

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO – UENF

FABIO JUNIO DOS SANTOS COELHO

**MODELO DE COMPARTILHAMENTO DE ESTAÇÕES TERRESTRES
EM REDE COM NEGOCIAÇÃO**

**CAMPOS DOS GOYTACAZES – RJ
MARÇO/2017**

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO – UENF

FABIO JUNIO DOS SANTOS COELHO

**MODELO DE COMPARTILHAMENTO DE ESTAÇÕES TERRESTRES
EM REDE COM NEGOCIAÇÃO**

Dissertação apresentada ao Centro de Ciências e Tecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Rogério Atem de Carvalho, D. Sc.

**CAMPOS DOS GOYTACAZES – RJ
MARÇO/2017**

MODELO DE COMPARTILHAMENTO DE ESTAÇÕES TERRESTRES EM REDE COM NEGOCIAÇÃO

FABIO JUNIO DOS SANTOS COELHO

Dissertação apresentada ao Centro de Ciências e Tecnologia da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção.

Aprovada em 29 de Março de 2017

Comissão Examinadora:

Prof. Rogério Atem de Carvalho, D. Sc. – IFF/UENF
Orientador

Prof. Milton Erthal Junior, D. Sc. – IFF/UCAM

Prof.^a. Annabell Del Real Tamariz, D. Sc. – UENF

Prof. Luiz Gustavo Lourenço Moura, D. Sc. – IFF

Agradecimentos

Primeiramente, agradeço a Deus por ter me guiado até aqui e por ter me dado forças para correr atrás dos meus sonhos.

Aos meus pais pelo incentivo constante para os estudos e pela educação dada em todas as etapas da minha vida. Essa conquista também é de vocês.

A minha amada esposa, por estar ao meu lado nessa fase da minha vida. Agradeço por todo o amor, compreensão e companheirismo. Obrigado pelo grande apoio dado.

Ao meu orientador, Rogério Atem de Carvalho, pela dedicação e conhecimento compartilhado durante esse curso de Mestrado. Obrigado pela prontidão em me atender e tirar as inúmeras dúvidas ao longo dessa jornada.

Aos professores do LEPROD, pelo conhecimento passado nas disciplinas cursadas e aos membros da banca pelo tempo despendido na leitura e defesa dessa dissertação de Mestrado.

RESUMO

Os satélites artificiais desempenham funções importantes em diversas atividades como, nos sistemas de telecomunicações, posicionamento global e na área de pesquisas atmosféricas e espaciais. A pesquisa com satélites de pequeno porte vem crescendo, devido principalmente ao baixo custo associado a uma infraestrutura bem mais simples do que as dos satélites convencionais. Esses satélites normalmente trabalham em rede, já que um único satélite não suporta muitos sensores. Para realizar a comunicação com esses satélites existe um conjunto de estações que formam o segmento terrestre. Em determinados momentos podem ocorrer conflitos de visibilidade, caracterizados quando mais de uma estação tem a possibilidade de comunicar com um mesmo satélite ou vice-versa. Esse trabalho solucionou o problema de alocação das estações terrestres nessas situações de conflito de forma descentralizada, buscando permitir a negociação entre as estações envolvidas em cada conflito. Num primeiro momento foi gerado um modelo de decisão centralizada, utilizando otimização linear. A partir de um cenário obtido através de simulação computacional contendo as efemérides dos satélites foi aplicado o modelo, sendo resolvido pelo método simplex e foi obtida a solução global para ser comparada com as soluções parciais do segundo modelo. Este segundo trata-se de um modelo descentralizado, que é o objetivo principal deste trabalho, baseado em teoria dos grafos onde o problema global foi subdividido em grafos menores, cada um representando um domínio de conflito com estações e satélites específicos, baseados no mesmo cenário utilizado no modelo centralizado. Cada subproblema foi solucionado através de uma meta-heurística baseada em inteligência de enxames, denominada ACO (Otimização por Colônia de Formigas). Os resultados mostram que o modelo descentralizado se mostrou eficiente, onde a união de suas soluções parciais forneceram a mesma solução do modelo centralizado aplicado ao mesmo problema de otimização.

Palavras-chave: satélites artificiais, conflitos de visibilidade, Inteligência de Enxames, algoritmo ACO.

ABSTRACT

Artificial satellites play important roles in various activities such as telecommunications systems, global positioning and atmospheric and space research. Research with small satellites has been growing, mainly due to the low cost associated with a much simpler infrastructure, when compared to conventional satellites. These satellites usually work in a network, since a single satellite does not support many sensors. To perform the communication with these satellites there is a set of stations that form the ground segment. Visibility conflicts can occur at certain moments, characterized when more than one station has the possibility to communicate with the same satellite or vice versa. This work solved the problem of allocating of ground stations in these conflict situations in a decentralized way, seeking to allow negotiation between the stations involved in each conflict. At first, a centralized decision model was generated, using linear optimization. From a scenario obtained through a computer simulation containing the satellite ephemeris, the model was applied, being solved by the simplex method and the global solution was obtained to be compared with the partial solutions of the second model. This second is a decentralized model, which is the main objective of this work, based on graph theory where the global problem was subdivided into smaller graphs, each representing a domain of conflict with specific stations and satellites, based on the same scenario used in the centralized model. Each subproblem was solved through a metaheuristic based on swarm intelligence, called ACO (Optimization by Ant Colony). The results show that the decentralized model was efficient, where the union of its partial solutions provided the same solution of the centralized model applied to the same optimization problem.

Keywords: artificial satellites, visibility conflicts, Swarm Intelligence, ACO algorithm.

LISTA DE FIGURAS

Figura 1 - Conflito de visibilidade	14
Figura 2 - Configuração hipotética do sistema	15
Figura 3 - Otimização e Simulação de eventos	18
Figura 4 - Definição de segmento espacial e terrestre	22
Figura 5 - Evolução dos sistemas de telecomunicações	22
Figura 6 - Cobertura de serviços por satélite.....	23
Figura 7 - Constelação de satélites	25
Figura 8 - Sistema IoT	28
Figura 9 - Cubesat 2U	29
Figura 10 – RIBRAS.....	32
Figura 11 - Exemplo de grafo	34
Figura 12 - Grafos dos estados brasileiros.....	35
Figura 13 - Matriz adjacente.....	38
Figura 14 - Matrix incidente	39
Figura 15 - Caminho, cadeia, circuito, ciclo e árvore.....	41
Figura 16 - Exemplo de grafo direto	42
Figura 17 - Matriz incidente com valores negativos	44
Figura 18 - Fluxograma algoritmo PSO.....	55
Figura 19 – Comportamento individual e social das bactérias	57
Figura 20 – Tipos de comunicação no BCO.....	58
Figura 21 – Modelo do ciclo de vida da bactéria	59
Figura 22 - Arranjo metodológico da dissertação	62
Figura 23 - Gráfico Cenário 1	67
Figura 24 - Disputa entre estações 4 e 5	70
Figura 25 - Grafo PCV utilizado no Ant System	71
Figura 26 - Domínios de conflitos.....	73
Figura 27 - Exemplo de Grafo aplicado à alocação de estações	74
Figura 28 – Grafo do domínio de conflito D_5	75
Figura 29 - Gráfico do Cenário 1 após a otimização via solver	78
Figura 30 - Trajeto das formigas	81
Figura 31 - Solução dos Domínios de Conflitos	82
Figura 32 - Solução dos Domínios de Conflitos	84

LISTA DE QUADROS

Quadro 1 - Parâmetros do grafo.....	34
Quadro 2 - Exemplos de redes.....	36

LISTA DE TABELAS

Tabela 1- Exemplo de serviços e setores do lot.....	27
Tabela 2 – Nomenclaturas utilizadas no PSO	53
Tabela 3 - Dados Cenário 1	65
Tabela 4 - Dados Normalizados Cenário 1	65
Tabela 5 - Resultados do Solver para o cenário 1	77

LISTA DE SIGLAS

ACO – *Ant Colony Optimization*

AEB – Agência Espacial Brasileira

AG – Algoritmo Genético

AOS – *Acquisition of Signal*

AS – *Ant System*

BCO – *Bacterial Colony Optimization*

BF-PSO - *Bacterial Foraging Optimization oriented by Particle Swarm Optimization*

DDO - *Data Downlink Opportunitie*

EOS – *End of Signal*

GEO - *Geostationary orbit*

HEO - *HighEarthOrbit*

INPE – Instituto Nacional de Pesquisas Espaciais

IoT - *Internet of Things*

ISL - *Intersatellite link*

LEO - *Low Earth Orbit*

MEO - *Medium Earth Orbit*

PCV – Problema do Caixeiro Viajante

PEB – Programa Espacial Brasileiro

PID – Proporcional + Integral + Derivativo

PNL – Programação não Linear

PSO – *Particle Swarm Optimization*

RFID - Identificação por Rádio Frequência

TNL - *Terrestrial network link*

UDL - *Up/Downlink*

Sumário

LISTA DE FIGURAS	7
LISTA DE QUADROS	8
LISTA DE TABELAS	9
LISTA DE SIGLAS	10
1. INTRODUÇÃO	12
1.1. Contextualização	12
1.2. Problematização	13
1.3. Objetivo	16
1.3.1. Objetivo geral	16
1.3.2. Objetivos específicos	16
1.4. Justificativa	16
1.5. Estrutura do trabalho	19
2. COMUNICAÇÃO POR SATÉLITES	21
2.1. Evolução Histórica da Comunicação Via Satélite	24
2.2. Satélites de pequeno porte e Projeto QB-50	29
2.3. Rede Integrada Brasileira de estações de Rastreamento de Satélites ..	31
3. TEORIA DOS GRAFOS	33
3.1. Exemplos de redes representadas por grafos	35
3.2. Representações de grafos	37
3.3. Conceitos relevantes	40
3.4. Princípios da Otimização em Grafos	42
4. OTIMIZAÇÃO POR INTELIGÊNCIA DE ENXAMES	46
4.1. Otimização por colônia de formigas (ACO)	49
4.1.1. Algoritmo ACO e suas aplicações	50
4.2. Otimização por enxame de partículas (PSO)	52
4.2.1. Algoritmo de otimização por enxame de partículas	52
4.3. Otimização por colônia de bactérias (BCO)	56
4.4. Aplicações do algoritmo BCO e BFO	60
5. METODOLOGIA E DESENVOLVIMENTO	61
5.1. Modelagem	63
5.2. ADAPTAÇÃO DO ALGORITMO Ant-System (AS) AO PROBLEMA DE ALOCAÇÃO DAS ESTAÇÕES	71

6.	RESULTADOS E DISCUSSÃO	77
6.1.	RESULTADOS DO ALGORITMO ANT SYSTEM	78
6.1.1.	Resultados do teste do Ant System no PCV	78
6.1.2.	Resultados do Ant System Aplicado ao Problema de Alocação de Estações Terrestres	80
7.	CONCLUSÕES E CONSIDERAÇÕES FINAIS	86
8.	REFERÊNCIAS	88
	APÊNDICE A - ANÁLISE BIBLIOMÉTRICA “SWARM INTELLIGENCE”	93
	APÊNDICE B - CÓDIGO DO ALGORITMO ANT SYSTEM APLICADO AO PCV ...	99
	APÊNDICE C - CÓDIGO DO ALGORITMO ANT SYSTEM APLICADO AO DOMÍNIO DE CONFLITO D5	103

1. INTRODUÇÃO

O primeiro capítulo desse projeto de dissertação apresenta o tema de pesquisa, onde o objetivo principal é a construção de um modelo de compartilhamento em rede para estações terrestres de comunicação com nanosatélites e em seguida, a aplicação de algoritmos de otimização baseados em inteligência de enxames. Inicialmente é feita a contextualização do tema de maneira mais ampla, seguida da apresentação do problema de pesquisa, objetivos e justificativa. Encerrando a introdução, apresenta-se a estrutura do trabalho.

1.1. Contextualização

A sociedade atual é altamente dependente dos satélites. Independentemente da sua função, normalmente os satélites artificiais têm que trocar informações com a Terra. Devido a esse fato, as estações terrestres devem ser capazes de saber a localização dos satélites, enviar comandos e receber informações. Alguns exemplos das informações enviadas são: telemetria com o status do satélite, imagens capturadas e ou informações provenientes de outros satélites. Sem as estações terrestres os satélites se tornariam sem utilidade. As principais missões de satélites têm como finalidade observações da Terra, telecomunicações, e sistemas de localização global (SOUZA, 2013).

O programa espacial brasileiro (PEB) surgiu em 1960, quando foi instituída uma comissão de estudos que dez anos depois daria origem ao INPE (Instituto Nacional de Pesquisas Espaciais). Reúne militares, cientistas e técnicos que buscam conceder ao Brasil domínio da tecnologia de exploração do espaço (NETO, 2015).

Foi a partir da década de 1990 que o programa brasileiro começou a demonstrar resultados mais expressivos, com a criação da Agência Espacial Brasileira (AEB), em 1994, que passou a regular o PEB (RODRIGUES, 2015).

Porém, segundo Rodrigues (2015), o país apresentou uma evolução muito lenta nessa área quando comparada a países em desenvolvimento como a Índia e a China, que investiram em 2015 respectivamente US\$ 1 bilhão e US\$ 3 bilhões, enquanto o orçamento da AEB nesse mesmo ano foi de US\$ 122 milhões. A falta de investimentos e corte de recursos é um dos principais motivos dessa lentidão. Outro problema do programa espacial brasileiro são os acidentes. O mais grave foi em

2003, quando a explosão do terceiro protótipo do VSL-1 (Veículo Lançador de Satélites) fez 21 vítimas na base de Alcântara.

Uma saída promissora para a melhoria das pesquisas brasileiras na área aeroespacial é o investimento nos satélites de pequeno porte, denominados de *smallsats*, *cubesats* ou nanosatélites. Segundo Becanovic et al., (1999) esses pequenos satélites são ferramentas importantes de exploração e teste de novas ideias para missões espaciais, já que podem ser desenvolvidos com infraestrutura mais simples e baixo custo.

Um *cubesat* é um satélite em miniatura, com dimensões de 10x10x10 cm e peso de 1kg, e que oferece as funções padrões dos satélites de grande porte, como controle e determinação de atitude, *uplink* e *downlink* de telecomunicações, sistema de energia com bateria e painéis solares, sistema de processamento de dados, sensores e câmeras (QB50 PROJECT, 2016).

Muitas universidades distribuídas pelo mundo, inclusive no Brasil, vêm investindo em pesquisas com os *cubesats* com objetivo de desenvolvê-los, lançá-los e operá-los, já que os mesmos têm grande potencial para o desenvolvimento das tecnologias aeroespaciais.

1.2. Problematização

Existem vários projetos de *cubesats* em andamento no Brasil. Esse trabalho de dissertação vem contribuir com o do Instituto Federal Fluminense (IFF), em parceria com a Universidade do Porto (Portugal), contribuindo para a missão da Agência Espacial Europeia chamada QB50, que lançará 50 *cubesats* para medidas atmosféricas. De acordo com o QB50 Project (2016), os objetivos da missão QB50 são: facilitação de acesso ao espaço, pesquisas científicas, demonstração da tecnologia em órbita e integração entre universidades.

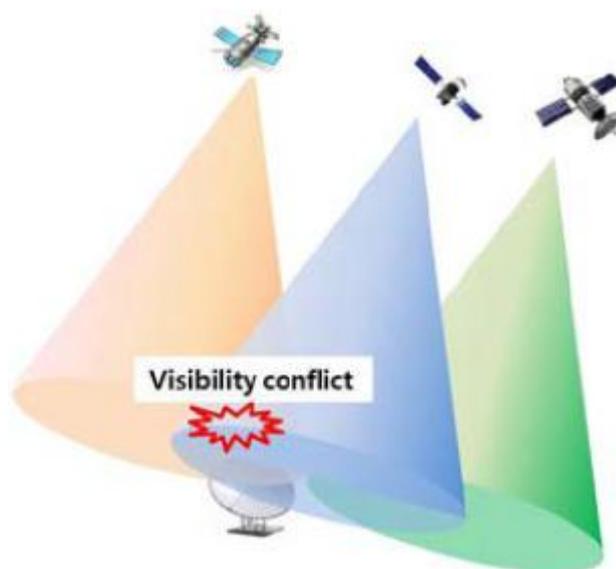
Normalmente, um único *cubesat* não é capaz de transportar muitos sensores, logo sua finalidade é mais educacional. No entanto, quando estes são organizados formando uma rede, as possibilidades são ampliadas, como o caso de monitoração atmosférica. Para essa rede de pesquisa, 50 nanosatélites *double-unit* (10x10x20 cm) estão previstos, os quais devem estabelecer comunicação com as estações terrestres, que deverão ser capazes de suportar uma constelação de satélites LEO (*Low Earth Orbit*) (QB50 PROJECT, 2016).

De acordo com Lee et al. (2013), quando um satélite está na linha de visão da antena terrestre, os mesmos têm a possibilidade de estabelecer uma comunicação. Os sinais são trocados durante este período de tempo, onde, o início da visibilidade é chamado de *acquisition of signal* (AOS), e o fim é chamado de *end of signal* (EOS). Durante esse intervalo de tempo o apontamento da antena deve ser adequado de modo a obter-se uma boa transmissão de dados.

Durante sua janela de visibilidade, cada um dos satélites tem a oportunidade de estabelecer um link de transmissão de dados, chamado de DDO (*Data Downlink Opportunite*). Alguns desses contatos podem ser livres de conflitos, enquanto outros não, o que exige ações apropriadas para contorna-los (CORRAO et al. 2012).

A Figura 1 mostra três satélites na linha de visão de uma antena da estação terrestre, caracterizando o conflito de visibilidade.

Figura 1 - Conflito de visibilidade

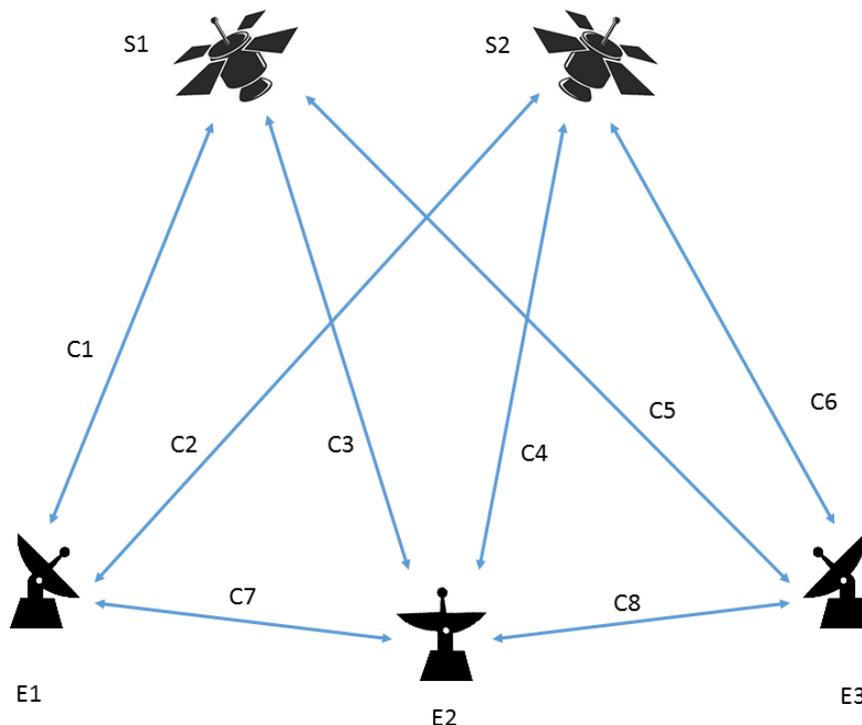


Fonte: Lee et al. (2013)

Cada satélite tem uma região de visibilidade em formato de cone, que depende da altitude e especificações da antena. O conflito de visibilidade ocorre quando dois ou mais satélites passam na linha de visão de uma antena terrestre simultaneamente, ou quando os satélites passam de forma sequenciada, porém o EOS de um satélite é mais rápido que o AOS do subsequente (SOMA et al., 2004).

A Figura 2 ilustra de forma simplificada o problema de comunicação e as possibilidades de caminhos diferentes de fluxo de dados para uma situação hipotética.

Figura 2 - Configuração hipotética do sistema



Fonte: Próprio Autor (2016)

A Figura 2 ilustra um possível cenário para o sistema, onde em determinado instante, dois satélites (S_1 e S_2) estão passando na linha de visão de três estações terrestres (E_1 , E_2 e E_3).

De acordo com Corrao et al. (2012) o objetivo de um sistema de planejamento de alocação de estações satélites é encontrar as melhores combinações de DDO's livres. O primeiro passo é obter a lista completa de AOS's e EOS's dos satélites a partir de um calculador de órbitas.

O problema consiste em determinar os melhores arranjos com links de comunicação (DDO's) livres de conflitos, que otimizem determinados parâmetros, como taxa de download de uma forma global, sujeitos as restrições do sistema, representados por uma função objetivo adequada.

O problema global já foi solucionado por Carvalho et al (2013), com um modelo de decisão centralizada.

O presente trabalho consiste em gerar um modelo descentralizado baseado em otimização linear e em teoria dos grafos. Em seguida aplicar o algoritmo de otimização baseado em inteligência de enxames, chamado de ACO (*Ant Colony Optimization*), para obter a solução ótima para o problema. O uso do algoritmo de formigas permite fazer a subdivisão do problema e resolver de forma negociada entre as estações.

1.3. Objetivo

Nessa seção são apresentados os objetivos geral e específicos da pesquisa.

1.3.1. Objetivo geral

O objetivo geral deste trabalho de pesquisa é solucionar o problema de alocação das estações terrestres para comunicação com satélites de forma descentralizada através do algoritmo ACO, maximizando a taxa de download e upload entre a estação terrestre e os nanosatélites.

1.3.2. Objetivos específicos

Os objetivos específicos são:

- Desenvolver um modelo centralizado que explique o compartilhamento de informações da rede de estações terrestres do programa de nanosatélites;
- Aplicar o método simplex ao modelo centralizado para obtenção da solução ótima global para problema;
- Desenvolver um modelo descentralizado, que permita que os conflitos sejam tratados entre as estações envolvidas, dividindo o problema geral em subproblemas;
- Desenvolver o código do algoritmo de otimização por colônia de formigas para a solução dos subproblemas;
- Aplicar o algoritmo de otimização ao problema de alocação das estações terrestres, de forma a obter as solução parciais.
- Comparar as soluções parciais com a solução obtida no modelo centralizado para verificar a viabilidade do modelo descentralizado.

1.4. Justificativa

Segundo Ávila (2002), a busca por eficiência se faz cada vez mais necessária nos dias atuais. A sociedade vive imersa em conceitos como desempenho, custos e

demais parâmetros que devem ser atingidos para que se consiga viver num meio extremamente competitivo.

Um exemplo clássico é o problema de transporte. Existem n produtores de determinado produto, que devem suprir a demanda de m mercados. Cada mercado possui uma demanda individual que precisa ser suprida. Existe um custo associado ao transporte de uma unidade de produto entre cada produtor e cada mercado. O problema consiste em suprir todos os mercados com um custo de transporte mínimo (FERGUSON; SARGENT, 1958).

De acordo com Gale (2007), a otimização linear pode ser definida como um problema de maximização ou minimização de uma função linear nas quais as variáveis devem satisfazer a um conjunto de restrições.

De acordo com as características do problema, podem-se dividir os métodos otimizadores em dois grandes grupos: programação linear e programação não linear. Para os problemas que são descritos por equações não lineares, utiliza-se a programação não linear (PNL), que pode ser dividida em três grandes famílias de métodos: os Determinísticos, os Estocásticos e os Enumerativos (ÁVILA, 2002).

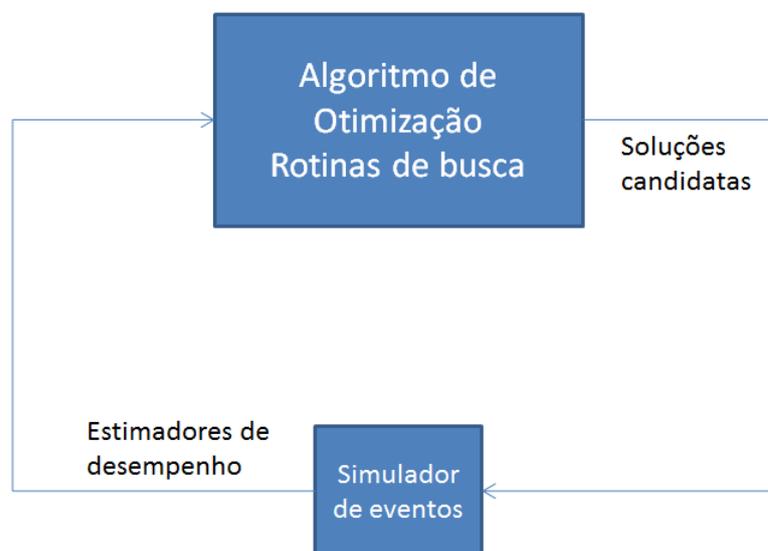
Nesse trabalho é proposta a utilização dos Algoritmos de Enxames, que são técnicas de otimização estocásticas. Os métodos estocásticos têm como principal característica a busca pelo ótimo através de regras de probabilidade. Principalmente quando a função objetivo é não linear ou não diferenciável, esses métodos são indicados (STORN; PRICE, 1995).

O uso da simulação proporciona a resolução de questões complexas sem os custos elevados das tentativas da vida real. Aliando-se otimização a simulação, pode-se assegurar que as soluções implementadas são ou estão próximas dos resultados ótimos (JUNIOR; FERREIRA, 2001).

Para Fu (2002), a otimização deve ser encarada como uma ferramenta complementar à simulação. Fornecendo as variáveis de uma possível solução (inputs) à simulação, e esta última, fornecendo as respostas (outputs) para a situação proposta. A otimização gera novas variáveis, utilizando técnicas de otimização específicas, que serão novamente testadas pela simulação.

A Figura 3 mostra um diagrama em blocos que ilustra a simulação associada a otimização.

Figura 3 - Otimização e Simulação de eventos



Fonte: Fu (2002)

O algoritmo de otimização fornece as candidatas a soluções do problema, que são as entradas para o modelo de simulação, do qual os parâmetros de saída servem como estimadores de performance dessas soluções (FU, 2002).

Ao longo da pesquisa foram encontrados alguns trabalhos que tratavam do problema de alocação em redes de estações terrestres, sendo que a maioria utiliza dos Algoritmos Genéticos (AG's) como ferramenta de otimização. Um exemplo é o trabalho de Corrao et al. (2012), que aplicou uma estratégia de otimização baseada na integração entre Algoritmos Genéticos, Teoria dos Grafos e Programação Linear.

A principal contribuição desse trabalho é o modelo descentralizado para a alocação das estações, onde as mesmas poderão negociar apenas com as estações envolvidas nos conflitos. O modelo não foi aplicado ao problema real porque até o presente momento os satélites do projeto QB 50 ainda não foram lançados. Então se utilizou de um cenário obtido por simulação computacional.

Carvalho et al. (2013) apresentaram um modelo centralizado para a alocação automática, que incluía também o controle de rastreamento das antenas. O problema é que esse modelo atende muito bem à Rede Integrada Brasileira de Rastreamento de Satélites (RIBRAS), onde apenas o centro de comando pode gerar os planos de rastreamento e enviar para todas as estações. Tratando-se da rede integrada com estações de outros países é necessário que ocorra essa negociação

entre os pares envolvidos naquele conflito, para que se chega a uma solução compartilhada.

Para solução dos subproblemas, utilizou-se o algoritmo ACO, porque dentro os três (ACO, PSO e BCO) apresentados na revisão sobre inteligência de enxames, ele é o mais recomendado para problemas com variáveis discretas. No modelo descentralizado as variáveis de solução são binárias, justificando a escolha desse algoritmo.

Este trabalho de pós-graduação justifica-se pela contribuição que o mesmo pode oferecer ao Centro de Referência em Educação, Pesquisa e Desenvolvimento em Sistemas Embarcados e Aeroespaciais do Instituto Federal Fluminense – IFF/Campos, onde o modelo poderá ser utilizado para integrar a rede de estações terrestres e estabelecer as regras de comunicação de forma a otimizar a transmissão dos dados entre estações e satélites. Além disso, este projeto pode ser de grande contribuição para a área de pesquisa aeroespacial brasileira, que ainda é tem muito a evoluir.

1.5. Estrutura do trabalho

Essa seção mostra de forma resumida a estrutura deste projeto de dissertação de mestrado.

O **capítulo 1** trouxe a introdução, apresentando a contextualização do tema, seguida do problema de pesquisa, objetivos geral e específicos, justificativa e a estrutura do trabalho.

O **capítulo 2** apresenta uma revisão bibliográfica sobre a comunicação via satélite e os mais recentes projetos envolvendo redes de satélite de pequeno porte, sistema que forma o objeto de estudo deste trabalho.

No **capítulo 3** são apresentados os fundamentos básicos sobre a teoria dos grafos.

O **capítulo 4** traz informações pertinentes aos algoritmos de enxames utilizados como ferramenta de otimização do modelo proposto.

O **capítulo 5** discorre sobre a metodologia utilizada para o desenvolvimento do presente projeto de dissertação.

O **capítulo 6** apresenta as informações relativas à continuação e ao cronograma de atividades da pesquisa.

O **capítulo 7** mostra as referências bibliográficas utilizadas até o presente momento neste projeto de dissertação.

O **apêndice A** mostra a análise bibliométrica realizada para levantar os artigos científicos que formaram o núcleo de partida para a construção das revisões bibliográficas.

O **apêndice B** apresenta o código do algoritmo de formigas para solução do problema do caixeiro viajante desenvolvido no software Matlab.

O **apêndice C** apresenta o código do algoritmo de formigas para solução do problema de alocação das estações terrestres.

2. COMUNICAÇÃO POR SATÉLITES

A comunicação via satélite pode ser considerada uma das maiores evoluções no que diz respeito a comunicação. Dentre os avanços, podemos citar, as contribuições para os estudos da área de geociências, meteorologia, telecomunicações, e transporte aéreo.

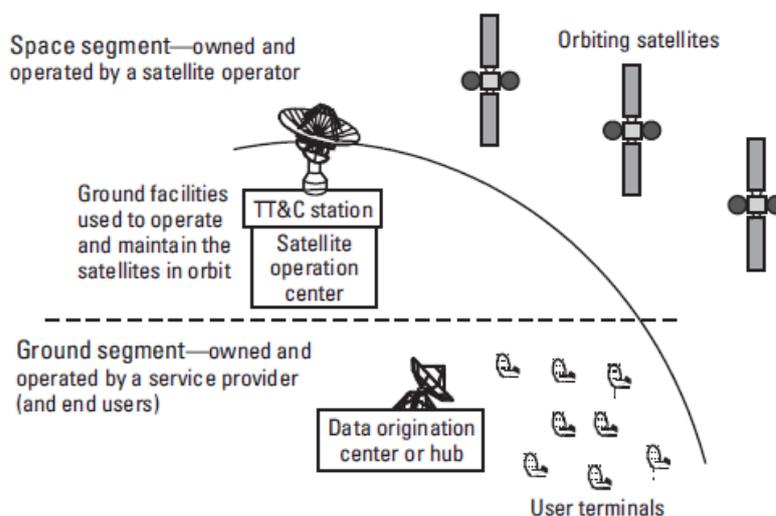
Em relação à transmissão, os satélites podem ser vistos como uma estação repetidoras de sinais que são recebidos da Terra. Em várias aplicações, o usual é que os satélites tenham que enviar informações para a Terra (SOUZA, 2003). Os mesmos recebem esses sinais, fazem seu deslocamento em frequência, amplificam e retransmitem para a terra. Um satélite artificial também pode ser utilizado para medições meteorológicas, pesquisas atmosféricas, telemetria, observação da terra e visualização do espaço, situação na qual, após a leitura das variáveis, os mesmos transmitem os dados via rádio frequência para as estações terrestres.

De acordo com Werner (1997), existem três segmentos que podem ser claramente separados no sistema de comunicação via satélites.

- i) *Intersatellite link (ISL)* compreende os *links* de radio entre pares de satélite, formando uma rede dinâmica no espaço;
- ii) *Up/downlink (UDL)* compreende os *links* entre satélites e estações terrestres;
- iii) *Terrestrial network link (TNL)* engloba as redes de estações terrestres e de dispositivos móveis.

O sistema pode ser separado em segmento espacial, formado pelo satélite ou conjunto de satélites e segmento terrestre, formado pelas estações terrestres e pelos terminais (ELBERT, 2001). A Figura 4 mostra esses dois segmentos.

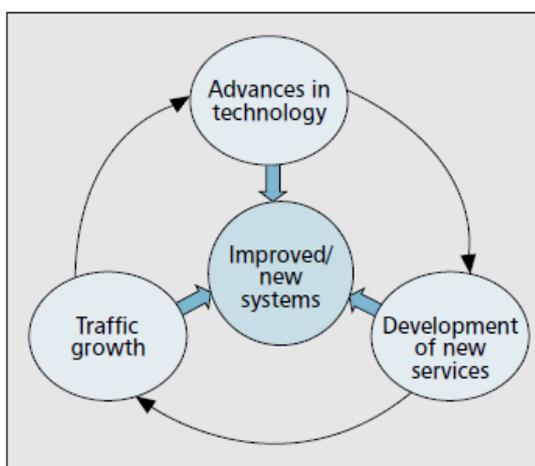
Figura 4 - Definição de segmento espacial e terrestre



Fonte: Elbert (2001)

É importante destacar o desenvolvimento tecnológico desses sistemas de comunicação nos últimos anos. Segundo Evans et al. (2005), pode-se atribuir a evolução dos sistemas de telecomunicação a três fatores: avanço tecnológico, desenvolvimento de novos serviços e crescimento de tráfego de dados nas redes, devido ao aumento de demanda dos consumidores. Os avanços tecnológicos, como processadores de alta performance e dispositivos de armazenamento de memória com tamanho reduzido, impulsionaram novos serviços de multimídia e baratearam preços para o consumidor final. A Figura 5 ilustra a interação desses três fatores.

Figura 5 - Evolução dos sistemas de telecomunicações

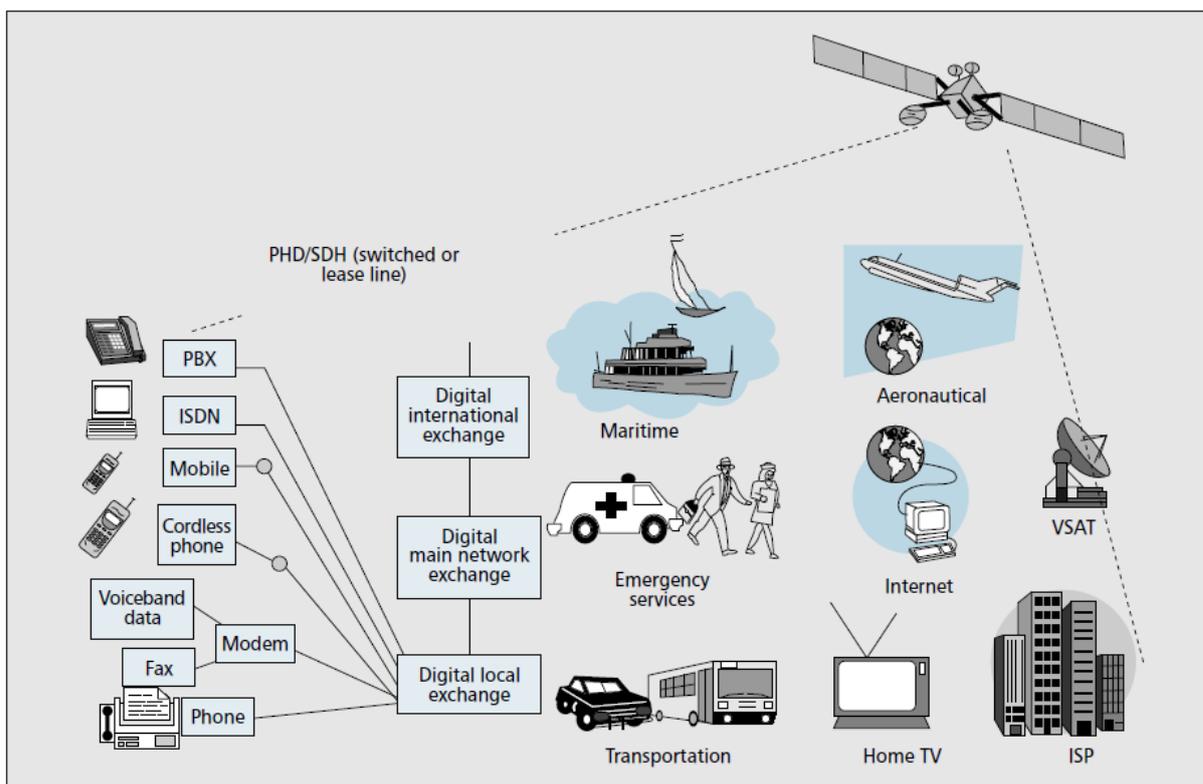


Autor: Evans et al. (2005)

A demanda tecnológica pode ser atendida através da integração de redes, especificamente de redes terrestres e de satélites. Os satélites possuem uma grande área de cobertura, o que garante um bom desempenho em serviços das áreas marinha e aeronáuticas. Com o crescimento da demanda tecnológica, os satélites também tiveram que melhorar, se tornando mais eficientes, explorando novas tecnologias, como processamento a bordo e antenas multifeixe. Um cenário promissor para utilização da comunicação via satélite é a telefonia móvel com as tecnologias 3G e 4G. Outra aplicação importante é na transmissão de sinais de TV, principalmente com o advento do sinal digital (EVANS et al.,2005).

A Figura 6 apresenta a flexibilidade dos sistemas de comunicação por satélite em atender uma grande variedade de serviços de multimídia.

Figura 6 - Cobertura de serviços por satélite



Autor: Evans et al. (2005)

Evans et al. (2005) apresentam diversas aplicações para a comunicação via satélite, das quais destaca: aeronáutica, marinha, serviços de emergência, internet, transporte, sinais de TV, etc.

2.1. Evolução Histórica da Comunicação Via Satélite

O primeiro satélite artificial foi colocado em órbita na época da Guerra Fria, quando em 1957 a União Soviética lançou o Sputnik, que transmitiu sinais de telemetria por 21 dias. Alguns meses depois em 1958 foi a vez dos americanos lançarem seu primeiro satélite, denominado Explorer 1 que transmitiu sinais por aproximadamente cinco meses (HAYKIN, 2000).

Em 1976, três satélites para comunicação marítima foram lançados, formando o MARSAT. Com o lançamento de novos satélites, foi possível a expansão da rede, e em 1979 formaram o INMARSAT. Em 1990 – 1991, INMARSAT adicionou serviços aeronáuticos para aeronaves de passageiros e alguns veículos terrestres com a introdução de satélites com poder maior de transmissão multifeixe (EVANS et al., 2005).

A rede INMARSAT concentrou-se no uso de satélites (GEO) geoestacionários, e nos meados dos anos 90 vários sistemas GEO regionais surgiram em concorrência, como por exemplo, OmniTRACS e REALTRACS, concentrando-se em veículos terrestres. O sucesso desses foi apenas moderado em relação a INMARSAT que construiu sua base de clientes com cerca de 250.000. A principal investigação nos anos 80 e início dos anos 90 foi direcionado para constelações não-Geoestacionária (EVANS et al., 2005).

Em 1994, constelação de satélites e os equipamentos de suporte terrestre da rede NAVSTAR – GPS foram declarados operacionais. O Sistema de Posicionamento Global é uma ampla rede mundial de satélites que fornecem dados de latitude, longitude, altitude e tempo. Foi desenvolvido pelo Departamento de Defesa dos Estados Unidos para ser o principal sistema de navegação das forças armadas (BRADFORD W. PARKINSON; JAMES J. SPILKER JR.; PENINA AXELRAD, 1996).

Uma discussão que se fez presente ao longo da evolução histórica dos sistemas de comunicação via satélite foi a questão das órbitas. De acordo com Nassif (2001), os satélites podem se dividir em três categorias em relação à sua órbita:

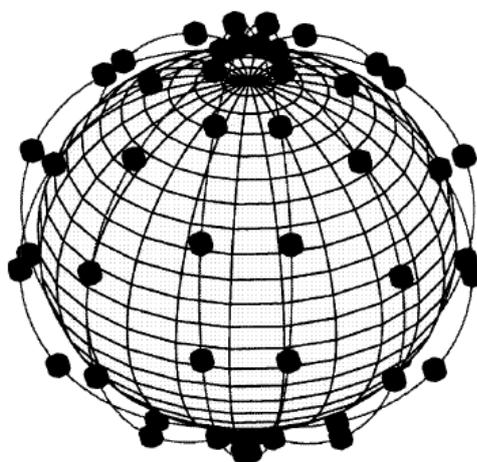
- LEO (*Low Earth Orbit*): De 700 a 1500 km da superfície da terra;
- MEO (*Medium Earth Orbit*): De 10000 a 20000 km;

- HEO (*HighEarthOrbit*): a partir de 20000 km (onde se inclui GEO: (*Geostationary orbit*) Aproximadamente 36000 km).

Nas órbitas LEO os satélites têm que viajar em maiores velocidades quando comparadas as demais órbitas. Sua velocidade angular deve ser maior do que a da Terra para que a força gravitacional não os atraia para dentro da atmosfera. As pesquisas voltadas para órbitas baixas se intensificaram, pois são muito utilizadas para telecomunicações, tornando possível reduzir tamanho das antenas, já que a potência de transmissão pode ser menor (NASSIF, 2001).

Segundo Werner (1997), todos os sistemas LEO e MEO são baseados em constelações com órbitas baixas entre 700 e 1400 Km e órbitas médias de 10000 Km de altitude. Todas as órbitas têm a mesma inclinação em relação ao plano do equador, e em cada uma delas circulam o mesmo número de satélites. Cada satélite possui sua área de cobertura, que somadas, cobrem todo o globo terrestre. A Figura 7 mostra uma possível geometria para a constelação de satélites.

Figura 7 - Constelação de satélites



Autor: Werner (1997)

Já os satélites em órbitas geoestacionárias possuem mesma frequência angular da Terra, fazendo com que um observador no referencial da Terra mantenha a mesma posição relativa ao satélite. Essa órbita se dá aproximadamente a exatos 42164 Km do centro da Terra (aproximadamente 36000 Km de altitude da superfície) e tem sua maior utilização como meio de transmissão de sinais de televisão. De modo geral, satélites para telecomunicações situam-se em órbitas geoestacionárias (GEO), satélites para observação voam em órbitas baixas (LEO), e satélites para localização encontram-se em orbitas médias (RIEBEEK, 2009).

Alguns trabalhos na década de 90 estavam voltados para a utilização dos satélites em órbitas baixas ou médias para serviços de telecomunicação. Werner (1997) propôs um conceito de roteamento dinâmico para constelações de satélites LEO e MEO, os quais considerou como principais candidatos a infraestrutura de comunicação pessoal. Sua justificativa para esses tipos de satélites era o menor *delay* de comunicação entre estação terrestre e segmento espacial. Para uma ampla cobertura seriam necessários mais satélites, formando uma rede global, chamada de *ISL's (intersatellite links) networking*.

Outro trabalho voltado para sistemas de satélites LEO e MEO foi desenvolvido por Werner et al. (1995), onde destacava que muitos esforços estavam voltados para o desenvolvimento e implantação dessas redes, porém, destacava que muitos problemas precisavam ser resolvidos para sua operação final.

De acordo com Wu et al. (1994), nos anos de 1960 houve forte discussão sobre qual órbita seria mais adequada para a comunicação móvel, onde a conveniência das órbitas GEO foram pesadas contra a dificuldade técnica para a lançamento dos satélites em grandes altitudes, e pelo fato de as órbitas LEO apresentarem vantagens técnica inerentes, como o menor atraso de comunicação .

Nesse contexto, o autor destacou duas principais objeções que haviam em relação as órbitas geoestacionárias. Como a altitude da órbita é de 35786 Km, o tempo de comunicação, levando em conta o uplink e o downlink é de cerca de 0.6 segundos. Em transmissão de voz, por exemplo, isso é um atraso considerável. A segunda objeção fundamental para a GEO é a falta de cobertura para latitudes mais próximas ou Norte e ao Sul. O limite prático é em cerca de 75° e um efeito grave é o bloqueio simples causado por edifícios altos em áreas urbanas.

Por outro lado, ângulos altos de elevação em órbita de constelações LEO são facilmente alcançados. Essa discussão em relação a melhor aplicabilidade das órbitas sofreu uma evolução cíclica, onde em épocas diferentes, uma órbita foi considerada melhor que as demais para esse tipo de aplicação (WU et al., 1994).

Um estudo atual foi realizado por Sanctis et al. (2016), no qual apresenta uma aplicação mais recente para as tecnologias de comunicação via satélite. Seu artigo foca na aplicação do sistema de comunicação por satélite no apoio à *Internet of Things* (IoT). O autor considera esse termo como a comunicação entre sensores e atuadores, como o paradigma de receber os dados de sensores ou *tags* de RFID

(identificação por rádio frequência) e para enviar sinais de controle aos atuadores. Nesse contexto, os satélites podem servir como rede complementar para interligar esses dispositivos Inteligentes em áreas remotas.

A Tabela 1 mostra exemplo de setores e serviços que utilizam a tecnologia IoT.

Tabela 1- Exemplo de serviços e setores do IoT

Setor de Serviço	Grupo de aplicação	Localização	Dispositivos
Construção	Comercial, institucional, industrial	Escritórios, escolas, hospitais, aeroportos e estádios	Iluminação, sistemas de segurança e de combate à incêndios
Energia	Geração, distribuição e demanda	Energia solar, eólica, linhas de transmissão e distribuição	Turbinas, aerogeradores, baterias, medidores de potência
Saúde	Cuidado público, cuidado doméstico	Hospitais e residências	Equipamentos médicos
Indústria	Controle de processos, automação de processos	Plantas de controle de processos, controle de manufatura	Bombas, válvulas, motores, sensores
Transporte	Terrestre, aéreo e marítimo	Terra, céu, mar, rodovias	Carros, embarcações, aviões
Varejo	Lojas, hotelaria, entretenimento	Cinema, teatro, posto de combustível, supermercado, shopping center, hotel, restaurante	Máquinas de venda, Tags, registradores de caixa
Segurança pública	Serviços de emergência e etc	Nível nacional, nível municipal	Carros, ambulâncias
Redes	Publica, empresarial e e-commerce	Data centers, provedores de serviços, redes de telecomunicações	Servidores, roteadores, computadores, Tablets e smartphones

Fonte: Adaptado de Sanctis et al. (2016)

A Figura 8 mostra as três camadas que forma um sistema de IoT. A primeira camada é da operação e gestão da internet, a segunda é das tecnologias de acesso, e a camada mais baixa é a dos dispositivos (sensores e atuadores).

Figura 8 - Sistema IoT



Fonte: Sanctis et al. (2016)

O uso da comunicação via satélite, nesse caso, se justifica quando os sensores estão localizados em áreas remotas, onde redes convencionais não alcançam. Os dois principais fatores favoráveis são: a interoperabilidade existente entre sistemas de satélites e sensores/atuadores (SANCTIS et al., 2016).

Segundo o autor, a comunicação por satélite fornece uma solução com melhor custo benefício em relação a outras tecnologias terrestres para a interligação e comunicação com os dispositivos inteligentes quando estão dispersos remotamente. Nesses casos o satélite pode facilmente realizar uma transmissão em *broadcast* (quando todos os nós da rede precisam receber a mesma mensagem), ou *multicast* (quando uma parte dos nós precisa receber a mesma mensagem). A comunicação IoT via satélite é uma realidade, fornecendo uma integração completa, o que representa uma boa oportunidade para os mercados de satélite.

É possível concluir que as aplicações dos satélites como meio de comunicação evoluiu muito aos longos dos anos, impulsionado principalmente pela evolução tecnológica.

2.2. Satélites de pequeno porte e Projeto QB-50

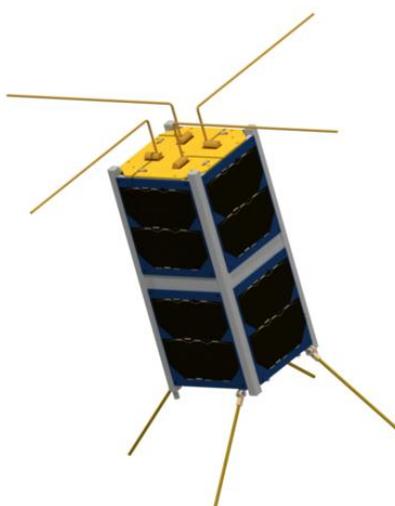
A partir da década de 1980, um novo paradigma no campo das missões espaciais surgiu com o objetivo de reduzir os custos envolvidos com a área aeroespacial. As tendências tecnológicas permitiram o desenvolvimento de satélites de pequeno porte, considerados assim, quando possuem sua massa entre 1 e 500 Kg (XUE et al., 2008).

De acordo com sua massa, os satélites podem ser classificados como:

- Satélites grandes: Massa > 1000 kg;
- Satélites médios: Massa 500 – 1000 Kg;
- Mini-satélites: Massa 100 – 500 Kg;
- Micro-satélites: Massa 10 – 100 Kg;
- Nano-satélites: 1 – 10 Kg;
- Pico-satélites: 0.1 – 1 Kg;
- Femto-satélites: Massa < 100g.

Os nano-satélites, em particular, têm despertado o interesse de muitas instituições educacionais que querem desenvolver pesquisas espaciais com um baixo custo, como é o caso do projeto QB-50, que utiliza satélites conhecidos como cubesats. Segundo Gavrilovich et al. (2015), os cubesats são satélites miniaturizados com tamanho padrão de 10 x 10 x 10 mm, formando uma unidade (1U), ou múltiplos de uma unidade, como por exemplo os cubesats 2U ou 3U. A Figura 9 mostra um cubesats 2U, que possui a forma de dois 1U's juntos.

Figura 9 - Cubesat 2U



Fonte: (QB50 PROJECT, 2016)

Muitos trabalhos recentes relacionados aos cubesats foram desenvolvidos nos últimos anos, impulsionados principalmente pelo baixo custo associados aos satélites de pequeno porte, quando comparados com os de grande porte. Segundo Xue et al. (2008), micro e nano-satélites são ferramentas importantes para a exploração e teste de novas ideias e de novos dispositivos com a necessidade de menos recursos financeiros.

Um artigo recente foi desenvolvido por Guertin et al. (2015) que realizou testes de radiação nos microcontroladores e microprocessadores dos *cubesats*. Sanchez et al. (2016) fizeram uma comparação entre a energia solar incidente e a carga da bateria num 3U *cubesat*. Esses exemplos mostram que as missões espaciais e pesquisas envolvendo o desenvolvimento das tecnologias empregadas nos *small sats* têm despertado interesse de muitos pesquisadores.

Os satélites de pequeno porte também possuem algumas desvantagens, como concluíram Xue et al. (2008). Além das vantagens já destacadas, como baixo custo, rapidez no desenvolvimento, simplicidade construtiva e custo-benefício, esses pequenos satélites desvantagens, como, pequena área de superfície, que resulta em restrições de carga, restrições térmicas e restrições de potência. O espaço para painéis solares também é bem reduzido.

Uma saída para as desvantagens apresentadas é a utilização de um conjunto de *cubesats*, formando uma rede, como é feito no projeto QB-50. As redes de pequenos satélites estão sob discussão na comunidade acadêmica a alguns anos, sendo que nenhuma universidade ou agência espacial, havia tomado uma iniciativa de criação e coordenação de uma rede tão poderosa como a QB-50 (QB50 PROJECT, 2016).

O principal objetivo da missão QB-50 é demonstrar a possibilidade de lançamento a um baixo custo de uma rede 50 *cubesats* construídos por universidades de todo o mundo para realizar pesquisas na termosfera.

Um dos objetivos parciais do projeto é conseguir uma forma de acesso sustentado e disponível ao espaço, para missões espaciais de pesquisa de pequena escala.

Outro objetivo do projeto QB50 é a realização de pesquisa atmosférica dentro da termosfera inferior, entre 200 - 380 km de altitude, que é a camada menos explorada da atmosfera. QB50 irá fornecer medições de multi-ponto, por um período de tempo de alguns meses, em vez de minutos, como foi feito anteriormente testes

anteriores. Para essa pesquisa, três tipos diferentes de sensores atmosféricos serão distribuídos entre os satélites.

O terceiro objetivo do Projeto QB50 é funcionar como uma plataforma para demonstração e desenvolvimento tecnológico voltado para missões espaciais.

O último objetivo tem caráter educacional. Diversas universidades de todo o mundo foram convidadas para participar do projeto e enviar um satélite ao espaço. Então, os *cubesats* serão projetados e construídos por um grande número de jovens estudantes de engenharias que são coordenados por um corpo docente de suas universidades e guiados pelo Projeto QB50.

2.3. Rede Integrada Brasileira de estações de Rastreamento de Satélites

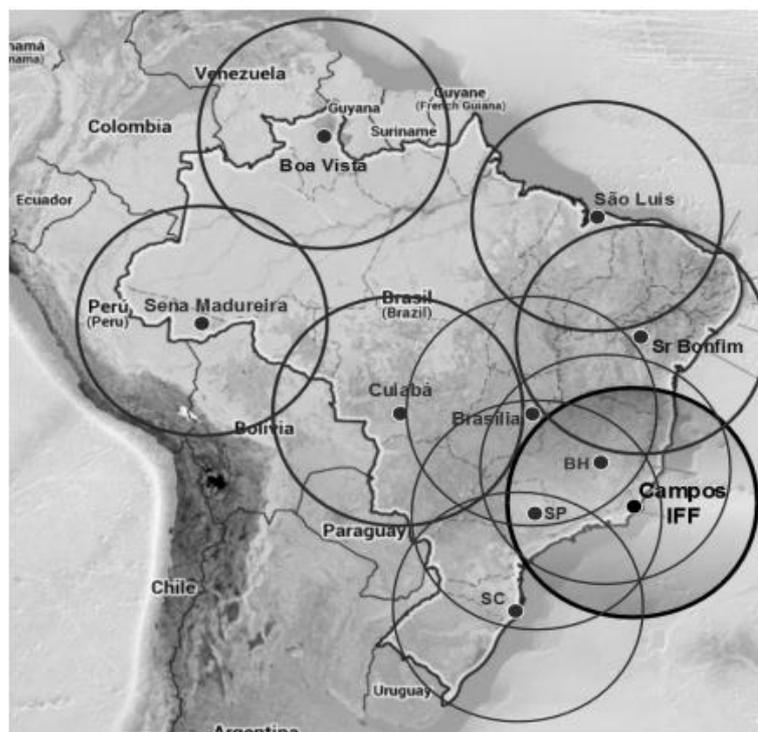
A RIBRAS (Rede Integrada Brasileira de Estações de Rastreamento de Satélites) é um conjunto de softwares destinados a controlar a rede de estações terrestres para receber dados de satélites durante suas passagens sobre o território brasileiro. As estações trabalham de forma sincronizada, seguindo planos de rastreamentos gerados previamente, com o objetivo de obter a maior quantidade de dados durante esses voos (Carvalho et al., 2014).

Para aplicação dos modelos gerados nessa dissertação foi utilizado um cenário de passagens de satélites baseados na RIBRAS.

De acordo com Carvalho et al. (2014), o CRSEA (Centro de Referência em Sistemas Embarcados e Aeroespaciais) do Instituto Federal Fluminense (IFF, Brasil) tomou a iniciativa no desenvolvimento dessa rede para abranger a maior parte do território brasileiro, estabelecendo comunicação com satélites em LEO. Essas redes precisam ser automatizadas para a comunicação ótima com essa constelação de satélites.

A Figura 10 mostra a distribuição das estações de satélites e as pegadas simuladas de satélites a 200 Km de altitude e 12° de elevação.

Figura 10 – RIBRAS



Fonte: Carvalho et al. (2014)

Cada estação é composta por duas antenas que são movimentadas por motores industriais capazes de movimentá-las de forma suave permitindo que os satélites sejam rastreados.

3. TEORIA DOS GRAFOS

Este capítulo procura mostrar os conceitos básicos sobre a teoria dos grafos, destacando alguns exemplos e os principais métodos de otimização aplicados a problemas dessa natureza.

De acordo com Steen (2010), a Teoria dos Grafos é um campo da matemática que ganhou muita popularidade nos nas últimas décadas, devido principalmente, a sua capacidade de descrever fenômenos de diferentes áreas, como infraestrutura de comunicações, desenho de mapas, agendamento de tarefas e estruturas sociais.

Os grafos são as representações matemáticas de redes que descrevem diversos problemas ou fenômenos. Esse fato motivou a utilização dessa ferramenta matemática como parte desse projeto de dissertação, onde os grafos podem ser utilizados para descrever a estrutura de comunicação da rede de nanosatélites. De uma forma simples, um grafo é um conjunto de vértices ou nós, conectados por linhas, denominadas como arestas ou arcos, onde cada aresta conecta apenas dois vértices.

Uma definição mais formal é dada por Hofstad (2016): um grafo $G = (V, E)$ consiste numa coleção ou conjunto de vértices V e uma coleção de bordas ou arestas E . Os vértices indicam os objetos do modelo e as arestas correspondem às interações entre pares de vértices.

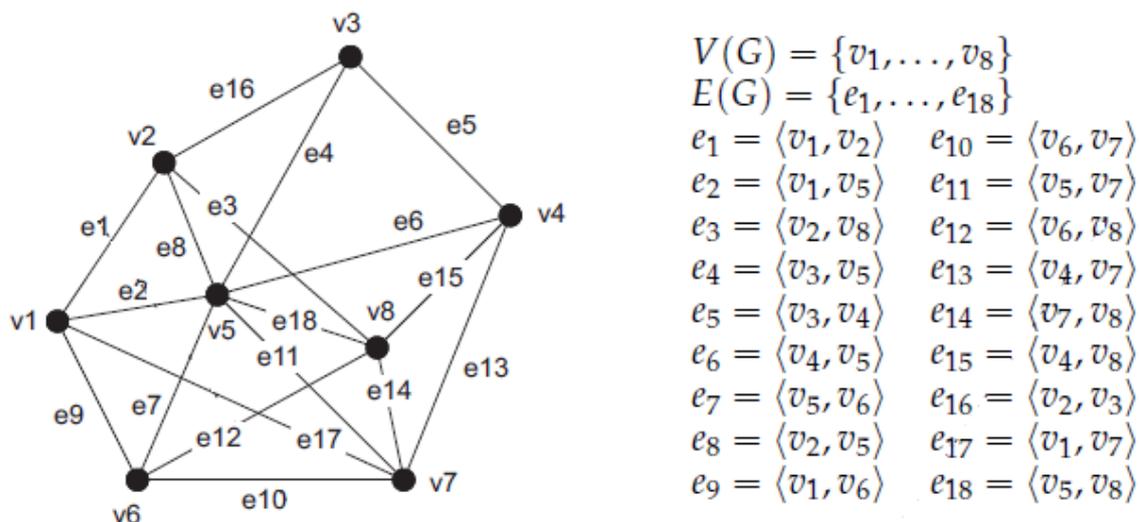
Cada aresta $e \in E$ é dito como uma junta para dois vértices, que são seus pontos finais. Se e une $v, u \in V$, escreve-se $e = \langle u, v \rangle$. Nesse caso, os vértices u e v são chamados de **adjacentes**. A aresta e é dita como **incidente** aos vértices u e v , respectivamente (STEEN, 2010).

É importante destacar que o E vem do inglês *edge*. Em português pode-se utilizar da letra A para descrever o conjunto de arestas. Um grafo pode ter uma aresta unindo o mesmo vértice, formando o que é chamado de *loop* ou laço. Ou ainda, possuir dois vértices sendo unidos por um conjunto de arestas que tem os mesmos pontos finais. Quando um grafo não apresenta nenhuma dessas situações, esse é chamado de grafo simples.

Existe ainda, mesmo sem ter uma aplicação prática, o grafo vazio, o qual não possui nenhum vértice ou aresta. Um outro tipo especial é o grafo completo, que ocorre quando o grafo tem um número n de vértices, com cada vértice sendo adjacente a todos os outros. Nesse caso a rede é chamada de grafo completo.

A Figura 11 mostra um exemplo de grafo com 8 vértices e 18 arestas.

Figura 11 - Exemplo de grafo



Fonte: Steen (2010)

Cada vértice é representado, nesse caso, por um ponto escuro e cada aresta por uma linha. Além disso, cada elemento possui uma legenda, indicando qual a numeração das arestas e vértices, como é possível visualizar na parte esquerda do grafo. O Quadro 1 completa o entendimento do grafo acima.

Quadro 1 - Parâmetros do grafo

Vértice	Grau	Arestas incidentes	Vizinhos
v_1	4	$\langle v_1, v_2 \rangle, \langle v_1, v_5 \rangle, \langle v_1, v_6 \rangle, \langle v_1, v_7 \rangle$	v_2, v_5, v_6, v_7
v_2	4	$\langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_2, v_5 \rangle, \langle v_2, v_8 \rangle$	v_1, v_3, v_5, v_8
v_3	3	$\langle v_2, v_3 \rangle, \langle v_3, v_4 \rangle, \langle v_3, v_5 \rangle$	v_2, v_4, v_5
v_4	4	$\langle v_3, v_4 \rangle, \langle v_4, v_5 \rangle, \langle v_4, v_7 \rangle, \langle v_4, v_8 \rangle$	v_3, v_5, v_7, v_8
v_5	7	$\langle v_1, v_5 \rangle, \langle v_2, v_5 \rangle, \langle v_3, v_5 \rangle, \langle v_4, v_5 \rangle, \langle v_5, v_6 \rangle, \langle v_5, v_7 \rangle, \langle v_5, v_8 \rangle$	$v_1, v_2, v_3, v_4, v_6, v_7, v_8$
v_6	4	$\langle v_1, v_6 \rangle, \langle v_5, v_6 \rangle, \langle v_6, v_7 \rangle, \langle v_6, v_8 \rangle$	v_1, v_5, v_7, v_8
v_7	5	$\langle v_1, v_7 \rangle, \langle v_4, v_7 \rangle, \langle v_5, v_7 \rangle, \langle v_6, v_7 \rangle, \langle v_7, v_8 \rangle$	v_1, v_4, v_5, v_6, v_8
v_8	5	$\langle v_2, v_8 \rangle, \langle v_4, v_8 \rangle, \langle v_5, v_8 \rangle, \langle v_6, v_8 \rangle, \langle v_7, v_8 \rangle$	v_2, v_4, v_5, v_6, v_7

Fonte: Adaptado de Steen (2010)

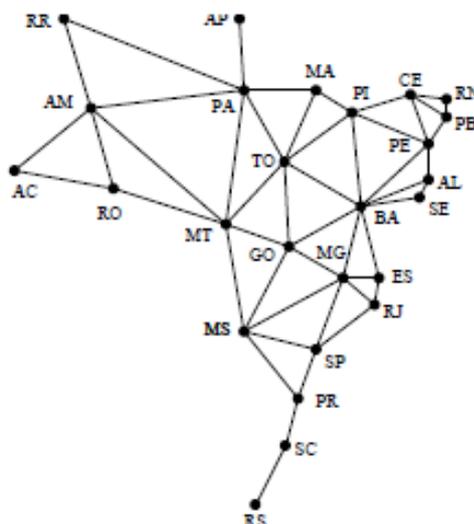
No Quadro 1 é possível verificar o número de arestas incidentes em cada vértice, o que é chamado de grau do vértice. Além disso, verifica-se os vizinhos de cada um dos vértices ou vértices adjacentes.

3.1. Exemplos de redes representadas por grafos

Como dito anteriormente, um grafo tem aplicações em diversas áreas de estudo. Os grafos constituem-se num bom modelo para muitos problemas em várias áreas da matemática, da informática, da engenharia e da indústria, tendo importantes aplicações práticas (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

Um exemplo simples de grafo dado por Feofiloff et al. (2011) é a representação dos estados do Brasil, onde cada vértice é um estado da República Federativa do Brasil e as arestas são as fronteiras entre eles. Dois estados são adjacentes se têm uma fronteira comum. A Figura 12 mostra esse grafo.

Figura 12 - Grafos dos estados brasileiros



Fonte: Feofiloff et al. (2011)

A internet é um exemplo de rede complexa que normalmente é utilizada como referência para estes tipos de grafos. Segundo Steen (2010), é possível definir a internet como um grande conjunto de dados interconectados. Um nó pode ser uma pessoa, uma empresa, um computador, ou qualquer dispositivo conectado à rede.

As arestas são os links estabelecidos pelas conexões físicas, seja por cabo ou sem fio.

Outro exemplo são as redes que envolvem pessoas, que são as modernas redes sociais, formando as comunidades online. As redes sociais tem sido objeto de estudo de muitos pesquisadores nos últimos anos, o que destaca sua relevância como rede complexa. Porém, antes da popularização da Internet, pesquisadores já estudavam as estruturas de grupos de pessoas. Na década de 1930 foram introduzidos os sociogramas, representações gráficas de uma rede onde as pessoas são os nós e as relações interpessoais são as arestas. Anos mais tarde, os sociogramas foram formalizados em grafos (STEEN, 2010).

O Quadro 2 mostra uma síntese de diferentes tipos de redes na qual a Teoria dos Grafos é aplicada.

Quadro 2 - Exemplos de redes

Rede	Vértices	Arestas	Descrição
Transporte aéreo	Aeroportos	Voos	Voos regulares entre os aeroportos
Planejamento de ruas	Esquinas	Segmentos de ruas	Um trecho de estrada estende-se exatamente entre duas esquinas. Uma variação é distinguir entre os segmentos de sentido único e em dois sentidos.
Transporte por trem	Estações	Conexões	Duas estações só são conectadas se houver uma conexão de trem agendada que passa sem parar por quaisquer estações intermediárias.
Cérebro	Neurônios	Sinapses	Cada neurônio pode ser considerado como um elemento com entradas de sinais, chamados dendritos e saídas, chamados de axônios. As sinapses carregam sinais elétricos entre os neurônios.
Colônia de formigas	Junções	Trilhas de feromônio	As formigas depositam uma substância química chamada de feromônio quando se movimentam em busca de alimentos. A medida que muitas formigas se deslocam por um mesmo caminho a trilha de feromônio fica mais forte e atrai

			mais formigas.
Rede de citações	Autores	Citações	Na literatura científica, é comum fazer referência a trabalhos relacionados publicados por outros autores. Esse trabalho pode ser citado por outros autores, formando assim uma rede.
Chamadas de telefones	Números de telefones	Chamadas	Redes de telefonemas se referem a pares de pessoas trocando informações, formando assim uma rede social tecnicamente representada por números de telefone e as chamadas reais.
Rede de reputação	Pessoas	Classificação	Em redes de negociação eletrônica, tais como e-Bay, transações de taxa de compradores. Como compradores, por sua vez, também pode ser vendedor, obtém-se uma rede em que as taxas refletem a reputação entre as pessoas.

Autor: adaptado de Steen (2010)

O Quadro 2 mostrou apenas alguns entre os inúmeros exemplos de redes de comunicações. A próxima seção demonstra como são as representações matemáticas para os grafos.

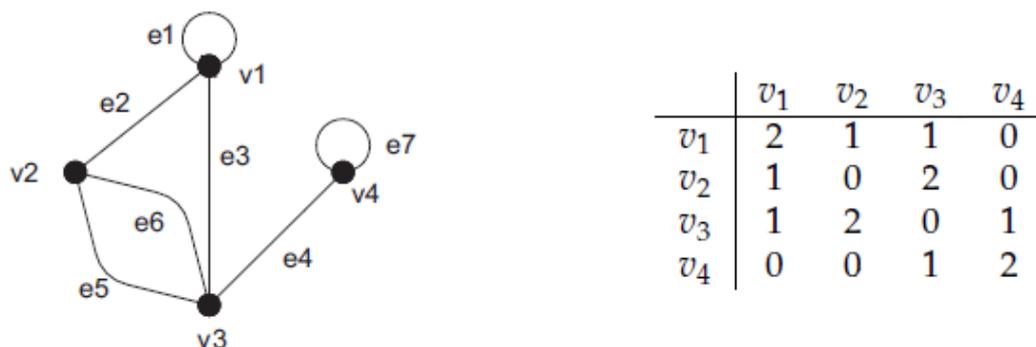
3.2. Representações de grafos

Ficou claro até aqui que os inúmeros tipos de problemas nos quais se aplicam a teoria dos grafos, podem ser modelados como um grafo genérico em função dos seus vértices e arestas. Essa seção mostra a representação dos grafos em forma matemática para permitir o processamento automático via computadores.

Existem diversas formas de representar um grafo, sendo que as mais utilizadas são através de matrizes, como verificado em alguns artigos relacionados. Alguns exemplos destes são demonstrados nos trabalhos de Corrao et al. (2012), Cho e Byun (2016), e Gong et al. (2016).

Uma das formas de representação se dá através da matriz adjacente, como explica Steen (2010). Um exemplo de grafo e sua respectiva matriz adjacente é representado na Figura 13.

Figura 13 - Matriz adjacente



Fonte: Steen (2010)

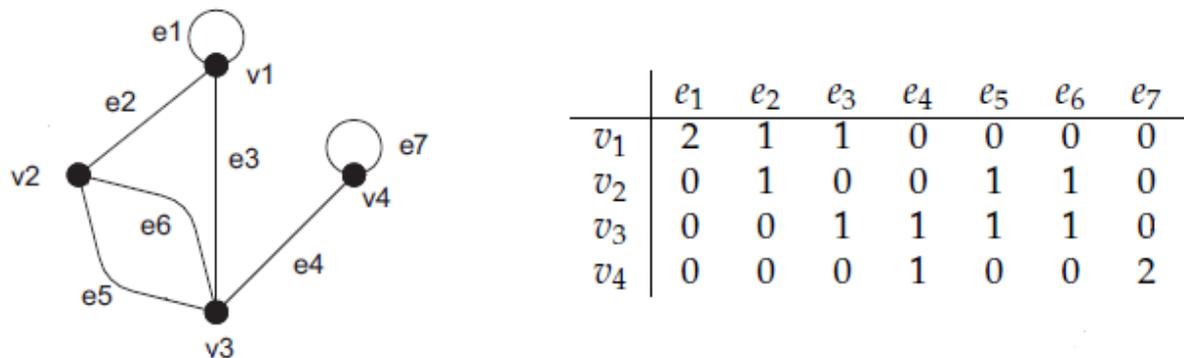
De forma genérica, um grafo G , com n vértices e m arestas, possui uma matriz adjacente A com n linhas e n colunas. Cada elemento $A[i, j]$ representa o número de arestas que ligam o vértice v_i ou v_j . Na Figura 13 é possível verificar que a matriz adjacente gerada possui 4 linhas e 4 colunas, já que o grafo possui 4 vértices.

As propriedades dessa matriz são:

- Para todo i, j , $A[i, j] = A[j, i]$, ou seja, a matriz é simétrica.
- Um grafo G é considerado simples se e apenas se para todo i, j , $A[i, j] \leq 1$ e $A[i, i] = 0$. O grafo da Figura 13, não é um grafo simples, pois ele possui duas juntas ligando os vértices v_2 e v_3 , e possui um loop no vértice v_1 .
- $\delta(v_i) = \sum_{j=1}^n A[i, j]$, isto é, o grau do vértice v_i é igual ao somatório dos valores da linha i .

De forma semelhante, existe a representação de um grafo através de sua matriz incidente. O mesmo grafo da Figura 13 é mostrado na Figura 13, agora com sua matriz incidente.

Figura 14 - Matrix incidente



Fonte: Steen (2010)

Segundo Steen (2010), a matriz incidente M que representa um grafo G possui n linhas e m colunas, e cada elemento $M[i, j]$ contém o número de vezes que a aresta e_j é incidente ao vértice v_i . Uma observação importante nesse caso, é que os elementos da matriz incidente só podem assumir os valores 0, 1 ou 2.

Suas propriedades são:

- O grafo G não possui nenhum loop se para todo $i, j, M[i, j] \leq 1$.
- O somatório dos elementos da linha i é igual ao grau do vértice v_i , isto é, $\forall i : \delta(v_i) = \sum_{j=1}^m M[i, j]$.
- Como cada aresta possui apenas dois *end points*, o somatório de cada coluna é igual a 2. Para toda coluna $j, \sum_{i=1}^n M[i, j] = 2$.

Apesar de serem formas simples e eficientes de representar um grafo, a matriz adjacente e a matriz incidente não são adequadas quando o grafo é muito grande, ou seja, possui muitos nós. Como elas precisam armazenar $n \times n$ ou $n \times m$ elementos, caso o grafo possua centenas de milhares de vértices, uma matriz desse tipo pode precisar de mais de 10 Gb de espaço de armazenamento num computador.

Uma saída para esse problema é a representação por lista de arestas. No caso do grafo G , a lista de arestas que representa o grafo é:

$$((v_1, v_1), (v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_3), (v_3, v_4), (v_4, v_4))$$

Para um grafo com m arestas, são necessários apenas $2 \cdot m$ dados de armazenamento.

3.3. Conceitos relevantes

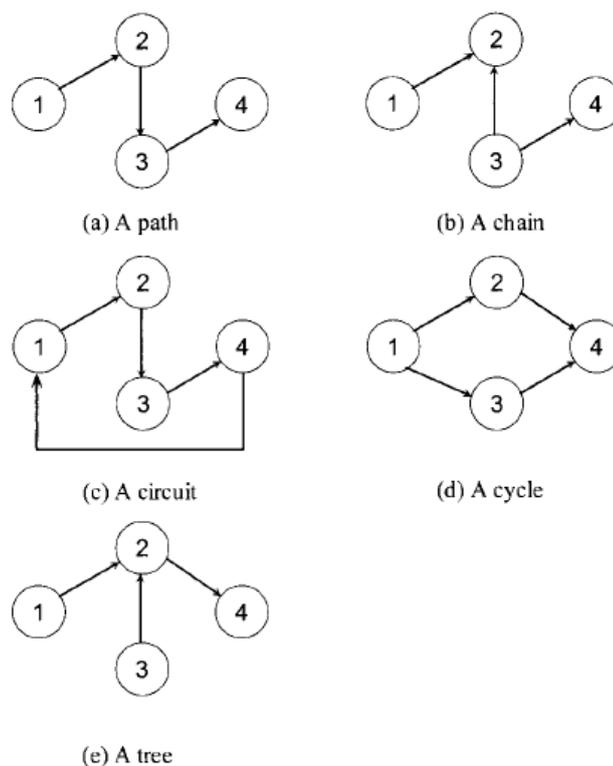
Feofiloff et al. (2011) apresentam conceitos que também são importantes em relação a teoria de grafos são o **isomorfismo** e a **conectividade**. As representações citadas anteriormente independem da maneira como é desenhado um grafo. A partir de uma mesma representação é possível desenhar diferentes grafos, que possuem o mesmo número de arestas e vértices, com mesma matriz adjacente, mesma matriz incidente e mesma lista de arestas. Esses diferentes grafos são ditos isomorfos entre si.

Segundo Feofiloff et al. (2011), de maneira simples, dois grafos são isomorfos se é possível alterar os nomes dos vértices de um deles de tal modo que os dois grafos fiquem iguais.

Nos grafos utilizados como exemplo anteriormente, cada vértice v pode ser alcançado a partir de outro vértice, sendo de forma direta ou passando por uma sequência de vértices e arestas. Essa noção de caminho entre vértices leva ao conceito de conectividade.

A **trilha** é um trecho no qual todas as arestas são distintas; um **caminho** é uma trilha em que todos os vértices também são distintos. Um **ciclo** é uma trilha fechada, na qual todos os vértices, exceto o primeiro e o último são distintos. Um grafo G é dito **conectado** quando todos os pares de vértices distintos são conectados (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011). A Figura 15 mostra esses conceitos de forma gráfica.

Figura 15 - Caminho, cadeia, circuito, ciclo e árvore



Fonte: Bazaraa et al. (2010)

Como é possível observar na Figura 15, um caminho (*path*) é um trecho ou trilha que liga os vértices em uma única direção ou sentido. Uma **cadeia** (*chain*), Figura 15 – b, tem estrutura semelhante a de um caminho, com a exceção de que os arcos ou arestas não tem a mesma direção de fluxo, ou seja, nem todos os arcos vão no sentido do último nó. Considerando o nó inicial como v_0 e o final como v_k , um **circuito** é um caminho que liga do nó v_0 ao v_k , mais a aresta (v_k, v_0) , formando um caminho fechado. Um **ciclo** é uma cadeia fechada (BAZARAA; JARVIS; SHERALI, 2010).

Bazaraa et al. (2010) trazem ainda a definição de árvores no que diz respeito aos grafos. Árvore é um grafo conectado que não possui nenhum ciclo ou caminho fechado. Um exemplo clássico de árvore é mostrado Figura 15 – e, porém, os grafos da Figura 15 – a e b também são árvores, já que não possuem caminhos fechados ou ciclos. Uma árvore que possui n vértices possui sempre $n - 1$ arestas.

Um tipo muito importante de grafo, com muitas aplicações práticas, são os **grafos ponderados**. Nesses tipos de grafos, as arestas possuem pesos, que são números reais associados a cada uma delas. Essa extensão é muito comum na

modelagem de redes reais. Como exemplo, pode-se citar o grafo que representa o tráfego aéreo numa região. Nesse caso, os pesos das arestas podem ser definidos como as distâncias entre cada voo que conecta os aeroportos.

Estes são conceitos básicos e iniciais para a compreensão da Teoria dos Grafos, para posterior aplicação no objeto de estudo desse trabalho de dissertação.

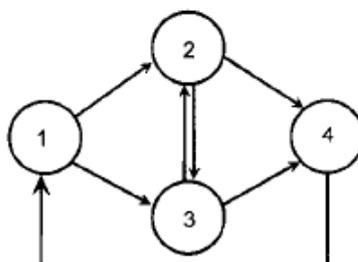
3.4. Princípios da Otimização em Grafos

Existem diversos métodos de otimização voltados para os fluxos em redes ou grafos. Alguns desses métodos tem origem na otimização linear, a partir da modelagem e posterior aplicação dos algoritmos. Como dito anteriormente, a proposta nesse trabalho é aplicação de algoritmos de enxames para a solução desses problemas de fluxo modelados a partir dos grafos. Nessa seção, são apresentados os fundamentos iniciais da otimização dos problemas de fluxo em redes.

Bazaraa et al. (2010) apresentam uma especialização do algoritmo simples, conhecida como algoritmo simplex de rede. O autor trata de problemas clássicos, como o do custo mínimo de fluxo em rede, fluxo máximo e caminho mais curto que são apresentados a seguir.

Considerando um grafo G , formado por um número finito m de vértices e arestas n , considerando que estas sejam arcos diretos. A Figura 16 apresenta um exemplo com quatro nós e sete arcos.

Figura 16 - Exemplo de grafo direto



Fonte: Bazaraa et al. (2010)

Para cada nó i desse grafo G , é associado um número b_i , o qual representa a oferta disponível de um item se $b_i > 0$. Nesse caso o nó é chamado de fonte. Se $b_i < 0$, o nó é chamado de dreno e representa a demanda por determinado item. Se

$b_i = 0$, nenhum item está disponível ou é demandado pelo nó i , que nessa situação é chamado nó intermediário. Associado com cada aresta (i, j) pode-se associar uma variável x_{ij} que representa a quantidade de fluxo no arco, considerando que o fluxo não pode ser negativo. c_{ij} representa o custo unitário de transporte ao longo desse arco (BAZARAA; JARVIS; SHERALI, 2010).

O problema de custo mínimo de fluxo em rede é enviar a oferta disponível através da rede para satisfazer a demanda a um custo mínimo, sendo definido matematicamente como:

Equação 1 - problema de custo mínimo de fluxo em rede

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \\ \text{Sujeito a} \quad & \sum_{i=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = b_i, \quad i = 1, \dots, m \\ & x_{ij} > 0, \quad i, j = 1, \dots, m \end{aligned}$$

As restrições são chamadas de balanço nodal ou conservação de fluxo e indica que o fluxo não pode ser criado e nem destruído na rede. A equação $\sum_{i=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = b_i$ descreve que o fluxo que saí ou entra no nó é igual a diferença do somatório dos fluxos que saem do nó e do que entra.

O problema do custo mínimo pode ser aplicado a vários problemas, como, logística, transporte ferroviário, sistemas de comunicações.

Para os grafos direcionados, onde os arcos possuem orientação, a matriz incidente pode assumir valores negativos, caso em que o fluxo “entra” no nó. Como exemplo, a Figura 17 abaixo mostra a matriz adjacente relativo ao grafo da Figura 16.

Figura 17 - Matriz incidente com valores negativos

$$\mathbf{A} = \begin{matrix} & (1,2) & (1,3) & (2,3) & (2,4) & (3,2) & (3,4) & (4,1) \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 1 \end{bmatrix} \end{matrix}$$

Fonte: Bazaraa et al. (2010)

Para a solução desse tipo de problema, é possível a utilização do método simplex e outros métodos citados na literatura científica, como o algoritmo *out-of-kilter* (BAZARAA; JARVIS; SHERALI, 2010).

Outros dois problemas definidos pelos autores como clássicos em redes são: o fluxo máximo e o caminho mínimo. Para a descrição formal do problema de fluxo máximo, considera-se uma rede com m vértices e n arestas. Para cada aresta (i, j) é associado um limite inferior de fluxo $l_{ij} = 0$ e um limite superior u_{ij} , considerando que esse valor é um número inteiro e finito. No problema de fluxo máximo não há custos envolvidos, onde o objetivo é encontrar o fluxo máximo entre o primeiro e o último nó.

Bazaraa et al. (2010) definem o problema de fluxo máximo como:

Equação 2 – problema do fluxo máximo

$$\begin{aligned}
 & \text{Maximize} && f \\
 & \text{Sujeito a} && \sum_{i=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = \begin{cases} f, & \text{if } i = 1 \\ 0, & \text{if } 1 < i < m \\ -f, & \text{if } i = m \end{cases} \\
 & && x_{ij} \leq u_{ij}, \quad i, j = 1, \dots, m \\
 & && x_{ij} \geq 0, \quad i, j = 1, \dots, m
 \end{aligned}$$

Onde f representa o fluxo total.

Já o problema do caminho mínimo pode ser definido da seguinte maneira: um grafo G com m vértices e n arestas, cada uma com seu custo associado c_{ij} , onde deseja-se encontrar qual o caminho com menor custo para transportar o fluxo do nó 1 até o m . Nesse caso, deseja-se enviar uma unidade de fluxo de um nó inicial até

um nó final, com o menor custo, então a modelagem matemática é definida na Equação 3.

Equação 3 – problema do caminho mínimo

$$\begin{aligned}
 & \text{Minimize} && \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \\
 & \text{Sujeito a} && \sum_{i=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = \begin{cases} 1, & \text{if } i = 1 \\ 0, & \text{if } 1 < i < m \\ -1, & \text{if } i = m \end{cases} \\
 & && x_{ij} = 0 \text{ ou } 1, \quad i, j = 1, \dots, m
 \end{aligned}$$

Alguns trabalhos recentes mostram soluções de problemas de otimização através de outros algoritmos. Corrao et al. (2012) resolveram o problema de otimização em grafos via algoritmos genéticos, com o objetivo de solucionar o problema de conflito de comunicação entre satélites LEO e estações terrestres. Já Jovanovic et al. (2016), utilizaram um algoritmo de otimização por colônia de formigas e Gong et al. (2016) usaram a otimização por enxame de partículas.

O próximo capítulo traz uma revisão sobre a otimização por inteligência de enxames, método que será utilizado para solução do problema de alocação discutido nessa dissertação.

4. OTIMIZAÇÃO POR INTELIGÊNCIA DE ENXAMES

O homem busca inspiração na natureza para resolver diversos problemas práticos. Um exemplo é o radar ultrassônico utilizado em submarinos, que teve seu princípio baseado no sonar dos morcegos.

A partir dos anos 90, com o desenvolvimento da computação, começaram a surgir soluções para problemas de otimização na forma de algoritmos baseados em comportamentos de insetos sociais. Essa área de estudo passou a ser conhecida como Inteligência de enxames. Em muitos problemas de engenharia e problemas operacionais ocorrem não linearidades e as interações entre as variáveis podem ser descritas por funções complexas. Nesse contexto, alguns problemas de otimização podem ser difíceis de resolver por métodos analíticos, baseados em modelos matemáticos conhecidos ou em métodos que utilizem gradiente (SERAPIÃO, 2009).

Segundo Serapião, (2009) métodos analíticos são inviáveis quando existe a dificuldade em formular a modelagem matemática ou o esforço matemático exigido na solução é relativamente alto, principalmente quando estão envolvidas funções não diferenciáveis e descontínuas.

Inteligência de Enxames é o estudo de sistemas computacionais inspirados em inteligência coletiva. Essa inteligência coletiva vem da cooperação de um grande número de agentes trabalhando uniformemente no ambiente. Exemplos da natureza são cardumes de peixes, bando de aves e colônias de formigas (BROWNLEE, 2011).

Esses algoritmos partem do princípio da geração de uma população de possíveis soluções para o problema. Em seguida são utilizadas regras de busca locais para a evolução dessas soluções em busca da solução ótima para o problema. Esses algoritmos representam um ramo de estudo conhecido como computação evolutiva. Os algoritmos evolutivos incluem tradicionalmente os algoritmos genéticos (AG). Esse trabalho foca no outro ramo da computação evolutiva conhecida como inteligência de enxames, também conhecida como inteligência coletiva, ou inteligência de colônias. Estes sistemas são formados por uma população de agentes computacionais simples que possuem a capacidade de perceber e modificar o seu ambiente de maneira local (SERAPIÃO, 2009).

Segundo Denby e Mascle (2003), técnicas de computação baseadas em agentes se tornaram populares nos últimos anos. Um agente autônomo pode ser definido como um indivíduo que detecta o seu ambiente e age sobre ele ao longo do tempo, de forma a atingir algum objetivo. Em sistemas multiagente, eles podem interagir com outros agentes do sistema, bem como os usuários finais do sistema, e podem cooperar com outros agentes para realizar tarefas mais complexas do que eles próprios podem lidar. É possível concluir, por exemplo, que formigas são agentes, de um sistema multiagente.

Uma característica muito importante desses sistemas é que não existe uma estrutura de controle centralizado. No entanto a interação entre os agentes leva a uma solução global que se aproxima da solução ótima, ou seja, um objetivo coletivo é atingido.

De acordo com Denby e Mascle (2003), Inteligência de Enxames é um conjunto de paradigmas computacionais inspirados na natureza, essencialmente no comportamento coletivo de determinados seres vivos, que são aplicados com sucesso em problemas de otimização em uma variedade de áreas.

Para Liu, Abraham e Clerc, (2007), inteligência de Enxames é um paradigma inovador de inteligência distribuída no qual o comportamento de indivíduos pouco sofisticados interagindo com o seu ambiente geram padrões funcionais coerentes.

Swarm Intelligence é o nome dado a um campo relativamente novo de pesquisa interdisciplinar, que ganhou uma grande popularidade nos últimos tempos. Algoritmos que pertencem a este campo inspiram-se na inteligência coletiva que emerge do comportamento de um grupo de insetos sociais (como abelhas, cupins e vespas). Estes insetos, mesmo com capacidade individual muito limitada, em conjunto podem (cooperativamente) executar muitas tarefas complexas, necessárias para a sua sobrevivência (DAS; KONAR, 2007).

Segundo Brownlee (2011), o paradigma da inteligência de enxames compreende dois subcampos dominantes: 1) Otimização por colônia de formigas, com mecanismos de busca baseados em estigmergia e busca por alimentos pelas formigas; 2) Otimização por enxame de partículas, baseado no mecanismo de voo de pássaros ao buscar alimentos.

É possível verificar que diversos autores nos últimos anos vêm pesquisando e desenvolvendo algoritmos nessa área.

Um sistema baseado em inteligência de enxames devem possuir algumas propriedades definidas por Millonas (1994),

- **Proximidade:** Os agentes devem ser capazes de interagir.
- **Qualidade:** Os agentes devem ser capazes de avaliar seus comportamentos.
- **Diversidade:** Permite ao sistema reagir a situações inesperadas.
- **Estabilidade:** Nem todas as variações ambientais devem afetar o comportamento de um agente.
- **Adaptabilidade:** Capacidade de se adequar a variações ambientais.

Segundo Denby e Mascle (2003), as aplicações da inteligência de enxames incluem o problema do caixeiro viajante, atribuição quadrática, coloração de grafos, otimização de roteamento da rede, descoberta cluster, agendamento de tarefas, motores de busca, e balanceamento de carga. SI parece, assim, adaptar-se bem para os problemas que envolvem complexidade combinatória.

São diversas as aplicações dos algoritmos inspirados em *swarm intelligence* nas diversas áreas da engenharia e da automação de processo. Uma área que vem sendo objeto de pesquisa de diversos pesquisadores é a união da robótica com a inteligência de enxames. Essa área é conhecida como *Swarm Robotics*. Para Valdastrì et al. (2006), a microrrobótica tem uma importância muito significativa no contexto da “*Swarm Robotics*” (Robótica de Enxame) – Pode ser definido como o estudo de como um número relativamente grande de agentes simples podem ser construídos e programados para realizar tarefas coletivas que vão além da capacidade de um único agente. Consiste na utilização de pequenos robôs como agentes autônomos que interagem entre si, com o objetivo de realizar alguma tarefa que não pode ser realizada por um único robô, mas sim por um conjunto deles. Alguns trabalhos nessa área são demonstrados em alguns artigos. Um exemplo é descrito por Ishii e Miki, (2007). Segundo Takemura et al. (2006), inteligência coletiva é um dos temas-chave de pesquisa em robótica como o problema

multiagente, que são ativamente investigados através de simulações numéricas e experimentos usando muitos robôs móveis.

Existem diversos algoritmos, como, otimização por colônia de formigas, otimização por enxame de partículas, algoritmo vaga-lume, *Cuckoo Search Algorithm*, otimização por colônia de bactérias, colônia de abelhas, algoritmo *shuffled frog-leaping* e algoritmo de enxame de andorinhas. No entanto, nessa dissertação são descritos três deles, que são: otimização por colônia de formigas, otimização por enxame de partículas e otimização por colônia de bactérias.

4.1. Otimização por colônia de formigas (ACO)

Quando as formigas saem do formigueiro em busca de alimentos, vão depositando no solo uma substância química chamada de feromônios. No caminho mais curto as formigas depositam mais feromônio, pois vão e retornam num intervalo de tempo mais curto. Essa substância é volátil, evaporando rapidamente no ambiente. As trilhas mais curtas ficam com mais feromônios, atraindo mais formigas, que por sua vez depositam mais feromônios reforçando ainda mais aquela trilha química. Esse tipo de comunicação baseado em marcas é chamado de estigmergia (SERAPIÃO, 2009).

Esse comportamento foi observado por Deneubourg et al. (1990) através de um experimento com formigas reais, onde formigas argentinas do tipo *Iridomyrmex humilis* foram colocadas numa área com dois caminhos até o alimento. Depois de certo tempo foi verificado que a trilha mais curta estava repleta de formigas e a mais longa possuía poucas.

Dorigo et al. (1991) foram os pioneiros no desenvolvimento de um algoritmo baseado no comportamento das formigas, e o chamaram de *Ant System (AS)*. A aplicação desse algoritmo objetivou resolver o problema do caixeiro viajante. Um grupo de “formigas artificiais” foram implementadas com o objetivo de encontrar o caminho mais curto entre as cidades, através da utilização de “feromônios artificiais”. As formigas se movem inicialmente de forma aleatória entre as cidades e depositando os feromônios. A quantidade desses é inversamente proporcional ao tamanho do caminho percorrido, e como a substância evapora com o tempo, após

alguns movimentos as trilhas mais curtas terão quantidades maiores, atraindo cada vez mais formigas.

4.1.1. Algoritmo ACO e suas aplicações

Dorigo et al (1991), descreveram pela primeira as características gerais dos algoritmos de colônia de formiga, desenvolvendo o Ant System (AS) com três variantes: *AS-density*, *AS-quantity* e *AS-cycle*. Essas variantes modificavam apenas a maneira como as trilhas de feromônios eram atualizadas, sendo a última mais utilizada.

Os autores aplicaram o algoritmo de formigas para resolver o problema do caixeiro viajante, devido à possibilidade de comparar o desempenho com outros métodos. Dado um conjunto de n cidades, o problema é encontrar o caminho mais curto entre elas, passando por cada cidade apenas uma vez. O problema pode ser representado por um grafo ponderado (N,E) , onde N é o conjunto de cidades e E é o conjunto de arestas entre as cidades, ponderada pelas distâncias. Em cada um dos vértices é colocada uma formiga artificial, que percorre um caminho seguindo uma regra probabilística em função do feromônio depositado em cada trilha.

Para evitar que uma formiga visite um mesmo vértice, existe uma lista tabu, que é uma estrutura de dados que guarda os vértices já visitados. Para construção da solução, cada formiga (k) utiliza iterativamente uma regra de transição de estado ($p_{i,j}$), que é a função probabilística para decidir se determinada aresta entrará na solução:

$$p_{i,j}^k(t) = \begin{cases} \frac{\tau_{i,j}^\alpha(t) * \eta_{i,j}^\beta}{\sum_{j \in J^k} \tau_{i,j}^\alpha(t) * \eta_{i,j}^\beta}, & \text{se } j \in J^k \\ 0, & \text{caso contrário} \end{cases}$$

Onde τ_{ij} é a trilha de feromônio da combinação (i,j) , η_{ij} é a heurística local da combinação (i,j) , α é a importância relativa da trilha e β é a importância da heurística local. Não existe uma regra para a escolha desses valores de α e β , os quais devem ser determinados experimentalmente. A atualização das trilhas de feromônio nas arestas são calculadas em função da qualidade das soluções da qual elas pertencem.

$$\tau_{i,j}(t+1) = (1 - \rho) * \tau_{i,j}(t) + \rho * \Delta\tau_{i,j}(t),$$

$$\Delta\tau_{i,j}(t) = \begin{cases} \frac{1}{f(S)}, & \text{se } (i,j) \in S \\ 0, & \text{caso contrário} \end{cases}$$

Onde ρ é a taxa de evaporação, usada para evitar a rápida convergência das formigas. $\Delta\tau_{ij}$ é a quantidade de feromônio depositada na aresta (i,j) , que depende função de custo f da solução S . O valor de τ é atualizado a cada iteração do algoritmo de busca para aumentar os valores dos movimentos que resultaram numa boa solução.

Segundo Serapião (2009), um algoritmo ACO pode ser resumido no seguinte procedimento:

Algoritmo 1 – Procedimento geral de um algoritmo baseado em colônia de formigas

1. Inicializar os parâmetros k, α, ρ e β e as trilhas de feromônio τ_{ij} com um mesmo valor inicial;
2. Colocar cada formiga k em uma aresta selecionada de forma aleatória;
3. Para cada formiga k , construir soluções baseadas na regra de transição de estado $p_{i,j}^k$;
4. Calcular $f(S)$;
5. Guardar a melhor solução até o momento;
6. Para cada aresta do grafo, aplicar a regra de atualização da trilha de feromônio (calcular $\Delta\tau_{ij}$);
7. Retornar ao passo 3 até a condição de término ser atendida.

Diversas aplicações desse algoritmo foram utilizadas. Denby e Mascle (2003) trazem uma aplicação do ACO no roteamento de satélites de telecomunicações. A simulação foi feita utilizando 72 satélites em LEO e 121 estações terrestres, com os resultados comparados com os obtidos com a técnica padrão de roteamento. Zhang

et al. (2007) utilizaram um algoritmo Ant-system para fazer o controle descentralizado. O método proposto permite que os membros da equipe robô se adaptem aos ambientes dinâmicos desconhecidos, respondendo de forma flexível e robusta às perturbações do ambiente às modificações na equipe decorrentes de falha mecânica.

Existem diversas variações do ACO original desenvolvido por diversos pesquisadores nos últimos anos.

4.2. Otimização por enxame de partículas (PSO)

A otimização por enxame de partículas é uma técnica de inteligência de enxames desenvolvida por Kennedy e Eberhart em 1995.

O PSO é um modelo que surgiu através da inspiração do comportamento social baseado em algumas espécies de pássaros e cardume de peixes. A busca por alimentos e a interação entre as aves ao sobrevoar determinada área são modeladas como um mecanismo de otimização. A área sobrevoada na busca por alimento é o espaço das soluções e encontrar a comida significa encontrar a solução ótima. No algoritmo, os pássaros, que são as partículas fazem uso da sua própria experiência e da experiência do bando para encontrar a solução ótima (ESPOSITO, 2012).

A probabilidade de uma partícula encontrar a melhor solução é uma função da sua experiência (desempenho passado) e da experiência coletiva do bando. O algoritmo PSO é baseado em uma população de indivíduos capazes de interagir entre si e com o ambiente, onde as ações dos indivíduos são baseadas em comportamentos com propriedades de auto avaliação, e onde os indivíduos sentem o ambiente de forma a estimar seu próprio comportamento, fazendo comparação com os demais indivíduos e imitação das habilidades individuais (SERAPIÃO, 2009).

4.2.1. Algoritmo de otimização por enxame de partículas

Para que o bando de pássaros sempre se aproxime do objetivo, ao invés de se perder ou nunca alcançar o alvo focado, utiliza-se o indicador denominado **fitness**, função que irá avaliar o desempenho das partículas. Comparando com os pássaros reais, estes para alcançar o alvo focado, sejam os alimentos ou os ninhos, os pássaros fazem uso de suas experiências e da experiência do próprio bando. O

termo que indica a melhor experiência de uma determinada partícula é o **pbest**, já a experiência do bando é o **gbest** (ESPOSITO, 2012).

A Tabela 2 sintetiza as nomenclaturas utilizadas no algoritmo.

Tabela 2 – Nomenclaturas utilizadas no PSO

Termo no algoritmo	Significado
Partícula	Pássaro
Enxame	Bando de pássaros
Espaço de Busca	Área sobrevoada pelos pássaros
Posição	Localização de cada pássaro durante o vôo.
Solução ótima	Localização do pássaro onde este encontrou o alimento
Fitness	Função de avaliação
pbest	Melhor posição conhecida pelo pássaro (experiência individual)
gbest	Melhor posição conhecida pelo bando (experiência coletiva)

Fonte: Próprio autor

O conceito desse algoritmo de otimização de otimização é a cada intervalo de tempo, mudar a velocidade de cada partícula em função do *pbest* e *gbest* (EBERHART; KENNEDY, 1995).

Serapião (2009) descreve um modelo de otimização por enxame de partículas sendo composto por diversas partículas que se movem num espaço de busca n -dimensional, onde existe uma aptidão $f(p)$ que pode ser calculada como uma medida de qualidade. Cada partícula p_i tem sua posição determinado por um vetor de posição $x_i(t)$ e com velocidade representada respectivamente por um vetor $v_i(t+1)$. Cada partícula guarda sua melhor posição até o instante num vetor p_B (*pbest*) A melhor posição para o enxame também é armazenada, essa num vetor p_g (*gbest*). Durante o tempo de interação t a velocidade é atualizada segundo a equação 1:

$$v_i(t+1) = v_i(t) + \varphi_1 * (p_B - x_i(t)) + \varphi_2 * (g_B - x_i(t)) \quad (1)$$

Onde φ_1 e φ_2 são constantes limitadas dentro de um intervalo finito, em que Kennedy denomina respectivamente de componentes “cognitiva” e “social”.

Na próxima iteração, a nova posição da partícula i é calculada como a soma da posição anterior e a componente de velocidade, como mostra a equação 2.

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2)$$

Para que a posição da partícula não extrapole o espaço de busca, a velocidade em cada dimensão d é limitada a uma velocidade máxima. O algoritmo é repetido até que um critério de parada seja atingindo, como mudanças muito pequenas na velocidade.

Um algoritmo em pseudocódigo para o modelo de enxame de partículas é dado por Serapião (2009):

Algoritmo 2 – Pseudocódigo de um algoritmo PSO

1. Determinar o número de partículas P da população.
2. Inicializar a posição de todas as partículas de forma aleatória.
3. Inicializar as velocidades das partículas de forma aleatória.
4. Para cada partícula p_i faça:
 - a. Calcular a aptidão (*fitness*) $f(p)$;
 - b. Calcular a melhor posição de p_i até o momento (pb);
5. Verifique qual a partícula possui a melhor aptidão de todo o bando (gb):
6. Para cada partícula p_i faça:
 - a. Atualize a velocidade de acordo com a equação 1;
 - b. Atualize a posição de acordo com a equação 2;
7. Se a condição de término não for alcançada , retorne ao passo 4:

Mediante as iterações do sistema, agentes individuais percorrem o espaço em busca do ótimo global, baseados nas iterações com os demais (experiência global) e em sua busca individual (experiência individual) (EBERHART; KENNEDY, 1995).

O algoritmo pode ser demonstrado na forma de um fluxograma, como é possível observar na Figura 18.

Figura 18 - Fluxograma algoritmo PSO



Fonte: Próprio autor (2016)

O algoritmo PSO é largamente utilizado em diversas aplicações. Existem aplicações para melhoria de outros métodos ou algoritmos de otimização, como redes neurais, algoritmos genéticos ou lógica fuzzy. Zahiri e Seyedin (2007) criaram um controlador chamado de classificador Fuzzy, controlado por enxame de partículas (FCPS-classificador). As regras do controlador fuzzy foram extraídas de uma descrição linguística a partir de pesquisas anteriores sobre o estudo dos efeitos dos parâmetros PSO no seu processo de pesquisa.

O PSO também é largamente utilizado em áreas de engenharia, como otimização de controladores PID, controle robusto e identificação de sistemas. Das e Konar (2007) utilizaram o algoritmo de otimização por enxame de partículas para projeto de filtros digitais. Uma aplicação de otimização de controlador PID é demonstrada por Sabir e Ali (2016).

Bhattacharyya e Raj (2016) aplicaram algoritmos baseados inteligência de enxame para a coordenação eficaz do sistema de transmissão AC flexível (FACTS). Os autores utilizaram SPSO (Simple Particle Swarm Optimization), APSO (Adaptive Particle Swarm Optimization) e EPSO (Evolutionary Particle Swarm Optimization).

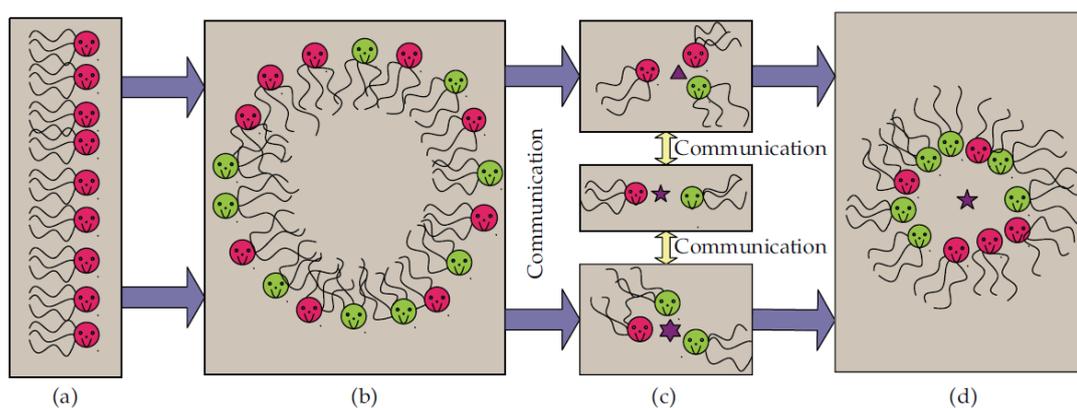
4.3. Otimização por colônia de bactérias (BCO)

Seguindo a teoria da seleção natural, espécies que possuem estratégias eficientes de coleta de alimentos sobrevivem, enquanto as que não têm morrem ou evoluem para espécies mais adaptadas. Com o objetivo de melhorar a busca por alimentos uma espécie age de forma a otimizar a energia gasta nessa tarefa, com todas as restrições existentes no meio. Essa atividade que motivou pesquisadores a se basearem nesse processo para criar determinado algoritmo de otimização.

O primeiro algoritmo baseado no comportamento de bactérias foi o de coleta de alimento por bactérias (BFO), proposto por Passino (2000). Esse algoritmo é inspirado no ciclo de vida da bactéria *Escherichia coli*. Segundo o autor, o ciclo de vida dessa bactéria é dividido em quatro estágios: quimiotaxia e comunicação, eliminação, reprodução e migração.

No entanto, algoritmos baseados no comportamento tradicional da bactéria BFO consideram apenas os comportamentos individuais em vez de comportamentos sociais com inteligência de enxame. Cada indivíduo na colônia procura independentemente por comida por sua própria experiência, sem qualquer troca de informações com os outros. Com o objetivo de melhorar o algoritmo, Niu e Wang (2012) propõem um novo algoritmo de otimização por colônia de bactérias (BCO). A Figura 19 ilustra o comportamento individual e social das bactérias.

Figura 19 – Comportamento individual e social das bactérias



Fonte: Niu e Wang (2012)

As bactérias podem nadar impulsionadas pelo giro dos flagelos, guiadas por um propulsor reversível encaixado na parede da célula, como pode ser observado na figura 18.a. Devido ao consumo dos nutrientes, os indivíduos mais fortes sobrevivem e geram novos indivíduos, como na figura 18.b. Esses indivíduos migram para diferentes áreas na busca de novos nutrientes como mostra a figura 18.c, compartilhando suas experiências. As bactérias se reproduzem novamente na região com nutrientes, como se pode observar na figura 18.d. O comportamento básico de bactérias no ciclo de vida pode ser simplesmente dividido em cinco partes: a quimiotaxia, eliminação, reprodução, migração, e comunicação.

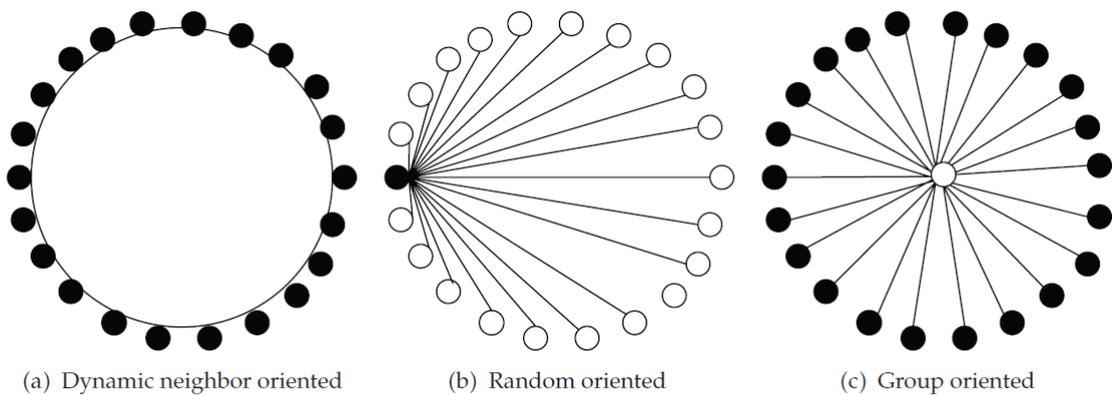
Niu e Wang (2012), descrevem ainda os quatro passos reproduzidos no algoritmo: na **quimiotaxia** as bactérias se movem em busca da maior concentração de nutrientes. Esse movimento depende do sentido de giro dos flagelos: se o giro é no sentido anti-horário, as bactérias se movem numa trajetória helicoidal com uma única direção. Se o giro é horário, a bactéria se move de forma aleatória. Uma combinação dos movimentos permite a bactéria buscar por alimentos em diferentes direções.

Bactérias com menor capacidade de busca teriam maior chance de ser eliminadas. Em contraste, aqueles que um bom desempenho no processo de quimiotaxia obteriam mais energia para a sobrevivência e, portanto, tem uma alta probabilidade de reprodução. Depois de um longo tempo de quimiotaxia, **eliminação** e **reprodução** na mesma área ou ambiente, a nutrição não pode satisfazer todas as

bactérias. Neste momento, algumas bactérias devem migrar para um novo local com nutrientes, e este processo é chamado de "**migração**".

A **comunicação** pode ser feita de três formas, que representam três topologias diferentes: comunicação dinâmica entre os vizinhos, comunicação orientada aleatória ou comunicação em grupo. A figura 19 mostra esses 3 tipos de comunicação implementados no BCO.

Figura 20 – Tipos de comunicação no BCO



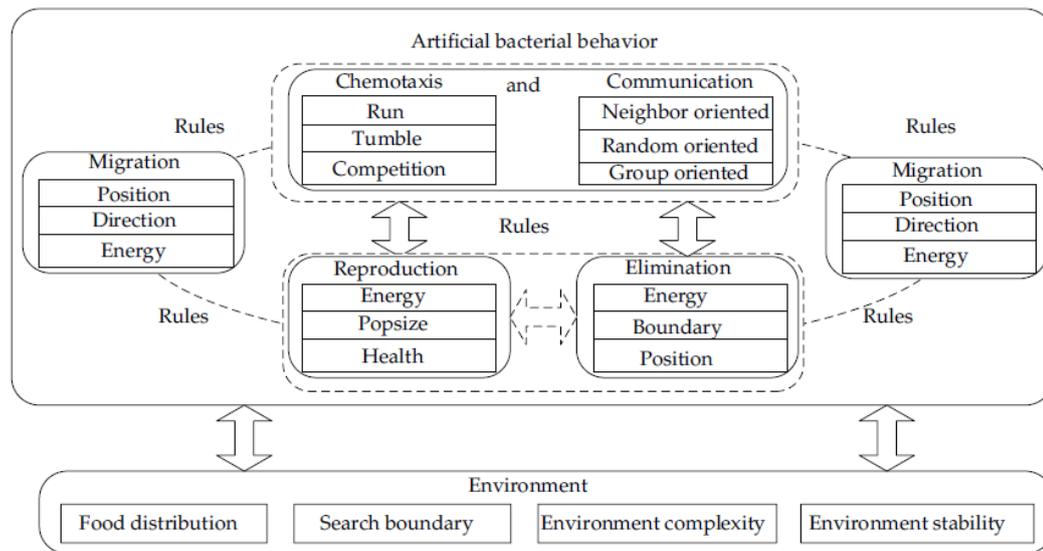
Fonte: Niu e Wang (2012)

A quimiotaxia é sempre acompanhada da comunicação por todo o ciclo de vida da bactéria. No BCO, quimiotaxia e comunicação são tratados como um modelo. Após longos períodos desse processo, as bactérias podem morrer por falta de alimento ou se reproduzir se suas capacidades de busca de alimento são eficientes. A migração é conduzida como um modelo independente, que envolve diminuição de energia, diversidade de grupo e quimiotaxia eficiente. O modelo foi baseado num esquema com três elementos fundamentais:

- A: bactérias artificiais;
- E: ambiente artificial;
- R: mecanismo de interações entre bactérias e entre o ambiente.

Em geral, o modelo de ciclo de vida das bactérias artificiais em BCO pode ser dividido em quatro submodelos, como ilustra a *Figura 20*: quimiotaxia e modelo de comunicação, o modelo de reprodução, modelo eliminação, e modelo de migração.

Figura 21 – Modelo do ciclo de vida da bactéria



Fonte: Niu e Wang (2012)

Niu et al. (2015) trazem de forma simplificada os procedimentos do algoritmo BCO, descritos nos próximos parágrafos. Na primeira etapa, as bactérias caem na direção ideal seguindo as informações: informação prévia pessoal, informações de grupo e direção aleatória.

$$\begin{aligned}
 & Posição_i(t) = Posição_i(t-1) + C(i) \\
 & * [f_i * (G_{best} - Posição_i(t-1)) + (1 - f_i) \\
 & * (P_{besti} - Posição_i(t-1)) + turbulent_i]
 \end{aligned}$$

$$\begin{aligned}
 & Posição_i(t) = Posição_i(t-1) + C(i) \\
 & * [f_i * (G_{best} - Posição_i(t-1)) + (1 - f_i) \\
 & * (P_{besti} - Posição_i(t-1))]
 \end{aligned}$$

Onde i representa a i -ésima bactéria. P_{best} e G_{best} representam respectivamente a melhor posição anterior da i -ésima bactéria e a melhor posição de todo o enxame. f_i é um escalar entre 0 e 1. E $turbulent_i$ é a direção estocástica atribuída a i -ésima bactéria. $C(i)$ representa o tamanho do passo da quimiotaxia.

Tal como para a eliminação e a reprodução do modelo, a bactéria reproduzir ou ser eliminada, depende da qualidade solução de bactéria que é indicado pelo nível de energia. Após obtenção dos graus distribuição de energia bacteriana,

podemos defini-lo como um juízo sobre a qualificação de bactérias conforme apresentado a seguir:

$$\begin{aligned} & \text{se } L_i > L_{\text{dado}}, \text{ e } i \in \text{saudável}, \text{ então } i \in \text{Candidatos}_{\text{repr}}, \\ & \text{se } L_i < L_{\text{dado}}, \text{ e } i \in \text{saudável}, \text{ então } i \in \text{Candidatos}_{\text{eli}}, \\ & \text{se } i \in \text{não - saudável}, \text{ então } i \in \text{Candidatos}_{\text{eli}}, \end{aligned}$$

No modelo de migração da bactéria, a mesma migra para uma área aleatória quando determinada condição é satisfeita.

$$\text{Posição}_i(t) = \text{rand} * (ub - lb) + lb$$

Onde lb e ub são respectivamente os limites inferior e superior. Rand é um valor aleatório entre 0 e 1.

O algoritmo BCO usa o passo adaptativo da quimiotaxia $C(i)$ definido pela equação abaixo:

$$C(i) = C_{\min} + \left(\frac{\text{iter}_{\max} - \text{iter}_j}{\text{iter}_{\max}} \right)^n * (C_{\max} - C_{\min})$$

Onde iter_{\max} é o número máximo de iterações, e iter_j é o número atual de iterações.

4.4. Aplicações do algoritmo BCO e BFO

Diversos pesquisadores continuam desenvolvendo melhorias para o algoritmo original e diversas aplicações surgiram. Biswas et al. (2007) desenvolveram um algoritmo que acopla as funcionalidades e características dos algoritmos BCO e PSO. Korani, Dorrah e Emara (2009) também desenvolveram um algoritmo híbrido BCO e PSO. O algoritmo foi denominado de *Bacterial Foraging Optimization oriented by Particle Swarm Optimization* (BF-PSO) e foi aplicado ao problema de sintonia de controladores PID. Kim et al. (2007) trouxeram uma abordagem híbrida unindo BCO e algoritmos genéticos para a otimização de funções.

5. METODOLOGIA E DESENVOLVIMENTO

Em relação à abordagem do problema, a pesquisa está caracterizada como pesquisa quantitativa, já que seus resultados podem ser expostos em valores numéricos. Os modelos gerados traduzem o problema de alocação de DDO's livres numa função objetivo para ser otimizada, sujeita as restrições do sistema. As restrições nos modelos são geradas em função de evitar os conflitos de visibilidade, assegurando que cada antena comunique com um satélite por vez e vice-versa.

Após a modelagem, foi realizada a otimização, com objetivo de obter taxas máximas de download, utilizando-se, para o modelo centralizado o algoritmo Simplex, e para o descentralizado o algoritmo de inteligência de enxames conhecido como ACO, otimização por colônia de formigas.

Antes de aplicar o ACO, o problema foi modelado com variáveis contínuas e resolvido utilizando o método simplex, já que o modelo é linear. O objetivo foi obter a solução de forma centralizada para poder comparar posteriormente com o modelo solucionado através do algoritmo de formigas.

Os modelos de programação linear foram aplicados considerando as taxas de download de cada par estação/satélite como constantes e utilizando um cenário de passagem de satélites obtido a partir de simulação computacional, baseado na RIBRAS. Esse cenário é apresentado posteriormente na seção de modelagem, na *Figura 23*, mostrando os intervalos de tempo em que os satélites estão visíveis para determinadas estações. Já que os satélites estão situados em órbita baixa, suas *footprints* abrangem áreas pequenas, fazendo com que apenas algumas antenas visualizem esses satélites durante o voo.

Cada antena tem possibilidade de comunicar com um ou mais satélites em determinados intervalos de tempo, delimitados pelos AOS's e EOS's de cada satélite. Quando ocorrem conflitos de visibilidade (mais de um satélite sobre a mesma antena) ou disputas (mais de uma antena com possibilidade de comunicar com o mesmo satélite) é necessário contornar através da alocação adequada dos slots de tempo de cada estação.

O modelo centralizado é descrito na seção de desenvolvimento, onde as variáveis de decisão são os instantes de tempo de início e de término de

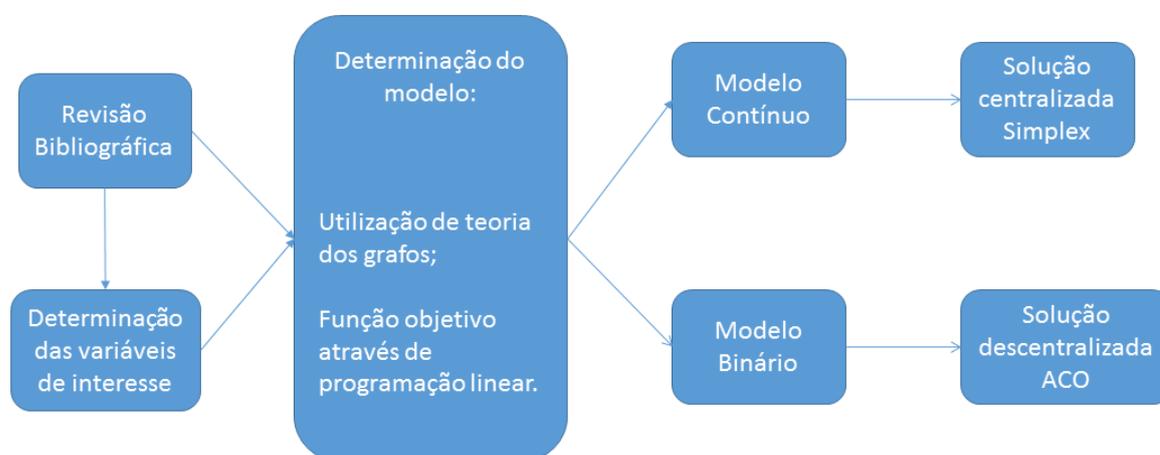
comunicação entre cada par estação/satélite envolvidos no problema. As restrições foram determinadas com o objetivo de evitar os possíveis conflitos.

O ACO é utilizado para obter a solução de forma descentralizada, onde cada região de conflito é resolvida como um grafo diferente. Essa solução descentralizada permitirá a negociação entre as estações envolvidas no conflito. Cada grafo foi chamado de Domínio de conflito. Como base para esse desenvolvimento foram empregados os conceitos de slots de tempo apresentados por Carvalho et al. (2013).

Nesse caso o modelo precisou ser baseado em variáveis binárias, já que o ACO trabalha com problemas de complexidade combinatória. Nesse sentido, as variáveis deixam de ser tempo e passam a decidir se determinada estação deve comunicar com determinado satélite durante aquele intervalo de tempo ($x = 1$) ou não comunicar ($x = 0$).

A Figura 22 mostra através de um diagrama como ficou o arranjo metodológico da dissertação.

Figura 22 - Arranjo metodológico da dissertação



Fonte: Próprio Autor (2016)

O ponto de partida é uma pesquisa bibliográfica conduzida por uma análise bibliométrica feita em bases científicas relevantes, com o objetivo de construir as bases teóricas da dissertação. A partir deste ponto, a pesquisa segue um caráter experimental, onde foi definido o objeto de estudo como a rede de nanosatélites e estações terrestres.

Para a modelagem, foi utilizada a Teoria dos Grafos e Programação Linear, gerando assim uma função objetivo que representa de forma adequada o problema com suas determinadas restrições.

O primeiro modelo é solucionado através do Solver do Excel e o segundo é solucionado através do algoritmo de formigas implementado no Matlab.

5.1. Modelagem

Para realizar a modelagem matemática do problema de forma centralizada utilizou-se otimização linear, já que as variáveis de decisão respeitam as condições de linearidade.

Foi desenvolvido um modelo genérico como segue:

Função objetivo:

$$\max z = \sum_{i=1}^n \sum_{j=1}^m T_{Xij} (x_{Eij} - x_{Aij})$$

Onde:

- x_{Aij} é o instante de tempo, em segundos, que a estação j inicia a comunicação com o satélite i ;
- x_{Eij} é o instante de tempo, em segundos, que a estação j finaliza a comunicação com o satélite i ;
- T_{Xij} é a taxa de download e em bps da estação j com o satélite i ;

Obs: é importante destacar que, para os testes preliminares, foram arbitrados valores constantes de taxas de download para cada estação baseados em taxas de transmissão típicas;

- n é o número de satélites considerado no modelo;
- m é o número de estações considerado no modelo.

Restrições:

$$x_{Eij} - x_{Aij} \geq 0$$

$$x_{Aij} \geq AOS_{ij}$$

$$x_{Eij} \leq EOS_{ij}$$

Essas restrições estabelecem os limites para as variáveis em função do cenário de simulação que deverá ser otimizado, onde:

- AOS_{ij} é o instante de tempo em segundos no qual o satélite i entra na região de visibilidade da estação j ;
- EOS_{ij} é o instante de tempo em segundos no qual o satélite i sai da região de visibilidade da estação j ;

$$x_{Eij} < x_{Ai(j+k)}, k = 1, \dots, m$$

$$x_{Eij} < x_{A(i+l)j}, l = 1, \dots, n$$

As restrições acima tem por objetivo evitar os conflitos de visibilidade entre estações e satélites. A primeira é voltada aos conflitos que podem ocorrer em determinados instantes de tempo, onde o satélite i entra na região de visibilidade da estação j e da estação $(j + k)$ simultaneamente. Já a segunda restrição está relacionada à situação em que os satélites i e $(i + l)$ estão na região de visibilidade da mesma antena j .

Além disso, as variáveis de decisão são inteiras e devem atender as condições de não-negatividade.

Para aplicação do modelo foi utilizado um cenário obtido a partir de simulações computacionais, baseado na Rede Integrada Brasileira de Rastreamento de Satélites (RIBRAS).

Cenário 1: O cenário 1 representa uma simulação com 5 estações e 9 satélites, sendo:

- m_1 : estação de Campos – RJ
- m_2 : estação de Salvador - BA
- m_3 : estação de Natal – RN
- m_4 : estação de Cuiabá – MT
- m_5 : estação de Brasília– DF

- n_1 : IFFSat01
- n_2 : IFFSat04
- n_3 : IFFSat05
- n_4 : IFFSat06
- n_5 : IFFSat08

- n_6 : IFFSat09
- n_7 : IFFSat10
- n_8 : IFFSat11
- n_9 : IFFSat12

A Tabela 3 apresenta os dados utilizados na aplicação do modelo de otimização linear ao Cenário 1.

Tabela 3 - Dados Cenário 1

AOS	EOS	Duração	Satélite	Estação
19:26:38	19:28:38	00:02:00	IFFSat01	Salvador
19:28:01	19:31:02	00:03:01	IFFSat01	Natal
19:37:49	19:40:54	00:03:05	IFFSat04	Campos
19:38:55	19:42:17	00:03:22	IFFSat04	Brasília
19:40:11	19:43:41	00:03:30	IFFSat05	Brasília
19:40:27	19:42:43	00:02:16	IFFSat04	Salvador
19:40:30	19:42:52	00:02:22	IFFSat05	Cuiabá
19:40:53	19:44:12	00:03:19	IFFSat06	Campos
19:43:02	19:46:32	00:03:30	IFFSat06	Salvador
19:44:51	19:48:28	00:03:37	IFFSat06	Natal
19:52:29	19:55:54	00:03:25	IFFSat08	Cuiabá
19:53:05	19:55:44	00:02:39	IFFSat08	Brasília
19:55:35	19:58:05	00:02:30	IFFSat09	Campos
19:57:37	20:00:34	00:02:57	IFFSat09	Salvador
19:59:15	20:02:43	00:03:28	IFFSat09	Natal
20:01:14	20:04:42	00:03:28	IFFSat10	Brasília
20:01:28	20:03:58	00:02:30	IFFSat10	Cuiabá
20:04:00	20:07:34	00:03:34	IFFSat11	Campos
20:05:41	20:08:20	00:02:39	IFFSat11	Brasília
20:06:24	20:09:37	00:03:13	IFFSat11	Salvador
20:09:49	20:13:25	00:03:36	IFFSat12	Cuiabá

Fonte: próprio Autor (2016)

A partir dessa tabela inicial, foi gerada a Tabela 4 com o tempo normalizado em segundos, onde 19:26:38 foi considerado o instante inicial $t_0 = 0$ s.

Tabela 4 - Dados Normalizados Cenário 1

	AOS_{ij}	EOS_{ij}	Duração	T_{xij} (bps)
ΔT_{12}	0	120	120	22000
ΔT_{13}	83	264	181	18000

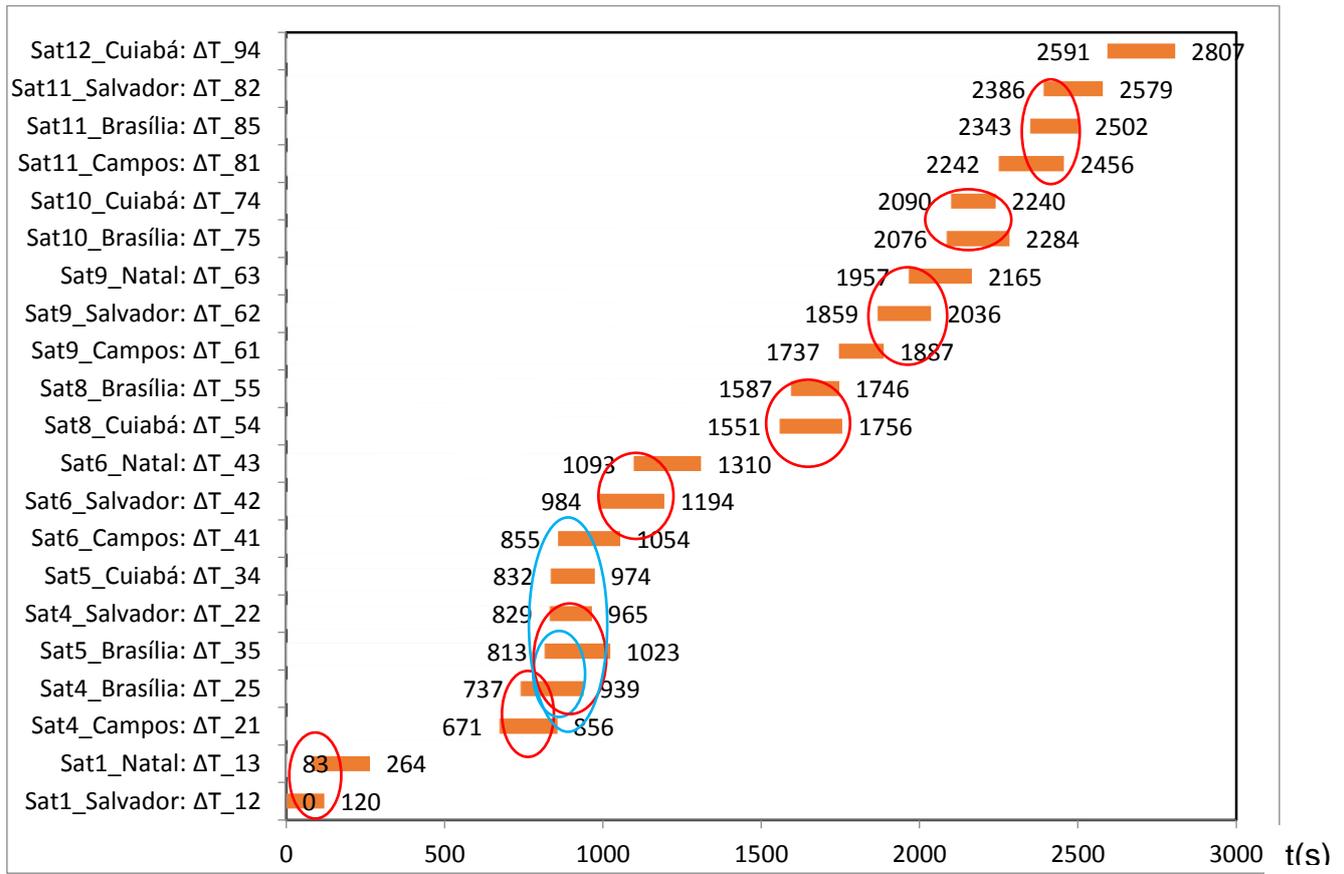
ΔT_{21}	671	856	185	30000
ΔT_{25}	737	939	202	22000
ΔT_{35}	813	974	210	25000
ΔT_{22}	829	965	136	20000
ΔT_{34}	832	974	142	36000
ΔT_{41}	855	1054	199	26000
ΔT_{42}	984	1194	210	22000
ΔT_{43}	1093	1310	217	24000
ΔT_{54}	1551	1746	205	15000
ΔT_{55}	1587	1746	159	18000
ΔT_{61}	1737	1887	150	36000
ΔT_{62}	1859	2036	177	30000
ΔT_{63}	1957	2165	208	28000
ΔT_{75}	2076	2240	208	22000
ΔT_{74}	2090	2240	150	25000
ΔT_{81}	2242	2456	214	18000
ΔT_{85}	2343	2502	159	20000
ΔT_{82}	2386	2579	193	22000
ΔT_{94}	2591	2807	216	30000

Fonte: próprio Autor (2016)

A última coluna da tabela mostra a taxa de transmissão de cada par estação satélite em bits por segundo. Uma mesma estação pode ter taxas diferentes para diferentes satélites, como exemplo temos, $T_{x_{25}}$ e $T_{x_{35}}$ com respectivamente 22000 e 25000 bps. Esses valores foram arbitrados para a resolução do problema em questão e considerados constantes ao longo do tempo. Na prática essa taxa não é constante e depende de diversos fatores, como, posicionamento do satélite, condições climáticas, regiões de sombra formadas por relevos ou construções, entre outras.

Considerando esse cenário, foi gerado um gráfico mostrado na Figura 22, que permite a visualização dos conflitos de visibilidade.

Figura 23 - Gráfico Cenário 1



Fonte: Próprio autor (2016)

As áreas circuladas em vermelho correspondem a disputas entre estações que estão visualizando o mesmo satélite. Nesses casos, os índices i se repetem nas barras que se sobrepõem nos mesmos intervalos de tempo, como é o caso do primeiro conflito $\Delta T_{12} \rightarrow \Delta T_{13}$. Significa que o satélite n_1 está na linha de visão das antenas m_2 e m_3 no período de tempo compreendido entre 83 e 120 s.

Já as circuladas em azul correspondem aos conflitos em que diferentes satélites sobrevoam a mesma estação, como ocorre com $\Delta T_{25} \rightarrow \Delta T_{35}$.

O modelo de otimização linear aplicado ao cenário 1 apresenta as seguintes características:

$$\begin{aligned}
 \max Z = & T_{X12}(x_{E12} - x_{A12}) + T_{X13}(x_{E13} - x_{A13}) + T_{X21}(x_{E21} - x_{A21}) + T_{X22}(x_{E22} - x_{A22}) \\
 & + T_{X25}(x_{E25} - x_{A25}) + T_{X34}(x_{E34} - x_{A34}) + T_{X35}(x_{E35} - x_{A35}) + T_{X35}(x'_{E35} - x'_{A35}) \\
 & + T_{X41}(x_{E41} - x_{A41}) + T_{X42}(x_{E42} - x_{A42}) + T_{X43}(x_{E43} - x_{A43}) + T_{X54}(x_{E54} - x_{A54}) \\
 & + T_{X54}(x'_{E54} - x'_{A54}) + T_{X55}(x_{E55} - x_{A55}) + T_{X61}(x_{E61} - x_{A61}) + T_{X62}(x_{E62} - x_{A62}) \\
 & + T_{X63}(x_{E63} - x_{A63}) + T_{X74}(x_{E74} - x_{A74}) + T_{X75}(x_{E75} - x_{A75}) + T_{X75}(x'_{E75} - x'_{A75}) \\
 & + T_{X81}(x_{E81} - x_{A81}) + T_{X82}(x_{E82} - x_{A82}) + T_{X85}(x_{E85} - x_{A85}) + T_{X94}(x_{E94} - x_{A94})
 \end{aligned}$$

sujeito a

$$x_{E12} - x_{A12} \geq 0$$

$$x_{E13} - x_{A13} \geq 0$$

$$x_{E21} - x_{A21} \geq 0$$

$$x_{E22} - x_{A22} \geq 0$$

$$x_{E25} - x_{A25} \geq 0$$

$$x_{E34} - x_{A34} \geq 0$$

$$x_{E35} - x_{A35} \geq 0$$

$$x_{E41} - x_{A41} \geq 0$$

$$x_{E42} - x_{A42} \geq 0$$

$$x_{E43} - x_{A43} \geq 0$$

$$x_{E54} - x_{A54} \geq 0$$

$$x_{E55} - x_{A55} \geq 0$$

$$x_{E61} - x_{A61} \geq 0$$

$$x_{E62} - x_{A62} \geq 0$$

$$x_{E63} - x_{A63} \geq 0$$

$$x_{E74} - x_{A74} \geq 0$$

$$x_{E75} - x_{A75} \geq 0$$

$$x_{E81} - x_{A81} \geq 0$$

$$x_{E82} - x_{A82} \geq 0$$

$$x_{E85} - x_{A85} \geq 0$$

$$x_{E94} - x_{A94} \geq 0$$

$$x_{A12} \geq 0$$

$$x_{A13} \geq 83$$

$$x_{A21} \geq 671$$

$$x_{A22} \geq 829$$

$$x_{A25} \geq 737$$

$$x_{E34} \geq 832$$

$$x_{A35} \geq 813$$

$$x_{A41} \geq 855$$

$$x_{A42} \geq 984$$

$$x_{A43} \geq 1093$$

$$x_{A54} \geq 1551$$

$$x_{A55} \geq 1587$$

$$x_{A61} \geq 1737$$

$$x_{A62} \geq 1859$$

$$x_{A63} \geq 1957$$

$$x_{A74} \geq 2090$$

$$x_{A75} \geq 2076$$

$$x_{A81} \geq 2242$$

$$x_{A82} \geq 2386$$

$$x_{A85} \geq 2343$$

$$x_{A94} \geq 2591$$

$$x_{E12} \leq 120$$

$$x_{E13} \leq 264$$

$$x_{E21} \leq 856$$

$$x_{E22} \geq 965$$

$$x_{E25} \leq 939$$

$$x_{E34} \leq 974$$

$$x_{E35} \leq 974$$

$$x_{E41} \leq 1054$$

$$x_{E42} \leq 1194$$

$$x_{E43} \leq 1310$$

$$x_{E54} \leq 1746$$

$$x_{E55} \leq 1746$$

$$x_{E62} \leq 2036$$

$$x_{E63} \leq 2165$$

$$x_{E74} \leq 2240$$

$$x_{E75} \leq 2240$$

$$x_{E81} \leq 2456$$

$$x_{E82} \leq 2579$$

$$x_{E85} \leq 2502$$

$$x_{E94} \leq 2807$$

$$x_{E12} < x_{A13}$$

$$x_{E21} < x_{A25}$$

$$x_{E21} < x_{A22}$$

$$x_{E25} < x_{A22}$$

$$x_{E35} < x_{A34}$$

$$x_{E41} < x_{A42}$$

$$x_{E42} < x_{A43}$$

$$x_{E54} < x_{A55}$$

$$x_{E61} < x_{A62}$$

$$x_{E62} < x_{A63}$$

$$x_{E75} < x_{A74}$$

$$x_{E81} < x_{A85}$$

$$x_{E81} < x_{A82}$$

$$x_{E85} < x_{A82}$$

$$x_{E25} < x_{A35}$$

$$x_{E21} < x_{A41}$$

$$x'_{E35} = 1023$$

$$x'_{A35} = 974$$

$$x'_{E54} = 1766$$

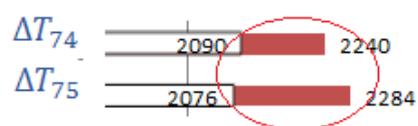
$$x'_{A54} = 1756$$

$$x'_{E75} = 2284$$

$$x'_{A75} = 2240$$

O último conjunto de restrições foi inserido para evitar perda de sinal durante as disputas nas situações em que o $EOS_{ij} > EOS_{i(j+k)}$, como ocorre com EOS_{75} . A Figura 24 ilustra essa situação.

Figura 24 - Disputa entre estações 4 e 5



Fonte: próprio autor (2016)

Sem essas últimas seis restrições é possível observar que se $T_{X74} > T_{X75}$, o modelo iria determinar que x_{E75} seria igual 2090 e durante o intervalo de tempo compreendido entre 2240 e 2284 segundos a estação m_5 não iria comunicar com o satélite n_7 . Ao fixar outro intervalo de tempo definido nesse caso como $x'_{E75} - x'_{A75}$, garante-se a cobertura do satélite nesse período.

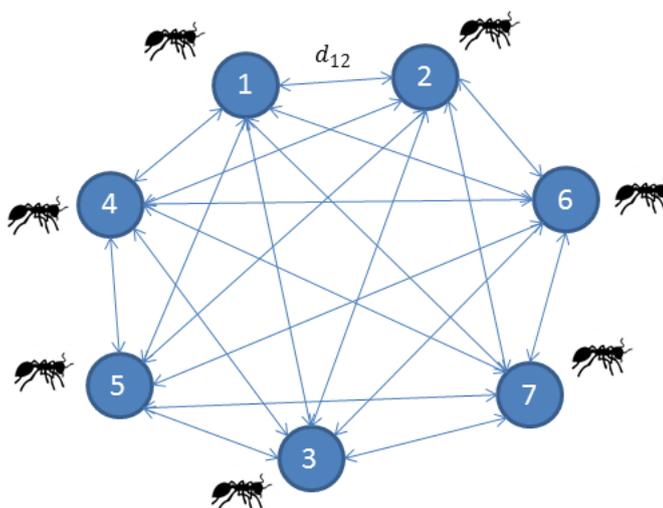
A próxima seção mostra a modelagem do problema para a solução descentralizada, via algoritmo ACO.

5.2. ADAPTAÇÃO DO ALGORITMO Ant-System (AS) AO PROBLEMA DE ALOCAÇÃO DAS ESTAÇÕES

Para realizar a adequação do algoritmo de colônia de formigas ao problema de alocação das estações, optou-se por reproduzir primeiramente o algoritmo desenvolvido originalmente para solução do problema do caixeiro viajante (PCV). Dado um grafo com n vértices que representam as cidades e m arestas, o problema consiste em determinar o caminho mais curto para percorrer todas as cidades e retornar a inicial, sem que haja repetição de cidades ao longo do trajeto.

Inicialmente foi adotado um problema com 7 cidades, apresentado na Figura 25. As distâncias são geradas aleatoriamente, podendo assumir valores entre 1 e 5.

Figura 25 - Grafo PCV utilizado no Ant System



Fonte: próprio autor (2016)

Na primeira iteração, uma formiga é colocada em cada cidade. Em seguida, cada formiga k utiliza a regra de probabilidade baseada no valor heurístico (inverso

da distância) e quantidade de feromônio de cada aresta ainda não percorrida por ela para construir seu trajeto, retornando a solução inicial.

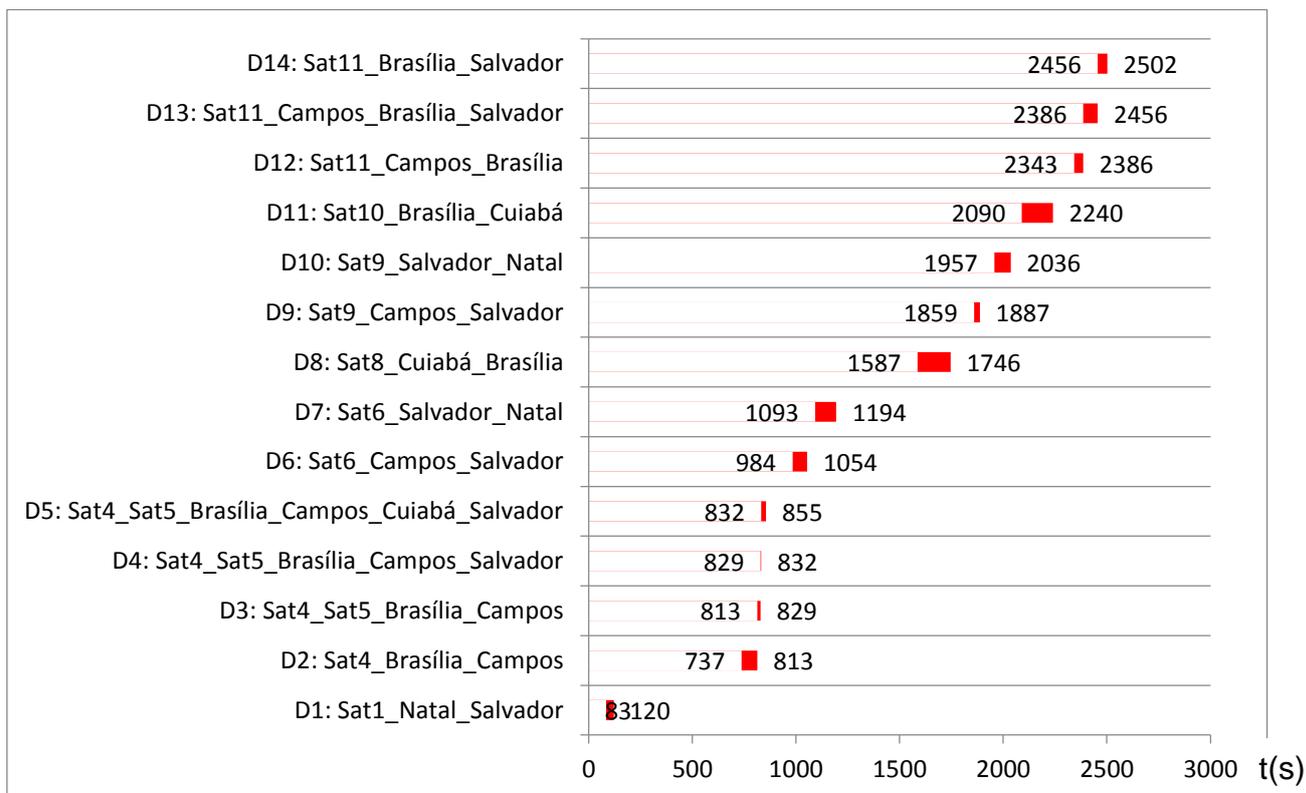
A qualidade de cada solução é avaliada a cada iteração através da função objetivo e as formigas atualizam as trilhas de feromônio de cada aresta que percorreram. As formigas que percorreram caminhos mais curtos depositam mais feromônio. Além disso, ocorre evaporação em todas as trilhas para evitar a rápida convergência das formigas, o que levaria a soluções sub-ótimas.

Esse ciclo é executado até o número máximo de iterações determinadas, ou até que todas as formigas percorram o mesmo caminho. O código implementado no software MatLab é apresentado no APÊNDICE B. Os valores escolhidos para α e β foram "1", determinados a partir de testes com o próprio algoritmo. Esses valores um número de iterações satisfatório.

Para utilização do AS no problema de alocação das estações terrestres, o problema foi dividido em subgrafos, onde cada um representa o problema de conflito para um determinado conjunto de estações e satélites, nos intervalos de tempo onde ocorrem as disputas. Onde não existe conflito não há necessidade de utilização do algoritmo. A soma das contribuições das soluções ótimas de cada subproblema fornece a quantidade máxima de dados transmitidos pela rede de satélites.

Observando novamente o gráfico da figura 24 é possível verificar as regiões de conflito. Cada área circulada representa um domínio e conflito, que gerou um subproblema representado em forma de grafo a ser resolvido pelo algoritmo de formigas. A Figura 25 ilustra de forma mais clara essas regiões de conflito, destacando as estações e satélites envolvidos em cada um deles.

Figura 26 - Domínios de conflitos

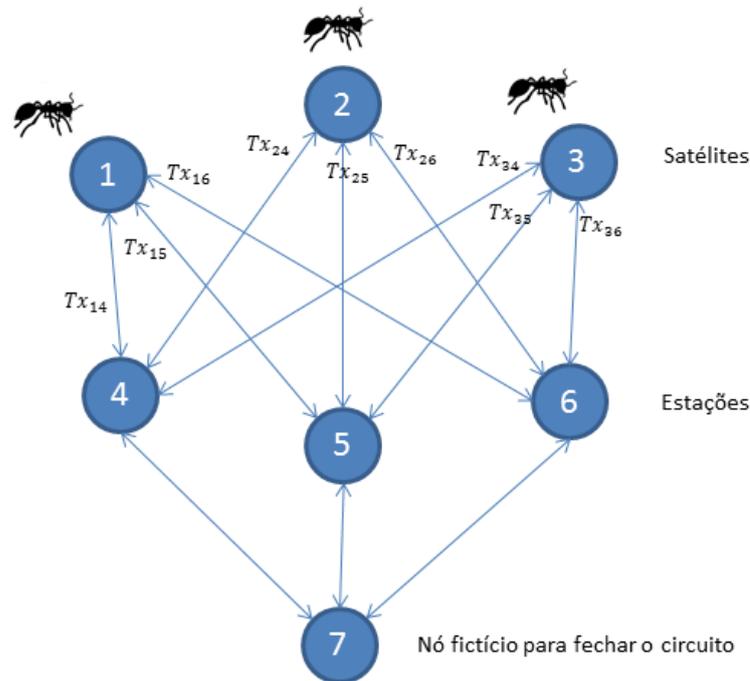


Fonte: próprio autor (2016)

As estações e os satélites envolvidos na disputa em cada domínio de conflito formam o conjunto de vértices de cada grafo. Por exemplo, o domínio de conflito 1 é gerado devido ao satélite Sat1 (n_1) estar na linha de visão das antenas de Natal (m_3) e Salvador (m_2) simultaneamente no intervalo de tempo compreendido entre 83 e 20 segundos. Para esse caso, o grafo D_1 gerado possui apenas 3 vértices: n_1 , m_3 , e m_2 , o que torna a solução bastante simples.

As arestas de cada grafo D_i , $\forall i = 1, 2, \dots, 14$ são as taxas de transmissão entre cada par estação-satélite, o que equivale “distância” do PCV. Como o problema é de maximização, a própria taxa foi a informação heurística e as formigas com trajetos mais longos são as que depositarão mais feromônio a cada iteração, modificando o algoritmo original. Um exemplo genérico de conflito com 3 estações e 3 satélites é mostrado na Figura 27.

Figura 27 - Exemplo de Grafo aplicado à alocação de estações



Fonte: próprio autor (2016)

O nó 7 foi acrescentado somente para fechar os trajetos realizados pelas formigas. Uma formiga será colocada em cada “satélite”, sendo $k = n$, o que irá garantir que todos os satélites envolvidos na disputa serão cobertos.

A divisão em subproblemas é necessária para realizar a decisão negociada. Em cada estação será processado o conflito que envolve a mesma, não necessitando de tomada de decisão centralizada.

Para a aplicação do ACO, o modelo de otimização foi modificado, já que esse algoritmo trabalha melhor com problemas de complexidade combinatória. O modelo genérico ficou da seguinte forma:

$$\max z = \sum_{i=1}^n \sum_{j=1}^m T_{Xij} * x_{ij}$$

Sujeito a

$$\sum_{j=1}^m x_{ij} \leq 1 \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} \leq 1 \quad j = 1, 2, \dots, m$$

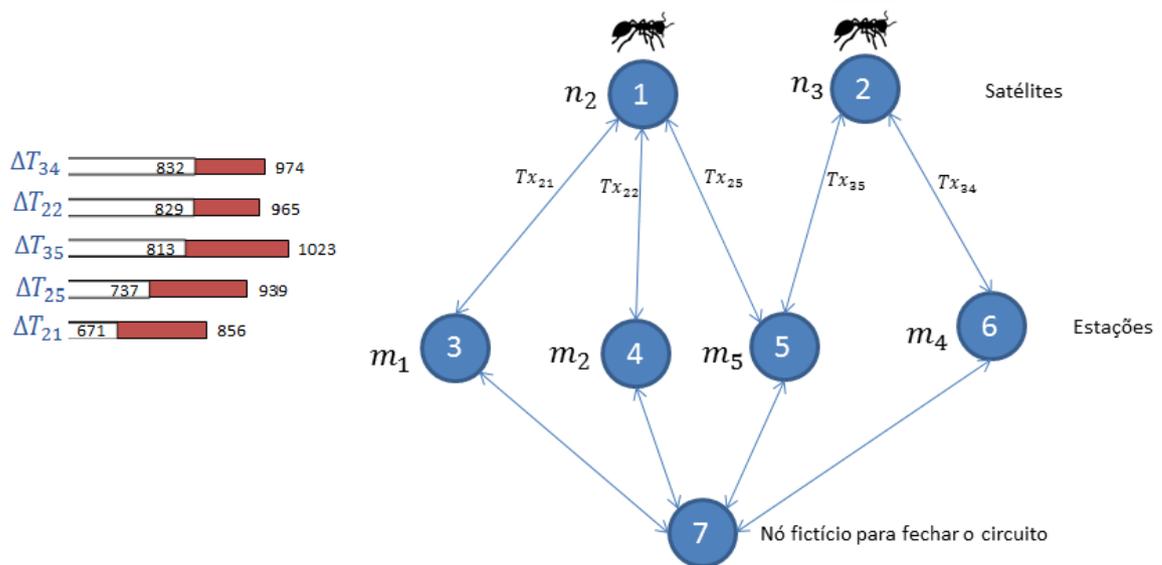
$$x_{ij} \in B;$$

Onde:

- x_{ij} é uma variável de decisão binária: assume 0 quando a estação j comunicará com o satélite i , ou 1 quando não assumirá a comunicação.
- $T_{x_{ij}}$ é a taxa de download e em bps da estação j com o satélite i ;
- n é o número de satélites considerados no modelo;
- m é o número de estações consideradas no modelo.

Para aplicar o algoritmo selecionou-se a região de conflito D_5 delimitada entre 832 e 856 segundos, como se pode observar na Figura 25. A Figura 28 apresenta o grafo gerado a partir dessa situação.

Figura 28 – Grafo do domínio de conflito D_5



Fonte: próprio autor (2016)

Para essa situação específica o submodelo apresentado é:

$$\max z_5 = T_{x_{21}} * x_{21} + T_{x_{22}} * x_{22} + T_{x_{25}} * x_{25} + T_{x_{35}} * x_{35} + T_{x_{34}} * x_{34}$$

Sujeito a

$$x_{21} + x_{22} + x_{25} \leq 1;$$

$$x_{35} + x_{34} \leq 1;$$

$$x_{25} + x_{35} \leq 1;$$

O grafo D_5 é o que apresenta o maior número de vértices nesse cenário, possuindo dois satélites: n_2 e n_3 e quatro estações: m_1 , m_2 , m_5 e m_4 . Nesse intervalo de tempo, o satélite n_2 é visto simultaneamente por pelas estações m_1 , m_2 e m_5 . n_3 é visto m_5 e m_4 . Por sua vez, a estação m_5 tem possibilidade de comunicar com os satélites n_2 e n_3 . O problema consiste em definir quais estações irão estabelecer comunicação de forma a maximizar a taxa de download nesse intervalo de tempo.

6. RESULTADOS E DISCUSSÃO

Antes de resolver o problema de otimização utilizando os métodos heurísticos baseados em inteligência de enxames, utilizou-se o método simplex para verificar a consistência do modelo.

O modelo aplicado ao cenário 1 foi resolvido utilizando-se o solver do Microsoft Excel. A Tabela 5 apresenta os resultados.

Tabela 5 - Resultados do Solver para o cenário 1

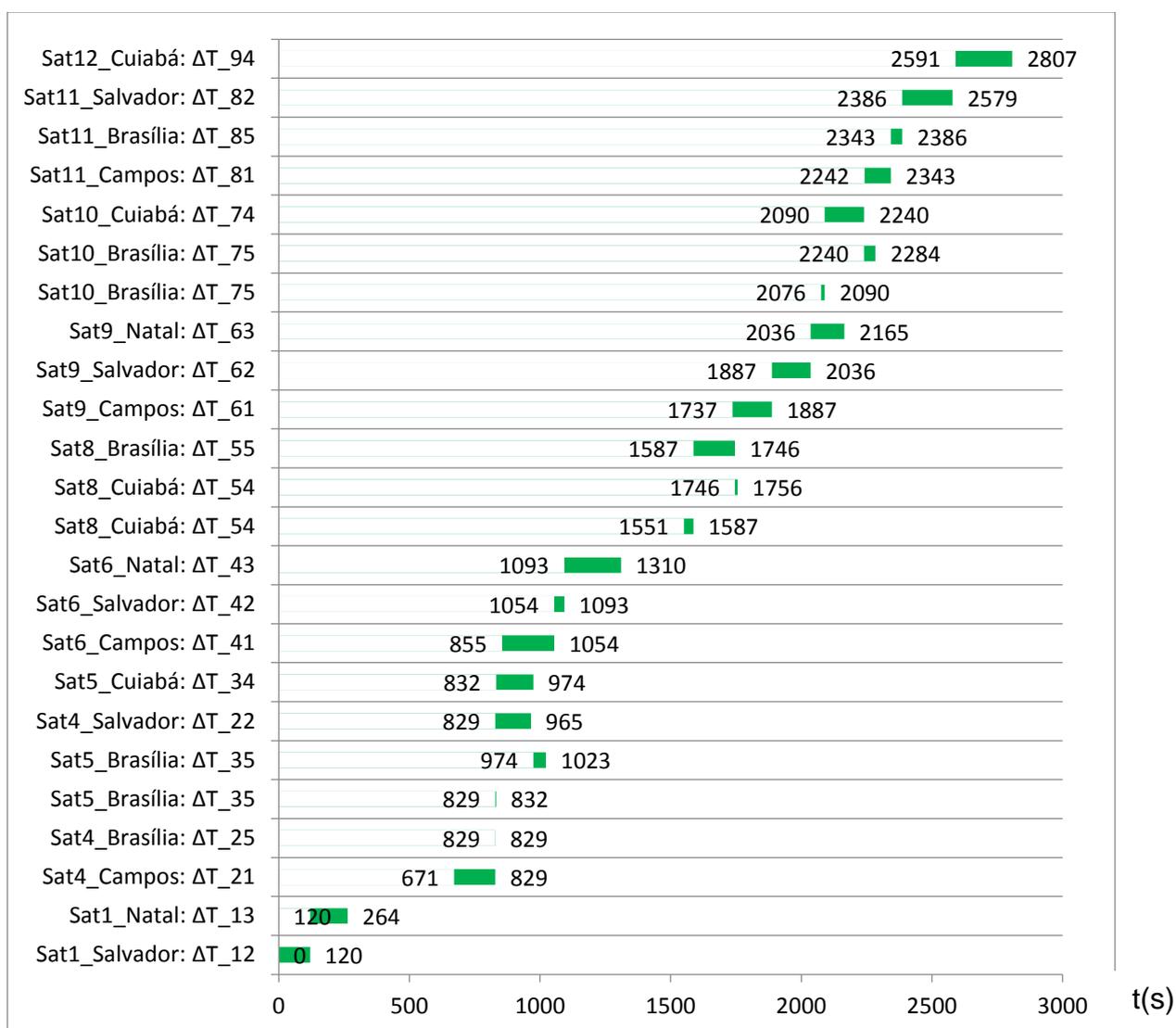
	Valores iniciais			Valores ótimos			
	AOS_{ij}	EOS_{ij}	$T_{Xij}(bps)$	X_{Aij}	X_{Eij}	Duração	$T_{Xij}(X_{Aij} - X_{Eij})$
ΔT_{12}	0	120	22000	0	120	120	2640000
ΔT_{13}	83	264	18000	120	264	144	2592000
ΔT_{21}	671	856	30000	671	829	158	4740000
ΔT_{25}	737	939	22000	829	829	0	0
ΔT_{35}	813	974	25000	829	832	3	75000
$\Delta T'_{35}$	974	1023	25000	974	1023	49	1225000
ΔT_{22}	829	965	20000	829	965	136	2720000
ΔT_{34}	832	974	36000	832	974	142	5112000
ΔT_{41}	855	1054	26000	855	1054	199	5174000
ΔT_{42}	984	1194	22000	1054	1093	39	858000
ΔT_{43}	1093	1310	24000	1093	1310	217	5208000
ΔT_{54}	1551	1746	15000	1551	1587	36	540000
$\Delta T'_{54}$	1746	1756	15000	1746	1756	10	150000
ΔT_{55}	1587	1746	18000	1587	1746	159	2862000
ΔT_{61}	1737	1887	36000	1737	1887	150	5400000
ΔT_{62}	1859	2036	30000	1887	2036	149	4470000
ΔT_{63}	1957	2165	28000	2036	2165	129	3612000
ΔT_{75}	2076	2240	22000	2076	2090	14	308000
$\Delta T'_{75}$	2240	2284	22000	2240	2284	44	968000
ΔT_{74}	2090	2240	25000	2090	2240	150	3750000
ΔT_{81}	2242	2456	18000	2242	2343	101	1818000
ΔT_{85}	2343	2502	20000	2343	2386	43	860000
ΔT_{82}	2386	2579	22000	2386	2579	193	4246000
ΔT_{94}	2591	2807	30000	2591	2807	216	6480000
					F. Objt		65808000

Fonte: próprio autor (2016)

As colunas dos valores iniciais foram utilizadas como as restrições das variáveis de decisão. As colunas dos valores ótimos mostram a solução do solver para o modelo 1, onde o valor da função objetivo que representa o total de dados recebidos nessa simulação pode ser verificado na última linha. A partir da tabela 6,

construiu-se o gráfico do cenário 1 otimizado, que permite visualizar de forma mais clara como os conflitos foram contornados.

Figura 29 - Gráfico do Cenário 1 após a otimização via solver



Fonte: próprio autor (2016)

Através da observação da Figura 29 é possível verificar que o modelo foi consistente, já que todos os conflitos foram contornados e nenhum satélite ficou fora de cobertura.

6.1. RESULTADOS DO ALGORITMO ANT SYSTEM

6.1.1. Resultados do teste do Ant System no PCV

Os resultados obtidos no algoritmo aplicado ao Problema do caixeiro viajante mostram que todas as formigas convergem até a quinta iteração, para uma

aplicação com 7 cidades. A matriz d de distâncias geradas aleatoriamente ficou como segue:

0	4	1	4	2	1	3
4	0	2	1	1	1	4
1	2	0	3	3	3	1
4	1	3	0	2	3	4
2	1	3	2	0	1	1
1	1	3	3	1	0	1
3	4	1	4	1	1	0

Na quinta iteração a matriz de soluções, onde cada linha representa o trajeto de cada formiga, ficou da seguinte maneira:

1	3	7	5	4	2	6	1
2	4	5	7	3	1	6	2
3	1	6	2	4	5	7	3
4	2	6	1	3	7	5	4
5	7	3	1	6	2	4	5
6	1	3	7	5	4	2	6
7	3	1	6	2	4	5	7

Observando com atenção, verifica-se que todos os trajetos são iguais, mudando apenas o ponto de partida de cada formiga e o sentido do percurso. Qualquer uma das linhas apresenta a solução ótima, com percurso total (soma das distâncias entre cada cidade do trajeto) igual a 8. Conclui-se que o algoritmo foi eficaz, podendo prosseguir para a adaptação problema de alocação das antenas terrestres.

6.1.2. Resultados do Ant System Aplicado ao Problema de Alocação de Estações Terrestres

Para cada grafo D_i , $\forall i = 1, 2, \dots, 14$ gerado a partir dos domínios de conflitos apresentados na seção anterior, foi adaptado o algoritmo inicial com o objetivo de obter as soluções ótimas de cada subproblema.

O domínio de conflito D_1 apresenta a seguinte função o seguinte modelo:

Max

$$\max z_1 = Tx_{12} * x_{12} + Tx_{13} * x_{13}$$

Sujeito a

$$x_{12} + x_{13} \leq 1;$$

$$x_{ij} \in B;$$

Como é possível verificar na Tabela 4, as taxas Tx_{12} e Tx_{13} são respectivamente 22000 e 18000 bps, respectivamente. A solução desse problema é trivial, e como esperado, o algoritmo atribui valor 1 para x_{12} , já que a taxa de download da estação m_2 é maior que a de m_3 . A solução ótima é:

$$x_{12} = 1;$$

$$x_{13} = 0;$$

Isso significa que durante o intervalo de tempo compreendido entre 83 e 120 segundos, a estação m_2 assumirá a comunicação com o satélite n_1 . Para o $t < 83$ s, m_2 também comunica com n_1 , já que não há nenhuma outra estação na linha de visão desse satélite. A taxa de download de 0 à 120 segundos será de 22000 bps.

De 120 à 264 segundos, a estação m_3 assume a comunicação com n_1 à uma taxa de 18000 bps, garantindo a cobertura durante toda a janela de visibilidade de n_1 .

Para todos os domínios de conflitos onde mais de uma estação comunica com um mesmo satélite, fica claro que a estação com a maior taxa de download deverá assumir a comunicação.

O domínio de conflito D_5 apresentou uma maior complexidade, envolvendo a disputa entre 2 satélites e 4 estações. O grafo foi apresentado na Figura 29. O modelo ficou da seguinte maneira:

max

$$z_5 = Tx_{21} * x_{21} + Tx_{22} * x_{22} + Tx_{25} * x_{25} + Tx_{35} * x_{35} + Tx_{34} * x_{34}$$

Sujeito a

$$x_{21} + x_{22} + x_{25} \leq 1;$$

$$x_{35} + x_{34} \leq 1;$$

$$x_{25} + x_{35} \leq 1;$$

Os valores das taxas envolvidas são:

$$Tx_{21} = 30000 \text{ bps};$$

$$Tx_{22} = 20000 \text{ bps};$$

$$Tx_{25} = 22000 \text{ bps};$$

$$Tx_{35} = 25000 \text{ bps};$$

$$Tx_{34} = 36000 \text{ bps};$$

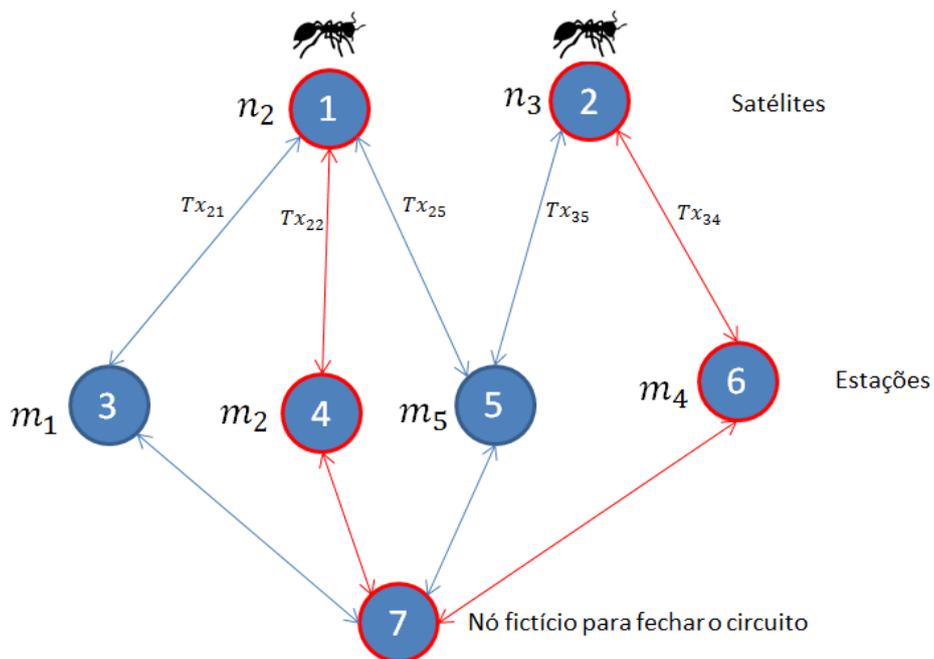
Para a solução, o algoritmo Ant System foi adaptado, onde duas formigas foram colocadas nos vértices que representam os satélites e inseridas as restrições. Os trajetos realizados pelas formigas em todos os ciclos (iterações) do algoritmo foram:

$$k_1 = 1 \quad 4 \quad 7;$$

$$k_2 = 2 \quad 6 \quad 7;$$

A Figura 30 ilustra esses trajetos ótimos.

Figura 30 - Trajeto das formigas



Fonte: próprio autor (2016)

Significa que, para maximizar a taxa total de download do domínio de conflito D_5 , a estação m_2 deverá comunicar com o satélite n_2 , enquanto m_4 comunica com n_3 . A solução ótima é:

$$x_{21} = 0;$$

$$x_{22} = 1;$$

$$x_{25} = 0;$$

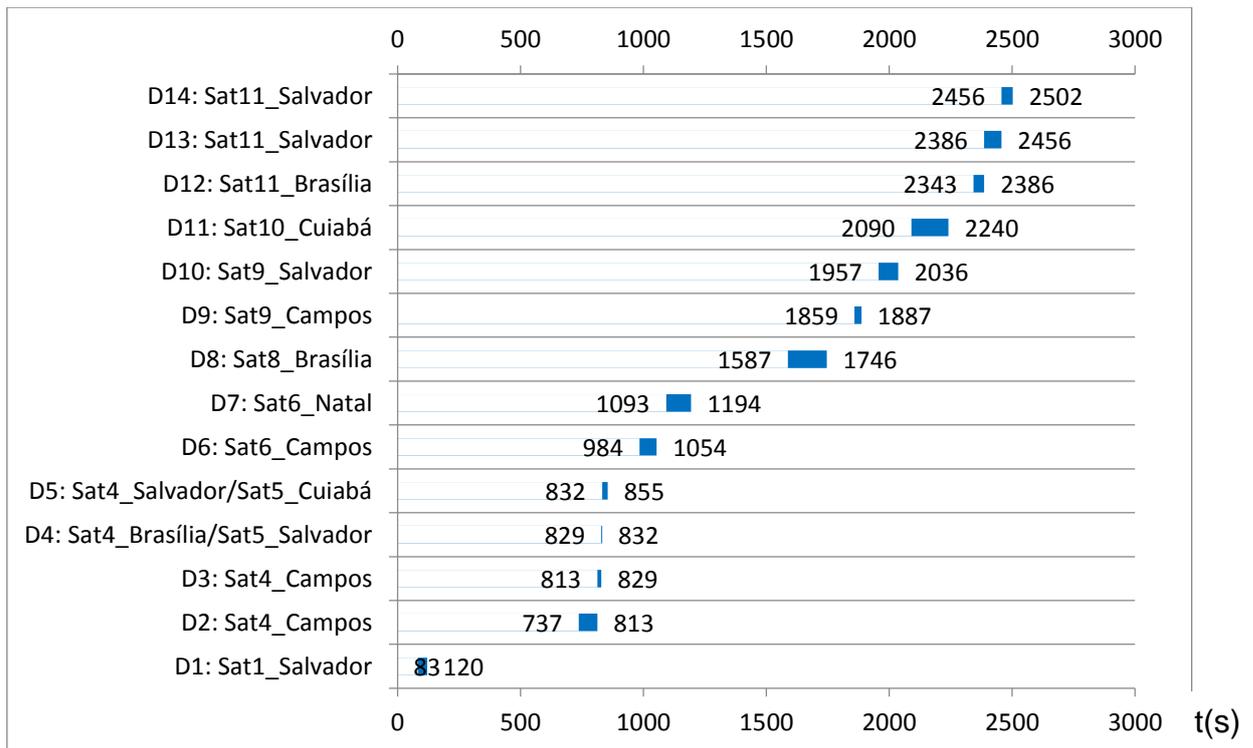
$$x_{35} = 0;$$

$$x_{34} = 1;$$

O algoritmo de formigas adaptado para esse problema é apresentado no APÊNDICE C.

As soluções ótimas para cada um dos 14 domínios de conflito podem ser visualizadas no gráfico abaixo. A solução de cada domínio é obtida de forma descentralizada através do algoritmo de cada estação envolvida naquela disputa. A Figura 31 apresenta a solução de todos os domínios.

Figura 31 - Solução dos Domínios de Conflitos



Fonte: próprio autor (2016)

Domínios:

D_1 : A solução ótima prevê que a estação de Salvador (m_2) comunique com o satélite Sat 1 (n_1) no intervalo de tempo entre 83 e 120 segundos. A outra estação envolvida na disputa é a de Natal (m_3). A solução é $x_{12} = 1$ e $x_{13} = 0$;

D_2 : A estação de Campos (m_1) deverá comunicar com o Sat 4 (n_2) entre 737 e 813 segundos. A solução é $x_{21} = 1$ e $x_{25} = 0$;

D_3 : A estação de Campos (m_1) deverá comunicar com o Sat 4 (n_2) entre 813 e 829 segundos. A solução é $x_{21} = 1$ e $x_{22} = 0$;

D_4 : A estação de Brasília (m_5) deverá comunicar com o Sat 5 (n_3) e a estação de Salvador (m_2) com o Sat 4 (n_2) entre 829 e 832 segundos. A solução é $x_{35} = 1$, $x_{22} = 1$ e $x_{25} = 0$;

D_5 : A estação de Salvador (m_2) deverá comunicar com o Sat 4 (n_2) e a estação de Cuiabá (m_4) com o Sat 4 (n_2) entre 832 e 855 segundos. A solução é $x_{21} = 0$, $x_{22} = 1$, $x_{25} = 0$, $x_{35} = 0$ e $x_{34} = 1$;

D_6 : A estação de Campos (m_1) deverá comunicar com o Sat 6 (n_4) entre 984 e 1054 segundos. A solução é $x_{41} = 1$ e $x_{42} = 0$;

D_7 : A estação de Campos (m_1) deverá comunicar com o Sat 11 (n_8) entre 1093 e 1194 segundos. A solução é $x_{41} = 1$ e $x_{42} = 0$;

D_8 : A estação de Brasília (m_5) deverá comunicar com o Sat 8 (n_5) entre 1587 e 1746 segundos. A solução é $x_{55} = 1$ e $x_{54} = 0$;

D_9 : A estação de Campos (m_1) deverá comunicar com o Sat 9 (n_6) entre 1859 e 1887 segundos. A solução é $x_{61} = 1$ e $x_{62} = 0$;

D_{10} : A estação de Salvador (m_2) deverá comunicar com o Sat 9 (n_6) entre 1957 e 2036 segundos. A solução é $x_{62} = 1$ e $x_{63} = 0$;

D_{11} : A estação de Cuiabá (m_4) deverá comunicar com o Sat 10 (n_7) entre 2090 e 2240 segundos. A solução é $x_{74} = 1$ e $x_{75} = 0$;

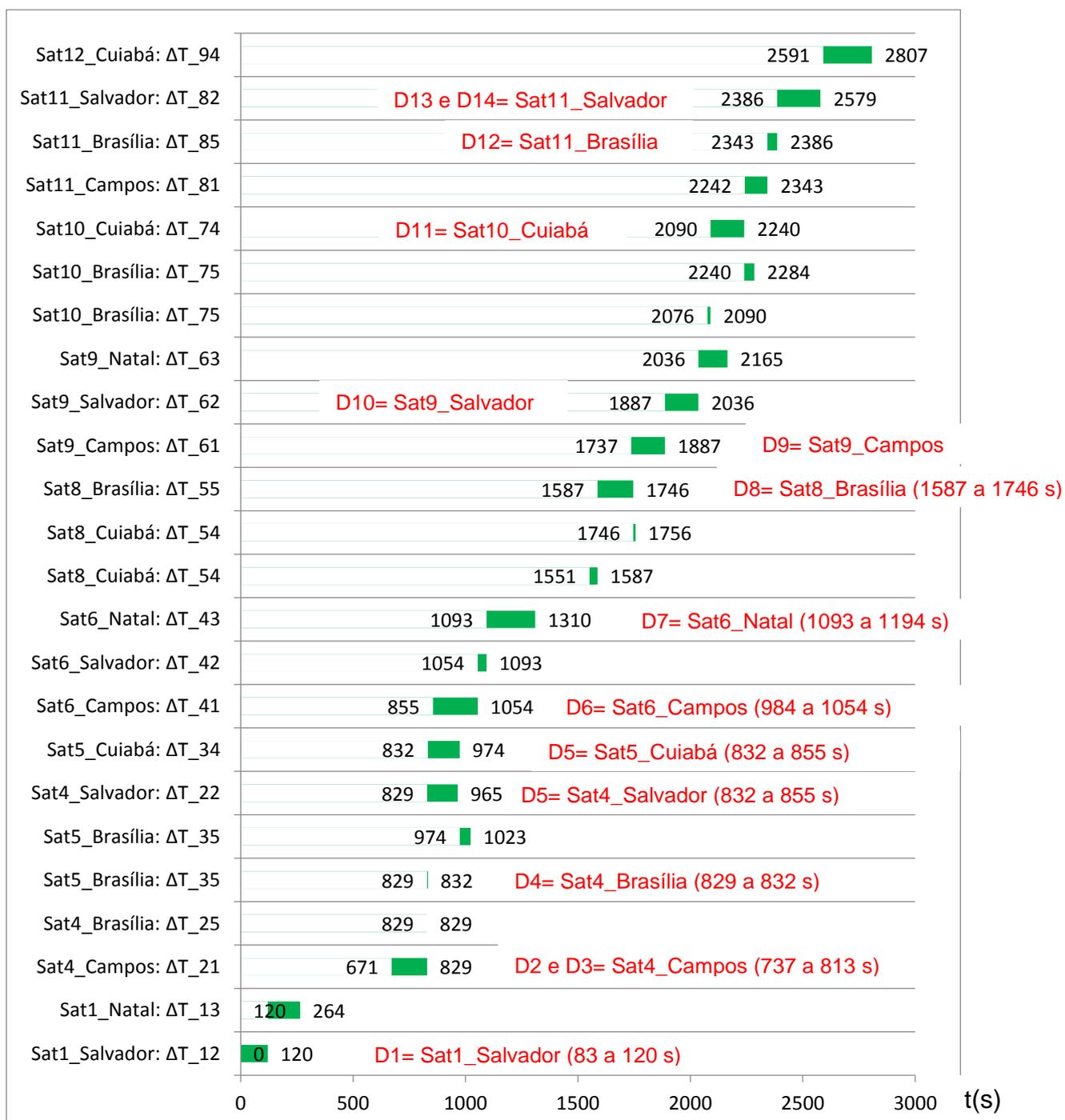
D_{12} : A estação de Brasília (m_5) deverá comunicar com o Sat 11 (n_8) entre 2343 e 2386 segundos. A solução é $x_{85} = 1$ e $x_{81} = 0$;

D_{13} : A estação de Salvador (m_2) deverá comunicar com o Sat 11 (n_8) entre 2386 e 2456 segundos. A solução é $x_{82} = 1$, $x_{85} = 0$ e $x_{81} = 0$;

D_{14} : A estação de Salvador (m_2) deverá comunicar com o Sat 11 (n_8) entre 2456 e 2502 segundos. A solução é $x_{82} = 1$, $x_{85} = 0$.

A união das soluções fornece um gráfico de alocação de estações exatamente igual ao resultado obtido pelo simplex, que pode ser observado na Figura 32. Isso mostra que o modelo descentralizado funciona adequadamente e que a solução via algoritmo de formigas foi eficaz.

Figura 32 - Solução dos Domínios de Conflitos



Fonte: próprio autor (2016)

A Figura 32 mostra em vermelho onde entram as soluções de cada domínio de conflitos solucionado pelo ACO. Por exemplo, no domínio D1 ocorria conflito entre as estações de Natal e Salvador entre 83 e 120 segundos. A solução prevê que nesse intervalo de tempo a estação de Salvador assumia a comunicação com o IFF_Sat 1. Então esse intervalo de tempo está compreendido entre o tempo total de comunicação da estação de Salvador com o Satélite 1, que é de 0 à 120 segundos.

Essa noção pode ser estendida para os demais conflitos mostrados na Figura 31 e incluídos na Figura 32.

7. CONCLUSÕES E CONSIDERAÇÕES FINAIS

As pesquisas aeroespaciais envolvendo satélites de pequeno porte têm se mostrado uma saída promissora para os problemas de investimentos, já que oferecem acesso ao ambiente espacial com um custo bem inferior aos satélites convencionais. Muitas pesquisas com os satélites conhecidos como *cubesats* estão sendo desenvolvidos e projetos envolvendo diversas universidades, como o QB-50, surgem com o objetivo de desenvolver tecnologias voltadas a esse ramo.

Como o projeto conta com diversos satélites que deverão estabelecer comunicação em rede com estações terrestres espalhadas pelo planeta, conflitos de visibilidade podem ocorrer soluções adequadas devem ser tomadas para solucionar esse problema.

Os objetivos do trabalho foram alcançados, uma vez que foram gerados dois modelos para a solução do problema de alocação das estações terrestres. O primeiro modelo de otimização linear tratou o problema de forma centralizada, obtendo a solução através do método Simplex. Já o segundo modelo, apesar de também ser linear, possui variáveis de decisão binária. O problema apresentado foi subdividido em problemas menores denominados domínios de conflitos e representados através de grafos. A solução ótima de cada grafo foi obtida através de um algoritmo de enxames baseado em colônia de formigas, chamado de ACO (*Ant Colony Optimization*).

O modelo descentralizado apresentou a mesma solução global que o modelo centralizado, se mostrando eficiente. Esse modelo permite que cada domínio de conflito seja negociado entre as estações pertencentes àquela disputa, sendo esta a principal contribuição deste trabalho.

Esse trabalho pode trazer uma contribuição importante para o projetos com nanosatélites desenvolvidos pelo Instituto Federal Fluminense em parceria com a Universidade do Porto – Portugal, fornecendo um modelo para a alocação das antenas que formam o segmento terrestre com negociação, posto que o modelo atual desenvolvido por Carvalho et al (2013) possui decisão centralizada.

Uma das limitações desse trabalho é que o modelo foi aplicado a um cenário obtido através de simulação computacional, já que não existiam ainda dados reais da rede de nanosatélites. Outra limitação consiste nos valores das taxas de

transmissão consideradas no estudo de caso ter sido arbitradas e consideradas constantes.

Um estudo posterior pode calcular e levantar as curvas taxa x tempo, que na prática não é um valor constante, já que depende de diversos fatores ambientais, características das antenas, entre outros. Em seguida aplicar esses valores no modelo.

8. REFERÊNCIAS

- ÁVILA, S. L. **Algoritmos genéticos aplicados na otimização de antenas refletoras**. Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica., 2002.
- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear programming and network flows**. 4th ed ed. Hoboken, N.J: John Wiley & Sons, 2010.
- BECANOVIC, V. et al. **HUGIN: a small satellite trying to be intelligent**. (T. Lindblad, M. L. Padgett, J. M. Kinser, Eds.), 1999. Disponível em: <<http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=990265>>. Acesso em: 9 jan. 2017
- BHATTACHARYYA, B.; RAJ, S. Swarm intelligence based algorithms for reactive power planning with Flexible AC transmission system devices. **International Journal of Electrical Power & Energy Systems**, v. 78, p. 158–164, jun. 2016.
- BISWAS, A. et al. Synergy of PSO and bacterial foraging optimization—a comparative study on numerical benchmarks. In: **Innovations in Hybrid Intelligent Systems**. Springer, 2007. p. 255–263.
- BRADFORD W. PARKINSON; JAMES J. SPILKER JR.; PENINA AXELRAD. **Global Positioning System: Theory and Applications**. American Institute of Aeronautics and Astronautics, 1996. v. Volume I
- BROWNLEE, J. **Clever Algorithms: Nature-Inspired Programming Recipes**. Jason Brownlee, 2011.
- CARVALHO, R. A. de; LISBOA, W. S.; HISSA, L. R.; MOURA, G. L.; AMADURO, L. R.; CORDEIRO, C. S. . **The Software Architecture for the RIBRAS Ground Station Network**. In: 1st IAA Latin America CubeSat Workshop, 2014, Brasília. Proceedings of the 1st IAA Latin America CubeSat Workshop, 2014.
- CARVALHO, R. A. de; CORDEIRO, C. S.; MOURA, G. L. **Optimizing Data Download using Clusters of Networked Ground Stations: A Model for the QB50 Project**. In: 5TH EUROPEAN CUBESAT SYMPOSIUM. Bruxelas, 2013.
- CHO, S.; BYUN, H. A Space-Time Graph Optimization Approach Based on Maximum Cliques for Action Detection. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 26, n. 4, p. 661–672, abr. 2016.
- CORRAO, G. et al. **Ground station activity planning through a multi-algorithm optimisation approach**. Satellite Telecommunications , 2012 IEEE First AESS European Conference on. **Anais...IEEE**, 2012. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6400125>. Acesso em: 25 mar. 2016

DAS, S.; KONAR, A. A swarm intelligence approach to the synthesis of two-dimensional IIR filters. **Engineering Applications of Artificial Intelligence**, v. 20, n. 8, p. 1086–1096, dez. 2007.

DENBY, B.; LE HÉGARAT-MASCLE, S. Swarm intelligence in optimisation problems. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 502, n. 2–3, p. 364–368, abr. 2003.

DENEUBOURG, J.-L. et al. The self-organizing exploratory pattern of the argentine ant. **Journal of insect behavior**, v. 3, n. 2, p. 159–168, 1990.

DORIGO, M. et al. **Positive Feedback as a Search Strategy**, 1991.

DOS SANTOS COELHO, L.; DA COSTA SILVEIRA, C. **Improved bacterial foraging strategy for controller optimization applied to robotic manipulator system**. 2006 IEEE Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control. **Anais...** In: 2006 IEEE COMPUTER AIDED CONTROL SYSTEM DESIGN, out. 2006

EBERHART, R. C.; KENNEDY, J. **A new optimizer using particle swarm theory**. Proceedings of the sixth international symposium on micro machine and human science. **Anais...** New York, NY, 1995 Disponível em: <http://www.ppgia.pucpr.br/~alceu/mestrado/aula3/PSO_2.pdf>. Acesso em: 11 jan. 2017

ELBERT, B. R. **The satellite communication ground segment and earth station handbook**. Boston: Artech House, 2001.

ESPOSITO, A. **Programação paralela e sequencial aplicada à otimização de estruturas metálicas com o algoritmo PSO**. Dissertação de Mestrado—Porto Alegre - RS: Universidade Federal do Rio Grande do Sul, Dezembro de 2012.

EVANS, B. et al. Integration of satellite and terrestrial systems in future multimedia communications. **IEEE Wireless Communications**, v. 12, n. 5, p. 72–80, out. 2005.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta à teoria dos grafos. **Disponível em <http://www.ime.usp.br/~pf/teoriadosgrafos>**, 2011.

FERGUSON, R. O.; SARGENT, L. F. **Linear programming**. McGraw-Hill, 1958.

FU, M. C. Optimization for simulation: Theory vs. practice. **INFORMS Journal on Computing**, v. 14, n. 3, p. 192–215, 2002.

GALE, D. Linear programming and the simplex method. **Notices of the AMS**, v. 54, n. 3, p. 364–369, 2007.

GAVRILOVICH, I. et al. **Robotic test bench for CubeSat ground testing: Concept and satellite dynamic parameter identification**. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). **Anais...** In: 2015 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS). set. 2015

GONG, M. et al. Discrete particle swarm optimization for high-order graph matching. **Information Sciences**, v. 328, p. 158–171, 20 jan. 2016.

GUERTIN, S. M.; AMRBAR, M.; VARTANIAN, S. **Radiation Test Results for Common CubeSat Microcontrollers and Microprocessors**. 2015 IEEE Radiation Effects Data Workshop (REDW). **Anais...** In: 2015 IEEE RADIATION EFFECTS DATA WORKSHOP (REDW). jul. 2015

HAYKIN, S. **Communication Systems**. 4^a ed. [s.l.] John Wiley & Sons, Inc., 2000.

HOFSTAD, R. VAN DER. **Random Graphs and Complex Networks**. [s.l.: s.n.]. v. 1

ISHII, K.; MIKI, T. Mobile robot platforms for artificial and swarm intelligence researches. **International Congress Series**, v. 1301, p. 39–42, jul. 2007.

JOVANOVIC, R.; TUBA, M.; VOSS, S. An ant colony optimization algorithm for partitioning graphs with supply and demand. **Applied Soft Computing**, 2016.

JUNIOR, P.; FERREIRA, O. P. Simulação e otimização: desenvolvimento de uma ferramenta de análise de decisão para suprimento de refinarias de petróleo através de uma rede de oleodutos. 2001.

KIM, D. H.; ABRAHAM, A.; CHO, J. H. A hybrid genetic algorithm and bacterial foraging approach for global optimization. **Information Sciences**, v. 177, n. 18, p. 3918–3937, 15 set. 2007.

KORANI, W. M.; DORRAH, H. T.; EMARA, H. M. **Bacterial foraging oriented by Particle Swarm Optimization strategy for PID tuning**. 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA). **Anais...** In: 2009 IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION (CIRA). dez. 2009

LEE, J. et al. **Visibility conflict resolution for multiple antennae and multi-satellites via genetic algorithm**. Aerospace Conference, 2013 IEEE. **Anais...IEEE**, 2013Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6496849>. Acesso em: 28 abr. 2016

LIU, H.; ABRAHAM, A.; CLERC, M. Chaotic dynamic characteristics in swarm intelligence. **Applied Soft Computing**, v. 7, n. 3, p. 1019–1026, jun. 2007.

MILLONAS, M. M. **Swarms, phase transitions, and collective intelligence**. SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-. **Anais...**ADDISON-WESLEY PUBLISHING CO, 1994

NASSIF, L. N. Internet via Satélite: as expectativas da comunicação em banda larga e as implicações tecnológicas. **Revista IP**, v. 3, p. 2, 2001.

NETO, A. O que é o programa espacial brasileiro? **Superinteressante**, 19 maio 2015.

NIU, B. et al. Swarm intelligence algorithms for Yard Truck Scheduling and Storage Allocation Problems. **Neurocomputing**, nov. 2015.

NIU, B.; WANG, H. Bacterial Colony Optimization. **Discrete Dynamics in Nature and Society**, v. 2012, p. 1–28, 2012.

PASSINO, K. M. **Distributed optimization and control using only a germ of intelligence**. Proceedings of the 2000 IEEE International Symposium on Intelligent Control, 2000. **Anais...** In: PROCEEDINGS OF THE 2000 IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, 2000. 2000

QB50 PROJECT, What is a CubeSat? Disponível em: <<https://www.qb50.eu/index.php/project-description-obj/what-is-a-cubesat>>. Acesso em: 27 mar. 2016.

RIEBEEK, H. **Catalog of Earth Satellite Orbits : Feature Articles**. Setembro de 2009. Disponível em: <<http://earthobservatory.nasa.gov/Features/OrbitsCatalog/page2.php>>. Acesso em: 10 jan. 2017.

RODRIGUES, A. Qual é o problema do programa espacial brasileiro? **revistagalileu.globo.com**, 7 jan. 2015.

SABIR, M. M.; ALI, T. Optimal PID controller design through swarm intelligence algorithms for sun tracking system. **Applied Mathematics and Computation**, v. 274, p. 690–699, fev. 2016.

SANCHEZ-SANJUAN, S.; GONZALEZ-LLORENTE, J.; HURTADO-VELASCO, R. Comparison of the Incident Solar Energy and Battery Storage in a 3U CubeSat Satellite for Different Orientation Scenarios. **Journal of Aerospace Technology and Management**, v. 8, n. 1, p. 91–102, 7 mar. 2016.

SANCTIS, M. D. et al. Satellite Communications Supporting Internet of Remote Things. **IEEE Internet of Things Journal**, v. 3, n. 1, p. 113–123, fev. 2016.

SERAPIÃO, A. B. DE S. Fundamentos de otimização por inteligência de enxames: uma visão geral. **Sba: Controle & Automação. Sociedade Brasileira de Automatica**, v. 20, n. 3, p. 271–304, set. 2009.

SOMA, P. et al. **Multi-Satellite Scheduling Using Genetic Algorithms**. American Institute of Aeronautics and Astronautics, 17 maio 2004. Disponível em: <<http://arc.aiaa.org/doi/10.2514/6.2004-743-515>>. Acesso em: 9 jan. 2017

SOUZA, A. L. G. DE. **Sistema de apontamento de antena para estação de comunicação com satélites**. 2013. 77 f., il. Dissertação (Mestrado em Engenharia Elétrica)—Universidade de Brasília, Brasília, 2013. Disponível em: <<http://repositorio.unb.br/handle/10482/13847>>. Acesso em: 6 dez. 2016.

STORN, R.; PRICE, K. **Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces**. [s.l.] ICSI Berkeley, 1995. v. 3

TAKEMURA, Y.; SATO, M.; ISHII, K. Toward realization of swarm intelligence mobile robots. **International Congress Series**, v. 1291, p. 273–276, jun. 2006.

VALDASTRI, P. et al. Micromanipulation, communication and swarm intelligence issues in a swarm microrobotic platform. **Robotics and Autonomous Systems**, v. 54, n. 10, p. 789–804, out. 2006.

VAN STEEN, M. **Graph theory and complex networks**, 2010. v. 144

WERNER, M. et al. Analysis of system parameters for LEO/ICO-satellite communication networks. **IEEE Journal on Selected Areas in Communications**, v. 13, n. 2, p. 371–381, fev. 1995.

WERNER, M. A dynamic routing concept for ATM-based satellite personal communication networks. **IEEE Journal on Selected Areas in Communications**, v. 15, n. 8, p. 1636–1648, out. 1997.

WU, W. W. et al. Mobile satellite communications. **Proceedings of the IEEE**, v. 82, n. 9, p. 1431–1448, set. 1994.

XUE, Y. et al. Small satellite remote sensing and applications – history, current and future. **International Journal of Remote Sensing**, v. 29, n. 15, p. 4339–4372, ago. 2008.

ZAHIRI, S.-H.; SEYEDIN, S.-A. Swarm intelligence based classifiers. **Journal of the Franklin Institute**, v. 344, n. 5, p. 362–376, ago. 2007.

ZHANG, D. et al. Adaptive task assignment for multiple mobile robots via swarm intelligence approach. **Robotics and Autonomous Systems**, v. 55, n. 7, p. 572–588, jul. 2007.

APÊNDICE A - ANÁLISE BIBLIOMÉTRICA “SWARM INTELLIGENCE”

Com o objetivo de realizar uma busca coerente de artigos para constituírem o núcleo da revisão bibliográfica optou-se por realizar uma pesquisa bibliométrica a fim de selecionar artigos relevantes dentre os mais antigos, mais citados e mais recentes. O levantamento bibliográfico foi realizado nas fontes digitais, utilizando-se de mecanismos de busca como Google e Google Acadêmico, Biblioteca Digital de Teses e Dissertações (BDTD) e na base SciELO.

Como principal base de pesquisa foi selecionada a Scopus (Elsevier), devido ao seu grande número de artigos indexados.

Para o levantamento bibliométrico foram utilizados alguns indicadores, como ano de publicação, autor, tipo de trabalho e idioma.

Ano de publicação: Inicialmente não se limitou a busca por data para observar os arquivos mais relevantes, obtendo trabalhos recentes e antigos. O objetivo era verificar a evolução ao longo dos anos.

Idiomas: Buscou-se por publicações em inglês, português e Espanhol.

Autores: Inicialmente a busca não foi limitada a autores, e num segundo momento foi dada preferência a autores mais citados.

Periódicos: Revistas e periódicos relacionados com o tema.

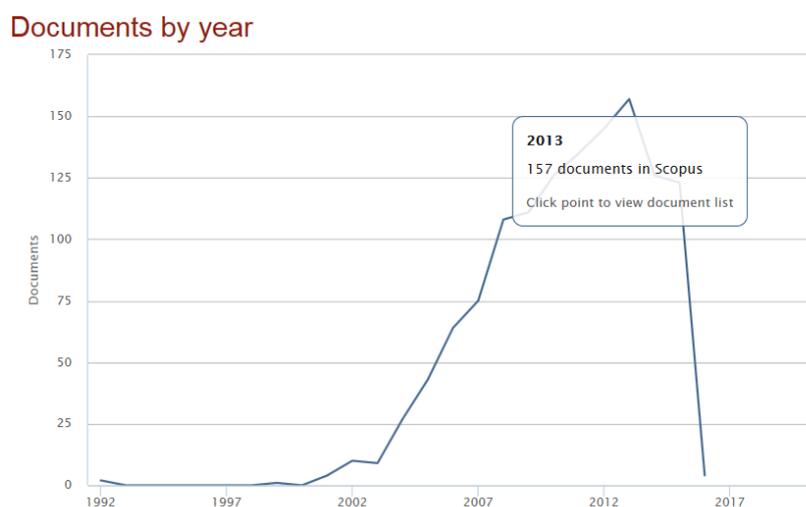
Tipos de Documentos: Trabalhos acadêmicos, artigos e trabalhos apresentados em eventos.

Pesquisa na amostra

A base de pesquisa escolhida aqui foi a Scopus (Elsevier), devido a sua representatividade. Não se limitou a janela de tempo da amostra.

Foi utilizada a frase “Swarm Intelligence” limitada apenas aos títulos. Nesta seção serão apresentados alguns gráficos em relação aos indicadores bibliométricos apresentados. A busca retornou 1270 documentos, onde a distribuição temporal pode ser vista na figura 2.

Figura 2 – Distribuição temporal das Publicações



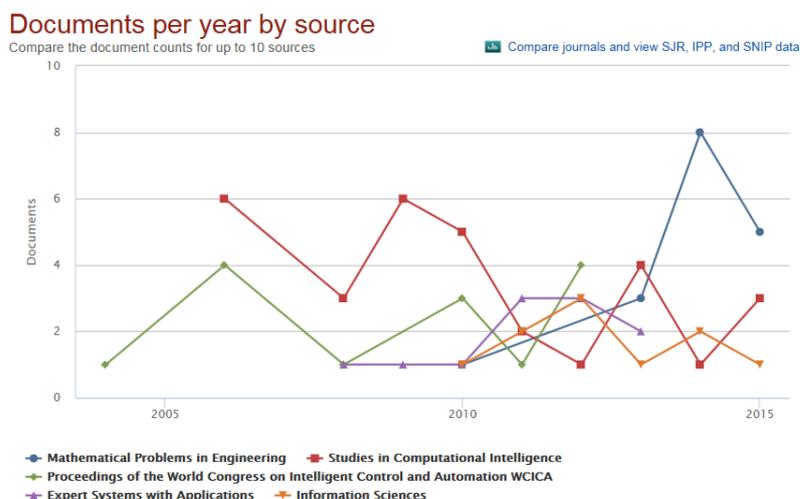
Fonte: Scopus (2014)

As primeiras publicações falando sobre inteligência de exames ocorreram nos anos noventa, o que coincide com o início da difusão da informática. Porém os estudos seguiram uma tendência crescente após os anos 2000, tendo seu pico com 157 artigos publicados em 2013. Essa tendência crescente mostra que essa área de estudo ainda está sendo bastante explorado pelos pesquisadores.

Identificação dos periódicos com maior número de artigos relacionados

Foram identificados os periódicos que mais continham artigos relacionados ao tema em questão. A figura 3 apresenta a distribuição de artigos por periódicos.

Figura 3 – Artigos por periódicos



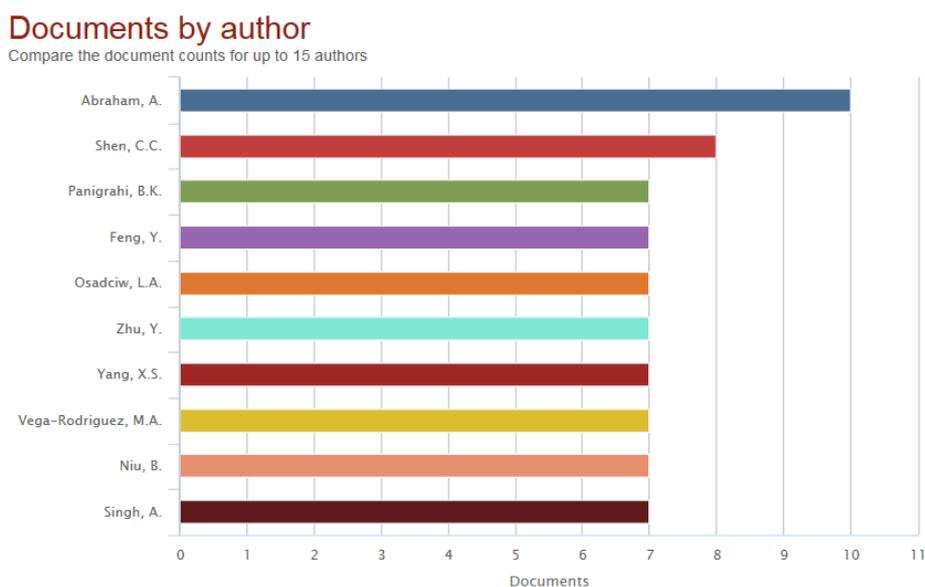
Fonte: Scopus (2015)

A figura 3 lista os cinco periódicos com maior número de artigos publicados sobre inteligência de enxames em todo período analisado.

Identificação dos autores com maior número de publicações

Foram selecionados os 10 autores que mais publicaram em relação ao tema. Na figura 4 é possível observar os números de publicações de cada um deles.

Figura 4 – Artigos por autor



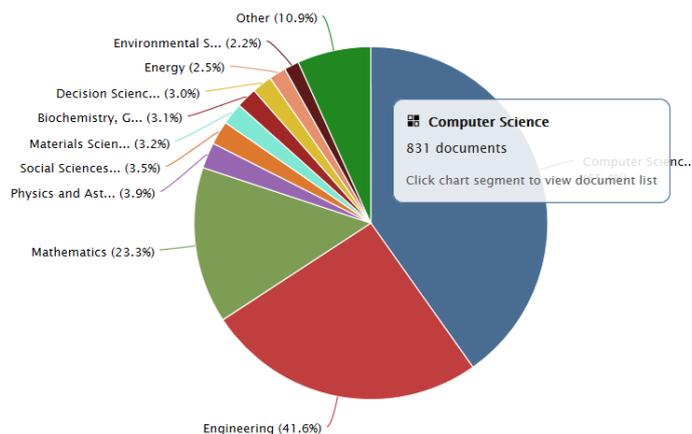
O autor com mais publicações indexadas nessa base foi o Ajith Abraham, pesquisador do Machine Intelligence Research Labs (MIR Labs) - Scientific Network for Innovation and Research Excellence, Auburn, United States. Esse autor possui 854 documentos publicados nas áreas de Ciência da Computação, Engenharias, entre outras, citados por 6332 artigos.

Através da leitura posterior dos artigos identificou-se o Marco Dorigo como um autor relevante e pioneira na otimização baseada em colônia de formigas.

Identificação das áreas de concentração dos artigos

Figura 5 – artigos por áreas de estudo

Documents by subject area



Fonte – Scopus (2015)

Observando a figura 5 é possível identificar que a maioria dos artigos publicados sobre inteligência de enxames é na área de Ciências da Computação, com 831 artigos, o que representa 65,4 % do total de documentos publicados. Na sequência vem a área de Engenharia com 41,6 %. Justamente nessa área que se encontra a linha de Pesquisa Operacional, onde a swarm intelligence vem sendo muito aplicada para otimização de processos.

Seleção dos artigos para o núcleo de partida

Foram selecionados 22 artigos através da busca direta pelo termo “swarm intelligence”, divididos em 2 grupos: Mais recentes e relevantes, segundo critérios de relevância da própria base, e mais antigos. Foram selecionados apenas os artigos que se encontravam disponíveis para download. Esses artigos se encontram discriminados por título, autor e ano nas tabelas 1 e 2.

Tabela 1 – 12 artigos mais antigos

Título	Autor	Ano
A swarm intelligence approach to the synthesis of two-dimensional IIR filters.	Swagatam Das, Amit Konar	2007
A first mathematical model of brood sorting by ants: Functional self-organization without swarm-intelligence, Ecological Complexity.	Laurent Gaubert, Pascal Redou, Fabrice Harrouet, Jacques Tisseau	2007

Swarm intelligence based classifiers.	Seyed-Hamid Zahiri, Seyed-Alireza Seyedin	2007
Adaptive task assignment for multiple mobile robots via swarm intelligence approach.	Dandan Zhang, Guangming Xie, Junzhi Yu, Long Wang	2007
Mobile robot platforms for artificial and swarm intelligence researches.	Kazuo Ishii, Tsutomu Miki	2007
Chaotic dynamic characteristics in swarm intelligence.	Hongbo Liu, Ajith Abraham, Maurice Clerc	2007
Micromanipulation, communication and swarm intelligence issues in a swarm microrobotic platform	P. Valdastri, P. Corradi, A. Menciassi, T. Schmickl, K. Crailsheim, J. Seyfried, P. Dario, Micromanipulation	2006
ANSI: A swarm intelligence-based unicast routing protocol for hybrid ad hoc networks.	Sundaram Rajagopalan, Chien-Chung Shen,	2006
Toward realization of swarm intelligence mobile robots.	asunori Takemura, Masanori Sato, Kazuo Ishii	2006
Collective and distributive swarm intelligence: evolutionary biological survey.	Harold Szu, Pornchai Chanyagorn, William Hwang, Mike Paulin, Takeshi Yamakawa	2004
Swarm intelligence in optimisation problems.	B. Denby, S. Le Hégarat-Masclé	2003
Formation of an ant cemetery: swarm intelligence or statistical accident?	Marc Martin, Bastien Chopard, Paul Albuquerque	2002

Fonte: autor

É importante destacar a respeito da tabela 1 que os artigos discriminados são os mais antigos que se obteve acesso através das bases eletrônicas de pesquisa. Existiam artigos mais antigos, porém apenas com o abstract disponível.

Tabela 2 – 10 artigos mais novos e relevantes

Título	Autor	Ano
Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms.	Alan Oliveira de Sá, Nadia Nedjah, Luiza de Macedo Mourelle	2016
Swarm intelligence algorithms for macroscopic traffic flow model validation	Adam Poole, Apostolos Kotsialos	2016

with automatic assignment of fundamental diagrams.		
Swarm intelligence based algorithms for reactive power planning with Flexible AC transmission system devices.	Biplab Bhattacharyya, Saurav Raj	2016
Optimal PID controller design through swarm intelligence algorithms for sun tracking system.	Mirza Muhammad Sabir, Tariq Ali	2016
Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems - Part 2: Constrained optimization.	Adil Baykasoğlu, Şener Akpınar	2015
Swarm intelligence algorithms for Yard Truck Scheduling and Storage Allocation Problems.	Ben Niu, Ting Xie, Lijing Tan, Ying Bi, Zhengxu Wang	2015
Optimization of retaining wall design using recent swarm intelligence techniques.	Amir H. Gandomi, Ali R. Kashani, David A. Roke, Mehdi Mousavi	2015
A new hybrid optimization method inspired from swarm intelligence: Fuzzy adaptive swallow swarm optimization algorithm (FASSO).	Mehdi Neshat, Ghodrat Sepidname	2015
Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems - Part 1: Unconstrained optimization.	Adil Baykasoğlu, Şener Akpınar	2015
An efficient two-level swarm intelligence approach for RNA secondary structure prediction with bi-objective minimum free energy scores.	Soniya Lalwani, Rajesh Kumar, Nilama Gupta	2015

Fonte: autor

Na tabela 2 encontram-se listados os artigos mais recentes encontrados na base Scopus e na Science Direct sobre “swarm intelligence”. Ao longo do desenvolvimento do artigo foram verificadas outras fontes e referências relevantes ao tema pesquisado.

APÊNDICE B - CÓDIGO DO ALGORITMO ANT SYSTEM APLICADO AO PCV

```

%inicialização de constantes e variáveis
m = 7; %número de formigas utilizadas;
n = 7; %número de cidades;
rho=0.5; %taxa de evaporação
max_iter =5; %número máximo de iterações
alpha=1; %influência do feromônio;
Beta = 1; %influência da informação heurística;
tau=ones([n n]); % Inicializando matriz de feromônio (inicializa com 1,
pois tem que ser números maiores que zero);
Eta=zeros([n n]);
soma= zeros ([1 m])
p= zeros ([m n])
pbest=0;
L=zeros([1 m]); %Distância total da viagem S de cada formiga k%
Lmin=1000; %menor distância total encontrada por uma formiga k%
S=zeros([1 m+1]); %Viagem ótima%
Q=1;

%gerando matriz aleatória de distâncias d
d = randi([1 5], n);
for i = 1:1:n
    for j = 1:1:i
        if (i==j)
            d(i,j) = 0;
        else
            d(j,i) = d(i,j);
        end
    end
end
d

%Gerando valores heurísticos Eta(Eta=1/d)
for i= 1:1:n
    for j=1:1:n
        if d(i,j)==0
            Eta(i,j)=0;
        else
            Eta(i,j)= 1/d(i,j);
        end
    end
end
Eta

%Inicializando matriz de feromônios
for i= 1:1:n
    for j=1:1:n
        if i==j
            tau(i,j)=0;
        else
            tau(i,j)= 1;
        end
    end
end
tau

iter=1;%inicializando contador de iterações%

```

```

while iter <= max_iter

%Inicializando uma formiga em cada cidade
for k=1:1:m
    for j=1:1:n+1
        if j==1
            formigas(k,j)=k;
        else
            formigas(k,j)=0;
        end
    end
end

formigas;

for a=1:1:n %transições de cidades para cada formiga
a;
% soma das possibilibilades de transições factíveis das formigas
for k=1:1:m %formigas

    for i= 1:1:n %possibilidade de transição para cidade i
        aux=0;
        for j=1:1:a
            if formigas(k,j)==i
                aux=1;
            end
        end
        if aux==0

soma(k)=soma(k)+(tau(formigas(k,a),i)^alpha*Eta(formigas(k,a),i)^Beta);
            end

        end
end
%soma;

%cálculo das probabilidades de transição de cada formiga
for k=1:1:m %formigas

    for i= 1:1:n %cidade
        aux=0;
        for j=1:1:a
            if formigas(k,j)==i
                aux=1;
            end
        end
        if aux==0
            p(k,i)=
((tau(formigas(k,a),i)^alpha*Eta(formigas(k,a),i)^Beta))/soma(k);%cálculo
da probabilidade de transição de cidade i para cidade j%
            end

            if p(k,i)> pbest
                formigas(k,a+1)=i;
                pbest=p(k,i);
            end
        end
    end
end

```

```

        if a==n % a última coluna da matriz formigas é sempre igual a primeira
        coluna (formiga sempretorna a cidade de partida%
            formigas(k,a+1)= formigas(k);
        end

    end
    pbest=0;
end
p;
iter;
formigas;
soma= zeros ([1 m]);
p= zeros ([m n]);
end
formigas
fprintf ('Ciclo %d', iter);
%Calculando distância total do percurso de cada formiga%
for k=1:1:m
    L(k)=0;
    for j=1:1:n
        x=formigas(k,j);
        y=formigas(k,j+1);
        L(k)=L(k)+d(x,y);
    end
    if L(k)<= Lmin % menor ou igual para alterar o valor da solução ótima
        Lmin=L(k);
        formiga=k;
        for j=1:1:n+1
            S_otimo(j)=formigas(formiga,j);
        end
    end
end

end
L
Lmin
%mostra qual melhor solução até o momento
S_otimo

%inicializando matriz de acréscimo de feromônios%
delta_tau=zeros([n n]);

%Cálculo da acréscimo de ferômonio para cada caminho percorrido pelas
formigas
for k=1:1:m %formigas
    for a=1:1:n %auxiliar para varrer matriz das formigas
        for i= 1:1:n %varrer matriz de feromônios%
            for j= 1:1:n
                if (formigas(k,a)==i && formigas(k,a+1)==j)
                    delta_tau(i,j)= delta_tau(i,j) + Q/L(k);
                    delta_tau(j,i)= delta_tau(i,j); %o acréscimo de
feromônio é feito em tau(i,j) e tau (j,i) simultaneamente, gerando uma
matriz simétrica%
                end
            end
        end
    end
end
end
delta_tau;

```

```
%Atualização da matriz de feromônio: acréscimo e evaporação%
for i=1:1:n
    for j=1:1:n
        tau(i,j)=(1-rho)*tau(i,j)+ delta_tau(i,j);
    end
end
tau
Eta
iter=iter+1;
end
```

APÊNDICE C - CÓDIGO DO ALGORITMO ANT SYSTEM APLICADO AO DOMÍNIO DE CONFLITO D5

```

clear all
clc
%inicialização de constantes e variáveis
m = 2; %número de formigas utilizadas;
n = 7; %número de cidades;
trans = 3;
rho=0.5; %taxa de evaporação
max_iter =5; %número máximo de iterações
alpha=1; %influência do feromônio;
Beta = 1; %influência da informação heurística;
tau=ones([n n]); % Inicializando matriz de feromônio (inicializa com 1,
pois tem que ser números maiores que zero);
Eta=zeros([n n]);
soma= zeros ([1 m])
p= zeros ([m n])
pbest=0;
L=zeros([1 m]); %Distância total da viagem S de cada formiga k%
zmax=zeros([1 m])
S=zeros([1 m+1]); %Viagem ótima%
Q=1;

%Gerando valores heurísticos Eta (7 X 7)

Eta=[0 0 2 3 2.2 0 0;0 0 0 0 2.5 3.6 0;0 0 0 0 0 0 1;0 0 0 0 0 0 1;0 0 0 0
0 0 1;0 0 0 0 0 0 1;0 0 0 0 0 0 0]

%Inicializando matriz de feromônios
for i= 1:1:n
    for j=1:1:n
        if i==j
            tau(i,j)=0;
        else
            tau(i,j)= 1;
        end
    end
end
tau

iter=1;%inicializando contador de iterações%

while iter <= max_iter

%Inicializando uma formiga em cada satélite (os primeiros m nós representam
%os satélites)
for k=1:1:m
    for j=1:1:trans
        if j==1
            formigas(k,j)=k;
        else
            formigas(k,j)=0;
        end
    end
end
end

```

formigas

```

for a=1:1:trans %transições de cidades para cada formiga
a;
% soma das possibilibilades de transições factíveis das formigas
for k=1:1:m %formigas

    for i= 1:1:n %possibilidade de transição para cidade i
        aux=0;
        for j=1:1:a
            if formigas(k,j)==i
                aux=1;
            end
        end
        if aux==0

soma(k)=soma(k)+(tau(formigas(k,a),i)^alpha*Eta(formigas(k,a),i)^Beta);
            end

        end
end
%soma;

%cálculo das probabilidades de transição de cada formiga
for k=1:1:m %formigas

    for i= 1:1:n %cidade
        aux=0;
        for j=1:1:a
            if formigas(k,j)==i
                aux=1;
            end
        end
        if aux==0
            p(k,i)=
((tau(formigas(k,a),i)^alpha*Eta(formigas(k,a),i)^Beta))/soma(k);%cálculo
da probabilidade de transição de cidade i para cidade j%
            end

            if p(k,i)> pbest
                formigas(k,a+1)=i;
                pbest=p(k,i);
            end
            if a==n % a última coluna da matriz formigas é sempre igual a primeira
coluna (formiga sempre retorna a cidade de partida%
                formigas(k,a+1)= formigas(k);
            end

        end
        pbest=0;
end
p;
iter;
formigas;
soma= zeros ([1 m]);
p= zeros ([m n]);

```

```

end
formigas
fprintf ('Ciclo %d', iter);

%Calculando resultado da função objetivo para cada solução gerada%
for k=1:1:m
    z(k)=0;

    x=formigas(k,1);
    y=formigas(k,2);
    z(k)=Eta(x,y);

    if z(k)>= zmax(k) %botei maior ou igual para alterar o valor da solução
ótima
        zmax(k)=z(k);
    end
end

end

%mostra qual melhor solução até o momento
zmax

%inicializando matriz de acréscimo de feromônios%
delta_tau=zeros([n n]);

%Cálculo da acréscimo de feromônio para cada caminho percorrido pelas
formigas
for k=1:1:m %formigas
    for a=1:1:trans-1 %auxiliar para varrer matriz das formigas
        for i= 1:1:n %varrer matriz de feromônios%
            for j= 1:1:n
                if (formigas(k,a)==i && formigas(k,a+1)==j)
                    delta_tau(i,j)= delta_tau(i,j) + Q*z(k);
                    delta_tau(j,i)= delta_tau(i,j); %o acréscimo de
feromônio é feito em tau(i,j) e tau (j,i) simultaneamente, gerando uma
matriz simétrica%
                end
            end
        end
    end
end

delta_tau

%Atualização da matriz de feromônio: acréscimo e evaporação%
for i=1:1:n
    for j=1:1:n
        tau(i,j)=(1-rho)*tau(i,j)+ delta_tau(i,j);
    end
end
tau
Eta
iter=iter+1;
end

```