica que pode gerar algum desconforto está relacionada com o espaço que cada usuário dispõe dentro do elevador, o qual pode ser definido pela distância mínima que deve haver entre um usuário e outro sem que estes sintam algum tipo de incômodo. Na Figura 4 a) apresenta-se a situação onde o espaço pessoal necessário é ultrapassado, e na Figura 4 b) apresenta-se o espaço pessoal padrão sugerido na literatura [2]. Esse tipo de desconforto também depende de cada pessoa e da sua tolerância diante dessa situação.

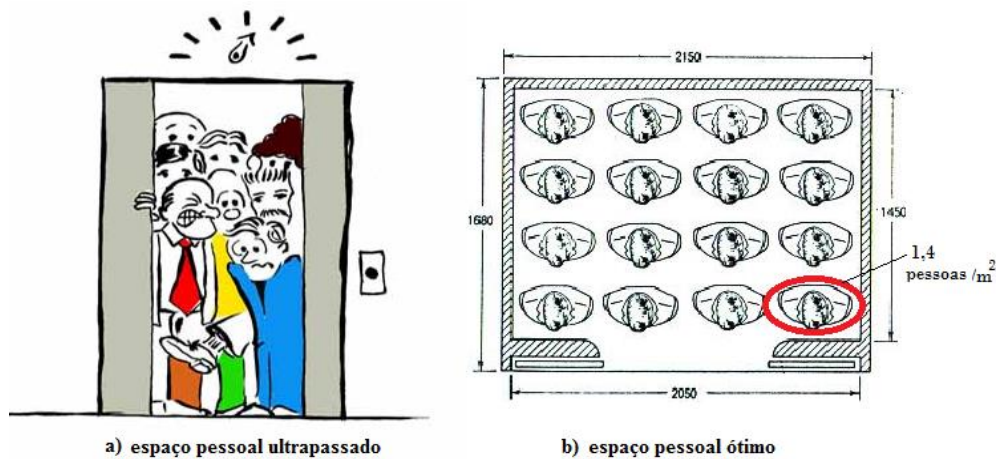


Figura 4. a) Exemplo de capacidade ultrapassada, b) Espaço padrão por pessoa [2].

Por outro lado, as restrições do tipo psicológico são determinadas pelo tempo de espera, isto é, pelo tempo que o usuário espera por um elevador e o tempo que demora a viagem do andar de origem até o andar de destino. Na Tabela 1 são apresentados os valores máximos de tempo de espera em diferentes tipos de prédios, de forma a evitar o desconforto dos usuários. Por exemplo, no caso de um prédio de escritórios convencional, o tempo de espera máximo é de aproximadamente 30 segundos, diferente do tempo no prédio residencial, que pode ser de até 60 segundos. Na Tabela 2, apresentam-se os valores típicos do tempo de voo em edificações residenciais e comerciais.

Tabela 1. Padrões dos tempos de espera por tipo de prédio [2].

Tipo de prédio	Intervalo de tempo (s)
Hotel	30-50
Residencial	40-90
Hospitais	30-50
Escolas	30-50
Escritórios	
Normal	25-30
Prestígiado	20-25

Tabela 2. Tempos de voo padrões por tipo de prédio [10]

Tipo de prédio	Tempo de Voo (s)
Hotel	<100s
Residencial	<110s
Comercial	<120s

2.1.2. Quantidade de elevadores ótima

Para determinar o número de elevadores e sua capacidade é preciso conhecer as características do prédio (tipo de prédio, número de andares, população, intensidade da demanda, etc.). A importância desses valores reside no fato de que permitirão especificar um sistema de elevadores adequado que tenha a capacidade de servir de forma ótima aos usuários, evitando sobredimensionamentos (prédios com mais elevadores do que é necessário) ou subdimensionamentos (prédios com elevadores insuficientes para cumprir com a demanda de serviço). O cálculo do número de elevadores e da sua capacidade não é simples, precisando-se, em muitas ocasiões, da ajuda de especialistas nessa área. Contudo, quando se constrói um prédio, na maioria dos casos, não se faz esse tipo de análise, possibilitando a existência de sistemas de transporte mal projetados e com pouca capacidade de prestar um serviço satisfatório [3].

Da mesma forma, projetar um sistema de elevadores para prédios já construídos (reais) é mais difícil, devido aos seguintes fatores: custo econômico, espaço disponível, modificações do prédio, interferência na mobilidade e adequações de instalações tais como sistemas elétricos, sistemas de potência, cabine de manutenção, entre outros. Tais fatores dificultam a instalação, em muitos casos, do número adequado de elevadores [2].

As variáveis como o número de elevadores e capacidade do sistema de grupo de elevadores devem-se projetar para atender a baixas e altas demandas de serviço por parte dos usuários, buscando oferecer uma qualidade de serviço aceitável. Para isso, é importante conhecer o planejamento desses sistemas, compreendidos em dois critérios fundamentais: (1) Capacidade de Transporte (HC); e (2) Intervalos de Tempo (I) [3]. HC é um critério que ajuda a definir o tamanho dos elevadores. Por outro lado, o intervalo de tempo (I) permite calcular o número de elevadores em um sistema de grupos de elevadores.

Para o cálculo de HC e I é necessário usar a equação de RTT (tempo de viagem de ida e volta) que depende do número de paradas durante um percurso de ida e volta desde o andar térreo até o último andar do prédio, considerando uma demanda alta de passageiros (momento do dia com um fluxo maior de pessoas), tal como mostra a Equação (2-1). Observa-se que RTT depende de dois conjuntos de dados em relação ao prédio: os dados do sistema de elevadores e os dados dos usuários [2].

$$RTT = \sum \text{moving times} + \sum \text{stop times}$$

$$RTT = \underbrace{2Ht_v + (S + 1)t_s}_{\text{Dados do prédio e do sistema de elevadores}} + \underbrace{2Pt_p}_{\text{Dados dos usuários}} \quad (2-1)$$

H remete ao número de andares do prédio. O valor de H pode ser calculado usando teorias probabilísticas, as quais dependem diretamente do número de andares do prédio e do número de passageiros. S é o número de paradas esperadas no prédio, e o seu cálculo se baseia em leis probabilísticas. Os valores de S e H podem ser encontrados em tabelas oferecidas pelos fabricantes de elevadores em conformidade com a normatividade de segurança e de instalação de elevadores BS EN81 e BS ISO4190 [2]. P é o número médio de passageiros dentro do elevador (geralmente trabalha-se com 80% da capacidade total de carga do elevador). A variável t_v é o tempo utilizado para percorrer um andar com velocidade nominal V (velocidade máxima do elevador); t_s é o tempo que um elevador fica parado, incluindo o tempo de abrir e fechar as portas, mais o tempo de aceleração e desaceleração; e t_p é o tempo que um usuário leva para entrar ou sair do elevador.

O primeiro critério é a capacidade de transporte, que representa a quantidade de pessoas ou usuários que o elevador pode transportar em um tempo de 5 minutos com uma demanda alta de serviço (HC é comumente conhecido como um indicador da qualidade do serviço). A equação (2-2) mostra como calcular HC , onde cc é a capacidade de carga do elevador (número de pessoas possíveis no elevador) e L é o número de elevadores [2] [3].

$$HC = \frac{(0,8ccL)}{RTT} \quad (2-2)$$

O segundo critério importante para o planejamento de um sistema de elevadores é o tempo médio entre as chegadas sucessivas de um elevador ao andar principal com chamadas agendadas em qualquer andar [2]. Esse intervalo de tempo, aqui representado por I , é calculado por meio da equação (2-3).

$$I = \frac{RTT}{L} \quad (2-3)$$

Este parâmetro representa a capacidade do sistema de estar continuamente prestando serviço aos usuários que chegam ao prédio e que fazem chamadas, especialmente em condições críticas ou de alta demanda. Intervalos de tempo muito grandes poderiam indicar que o sistema não é adequado para o prédio. Por sua vez, intervalos de tempo muito curtos indicam que o sistema está sobredimensionado para o prédio.

2.1.3. Tipos de prédios

Dependendo do tipo de prédio há considerações relevantes que caracterizam e determinam os parâmetros mais importantes para projetar um EGCS [12]. No entanto, é preciso primeiramente definir claramente os principais tipos de prédios.

- **Prédio residencial**

Prédios onde vivem pessoas com características e necessidades de transporte não exigentes e padrões de tráfego de pouca intensidade. Nesse tipo de prédios, podem-se ter tempos de espera e de voo elevados, como mostra a Tabela 1 e a Tabela 2. Além disso, a demanda de serviço nesses prédios não é alta, sendo na maior parte do dia quase constante.

- **Hospitais**

Nos hospitais, o sistema de elevadores requer um planejamento detalhado, considerando aspectos como a arquitetura, necessidades especiais do edifício e necessidade de transportar camas, pacientes, equipamentos, visitantes e trabalhadores. Destacam-se fatores de restrições para transporte de pessoas a andares restritos como: salas de operações, unidades de cuidados intensivos, entre outras áreas.

- **Comerciais**

Correspondem a prédios de escritórios governamentais ou de negócios, onde as necessidades de transporte são exigentes dependendo da hora do dia, uma vez que a demanda dos usuários pelo serviço dos elevadores é elevada em certas horas conhecidas como *peak*. Durante esses períodos de tempo é indispensável o ótimo funcionamento e controle do despacho dos elevadores.

2.1.4. Padrões de tráfego

A demanda gerada pelos usuários dos elevadores representa um desafio para os seus sistemas de controle. Estes controladores têm como função responder eficientemente a cada chamada de serviço e, portanto, pode-se dizer que a demanda não é constante no decorrer do dia. Ditas variações na demanda de serviço determinam os tipos de padrões de tráfego, os quais estão relacionados com o tipo de prédio e com a dinâmica do transporte dos usuários. Para o caso do estudo deste trabalho foi escolhido um prédio comercial com os padrões de tráfego mostrados na Figura 5.

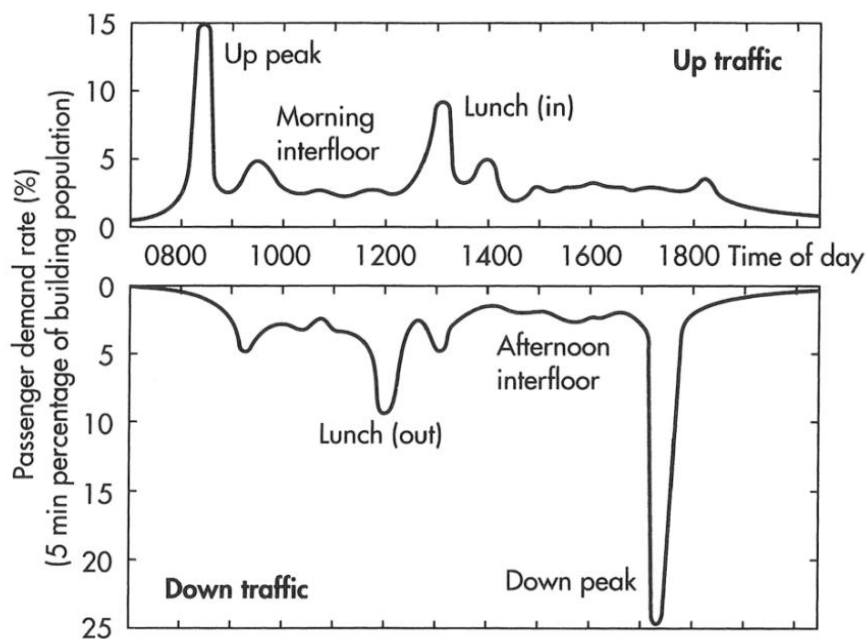


Figura 5. Padrões de tráfego em um prédio comercial [2].

Na Figura 5, podem-se observar quatro tipos de padrões de tráfego nas diferentes horas de trabalho em um prédio comercial (das 7 horas até às 20 horas), assim:

- **UP-PEAK:** é o tráfego de usuários no início do dia, quando os usuários chegam ao prédio para trabalhar. Apresentam-se entre as 7 horas e as 9 horas da manhã com uma intensidade máxima de tráfego de 15% do total da população do prédio. As chamadas de serviço para os elevadores originam-se em uns 90% dos casos no térreo e o andar de destino pode ser qualquer outro andar [2].

Esse padrão é considerado como uma condição crítica para o sistema de controle de elevadores, pois durante esse momento do dia tem-se uma faixa de tempo de 5 minutos

na qual 15% da população precisa do serviço dos elevadores, gerando a necessidade de usar os elevadores de forma ótima. Na Figura 6, pode-se observar esse padrão de tráfego em uma faixa de tempo de 1 hora, onde a demanda é representada por uma curva Gaussiana.

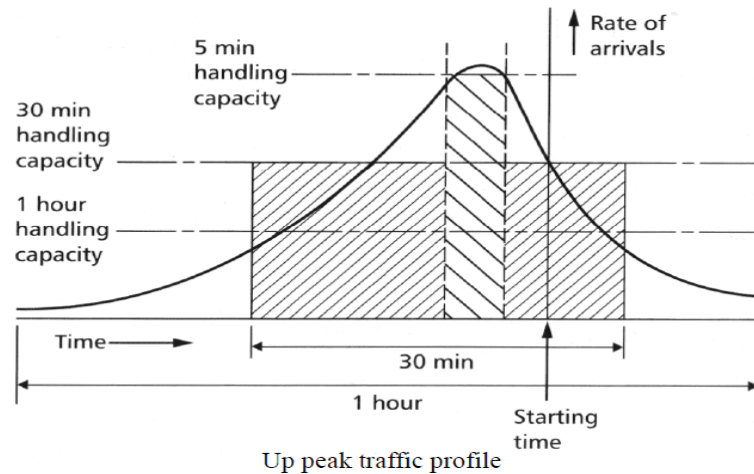


Figura 6. Intensidade de tráfego *up-peak* [2].

- **INTERFLOOR:** é o tráfego que se apresenta na maior parte do tempo, em prédios comerciais, caracterizando-se por uma demanda baixa de serviço dos elevadores com o andar de origem e de destino aleatórios. A porcentagem de demanda é de 5% da população subindo e outros 5% descendo. Esse padrão acontece aproximadamente das 9 às 11 horas da manhã e das 14 às 17 horas da tarde [9].
- **LUNCH:** é o padrão de tráfego que se apresenta na hora do almoço em um prédio comercial. A demanda está dividida em duas partes: (1) usuários saindo para almoçar entre as 11h30min e às 12h30min, onde a maioria das chamadas tem como destino o andar térreo; e, (2) depois do almoço, quando os usuários voltam entre as 13 e as 14 horas, onde a maioria das chamadas se origina no térreo com andar de destino aleatório. Cada uma das partes do padrão *lunch* tem uma intensidade de 10% da população [2]. É importante salientar que nesse tipo de padrão existe uma exceção quando o prédio conta com um restaurante no último andar, situação na qual o andar de origem e destino pode mudar.
- **DOWN-PEAK** é o padrão de tráfego que se apresenta quando o dia de trabalho acaba no prédio comercial, entre as 17 e 20 horas. Nesse tipo de padrão, a característica principal é que as chamadas têm como andar destino o térreo, e há uma faixa de 10

minutos com o maior desafio para o funcionamento ótimo dos elevadores, pois nesse intervalo a demanda de serviço chega a ser de aproximadamente 25% da população do prédio [2]. O padrão de tráfego *down-peak* é representado na Figura 7. Nessa figura pode-se observar uma curva Gaussiana invertida com a maior quantidade de chamadas feitas em uma faixa de 10 minutos.

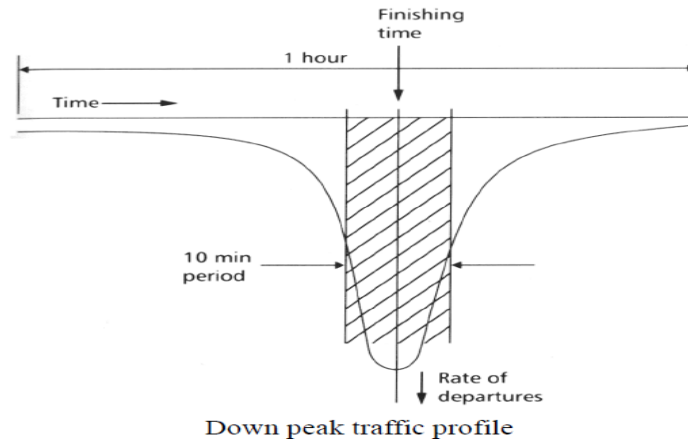


Figura 7. Intensidade de tráfego *Down-peak* [2].

Fazendo um resumo das porcentagens da população de padrões de tráfego estão a Figura 8 e a Tabela 3, na qual **A** representa a porcentagem do total da população viajando do térreo (*Up-peak*); **B** é porcentagem do total da população que se encontra viajando e que tem como andar de destino o térreo (*Down-peak*); e **C** é a porcentagem do total da população que se encontra viajando e que tem como origem e destino andares diferentes do térreo (*Interfloor*).

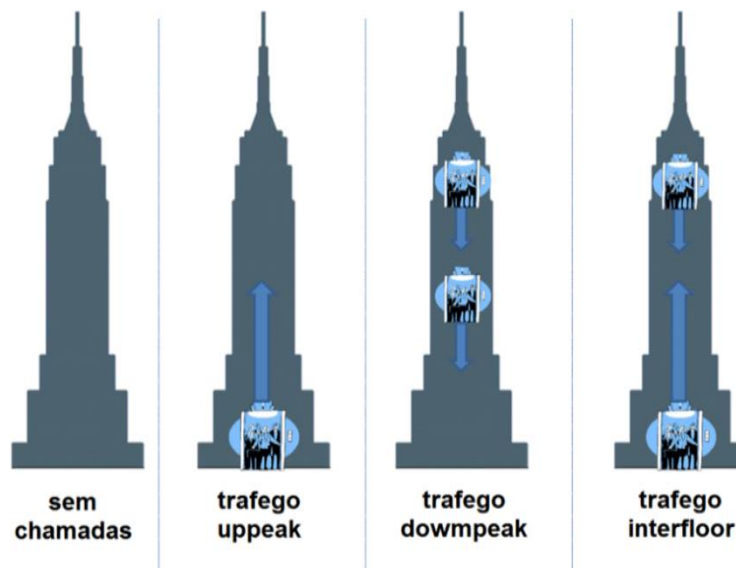


Figura 8. Chamadas de acordo com os padrões de tráfego

Tabela 3. Porcentagens de padrões de tráfego [13].

Padrão	Parâmetros		
	A	B	C
<i>Up-peak</i>	90	5	5
<i>Interfloor</i>	45	45	10
<i>Lunch</i> (Saída almoço)	20	60	20
<i>Lunch</i> (Entrada almoço)	70	10	10
<i>Down-peak</i>	5	90	5

2.1.5. Outros tipos de padrões de tráfego

Podem-se ter diferentes tipos de padrões de tráfego relacionados com a dinâmica dos usuários que fazem uso do prédio a ser analisado. Assim, pode-se pensar em um prédio comercial com tempos de trabalho denominados flexíveis. Nesses tipos de prédios, os trabalhadores têm horas diferentes de entrada e saída, porém, é comum ter mais de um padrão de tráfego repetido, por exemplo, no caso de se ter dois momentos de entrada dos usuários (duas vezes o padrão *up-peak*) (vide Figura 9) [2].

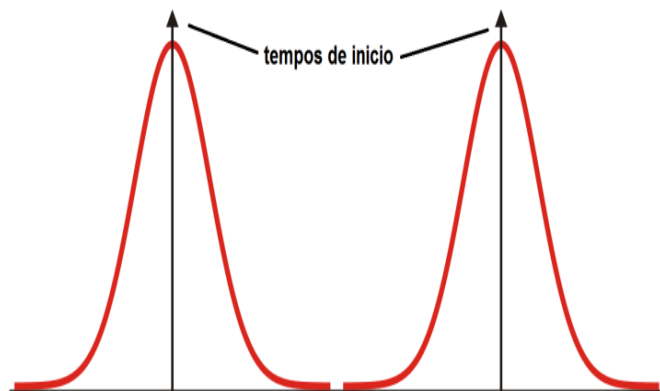


Figura 9. Prédio com dois momentos de tráfego *Up-peak*.

2.2. SOFTWARE ESCOLHIDO PARA A SIMULAÇÃO DOS SISTEMAS DE GRUPOS DE ELEVADORES: ARENA®

A escolha do software de simulação *Arena* reside basicamente na inspiração da ideia deste trabalho, uma pesquisa desenvolvida na mesma área e na mesma universidade planteava

como trabalho futuro o uso dessa ferramenta, baseando-se nas vantagens e potencialidades desse software na simulação de processos industriais, além da existência e disponibilidade dele na universidade (licenças tipo estudante e *basic*)

Arena é um ambiente gráfico integrado de simulação, dotado de recursos para modelagem de processos, desenho e animação, análise estatística e análise de resultados. Com ele, podem- se modelar e simular diversos processos, entre os quais estão análises de processos industriais, de filas e de linhas de produção. Este software também permite prever o comportamento dos ambientes modelados, sem que haja necessidade de fazê-lo no mundo real. (vide Figura 10).



Figura 10. Software proprietário *Arena* (*Rockwell Automation*).

A simulação de processos é uma das ferramentas para o estudo de sistemas baseada em protótipos ou modelos. Modelos matemáticos podem ser usados para se obterem soluções analíticas dos problemas formulados ou para fins de análise de comportamento de sistemas mediante sua simulação em sistemas computacionais. O software *Arena* foi desenvolvido com o intuito de possibilitar a simulação de sistemas, fornecendo para este caso uma interface gráfica baseada na linguagem *SIMAN* [14] [15]. As simulações são ferramentas para o auxílio na tomada de decisões e possibilitam a criação de testes simulados dos diversos modelos que representam sistemas dentro da indústria, permitindo com os resultados das simulações inferir os comportamentos dos sistemas sob influência de diversos cenários possíveis, sem a necessidade da implementação física dos sistemas. A Figura 11 mostra a simulação de uma linha de montagem em que se podem simular diferentes condições e verificar os tempos de ciclo de produção resultantes.

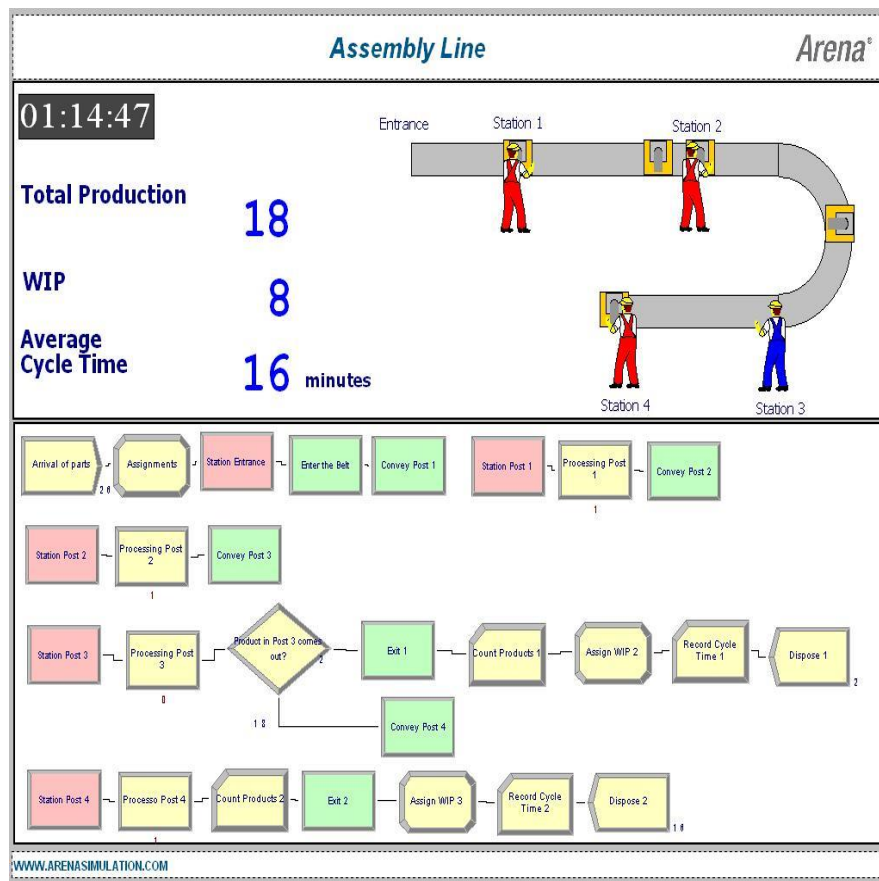


Figura 11. Simulação de processos com *Arena*. [14]

Os sistemas modelados e simulados normalmente são reduzidos a um conjunto de estações de trabalho que prestam diferentes tipos de serviços a entidades (clientes ou matérias primas), as entidades são criadas de acordo com uma distribuição probabilística (normal, exponencial, uniforme, entre outras) e se movimentam através de estações ou módulos que estão interconectadas com outras, seguindo uma sequência definida e lógica, até sair do sistema (por exemplo, termina o processo de produção de um objeto). As entidades têm diferentes tipos de características conhecidas como atributos (tamanho, peso, cor, etc.) [14]. Os diferentes valores dos atributos podem definir as ações das entidades dentro do sistema modelado (vide Figura 12)

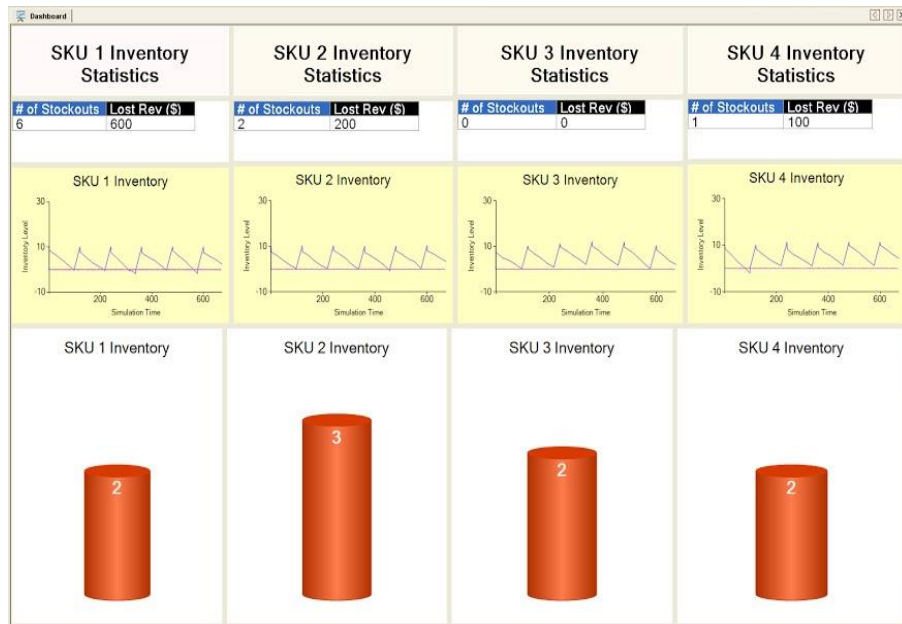


Figura 12. Entidades, filas, processos, e outros. [14]

Além das possibilidades de simular, *Arena* fornece ferramentas de acompanhamento e ajuda na simulação com as quais é possível verificar o modelo criado, identificando erros lógicos e melhorando a modelagem de forma a obter modelos mais claros e concisos. A possibilidade de gerar animação dos modelos desenvolvidos permite ver mais claramente os elementos do sistema representado

O software *Arena* acrescenta também vantagens na parte estatística, pois o modelo simulado coleta dados referentes aos processos, os armazena em um banco de dados para ao final gerar relatórios referentes à simulação. Entre os dados mais importantes estão os relacionados com os tempos do transcurso das entidades no sistema simulado, bem como para cada processo pelo qual a entidade passou durante a simulação (*wait time*, *transfer time*, entre outros). Estes tempos estão relacionados individualmente às entidades, gerando valores médios de tempos de espera, de tempos em filas e em processamento de serviços, necessários para gerar análises mais completas dos sistemas simulados [14] [15].(vide Figura 13).

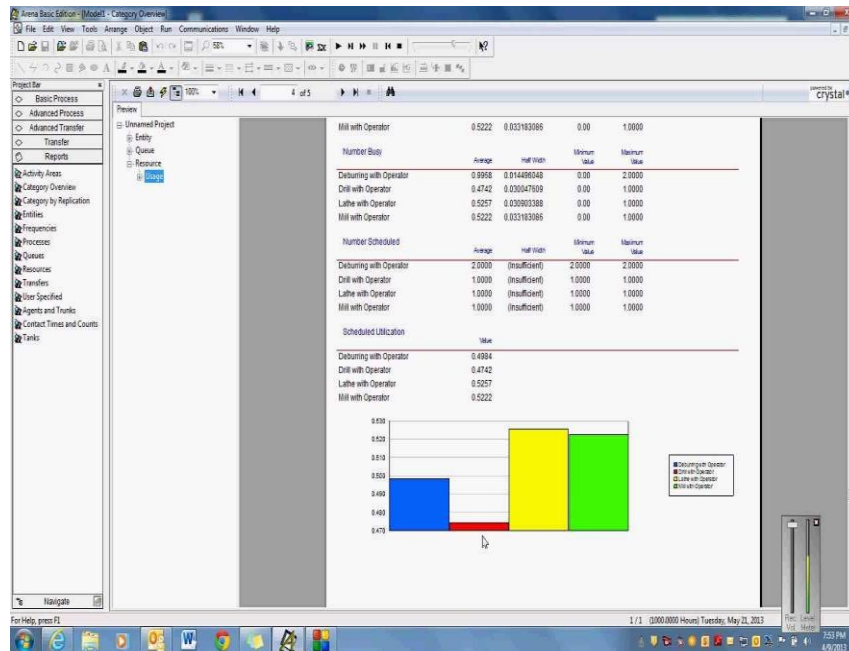


Figura 13 reportes depois da simulação [14]

Por último, é importante entender que as vantagens descritas foram as razões principais para escolher esta ferramenta como software para desenvolver o simulador pretendido neste trabalho. Além disso, é importante saber que este software é compatível com o sistema operacional *Windows*[®] da empresa *Microsoft*, sendo compatível com programas do pacote *MS office* (Visio, Excel, Access). Esta poderosa ferramenta de software permite desenvolver parte de sua programação em *VBA* (*Visual Basic for Applications*). Estes códigos são usados para potencializar e fazer uma melhor integração entre os usuários e o modelo de simulação [16]. Vide Figura 14

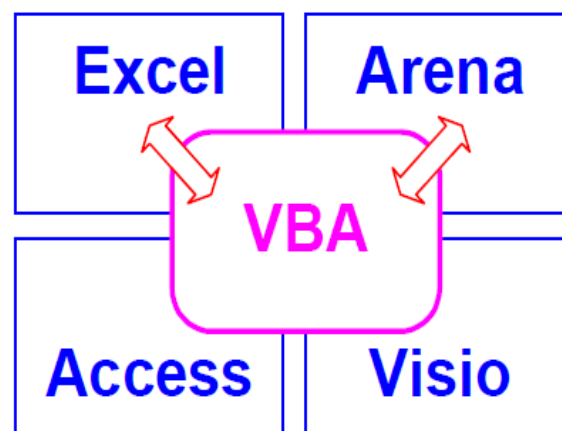


Figura 14. Arena e conexão com outros aplicativos. [16]

Esta possibilidade de fazer uso de VBA foi de muita utilidade no desenvolvimento do simulador, pois nessa linguagem foi feita a programação do algoritmo PSO de tipo binário encarregado de determinar o elevador indicado ao atendimento da chamada analisada. Nos seguintes itens, apresenta-se cada parte desenvolvida do simulador.

2.3. CONTROLE DE TRÁFEGO EM SISTEMAS DE GRUPOS DE ELEVADORES

As características dos sistemas de elevadores apresentadas nos tópicos anteriores demonstram a importância e a necessidade de se ter um ótimo controle do uso dos elevadores com o fim de garantir tempos adequados de serviço prestado aos usuários de um prédio em qualquer situação de tráfego. Nesta seção, apresenta-se uma breve visão dos sistemas mais utilizados na literatura para o controle de sistemas de grupos de elevadores. As diferentes técnicas estão, em grande parte, relacionadas com os avanços tecnológicos; e, neste caso, antes de abordar os tipos de controle pode-se restringir o problema definindo a forma como os usuários fazem as chamadas de elevadores.

2.3.1. Sistemas de Controle Destino DCS

Quando uma pessoa precisa utilizar um elevador há, duas possibilidades para fazer a chamada de serviço:

- **Painéis clássicos:** esta tecnologia é comumente usada na maioria dos prédios, especialmente nos mais antigos. As chamadas são feitas através de painéis instalados em cada andar que permitem indicar o sentido do andar de destino, como mostrado na Figura 15 [2]. Além disso, essa tecnologia inclui uma botoeira na cabine do elevador para indicar o andar de destino. Esse sistema de chamada funciona adequadamente em prédios com um ou dois elevadores e em prédios com padrões de tráfego pouco exigentes. Uma situação particular que se apresenta com esse tipo de painéis é a possibilidade do usuário chamar vários elevadores, possibilitando reduzir o tempo de espera. Entretanto, deve-se considerar que todos os elevadores chegarão à esse andar para atender a mesma chamada e que o usuário pode entrar no elevador inadequado (usar um elevador que se dirige na direção contrária, ou um elevador com muitas

paradas agendadas). Esses tipos de painéis permitem mudar em qualquer momento o andar de destino aumentando, assim, o tempo de voo [3].



Figura 15. Método clássico de chamada de serviço [10].

Essa técnica não permite um adequado controle e despacho dos elevadores de forma ótima, uma vez que, os andares de destino são desconhecidos no momento que se despacha um elevador e já que estes podem ser alterados quando o usuário está dentro da cabine.

- **DCS (*Destination Control System*)**: este sistema é muito usado em prédios modernos; caracteriza-se pelo fato de que o usuário faz sua chamada somente em um painel que se encontra fora dos elevadores em cada andar [3]. Nesse sistema, o usuário insere o andar de destino e o sistema devolve o melhor elevador para realizar a viagem (aquele que cumpre com requisitos de tempo de espera, tempo de voo e disponibilidade de carga); vide Figura 16.

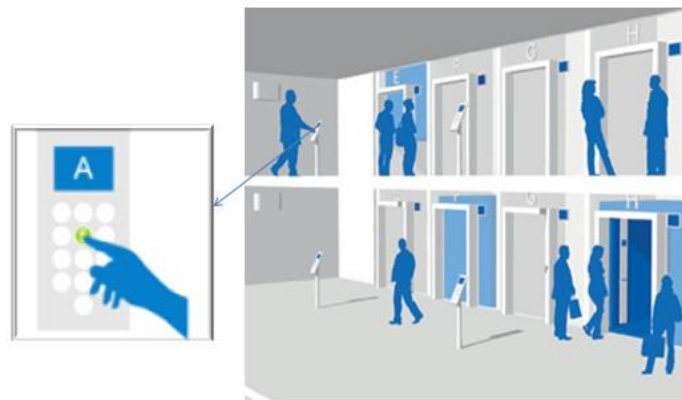


Figura 16. DCS: Sistemas de Controle de Destino.

Com o uso dos DCS pode-se adaptar adequadamente o funcionamento dos elevadores ao padrão de tráfego presente nos prédios [17]. Contudo, esse sistema apresenta uma dificuldade para o usuário, a qual radica em que não é possível mudar o andar de destino uma vez dentro do elevador. Se o usuário ainda não tiver entrado no elevador, ele pode fazer uma nova chamada no DCS, gerando a possibilidade de alocar novas chamadas por erros dos usuários.

2.3.2. Controle de tráfego

É importante salientar a necessidade de se ter um controle ótimo para o uso dos sistemas de grupos de elevadores. O projeto e seleção de um sistema de controle representa um desafio para pesquisadores na área da Automatização Predial. Na literatura científica há dois tipos de técnicas, convencionais ou clássicas e as modernas, que visam otimizar os recursos disponíveis no intuito de reduzir o tempo dos usuários e o consumo de potência [3].

- **Controle clássico ou convencional**

Trata-se de técnicas largamente empregadas por grandes empresas de produção, comercialização e instalação de elevadores tais como Otis, Westinghouse, Schindler, entre outras. Encontram-se geralmente em prédios com padrões de tráfego não exigentes e também na maioria dos prédios antigos. Esse tipo de técnica de controle se caracteriza por usar lógicas simples de funcionamento, onde, em muitos casos, a otimização do uso dos elevadores deixa a desejar, como é possível ver a seguir [18].

Control Near Car (NC): controle baseado na escolha do elevador para atender uma chamada de acordo com as seguintes condições e parâmetros: a direção da chamada (subir ou descer) e o número de andares entre a posição do elevador e do andar onde a chamada foi feita. A ordem de atenção depende da ordem de ocorrência, assim, o elevador mais próximo e na mesma direção da chamada é o encarregado de prestar o serviço. Quando são atendidas todas as chamadas, os elevadores retornam para o andar principal (por exemplo, o térreo) [3].

Fixed sectoring common sector system (FSO): nesse tipo de controle o prédio está dividido em setores, permitindo que um elevador atenda apenas as chamadas feitas em um determinado setor. Os elevadores só podem sair do seu setor para levar usuários

para outros setores. Entretanto, para responder chamadas de outros setores eles são considerados sem capacidade disponível. A divisão do prédio pode não ser equitativa e é possível ter setores com mais andares que outros, bem como ter mais do que um elevador associado à um setor, indicando maior importância ou uma população maior [2].

Fixed sectoring priority sector system (FS4): esse tipo de controle divide o prédio em setores com diferentes prioridades de atendimento pelos elevadores. As prioridades definidas se baseiam na importância de alguns andares ou setores e também na possibilidade de haver uma demanda de chamadas maior que em outros setores do prédio. Nesse tipo de controle, uma chamada feita em um andar com prioridade alta é atendida pelo elevador disponível que estiver mais próximo ao andar de origem, ou que estiver indo para ele em voo direto para atender outras chamadas feitas com prioridades menores [2].

Em contrapartida às técnicas apresentadas, é importante conceitualizar o grupo de técnicas de controle consideradas como modernas. Esses tipo de técnicas se baseiam no uso de processos de predição, aprendizagem e ponderação de critérios, tais como, tempo médio de espera, tempo de voo, capacidade do elevador, direção, consumo de energia, entre outros, otimizando e melhorando o uso dos sistemas de elevadores. Estas técnicas usam lógicas de controle complexas e, geralmente, envolvem um custo computacional elevado.

- **Sistemas de controle moderno**

Existem diversas técnicas de controle de elevadores modernas presentes na literatura, dentre as quais pode-se destacar as que se baseiam em técnicas de otimização não clássica, como as técnicas de busca estocástica e as técnicas de busca bioinspiradas nos comportamentos de seres vivos. Na literatura pode-se encontrar algumas técnicas modernas já empregadas e também propostas para o controle de sistemas de grupos de elevadores [2].

Controle com lógica nebulosa: essa técnica é uma das mais vistas na literatura científica, empregada como solução em EGCS [4] [19]. Ela possibilita trabalhar com sistemas múltiplos-variáveis a fim de selecionar o melhor elevador para uma chamada. Essas variáveis se caracterizam por estar representadas linguisticamente (vide Figura

17). Essa representação é conhecida como *conjuntos nebulosos*, [4] [19], diferenciando-se dos conjuntos clássicos pelo fato de que um elemento de um conjunto nebuloso não precisa obrigatoriamente pertencer inteiramente a ele (grau de pertinência ao conjunto nebuloso) sem ter uma completa exclusão, (vide Figura 18)

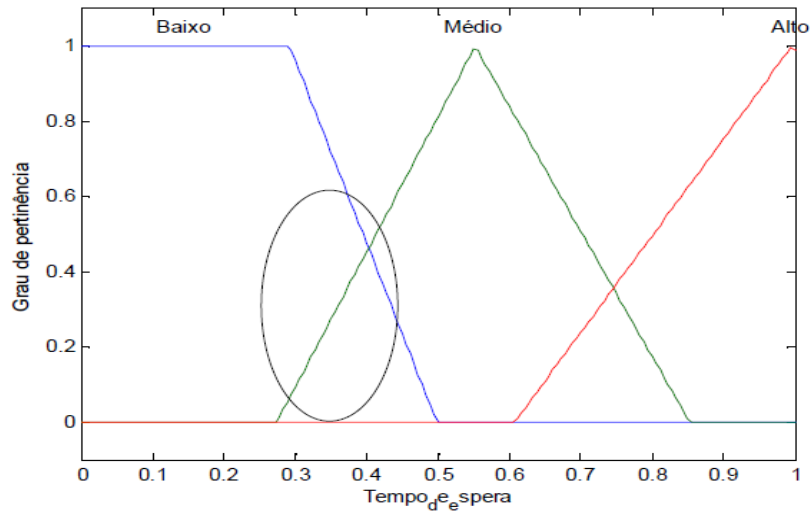


Figura 17. Variáveis com graus de incerteza

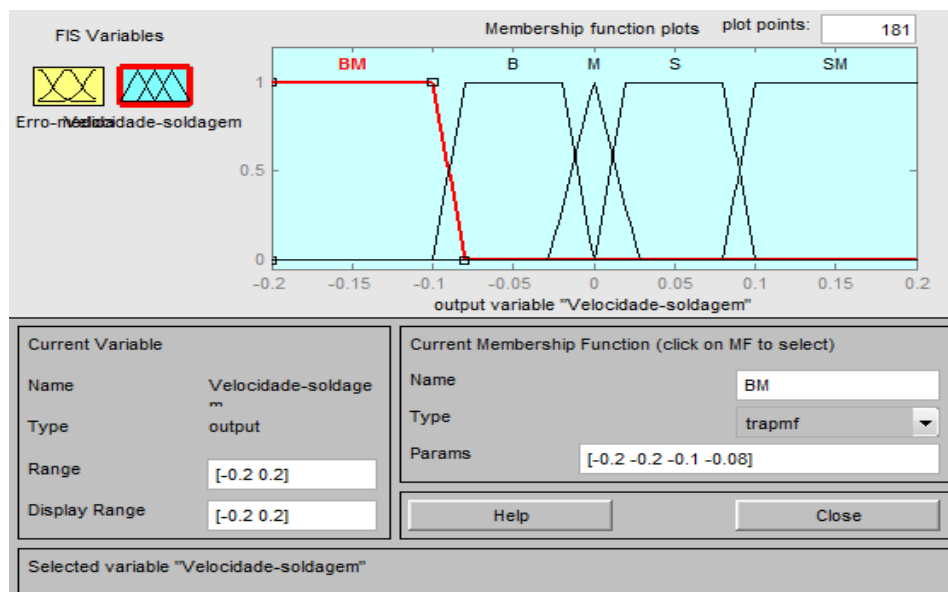


Figura 18. Variáveis em conjuntos no Matlab [10].

A combinação das variáveis linguísticas gera regras de decisão do tipo linguístico. Essas regras representam o conhecimento do problema, e a definição adequada dessas regras permite determinar com maior probabilidade o melhor elevador para uma determinada chamada (vide Tabela 4).

Tabela 4. Base de regras de um controle nebuloso [10].

	Regras de inferência	Prioridade
R1	SE(Distância ==Alto & Tempo de espera == Alto)	Baixa
R2	SE(Distância == Alto& Tempo de espera == Baixo)	Média
R3	SE(Distância == Alto& Tempo de espera == Médio)	Baixa
R4	SE(Distância == Baixo & Tempo de espera == Alto)	Média
R5	SE(Distância == Baixo & Tempo de espera == Baixo)	Alta
R6	SE(Distância == Baixo & Tempo de espera == Médio)	Média
R7	SE(Distância == Médio& Tempo de espera == Médio)	Média
R8	SE(Distância == Médio & Tempo de espera == Alto)	Baixa
R9	SE(Disponibilidade de Carga == Baixo&Tempo de espera == Alto)	Alta
R10	SE(Distância == Baixo&Disponibilidade de Carga == Baixo)	Baixa
R11	SE (Disponibilidade de Carga == Médio &Tempo de espera == Baixo)	Baixa

Na Figura 19, apresenta-se o processo de avaliação de entradas relacionadas a um possível elevador para servir a uma chamada. Note que o valor de saída depende das regras anteriormente descritas.

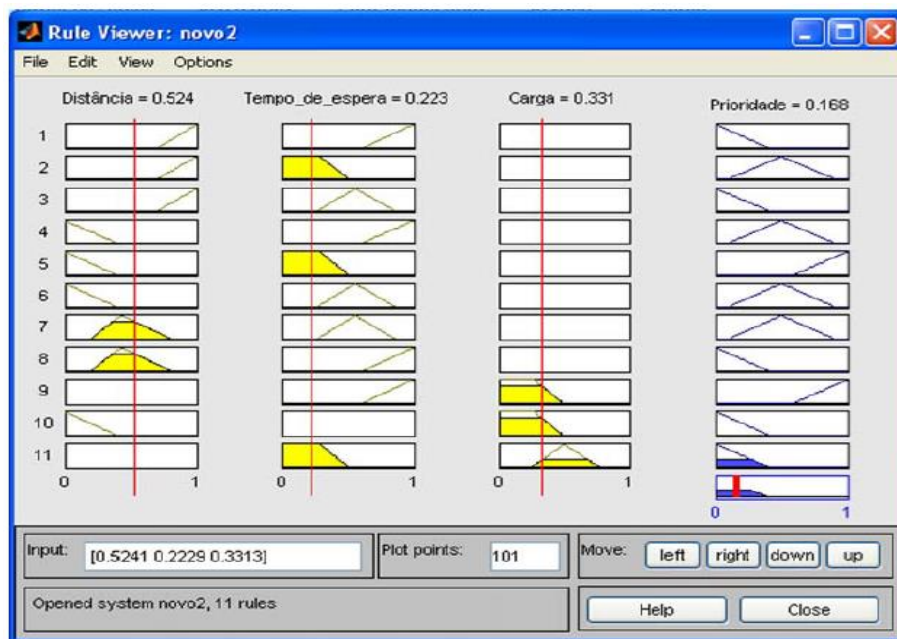


Figura 19. Interação de variáveis de entrada e a saída de um controle nebuloso.

Controle com redes neurais: essa técnica também é utilizada para solucionar problemas como os do tipo dos sistemas de elevadores e é inspirada no modelo de processamento de informações dos neurônios do cérebro dos animais. Esse modelo recebe múltiplas entradas (variáveis do problema), e a interação que se dá entre elas

(sinapses - as entradas superam um umbral que polariza o neurônio ativando sua saída) através dos neurônios, gera uma saída ou possível solução para o problema (vide Figura 20) [20]. O funcionamento da rede depende de parâmetros denominados *pesos* que são ajustados na medida em que a rede é treinada melhorando, assim, sua resposta para o problema.

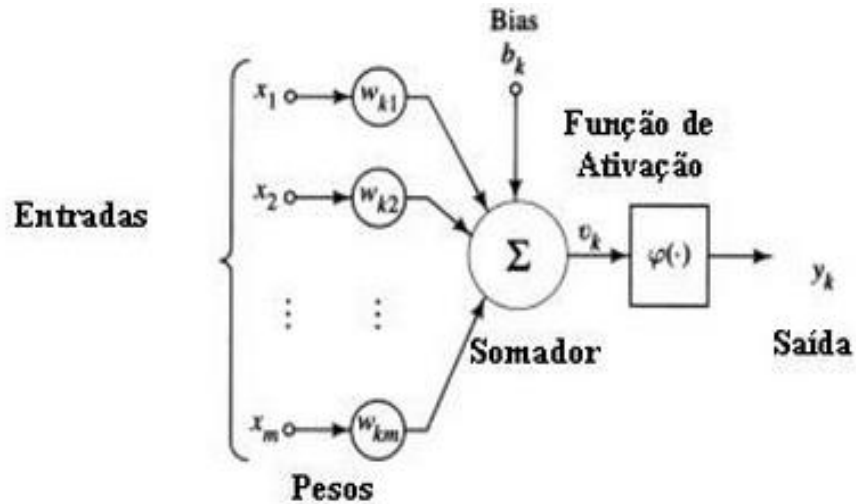


Figura 20. Neurônio artificial.

Na Figura 21, pode-se observar o diagrama de um neurônio artificial e de uma rede neural formada por vários neurônios.

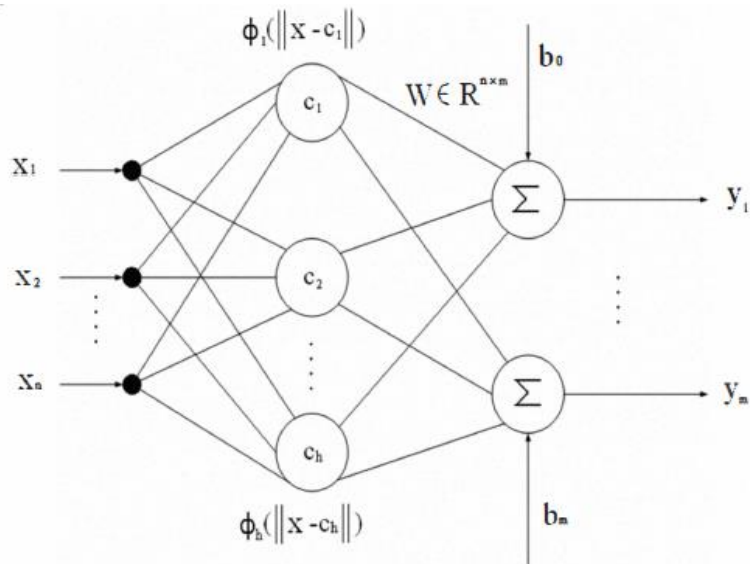


Figura 21. Rede neural artificial

As técnicas anteriores podem se associar a outras existentes na literatura que podem ser usadas para resolver o problema do controle de sistemas de elevadores, como por

exemplo, os Algoritmos Genéticos (AG), os Algoritmos Evolutivos (AE), a Programação Evolutiva (PE) e a Programação Genética (PG), todas inspiradas no comportamento da teoria da evolução de Darwin [12], [21].

Neste trabalho serão analisadas as técnicas baseadas em algoritmos evolutivos, especificamente os algoritmos de otimização bioinspirados. De preferencia, serão exploradas particularmente, as capacidades de busca local e global dos algoritmos baseados em inteligência de enxames em espaços discretos, de forma a encontrar o melhor elevador para atender uma determinada chamada, minimizando, assim, o critério de tempo de espera dos usuários.

3. ALGORITMOS DE OTIMIZAÇÃO BASEADOS EM INTELIGÊNCIA DE ENXAMES

A definição de otimização matemática é a capacidade de obter o melhor resultado possível dentro de um conjunto de soluções em um espaço finito ou infinito visando minimizar os esforços ou maximizar os lucros. Para verificar se uma solução é ótima é necessário obter uma representação matemática, chamada de “*função objetivo*” ou “*função custo*”, que permita avaliar cada solução. Esta representação é formada por uma ou várias variáveis de decisão e um conjunto de restrições que afetam o problema de otimização. Desta forma, o objetivo da otimização é encontrar o melhor valor ou valor aceitável da função custo (o maior valor numérico possível implica em maximização e o menor valor numérico possível implica em minimização) [22].

Algoritmos bio-inspirados são técnicas de otimização muito usadas nos últimos anos para resolver problemas complexos com múltiplas variáveis nos quais soluções robustas são difíceis ou impossíveis de encontrar usando métodos clássicos (baseados no cálculo do gradiente). Esse tipo de algoritmo imita o comportamento social de algumas espécies da natureza, mas precisamente de espécies que possuem inteligência coletiva. A característica principal destes algoritmos é o processo de interação entre os indivíduos do enxame os quais estão baseados em regras simples, porém, usam suas capacidades de inteligência coletiva para realizar tarefas complexas [22].

Os algoritmos baseados em enxames estão conformados por agentes que representam individualmente uma possível solução a um problema. Os agentes podem mudar de posição no espaço de busca procurando encontrar uma posição na qual é melhorada a função custo que modela o problema de otimização. A mudança de posição é influenciada pela interação com os outros agentes (todo o enxame ou só os vizinhos). Cada agente sabe qual é a melhor posição encontrada pelos vizinhos, conformando o conhecimento coletivo que apresenta o enxame [22].

Nos subtítulos que se seguem, são apresentados os algoritmos de otimização por inteligência de enxames que vão ser explorados nesta pesquisa visando obter uma possível solução ao problema de controle de grupo de elevadores.

3.1. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO)

A técnica PSO (*Particle Swarm Optimization*) é um algoritmo de otimização inspirada no comportamento social do voo dos bandos de pássaros e também no movimento dos cardumes de peixes durante a busca por alimento. Este algoritmo foi desenvolvido em 1995 por dois pesquisadores de duas áreas diferentes o psicólogo James Kennedy e o engenheiro Russell Eberhart [7].

A inspiração natural deste algoritmo pode ser explicada assim: seja um bando de pássaros que procura uma fonte de alimento em uma área delimitada. No início, os pássaros do bando voam aleatoriamente na área de busca e comunicam-se entre eles quando encontram uma nova fonte de alimento. Ao interagir entre si, o enxame segue o pássaro que esteja mais perto da melhor fonte de alimento. O algoritmo PSO emula este comportamento para resolver problemas de otimização, de forma que as características fundamentais para seu funcionamento são o conhecimento individual, representado pelo histórico de cada partícula, e o conhecimento social, representado pelo histórico das partículas vizinhas.

O PSO é muito usado em diferentes áreas para resolver problemas de otimização ou como complemento para outras técnicas de controle [23] [24].

3.1.1. Algoritmo PSO básico

O algoritmo PSO pode ser descrito, de forma geral, como um conjunto de vetores que contem dados ligados à posição das partículas, as quais se movimentam em uma região ou área definida (vide Figura 22). Neste algoritmo, a nova posição de uma partícula é definida por sua experiência particular, conhecida como memória individual (a partícula lembra-se da melhor posição determinada pela avaliação da função custo). Da mesma forma a partícula é influenciada pela experiência global do enxame, conhecida como memória coletiva (a partícula lembra a melhor posição do enxame determinada pela melhor aptidão ao ser avaliada na função custo) [7].

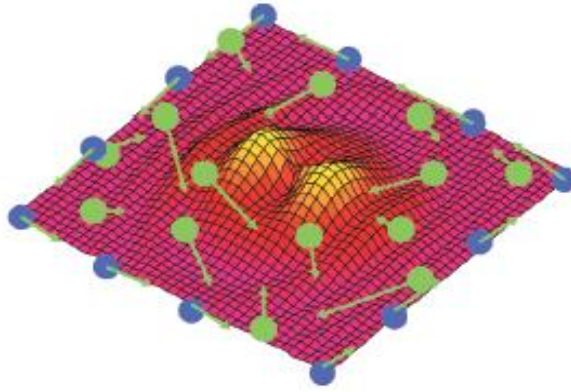


Figura 22. Movimentos das partículas.

A nomenclatura do PSO é a seguinte [7]

- Partícula: indivíduo do enxame.
- Enxame: conjunto de partículas.
- Posição (\mathbf{x}): coordenadas de uma partícula no espaço N -dimensional (possível solução de um problema)
- Aptidão: valor que representa quão boa é uma solução. Resultado da avaliação de uma posição por meio de uma função custo.
- Pbest (\mathbf{y}_i): memória individual da partícula determinada pela posição do melhor valor de aptidão encontrado pela partícula
- Gbest (\mathbf{y}_s): memória coletiva do enxame determinada pela posição do melhor valor de aptidão encontrado entre todas as partículas.
- Vmax: velocidade máxima possível para uma partícula.

Estes parâmetros são a base para o algoritmo. É importante descrever as equações que os representam e definem as mudanças de posições das partículas no tempo, quando elas se movimentam procurando a melhor solução em um espaço de N dimensões. As equações (3-1) e (3-2), mostram a atualização da posição da i -ésima (i^{th}) partícula na j -ésima (j^{th}) dimensão, $x_{ij}^{(t+1)}$, para um enxame com S partículas.

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1U_{1j}(y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2U_{2j}(y_{sj}^{(t)} - x_{ij}^{(t)}) \quad (3-1)$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (3-2)$$

Em que:

- U_{1j} e U_{2j} são números aleatórios uniformemente distribuídos entre 0 e 1
- c_1 e c_2 são os coeficientes cognitivo individual e social, respectivamente. Um valor grande de c_1 indica partículas com alta confiança na sua experiência, enquanto um valor grande de c_2 proporciona maior confiança das partículas na experiência do enxame.
- w é denominado o fator de inercia. Este parâmetro é aplicado durante o cálculo da velocidade, sendo utilizado como um fator de escala para a velocidade atual de cada partícula. Desta forma define a influência da velocidade na medida em que o algoritmo vai sendo executado.
- y_{ij} é a melhor posição individual da i^{th} partícula na j^{th} dimensão.
- y_{sj} é a melhor posição global entre todas as partículas na j^{th} dimensão.
- v_{ij} estão limitadas na faixa $[-Vmax, Vmax]$ evitando assim que as partículas abandonem o espaço de busca.

Nesta equação, pode-se observar que a posição para o próximo instante de tempo depende da posição atual e do cálculo da velocidade. Entretanto, a velocidade para o próximo instante de tempo da partícula depende diretamente da velocidade atual, da melhor posição encontrada pela partícula (y_i) e da melhor posição encontrada pelo enxame (y_s).

A nova posição de uma partícula pode se observar como uma soma vetorial dos parâmetros y_{sj} e y_{ij} (vide Figura 23).

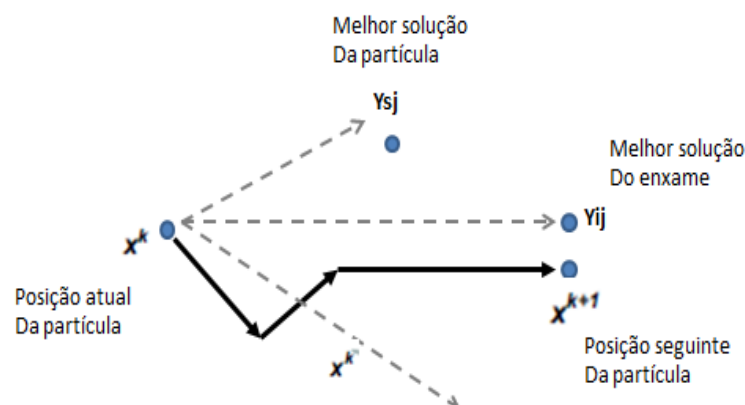


Figura 23. Soma vetorial para definir a nova posição da partícula.

Com base nos parágrafos anteriores a Figura 24 mostra o pseudocódigo do algoritmo PSO na sua forma básica.

Algoritmo: PSO básico

Entrada: $S, N, f, c_1, c_2, x_{max}, v_{max}, Max_{iter}$

Saída: posição da particular: x e seu melhor valor de aptidão: $f(x)$

Início

// Inicializa o enxame

Para $k=1: S$ **faça**

Para $j=1: N$ **faça**

$$v_{kj} = -v_{max} + 2U[0,1]_{V_{max}}$$

$$x_{kj} = -x_{max} + 2U[0,1]_{X_{max}}$$

Fim para

Fim para

Repita

 //Avaliação e detecção

Para $k=1: S$ **faça**

Se $f(x_k) = f_{min_k}$ **então**

$$y_k = x_k$$

$$f_{min_k} = -f(x_k)$$

Fim se

Fim para

 //Calcule y_s usando os S valores de aptidão $f(y_k)$

 //Atualização usando equações

Para $k=1: S$ **faça**

Para $j=1: N$ **faça**

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1U_{1j}(y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2U_{2j}(y_{sj}^{(t)} - x_{ij}^{(t)})$$

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

Fim para

Fim para

Até iterações = Max_{iter}

Fim

Figura 24. Pseudocódigo do PSO [7].

3.2. ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE ABELHAS (ABC)

Na atualidade, existem vários algoritmos inspirados em diferentes tipos de comportamento das colônias de abelhas usados para resolver problemas de otimização. Neste trabalho será usado o algoritmo ABC (*Artificial Bee Colony*), inspirado no comportamento inteligente das colônias de abelhas quando procuram novas fontes de alimento. Este algoritmo foi proposto por Dervis Karaboga em 2005 [25]. A característica principal desta técnica é a

inteligência coletiva como papel fundamental, representada pela capacidade que tem as abelhas (agentes) de tomar decisões mediante o intercâmbio de informações entre elas.

Karaboga modela o comportamento das abelhas quando encontram uma fonte nova de alimento da seguinte maneira: A abelha que encontrou a nova fonte a descreve por meio de uma dança que permite compartilhar informações. A forma da dança é relacionada com a quantidade de alimento que a fonte contém (néctar), a distância da colmeia até a fonte e a direção em que a fonte está com relação ao sol [26] (vide Figura 25). Na colônia nem todas as abelhas procuram fontes de alimento. Há abelhas que cumprem funções diferentes. Podem-se classificar as abelhas em três tipos:

- Abelhas operárias: estão relacionadas com uma fonte de alimento e são as encarregadas de compartilhar a informação das novas fontes na colmeia por meio da dança.
- Abelhas seguidoras: aguardam na colmeia para olhar a dança das operárias e escolher se devem ou não seguir uma delas até a fonte de alimento (a melhor fonte de alimento tem maior probabilidade de ser seguida). As seguidoras podem aguardar na colmeia até encontrar uma representação que elas gostem e segui-la. Se depois de um tempo elas não gostam de nenhuma fonte, a abelha seguidora torna-se uma escoteira e sai da colmeia na busca aleatória de uma fonte nova de alimento.
- Abelhas escoteiras: são as que procuram novas fontes de alimento, uma abelha operária que tem uma fonte de alimento que se esgotou torna-se uma escoteira.

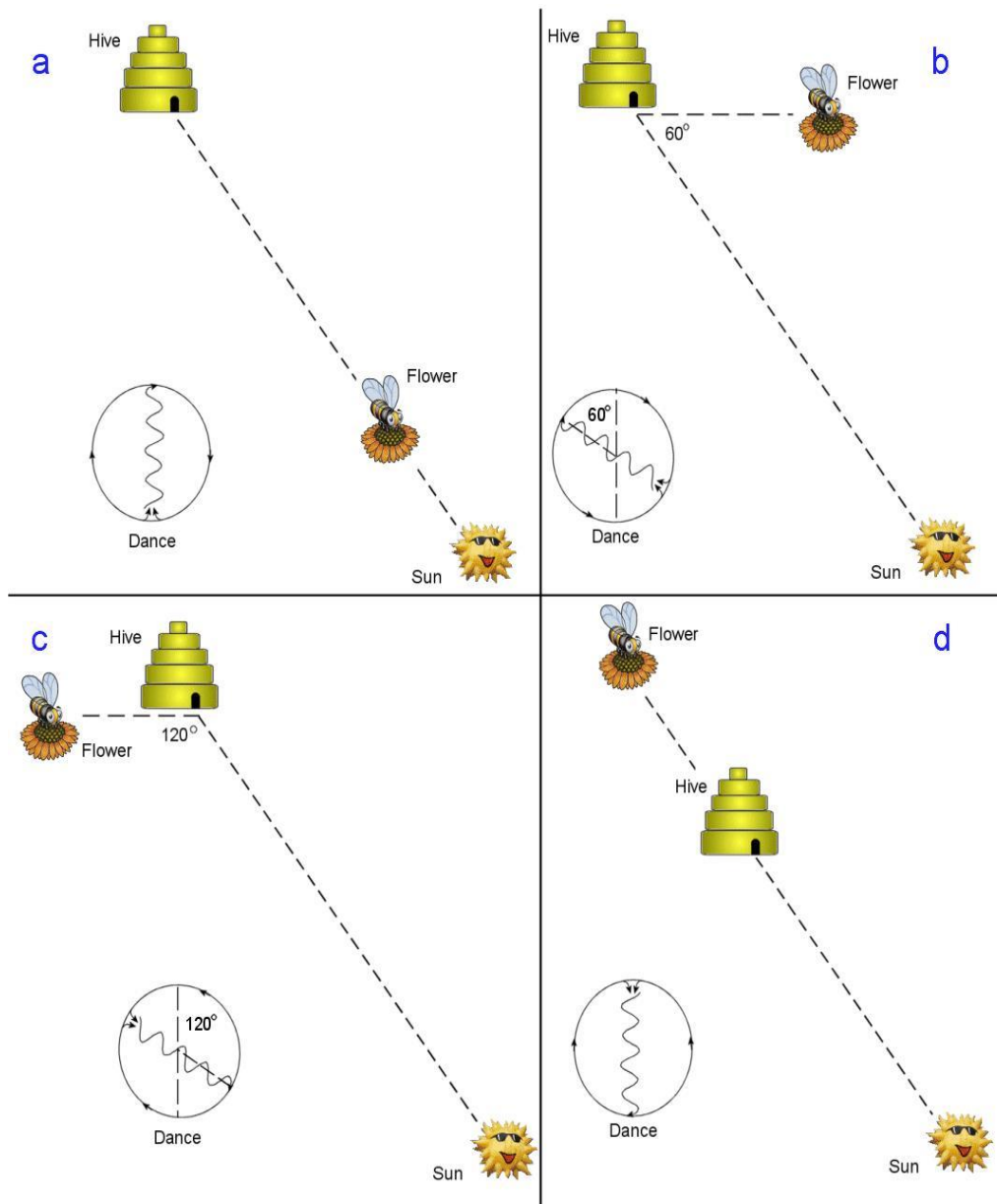


Figura 25 exemplo de fontes de alimento e representação pelas abelhas

3.2.1. Algoritmo ABC

O algoritmo tem como função achar a solução ótima a um problema dentro de um conjunto de possíveis soluções encontradas pelas abelhas artificiais. Para saber se a solução é a ótima, precisa-se avaliar cada solução usando uma função custo (aptidão) que possa representar numericamente sua qualidade, lembrando que a definição dessa função depende da modelagem do problema. A quantidade de possíveis soluções a serem avaliadas em cada instante de tempo é igual ao número de abelhas operárias, ou seja, ao número de fontes de alimento [25]. Desta forma o intercâmbio de informação que existe

entre as abelhas pode ser representado por uma probabilidade calculada em função da qualidade da fonte de alimento (néctar), como mostrado na equação (3-3).

$$p_i = \frac{f_i}{\max(f)} \quad (3-3)$$

nesta representação f_i é o valor de aptidão da i^{th} fonte de alimento, com $i=1, \dots, s$, sendo s (minúscula) o tamanho do enxame de abelhas operárias, S (maiúscula) é o número de abelhas da colmeia ($S=2s$); p_i é a probabilidade da i^{th} solução e $\max(f)$ o valor máximo da função custo entre toda a população. As abelhas seguidoras fazem uso destas probabilidades para escolher as fontes de alimento e assim encontrar novas soluções a serem exploradas mediante um processo de busca local com mostrado na equação (3-4) [27].

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + \phi_{ij} (x_{ij}^{(t)} - x_{kj}^{(t)}) \quad (3-4)$$

onde $k \in \{1, 2, \dots, s\}$ e $j \in \{1, 2, \dots, N\}$ são índices escolhidos aleatoriamente com k diferente de i ; N é o número de dimensões do problema e ϕ_{ij} é um número aleatório com distribuição uniforme na faixa $[-1,1]$. O pseudocódigo do algoritmo ABC é apresentado na Figura 26. Pode-se observar que o algoritmo está dividido em três partes, referentes ao tipo de abelha que executa o trabalho: abelhas operárias, abelhas seguidoras e abelhas escoteiras [27].

Na primeira parte, para cada solução i determina-se aleatoriamente um vizinho k (k diferente de i) e uma dimensão j . Posteriormente, determina-se a nova posição de cada abelha operária, calculando-se para cada nova posição o valor de aptidão e atualizando-se a posição se a mesma melhora o valor de aptidão anterior.

Na segunda parte, calculam-se as probabilidades p_i com os valores de aptidão calculados na primeira etapa. Depois de conhecer as probabilidades p_i , é feita a simulação do comportamento das abelhas seguidoras, calculando uma nova posição para cada uma delas. Se o valor da função objetivo é menor que o valor anterior (caso de minimização), então a nova posição é atualizada [25]. Se pelo contrário a posição não melhora em um determinado número de tentativas (*trial*) a abelha torna-se escoteira.

Na terceira parte, é simulado o comportamento das escoteiras. Quando uma fonte é abandonada, isto é, quando o néctar é esgotado depois de um número de tentativas (*maxtrial*), envia-se uma abelha escoteira em procura de uma nova fonte ou solução em uma posição aleatória. Em cada iteração do algoritmo, a função custo é avaliada 2s vezes (primeiro para todas as operárias e depois para todas as seguidoras).

Algoritmo: ABC básico

Entrada: $S, N, f, [X_{min}, X_{max}], maxTrial, maxIter$

Saída: posição da melhor fonte de alimento: X_g e seu melhor valor de aptidão: $f(x_g)$

Início

// Gerar as posições aleatórias para as S fontes de alimento e determinar aptidões $f(x_g)$

Repita

//Enviar operarias para explorar as fontes de alimento

Para cada solução i determinar um vizinho k e dimensão j

Usando a equações para uma nova possível solução

Calcular os valores de aptidão $f(x)$

Atualizar posições se $f(x_i)$ melhora o valor anterior

//Enviar seguidoras para explorar as fontes de alimento p_i

Calculando o valor de p_i com a equação

Para $i=1: S$ faça

Se $rand() \leq P_i$ então

//Determinar um vizinho k e dimensões j

Criar uma nova solução x_i usando a equação correspondente

Calcular o valor de aptidão $f(x_i)$

Atualizar posições se $f(x_i)$ melhora o valor anterior

Fim se

Fim para

//determine as soluções abandonadas e envie as abelhas escoteiras para

Buscar novas fontes de alimento

//atualize a melhor solução x_g seguindo as aptidões

Até iterações $\geq maxIter$

Fim

Figura 26 pseudocódigo do algoritmo ABC [25]

3.3. OTIMIZACAO POR VAGA LUMES (FA)

Algoritmo de otimização desenvolvido por Xin-She Yang [28] [29] da universidade de Cambridge. Esta técnica é inspirada no comportamento social dos vaga-lumes (pirilampos) na natureza. Estes insetos fascinantes têm como característica a produção de luz por meio de órgãos bio-luminescentes que se localizam na parte inferior da superfície de seu corpo. Esta capacidade é usada por estes insetos para se comunicar fazendo rituais de acasalamento. A intensidade da luz e a intermitência permitem ao vaga-lume atrair um companheiro ou companheira.

A representação do comportamento dos vaga-lumes feita por Yang descreve a situação quando estes insetos são atraídos entre eles por meio da luminescência de seus corpos para se reproduzir. No intuito de imitar este comportamento, pode-se representar a luminescência de cada vaga-lume por meio da capacidade de atração (β) e da intensidade da sua luz (I) ao piscar a diferentes velocidades. Destes parâmetros, a intensidade, $I(x)$, pode ser associada proporcionalmente com o valor da função custo(x), que depende sempre do problema a ser otimizado. Dado que a luz é absorvida pelo meio onde se propaga, precisa-se representar este fenômeno com um coeficiente de absorção (γ) que depende do meio e determina como a intensidade decresce com a distância desde a fonte de luz (vaga-lume) [28]. No algoritmo proposto por Yang, representa-se este comportamento matematicamente como indicado na equação (3-5).

$$\beta = \beta_0 e^{-\gamma \cdot r_{ik}^m}, (m \geq 1) \quad (3-5)$$

sendo β_0 a atração com $r_{ik} = 0$, e r_{ik} a distancia entre os vaga-lumes i e k . O parâmetro γ de absorção do meio caracteriza a variação da atração entre os vaga-lumes, sendo considerado para definir a velocidade de convergência do algoritmo. Este parâmetro é importante defini-lo em uma faixa (γ) adequada de valores possíveis. O parâmetro m (geralmente $m=2$) permite relacionar o comprimento característico e o coeficiente de absorção da seguinte forma $\beta = \beta_0 / (\gamma^{d/2})$, definindo assim uma atração que varia significativamente na faixa $[\beta_0; \beta_0/e]$. A posição do vaga-lume i sendo atraído por uma vaga-lume k determina-se pela seguinte equação (3-6) [28].

$$x_i^{(t+1)} = x_i^{(t)} + \beta(x_k^{(t)} - x_i^{(t)}) + \alpha(U_i - 1/2) \quad (3-6)$$

Nesta equação, têm-se três componentes que definem a nova posição: (1) experiência individual (x_i); (2) componente de atração entre indivíduos (componente social) e (3)

componente de aleatoriedade, onde U_i é um número aleatório com distribuição uniforme entre $[0,1]$ e α é um parâmetro de ajuste [28]. Na prática, utiliza-se $\beta_o=1$ e $\alpha \in [0,1]$. Este último parâmetro é utilizado para refinar a convergência na medida em que as iterações do algoritmo vão se realizando, na equação (3-7).

$$\alpha^{(t+1)} = \alpha^{(t)} \cdot \delta \quad (3-7)$$

Em que $\delta \leq 1$, sendo próximo de 1. Os vaga-lumes sempre se movimentam a uma nova posição, mesmo quando a sua nova posição não seja melhor que a anterior, sendo influenciada a sua direção pelo melhor vizinho mais próximo. Este comportamento na prática gera uma perda de desempenho dado que permite ao vaga-lume explorar muitas vezes soluções que não são ótimas. Na Figura 27, pode-se observar o pseudocódigo do algoritmo *FA (Firefly Algorithm)* básico.

Algoritmo: FA básico

Entrada: $S, N, f, [X_{min}, X_{max}], \alpha^{(t)}, maxIter$

Saída: posição do melhor vaga-lume: X_g e seu melhor valor de aptidão: $f(x_g)$

Início

// Gerar as posições aleatórias para x_i as S fontes de alimento e

Calcular as intensidades I_i avaliando a função custo $f(x_i)$

Determinar o coeficiente de absorção γ

Repita

Para $i=1: S$ **faça**

Para $k=1: S$ **faça**

Se $I_k > I_i$ **então**

 Calcular a distancia r_{ik} e a atração β com a equação

 Criar uma nova solução x_i

 Avaliar a nova solução $f(x_i)$ e

 Atualizar a intensidade I

Fim se

Fim para

Fim para

 Ordena os vaga-lumes segundo o valor de aptidão e calcula o melhor

 Atualize a melhor solução x_g

 Calcula o novo valor de α com sua equação

Até iterações $\geq maxIter$

Fim

Figura 27. Pseudocódigo do algoritmo *FA* básico [28].

3.4. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS PSO BINÁRIO

Em muitos casos, os problemas de otimização não são somente de tipo contínuo. Há também problemas do tipo discreto e do tipo binário, para o qual é necessário adequar as técnicas de busca em espaços contínuos. O algoritmo clássico do PSO proposto por Eberthard e Kennedy [30] foi desenvolvido pensando em espaços de tipo contínuo, onde a velocidade das partículas é determinada por uma combinação linear, como foi descrita no início deste capítulo.

Eberthard e Kennedy [30] [31] fizeram a proposta do algoritmo PSO binário, no qual as variáveis de decisão assumem dois possíveis valores 1 ou 0. Para fazer a adaptação foi preciso codificar as soluções encontradas como uma solução de tipo binária [30]. O novo algoritmo conserva a estrutura do PSO original, isto é, a velocidade continua sendo calculada da mesma maneira (vide equação (3-8)), porém utiliza-se uma função sigmóide para normalizar os valores de velocidade das partículas (vide equação (3-9)). Para um espaço de busca N dimensional a posição de cada partícula é representada da forma $\mathbf{x} = (x_1, x_2, \dots, x_N)$ e os valores possíveis de cada termo são calculados comparando-se o valor da função sigmóide com um valor aleatório r_{ij} no intervalo $[0,1]$ (vide equação (3-10)).

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1 U_{1j} (y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2 U_{2j} (y_{sj}^{(t)} - x_{ij}^{(t)}) \quad (3-8)$$

$$v_{ij}^{(t+1)} = sig(v_{ij}^{(t+1)}) = \frac{1}{1 + e^{-v_{ij}^{(t+1)}}} \quad (3-9)$$

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } r_{ij} < sig(v_{ij}(t)) \\ 0 & \text{de outra maneira} \end{cases} \quad (3-10)$$

Para este tipo de algoritmo, faz-se a mesma avaliação da qualidade das possíveis soluções em uma função custo, obtendo a aptidão de cada uma delas. Além da versão destes autores outras propostas têm sido realizadas tentando melhorar o desempenho do PSO discreto original [32] [33] [34] [35]. Neste trabalho é utilizada a versão original do algoritmo PSO binário (vide Figura 28).

Algoritmo: PSO binário básico

Entrada: $S, N, f, c_1, c_2, x_{\max}, v_{\max}, Max_{iter}$ **Saída:** posição da particular: x e seu melhor valor de aptidão: $f(x)$ **Início**

// Inicializa o enxame

Para $k=1$: S **faça****Para** $j=1$: N **faça**

$$v_{kj} = -v_{\max} + 2U[0,1]v_{\max}$$

$$x_{kj} = -x_{\max} + 2U[0,1]x_{\max}$$

Fim para**Fim para****Repita**

//Avaliação e detecção

Para $k=1$: S **faça****Se** $f(x_k) = f_{\min k}$ **então**

$$y_k = x_k$$

$$f_{\min k} = -f(x_k)$$

Fim se**Fim para**//Calcule y_s usando os S valores de aptidão $f(y_{ik})$

//Atualização usando equações

Para $k=1$: S **faça****Para** $j=1$: N **faça**

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1U_{1j}(y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2U_{2j}(y_{ij}^{(t)} - x_{ij}^{(t)})$$

$$v'_{ij}^{(t+1)} = \text{sig}(v_{ij}^{(t+1)}) = \frac{1}{1 + e^{-v_{ij}^{(t+1)}}}$$

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } r_{ij} < \text{sig}(v'_{ij}(t)) \\ 0 & \text{de outra maneira} \end{cases}$$

Fim para**Fim para****Até** $iterações = Max_{iter}$ **Fim**

Figura 28. Pseudocódigo do algoritmo PSO binário básico.

3.5. TÉCNICAS DE MELHORIA DE DESEMPENHO DOS ALGORITMOS BIOINSPIRADOS

Estes métodos são modificações realizadas por pesquisadores na procura de melhorar o desempenho dos algoritmos bioinspirados baseados em populações. Esses métodos denominados na literatura como de adição de diversidade artificial, tentam evitar o problema de convergência prematura, especialmente quando se faz uso de topologias em que os agentes do enxame seguem exclusivamente o indivíduo com melhor desempenho. Nestes casos existe uma forte tendência a encontrar soluções sub-ótimas como consequência do melhor indivíduo encontrar-se preso em um mínimo local [8].

É importante saber que, com estas modificações pode-se obter a melhoria no desempenho dos algoritmos, porém, incrementa-se também a complexidade computacional do algoritmo. Os métodos que são apresentados neste trabalho dos muitos que podem-se encontrar na literatura são: 1) o método de atrativo repulsivo; 2) o método de congregação passiva seletiva; e 3) o método de aprendizado em oposição, a seguir uma breve explicação.

3.5.1. O método atrativo repulsivo

O método atrativo repulsivo foi proposto por Riget e Vesterstrom [36]. A ideia desta técnica é evitar a convergência prematura mediante a aplicação de comportamentos atrativos e repulsivos do enxame. Nesta técnica quando as partículas convergem a uma solução sub-ótima (fase de atração) o valor da diversidade do enxame diminui e, no caso de ultrapassar de um valor limite, aplica-se um processo de dispersão do enxame (fase de repulsão) que permite aumentar a diversidade do enxame.

3.5.2. O método de congregação passiva seletiva

Baseado nos estudos realizados com grupos de animais, Parrish e Hammer [37], encontraram dois padrões de organização dos animais: agregação e congregação. A agregação acontece por efeitos físicos externos, tais como os efeitos climáticos (agregação ativa) ou pela busca dos objetivos do enxame tais como procura de alimento e espaço (agregação passiva). A congregação acontece por meio de efeitos sociais. Quando os

agentes do enxame são atraídos de forma natural através de correlação genética encontra-se enxames com divisão de tarefas entre os indivíduos (congregação ativa). Por outro lado, na congregação passiva os indivíduos colaboram inclusive quando não se tem nenhuma relação genética.

3.5.3. O método de aprendizado em oposição

Tizhoosh [38] propôs o método de aprendizado em oposição, o qual direciona a busca da melhor solução pelos algoritmos bioinspirados, na direção oposta da busca atual. O processo consiste em que em um determinado momento os agentes encontram-se juntos ao redor da melhor posição encontrada, com um decremento da diversidade. Nesse caso, esta técnica permite trocar as posições de alguns dos seus agentes às coordenadas opostas, explorando novas possibilidades no espaço de busca [39].

4. TESTES DE ESCALABILIDADE PARA A SELEÇÃO DO MELHOR ALGORITMO BIOINSPIRADO

Esta seção apresenta os resultados obtidos em testes feitos em simulação dos algoritmos de otimização bioinspirados em inteligência de enxames. Para a realização da simulação foi usada a ferramenta de desenvolvimento *MATLAB*. Esses testes realizados com os algoritmos de otimização bioinspirados foram feitos testes usando funções custo tipo *benchmark*. As funções escolhidas são não lineares com dificuldade elevada para encontrar o mínimo ou máximo global (dependendo da necessidade do problema). Por outro lado, para observar melhor o desempenho dos algoritmos nos testes foram feitas mudanças em parâmetros como: número de agentes e número de dimensões, apresentando de forma estatística os resultados obtidos (escalabilidade).

4.1. FUNÇÕES DE TESTE *BENCHMARK*

Para serem avaliados os algoritmos foram usadas funções que representam problemas de otimização considerados *benchmark* (é uma técnica usada para medir o desempenho de um sistema) [31]. Estas funções de teste têm características diferentes em termos de complexidade, de composição e interação entre as variáveis de decisão. Na literatura se encontram muitos exemplos de estas funções, porém neste trabalho foram utilizadas as funções apresentadas a seguir.

4.1.1. Função esfera

Esta função é do tipo unimodal com o mínimo global em $f(\mathbf{x}) = 0$, localizado nas coordenadas $x(i) = 0$, com $i = 1, \dots, N$. Sendo N como o número de dimensões do problema. A equação (4-1) representa a função de forma matemática e na Figura 29 pode-se observar a gráfica da função com $N=2$.

$$f(\vec{x}) = \sum_{i=1}^N x_i^2 \quad (4-1)$$

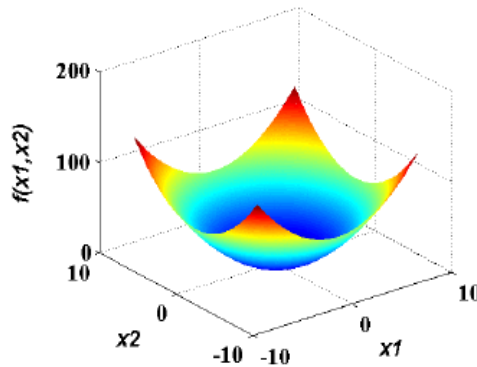


Figura 29. Função esfera

4.1.2. Função Rosenbrock

É uma função multimodal para um número de dimensões de $N \geq 2$. Esta função tem o mínimo global em $f(\mathbf{x}) = 0$, localizado nas coordenadas $x(i) = 1$ com $i = 1, \dots, N$. A equação (4-2) descreve a função de forma matemática, da mesma forma, na Figura 30 mostrasse a gráfica da função em duas dimensões ($N=2$), permitindo observar nela o mínimo o local e o mínimo global. Os mínimos da função estão em uma quase planície, possibilitando que os algoritmos fiquem facilmente estagnados no mínimo locais e que se gere em esta zona pouca diversidade das partículas.

$$f(\vec{x}) = \sum_{i=1}^{N/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \quad (4-2)$$

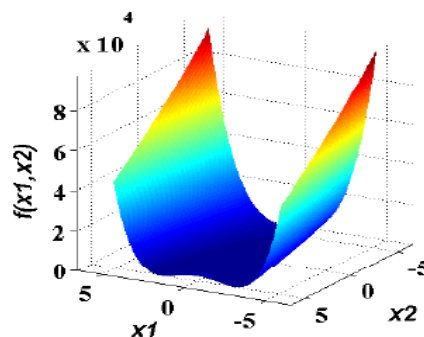


Figura 30. Função Rosenbrock.

4.1.3. Função *Rastrigin*

É uma função multimodal com um grau de complexidade maior devido à quantidade elevada de mínimos locais. Ela representa um desafio para os algoritmos de otimização por ter uma alta possibilidade convergir em um mínimo local e não chegar ao mínimo global (solução ótima). A equação (4-3) descreve esta função e a gráfica pode-se observar na Figura 31 e Figura 32.

$$f(\vec{x}) = \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (4-3)$$

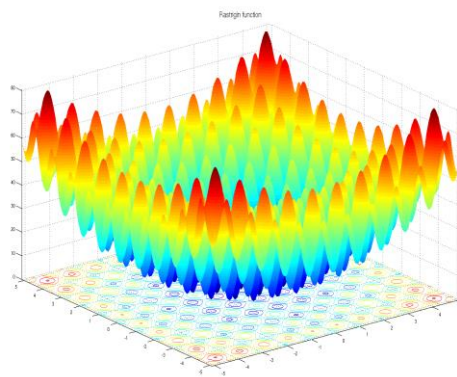


Figura 31. Função *rastrigin*

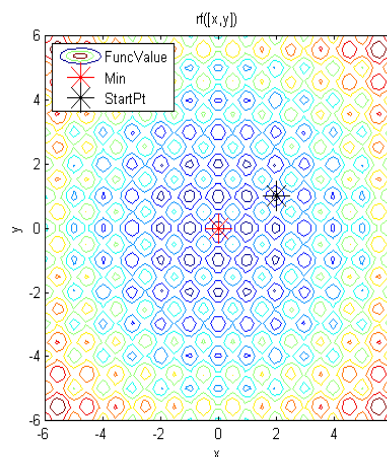


Figura 32. Gráfico de contorno da função *rastrigin*.

4.2. RESULTADOS DOS TESTES DOS ALGORITMOS DE OTIMIZAÇÃO USANDO AS FUNÇÕES BENCHMARK

Os resultados dos testes dos algoritmos de otimização são apresentados, para facilidade de entendimento, de forma estatística em tabelas que cotem a media, mediana, desvio padrão dos dados obtidos, valor mínimo e número de acertos. Esta abordagem permite fazer uma melhor análise de escalabilidade dos algoritmos quando são mudados o número de agentes e o número de dimensões [8]. Para cada algoritmo (PSO, ABC e FA) as condições iniciais e os valores de parâmetros de configuração são apresentados na Tabela 5.

Tabela 5. Parâmetros de configuração dos algoritmos

PSO		ABC		FA	
parâmetros	valores	parâmetros	valores	parâmetros	valores
Partículas	5,10,15,20	Fontes de Alimento	4,8,12,16	Vaga Lumes	5,10,15,20
dimensões	2,4,6,8	dimensões	2,4,6,8	dimensões	2,4,6,8
# iterações	2000	# iterações	2000	# iterações	2000
peso Inercia (w)	[0.8,0,1]	máx. Iterações Escoteiras	20	coeficiente de absorção (γ)	0,8
coeficiente cognitivo e social	$C1 = C2 = 2$			coeficiente de atração inicial (β_0)	0,8
velocidade máx.	[-3,3]			coeficiente de atração mín. (β_{min})	0,2
				parâmetro (m)	2
				coeficiente de aleatoriedade (α)	[1,0.001]

Os algoritmos foram executados 32 vezes para cada conjunto de parâmetros (número de partículas e número de dimensões). Para as 32 execuções as posições iniciais dos agentes foram geradas aleatoriamente. Dos resultados obtidos em cada teste foi escolhida a melhor posição encontrada pelas partículas e o respectivo valor de aptidão (valor mínimo encontrado). Com os resultados dos 32 experimentos foram calculados o valor da média, mediana, valor mínimo e desvio padrão para cada algoritmo.

4.2.1. Teste dos algoritmos PSO, ABC e FA na função esfera.

Nas seguintes tabelas podem-se observar os resultados alcançados. Nas Tabela 6, 7 e 8 apresentam-se os resultados numéricos obtidos pelos algoritmos PSO, ABC e FA, respectivamente.

Tabela 6. Resultados do algoritmo PSO na função esfera

Dados	S	5	10	15	20
	N				
Media	2	1,0083E-119	2,4239E-296	0	0
Mediana		2,2312E-183	2,17E-306	0	0
Menor valor		4,05E-236	0	0	0
Acertos		0--24	2--24	18--24	24--24
Desvio padrão		4,9398E-119	0	0	0
Media	4	1,95913E-44	6,2917E-115	5,0833E-169	9,7795E-232
Mediana		9,64E-58	1,0257E-153	1,201E-219	4,035E-240
Menor valor		1,4E-181	6,24E-198	5,37E-229	3,21E-256
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		7,94064E-44	3,0823E-114	0	0
Media	6	2,16923E-14	8,26087E-59	1,3333E-101	2,0355E-131
Mediana		8,205E-27	9,59E-80	1,8815E-124	2,4582E-161
Menor valor		1,38E-47	2,96E-101	1,34E-153	1,3E-175
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		9,75768E-14	3,96177E-58	6,532E-101	9,8339E-131
Media	8	5,14452E-13	1,29208E-42	1,5625E-57	1,45431E-88
Mediana		4,485E-16	3,25E-53	7,595E-80	3,005E-105
Menor valor		3,45E-21	3,57E-70	6,92E-103	5,13E-124
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		1,4458E-12	5,649E-42	7,65466E-57	5,50766E-88

Tabela 7. Resultados do algoritmo ABC na função esfera

Dados	S	4	8	12	16
	N				
Media	2	2,64272E-18	1,37869E-18	5,04683E-19	3,20241E-19
Mediana		1,995E-18	1,195E-18	4,07E-19	1,975E-19
Menor valor		4,47E-20	9,75E-20	3,34E-20	9,09E-21
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		2,82548E-18	1,01396E-18	4,40951E-19	3,23979E-19
Media	4	1,9485E-17	1,53917E-17	1,21575E-17	1,02779E-17
Mediana		1,89E-17	1,315E-17	1,195E-17	8,225E-18

Dados	S	4	8	12	16
	N				
Menor valor		1,09E-18	3,34E-18	2,54E-18	1,75E-18
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		1,32164E-17	9,41279E-18	5,76262E-18	5,79544E-18
Media		1,6E-08	4,35417E-17	3,46833E-17	3,54333E-17
Mediana	6	6,345E-17	4,26E-17	3,495E-17	3,635E-17
Menor valor		1,76E-17	1,95E-17	1,22E-17	1,62E-17
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		7,83837E-08	1,54301E-17	1,3379E-17	1,02848E-17
Media	8	3,582E-16	7,20167E-17	6,68542E-17	6,56292E-17
Mediana		8,065E-17	7,455E-17	7,06E-17	6,49E-17
Menor valor		2,52E-17	1,01E-17	1,79E-17	3,06E-17
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		1,269E-15	2,46565E-17	2,18714E-17	1,61934E-17

Tabela 8 resultados do algoritmo FA na função esfera

Dados	S	5	10	15	20
	N				
Media	2	8,40325E-17	4,29838E-17	3,43827E-17	2,40549E-17
Mediana		4,44E-17	3,795E-17	2,8E-17	2,24E-17
Menor valor		5,98E-18	1,25E-18	9,25E-19	2,07E-19
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		8,1766E-17	2,88393E-17	2,57888E-17	1,66496E-17
Media	4	5,46708E-15	3,0255E-15	2,73542E-15	2,67229E-15
Mediana		4,965E-15	2,715E-15	2,215E-15	2,5E-15
Menor valor		1,06E-15	2,71E-16	6,36E-16	9,5E-16
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		2,88638E-15	1,98072E-15	1,48745E-15	9,92942E-16
Media	6	2,62446E-14	1,80954E-14	1,65442E-14	1,53558E-14
Mediana		2,645E-14	1,715E-14	1,62E-14	1,57E-14
Menor valor		8,74E-15	5,509E-15	4,88E-15	4,4E-15
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		8,25456E-15	6,41474E-15	5,63508E-15	4,94218E-15
Media	8	4,89125E-14	4,2725E-14	4,01208E-14	3,46167E-14
Mediana		5,14E-14	4,59E-14	3,91E-14	3,52E-14
Menor valor		4,29E-19	1,04E-14	1,46E-14	1,82E-14
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		1,81279E-14	1,43109E-14	1,12654E-14	8,93774E-15

A partir dos resultados obtidos para a função podem ser ressaltados os seguintes pontos:

- O algoritmo PSO foi o algoritmo que consegue os melhores resultados, sendo o único que encontrou a solução ótima da função em varias oportunidades.
- O PSO foi o algoritmo que esteve mais perto de alcançar a solução ótima.
- O FA foi o algoritmo que ficou mais “longe” da solução ótima.
- Os resultados dos três algoritmos podem mostrar a potencialidade de eles como solução a problemas de otimização, pois na maioria dos testes chegam perto da solução ótima.
- Na medida em que são incrementadas o número de dimensões é aumentada a complexidade do problema e, portanto, é mais complicado para os algoritmos encontrar a solução ótima (número de agentes fixo).
- Na medida em que é mantido constante o número de dimensões e incrementasse o número de agentes dos algoritmos, conseguisse melhorar seu desempenho, alcançando melhores soluções.

4.2.2. Teste de PSO, ABC e FA para a função *Rosenbrock*

Os resultados destes algoritmos são apresentados nas tabelas da seguinte forma: Na Tabela 9 o algoritmo PSO, na Tabela 10 o algoritmo ABC e na Tabela 11 o algoritmo FA.

Tabela 9. Algoritmo PSO com a função *Rosenbrock*

Dados	S	5	10	15	20
	N				
Media	2	2,53453E-06	8,36405E-21	0	0
Mediana		4,525E-12	1,18E-28	0	0
Menor valor		2,02E-24	0	0	0
Acertos		0--24	2--24	24--24	24--24
Desvio padrão		1,24105E-05	4,09355E-20	0	0
Media		4	0,833541667	0,010547209	0,12071758
Mediana	0,145		0,000125	0,00001605	2,0045E-08
Menor valor	0,005		5,56E-09	2,47E-08	2,41E-11
Acertos	0--24		0--24	0--24	0--24
Desvio padrão	1,964514366		0,042983286	0,589855573	1,00802E-05
Media	6		1,247916667	0,109625	0,356166667
Mediana		0,395	0,045	0,02	0,008
Menor valor		0,09	0,001	0,0001	0,0002

Dados	S	5	10	15	20
	N				
Acertos	8	0--24	0--24	0--24	0--24
Desvio padrão		2,193365449	0,167108091	1,58651541	0,044915621
Media		3,402791667	0,794333333	0,228958333	0,131241667
Mediana		2,595	0,34	0,123	0,064
Menor valor		0,28	0,02	0,01	0,0035
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		3,826366434	1,192887715	0,349906507	0,165781829

Tabela 10. Algoritmo ABC na função *Rosenbrock*

Dados	S	4	8	12	16
	N				
Media	2	0,206945833	0,473805471	0,221554167	0,10645625
Mediana		0,0775	0,0945	0,0965	0,045
Menor valor		0,001	0,0000313	0,0056	0,0006
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		0,354746284	0,893029921	0,283129302	0,147527757
Media	4	7,7053125	2,103416667	1,183458333	1,437291667
Mediana		4,265	1,2415	0,48	1,1525
Menor valor		0,0035	0,057	0,043	0,11
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		8,512897711	2,94969932	1,644875441	1,246743531
Media	6	24,03375	5,629583333	3,225	1,74125
Mediana		6,485	2,905	1,005	0,815
Menor valor		0,76	0,32	0,1	0,03
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		46,34569422	6,40335135	6,799278734	2,855194066
Media	8	8,752708333	5,663583333	2,632041667	1,403891667
Mediana		4,855	2,485	1,475	0,57
Menor valor		0,003	0,05	0,134	0,037
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		10,2114389	8,118493135	2,749259513	2,204666013

Tabela 11. Algoritmo FA na função *Rosenbrock*

Dados	S	5	10	15	20
	N				
Media	2	2,83775E-05	9,47958E-06	6,1445E-06	5,36233E-06
Mediana		0,000011335	0,00000611	0,0000049	0,000004744
Menor valor		4,222E-07	9,429E-07	0,000000137	0,000000278

Dados	S	5	10	15	20
	N				
Acertos	4	0--24	0--24	0--24	0--24
Desvio padrão		3,86866E-05	8,07285E-06	4,99552E-06	3,52867E-06
Media		0,125416042	0,012159213	0,00449782	0,001236496
Mediana	4	0,00974	0,00565	0,0018285	0,00121375
Menor valor		0,001465	0,0005465	0,0002562	0,000163
Acertos		0--24	0--24	0--24	0--24
Desvio padrão	6	0,427757857	0,019956119	0,011585837	0,000713666
Media		2,865287083	0,457817375	0,329935875	0,318566042
Mediana		0,3267	0,128985	0,13864	0,07229
Menor valor	6	0,0911	0,0488	0,0219	0,01459
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		7,028717355	0,695643156	0,585569345	0,60911634
Media	8	3,45511625	1,8152	1,171940833	1,108775
Mediana		0,9414	0,524	0,3739	0,30475
Menor valor		0,2948	0,2675	0,1807	0,1437
Acertos	8	0--24	0--24	0--24	0--24
Desvio padrão		4,392216487	2,9662702	1,760889229	2,102878931

A partir dos resultados obtidos com esta função podem ser observados nos seguintes pontos:

- Em geral, a complexidade da função gera nos algoritmos um desempenho menor comparado com os resultados obtidos para a função esfera.
- O algoritmo PSO foi de novo o algoritmo com os melhores resultados, chegando mais perto da solução ótima.
- O algoritmo FA melhora seu desenvolvimento e foi melhor que o algoritmo ABC.
- De novo na medida em que são incrementadas o número de dimensões é aumentada a complexidade do problema, sendo mais difícil para os algoritmos encontrar a solução ótima (número de agentes fixo).
- Com o valor constante de dimensões e mudando o número de agentes os algoritmos conseguem melhorar seu desempenho, dado que conseguem explorar mais soluções. Entretanto, o custo computacional é incrementado.

4.2.3. Teste de PSO, ABC e FA na função *Rastrigin*

Os resultados dos testes para esta função são apresentados da seguinte forma: na Tabela 12 os resultados do algoritmo PSO, na Tabela 13 os resultados do algoritmo ABC e na Tabela 14 os resultados do algoritmo FA.

Tabela 12. Algoritmo PSO na função *Rastrigin*

Dados	S	5	10	15	20
	N				
Media	2	0,165	0	0	0
Mediana		0	0	0	0
Menor valor		0	0	0	0
Acertos		20--24	24--24	24--24	24--24
Desvio padrão		0,376886559	0	0	0
Media	4	1,2387625	0,61875	0,165	0,0825
Mediana		0,99	0,99	0	0
Menor valor		0	0	0	0
Acertos		5--24	10--24	20--24	22--24
Desvio padrão		1,284630201	0,570021453	0,376886559	0,279506553
Media	6	3,020833333	1,694583333	1,112916667	0,66
Mediana		2,98	1,98	0,99	0,99
Menor valor		0	0	0	0
Acertos		1--24	3--24	7--24	11--24
Desvio padrão		1,977734577	1,152830087	1,024073194	0,694944477
Media	8	7,171791667	3,475	2,439583333	2,149166667
Mediana		6,5265	2,98	2,98	1,98
Menor valor		1,98	0,99	0	0
Acertos		0--24	0--24	1--24	1--24
Desvio padrão		3,650988264	2,114580851	1,208162382	1,122392308

Tabela 13 algoritmo ABC na função *Rastrigin*

Dados	S	4	8	12	16
	N				
Media	2	0	0	0	0
Mediana		0	0	0	0
Menor valor		0	0	0	0
Acertos		24--24	24--24	24--24	24--24
Desvio padrão		0	0	0	0
Media	4	0,000182287	1,68926E-05	3,7579E-06	1,61838E-06

Dados	S	4	8	12	16
	N				
Mediana		1,942E-10	5,06E-10	2,1E-10	4,217E-10
Menor valor		0	0	0	0
Acertos		3--24	2--24	1--24	3--24
Desvio padrão		0,000577277	4,99929E-05	1,61465E-05	5,47864E-06
Media		0,358850756	0,065417762	0,063902707	0,035437458
Mediana	6	0,02315	0,01151	0,0167	0,008355
Menor valor		1,4548E-14	3,649E-09	0,00000383	0,0000366
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		0,561698164	0,109631664	0,141718039	0,056284111
Media		1,122975104	1,276209543	1,452670833	0,836145833
Mediana	8	0,9965	1,4075	1,4405	0,601
Menor valor		1,67E-10	0,00002902	0,01	0,0167
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		1,267023202	0,774301701	0,766892743	0,760606374

Tabela 14 algoritmo FA com a função *Rastrigin*

Dados	S	5	10	15	20
	N				
Media	2	0,006648608	0,000336673	8,68808E-05	0,000156333
Mediana		0,00044	0,000228	0,000058585	0,000107835
Menor valor		0,00001437	0,00001905	0,000003564	0,00001237
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		0,029261843	0,000278664	6,80446E-05	0,000173241
Media	4	1,273102417	0,561809958	0,313415083	0,221365575
Mediana		1,02735	0,061545	0,026095	0,012845
Menor valor		0,024158	0,005768	0,003657	0,0013329
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		0,983341929	0,658027812	0,465535855	0,409885467
Media	6	0,110958333	0,000208333	3,49542E-12	2,97542E-12
Mediana		5,64E-12	3,72E-12	3,285E-12	3,09E-12
Menor valor		1,29E-12	1,83E-12	1,31E-12	7,6E-13
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		0,284162152	0,001020621	1,14701E-12	1,12512E-12
Media	8	7,386232917	5,692125	4,911391667	3,84065625
Mediana		6,796	5,222	4,264	3,172
Menor valor		0,1226	1,178	1,267	1,2136
Acertos		0--24	0--24	0--24	0--24
Desvio padrão		4,158700193	2,700896149	2,319093112	2,157802343

O análises dos resultados obtidos dos testes dos algoritmos na função Rastrigin são:

- Apesar de a complexidade da função Rastrigin ser elevada os algoritmos conseguiram obter resultados adequados, em especial o PSO e o ABC que conseguem chegar perto da solução ótima.
- O algoritmo PSO foi de novo o algoritmo de melhor desempenho (encontrou a solução ótima da função o maior número de vezes).
- É mais difícil para os algoritmos chegar à solução ótima quando é incrementado o número de dimensões, pois aumenta a complexidade do problema.
- O incremento do número de agentes nos algoritmos melhora o desempenho deles nesta função, incrementando-se também o custo computacional.

É importante ressaltar que os resultados obtidos dão, em geral, uma visão das potencialidades dos algoritmos para resolver problemas de otimização unimodal e multimodal. Entretanto, no intuito de melhorar o desempenho é possível realizar um processo de ajuste dos parâmetros de cada algoritmo (valores de inercia, coeficiente cognitivo, entre outros), permitindo explorar as capacidades de busca local e global. Adicionalmente, o incremento do número de iterações permite que as partículas tenham mais oportunidades de convergir para uma solução. Por outro lado, poderiam ser usados os métodos para evitar a convergência prematura dos algoritmos no intuito de obter melhores resultados (vide Seção 3.5) [37] [36]. Estas últimas possibilidades podem ser objetivo de trabalhos futuros e não são involucrados nesta pesquisa.

5. IMPLEMENTAÇÃO DO ALGORITMO PSO BINÁRIO PARA O CONTROLE DE SISTEMAS DE GRUPOS DE ELEVADORES

Para fazer a alocação de chamadas nos sistemas de grupos de elevadores foi usado o algoritmo de otimização bioinspirado PSO binário na versão original de seus autores Eberthard e Kennedy [30]. Esta escolha se justifica dada a fácil implementação do PSO, assim como pela sua capacidade para resolver problemas de otimização multimodais se comparado com os resultados obtidos em testes com funções de tipo *benchmark* (apresentados no capítulo 4) deste algoritmo e os outros algoritmos que foram apresentados no capítulo 3 [7] [8]. Da mesma forma a escolha da versão binária tem relação com o problema dos elevadores nos quais, de forma geral, a função custo é não linear e de tipo discreta (número de elevadores do sistema é uma variável discreta), dependendo de parâmetros dinâmicos conforme o fluxo de passageiros [9].

Da mesma forma, a versão de tipo binário do PSO, apresenta uma adequada forma de busca da solução ótima para o problema de alocação em sistemas de grupos de elevadores. Nestes sistemas a solução para uma chamada pode ser representada como se mostra na Figura 33. Observa-se uma palavra de tipo binário na qual os elevadores que vão ser despachados para atender a chamada tem o valor de 1, e o valor de 0 para o elevador que não é despachado.

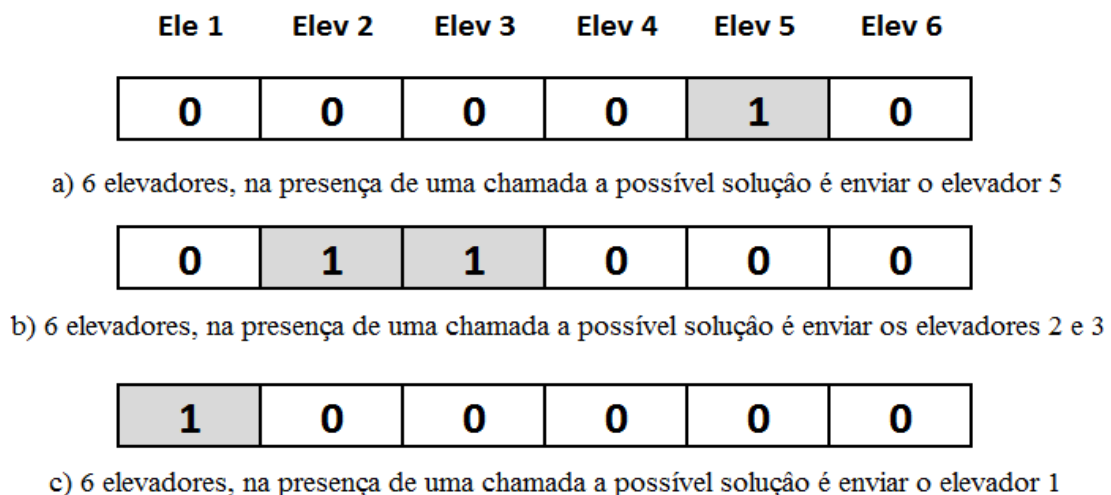


Figura 33. Exemplos de elevadores selecionados para atender uma chamada.

Pode-se pensar também que basta testar todas as possíveis combinações binárias (busca exaustiva) para encontrar a melhor solução. Contudo, esta técnica seria pouco adequada, considerando a quantidade de possíveis soluções. Entretanto, a importância do algoritmo PSO Binário na procura da solução está na forma como se geram as possíveis soluções (estratégia de busca) e como o algoritmo encontra a melhor solução usando a estratégia de inteligência coletiva. Desta forma é possível avaliar soluções em uma função custo (o elevador que cumpre com todos os requerimentos de tempo e conforto), ou chegando a uma solução que pode estar próxima da ótima, possibilitando, diminuir o custo computacional em fazer operações que não são necessárias.

Por outro lado, e depois de ter selecionado o algoritmo de otimização para fazer os testes primários do sistema de controle proposto, é importante apresentar o esquema de controle planejado de forma geral para os sistemas de grupos de elevadores. O esquema foi o resultado do estudo da base teórica do problema (metodologia *top down*). Vide Figura 34.

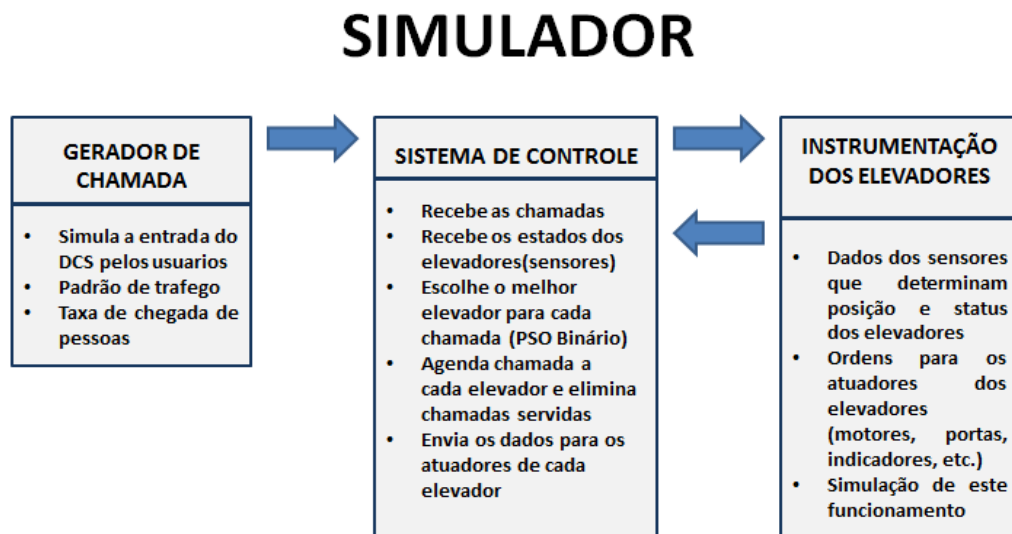


Figura 34. Esquema do simulador.

Neste esquema, Pode-se observar a estrutura do simulador desenvolvido, simulador implementado em primeira instância em Matlab, no intuito de testar e refinar a função custo que foi construída para o problema, e o funcionamento do algoritmo de otimização encarregado de fazer o controle do sistema de grupo de elevadores. Logo, o simulador foi implementado no software Arena de *Rockwell Automation*. Posteriormente, se faz uma

apresentação na Figura 35 do fluxograma em digrama de blocos do sistema controle de grupos de elevadores, e na Figura 36 apresenta-se o fluxograma do PSO binário.

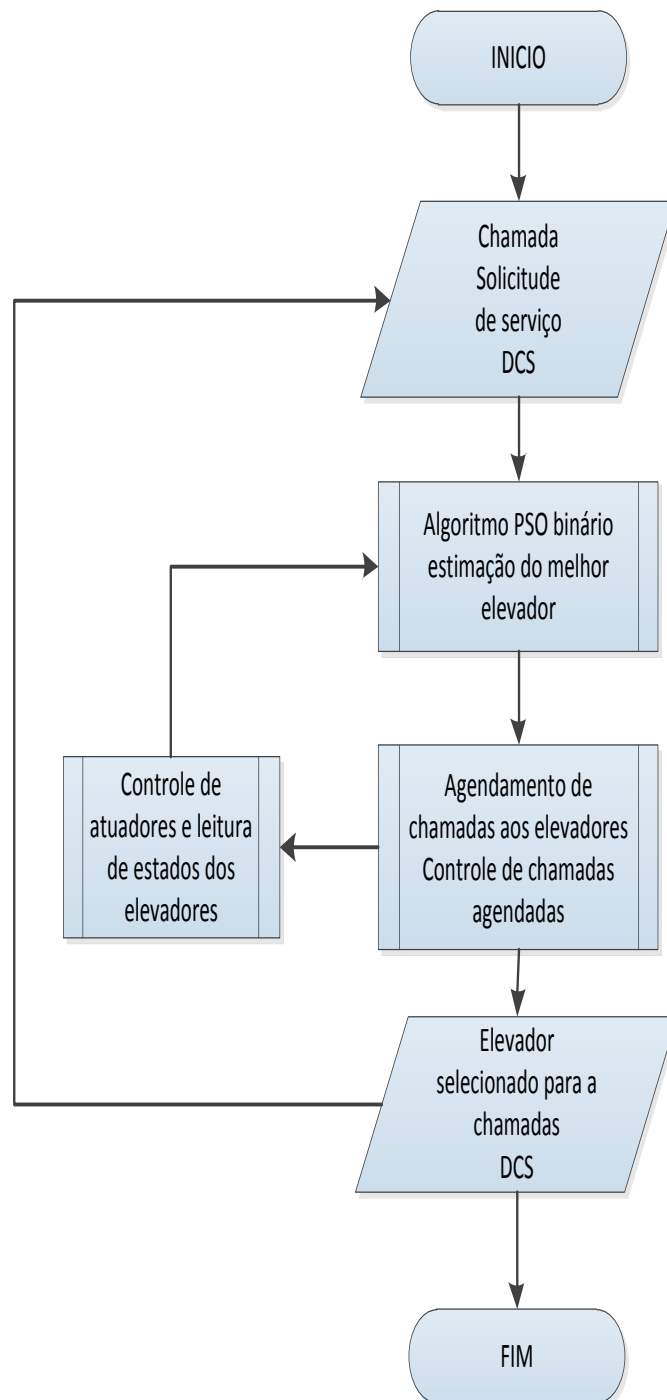


Figura 35. Esquema de controle do simulador.

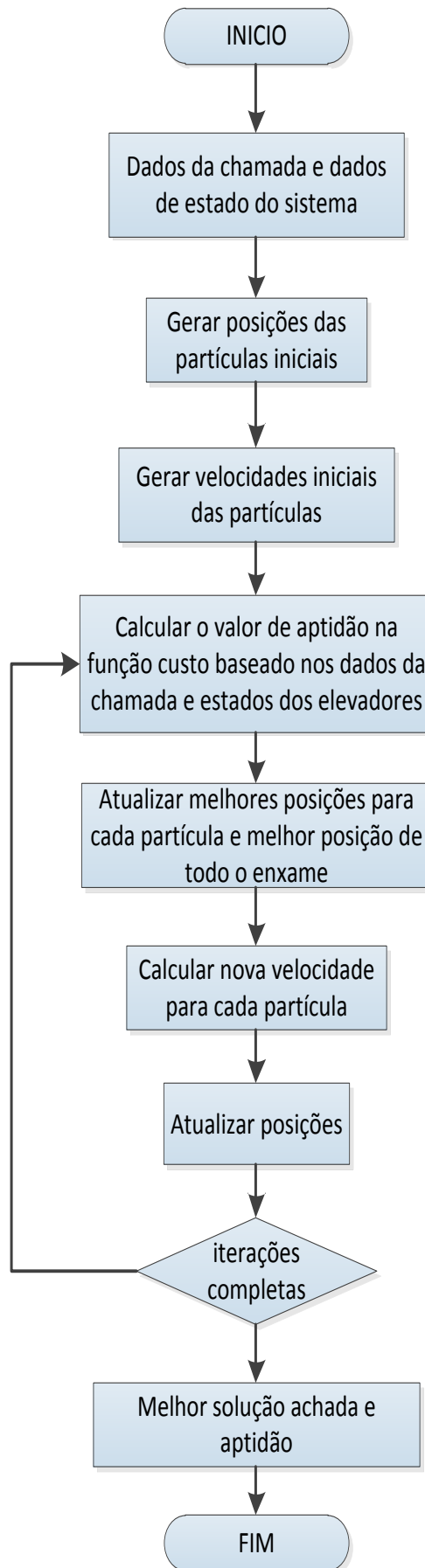


Figura 36. Esquema do PSO binário

No fluxograma apresentado na Figura 35, apresentam-se vários blocos que são parte importante dos testes, assim como do simulador final desenvolvido neste trabalho. É importante realçar que alguns blocos foram desenvolvidos de forma modular e integrados em sua totalidade no simulador industrial *Arena*. Desta forma, os testes apresentados neste trabalho fazem uso de uma abordagem dinâmica do funcionamento dos sistemas de elevadores (elevadores com movimento).

Os testes consistem em encontrar o elevador mais apropriado para chamadas feitas sucessivamente em um sistema de grupo de elevadores de um prédio comercial. Neste prédio é gerada cada uma das chamadas, a partir de qualquer andar (de acordo com o padrão de tráfego selecionado) e o programa simulador implementado no software *Arena* recebe a chamada e faz o uso do algoritmo PSO binário para encontrar o melhor elevador. O critério para escolher o elevador usa uma função custo baseada nas seguintes variáveis: (a) tempo de espera, (b) posição, (c) direção, (d) carga disponível e (e) quantidade de chamadas alocadas em cada elevador. Depois de ser feita a seleção do elevador (ou dos elevadores) para atender cada chamada, simula-se o comportamento dos elevadores (movimento, paradas, entrada e saída de pessoas, os tempos de espera dos usuários pelo elevador, os tempos de traslado dos usuários desde seu andar de origem até seu andar de destino, entre outras). Com o fim de avaliar o sistema, calculam-se os valores médios dos tempos de espera e dos tempos de voo como função do tamanho da população.

5.1. CARACTERÍSTICAS DO PRÉDIO DE ESTUDO

Neste caso de estudo, as características do prédio do tipo comercial mostram-se na Tabela 15 e na Figura 37. O prédio faz uso de *DCS's* em cada andar para que os usuários façam as chamadas. Em diferenciação com alguns prédios comerciais onde alguns andares têm prioridade de atendimento, neste caso de estudo assumiu-se que todos os andares têm a mesma prioridade de atenção. Foi escolhido o padrão de tráfego de tipo *up-peak*, como descrito no capítulo 2 para prédios de tipo comercial.

Tabela 15. Características do prédio comercial caso de estudo.

PREDIO	
Tipo de prédio	Comercial
População	460 pessoas
Número de Andares	10
Número de elevadores	3
Horas de funcionamento	7:00h até as 18:00h

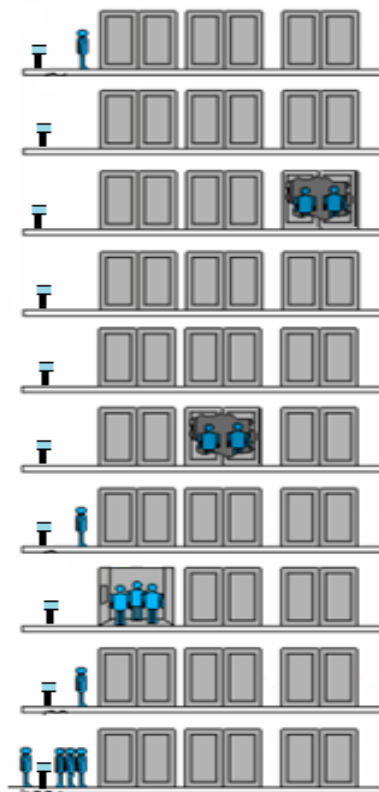


Figura 37. Prédio comercial do estudo de caso.

5.2. GERADOR DE CHAMADAS

O gerador de chamadas foi desenvolvido para simular padrões de tráfego em um prédio do tipo comercial (vide item 2.1.4) [2] [3]. Para isto foi preciso definir parâmetros como a representação da chegada de passageiros (taxa de chegada). De acordo com a literatura científica estudada [2] [3], pode-se representar a chegada de usuários como um processo do tipo *Poisson* com distribuição de probabilidade dada pela equação (5-1).

$$P = \frac{(\lambda T)^n e^{-\lambda T}}{n!}, n = 0,1,2,\dots \quad (5-1)$$

onde P é a probabilidade de chegada definida em um tempo T , com taxa de chegada λ e com n como o número de eventos (chegada de usuários). Desta forma, o intervalo de tempo entre chegadas atende a uma distribuição de probabilidade de tipo exponencial. Por outro lado, pode-se definir a taxa de chegada de pessoas fazendo uso da equação (5-2),

$$\lambda = \frac{0,01(TI)}{300} \sum_i POP_i \quad (5-2)$$

onde TI é a intensidade de tráfego (em porcentagem), POP_i é a população de cada andar e i é o número do andar, variando de 1 ao número total de andares do prédio. Assume-se aqui uma distribuição uniforme de população em todos os andares. No caso particular do prédio comercial do estudo de caso no padrão de tráfego *up-peak*, tem-se que na faixa crítica (aproximadamente 5 min com um 15% da população do prédio solicitando o serviço do sistema de grupo de elevadores). Substituindo os valores do prédio na equação (5-2), $TI = 15\%$ e $POP_i = 46$ (prédio com 10 andares e 460 pessoas), a taxa de chegada resultante é de $\lambda = 0.24$. O gerador de chamadas feito no software *Arena* apresenta, como se observa na Figura 38, a geração de uma chamada do tipo *up-peak*, e na Figura 39 a geração de uma chamada de outro padrão (*interfloor*). Finalmente, na Figura 40 pode-se observar a distribuição de chamadas acumuladas ao longo de um período de 5 minutos e a distribuição nesse tempo das chamadas contidas no pico de demanda de tráfego *up-peak*, geradas pelo programa simulador.

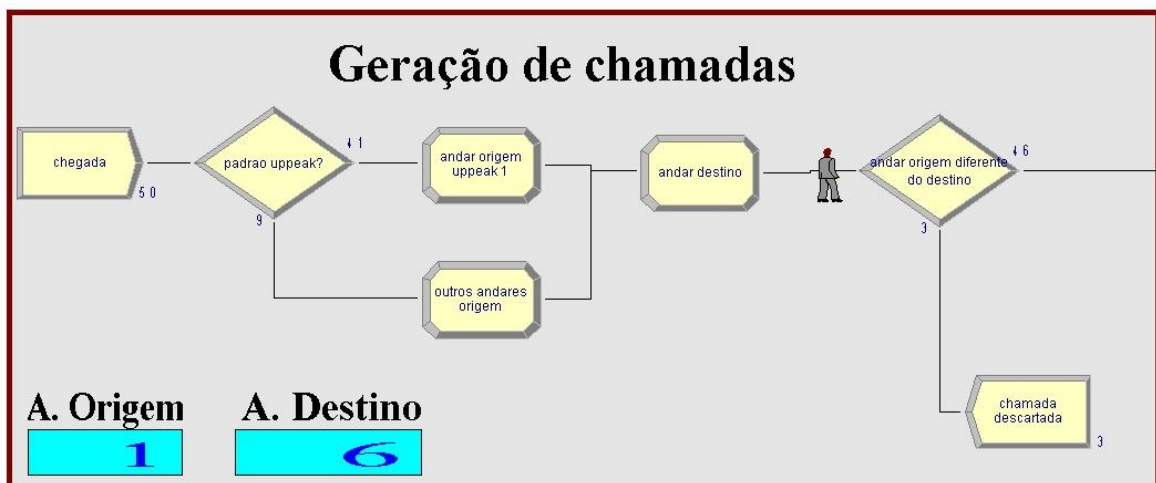


Figura 38. Gerador de chamadas no *Arena*.



Figura 39. Geração de chamadas não *up-peak*.

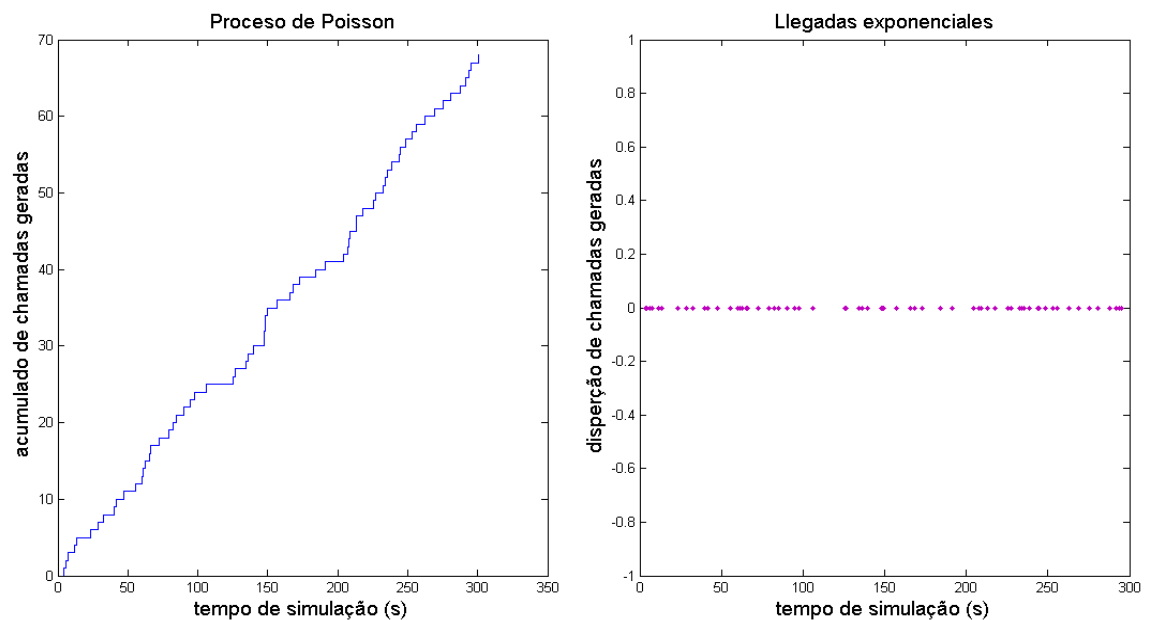


Figura 40. Gráfico de chamadas geradas pelo simulador (*up-peak*) em função do tempo de simulação.

Na Tabela 16 pode se observar os andares origem e destino das chamadas simuladas. Nela, as porcentagens das chamadas simuladas atendem o padrão de tráfego *up-peak* de um prédio comercial.

Tabela 16 chamadas geradas do tipo *up-peak*

núm. chamada	Fo(andar origem)	Fd(andar destino)	núm. chamada	Fo(andar origem)	Fd(andar destino)	núm. chamada	Fo(origem floor)	Fd(andar destino)
1	1	7	24	1	8	47	1	8
2	1	10	25	1	5	48	1	10
3	1	5	26	1	6	49	1	7
4	1	6	27	1	9	50	1	5
5	1	4	28	1	5	51	6	6
6	1	10	29	1	10	52	1	3
7	1	8	30	10	5	53	1	2
8	1	7	31	1	3	54	1	7
9	1	3	32	1	6	55	1	5
10	1	5	33	3	4	56	1	2
11	1	6	34	1	2	57	1	10
12	1	6	35	1	4	58	1	10
13	1	8	36	1	7	59	1	10
14	5	4	37	1	5	60	1	8
15	1	2	38	1	8	61	1	5
16	1	7	39	1	6	62	1	6
17	1	2	40	1	4	63	1	9
18	1	8	41	1	4	64	4	7
19	1	9	42	1	9	65	1	0
20	1	10	43	1	2	66	1	10
21	1	3	44	1	3	67	1	3
22	1	7	45	1	5	68	1	2
23	1	5	46	1	3	69	1	4

Observa-se que das 73 chamadas esperadas, de acordo com os cálculos com as equações (5-1) e (5-2) o programa gerou 69. Também pode-se observar e verificar que na maioria das chamadas o andar de origem é o térreo e que existem poucas chamadas que são realizadas de um andar de origem diferente, como é o caso do tráfego *up-peak*. Pode-se, ainda, ver que foi gerada uma chamada (número 51) com o mesmo andar origem e destino, ela foi descartada pelo o programa para não gerar erros.

5.3. FUNÇÃO CUSTO PARA AVALIAÇÃO DO PSO BINÁRIO

Para definir a função custo foram usados os fatores de tempo de espera e de tempo de voo, assim como outros parâmetros importantes que determinam as possibilidades de alocar uma chamada no elevador avaliado. Desta forma, pode-se definir a função custo com a soma de parâmetros ponderados, como indicado pela equação (5-3).

$$f(i) = k_1WT + k_2JT + k_3MWT + k_4NP + k_5NC + k_6RRI \quad (5-3)$$

Na função custo acima, os parâmetros descrevem-se da seguinte maneira:

WT: tempo de espera pelo elevador i .

JT: tempo de voo do elevador i , desde o andar origem ao andar de destino.

MWT: tempo máximo em atender uma chamada alocada no elevador i .

NP: número de pessoas no elevador i .

NC: número de chamadas alocadas no elevador i .

RRI: número de pessoas esperadas no elevador i quando chega ao andar origem.

K_i : fatores de ponderação $\sum_{i=1}^6 K_i = 1$

Os parâmetros que estão na função custo, são ponderados mediante pesos. [40]. O valor dos pesos foi escolhido de acordo com a importância de cada parâmetro e sua influência no conforto dos usuários em termos do tempo de espera ($k_1=0,4$, $k_2=0,3$, $k_3=0,1$, $k_4=0,08$, $k_5=0,04$ e $k_6=0,08$) [40], estes valores de ponderação estão inspirados na referência indicada, da mesma forma que neste trabalho foram escolhidos os pesos dos parâmetros, com os maiores valores para os parâmetros dos tempos de espera e voo. Para o caso do parâmetro NP , tem-se uma condição especial de penalidade, com a finalidade de evitar que elevadores com ocupação igual ou próxima a suas respectivas capacidades máximas tenham possibilidades de serem indicados pelo sistema. Assim, o valor de NP é incrementado para diminuir essas possibilidades.

Para o cálculo do valor dos parâmetros, foi definida uma forma de representar as variáveis do sistema de elevadores. Como por exemplo: o andar atual de cada elevador, os andares onde o elevador tem paradas agendadas, etc. para esta representação foram escolhidas de matrizes como pode-se observar na Figura 41. A primeira matriz contém os dados dos andares onde o elevador deve parar, a segunda e terceira matriz são para o número de pessoas que vão entrar e sair do elevador em cada andar. Na última matriz os estados globais.


```

RC vetor de paradas alocadas
RC=[ subindo          descendo   ]
RC=[0010011001  0000000000; elev. 1
    0000000000  0000100010; elev. 2
    0100101101  0000000000]; elev. 3

RCS vetor de pessoas que vão sair dos elevadores
RCS=[ subindo          descendo   ]
RCS=[0000020000  0000000000; elev.1
    0000000000  0000000000; elev.2
    0000100300  0000000000]; elev.3

RCB vetor de pessoas que vão sair dos elevadores
RCB=[ subindo          descendo   ]
RCB=[0000003003  0000000000; elev.1
    0000000000  0000100000; elev.2
    0000201105  0000000000]; elev.3

C_EST vetor de estado dos elevadores
C_EST= [ SOB(1) OU BAIXA , DESL(1) OU PARA , CARGA, ANDAR ATUAL ]
C_EST=[1033; elev.1
    0019; elev.2
    1142]; elev.3

```

Figura 41. Vetores de estado dos elevadores.

Com este tipo de vetores e sua manipulação se calculam os parâmetros de tempo do sistema de grupo de elevadores de cada possível solução (WT , JT , MWT), usando como base para o cálculo as equações (5-4) e (5-5). A equação (5-4) esta relacionada com os tempos de deslocamento do elevador (tempo que demora de um andar a outro, tempos de aceleração e desaceleração, tempos de voo em velocidade constante). A equação (5-5) relaciona os tempos do elevador nos andares (saída e entrada pessoas, abertura e fechamento de portas).

$$tt_i = (Df * (t_{1ms} / 2) / V_{ac}) + (Df * (t_{1ms} / 2) / V_{dac}) + (Df * (t_{3ms}) / V_{max}) \quad (5-4)$$

$$WT / JT / MWT = N_{rc} * ((T_{ip} + 2 * (ran() - 0,5)) + (T_{op} + 2 * (ran() - 0,5)) + (T_{od} + T_{dc})) + (N_{rc} + 1) * T_{op} + tt_i \quad (5-5)$$

As variáveis das equações anteriores são descritas a seguir:

tt_i = tempos de deslocamento do elevador i

Df = a distância entre os andares do prédio

T_{1ms} = seções de deslocamento quando o elevador i vai acelerando ou desacelerando

T_{3ms} = seções de deslocamento quando o elevador vai à velocidade máxima ou nominal

N_{rc} = número de paradas entre o andar origem e destino, ou número de andares entre o andar atual e o andar origem.

V_{ac} = aceleração do elevador

V_{dac} = desaceleração do elevador

V_{max} = velocidade máxima quando o elevador vai de um andar para outro (constante).

T_{ip} = tempo que tarda uma pessoa em entrar no elevador. Vai acompanhada de um termo de tempo aleatório emulando a demora na entrada.

T_{op} = tempo que tarda uma pessoa em sair do elevador. Vai acompanhada de um termo de tempo aleatório emulando a demora na saída.

T_{od} = tempo que o elevador tarda em abrir as portas

T_{dc} = tempo que o elevador tarda em fechar as portas

Para definir o valor de cada parâmetro, tem-se que fazer a análise dos possíveis cenários, quando é gerada uma chamada, os quais dependem da posição do elevador que vai ser avaliado, das características das paradas (para entrada de pessoas ou para saída de pessoas), direção do elevador, direção da chamada e se o elevador está parado ou em movimento. Desta forma, podem-se apresentar diferentes cenários. Assim:

- Cenário 1: elevador esta na mesma direção da chamada gerada, o andar de origem da chamada esta no caminho do elevador (andar origem > andar atual quando o elevador vai subindo, e andar origem < andar atual quando o elevador vai descendo)
- Cenário 2 : elevador esta na mesma direção da chamada gerada, mais o andar de origem da chamada não esta no caminho do elevador (andar origem < andar atual quando o elevador vai subindo, e andar origem > andar atual quando o elevador vai descendo)
- Cenário 3: a direção e a chamada têm direções opostas.
- Cenário 4: o elevador esta no mesmo andar de origem da chamada, com a mesma direção, e com o sim possibilidade de servi-la .

Para entender melhor cada cenário e o cálculo de cada parâmetro da função custo, apresentam-se os seguintes exemplos.

Exemplo 1: Na Figura 42 (a) o elevador esta subindo, com paradas agendadas de cor azul nos andares 1 (térreo), 7 e 10. O andar atual do elevador é $Fa = 1$. É criada uma chamada no andar de origem $Fo = 3$ e o andar de destino $Fd = 8$ na mesma direção de movimento do elevador.

Para o cálculo de WT , tem-se o tempo que o elevador demora para viajar Fa até Fo , considerando-se os tempos associados às paradas, tempos de aceleração e desaceleração, abertura e fechamento de portas, tempos de ingresso e/ou de saída de pessoas.

Para o cálculo de JT , considera-se o tempo de viagem do elevador entre Fo e Fd , considerando as paradas, tempos de aceleração e desaceleração, abertura e fechamento de portas, tempos de ingresso e/ou de saída de pessoas.

O máximo tempo de espera (MWT) é o tempo de espera dos usuários na última chamada a ser atendida, considerando as paradas intermediárias, os tempos de aceleração e de desaceleração, abertura e fechamento de portas, tempos de ingresso e/ou de saída de pessoas.

Para o cálculo de NC , NP e RRi são usadas matrizes com informação do número de paradas (de entrada e saída de pessoas) e do número de pessoas aproximado que sobe e desce em cada parada, assim como o vetor de estados que contém a carga atual do elevador, o andar atual e a direção atual de movimento.

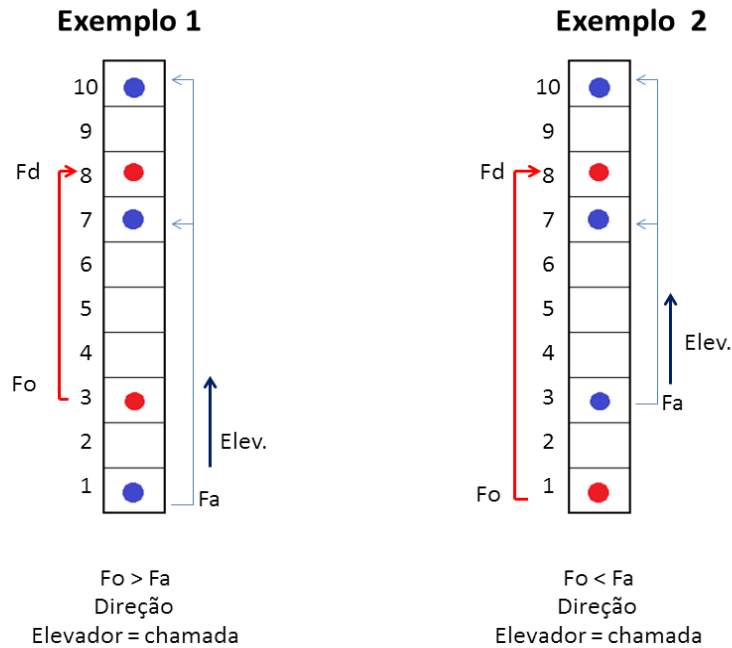


Figura 42. a) cenário do exemplo 1, b) cenário do exemplo 2

Exemplo 2: Na Figura 42 b) se tem o elevador subindo, com paradas agendadas de cor azul nos andares 3, 7 e 10. O andar atual do elevador $Fa = 3$. É criada uma chamada no andar de origem $Fo = 1$ com andar de destino $Fd = 8$ na mesma direção de movimento do elevador. Neste caso, o andar de origem é menor do que o andar atual do elevador, portanto, o elevador tem que atender às chamadas já alocadas na mesma direção de seu movimento e depois retornar ao andar $Fo = 1$ para atender esta nova chamada.

Para o cálculo de WT , tem-se o tempo que demora o elevador em atender às chamadas alocadas em cor azul na direção de subida desde Fa , somado ao tempo necessário ao deslocamento na direção contrária (descendo) até o andar de origem Fo (considerando todas as possíveis paradas intermediárias, os tempos de aceleração e de desaceleração, abertura e fechamento de portas e tempos de ingresso e/ou saída de pessoas).

Para o cálculo de JT , é considerado o tempo que o elevador demora para chegar de Fo até Fd (considerando as possíveis paradas intermediárias, tempos de aceleração e de desaceleração, abertura e fechamento de portas e tempos de ingresso e/ou saída de pessoas).

O máximo tempo de espera (MWT) é calculado do mesmo jeito do exemplo 1. Da mesma forma que nesse exemplo são calculados os valores de NC , NP e RRi utilizando as matrizes.

Exemplo 3: Na Figura 43 (a) mostra o elevador subindo, com paradas agendadas de cor azul nos andares 3, 7 e 10. O andar atual do elevador $Fa = 3$. É criada uma chamada no andar de origem $Fo = 8$ e o andar de destino $Fd = 1$ na direção contrária de movimento do elevador.

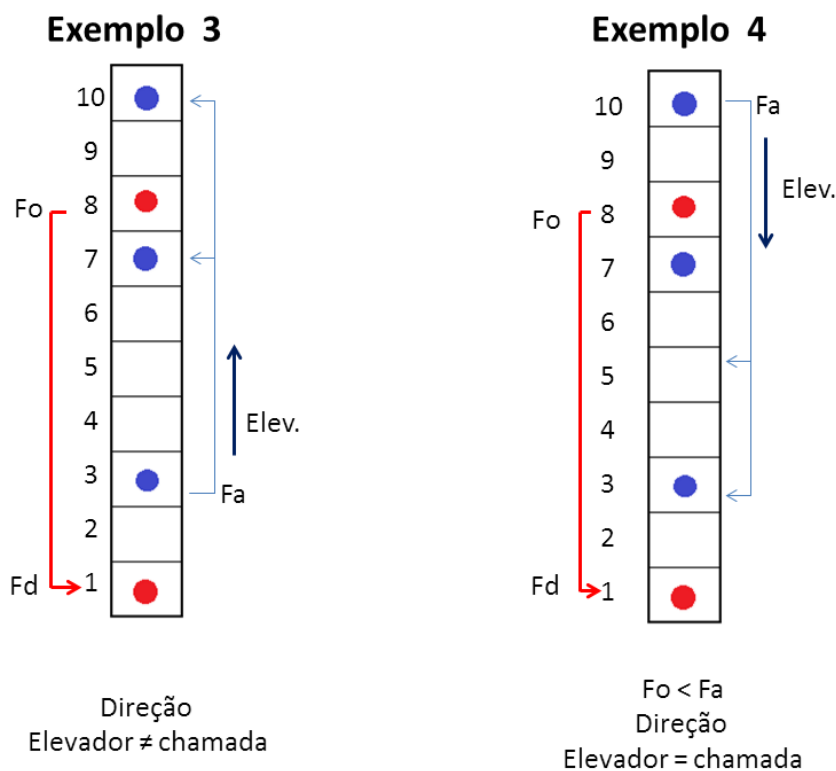


Figura 43. a) cenário do exemplo 3, b) cenário do exemplo 4

Para o cálculo WT , é o tempo em subir de Fa até o último andar a ser atendido na sua direção, atendendo todas as chamadas na cor azul, logo, é somado o tempo que tarda em descer até o andar de origem Fo , considerando-se as possíveis paradas intermediárias, tempos de aceleração e de desaceleração, abertura e fechamento de portas, tempo de ingresso e/ou saída de pessoas.

Para o cálculo de JT e MWT se faz do mesmo jeito do exemplo 1. Da mesma forma que nesse exemplo são calculados os valores de NC , NP e RRi utilizando as matrizes.

Os exemplos 4,5 e 6 das: Figura 43 (b), Figura 44 (a) e da Figura 44 (b), são analisadas como nos exemplos 1, 2, 3, mudando só a direção do elevador.

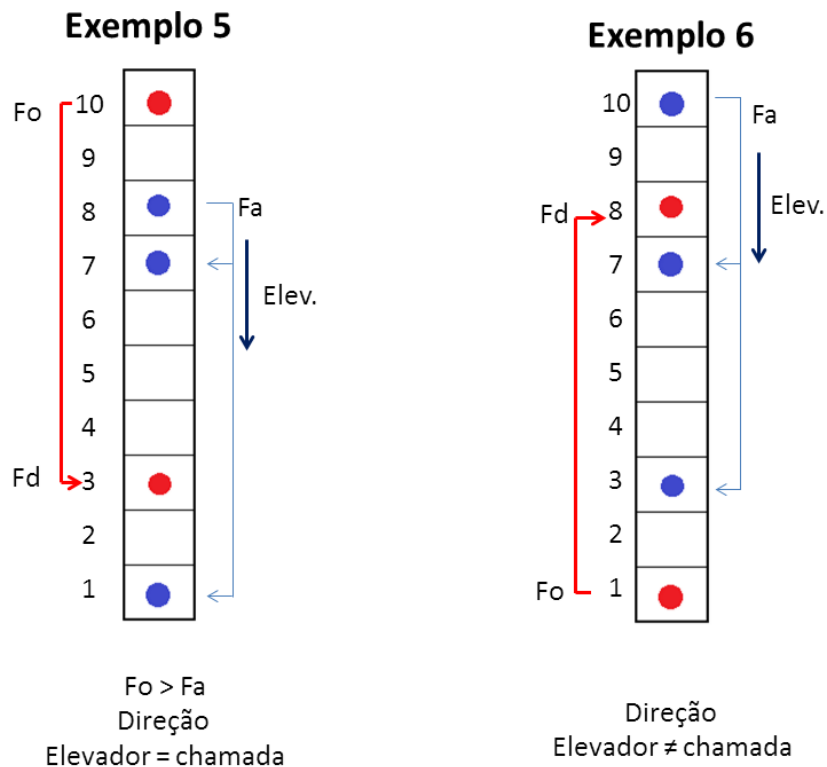


Figura 44. a) cenário do exemplo 5, b) cenário do exemplo 6.

Da mesma forma, pode-se considerar outros exemplos dentro dos cenários apresentados. Quando o andar atual é igual ao andar de origem de uma nova chamada ($Fa = Fo$), com a mesma direção de movimento do elevador, neste caso, se o elevador está em movimento, a análise é feita como no exemplo 2 ou 5. Por outro lado, se o elevador está parado, a análise é feita com o tempo de espera $WT = 0$ e para o cálculo de JT , MWT , NC , NP e RRi se procede da mesma maneira que nos exemplos anteriores.

Depois de avaliar cada possível solução por o algoritmo PSO binário, é escolhido o melhor elevador (solução com melhor aptidão) para atender à chamada que está sendo analisada.

Depois devem ser atualizados os parâmetros das chamadas agendadas para os elevadores, o número de pessoas a entrar e a sair do elevador e a posição atual. Desta forma, os elevadores podem ter controle sobre as chamadas alocadas.

A Tabela 17 apresenta os parâmetros de ajuste do PSO binário para a realização dos testes. As dimensões do algoritmo foram escolhidas para estes testes baseado no número de elevadores do prédio (3), o número de partículas e outros parâmetros foram escolhidos de acordo com o desempenho do algoritmo PSO nos testes desenvolvidos e realizados em *Matlab* (vide apêndice A).

Tabela 17. Parâmetros do PSO binário

PSO BINARIO	
parâmetros	valores
Partículas	10
dimensões	3
# iterações	50
peso Inercia (w)	[0,8,0,1]
coeficiente cognitivo e social	$C1 = C2 = 2$
velocidade máx.	[-3,3]

O controlador de alocação de chamadas com o algoritmo PSO binário foi implementado em VBA, como foi expresso na seção 5.1. No simulador desenvolvido foi representado por um bloco (contém o pseudocódigo do algoritmo), o qual tem como parâmetros de entrada os dados da chamada realizada por um usuário, os parâmetros atuais de posição, de capacidades, direção e paradas dos elevadores. Como saída desse bloco PSO produz o número do elevador que vai a servir a chamada analisada, também é realizada a atualização da agenda de paradas, de capacidade, de entrada e saída dos usuários do elevador escolhido. Na Figura 45, pode-se observar copia das telas do software *Arena* com o gerador de chamadas e com o bloco PSO binário. A Figura 46 mostra uma copia da tela do *Visual Basic*[®] com o pseudocódigo (Bloco PSO binário) explicitado. [14].

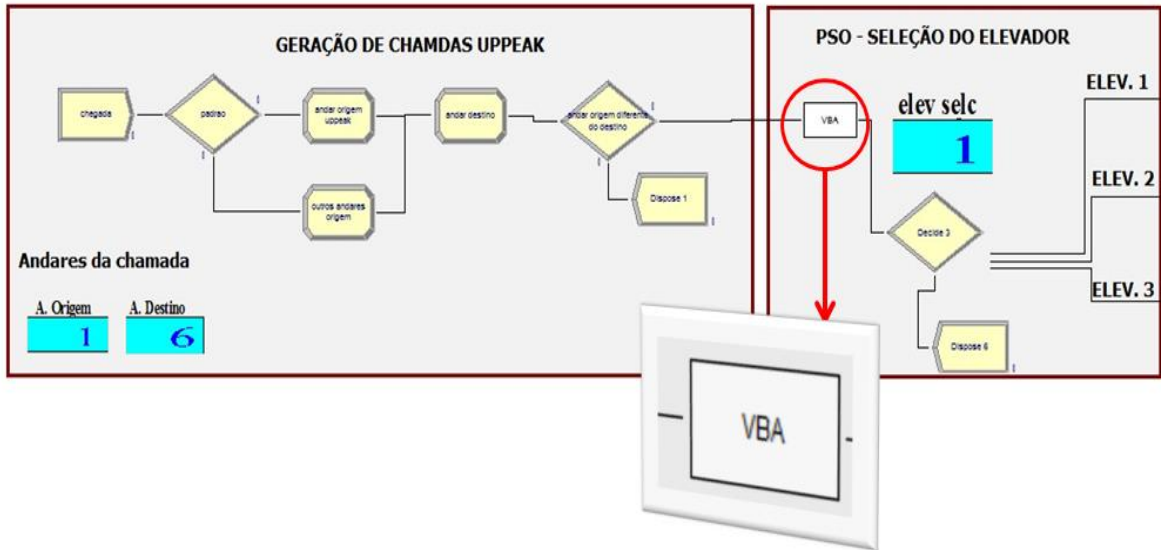


Figura 45. Bloco VBA do BPSO

The screenshot shows the Microsoft Visual Basic code editor with the following VBA code for "VBA_Block_1":

```

f_ind(i) = fx(i)
End If
Next i
'melhor posição das partículas global
'(bestfitness(k), p)= min(f_ind)
' geram as proximas posicoes das partículas

For j = 1 To N
For i = 1 To Ss

r1 = Rnd
r2 = Rnd
v(i, j) = w(k) * v(i, j) + c1 * r1 * (y(i, j) - x(i, j)) + c2 * r2 * (ys(j) - x(i, j))

If Abs(v(i, j)) > max_v Then
    If v(i, j) > 0 Then
        v(i, j) = max_v
    Else
        v(i, j) = -max_v
    End
End
sig = 1 / (1 + Exp(-v(i, j)))
r = Rnd
If sig > r Then
    x(i, j) = 1
Else
    x(i, j) = 0
End
Next i
Next j

k = k + 1

```

Pseudocódigo BPSO

Figura 46. Pseudocódigo do BPSO em VBA.

Para obter as variáveis de entrada ao controlador foi simulado o movimento de cada elevador, baseado nas paradas alocadas, simulando sistemas de filas para representar os tempos de espera fora do elevador e os tempos dentro do elevador. Na Figura 47 apresenta-

se a simulação do movimento dos três elevadores do caso (subindo e baixando), na Figura 48 pode-se observar a simulação de espera, entrada e saída de usuários no elevador. É importante entender que pelo ambiente do software *Arena*, em muitos casos não é possível obter boas imagens dos sistemas simulados (resolução).

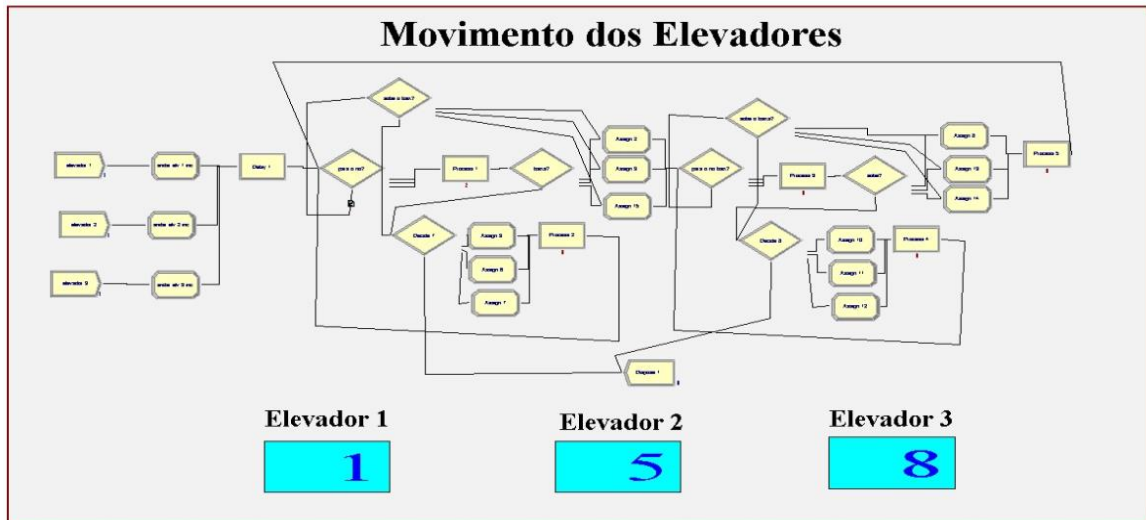


Figura 47. Simulação do movimento dos elevadores.

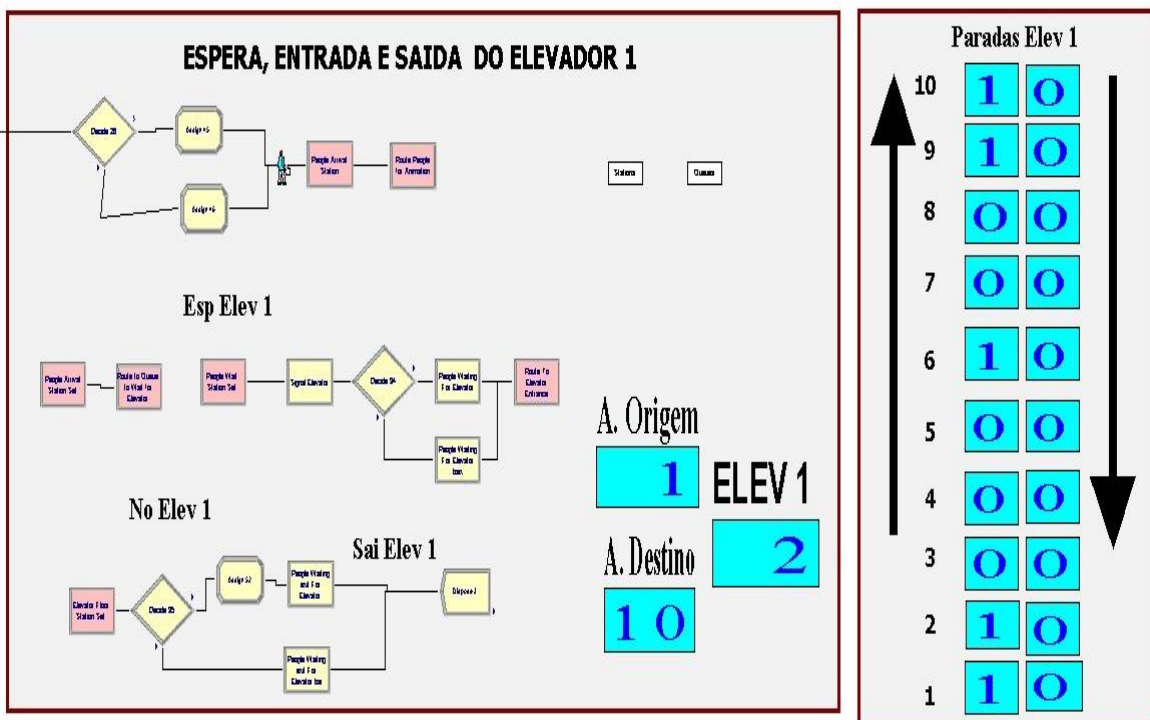


Figura 48. Simulação de usuários no elevador 1.

5.4. RESULTADOS EXPERIMENTAIS DO ALGORITMO DE CONTROLE COM O PSO DE TIPO BINARIO

Em uma primeira instancia o simulador foi desenvolvido em Matlab, sendo testado com chamadas do tipo *up-peak*, no intuito de analisar o desempenho do controlador proposto e a função custo desenvolvida. Os testes feitos foram denominados estáticos, isto é, simulando diferentes condições iniciais do sistema de elevadores, gerando para cada condição uma soa chamada, esta chamada é analisada e para ela é analisado o funcionamento do controlador (seleção do elevador ou elevadores encarregado de servi-la). Assim, desta forma foi sendo ajustado o controlador e a função custo desenvolvida em este trabalho de pesquisa.

Foram feitos quinze conjuntos de condições iniciais (simulando paradas, direção do elevador, capacidade disponível, diferente andar atual, etc.) para os três elevadores, para cada conjunto de condições foram analisadas 6 chamadas diferentes (6 chamadas por cada condição inicial dá um total de 90 chamadas analisadas, sendo um número de testes representativo). Para os três primeiros testes conjuntos de condições iniciais tem dois dos três elevadores com muitas chamadas alocadas, muitas paradas previstas e com capacidade de carga limitada quando comparados ao terceiro elevador, o qual tem só uma chamada alocada e a capacidade de carga é quase total. A ideia primaria para realizar estas provas é ter uma solução fácil de encontrar através de uma busca simples. No intuito de testar o desempenho do algoritmo, em cada um dos três testes escolheu-se um elevador diferente para ter as melhores características. Para os conjuntos de condições iniciais restantes os valores de paradas, direção e capacidade são especificadas de modo a não se ter uma solução fácil a primeira vista. Os resultados obtidos pelo algoritmo PSO binário proposto são apresentados no apêndice A. No apêndice B encontra-se uma parte do pseudocódigo desenvolvido em Matlab.

No decorrer dos testes, analisou-se a seleção do elevador solução pelo controlador, observando se apresentava compatibilidade com a lógica de controle pretendida, buscando as melhores características para atender à chamada gerada. Neste processo, por exemplo, foi melhorada a equação (5-4) pra o cálculo de tt_i , nela foram incluídos os tempos decorrentes de aceleração, desaceleração e viagem em velocidade máxima. Desta maneira, o número de paradas intermediárias impacta de forma proporcional no cálculo dos tempos

de espera e de voo. Com base nos resultados obtidos dos 15 testes com algoritmo PSO binário, os seguintes comentários podem ser destacados:

- Nos primeiros três testes onde a melhor solução poderia ser encontrada através de um processo de busca exaustiva, o algoritmo PSO binário conseguiu encontrá-la. Observe-se que para os testes 1, 2 e 3 foram escolhidos os elevadores 2, 1 e 3, respectivamente, como esperado através do processo de busca, considerando que são as soluções com menor tempo de espera, viajam na mesma direção da chamada e possuem a maior capacidade de carga.
- No teste 7, têm-se 5 chamadas com características do tipo *up-peak* para as quais o algoritmo PSO binário encontrou o elevador 2 como solução adequada, porém, quando se gerou uma chamada de outro tipo de padrão de tráfego(*interfloor*) o algoritmo mudou eficientemente de solução, escolhendo o elevador 3.
- No teste 13, apresentam-se três situações: a primeira para as chamadas de tipo *up-peak* para a qual encontrou-se, na maioria dos casos, a solução do elevador 2, mas em uma das chamadas a solução mudou para o elevador 3. As outras situações consideram a mudança de padrão de tráfego das chamadas. No caso da chamada com $Fo=6$ e $Fd=10$ (padrão *interfloor* subindo) foi selecionado o elevador 3. Por outro lado, no caso da chamada com $Fo=9$ e $Fd=6$ (padrão *interfloor* descendo), foi escolhido o elevador 2 dado que este elevador vai na mesma direção da chamada e encontra-se no mesmo andar de origem da chamada.
- Observa-se nos testes 8, 14 e 15 que os elevadores têm cargas similares, motivo pelo qual o algoritmo PSO binário encontrou soluções variadas, isto é, os três elevadores têm quase as mesmas possibilidades de serem a melhor solução.
- Nos outros testes é possível observar que foi encontrada sempre a mesma solução, pois a maioria das chamadas tem as mesmas características (de tipo *up-peak*) e, mesmo que mude o padrão de tráfego, o melhor elevador continua a ser o mesmo.

- Nos 15 testes observou-se que os valores de aptidão têm consistência para as diferentes tentativas (valores parecidos para chamadas com características similares).

Depois de realizados esses testes, apresentam-se os resultados de testes obtidos nas simulações no software de processos industriais *Arena* no senso dinâmico, isto é, simulando chamadas sucessivas em períodos de tempo de 5 min em tráfego *up-peak*. de acordo com o comportamento deste padrão, a porcentagem de população simulada, foi variando da seguinte forma: 2%, 4%, 6%, 8%, 10%, 12%, 14 % e 16% (o último valor é considerado a faixa crítica deste padrão). Para esses testes, foi tomado o tempo médio de espera para o atendimento pelo elevador selecionado pelo algoritmo de otimização, junto com o tempo médio de voo para cada chamada. Vide a Tabela 18.

Tabela 18. Resultados tempos com diferentes porcentagens de população.

Resultados obtidos				
% pop.	Pessoas esperadas	Tempo de espera (s)	Tempo de voo (s)	Tempo de destino (s)
2%	12	15,14	14,35	29,49
4%	21	24,6	20,86	45,46
6%	27	23,42	19,9	43,32
8%	36	29,35	18,36	47,71
10%	48	33,69	22,4	56,09
12%	54	35,89	26,07	61,96
14%	64	37,07	24,34	61,41
16%	73	43,37	25,5	68,87

Na Figura 49 e na Figura 50, pode-se observar o comportamento e a tendência do tempo médio de espera pelo elevador selecionado e do tempo de voo do andar de origem até o andar de destino, tudo com respeito às porcentagens de população atendidas. É importante lembrar que 90% das chamadas atendidas pelos elevadores têm como andar de origem o número 1 (padrão de tráfego *up-peak*). Também, é relevante entender que para os cálculos destes valores foram feitos diferentes testes (10 para cada porcentagem, lembrando que por cada teste têm-se dados de chamadas baseados nas porcentagens e nos valores esperados

mostrados Tabela 18, sendo assim, um valor de dados considerável e representativo). Por outro lado, para evitar os períodos transitórios no sistema durante as simulações [2] [41], foram simulados faixas de tempo de 6 min em cada teste, no qual os primeiros 30 seg. e os últimos 30 seg. se descartarão.

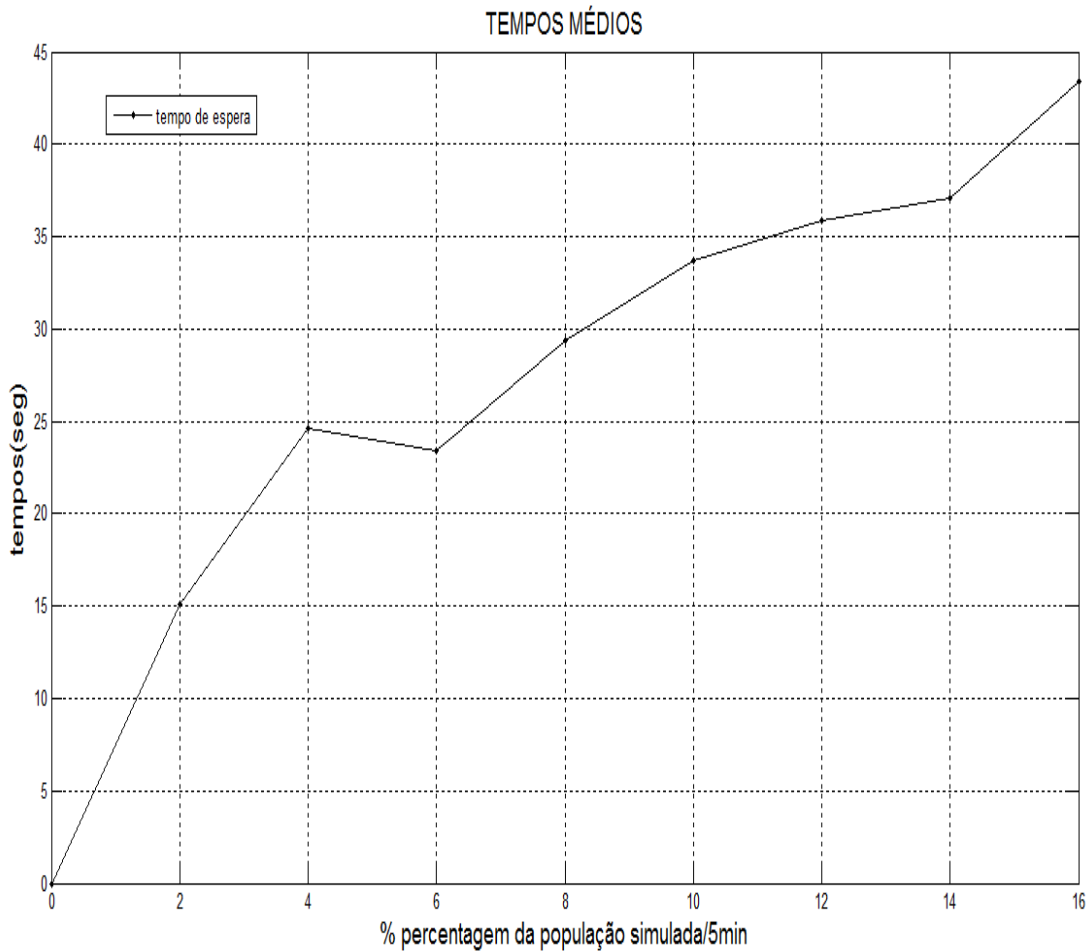


Figura 49. Tempos médios de espera pelo elevador escolhido.

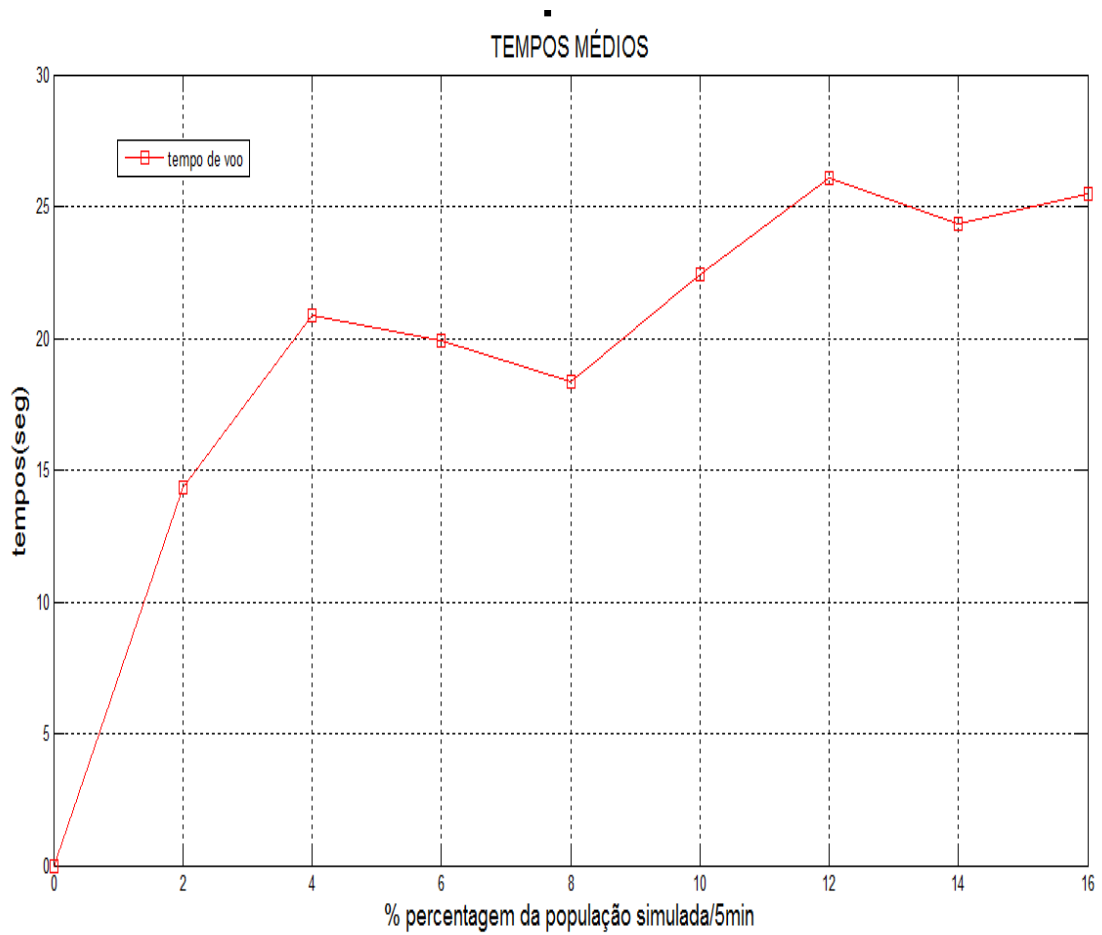


Figura 50. Tempo médio de voo.

Realizando a soma do tempo que o usuário do sistema tem que esperar pelo elevador e o tempo de traslado até seu destino, tem-se o tempo total do serviço (chamado tempo de destino na Tabela 18). Na Figura 51, pode-se observar este tempo e sua tendência com respeito ao percentual da população atendida e simulada, assim como os tempos de espera e de voo.

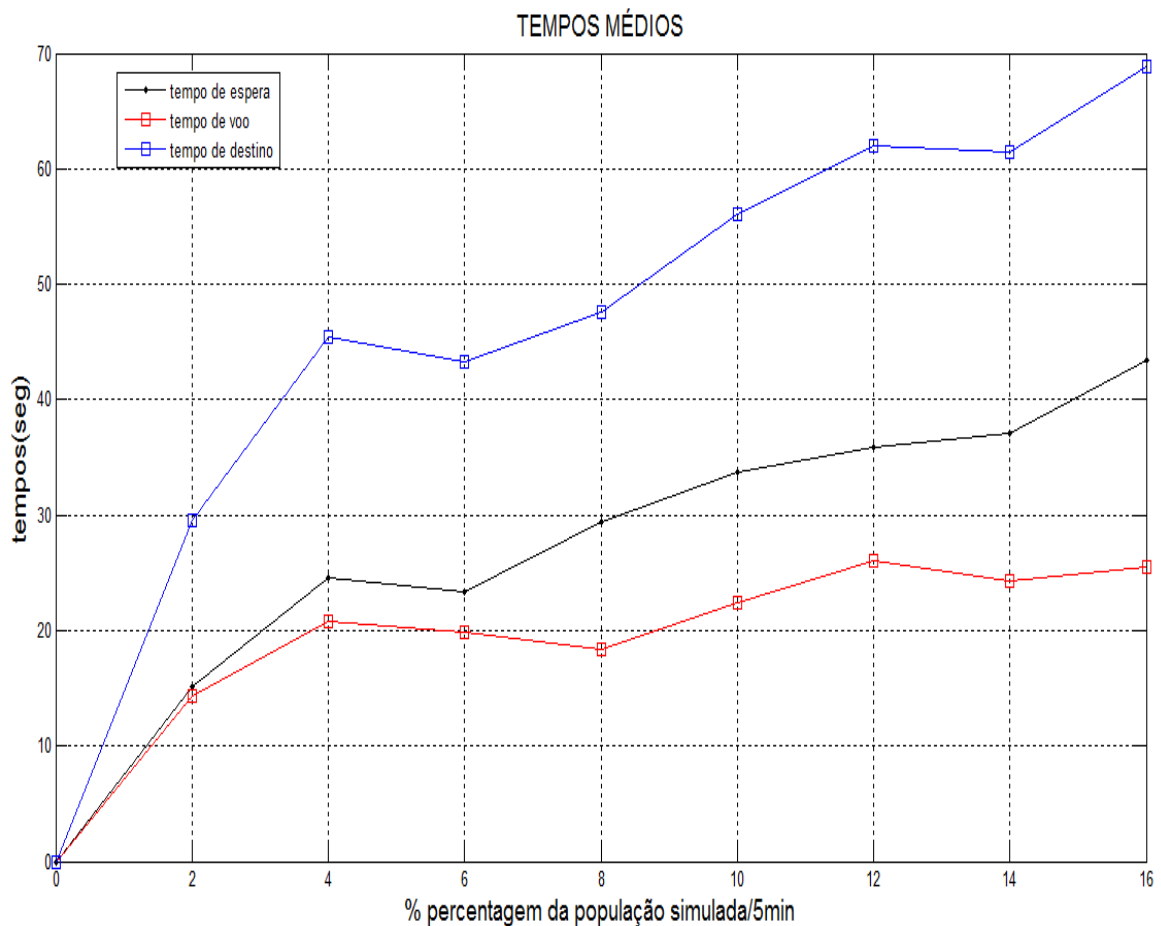


Figura 51. Tempos médios do sistema de elevadores

Outro parâmetro fundamental é a quantidade de paradas que os elevadores realizam em um período de tempo de 5 min. Este valor pode ser importante quando se pensa em realizar um controle multi-objetivo, em que além de tentar diminuir os tempos de serviço, pode-se pensar na diminuição do consumo de potência (no intuito de otimizar o consumo e o custo econômico que este item significa). Deve-se lembrar que o número de paradas está relacionado com o número de vezes que os motores que movimentam os elevadores são acelerados e desacelerados, quando se tem maior consumo de corrente e, conseqüentemente, elevado consumo de potência. Observa-se, entretanto, que esse tema não foi abordado neste trabalho, sendo sugerido como trabalho futuro. O parâmetro de paradas é utilizado aqui somente para possibilitar a realização de análises de desempenho. A Figura 52 ilustra o comportamento do sistema de elevadores em relação ao número de paradas em função do percentual da população atendida.

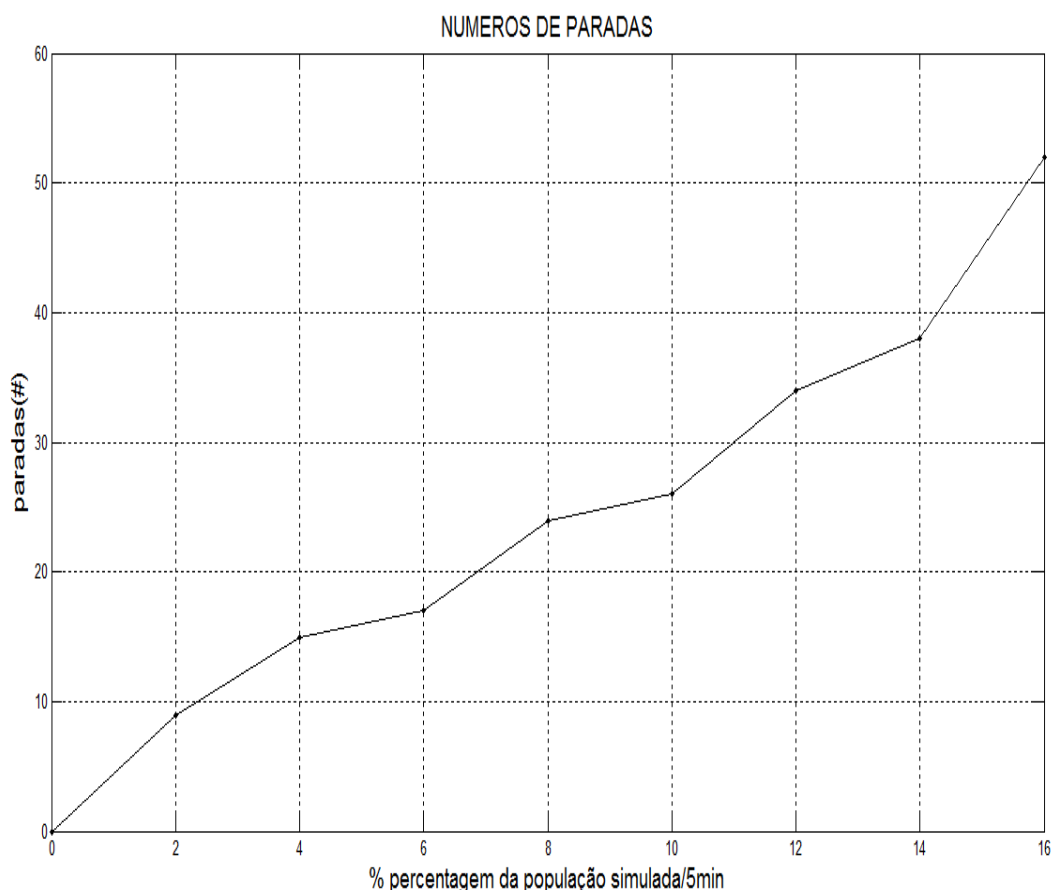


Figura 52. Número de paradas médio dos elevadores.

5.5. ANALISES DE RESULTADOS E COMPARAÇÃO COM OUTROS AUTORES

A importância e a relevância dos resultados obtidos nas simulações podem ser observadas mais facilmente quando são comparados com os resultados obtidos por outros autores que abordam o mesmo problema na literatura científica. Pontos a serem observados incluem a forma geral das curvas obtidas e tendências, em uma abordagem qualitativa, uma vez que uma comparação em termos absolutos se faz difícil, devido às diferenças entre simuladores utilizados, assim como aos parâmetros utilizados na definição dos problemas simulados, cujas informações não estão completas na literatura. Com a finalidade de realizar uma avaliação qualitativa, faz-se a comparação dos resultados deste trabalho com os relatados nas referências [10] e [41], e entre outros [6] [9] [17] [19] [41] [42].

5.5.1 – Comparação com os resultados de Patiño [10]

A primeira comparação é feita com os resultados obtidos no trabalho de Patiño [10], por ter antecedido a presente pesquisa na mesma instituição e por ter fornecido inspiração para a utilização de algoritmos de otimização como possível solução ao problema do controle de grupo de elevadores. Desta maneira, podem-se observar na Figura 53 as curvas de tempos de espera e de destino obtidas a partir das simulações realizadas neste e naquele trabalho. Os resultados obtidos por Patiño são melhores nos percentuais mais baixos de população atendida, enquanto que os resultados da presente pesquisa mostram-se melhores nos percentuais maiores, com a diferença que o edifício simulado neste estudo tem 2 andares a mais e população maior em 30 pessoas. Há também outras diferenças na dinâmica do movimento dos elevadores. Observa-se que as tendências das curvas são semelhantes e que o controlador proposto neste trabalho responde de uma maneira similar ao controlador de lógica nebulosa (*Fuzzy*) proposto por Patiño, melhorando, entretanto, o desempenho nos casos de maior demanda.

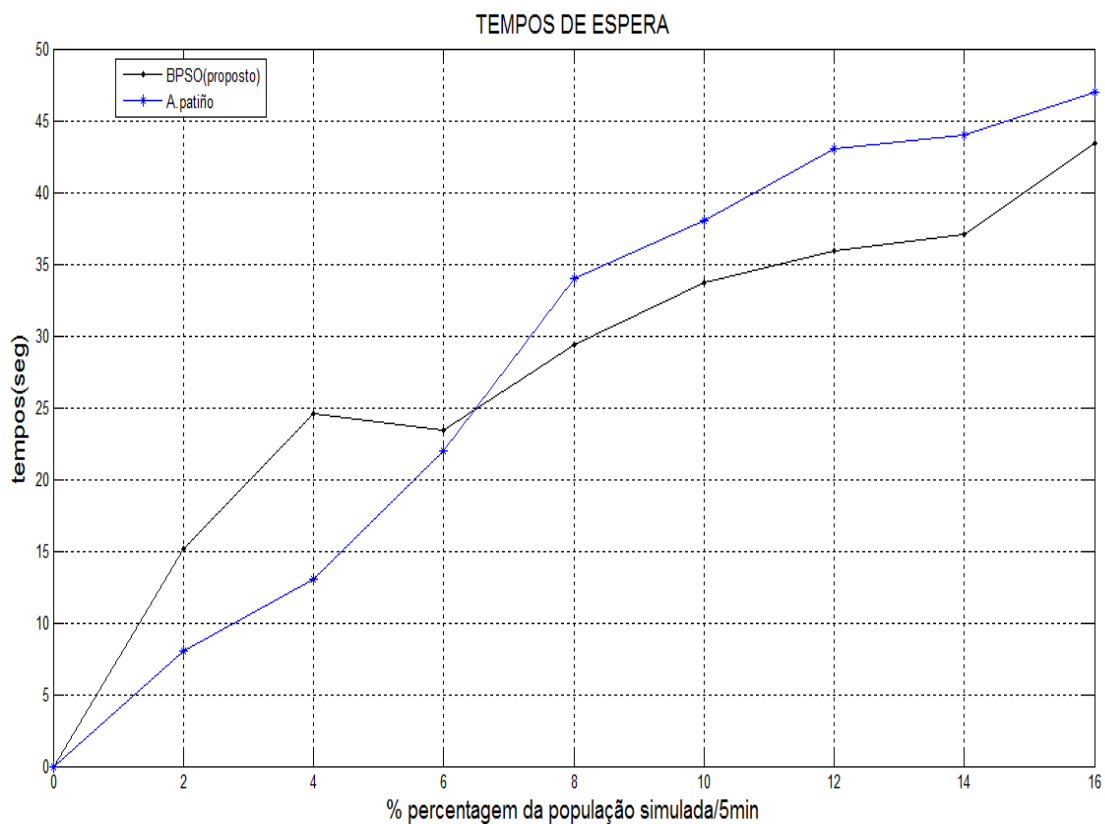


Figura 53. Comparação dos tempos médios de espera com os resultados de Patino [10],

Comparando, ainda, os tempos de destino (tempos de espera mais os tempos de voo), pode se observar a mesma tendência em ambos os trabalhos (vide a Figura 54). Como era de se esperar, o antecedente e o atual têm comportamentos similares.

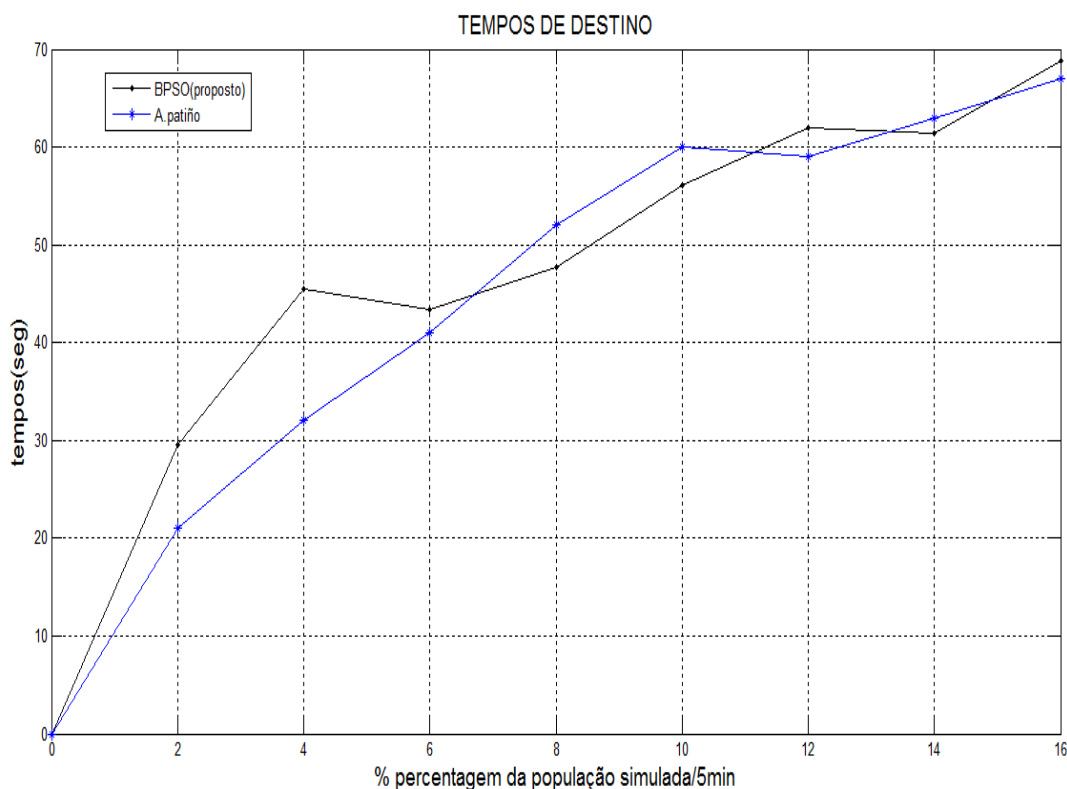


Figura 54 comparação de tempos de destino com Patino [10].

Por último, faz-se a comparação entre a quantidade de paradas realizadas pelos elevadores nos dois trabalhos, por meio da Figura 55, em que se pode observar que o número de paradas tem valores muito similares. As diferenças mais significativas localizam-se nas zonas de porcentagens médias (8% e 10%) e na alta (14% e 16%). Nesta zona (porcentuais de demanda críticos) o simulador proposto experimenta uma maior quantidade de paradas, o que pode significar um incremento nos tempos de espera e nos tempos de voo dos usuários do sistema, lembrando que esta parte é a zona mais crítica do padrão *up-peak*. Neste caso, o controlador proposto neste trabalho apresenta-se com melhor desempenho, uma vez que apresenta tempos de espera menores na zona crítica de demanda quando comparados aos obtidos por Patiño, apesar de seu sistema ter apresentado menor número de paradas. Deve-se observar, entretanto, que neste trabalho, o consumo de energia não constituiu um objetivo a ser minimizado, enquanto que no caso de Patiño, a minimização do consumo de energia era considerada.

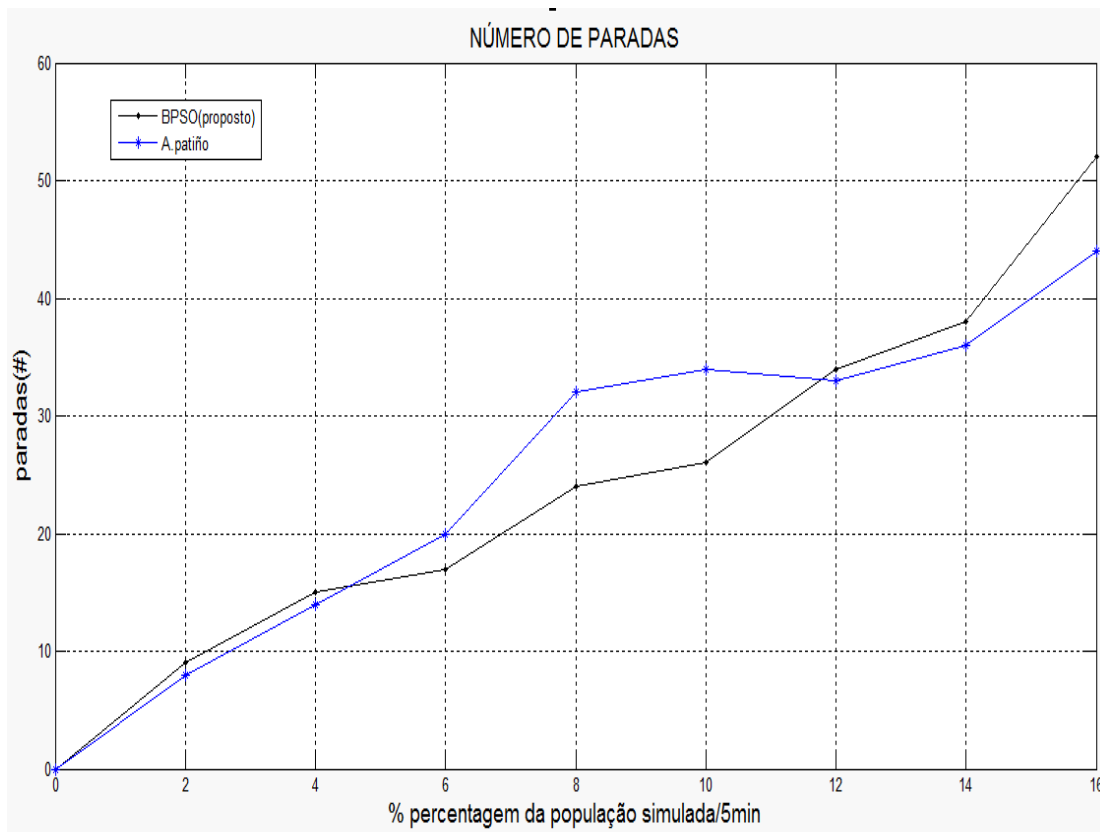


Figura 55. Comparação de numero de paradas médio com Patino [10].

5.5.2 – Comparação com os resultados de Siikonen [41]

Outra comparação interessante pode ser feita com os resultados obtidos por Siikonen [41], referência recorrente em trabalhos de pesquisa similares. Da mesma forma que com o autor anterior [10], apresenta-se na Figura 56 a comparação dos tempos de espera obtidos neste trabalho com os obtidos por Siikonen [41], os quais se apresentaram melhores que os resultados alcançados pelo controlador PSO binário desenvolvido aqui. Deve-se, entretanto, considerar que os dados apresentados na sua referência não especificam as condições do prédio simulado, tornando difícil a replicação da simulação que os gerou e não permitindo comparar diretamente os dois trabalhos. Desta forma a comparação só pode ser feita de forma qualitativa, observando-se, também, na Figura 56, que há uma similaridade entre as curvas, no tocante a sua tendência e a sua taxa de variação.

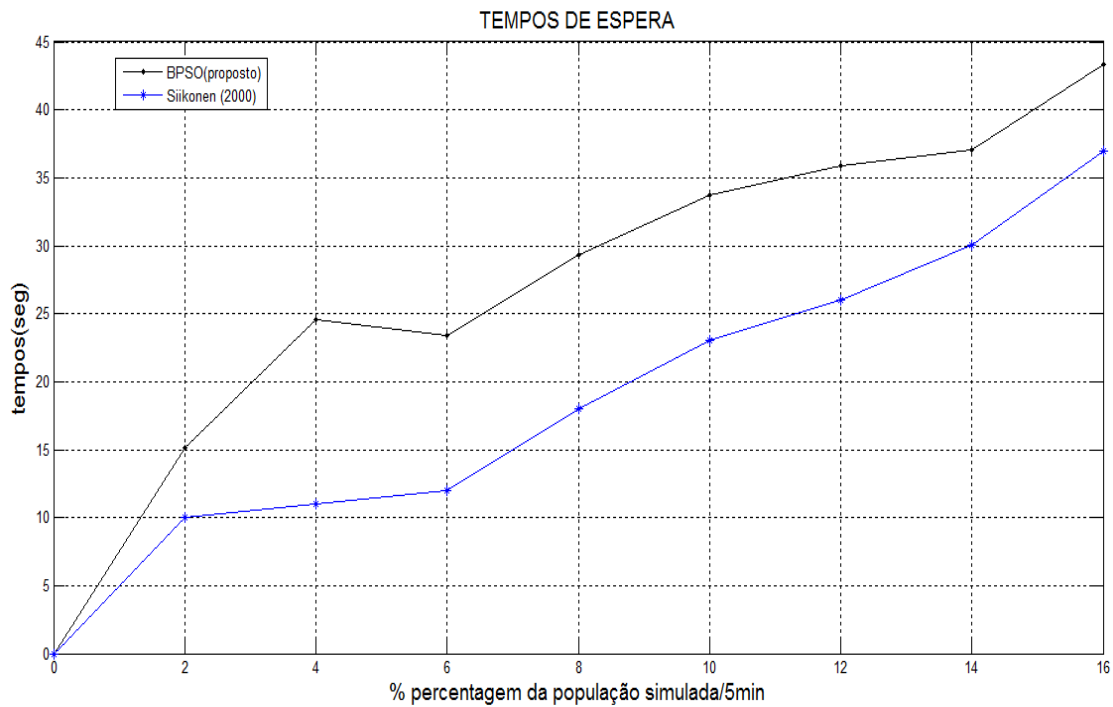


Figura 56. Comparação com o trabalho

Por último, em comparação com outros trabalhos relacionados, com os quais não se tem a possibilidade de fazer a mesma comparação gráfica, mas pode-se fazer a comparação com os tempos de espera, de voo e de destino obtidos. Na Tabela 19 observa-se que os resultados obtidos nesta pesquisa estão em concordância com os obtidos em pesquisas similares. Estas referências têm datas de publicação recentes e outras não tão recentes, relevantes nesta área de trabalho. Esta análise e comparação avalia a postulação dos algoritmos bioinspirados como possível solução para este tipo de problemas.

Tabela 19 comparação de resultados com outros autores

	ano	Autor	trabalho	tempo médio de espera	Tempo médio de voo	tempo médio de destino
1	2000	<i>Siikonen</i>	<i>On traffic planning methodology [41]</i>	37	x	x
2	2004	<i>Jian Liu</i>	<i>Dynamically dispatching method aiming to reduce the servicing time in the egcs [17]</i>	23,3	27,9	51,2

	ano	Autor	trabalho	tempo médio de espera	Tempo médio de voo	tempo médio de destino
3	2010	Morkon	<i>Design of elevator group control system simulation platform based on shortest distance algorithm [43]</i>	35.45	30.46	65.91
4	2011	Patiño	<i>Fuzzy elevator group control system using technology for industrial automation [19]</i>	47	20	67
5	2011	Sheng	<i>A Novel Elevator Group Control Scheduling Algorithm based on Pseudo Differential Feedback, [42]</i>	35,85	15,83	51,68
6	2012	Y.Gu	<i>Multi-objective optimization of multi-agent elevator group control system based on real-time particle swarm optimization algorithm [6]</i>	44.87	38.34	83.21
7	2013	Cortes	<i>A particle swarm optimization algorithm for optimal car-call allocation in elevator group control systems [9]</i>	x	x	33
8	2015	Diago	<i>Otimização com algoritmos bioinspirados de controle de tráfego em grupo de elevadores (proposto)</i>	43,37	25,5	68,87

Contudo, pode-se observar que os resultados aqui alcançados, em termos de tempos médios, situaram-se dentro das faixas de variação reportadas na literatura específica. No intuito e na tentativa de se melhorarem os tempos que caracterizam o comportamento do sistema. Na Figura 57, pode-se observar que os tempos médios de espera obtidos (cor vermelha) são similares aos tempos mais elevados obtidos por os outros autores. Fazendo uma comparação só qualitativa, pois as considerações de construção das simulações nos trabalhos apresentados são desconhecidas, o que não permite fazer uma comparação direta.

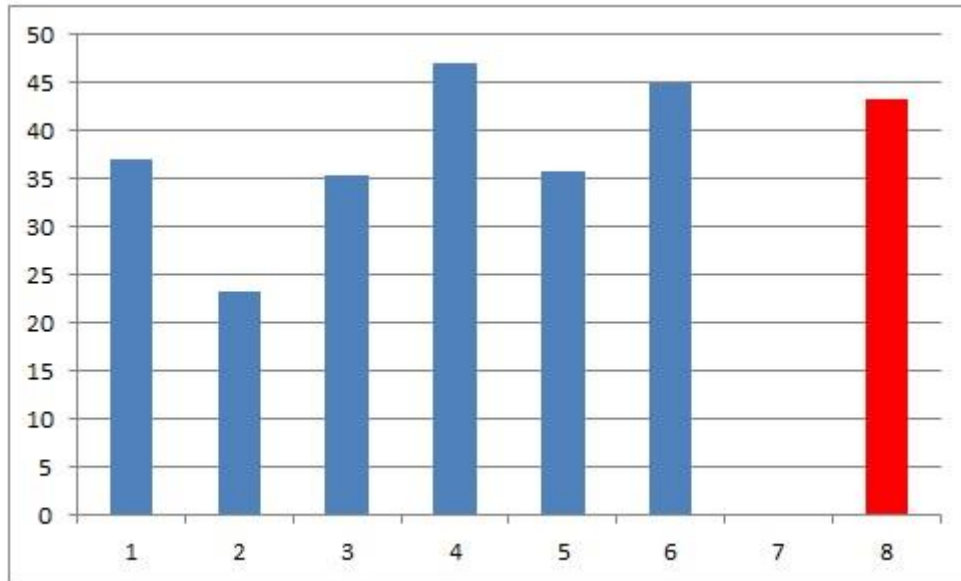


Figura 57. Tempos médios de espera

Na Figura 58, pode-se observar que os tempos médios de voo obtidos (cor vermelha) estão sobre na faixa média dos resultados comparados com outros autores

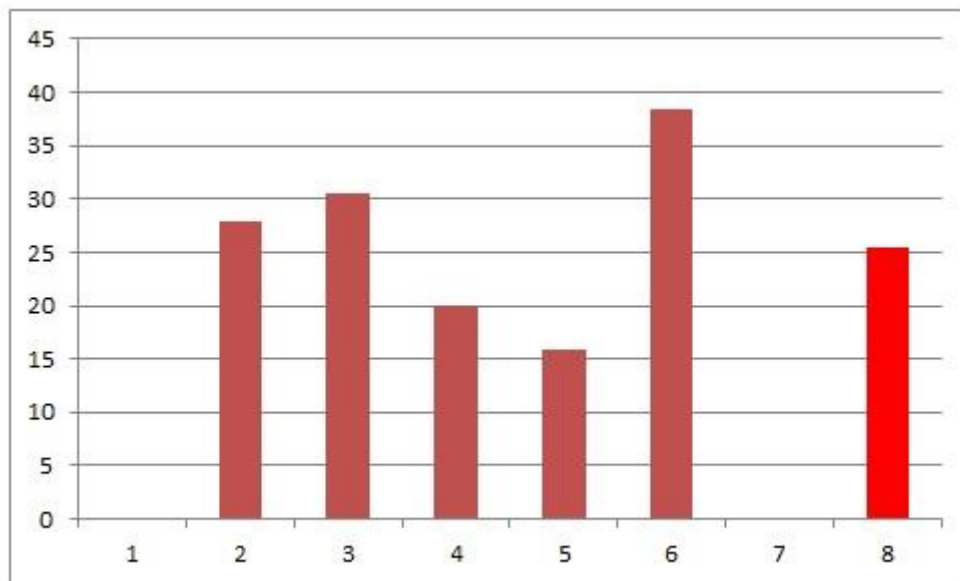


Figura 58. Tempos médios de voo

6. DISCUSSÃO DOS RESULTADOS

Neste trabalho propõe-se como uma nova abordagem na procura de uma possível solução ao problema de controle de sistema de grupos de elevadores. A proposta esta baseada em técnicas de otimização bioinspiradas, especificamente no algoritmo PSO de tipo binário. A implementação desta técnica corresponde, em primeira instância, ao caso estático, no qual não é considerada a dinâmica de movimentação dos elevadores. Esta etapa da pesquisa desenvolvida em Matlab, em cujo ambiente foram feitos testes de funcionalidade e de adequação do controlador desenvolvido e da sua função custo. Em uma segunda instância, considerou-se a dinâmica do sistema de grupo de elevadores e lembrando que o objetivo final do trabalho era realizar o controle de um sistema de grupo de elevadores de forma dinâmica respondendo a chamadas sucessivas realizadas por usuário baseado em um padrão de tráfego escolhido (neste caso *Up-peak*). Nesta etapa a ferramenta software usada para a simulação foi o *Arena* da *Rockwell Automation*. Esta ferramenta permite um ambiente adequado para a representação do problema, a programação, depuração, trabalhos com tempos e facilidade para o trabalho com variáveis estatísticas.

O primeiro passo na consecução do objetivo geral foi estudar diferentes técnicas de otimização bioinspiradas, entendendo-as e testando-as no intuito de escolher a técnica mais adequada para implementar o controle de grupos de elevadores. Com base nos resultados experimentais obtidos, foi possível concluir que o algoritmo PSO apresentava os melhores resultados finais de aproximação à solução mais ótima em problemas com diferentes graus de dificuldade (funções do tipo *benchmark*). Por outro lado, a facilidade de implementação do algoritmo e a existência de trabalhos prévios que abordam a discretização do PSO, justificam a escolha dele para resolver o problema de controle de grupo de elevadores.

A definição da função custo foi um passo necessário e um dos resultados mais relevantes desta pesquisa. Para a solução de qualquer problema de otimização, o primeiro passo é a determinação de uma função custo, que deve ser minimizada ou maximizada para se atingir o ponto ótimo. No caso particular deste trabalho a função custo foi definida como uma soma ponderada de seis critérios: (1) tempo de espera, (2) tempo de voo, (3) tempo máximo em atender uma chamada, (4) número de pessoas no elevador, (5) número de

chamadas alocadas no elevador, e (6) número de pessoas esperadas em cada elevador quando chega ao andar de origem, como indicado na equação (5-3). A escolha destes critérios foi baseada na abordagem apresentada por Yu L *et al.* [40]. Entretanto, uma contribuição original deste trabalho foi a adoção de penalidades quando os elevadores têm pouca capacidade de carga, induzindo o algoritmo a escolher soluções com menores tempos de espera e maior número de vagas, aumentando assim o conforto dos usuários. Uma contribuição adicional é a proposta nos modelos matemáticos para estimar os critérios escolhidos tal como é apresentado nas equações (5-4) e (5-5).

Com base nas simulações numéricas apresentadas no Capítulo 4, desenvolvidas em Matlab, é possível destacar os seguintes pontos: (1) as simulações foram feitas no intuito de escolher o melhor elevador quando uma chamada de tipo *up-peak* é realizada, considerando-se diversos cenários no sistemas de elevadores; (2) os resultados obtidos pelo algoritmo PSO discreto são satisfatórios considerando que para os cenários pré-condicionados a solução escolhida pelo algoritmo era a solução esperada; (3) foi observado que o algoritmo respondeu adequadamente para os casos em que chamadas *interfloor* foram eventualmente realizadas. Este tipo de chamadas atendem os modelos probabilísticos usados para a geração do padrão de tráfego *up-peak* e simulam situações reais encontradas nos prédios comerciais.

Depois dos resultados obtidos no Matlab, o seguinte passo foi a implementação do simulador no software de simulação de processos industriais *Arena* da *Rockwell Automation*. Nesta etapa foi necessário o estudo de manuais de operação desta ferramenta software encontrando a primeira dificuldade, a pouca literatura disponível sobre configuração e programação. Após estudo da literatura disponível, bem como de exemplos encontrados no software, conseguiram-se desenvolver modularmente as diferentes etapas do modelo do problema planejado. Uma segunda dificuldade foi a integração dos módulos desenvolvidos, pois as licenças disponíveis das versões de estudante e “*basic*” são limitadas em relação ao número de blocos, de variáveis, etc, que impossibilitaram realizar um teste com o simulador completo. Consequentemente, só foi possível realizar testes por partes, assim:

- (1) Geração e atribuição de chamadas de acordo ao padrão *up-peak* para cada elevador do sistema;

- (2) Foi simulado o processo de atendimento aos usuários pelos elevadores, analisando os tempos de espera, os tempos de voo e os tempos de destino em cada teste simulado. Deve-se lembrar que as porcentagens de população simulada foram variadas desde 2% até 16%, considerando a ocorrência dessa demanda em períodos de 6 min (5 minutos efetivos, sendo os 30 segundos iniciais e finais descartados);
- (3) Os resultados obtidos foram analisados e comparados com outros autores, só de forma qualitativa, pois os simuladores são construídos de forma diferente.

O simulador desenvolvido para avaliar a técnica de otimização BPSO apresentou desempenho adequado, respondendo às variações das condições do sistema e selecionando por parte do controlador proposto o elevador com melhores características para atender às chamadas. Entretanto, os resultados obtidos (tempo de espera e tempo de voo) nas simulações pode se observar que o sistema de simulação realizado satisfaz os requerimentos desta pesquisa. Comparando-se os resultados obtidos nesta pesquisa com outros resultados da literatura científica sobre o problema dos elevadores, como se pode observar graficamente na comparação com dois autores e numericamente com outros, na Tabela 19. Esta comparação foi feita para observar a compatibilidade dos resultados obtidos com os antecedentes e avaliar assim o controlador proposto (novamente é só qualitativamente). Desta comparação, o controlador proposto tem a mesma tendência dos resultados de pesquisas similares, o que pode ser observado como um desempenho adequado para o sistema de controle. Também, pode-se encontrar que é necessário em trabalhos futuros continuar fazendo melhorias ao controlador proposto e ao simulador desenvolvido, no intuito de melhorar os resultados obtidos. Neste mesmo senso é necessário desenvolver novos testes de escalabilidade, isto é, com números maiores de dados (sistemas de grupos de elevadores com: maiores números de elevadores, maior população, mais andares e a inclusão de outras variáveis que podem melhorar a simulação).

Por outro lado, espera-se que os resultados e as técnicas exploradas neste trabalho representem uma base importante para estudos futuros na área dos sistemas de grupos de elevadores, contribuindo com as linhas de pesquisa em sistemas de controle no laboratório GRACO (Grupo de Automação e controle) do programa de Sistemas Mecatrônicos da UnB.

7. CONCLUSÕES E TRABALHOS FUTUROS

7.1. CONCLUSÕES

No presente trabalho, foi apresentado-se um estudo da aplicação de algoritmos baseados em inteligência de enxames para resolver o problema de controle de grupo de elevadores em uma situação de tráfego tipo *up-peak*. De acordo com objetivos planejados no início da pesquisa se pode concluir que:

- Nos testes feitos com os algoritmos bioinspirados PSO, ABC e FA em problemas tipo *benchmark* os resultados numéricos mostram a competitividade dos algoritmos na solução de problemas de otimização multimodal. Mas o algoritmo PSO apresentou melhor desempenho se considerada a aproximação ao ótimo global. Adicionalmente, os resultados mostram que o incremento do número de agentes (tamanho do enxame) permite melhorar o desempenho dos algoritmos. Isto pode ser explicado considerando o aumento da diversidade na busca da solução ótima, em especial para problema de alta complexidade (número alto de dimensões). Entretanto, é importante lembrar que o uso de mais agentes pode gerar um consumo computacional maior.
- Um resultado importante desta pesquisa é formação e definição da função custo que representa o problema de otimização. Os resultados de simulação demonstraram que o algoritmo PSO binário consegue selecionar o melhor elevador para diversas chamadas de serviço, comprovando a escolha do elevador mais apropriado para diferentes cenários de acordo com o melhor valor de aptidão.
- Foi implementado no software *Arena* o simulador de grupo de elevadores com o controlador proposto baseado em inteligência de exames BPSO. Este simulador foi testado na faixa crítica de fluxo de usuários em um prédio comercial (*up-peak*), concluindo a partir dos resultados obtidos, que o algoritmo de controle fornece desempenho satisfatório ao tentar controlar o tráfego presente no sistema.
- Os resultados obtidos pelo controlador proposto têm tendências e concordância quando foram comparados e analisados com os resultados encontrados em pesquisas similares

na literatura científica. Porém, pode se estabelecer que o sistema de controle proposto postula-se como uma solução potencial para problemas do tipo trabalhado nesta pesquisa.

7.2. SUGESTÕES PARA TRABALHOS FUTUROS

- Podem-se realizar melhoras na construção do simulador e possivelmente ajustes finos ao controlador (possíveis melhoras em na técnica para binarizar o PSO e o uso de técnicas de diversidade nele), no intuito de melhorar os resultados obtidos na presente pesquisa.
- Novos testes do controlador proposto em sistemas de grupos de elevadores mais exigentes, isto é com um maior número de andares, maior número de elevadores e população. Para avaliar o desempenho do controlador proposto em novos cenários
- Pode ser realizada a postulação de uma modificação ao controlador, no intuito de ter uma proposta multi-objetivo, onde além da otimização de tempos de serviço a os usuários, se consiga otimizar o consumo de energia do sistema de grupo de elevadores.
- Pode ser realizado um trabalho com algoritmos bioinspirados para estimar de forma mais adequada os valores das constantes de ponderação da função custo que foi desenvolvida, no intuito de melhorar o desempenho do controlador proposto.
- Podem ser realizados novos testes com versões binarias de algoritmos bioinspirados como ABC e FA, utilizando a função custo desenvolvida, no intuito de comparar o desempenho desses algoritmos com o BPSO proposto neste trabalho.
- Por último, pode-se pensar no futuro em propor um controlador baseado em algoritmos de otimização bioinspirados junto com outra técnica de controle moderna como a *lógica fuzzy*. A interação de essas técnicas de controle poderia fornecer resultados interessantes.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] P Cortés, J Larrañeta, L Onieva, J Muñuzuri, and I Fernández, "Algoritmos de Optimización en Sistemas de Transporte Vertical," *II Conferencia de Ingeniería de Organización*, 2002.
- [2] Gina Barney, *Elevator Traffic Handbook Theory And Practice.*, 2003.
- [3] S.A. Markon, H. Kita, H. Kise, and Th. Bartz-Beielstein, *Control of Traffic Systems in Buildings.*, 2006.
- [4] pablo cortes, maria del carmen joaquin. fernandez, *fuzzy logic-based controller for vertical traffic detection purpose.*, 2009.
- [5] Tommi Tervonena, Henri Hakonenb, and Risto Lahdelmab, "Elevator planning with stochastic multicriteria," *the international journal of management science*, 2006.
- [6] Yanwu Gu, "Multi-Objective Optimization of Multi-Agent Elevator Group Control System Based on Real-Time Particle Swarm Optimization Algorithm," *Engineering, Vol. 4 No. 7*, 2012.
- [7] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *International Symposium Micro Machine and Human Science. Nagoya, Japan: IEEE*, 1995.
- [8] Daniel Muñoz, *otimizacao por inteligencia de enxames usando arquiteturas paralelas para aplicacoes embarcadas.*: UnB- Universidade de Brasilia, 2012.
- [9] Berna Bolata, Oguz Altunb, and Pablo Cortés, "A particle swarm optimization algorithm for optimal car-call allocation in elevator group control systems," *Applied Soft Computing 13* , 2013.
- [10] Alvaro Patiño, *Estudo e simulação de técnicas de controle de tráfego de grupo de elevadores usando automação industrial.*: Unb- Universidade de Brasilia, 2010.
- [11] Valentino Crespi, Aram Galstyan, and Kristina Lerman, "Top–Down vs Bottom–up

- Methodologies in Multi-Agent System Design," *Autonomous Robots*, 2008.
- [12] Jamaludin J., Rahim N.A., and Wooi Ping Hew, "An Elevator Group Control System With a Self-Tuning Fuzzy Logic Group Controller," *IEEE Transactions on Industrial Electronics (Volume:57)*, 2010.
- [13] George T. Hummet, Thomas D. Moser, and Bruce A. Powell, "Real time simulation of elevators," in *WSC '78 Proceedings of the 10th conference on Winter simulation - Volume 2*, 1978.
- [14] Rockwell Automation, *Arena Basic User's guide.*, 2005.
- [15] Tayfur Altiok and Benjamin Melamed, *Simulation Modeling And Analysis with Arena.*: Academic Press is an imprint of Elsevier, 2007.
- [16] Marvin S. Seppanen, "Developing industrial strength simulation models using Visual Basic for Applications (VBA)," in *Proceedings of the 2000 Winter Simulation Conference*, 2000.
- [17] Y Zhou and Q Ye, "Dynamically Dispatching Method Aiming to Reduce the Servicing Time in the EGCS," *Proceedings of the 5'World Congress on Intelligent Control and Automation*, 2004.
- [18] Marja-Liisa Siikonen, "Planning and Control Models for Elevators in High-Rise Buildings," KONE Corporation P.O. Box 8 SF-00331 Helsinki, Finland 1998.
- [19] Alvaro Patiño Forero, Daniel M Muñoz, Guilherme Caribé de Carvalho, and Carlos Llanos, "fuzzy elevator group control system using technology for industrial automation," *21st Brazilian Congress of Mechanical Engineering - COBEM*, 2011.
- [20] Jian Liu, Chengdong Wu, Xin Wang, Weize Wang, and Ting Zhang, "A Hybrid Control for Elevator Group System," *Third International Workshop on Advanced Computational Intelligence*, 2010.
- [21] A. Fujino, T. Tobita, K. Segawa, K. Yoneda, and A. Togawa, "An Elevator Group Control System with Floor-Attribute Control Method and System Optimization Using

- Genetic Algorithms," *IEEE Transactions on Industrial Electronics*, 1997.
- [22] Adriane Beatriz de Souza Serapião, "Fundamentos de otimização por inteligência de enxames: uma visão geral," *Revista Controle & Automação/Vol.20*, 2009.
- [23] Simoní Da Rosa et al., "A comparison among stochastic optimization algorithms for parameter estimation of biochemical kinetic models," *Applied Soft Computing*, 2013.
- [24] Jian Liu, Chengdong Wu, Meiju Liu, Enyang Gao, and Guojiang Fu, "RBF Optimization Control Based on PSO for Elevator Group System," *International Conference on Information Science and Technology*, 2011.
- [25] Dervis Karaboga and Bahriye Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, v. 39., 2007.
- [26] Liu y, Zhang L Liang y, "“An Improved Artificial Bee Colony (ABC) Algorithm for Large Scale Optimization”.", *International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, 2013.
- [27] Karaboga D. and Gorkemli B., "A Quick Artificial Bee Colony -qABC- Algorithm for Optimization Problems," in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2012.
- [28] Xin-She Yang, "Firefly algorithms for multimodal optimization," *Lecture Notes on Computers Sciences: Stochastic Algorithms: Foundations and Applications*, v. 5792, 2010.
- [29] Dervis Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06 2005.
- [30] R Eberhart and J Kennedy, "A discrete binary version of the particle swarm algorithm," in *International Conference on Systems, Man, and Cybernetics*, 1997.
- [31] Nelson Guerra Álvarez and Broderick Crawford Labrín, "Optimización de funciones a través de Optimización por Enjambre de Partículas y Algoritmos Genéticos," in

Conferencia Latinoamericana de Informática, Santiago, Chile, 2006.

- [32] Dina EL-Gammal, Amr Badr, and Mostafa Abd El Azeim, "New Binary Particle Swarm Optimization With Immunity-Clonal Algorithm," *Journal of Computer Science*, 2013.
- [33] M. Gomez-Gonzalez, A. López, and F. Jurado, "Optimization of distributed generation systems using a new discrete PSO and OPF," *European Journal of Operational Research*, 2012.
- [34] Alper Unler and Alper Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *European Journal of Operational Research*, 2010.
- [35] A. Bailey, B. Ornbuki-Bernan, and S. Asobiela, "Discrete PSO for the Uncapacitated Single Allocation Hub Location Problem," *Computational Intelligence In Production And Logistics Systems (CIPLS)*, 2013.
- [36] J. Riget and J.S. Vesterstrom, "A diversity-guided particle swarm optimizer - the ARPSO," EVALife Technical Report no. 2002-02 2002.
- [37] Julia K. Parrish and William M. Hamner, "Animal Groups in Three Dimensions How Species Aggregate," *Cambridge, UK: Cambridge University Press*, 1997.
- [38] Hamid R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," *Proc. Int. Conference on Computational Intelligence for Modelling, Control and Automation. Vienna, Austria*, 2005.
- [39] Shahryar Rahnamayan, Hamid R. Tizhoosh, and Magdy M.A. Salama, "Opposition versus randomness in soft computing techniques," *Journal Applied Soft Computing*, 2008.
- [40] Lu Yu et al., "Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming with Ant Colony Optimization with Evaporation," *IEEE Congress on Evolutionary Computation*, 2007.

- [41] Marja-Liisa Siikonen, "On traffic planning methodology," in *International Congress on Vertical Transportation*, Berlin, 2000.
- [42] Sheng Wu and Guifang Wu, "A Novel Elevator Group Control Scheduling Algorithm based on Pseudo Differential Feedback," *Proceeding of the IEEE, International Conference on Automation and Logistics*, 2012.
- [43] Wang Chuansheng and Chen Chunping, "Design of Elevator Group Control System Simulation Platform Based on Shortest Distance Algorithm," *International Conference on Electrical and Control Engineering*, 2010.
- [44] Jin Zhou et al., "Double-Deck Elevator Systems using Genetic Network Programming with Reinforcement Learning," *IEEE Congress on Evolutionary Computation* , 2007.
- [45] Gonçalves C. F., *Estatística.*: Universidade Estadual de Londrina, Brasil, 2002.

APÉNDICE A

Na Tabela 20 apresentam-se os resultados obtidos dos testes em Matlab do controlador e a função custo desenvolvidos, esses testes foram feitos no intuito avaliar o funcionamento deles.

Tabela 20 testes de seleção de elevador pelo BPSO

test	Condições iniciais					resultados			
		Fa	Direção	# paradas	# pessoas	Fo	Fd	Solução	f(i)
1	elev 1	3	sobe	4	3	1	8	elev 2 (0 1 0)	25,729
						1	9	elev 2 (0 1 0)	26,023
	elev 2	9	baixa	3	1	1	4	elev 2 (0 1 0)	25,750
						1	6	elev 2 (0 1 0)	25,350
	elev 3	2	sobe	5	4	1	2	elev 2 (0 1 0)	25,940
						1	3	elev 2 (0 1 0)	25,946
2	elev 1	9	baixa	3	1	1	8	elev 1 (1 0 0)	25,925
						1	3	elev 1 (1 0 0)	25,755
	elev 2	3	sobe	4	3	1	4	elev 1 (1 0 0)	25,602
						1	6	elev 1 (1 0 0)	25,721
	elev 3	2	sobe	5	4	1	8	elev 1 (1 0 0)	25,576
						1	7	elev 1 (1 0 0)	25,457
3	elev 1	3	sobe	4	3	1	9	elev 3 (0 0 1)	25,750
						6	9	elev 3 (0 0 1)	26,197
	elev 2	2	sobe	5	4	1	3	elev 3 (0 0 1)	25,814
						1	9	elev 3 (0 0 1)	25,664
	elev 3	9	baixa	3	1	1	4	elev 3 (0 0 1)	25,528
						1	9	elev 3 (0 0 1)	25,939
4	elev 1	10	baixa	5	3	6	9	elev 2 (0 1 0)	25,582
						1	3	elev 2 (0 1 0)	25,799
	elev 2	9	baixa	3	3	1	6	elev 2 (0 1 0)	25,903
						1	2	elev 2 (0 1 0)	25,280
	elev 3	2	sobe	6	4	1	5	elev 2 (0 1 0)	25,571
						1	2	elev 2 (0 1 0)	25,359
5	elev 1	9	baixa	3	3	1	7	elev 1 (1 0 0)	25,612
						1	2	elev 1 (1 0 0)	25,886
	elev 2	10	baixa	5	3	1	3	elev 1 (1 0 0)	25,578
						1	9	elev 1 (1 0 0)	25,406
	elev 3	2	sobe	6	4	1	6	elev 1 (1 0 0)	25,604
						1	3	elev 1 (1 0 0)	25,732
6	elev 1	10	baixa	5	3	1	9	elev 3 (0 0 1)	25,894
						1	7	elev 3 (0 0 1)	25,395

test	Condições iniciais						resultados		
		Fa	Direção	# paradas	# pessoas	Fo	Fd	Solução	f(i)
	elev 2	2	sobe	6	4	1	8	elev 3 (0 0 1)	25,796
						1	3	elev 3 (0 0 1)	25,640
	elev 3	9	baixa	3	3	6	4	elev 3 (0 0 1)	25,000
						1	2	elev 3 (0 0 1)	25,909
7	elev 1	3	sobe	2	3	1	10	elev 2 (0 1 0)	20,516
						4	3	elev 3 (0 0 1)	28,321
	elev 2	8	baixa	2	3	1	10	elev 2 (0 1 0)	20,365
						1	6	elev 2 (0 1 0)	20,231
	elev 3	2	sobe	2	3	1	4	elev 2 (0 1 0)	20,090
						1	7	elev 2 (0 1 0)	20,256
8	elev 1	5	sobe	3	5	7	3	elev 3 (0 0 1)	37,937
						2	6	elev 2 (0 1 0)	36,677
	elev 2	3	sobe	3	3	1	2	elev 2 (0 1 0)	36,353
						1	9	elev 1 (1 0 0)	36,182
	elev 3	1	sobe	4	1	4	7	elev 2 (0 1 0)	33,304
						8	6	elev 2 (0 1 0)	33,315
9	elev 1	10	baixa	3	2	1	9	elev 3 (0 0 1)	31,006
						1	3	elev 3 (0 0 1)	31,141
	elev 2	5	sobe	3	2	1	5	elev 3 (0 0 1)	31,359
						1	8	elev 3 (0 0 1)	30,853
	elev 3	9	baixa	2	5	1	6	elev 3 (0 0 1)	31,080
						1	5	elev 3 (0 0 1)	31,038
10	elev 1	1	sobe	5	6	1	8	elev 2 (0 1 0)	38,272
						1	5	elev 2 (0 1 0)	38,196
	elev 2	1	sobe	4	5	1	3	elev 2 (0 1 0)	37,771
						1	6	elev 2 (0 1 0)	38,183
	elev 3	1	sobe	5	4	1	8	elev 2 (0 1 0)	38,456
						1	5	elev 2 (0 1 0)	38,432
11	elev 1	9	baixa	3	2	1	4	elev 1 (1 0 0)	36,178
						1	8	elev 1 (1 0 0)	36,317
	elev 2	3	sobe	3	1	1	7	elev 1 (1 0 0)	36,315
						10	8	elev 1 (1 0 0)	34,314
	elev 3	1	sobe	4	2	8	6	elev 1 (1 0 0)	32,658
						1	3	elev 1 (1 0 0)	36,475
12	elev 1	2	sobe	1	1	1	2	elev 1 (1 0 0)	26,225
						1	8	elev 1 (1 0 0)	26,058
	elev 2	2	sobe	2	2	1	4	elev 1 (1 0 0)	26,146
						2	5	elev 1 (1 0 0)	25,831
	elev 3	10	baixa	4	3	1	8	elev 1 (1 0 0)	26,403
						1	9	elev 1 (1 0 0)	26,408
13	elev 1	2	sobe	3	3	6	10	elev 3 (0 0 1)	36,817
						1	5	elev 2 (0 1 0)	36,615
	elev 2	9	baixa	3	4	9	6	elev 2 (0 1 0)	33,572

test	Condições iniciais						resultados		
		Fa	Direção	# paradas	# pessoas	Fo	Fd	Solução	f(i)
						1	10	elev 2 (0 1 0)	36,787
	elev 3	7	baixa	3	4	1	3	elev 3 (0 0 1)	36,275
						1	5	elev 2 (0 1 0)	36,328
14	elev 1	9	baixa	3	3	1	5	elev 3 (0 0 1)	53,088
						1	10	elev 1 (1 0 0)	53,363
	elev 2	8	baixa	3	5	1	3	elev 1 (1 0 0)	52,440
						1	4	elev 1 (1 0 0)	52,602
	elev 3	7	baixa	3	3	9	4	elev 2 (0 1 0)	50,521
						1	8	elev 1 (1 0 0)	53,462
15	elev 1	9	baixa	3	3	1	9	elev 3 (0 0 1)	53,015
						1	2	elev 2 (0 1 0)	53,208
	elev 2	2	sobe	4	4	1	10	elev 3 (0 0 1)	52,954
						1	5	elev 3 (0 0 1)	52,406
	elev 3	1	sobe	4	4	1	7	elev 2 (0 1 0)	53,210
						1	5	elev 3 (0 0 1)	52,675

APÊNDICE B

A seguir apresenta-se o uma parte do pseudocódigo desenvolvido em Matlab (main e função do controlador PSO binário).

```
% otimização de controlador de trafego em grupo de elevadores com
% algoritmos bioinspirados (bpsi binary swarm particle optimization)
% primeira aproximação de 2014
%*****
% características gerais do sistema e constantes do sistema
Fl=10; % numero de andares do prédio
pop=460; % numero de pessoas do prédio
C=3; % numero de elevadores
CC=8; % capacidade de cada elevador
Tip= 3; % tempo médio que demora uma pessoa em entrar no elevador(seg)
Top= 3; % tempo médio que demora uma pessoa em sair do elevador(seg)
Tod= 3; % tempo de abertura de portas do elevador
Tcd= 3; % tempo de fechada de portas do elevador
Vmax= 3; % velocidade do elevador max
Df= 3.3; % espaço entre os andares em metrô
Vac= 1; % em aceleração
Vdac= 1; % em desaceleração
%-----
% inicialização de variáveis
% num_test=3;
[RC,C_EST,RCS,RCB]= test_vector_sol();
RC
C_EST
RCS
RCB
%-----
% geração de chamadas de acordo com os padroes de trafego
```

```

ttrf=1; % ttrf tipo de trafego uppeak(1), downpeak(2) interfloor(3)
% ainda falta "taxa de geracao de chamadas o de chegada de passageiros"
[Fo,Fd]= ger_cham (ttrf,Fl);
Fo
Fd
pause(5);
%-----
% avaliação de elevador selecionado para cada chamada
[C_SEL] = bpso_jp_v3(RC,C_EST,RCS,RCB,Fo,Fd,CHC);
C_SEL
%-----

function [C_SEL] = bpso_jp_v3(RC,C_EST,RCS,RCB,Fo,Fd,CHC)

maxiter=150;
S=10;
N=3;
f_ind = 1e10*ones(maxiter,1);
w0 = 0.8; % initial weight
wf = 0.1; % final weight
w=zeros(maxiter,1);
w(1)= w0;
slope = (wf-w0)/maxiter;
c1 = 2; % cognitive coefficient
c2 = 2; % 1.9999;% 1.9999; % social coefficient
max_v = 3; % max velocity
ini_v = 0.5;

k = 1; % index of iteration
% pesos dos parametros da funcao de custo
k1=0.4; % peso para o tempo de espera pelo elevador i
k2=0.3; % peso para o maximo tempo de espera para atender uma chamada alocada no
elevador i

```

```

k3=0.1; % peso para o tempo de Voo do andar origem ao andar destino da chamada no
elevador i
k4=0.05; % peso para a capacidade atual de receber pessoas no elevador i
k5=0.05; % peso para o numero de paradas alocadas no elevador i
k6=0.1; % peso para a capacidade esperada do elevador i quando chegar ao andar origem
da chamada
for j=1:N
    for i=1:S
        x(i,j) = ones;
        y(i,j) = 1e10;
        v(i,j) = ini_v;
    end
end
while k<= maxiter
    % Evaluates fitness and detection
    for i = 1:S
        %% Fitness functions %%
        % posible solucao da particula i para a chamada feita
        p_sol=x(i,:);
        [AT,ATV,AET,NP,NC,RRi] = fcn_awt_prb(p_sol,RC,C_EST,RCS,RCB,CHC,Fo,Fd);
        %
        fx(i)=(k1*AT+k2*ATV+k3*AET+k4*NP+k5*NC+k6*RRi);
        if fx(i) < f_ind(i)
            y(i,:) = x(i,:);
            ATs(i)=AT;
            ATV(i)=ATV;
            AETs(i)=AET;
            NPs(i)=NP;
            NCs(i)=NC;

            f_ind(i) = fx(i);
        end
    end
end
[bestfitness(k), p] = min(f_ind);

```

```

ys = y(p,:);
% update particles
for j = 1:N
    for i=1:S
        r1 = rand();
        r2 = rand();
        v(i,j) = w(k)*v(i,j) + c1*r1*(y(i,j)-x(i,j)) + c2*r2*(ys(j) - x(i,j));
        if abs(v(i,j)) > max_v
            if v(i,j) > 0
                v(i,j) = max_v;
            else
                v(i,j) = -max_v;
            end
        end
        sig=1/(1+exp(-v(i,j)));
        r=rand();
        if sig>r;
            x(i,j) = 1;
        else
            x(i,j)=0;
        end
    end
end
x;
X=bi2de(x);
X;

k=k+1;
w(k) = w(k-1) + slope;
end
fprintf('aptidao da melhor solucao');
out = bestfitness(maxiter)
% fprintf('parametros da melhor solucao');
C_SEL=ys;

```

```
fprintf('valores parametros');  
ATs(p)  
ATVs(p)  
AETs(p)  
NPs(p)  
NCs(p)  
fprintf('elevador escolhido');  
  
% plot (fx);  
% grid on;
```