



Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Ismael Érique Koch

O Independent Vortices Search no controle de qualidade em Redes Geodésicas:
Uma avaliação de Estimadores Robustos otimizados por Meta-Heurística para o ajustamento de Redes Geodésicas

São Leopoldo, 2019

Ismael Érique Koch

**O *INDEPENDENT VORTICES SEARCH* NO CONTROLE DE QUALIDADE EM
REDES GEODÉSICAS:**

**Uma avaliação de Estimadores Robustos otimizados por meta-heurística para o
ajustamento de redes geodésicas**

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos — UNISINOS

Orientador:

Prof. Dr. Luiz Gonzaga Silveira Jr

Coorientador:

Prof. Dr. Maurício Roberto Veronez

São Leopoldo
2018

Ismael Érique Koch

Título: O INDEPENDENT VORTICES SEARCH NO CONTROLE DE QUALIDADE EM REDES
GEODÉSICAS:
UMA AVALIAÇÃO DE ESTIMADORES ROBUSTOS OTIMIZADOS POR META-HEURÍSTICA PARA
O AJUSTAMENTO DE REDES GEODÉSICAS

Dissertação apresentada à Universidade do Vale do
Rio dos Sinos – Unisinos, como requisito parcial para
obtenção do título de Mestre em Computação
Aplicada.

Aprovado em 27 de fevereiro de 2019

BANCA EXAMINADORA

Prof. Sandro José Rigo – UNISINOS

Prof. Marcelo Carvalho dos Santos – UNB/Canadá

Prof. Felipe Geremia Nievinski – UFRGS

Prof. Dr. Luiz Gonzaga da Silveira Jr. (Orientador)

Visto e permitida a impressão
São Leopoldo,

Prof. Dr. Rodrigo da Rosa Righi
Coordenador PPG em Computação Aplicada

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior --- Brasil (CAPES) --- Código de Financiamento 001.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior --- Brasil (CAPES) --- Finance Code 001

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Computação Aplicada (PPGCA) da Universidade do Vale do Rio dos Sinos (Unisinos) pela oportunidade de realizar este mestrado e poder contribuir com a ciência.

Ao programa de Suporte à Pós-Graduação de Instituições Comunitárias de Educação (Capes/PROSUC) pela Bolsa Auxílio, sem a qual esta pesquisa não seria possível.

Aos professores, orientador e coorientador, Dr. Luiz Gonzaga Silveira Jr e Dr. Maurício Roberto Veronez, pela confiança depositada em mim e pelo tempo dedicado a esta dissertação com discussões e direcionamentos que agregaram muito ao trabalho.

Ao professor Dr. Ivandro Klein, do Departamento de Construção Civil do Instituto Federal de Santa Catarina (IFSC), pelas inúmeras e valiosas contribuições ao trabalho, sempre disposto e acessível.

Aos professores do PPGCA, Dr. Sandro José Rigo e Dr. Rodrigo da Rosa Righi, pelo ótimo acolhimento no ingresso ao PPGCA. Demais professores pelas aulas ministradas, em especial ao Prof. Dr. Gustavo Pessin, pelas discussões acerca das meta-heurísticas.

Aos colegas Eduardo Souza dos Reis, Felipe de Moraes, Gustavo Luis Schaab, Juarez Machado da Silva e Robson Pandolfi (*in memoriam*) pelo ótimo convívio, coleguismo e amizade desenvolvida durante o mestrado.

Aos membros da banca examinadora, pela disponibilidade e contribuições pertinentes ao trabalho.

A minha noiva, Maiara, pelo apoio e compreensão nas horas em que estive menos presente.

A Deus, e também a família e amigos que sempre estiveram ao meu lado mesmo quando estive ausente pela dedicação a esta pesquisa.

RESUMO

As redes geodésicas são a base para atividades de mapeamento, geoinformação, registro de terras e outros serviços baseados em localização, além de também apresentarem um papel importante na sociedade em obras de infraestrutura (estradas, pontes, túneis, abastecimento de água, esgoto, redes de eletricidade, telecomunicações, etc.) que são diretamente dependentes de pontos de controle tridimensionais altamente precisos. Dada a importância destes pontos de referência, é necessário um cálculo e ajustamento preciso no estabelecimento destas redes. A maioria delas é estabelecida usando os sinais GNSS (*Global Navigation Satellite System* - Sistema Global de Navegação por Satélite) que estão disponíveis em toda superfície terrestre. As redes geodésicas baseadas em técnicas GNSS têm as coordenadas de seus pontos estimadas a partir das distâncias relativas entre elas, chamadas observações, através de processos de ajustamento. Um ajustamento de rede leva em consideração todas as observações que formam a rede de modo a reduzir os resíduos resultantes do processo. Para tal, o Método dos Mínimos Quadrados (MMQ) é frequentemente aplicado por se tratar do melhor estimador linear imparcial, assumindo que não existam erros grosseiros (*outliers*) e/ou erros sistemáticos. No entanto, estes erros podem ocorrer na prática e fazer com que essa estimativa falhe, levando a erros sem precedentes em muitos pontos da rede. Como alternativa, Estimadores Robustos (ERs) podem ser aplicados, pois tendem a ser insensíveis a presença de *outliers*. Tais ERs podem ser implementados através de algoritmos meta-heurísticos (MHs) para buscar resolver a função do estimador, muitas vezes complexa e não solucionável por equações. Esta abordagem já foi trazida por outros trabalhos, porém, pouco explorada e em cenários muito específicos e limitados. Este trabalho traz uma pesquisa mais ampla desta estratégia, analisando o comportamento de diversos ERs conhecidos na literatura, através da aplicação de uma meta-heurística nova denominada *Independent Vortices Search* (Pesquisa de Vórtices Independentes), além de uma definição inédita para o espaço de busca. Para isso, gerou-se um grande número de cenários de erros para que os ERs fossem testados, junto a um método clássico iterativo e robusto, que não necessita de MH. Além disso, também testou-se a posição da mediana e de corte de alguns estimadores para verificar o impacto desse ajuste nos resultados. Os resultados apontam que a nova MH — IVS — é superior ao algoritmo original e todas as outras propostas de modificação. No ajustamento de redes, os ERs testados apresentam resultados melhores que o MMQ na presença de *outliers* de grande magnitude, porém, demonstram uma solução inferior nos cenários com nenhum *outlier* ou de pequena magnitude. Já para a detecção de *outliers*, o MMQ tem grandes dificuldades em cenários com erros grosseiros de pequena magnitude, enquanto que para situações com *outliers* de grande magnitude, tanto o LS quanto os demais ERs apresentam bons resultados.

Palavras-chave: Redes geodésicas. Controle de qualidade. Estimação robusta. Meta-heurística. Otimização numérica.

ABSTRACT

Geodetic networks are the basis not only for mapping activities, geoinformation, land registry and other location-based services, but also provide an important role in society as infrastructure works (roads, bridges, tunnels, water supply, sewage, electricity networks, telecommunications, etc.) which are directly dependent on highly accurate three-dimensional control points. Given the importance of this reference points, a precise computation and adjustment is highly necessary in the establishment of this networks. Most of them are settled using GNSS (Global Navigation Satellite System) signals which are available all over the Earth surface. The geodetic networks based on GNSS techniques have their points' coordinates estimated from the relative distances between them, called observations, through adjustment processes. When it comes to adjust all the observations to form a network, the least squares (LS) method is often applied because it is the best linear unbiased estimator, assuming that no outliers and/or systematic errors exist. Outliers may occur in practice, however, and cause such estimation to fail and leading to unprecedented errors over many points in the network. Alternatively, Robust Estimators (ERs) can be applied as to be insensitive to the presence of outliers. Such ERs can be implemented through metaheuristic algorithms (MHs) to solve the estimator's function, often complex and not solvable by equations. This approach has already been brought by other studies, however, they were not widely explored, mostly in very specific and limited scenarios. This work brings a new research of this strategy, analyzing the behavior of several ERs known in the literature, through the application of a new meta-heuristic called Independent Vortices Search, besides a new strategy for the search space definition. A large number of error scenarios were generated to test the ERs, along with a classic iterative and robust method, which does not require any MH algorithm. In addition, we also tested the median and cut position of some estimators to verify the impact of this adjustment on the solutions. Results indicate that the new MH — IVS — is superior to the original algorithm and all other modification proposals. In network adjustments, the ERs tested present better results than the LS method in the presence of large-scale outliers, however, they demonstrate a lower solution in scenarios with no outlier or small magnitude gross errors. For the detection of outliers, the LS method has great difficulties in scenarios with small errors, whereas for situations with large outliers, both LS and other ERs present good results.

Keywords: Geodetic networks. Quality control. Robust Estimation. Metaheuristics. Numeric Optimization.

LISTA DE FIGURAS

Figura 1 – Exemplo de vértices e vetores em uma rede geodésica	32
Figura 2 – Cenário hipotético de busca realizada pelo VS. São apresentados três momentos na convergência do raio do vórtex durante a execução. Ao final, vê-se a fragilidade do VS em ficar preso em um mínimo local. Os pontos indicam candidatos a soluções gerados pelo VS.	57
Figura 3 – Três situações hipotéticas em que o centro de um vórtex é direcionado para a melhor solução atual	58
Figura 4 – Cenário hipotético de busca realizada pelo IVS	59
Figura 5 – Valores encontrados no Cenário de Erros 01 para cada MH	63
Figura 6 – Valores encontrados no Cenário de Erros 02 para cada MH	64
Figura 7 – Valores encontrados no Cenário de Erros 03 para cada MH	64
Figura 8 – Valores encontrados no Cenário de Erros 04 para cada MH	65
Figura 9 – Rede com seis vértices da RBMC: MGIN, CHPI, UBA1, POLI, SPCA e SJSP. Um vértice de controle e treze vetores de observação tridimensionais.	68
Figura 10 – Fluxograma de execução do sistema de ajustamento, uma vez tendo definidas todas as informações da rede geodésica e configurações de execução.	71
Figura 11 – Ocorrência de cada classificação nos 900 cenários para os três estimadores e sua variação na posição de corte ou mediana.	73
Figura 12 – Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, comparando os ERs para os valores h_{n2} e h_{np2}	74
Figura 13 – Quantidade de <i>outliers</i> identificados, falsos e não detectados comparando os ERs para os valores h_{n2} e h_{np2}	75
Figura 14 – Diferença percentual para cada estimador divididas em erros identificados, falsos e não detectados, para os limites em h_{n2} e h_{np2}	76
Figura 15 – Diferença do afastamento médio das coordenadas, entre os limites h_{n2} e h_{np2} , em pacotes de cenários de 0 a 4 <i>outliers</i>	77
Figura 16 – Contagem das classificações “A” e “B” obtidas nos 100 cenários sem <i>outliers</i> para todos os estimadores.	78
Figura 17 – Quantidade de falsos positivos para os cenários sem <i>outliers</i> do pacote 01, agrupados por estimador.	78
Figura 18 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários sem <i>outliers</i>	79
Figura 19 – Contagem das classificações de “A” até “F” obtidas nos 400 cenários contendo de 1 a 4 <i>outliers</i> de menor magnitude, organizados por estimador.	80
Figura 20 – Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, para cada estimador, nos cenários com <i>outliers</i> de menor magnitude.	81
Figura 21 – Quantidade de <i>outliers</i> identificados, falsos e não constatados para cada estimador, nos 400 cenários com erros grosseiros de menor magnitude.	81

Figura 22 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 01 <i>outlier</i> de pequena magnitude.	83
Figura 23 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 02 <i>outliers</i> de pequena magnitude.	83
Figura 24 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 03 <i>outliers</i> de pequena magnitude.	84
Figura 25 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 04 <i>outliers</i> de pequena magnitude.	84
Figura 26 – Contagem das classificações de “A” até “F” obtidas nos 400 cenários contendo de 1 a 4 <i>outliers</i> de maior magnitude, organizados por estimador. . . .	86
Figura 27 – Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, para cada estimador, nos cenários com <i>outliers</i> de grande magnitude.	86
Figura 28 – Quantidade de <i>outliers</i> identificados, falsos e não constatados para cada estimador, nos 400 cenários com erros grosseiros de maior magnitude.	87
Figura 29 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 01 <i>outlier</i> de grande magnitude.	89
Figura 30 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 02 <i>outliers</i> de grande magnitude.	90
Figura 31 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 03 <i>outliers</i> de grande magnitude.	91
Figura 32 – Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 04 <i>outliers</i> de grande magnitude.	92

LISTA DE TABELAS

Tabela 1 – Principais características das meta-heurísticas apresentadas	61
Tabela 2 – Afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos de pequena magnitude.	82
Tabela 3 – Afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos de grande magnitude.	88

LISTA DE ABREVIATURAS

Eq., Eqs.	Equação, Equações
Fig., Figs.	Figura, -s
Tab., Tabs.	Tabela, -s

LISTA DE SIGLAS

ABC	<i>Artificial Bee Colony</i>
AG	Algoritmo genético
CR	Controle de Redundância
DNM	Distribuição Normal Multivariada
ER	Estimador Robusto
FE	<i>Fitness Evaluations</i>
GPS	<i>Global Positioning System</i>
GNSS	<i>Global Navigation Satellite Systems</i>
HVS	<i>Hybrid Vortex Search</i>
IRLS	<i>Iteratively Reweighted Least Squares</i>
IVS	<i>Independent Vortices Search</i>
LMS	<i>Least Median of Squares</i>
LS	<i>Least Squares method</i>
LTA	<i>Least Trimmed Absolute deviations</i>
LTS	<i>Least Trimmed Squares</i>
LTS-RC	<i>Least Trimmed Squares with Redundancy Constraint</i>
MH	Meta-heurística
MMQ	Método dos Mínimos Quadrados
ModVS	<i>Modified Vortex Search</i>
RBMC	Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS
SRLS	<i>Sign-constrained Robust Least Squares</i>
VS	<i>Vortex Search</i>

LISTA DE SÍMBOLOS

σ^2	Fator de variância <i>a priori</i> das observações
Σ	Matriz de covariâncias das observações
$\Sigma_{\hat{\theta}}$	Matriz de covariância dos resíduos ajustados
μ_{0j}	Solução e centro iniciais para o parâmetro j no <i>Vortex Search</i>
μ_l	Posição do centro l no <i>Modified Vortex Search</i>
$\boldsymbol{\mu}$	Vetor de centros no <i>Vortex Search</i>
γ	Representação da função gama inversa incompleta
τ_i	Resultado do teste estatístico para a posição i no IRLS
a_c	Parâmetro que define a forma (<i>shape</i>) a cada ciclo no <i>Vortex Search</i>
\mathbf{A}	Matriz das derivadas parciais (<i>design</i> ou Jacobiana) no modelo Gauss-Markov
c	Contagem de ciclos (iterações) no <i>Vortex Search</i> e na <i>Artificial Bee Colony</i>
\mathbf{c}_i	Vetor nulo com valor de 1 na posição i
C	Constante positiva definida entre o intervalo de $[1, 0; 2, 0]$ no SRLS e no IRLS
C_{max}	Constante que define o número máximo de ciclos (iterações) no <i>Vortex Search</i> e na <i>Artificial Bee Colony</i>
fit_i	Valor de aptidão (<i>fitness</i>) do i -ésimo candidato no <i>Vortex Search</i> e na <i>Artificial Bee Colony</i>
f_i	Resultado da função objetivo para o i -ésimo candidato no <i>Vortex Search</i>
h_{n2}	Limite de corte para o LTS e LTS-RC e posição da mediana para o LMS, em $(n + 1) / 2$
h_{np2}	Limite de corte para o LTS e LTS-RC e posição da mediana para o LMS, em $(n + p + 1) / 2$
\mathbf{I}	Matriz identidade
l	l -ésimo centro no <i>Modified Vortex Search</i>
\mathbf{l}	Vetor das observações do modelo linearizado de Gauss-Markov
L	Constante que define o valor de limite para abandono de uma fonte de alimento (solução) na <i>Artificial Bee Colony</i>
n	Dimensão do vetor de observações e do vetor de resíduos no modelo Gauss-Markov; quantidade de observações
p	Dimensão do vetor de parâmetros no modelo Gauss-Markov; quantidade de coordenadas a estimar
p_i	Valor de probabilidade de escolha da i -ésima solução na <i>Artificial Bee Colony</i>
r_{0j}	Raio inicial do j -ésimo parâmetro no <i>Vortex Search</i>
r_c	Tamanho do raio para geração de novas soluções candidatas para o ciclo c no <i>Vortex Search</i>

R	Constante fixada em 0.1 que controla a resolução da busca ao atingir aproximadamente metade do número de iterações no <i>Vortex Search</i>
s'	Candidato a melhor solução no <i>Vortex Search</i>
s_{best}	Melhor solução no <i>Vortex Search</i>
s_l	Melhor solução do centro l no <i>Modified Vortex Search</i>
s	Vetor de soluções candidatas no <i>Vortex Search</i>
SN	Constante que define o número de soluções na <i>Artificial Bee Colony</i>
T	Transposição de matriz ou vetor
v	Vetor de resíduos no modelo Gauss-Markov
W	Matriz peso das observações no modelo Gauss-Markov
$x_{min,j}, x_{max,j}$	Respectivamente, limite mínimo e máximo do j -ésimo parâmetro no <i>Vortex Search</i> e na <i>Artificial Bee Colony</i>
x	Vetor dos parâmetros no modelo Gauss-Markov
\hat{x}	Vetor dos parâmetros ajustados
x_0	Vetor dos parâmetros aproximados para as equações de observação
$\Delta X, \Delta Y, \Delta Z$	Componentes (observações) de uma linha-base GNSS entre dois vértices
X, Y, Z	Coordenadas de um vértice

SUMÁRIO

1 INTRODUÇÃO	25
1.1 Objetivos	28
1.1.1 Objetivo geral	28
1.1.2 Objetivos específicos	28
1.2 Hipóteses	28
1.3 Estrutura do trabalho	29
2 REVISÃO TEÓRICA	31
2.1 Redes geodésicas	31
2.2 Erros em observações	32
2.2.1 Erros grosseiros em redes geodésicas	32
2.3 Ajustamento da rede	33
2.3.1 Método dos mínimos quadrados	33
2.4 Estimadores robustos	35
2.4.1 Mínimos Quadrados Aparados (<i>Least Trimmed Squares</i>)	36
2.4.2 Mínima Mediana dos Quadrados (<i>Least Median of Squares</i>)	36
2.4.3 Mínimos Quadrados Aparados com Restrição de Redundância (<i>Least Trimmed Squares with Redundancy Constraint</i>)	37
2.4.4 Método dos Mínimos Quadrados Robusto com Sinais Restringidos (<i>Sign-Constrained Robust Least Squares</i>)	38
2.4.5 Mínimos Quadrados Reponderados Iterativamente (<i>Iteratively Reweighted Least Squares</i>)	40
2.4.6 Mínimos Desvios Absolutos Aparados (<i>Least Trimmed Absolute deviations</i>)	41
2.4.7 A posição da mediana no LMS e o limite de corte no LTS e LTS-RC	41
2.4.8 Aplicação dos estimadores robustos	42
2.5 Algoritmos meta-heurísticos	42
2.5.1 <i>Vortex Search</i> e <i>Modified Vortex Search</i>	44
2.5.2 <i>Artificial Bee Colony</i>	46
2.5.3 <i>Hybrid Vortex Search algorithm</i>	50
2.6 Trabalhos Relacionados	51
3 O INDEPENDENT VORTICES SEARCH	55
3.1 Análise das características e limitações do VS e ModVS	55
3.2 Desenvolvimento do IVS	59
3.3 Experimentos e resultados do IVS	60
4 CONTROLE DE QUALIDADE	67
4.1 Análise da qualidade da solução	67
4.2 A rede geodésica e a simulação dos cenários de erros	68
4.3 Desenvolvimento e implementação	70
5 RESULTADOS E DISCUSSÕES	73
5.1 Comparação do limite de corte (LTS e LTS-RC) e posição da mediana (LMS)	73
5.2 Resultados dos ajustamentos com ERs	77
5.2.1 O caso de nenhum <i>outlier</i>	77
5.2.2 Os casos com <i>outliers</i> de pequena magnitude	80
5.2.3 Os casos com <i>outliers</i> de grande magnitude	85

6	CONSIDERAÇÕES FINAIS	93
6.1	Sobre o IVS	93
6.2	Sobre o controle de qualidade com ERs	93
	REFERÊNCIAS	95

1 INTRODUÇÃO

A partir do início do século XXI, o uso de tecnologias baseadas em GNSS (*Global Navigation Satellite Systems* - Sistemas de Navegação Global por Satélite) cresceu de forma significativa. Tanto profissionais da área da geodesia quanto usuários de *smartphones* tem utilizado de forma crescente os produtos desenvolvidos a partir desta tecnologia. Popularmente conhecidos por GPS — sigla que, na verdade, se refere a apenas uma das constelações de satélites disponíveis hoje (*Global Positioning System* - Sistema de Posicionamento Global) mantida pelo governo dos Estados Unidos da América — estes receptores encontram-se atualmente nos mais diversos dispositivos, desde instrumentos de medição precisa, até celulares e carros. E, graças a introdução de novas constelações e novos tipos de sinais, é possível obter soluções mesmo em áreas menos favoráveis com pouca cobertura de sinais.

Além dos avanços e facilidades que a tecnologia GNSS trouxe no dia-a-dia, ela também facilitou a implementação e o controle de qualidade de redes geodésicas de alta precisão e exatidão. Antes desta tecnologia, o estabelecimento de uma rede geodésica necessitava de visada direta entre os vértices, o que poderia dificultar o planejamento e limitar a geometria da rede, ou então, fazia-se necessário usar de métodos de poligonação, que poderiam denegrir a qualidade da rede. Agora, com as coordenadas dos vértices podendo ser definidas a partir de sinais vindos do espaço, faz-se necessário somente uma boa cobertura de satélites no momento do rastreamento dos sinais. Isto também facilitou a utilização da tecnologia na densificação de redes geodésicas clássicas de primeira ordem, devido ao seu alto grau de flexibilidade no planejamento da rede e a acurácia alcançável (LEICK; RAPOPORT; TATARNIKOV, 2015).

As redes geodésicas são definidas por diversos vértices materializados, em locais de interesse, e desempenham um papel importante na sociedade. Além de servirem de apoio a atividades de mapeamento, geoinformação, registro de terras e outros serviços baseados em localização, também são a base de obras de infraestrutura como estradas, pontes, túneis, abastecimento de água, esgoto, redes de eletricidade, telecomunicações, etc., que dependem diretamente de pontos de controle tridimensionais altamente precisos.

A concepção de uma rede geodésica a partir dos sistemas GNSS, se dá pela observação dos vértices empregando no mínimo dois dispositivos receptores dos sinais. Os receptores devem trabalhar simultaneamente em cada par de pontos observados, registrando os sinais dos satélites por um tempo pré-estabelecido, até que todos os vértices tenham sido observados. O valor das coordenadas de cada vértice é obtido calculando os vetores de observação separadamente entre os pares observados e, em seguida, ajustando a rede com base nas componentes dos vetores calculados (GHILANI, 2010). Ainda, para o ajustamento da rede, é necessário que pelo menos um dos vértices sirva de referência e tenha coordenadas conhecidas no referencial (*datum*) adotado, servindo como ponto de apoio (ou vértice de controle), além da inexistência de erros sistemáticos e grosseiros (*outliers*) nos vetores de observação processados.

O chamado ajustamento das redes geodésicas — cálculo final das coordenadas dos vértices

— geralmente é obtido aplicando-se o Método dos Mínimos Quadrados (MMQ) (*Least Squares method — LS*), altamente difundido, e que assume a condição de que apenas erros aleatórios estejam presentes nas observações. Nessa situação, o MMQ é o melhor estimador linear imparcial para os parâmetros que, neste caso, são as coordenadas dos vértices a serem determinadas (KOCH, 1999; TEUNISSEN, 2000; GHILANI, 2010). Porém, o fato do MMQ não ser um estimador robusto, torna-o sensível¹ a presença de erros grosseiros nas observações, fazendo com que distorça a estimativa dos parâmetros, ou seja, das coordenadas da rede (HUBER; RONCHETTI, 2009).

Os erros aleatórios estão diretamente relacionados com o instrumento ou técnica utilizada na observação, não podendo ser evitados. Já os erros grosseiros, consistem em falhas inesperadas e desconhecidas nas observações, podendo ser de causa humana ou instrumental e de difícil detecção (KLEIN, 2011). Em redes geodésicas levantadas a partir das técnicas GNSS, apesar de grande parte do processo de observação não ter influência humana direta, os vetores não estão livres de falhas. Como causas destes tipos de desvios temos, por exemplo, o erro de multicaminhamento de sinal (reflexão do sinal), em que o sinal eletromagnético — que deveria se propagar diretamente ao longo da linha de visão entre o satélite e o receptor — é refletido ou dispersado por algum objeto antes de alcançar o instrumento (HOFMANN-WELLENHOF; LICHTENEGGER; WASLE, 2008). Também, podem haver erros grosseiros causados por falha humana como, por exemplo, a medição incorreta da altura da antena do receptor, neste caso acarretando na informação incorreta ao software de processamento. Embora os erros aleatórios sejam inerentes às observações, os erros grosseiros devem ser evitados e, sempre que possível, identificados e eliminados para a correta determinação das coordenadas na rede.

Tendo em vista a difícil detecção da ocorrência de um *outlier* e a limitação do MMQ frente a presença dos mesmos, a utilização de um estimador robusto (ER) é apresentada como uma opção para o cálculo dos parâmetros. Segundo Huber e Ronchetti (2009), o termo “robusto”, neste caso, significa uma insensibilidade a pequenos afastamentos das suposições feitas ao problema. Ou seja, caso algumas observações contenham erros grosseiros ou mesmo erros sistemáticos (assumindo de início que só haveriam erros aleatórios), os ERs serão insensíveis a estas distorções quando da estimativa dos parâmetros.

Porém, muitos destes ERs são complexos e não solucionáveis através de equações, necessitando de processos iterativos ou técnicas inteligentes para otimização da função do estimador. Uma das abordagens é a utilização de um algoritmo meta-heurístico, ou meta-heurística (MH), que possa buscar a otimização numérica de uma função objetivo.

As meta-heurísticas (MHs), são algoritmos iterativos que combinam de forma inteligente diferentes estratégias para exploração e investigação de soluções dentro de um espaço de busca, aliando estratégias de aprendizagem para estruturar a informação, com a finalidade de obter uma solução quase ótima ao problema de forma eficiente (OSMAN; KELLY, 1997). Na prática,

¹Sensível significa aquilo que recebe facilmente as impressões ou sensações externas, frágil, perturbado com facilidade. No nosso contexto, a estimação dos parâmetros pelo MMQ é sensível a erros ou desvios presentes nas observações ou no modelo.

no ajustamento de redes geodésicas, as MHs fazem a estimação dos parâmetros (coordenadas dos vértices), dentro do espaço de busca pré-estabelecido, e através da função objetivo do ER, calculam o valor de aptidão (*fitness*) da solução. Este processo se repete seguindo a estratégia de busca adotada pela MH até que uma solução aceitável seja encontrada ou um limite de tempo, ou computacional, seja atingido (KOCH et al., 2017).

A implementação de MHs no ajustamento de redes geodésicas não é novidade, porém, ainda recente e pouco explorada. Os trabalhos existentes (ver seção 2.6) limitam-se a trabalhar com apenas um ou dois ERs e muitas vezes em casos simples de ajustamento, sem geração de vários cenários de erros, ou sem erros aleatórios, apenas *outliers*. Pelo fato dos erros aleatórios fazerem parte de qualquer observação, é de suma importância que explore-se soluções de ajustamento onde eles possam ser simulados, aproximando-se ao máximo de uma situação real. Também é necessário que vários cenários de erros sejam testados com os mais variados ERs para tornar possível a obtenção de uma visão geral e mais factível da aplicabilidade da solução neste tipo de problema.

Desta forma, neste trabalho, avalia-se o comportamento de diversos ERs conhecidos na literatura para o ajustamento de uma rede geodésica, cujas observações foram geradas a partir das coordenadas oficiais dos vértices, e os desvios-padrões obtidos com o processamento dos sinais dos sistemas GNSS. Para isso, foram simulados diversos cenários de erros nas observações da rede, e os resultados dos ERs comparados com o MMQ clássico. Também foram testadas diferentes posições para a mediana e para o corte de resíduos nos ERs que aplicam tal estratégia. Assim, foi possível verificar o impacto destes ERs e dos ajustes nos resultados, além de verificar a potencialidade da estratégia de utilizar ERs e MH no ajustamento da rede. Para a análise dos resultados, levou-se em consideração a identificação ou não dos erros grosseiros (*outliers*), a classificação da solução, além da comparação numérica dos parâmetros estimados com as coordenadas oficiais da rede.

A rede geodésica utilizada para a realização dos experimentos é composta por seis (6) vértices da Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS (RBMC), formando uma rede de menor dimensão. Destes seis (6) vértices, um é de controle, utilizando treze (13) vetores de observação entre os pontos. Para o modelo, totalizam-se 15 parâmetros e 39 observações, tendo em vista que os pontos e vetores são tridimensionais. Foram estabelecidos nove (9) pacotes de cem cenários cada, contendo erros grosseiros em algumas observações e erros aleatórios em todas as observações. Todos os erros foram calculados a partir do desvio-padrão de cada observação, sendo que os *outliers* variam de magnitude entre $\pm [3\sigma; 6\sigma]$, configurando erros pequenos, e entre $\pm [6\sigma; 12\sigma]$, configurando erros grandes, distribuídos aleatoriamente. Cada pacote teve um número fixo de *outliers* por cenário, variando de nenhum até quatro erros grosseiros simultâneos. Os erros aleatórios também foram únicos em cada cenário, variando entre $\pm [0\sigma; 3\sigma]^2$

²O truncamento dos erros aleatórios em 3σ não deve afetar os resultados. Seguindo a distribuição normal, a ocorrência de valores maiores que 3σ é de apenas 0,3%.

Além disto, este estudo apresenta uma nova MH, denominada de *Independent Vortices Search* (Pesquisa de Vórtices Independentes — IVS), desenvolvida durante o trabalho e que foi aplicada na minimização das funções dos ERs testados. O IVS é baseado na modificação do algoritmo conhecido por *Vortex Search* (Pesquisa de vórtice — VS) (DOGAN; ÖLMEZ, 2015a), e traz alterações que objetivam minimizar algumas limitações encontradas no algoritmo original e já identificadas em Dogan (2016).

Em suma, a ideia deste trabalho se divide em duas: primeiro, contribuir no desenvolvimento de MHs melhores com a intridução do IVS. Segundo, colaborar no aprimoramento do controle de qualidade em redes geodésicas, aliando métodos inteligentes e robustos, de modo a permitir uma comparação com soluções já existentes.

1.1 Objetivos

1.1.1 Objetivo geral

Verificar a potencialidade do uso de Estimadores Robustos (ERs) otimizados pelo *Independent Vortices Search* (Pesquisa de Vórtices Independentes — IVS), para o controle de qualidade de redes geodésicas.

1.1.2 Objetivos específicos

- Apresentar o *Independent Vortices Search* como um novo algoritmo meta-heurístico, baseado no *Vortex Search*.
- Verificar a possibilidade de identificação de erros grosseiros (*outliers*) a partir do vetor de resíduos, usando ERs;
- Comparar a qualidade do ajustamento com o método clássico dos Mínimos Quadrados (MMQ);

1.2 Hipóteses

- I. O *Independent Vortices Search* é mais eficiente que o algoritmo original e que a alteração já proposta em Dogan (2016).
- II. A maioria dos ERs apresentam resultados melhores que o MMQ na maior parte dos cenários de erros testados;
- III. É possível detectar a maior parte dos *outliers* inseridos nas observações a partir dos vetores de resíduos gerados pelos ERs;

1.3 Estrutura do trabalho

Este trabalho é dividido em seis capítulos. O primeiro capítulo traz a introdução a pesquisa, justificando o trabalho, além de apresentar os objetivos.

O capítulo 2, traz uma revisão teórica, discutindo brevemente, as redes geodésicas, os erros envolvidos e o ajustamento de rede. Em seguida, apresenta os Estimadores Robustos (ERs) que foram aplicados nos experimentos, além de introduzir as meta-heurísticas, junto com os algoritmos estudados. Ao final, ainda apresenta uma relação de trabalhos relacionados, que serviram de base para esta pesquisa.

O capítulo 3, apresenta o *Independent Vortices Search*, trazendo uma análise das características e limitações dos algoritmos originais e já conhecidos, além do desenvolvimento e resultados obtidos com o nova MH. Em seguida, o capítulo 4, demonstra de que maneira foi procedida a análise da qualidade das soluções obtidas pelos experimentos, também traz mais detalhes da rede geodésica utilizada e da simulação dos cenários de erros e, por último, o desenvolvimento e implementação da solução.

O capítulo 5, apresenta a segunda contribuição deste trabalho, expondo os resultados obtidos nos experimentos de ajustamentos da rede GNSS. Inicia-se com a comparação do limite de corte e posição da mediana de alguns ERs e, em seguida, os resultados e discussões dos diversos cenários de erros. Por último, o capítulo 6 traz as considerações finais sobre o trabalho, além de apresentar questões para serem exploradas em pesquisas futuras.

2 REVISÃO TEÓRICA

Este capítulo traz uma revisão teórica dos assuntos que serão discutidos na dissertação. Primeiro, são abordadas as redes geodésicas seguidas pelos erros que as observações delas podem apresentar. Depois, uma breve explicação da teoria do ajustamento de redes, a apresentação dos estimadores robustos utilizados na pesquisa, o algoritmo *Vortex Search* e, por último, os trabalhos científicos relacionados a este trabalho.

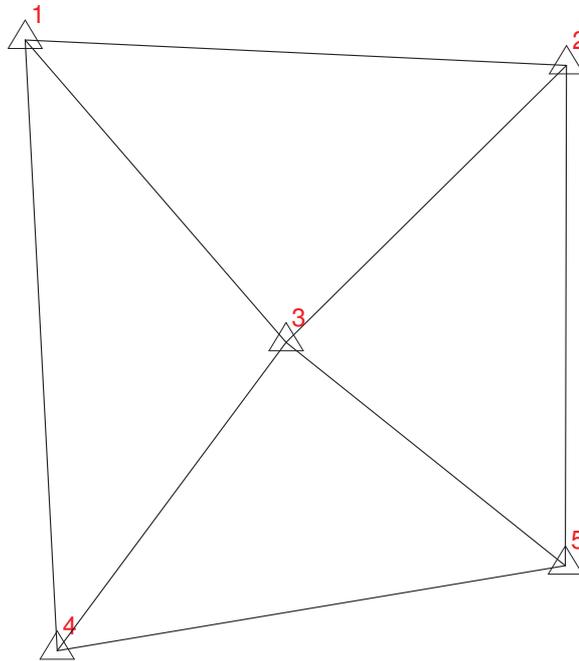
2.1 Redes geodésicas

Constituída por um conjunto de pontos materializados na Terra, uma rede geodésica tem por finalidade fornecer uma cadeia de vértices com coordenadas de alta precisão e confiabilidade. Estas redes objetivam servir de amarração (ou referência) para um posicionamento preciso para curtas e longas distâncias na superfície terrestre. As aplicações na ciência e engenharia são diversas, entre elas: realização de sistemas de referência; apoio em projetos de levantamento e mapeamento; gerenciamento territorial; monitoramento de deformações estruturais; locação de obras de engenharia; monitoramento de fenômenos dinâmicos na superfície da Terra; implementação e manutenção de vários serviços de infra-estrutura, entre outros. Todas estas aplicações dependem diretamente de pontos de controle com coordenadas tridimensionais altamente precisas e confiáveis.

A confiabilidade de uma rede está relacionada com a identificação de erros grosseiros e sistemáticos, enquanto a precisão está relacionada com a magnitude dos erros aleatórios (KLEIN et al., 2012). Os erros aleatórios estão relacionados aos equipamentos e técnicas de levantamento, enquanto os erros grosseiros devem ser totalmente evitados e podem ter diversas causas. Mesmo com a coleta de dados sendo uma rotina automatizada no levantamento das redes que fazem uso dos sistemas GNSS, ela não está livre de erros. São conhecidos erros de multicaminhamento de sinal, erros de “sem linha de visão” (*non-line-of-sight*), erro do relógio do satélite, erro do relógio do receptor, erros devido a ionosfera, erros de operação do equipamento, entre outros (HOFMANN-WELLENHOF; LICHTENEGGER; WASLE, 2008). No caso de uma rede GNSS, a observação dos vértices se dá com uso de no mínimo dois receptores observando simultaneamente um par de pontos por vez. Isto já elimina o erro do relógio do satélite, o erro do relógio do receptor e, se forem vértices próximos, também é eliminado o erro devido a ionosfera (HOFMANN-WELLENHOF; LICHTENEGGER; WASLE, 2008).

Para estabelecer uma rede geodésica, as observações dos vértices devem ser redundantes, ou seja, um mesmo vértice “P” deve formar mais de um vetor de observação. A Fig. 1 mostra cinco pontos de observação (1, 2, 3, 4 e 5) de uma rede, ligados pelos seus vetores de observação (linhas). O ponto 1, por exemplo, é observado três vezes com os vetores 1-2, 1-3 e 1-4 garantindo a redundância. Além disto, existe a possibilidade que entre os pares de pontos tenha-se mais de uma observação, aumentando a confiabilidade dos resultados (KLEIN et al., 2012).

Figura 1: Exemplo de vértices e vetores em uma rede geodésica



Fonte: Elaborado pelo autor

Em redes GNSS, uma observação se dá pelo posicionamento GNSS relativo entre dois vértices. Caso o leitor queira se aprofundar no assunto, pode consultar Leick, Rapoport e Tatarnikov (2015), Ghilani (2010) e Monico (2008).

2.2 Erros em observações

Conforme exposto acima, o processo de levantamento de uma rede geodésica está sujeito a erros na parte instrumental, com alguns destes passíveis de eliminação dependendo da técnica de levantamento. Observações, de um modo geral, estão sujeitas a vários tipos de erros, entre eles, erros aleatórios, sistemáticos e grosseiros. Neste trabalho, serão abordados e trabalhados os erros aleatórios e os grosseiros. Os primeiros, são erros inerentes às observações, ou seja, não há como evitá-los. Uma observação sempre conterà um erro aleatório, diretamente relacionado com o instrumento ou técnica utilizada na observação. Já os erros grosseiros, consistem em falhas inesperadas e desconhecidas nas observações, podendo ser de causa humana ou instrumental e de difícil detecção. Quando não identificados, os erros grosseiros podem conduzir a sérios equívocos na estimativa da grandeza observada (KLEIN, 2011).

2.2.1 Erros grosseiros em redes geodésicas

Em redes geodésicas levantadas a partir das técnicas GNSS, também podem ocorrer erros grosseiros nas observações. Mesmo que grande parte do processo de observação não tenha

influência humana direta, podemos citar como exemplo, o erro de multicaminhamento de sinal (reflexão do sinal), em que o sinal eletromagnético que deveria se propagar diretamente ao longo da linha de visão entre o satélite e o receptor, é refletido ou dispersado por algum objeto antes de alcançar o instrumento (HOFMANN-WELLENHOF; LICHTENEGGER; WASLE, 2008). Este erro pode causar falhas graves no vetor de observação e, conseqüentemente, na determinação das coordenadas do vértice e na qualidade da rede.

Podemos citar também, alguns os erros grosseiros causados por falha humana, entre eles: medição incorreta da altura da antena do receptor, neste caso acarretando na informação incorreta ao software de processamento; bastão ou tripé de apoio do receptor não nivelado (aprumado); anotação incorreta da altura da antena (algarismos invertidos, por exemplo); deslocamento do tripé de apoio do receptor durante a observação; entre outros.

Embora os erros aleatórios sejam inevitáveis, os erros grosseiros devem ser evitados e, sempre que possível, identificados e eliminados para a correta determinação das coordenadas na rede. De acordo com Hampel et al. (1986), a frequência de *outliers* varia consideravelmente de um experimento para outro. Porém, de forma geral, dados rotineiros podem conter de 1 a 10% de erros grosseiros, quantidade também adotada nas simulações desta pesquisa.

2.3 Ajustamento da rede

O ajustamento das redes geodésicas, tem por objetivo a estimativa final das coordenadas dos vértices. Geralmente, este procedimento é conduzido aplicando o Método dos Mínimos Quadrados (MMQ ou *Least Squares method* — LS). O MMQ é o melhor estimador linear imparcial para os parâmetros quando atendida a condição de que apenas os erros aleatórios (inevitáveis) estão presentes nas observações (KOCH, 1999). Quando isto não ocorre, o MMQ distribui os erros por vários parâmetros o que dificulta a detecção deles apenas pela busca nos resíduos.

Porém, outros estimadores podem ser aplicados, sobretudo, quando se desconfia da presença de erros grosseiros. Nestes casos, a utilização de um estimador robusto (ER) é uma opção para o cálculo dos parâmetros. Também há possibilidade de aplicar o procedimento de teste *data snooping* com o objetivo de localizar estes erros após o ajustamento por MMQ. Porém, testes estatísticos não são o escopo deste trabalho que focará nos ERs.

Um bom ER, pode estimar os parâmetros de maneira correta mesmo que algumas observações contenham erros grosseiros. Tanto o MMQ quanto os ERs, são métodos importantes e serão abordados em mais detalhe, a seguir.

2.3.1 Método dos mínimos quadrados

A estimativa dos parâmetros pelo MMQ é o método mais conhecido e aplicado dentro das ciências geodésicas. Isso se deve ao fato dele ser o melhor estimador linear imparcial para os

parâmetros, assumida a condição de que apenas erros aleatórios estejam presentes nas observações (KOCH, 1999).

Segundo Ghilani (2010), o ajustamento para redes geodésicas pelo MMQ é dado pelo Modelo de Gauss-Markov (ou Método Paramétrico), dado por:

$$\mathbf{A}_{n \times p} \mathbf{x}_{p \times 1} = \mathbf{l}_{n \times 1} + \mathbf{v}_{n \times 1} \quad (2.1)$$

onde, \mathbf{A} é a matriz das derivadas parciais (matriz *design* ou, ainda, Jacobiana), \mathbf{x} é o vetor de correção aos parâmetros para o modelo linearizado, que queremos estimar, \mathbf{l} o vetor das observações do modelo linearizado e \mathbf{v} o vetor de resíduos. n é igual a quantidade de observações e p o número de parâmetros, ou seja, a quantidade de valores de coordenadas a estimar.

A matriz \mathbf{A} é formada pelas derivadas parciais das equações de observação em relação a cada parâmetro. Para o caso das redes geodésicas que fazem uso exclusivo de métodos de levantamento GNSS, a matriz \mathbf{A} obtida é formada somente por escalares $(-1, 0, 1)$.

Para a formação do vetor \mathbf{l} , podemos tomar como exemplo um vetor de observação formado pelos pontos A e B . Os valores do vetor \mathbf{l} , correspondentes a esta observação, serão dados por:

$$l_{X_{AB}} = X_A + \Delta X_{AB}$$

$$l_{Y_{AB}} = Y_A + \Delta Y_{AB}$$

$$l_{Z_{AB}} = Z_A + \Delta Z_{AB}$$

onde, X_A , Y_A e Z_A , são as coordenadas do vértice A no sistema cartesiano do *datum* adotado e ΔX_{AB} , ΔY_{AB} e ΔZ_{AB} , são as distâncias relativas entre os pontos A e B .

Ocorre que o vetor \mathbf{v} tem na mesma posição, os resíduos calculados para as observações em \mathbf{l} . E o vetor resultante \mathbf{x} , é composto pelas coordenadas dos pontos no sistema de referência adotado (GHILANI, 2010).

De acordo com Koch (1999), a condução do ajustamento pelo MMQ é dada por:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{l} \quad (2.2)$$

onde \mathbf{A}^T representa transposição da matriz das derivadas parciais, $^{-1}$ a inversão matricial e \mathbf{W} a matriz peso das observações. Esta última, é obtida a partir da seguinte equação (GHILANI, 2010):

$$\mathbf{W}_{n \times n} = \sigma^2 \Sigma_{n \times n}^{-1} \quad (2.3)$$

sendo Σ a matriz de covariâncias das observações e σ^2 o fator de variância *a priori*, valor usado para escalar Σ^{-1} , que pode ser, por exemplo, arbitrado como sendo igual a 1.

A solução final, então, é dada por:

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \mathbf{x} \quad (2.4)$$

onde x_0 é o vetor de parâmetros aproximados para as equações de observação, x o vetor dos parâmetros dado pelo modelo para correção do x_0 e \hat{x} o vetor de parâmetros ajustados, contendo as coordenadas ajustadas dos vértices da rede.

2.4 Estimadores robustos

Uma das alternativas que permite o uso do MMQ no ajustamento de redes geodésicas com possíveis erros grosseiros, é a aplicação de métodos que possam detectar a presença destes *outliers* e proceder a rejeição das observações contaminadas, numa etapa anterior ao MMQ. Porém, conforme Huber e Ronchetti (2009), raramente é possível separar o processo nessas duas etapas.

Em um ajustamento com vários parâmetros, os erros grosseiros são difíceis de detectar podendo resultar em falsas rejeições ou falsas retenções. Neste caso, estas duas etapas podem se tornar mais difíceis do que a aplicação direta de um Estimador Robusto (ER). Além disso, estas técnicas dificilmente conseguem lidar com a presença de múltiplos erros grosseiros, pois pode ocorrer de um segundo erro grosseiro ocultar o primeiro (HUBER; RONCHETTI, 2009).

Os ERs são baseados na minimização de uma função (chamada de estimador) com a finalidade de obter uma solução que tenha resistência máxima ao não adequamento às suposições do modelo estatístico. Conforme Huber e Ronchetti (2009), o termo “robusto” significa uma insensibilidade a pequenos afastamentos (*deviations*) das suposições (*assumptions*) adotadas. Em outras palavras, os ERs não devem sofrer grandes distorções na estimativa dos parâmetros quando as observações contiverem alguns erros grosseiros, sistemáticos ou o modelo não estiver adequado.

Neste trabalho, serão abordados e testados diversos ERs, os quais serão detalhados separadamente nas próximas páginas. São eles:

- Mínimos Quadrados Aparados (*Least Trimmed Squares* — LTS) (ROUSSEEUW, 1984);
- Mínima Mediana dos Quadrados (*Least Median of Squares* — LMS) (ROUSSEEUW, 1984);
- Mínimos Quadrados Aparados com Restrição de Redundância (*Least Trimmed Squares with Redundancy Constraint* — LTS-RC) (KOCH et al., 2017);
- Método dos Mínimos Quadrados Robusto com Sinais Restringidos (*Sign-Constrained Robust Least Squares* — SRLS) (XU, 2005);
- Mínimos Quadrados Reponderados Iterativamente (*Iteratively Reweighted Least Squares* — IRLS). Uma abordagem clássica de ajustes dos pesos de forma iterativa para estimação robusta, usando a estratégia de ajustes de pesos I dada em Guo, Ou e Wang (2010);
- Mínimos Desvios Absolutos Aparados (*Least Trimmed Absolute deviations* — LTA) (TABLEMAN, 1994);

2.4.1 Mínimos Quadrados Aparados (*Least Trimmed Squares*)

O LTS foi desenvolvido em Rousseeuw (1984) e substitui a soma do quadrado dos resíduos do MMQ por uma parte da soma do quadrado dos resíduos. O método consiste em usar o vetor de resíduos (\mathbf{v}) do modelo de Gauss-Markov, aplicar o quadrado a cada um de seus valores e somar somente h menores quadrados dos resíduos, suprimindo os maiores. Desta forma, o algoritmo de otimização aplicado deve trabalhar de forma iterativa a fim de minimizar a soma. O modelo matemático do LTS é dado por:

$$\min \sum_{i=1}^h v_i^2 \quad (2.5)$$

onde h representa a quantidade de resíduos que permanecerão no vetor para a soma e v_i^2 corresponde ao quadrado do i -ésimo resíduo do vetor \mathbf{v} do modelo de Gauss-Markov.

Conforme Rousseeuw e Leroy (1987, p. 132), h é definido por:

$$h = (n + p + 1) / 2 \quad (2.6)$$

onde n é o número de observações e p o número de parâmetros. Caso $(n + p + 1)$ seja ímpar, o valor de $(n + p)$ é usado para a divisão por 2. Na prática, a equação 2.6 pode ser implementada como $\text{int} [(n + p + 1) / 2]$, onde a função $\text{int} ()$ extrai o valor inteiro do resultado.

Sabendo que no ajustamento de redes geodésicas por MMQ, a matriz peso das observações (\mathbf{W}) faz parte da estimativa dos parâmetros, faz-se necessário utilizá-la, também, no LTS. Além disso, o fato das observações correlacionadas nas redes geodésicas GNSS formarem uma matriz peso com alguns valores negativos de covariância, o corte dos resíduos deve ser executado somente após a extração dos valores absolutos dos resíduos ponderados ao quadrado, para eliminar a influência dos sinais no ordenamento dos valores no vetor. Isto resulta no seguinte modelo:

$$\min \sum_{i=1}^h |\mathbf{W}\mathbf{v}^2|_i \quad (2.7)$$

onde $|\mathbf{W}\mathbf{v}^2|_i$ é o i -ésimo quadrado do resíduo, ponderado pela matriz peso.

2.4.2 Mínima Mediana dos Quadrados (*Least Median of Squares*)

Também proposto em Rousseeuw (1984), o LMS substitui a soma do quadrado dos resíduos do MMQ pela mediana do quadrado dos resíduos. Assim como no LTS, é feito uso do vetor de resíduos (\mathbf{v}) do modelo de Gauss-Markov e aplicado o quadrado a cada um de seus valores. Após isso, extrai-se o valor de mediana a partir do vetor devidamente ordenado. O modelo

matemático, já adaptado às redes geodésicas GNSS é dado por:

$$\min_{i=1,2,\dots,n} med_h |\mathbf{W}\mathbf{v}^2|_i \quad (2.8)$$

onde W_i é o i -ésimo peso da i -ésima observação, que corresponde a seu resíduo em v_i . A posição da mediana (med_h), é definida mesma forma que o limite de corte do LTS, usando a equação 2.6.

2.4.3 Mínimos Quadrados Aparados com Restrição de Redundância (*Least Trimmed Squares with Redundancy Constraint*)

O LTS-RC é uma modificação do LTS clássico adicionando uma restrição ao processo de eliminação no vetor de resíduos (KOCH et al., 2017). Este ajuste, garante que cada parâmetro continue associado a pelo menos duas observação presentes na soma, evitando perda de redundância nas observações no momento da eliminação dos resíduos.

Como o estimador LTS resolve o modelo removendo os maiores resíduos quadrados ponderados do vetor e , então, executa a soma, a restrição é baseada na ideia de evitar uma quebra na redundância, evitando a eliminação de todos os resíduos associados a um único parâmetro. Ao eliminar os valores mais altos de maneira direta, o somatório pode levar a um valor menor, mas com uma solução irreal, se todos os resíduos relacionados a um único parâmetro forem eliminados.

Conforme o pseudo-código apresentado em Koch et al. (2017), a aplicação da restrição é feita após o cálculo do quadrado dos resíduos ponderados absolutos ($v_W = |\mathbf{W}\mathbf{v}^2|$) quando um processo iterativo começa a eliminar os maiores valores. Enquanto o procedimento iterativo é executado, uma instrução condicional verifica se, ao excluir o resíduo, restam pelo menos dois resíduos associados a duas observações para cada parâmetro. Se esta condição for satisfeita, o resíduo é eliminado; caso contrário, ele será ignorado e a iteração seguirá até o próximo valor no vetor. Ou seja, a solução é penalizada ao manter um valor maior na soma, indicando que não é o caminho ideal para a estimativa. Isso significa que, para uma ajustamento correto, todos os parâmetros devem ter um número suficiente de observações representadas na soma do vetor de resíduos. O processo continua até que todos os $u = n - h$ resíduos estejam eliminados do vetor. Em seguida, o valor final é obtido somando os valores restantes, como no LTS clássico. O pseudo do LTS-RC é apresentado no Algoritmo 1.

Para o modelo de ajustamento das redes geodésicas estabelecidas exclusivamente com observações GNSS, um parâmetro pode ser definido por uma única observação, sendo assim, poderia-se cogitar restringir o algoritmo para $obsCount [p_i] > 1$. Porém, não haveria redundância neste caso e o parâmetro seria definido somente por esta observação. Por isso, o algoritmo procura manter, no mínimo, duas observações por parâmetro garantindo algum ajuste e ponderação.

Data: v_W : Vetor ordenado do quadrado dos resíduos ponderados absolutos e parâmetros relacionados a cada resíduo.
obsCount: Uma matriz com o número total de observações para cada parâmetro, que corresponde ao número total de resíduos.
n: A quantidade de observações.
u: A quantidade de resíduos a serem excluídos de v_W .
Result: v_W com os restantes resíduos para somatório

```

1  $i \leftarrow n - 1$ ; // Recebe posição do maior resíduo no vetor
   ordenado  $v_W$ 
2 while  $u > 0$  do
3   // Recebe os parâmetros ( $p_1, p_2$ ) relacionados ao  $i$ -ésimo
   resíduo
4    $p_1 \leftarrow v_W^i[p_1]$ ;
5    $p_2 \leftarrow v_W^i[p_2]$ ;
6   // Se restam pelo menos mais de duas observações para  $p_1$ 
   e  $p_2$ 
7   if  $obsCount[p_1] > 2$  and  $obsCount[p_2] > 2$  then
8     unset( $v_W^i$ ); // Exclue o atual resíduo
9      $u \leftarrow u - 1$ ; // Diminui a quantidade de resíduos a
   eliminar
10    update(obsCount); // Atualiza o número de observações
   restantes para cada parâmetro
11  end
12   $i \leftarrow i - 1$ ; // Define a posição do próximo maior resíduo no
   vetor ordenado
13 end
14 return  $v_W$ 

```

Algorithm 1: pseudo-código da restrição proposta em Koch et al. (2017).

Nos casos de problemas em que um parâmetro dependa de mais de um tipo de observação para poder ser calculado, o algoritmo deve ser adaptado para garantir que cada tipo de observação esteja presente no mínimo duas vezes. Por exemplo, no caso de uma rede planimétrica por métodos clássicos, uma coordenada precisaria de duas leituras de ângulo (ou azimute) e duas medidas de distância para o ajuste. Sendo assim, o algoritmo deve ser adaptado para garantir a presença destes dois tipos de grandezas por parâmetro, totalizando quatro observações.

2.4.4 Método dos Mínimos Quadrados Robusto com Sinais Restringidos (*Sign-Constrained Robust Least Squares*)

Proposto em Xu (2005), o SRLS, ao contrário dos ERs apresentados até aqui, possui mais de uma função a ser otimizada. Isso torna a sua aplicação mais complexa, porém, segundo o autor, melhora a estimativa dos parâmetros na presença de erros grosseiros.

O SRLS tem como estimador a função abaixo:

$$\min (\mathbf{v}^T \mathbf{W}_R \mathbf{v}) / r_{W_R} \quad (2.9)$$

onde os elementos da diagonal de \mathbf{W}_R são definidos por:

$$w_i^R = \begin{cases} w_i & \text{se } |v_i| \leq C\hat{\sigma}/\sqrt{w_i} \\ 0 & \text{caso contrário} \end{cases} \quad (2.10)$$

sendo C uma constante positiva definida entre o intervalo de $[1, 0; 2, 0]$ de acordo com a quantidade estimada de erros grosseiros nas observações. Para casos com 1% de *outliers*, $C = 2, 0$, para 10% $C = 1, 0$. $\hat{\sigma}$ é uma estimativa do desvio padrão do peso unitário, na prática dado por $1,483 \times med(\sqrt{w_i} |v_i|)$, sendo $med(\sqrt{w_i} |v_i|)$ a mediana dos valores absolutos dos resíduos de $\sqrt{w_i} |v_i|$. r_{W_R} é o posto matricial (*rank*) de \mathbf{W}_R .

Porém, devido a presença de valores de covariância na matriz peso nas redes geodésicas GNSS, a montagem final da matriz \mathbf{W}_R pode ser dada pela Eq. 2.11 (YETKIN; BERBER, 2012; GUO; OU; WANG, 2010):

$$\mathbf{W}_R = \text{diag}(w_1^R, \dots, w_n^R) \cdot \mathbf{W} \cdot \text{diag}(w_1^R, \dots, w_n^R) \quad (2.11)$$

onde $\text{diag}(w_1^R, \dots, w_n^R)$ são os elementos calculados pela equação 2.10, distribuídos ao longo de uma matriz diagonal.

Além da minimização da Eq. 2.9, e da restrição para a montagem de \mathbf{W}_R , o SRLS ainda é condicionado ao seguinte somatório:

$$\sum_{i=1}^n \text{sign}(v_i) = 0 \quad (2.12)$$

com sign sendo a função que analisa o sinal dos resíduos, dada por:

$$\text{sign}(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (2.13)$$

Para os casos em que o número de observações é ímpar, a Eq. 2.12 torna-se $|\sum_{i=1}^n \text{sign}(v_i)| = 1$, sem perda da solução robusta final (XU, 2005). A Eq. 2.12, na prática, é utilizada para penalizar a Eq. 2.9, caso seu resultado seja diferente 0 para n par, ou, 0 ou 1, no caso de n ser ímpar.

2.4.5 Mínimos Quadrados Reponderados Iterativamente (*Iteratively Reweighted Least Squares*)

O método dos Mínimos Quadrados Reponderados Iterativamente (IRLS) representa neste trabalho uma classe de ERs que não necessitam de métodos de otimização numérica para serem solucionados. Estes estimadores trazem consigo um método iterativo próprio para obtenção da solução robusta, trabalhando com alterações nos valores dos pesos de acordo com a estratégia e característica de cada ER. Existem diversos outros ERs deste tipo, como por exemplo, os Estimadores M de Huber. Porém, neste trabalho priorizou-se ERs que não possuíssem equacionamento pronto para resolução do mesmo. Ainda assim, o IRLS foi escolhido e implementado para fins de comparação com os demais ERs.

O IRLS abordado neste trabalho é o proposto em Guo, Ou e Wang (2010) foi escolhido por se tratar de um dos mais recentes e, também, por ter apresentado bons resultados (GUO; OU; WANG, 2010; KOCH et al., 2017). Objetiva-se, com isso, a comparação dos resultados deste ER de abordagem iterativa própria, com outros ERs que necessitam de otimização numérica.

Guo, Ou e Wang (2010) apresentaram duas estratégias de ajuste de pesos, com a estratégia I apresentando melhores resultados. Sendo assim, neste trabalho também é adotada a mesma técnica, que é dada por:

$$w_i = \begin{cases} 1 & \text{se } |\tau_i| \leq C \\ C/|\tau_i| & \text{se } |\tau_i| > C \end{cases} \quad (2.14)$$

onde C uma constante positiva definida entre o intervalo de $[1, 0; 2, 0]$ de acordo com a quantidade estimada de erros grosseiros nas observações. Para casos com 1% de *outliers*, $C = 2, 0$, para 10% $C = 1, 0$. τ_i é dado por:

$$\tau_i = \frac{\mathbf{c}_i^T \mathbf{W} \mathbf{v}}{\sigma \sqrt{\mathbf{c}_i^T \mathbf{W} \mathbf{R} \mathbf{c}_i}} \quad (2.15)$$

com \mathbf{c}_i sendo um vetor nulo com valor de 1 na posição i e σ dado pela mediana normalizada dos resíduos absolutos dividida por 0,6745. Por fim, \mathbf{R} é calculado aplicando a Eq. 2.16:

$$\mathbf{R} = \mathbf{I} - \mathbf{A} (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \quad (2.16)$$

sendo \mathbf{I} uma matriz identidade.

Conforme Guo, Ou e Wang (2010) e utilizando a estratégia de ajuste de pesos I, o IRLS é resolvido conforme os passos a seguir:

1. Definir o contador para $i = 0$ e calcula a solução inicial dos parâmetros $\hat{\mathbf{x}}^{(0)}$ usando a equação 2.2.
2. Definir $i = i + 1$ e implementar a estratégia de ajustes de pesos I com 2.14.

3. Construir a matriz peso equivalente \mathbf{W}_R usando a equação 2.11 aplicando os valores calculados para w_i
4. Parar a iteração se: a diferença $\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(i-1)}\|$ é menor que um valor de parada; ou, um número máximo de iterações foi atingido. Caso contrário, volta ao passo 2 e começa uma nova iteração.

2.4.6 Mínimos Desvios Absolutos Aparados (*Least Trimmed Absolute deviations*)

O LTA desenvolvido em Tableman (1994) tem uma formulação muito parecida com o LTS. Em vez de somar parte da soma do quadrado dos resíduos, o LTA soma parte dos resíduos absolutos. Isto consiste em usar o vetor de resíduos (\mathbf{v}) do modelo de Gauss-Markov, extrair o valor absoluto de cada um de seus valores e somar somente h menores resíduos, suprimindo os maiores. O modelo matemático, já adaptado para o caso de ajustamento de redes geodésicas GNSS é dado por:

$$\min \sum_{i=1}^h |W_i \mathbf{v}|_i \quad (2.17)$$

onde W_i é o i -ésimo peso da i -ésima observação, que corresponde a seu resíduo em v_i . A posição de corte (h), é definida mesma forma que o limite de corte do LTS, usando a equação 2.6.

2.4.7 A posição da mediana no LMS e o limite de corte no LTS e LTS-RC

Conforme exposto anteriormente, o LMS tem a posição da mediana definida por h aplicando a Eq. 2.6. Do mesmo modo, o limite de corte (resíduos restantes para a soma) do LTS e do LTS-RC, é definido com base na mesma equação.

Alguns estudos, porém, utilizam uma variação desta equação, definindo-a conforme abaixo (YETKIN; BERBER, 2014):

$$h = (n + 1) / 2 \quad (2.18)$$

onde n é o número de observações. Neste caso, a implementação pode seguir a mesma lógica da Eq. 2.6 utilizando a função $int()$ que extrai o valor inteiro do resultado, ficando $int[(n + 1) / 2]$.

Percebe-se que a única diferença em relação à Eq. 2.6, é a ausência da variável p (número de parâmetros). A justificativa apresentada para definir o valor de h desta maneira, é de que o estimador apresentaria uma maior robustez e resistência à *outliers*, uma vez que mais resíduos estariam sendo cortados (no caso do LTS e LTS-RC), ou, a mediana deixaria mais e maiores resíduos para além dela (no caso do LMS). Em outras palavras, o estimador pode perder a sua robustez se h for um valor muito rigoroso, ou seja, um limite maior. No caso do LTS, na hipótese de um valor máximo para h , com $h = n$, ele se tornaria igual ao LS (MMQ), pois não haveria eliminações no vetor antes da soma.

Em termos práticos, para os ERs LTS e LTS-RC, um limite menor indica que menos resíduos serão levados em consideração na estimativa, ou seja, mais valores serão ignorados no vetor de resíduos. Enquanto que com um limite maior, mais valores serão somados para a estimativa, ou seja, menos resíduos serão descartados. Para o LMS, a ideia do limite define a posição da mediana. Com o limite maior, a localização do resíduo que será considerado a mediana estará mais afastada do centro em direção aos valores maiores. Ao passo que com o limite menor, a mediana será considerada mais ao centro do vetor de resíduos.

Deste modo, além de testar e analisar todos os ERs apresentados neste capítulo, este estudo também verificou o comportamento dos resultados quando adotada, tanto a Eq. 2.6 quanto a Eq. 2.18 para o valor de h nos ERs LMS, LTS e LTS-RC.

2.4.8 Aplicação dos estimadores robustos

Com exceção do IRLS, todos os outros ERs necessitam de um algoritmo de otimização numérica. Para isso, o algoritmo deve explorar os valores dos parâmetros (coordenadas dos vértices), gerar os resíduos com o modelo de Gauss-Markov e, então, através da função objetivo do ER, calcular um valor de aptidão (*fitness*) da solução. O processo se repete até que um limite de tempo ou computacional seja atingido (KOCH et al., 2017).

Como apresentado anteriormente, neste trabalho, tal tarefa foi realizada por um algoritmo meta-heurístico. As MHs e suas características serão objeto de discussão na seção seguinte.

2.5 Algoritmos meta-heurísticos

Os algoritmos meta-heurísticos, ou meta-heurísticas (MHs), que compreendem o método da meta-heurística, são desenvolvidos para resolver problemas de otimização de maneira genérica sem ter que se adaptar profundamente ao problema em questão. Daí o prefixo “meta” indicando que os algoritmos são um nível superior aos métodos heurísticos de solução de problemas específicos (TALBI, 2009).

As meta-heurísticas geralmente são aplicadas em problemas que não possuem um procedimento específico para resolução. Muitas vezes, remetendo a problemas complexos das mais variadas áreas incluindo a engenharia. Apesar desses algoritmos não terem um propósito específico, a maioria possui características comuns entre eles (BOUSSAÏD; LEPAGNOT; SIARRY, 2013):

- São inspirados na natureza baseando-se na física, biologia ou etologia (comportamento dos animais);
- Fazem uso de componentes estocásticos, envolvendo variáveis randômicas;
- Não fazem uso da matriz de derivadas parciais (*design*) A da função objetiva¹;

¹Apesar dos algoritmos meta-heurísticos não fazerem uso da matriz A , neste trabalho ela não é desprezada.

- Podem possuir diversos parâmetros que devem ser ajustados manualmente de acordo com o problema abordado.

Para conseguirem chegar a solução de um dado problema de otimização, estes algoritmos possuem dois aspectos divergentes e comuns a todos: **exploração** do espaço de procura (diversificação) e **investigação** das melhores soluções encontradas (intensificação).

A exploração é explicada pela capacidade do algoritmo encontrar regiões que possam conter boas soluções dentro de um determinado espaço de busca que, na programação do algoritmo, são buscas aleatórias (TALBI, 2009). Ou seja, aqui a meta-heurística faz uma busca global dentro do limite de procura estabelecido, tentando identificar áreas com resultados promissores para a solução do problema estabelecido.

Na investigação, regiões com soluções promissoras são apuradas mais detalhadamente na expectativa de encontrar resultados melhores (TALBI, 2009). Isto é, o algoritmo intensifica a procura pela solução em uma determinada região, fazendo um tipo de busca local.

Atualmente, no meio científico, é possível encontrar centenas de algoritmos meta-heurísticos. Conforme Sörensen (2015), a maioria das MHs novas, são baseadas em uma nova metáfora apenas renomeando conceitos existentes. Mesmo que os métodos possam ser chamados de novos por seus desenvolvedores, muitos não apresentam novas ideias, exceto uma ocasional variante de uma MH já existente, sem agregar nada a ciência. Em geral, estas metáforas tendem a dificultar o entendimento e a estratégia por trás do método de otimização e, às vezes, nem remotamente se assemelham ao processo em que foram modeladas. Ainda segundo o autor, a verdadeira inovação na pesquisa meta-heurística, não deve vir somente de um novo método que tenha um desempenho melhor do que seus concorrentes, mas, principalmente, deve ser compreendido porque exatamente esse método funciona bem.

Um bom algoritmo meta-heurístico deve ser explicado usando a terminologia de otimização, evitando uma linguagem de alguma metáfora obscura. Em outras palavras, todo projeto meta-heurístico deveria regressar a uma situação na qual os métodos são desenvolvidos com base na percepção da estrutura do problema (SÖRENSEN, 2015).

Neste trabalho, serão abordados cinco algoritmos ao todo². Primeiro, o *Vortex Search* (VS) (DOGAN; ÖLMEZ, 2015a) e o *Modified Vortex Search* (ModVS) (DOGAN, 2016), ambos baseados em uma estratégia de busca que lembra as linhas de corrente de um vórtice, com o último sendo uma proposta de aperfeiçoamento do primeiro. Em seguida, a *Artificial Bee Colony* (Colônia de Abelhas Artificiais — ABC) (KARABOGA, 2005; KARABOGA; BASTURK, 2007) bastante difundida e aplicada no meio científico, que lembra o trabalho coletivo e inteligente das abelhas na busca por comida. Outro algoritmo que será apresentado é o *Hybrid Vortex Search algorithm* (Pesquisa de vórtices híbrida — HVS) (WANG; WU; WAN, 2017) que faz um junção entre a ABC e o VS tentando aproveitar o melhor de cada algoritmo. E, por

Conforme visto, através do modelo de Gauss-Markov, a matriz de derivadas parciais se faz necessária para obtenção do vetor de resíduos v , que é objeto de análise e cálculo dos ERs.

²Caso o leitor queira se aprofundar sobre o tema das MHs, pode consultar Osman e Kelly (1997); Talbi (2009); Boussaïd, Lepagnot e Siarry (2013), entre outros.

último, no capítulo 3, o *Independent Vortices Search* (Pesquisa de Vórtices Independentes — IVS) como proposta de uma nova MH, mais eficiente, com melhores resultados e trazendo uma análise mais profunda do funcionamento dos algoritmos na busca por soluções. Todas as MHs citadas serão explicadas e detalhadas nas subseções seguintes, e servirão de comparação com o desempenho do IVS, que será apresentado no capítulo 3.

2.5.1 *Vortex Search* e *Modified Vortex Search*

O *Vortex Search* (Pesquisa de vórtice — VS) é uma MH de solução única para otimização de funções numéricas. A otimização se dá pela geração de novos candidatos (soluções) ao redor da melhor solução de momento, movendo-se ao longo do espaço de busca quando uma solução melhor é obtida. O raio que define o limite para geração de novas soluções candidatas diminui ao longo dos ciclos e consegue transitar bem de uma pesquisa global para uma local, proporcionando, assim, um equilíbrio adequado entre exploração e investigação.

A limitação do VS, se dá pelo fato de um único centro com raio adaptável fazer a busca, podendo ficar preso em um mínimo local. Deste modo, o *Modified Vortex Search* (Pesquisa de vórtices modificada - ModVS) veio com a proposta de utilizar vários centros ao mesmo tempo na busca do mínimo global.

Tanto o VS quanto o ModVS definem o raio inicial do(s) centro(s) aplicando a Eq. 2.19, abrangendo todo espaço de busca. De modo similar, a solução inicial é obtida aplicando-se a Eq. 2.20 fazendo com que a busca comece no centro dos limites do espaço.

$$r_{0j} = \frac{x_{min,j} - x_{max,j}}{2}, \quad (2.19)$$

$$\mu_{0j} = \frac{x_{min,j} + x_{max,j}}{2}. \quad (2.20)$$

Aqui, r_{0j} é o raio inicial do j -ésimo parâmetro, $x_{min,j}$ e $x_{max,j}$ são, respectivamente, os limites mínimos e máximos do j -ésimo parâmetro. μ_{0j} é, ao mesmo tempo, o centro do vórtice e a solução inicial para o parâmetro j .

Em cada ciclo, o raio diminui para limitar o espaço de geração de novas soluções candidatas, de modo que elas ficam cada vez mais próximas da melhor solução encontrada. A abordagem de um raio decrescente corresponde à definição em Dogan e Ölmez (2015a), que é obtida pela Eq. 2.21 e fornece um controle satisfatório do processo de exploração e investigação.

$$r_c = r_0 \cdot (1/R) \cdot \gamma(R, a_c) \quad (2.21)$$

Sendo r_c o tamanho do raio no ciclo c , R uma constante fixada em 0.1 que controla a resolução da busca do algoritmo ao atingir aproximadamente a metade do número de iterações, e $\gamma(R, a_c)$

a função gama inversa incompleta dada pela Eq. 2.22:

$$\gamma(R, a_c) = \int_0^R e^{-t} t^{a_c-1} dt \quad a_c > 0 \quad (2.22)$$

onde a_c é o parâmetro que define a forma (*shape*) a cada ciclo, definido pela Eq. 2.23 abaixo.

$$a_c = 1 - \frac{c}{C_{max}} \quad (2.23)$$

Aqui, a equação do parâmetro de forma (*shape*) usa a contagem de ciclos c e o número máximo de iterações C_{max} .

O vetor de soluções candidatas s é gerado aleatoriamente sobre o centro μ usando a distribuição Normal Multivariada (DNM) dada pela Eq. 2.24 com desvio padrão de r :

$$p(\mathbf{z}|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{z} - \mu)^T \Sigma^{-1}(\mathbf{z} - \mu)\right) \quad (2.24)$$

onde p representa o número de dimensões (quantidade de parâmetros), Σ a matriz de covariância do parâmetros, \mathbf{z} o $p \times 1$ vetor de variáveis aleatórias, e μ o $p \times 1$ vetor de centros do parâmetro.

Uma vez que as soluções candidatas foram geradas, o algoritmo testa os valores para confirmar se estão dentro do espaço de busca usando a Eq. 2.25, com “rand” definindo um valor real aleatório entre [0;1]:

$$x_{i,j} = \begin{cases} x_{i,j}, & \text{se } x_{min,j} \leq x_{i,j} \leq x_{max,j}; \\ \text{rand} \times (x_{max,j} - x_{min,j}) + x_{min,j}, & \text{caso contrário.} \end{cases} \quad (2.25)$$

Para cada solução candidata, uma função de aptidão avalia o resultado com base na função do problema escolhido. Se algum candidato produzir uma solução melhor que a atual, o centro (μ) será deslocado para a nova solução e os demais candidatos serão descartados. Caso contrário, a melhor solução atual (também centro atual) será mantida e todos os candidatos são rejeitados. Para tal avaliação, uma função de aptidão (Eq. 2.26) pode ser aplicada, onde f_i é o resultado da função objetivo do i -ésimo candidato:

$$\text{fit}_i = \frac{1}{1 + |f_i|} \quad (2.26)$$

Esta função de aptidão não tem influência direta no resultado, apenas normaliza os valores da f_i para que fiquem no intervalo real de [0; 1].

Além do espaço de busca, os demais parâmetros a serem definidos no VS são o número de ciclos e a quantidade de soluções candidatas. Isso ajuda a manter a configuração do algoritmo bastante simples, já que não requer parâmetros adicionais, exceto os citados acima, comuns a todas as meta-heurísticas.

Todas as equações acima são utilizadas tanto pelo VS quanto pelo ModVS. Porém, por fazer uso de vários vórtices, o ModVS ajusta as posições dos seus centros a cada ciclo, baseando-se no vórtice com melhor solução. Isso faz com que haja uma separação entre a posição do centro μ_l para geração de novas soluções, e a melhor solução s_l de um dado vórtice l . Para isso, o ModVS aplica a Eq. 2.27

$$\mu_l = s_l + \text{rand}(s_l + s_{best}) \quad (2.27)$$

onde μ_l é a nova posição do centro l , s_l a melhor solução do centro l e s_{best} o resultado do vórtice com melhor solução. Ou seja, a cada ciclo, todos os centros — exceto o vórtice com melhor solução — ganham novas posições a partir da equação acima. O pseudo-código do *Vortex Search* é apresentado no Algoritmo 2, neste caso, utilizando o número ciclos como critério de parada.

<p>Data: r_0: Centro inicial usando a Eq. 2.19 μ_0: Solução inicial calculada com a Eq. 2.20 $\text{fit}_{s_{best}}$: Aptidão da melhor solução aplicando Eq. 2.26 Result: s_{best}: Melhor solução encontrada</p> <pre> 1 $c \leftarrow 0$; // define o ciclo inicial 2 while $c < C_{max}$ do 3 $s \leftarrow \text{gerar}(\mu)$; // Gera vetor soluções candidatas s ao redor do centro μ 4 $\text{verificarLimites}(s)$; // Verifica os limites do espaço de busca para os candidatos gerados aplicando a Eq. 2.25 5 $s' \leftarrow \text{seleciona}(s)$; // Seleciona melhor solução candidata 6 // Se a aptidão da melhor solução candidata superar a aptidão da melhor solução atual, salva a nova solução 7 if $\text{fit}_{s'} < \text{fit}_{s_{best}}$ then 8 $s_{best} \leftarrow s'$; 9 $\text{fit}_{s_{best}} \leftarrow \text{fit}_{s'}$; 10 end 11 $\mu_{c+1} \leftarrow s_{best}$; // Centro é movido para a melhor solução 12 $r_{c+1} \leftarrow \text{decreceRaio}(r_c)$; // Decresce tamanho do raio 13 $c = c + 1$; 14 end 15 return s_{best} </pre>

Algorithm 2: Pseudo-código do *Vortex Search* original

2.5.2 Artificial Bee Colony

Entre os algoritmos de inteligência coletiva mais conhecidos está a Colônia Artificial de Abelhas (*Artificial Bee Colony* - ABC) proposto inicialmente em Karaboga (2005) e disponível

gratuitamente¹. A ABC é um algoritmo de minimização numérica, baseado no comportamento das abelhas, fazendo a procura dos valores ótimos que atendem a uma dada função objetivo. Suas principais características serão abordadas a seguir, divididas em, organização da colônia e estrutura do algoritmo, parâmetros e execução do algoritmo, e, processos de seleção envolvidos.

2.5.2.1 Organização da colônia e estrutura do algoritmo

A organização da colônia artificial se dá pela divisão em três grupos de abelhas, conforme proposto e desenvolvido em Karaboga (2005) e Karaboga e Basturk (2007):

- Abelhas trabalhadoras artificiais: são alocadas nas fontes de alimento (ou comidas) e representam 50% das abelhas da colônia. Para cada fonte de alimento existente, apenas uma abelha trabalhadora é alocada, logo, o número de trabalhadoras é igual ao número de fontes de alimento;
- Abelhas observadoras artificiais: representam a outra metade da colônia. São as abelhas que ficam na chamada área de dança da colônia e esperam as informações sobre fontes de alimento que as abelhas trabalhadoras trazem. Assim, podem escolher ir em uma fonte de acordo com uma probabilidade proporcional a qualidade da mesma;
- Abelhas exploradoras artificiais: visto que uma fonte de alimento tenha sido investigada sem sucesso, a abelha trabalhadora abandona-a e se torna uma exploradora, procurando uma nova comida aleatoriamente.

No algoritmo, a posição das fontes de alimento se refere a uma possível solução para o problema de otimização enquanto a quantidade de néctar de uma fonte corresponde a qualidade (aptidão ou *fitness*) daquela solução (KARABOGA; BASTURK, 2007). Segundo Karaboga et al. (2012), o algoritmo genérico da ABC é dado pelas seguintes etapas:

- Fase de inicialização
- REPETE
 - a) Fase das abelhas trabalhadoras
 - b) Fase das abelhas observadoras
 - c) Fase das abelhas exploradoras
 - d) Memoriza a melhor solução encontrada até então
- ATÉ (Ciclos = Número Máximo de Ciclos ou Tempo de Execução Máximo da CPU)

¹<http://mf.erciyes.edu.tr/abc/>

Ao iniciar, são definidos alguns parâmetros de execução e o algoritmo gera aleatoriamente as fontes de alimento com uso de abelhas exploradoras artificiais, compreendendo a fase de inicialização (KARABOGA; BASTURK, 2007).

Em seguida, o algoritmo entra no processo de repetição (ciclos), iniciando com a fase das abelhas trabalhadoras. Nesta fase, cada uma das abelhas vai a uma fonte de alimento e trabalha na vizinhança próxima a procura de uma fonte que tenha mais néctar do que a fonte salva na memória (atual). Ao encontrar uma nova fonte elas verificam a quantidade de néctar e calculam a aptidão (*fitness*). Após isto, elas compartilham as informações com as abelhas observadoras que estão a espera na área de dança da colônia (KARABOGA; BASTURK, 2007).

Na fase seguinte, as abelhas observadoras artificiais escolhem uma fonte de alimento de forma probabilística, dependendo da informação trazida pelas abelhas trabalhadoras, com o objetivo de explorá-la. Quanto mais néctar (mais aptidão) uma fonte tiver, maior a chance de ser escolhida pelas abelhas observadoras. E, assim como na fase anterior, ao chegar na fonte, a abelha observadora também procura uma nova fonte na vizinhança que tenha mais néctar do que a fonte salva na memória (KARABOGA; BASTURK, 2007).

Na última fase, as abelhas trabalhadoras que não conseguiram encontrar fontes de alimento melhores na vizinhança, depois de uma certa quantidade de tentativas, passam a ser abelhas exploradoras e partem para procura de novas fontes de alimento. Esta nova posição substituirá a fonte anterior na memória. No modelo de algoritmo proposto em Karaboga e Basturk (2007), no máximo uma abelha exploradora poderá sair a procura de novas fontes por ciclo de execução.

As três fases descritas acima, são repetidas até que um critério de parada seja atendido, como número máximo de ciclos ou tempo de execução da CPU (KARABOGA et al., 2012).

2.5.2.2 Parâmetros e execução do algoritmo

Os parâmetros a serem definidos para a execução do algoritmo da ABC são: o número de fontes de alimento (número de soluções) SN que serão geradas na fase de inicialização e que será igual ao número de abelhas trabalhadoras; o valor de limite L para abandono de uma fonte (solução), após sem sucesso tentar aprimorar o valor de aptidão; e o número máximo de ciclos de execução (KARABOGA; BASTURK, 2007).

Aqui, também pode-se incluir o limite do espaço de busca para cada parâmetro, $x_{min,j}$, $x_{max,j}$, sendo respectivamente, o limite mínimo e máximo do j -ésimo parâmetro.

Na fase de inicialização, o algoritmo produz aleatoriamente SN soluções, que são a quantidade de posições de fontes de alimento. Ou seja, para um SN de 25, teremos 25 abelhas trabalhadores e 25 observadoras, trabalhando 50 soluções por ciclo. Cada solução x_i com $i = 1, 2, \dots, SN$, é um vetor de dimensão j , onde j que corresponde a quantidade de parâmetros do problema (KARABOGA; BASTURK, 2007).

Após a inicialização, começa o processo de buscas por soluções com a contagem dos ciclos, $c = 1, 2, \dots, C_{max}$, onde C_{max} é o limite máximo de ciclos estabelecido. As abelhas

trabalhadoras e observadoras produzem mutações (modificações) de forma probabilística nas soluções (fontes) encontrando um novo candidato a solução. Esta nova fonte então é testada analisando a aptidão (quantidade de néctar). Se for melhor que a solução anterior, então a nova fonte é substituída na memória, caso contrário a anterior é mantida. A diferença entre as abelhas trabalhadoras e as observadoras está no fato de que há apenas uma abelha trabalhadora trabalhando por fonte, enquanto as abelhas observadoras escolhem probabilisticamente em qual solução irão trabalhar. Isto permite ter várias abelhas observadoras fazendo mutações numa mesma fonte durante a fase delas. Esta escolha probabilística p_i é dada pela Eq. 2.28:

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n}, \quad (2.28)$$

onde fit_i é o valor de aptidão (*fitness*) da i -ésima solução calculada pela abelha trabalhadora (proporcional a quantidade de néctar encontrado), dividido pelo somatório de todos os valores de aptidão das SN soluções. Quanto maior o valor de p_i maior a chance de uma abelha observadora escolher aquela solução para explorar (KARABOGA; BASTURK, 2007).

No caso de um problema de minimização, o valor de aptidão fit_i é o inverso do resultado da função objetivo, ou seja, quanto menor o resultado da função objetivo calculado a partir das soluções, maior o valor de aptidão. Sendo assim, temos:

$$\text{fit}_i = \frac{1}{f_i}, \quad (2.29)$$

onde f_i é o resultado da função objetivo, obtido a partir da x_i solução (fonte) (KARABOGA; BASTURK, 2007).

É importante notar que quando f_i for igual a 0 esta equação apresentará uma divisão por 0 que resultará em erro no algoritmo. Como alternativa, Akay e Karaboga (2012) propuseram que o valor de aptidão pode ser produzido pela Eq. 2.30:

$$\text{fit}_i = \begin{cases} \frac{1}{1+f_i} & \text{se } f_i \geq 0 \\ 1 + |f_i| & \text{se } f_i < 0 \end{cases} \quad (2.30)$$

Para produzir as mutações nas soluções, tanto as abelhas trabalhadoras quanto as observadoras, aplicam a Eq. 2.31, conforme Karaboga e Basturk (2007):

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (2.31)$$

onde $k \in \{1, 2, \dots, SN\}$ e $j \in \{1, 2, \dots, p\}$. Tanto k quanto j são índices escolhidos aleatoriamente, porém k deve ser diferente i . x_{ij} é a i -ésima solução (atual) para o parâmetro j , ϕ_{ij} é um número real aleatório entre $[-1, 1]$ que produz a nova solução (fonte) em torno de x_{ij} , e v_{ij} é a mutação gerada para a i -ésima solução do parâmetro j . Caso a Eq. 2.31 produza um valor fora dos limites estabelecidos nos parâmetros, ele pode ser alterado para um valor aceitável, como o próprio limite do espaço de busca (AKAY; KARABOGA, 2012).

Se após um dado número de ciclos estipulados, uma determinada solução x_{ij} não obtiver melhoras com as modificações geradas, ela é abandonada. Quando isso acontece a fase das abelhas exploradoras gera uma solução completamente nova que é dada a partir da Eq. 2.32 (KARABOGA; AKAY, 2009):

$$x_{ij} = x_{min,j} + \text{rand}(x_{max,j} - x_{min,j}) \quad (2.32)$$

onde, $x_{min,j}$ e $x_{max,j}$ são respectivamente os limites inferiores e superiores para o parâmetro j e “rand” é um número real aleatório entre $[0, 1]$.

2.5.2.3 Processos de seleção envolvidos

Segundo Karaboga e Basturk (2007), o algoritmo ABC envolve quatro processos de seleção:

1. Uma seleção global usada pelas observadoras artificiais para descoberta de regiões promissoras, com base na Eq. 2.28.
2. Uma seleção local executada pelas trabalhadoras e observadoras artificiais para determinar fontes de alimento ao redor da fonte na memória, aplicando a Eq. 2.31.
3. Uma seleção local praticada por todas as abelhas artificiais em que avaliam se a quantidade de néctar do novo candidato é maior do que a atual, esquecendo a atual e memorizando a nova posição.
4. Um processo de seleção aleatório realizado pelas exploradoras artificiais, conforme a Eq. 2.32.

Todos estes processos de seleção garantem um algoritmo poderoso que não somente trabalhe em uma busca local (baseando-se nos valores iniciais para gerar novos) mas também execute uma busca global por novas soluções dentro do espaço de procura. Para mais detalhes sobre a ABC, o leitor pode consultar Karaboga (2005), Karaboga e Basturk (2007, 2008), Karaboga e Akay (2009), Karaboga et al. (2012), Akay e Karaboga (2012) e Mernik et al. (2015).

2.5.3 *Hybrid Vortex Search algorithm*

O *Hybrid Vortex Search algorithm* (Pesquisa de vórtices híbrida — HVS) apresenta um algoritmo que combina as MHs VS e ABC. Segundo Wang, Wu e Wan (2017), embora o VS tenha bom desempenho na solução de problemas de otimização numérica, ocasionalmente, pode ocorrer do algoritmo não pesquisar por todo o espaço busca. Combinando o VS e a ABC — que apresenta bom desempenho na exploração — o trabalho traz um algoritmo híbrido para solução de problemas de otimização numérica.

Uma vez compreendidos os algoritmo do VS e da ABC, o funcionamento do HVS é bastante simples, pois as MHs são empregadas de forma sequencial e com pouquíssimas modificações nos algoritmo originais. Ao iniciar a otimização pelo HVS, o algoritmo executa a ABC. Nesta fase, a única diferença para o algoritmo original, é a ausência da fase das abelhas exploradoras. A ABC é executada até atingir 10% do total de ciclos (ou limite definido). Uma vez atingido este limite, o HVS pega a melhor solução salva na memória, e a define como sendo o centro inicial do VS, pulando a fase de geração da solução inicial do VS.

A partir daí, o HVS executa o VS até atingir o limite de ciclos (ou outro) estabelecido, com uma pequena modificação. A cada geração de candidatos, o algoritmo verifica: se a solução atual s_{best} for inferior a melhor solução dentre os candidatos s' , então s_{best} é substituída pelo melhor candidato — como ocorre no VS; porém, se a solução atual continuar superior ao melhor candidato, então, o algoritmo gera uma componente aleatória — um número real aleatório “rand” no intervalo de $[0; 1]$ — e verifica: se “rand” for menor que 0,5, o centro permanece com a posição da melhor solução global s_{best} ; caso contrário, o centro é definido na posição da melhor solução entre os candidatos s' . A aplicação desta estratégia, segundo Wang, Wu e Wan (2017), se dá pelo fato de ser perigoso tratar a melhor solução s_{best} como centro o todo tempo, pois pode ocorrer dela ser um mínimo local e o algoritmo do VS estaria se precipitando. O algoritmo 3, apresenta o pseudo-código desta estratégia.

```

1 // Se a solução candidata apresentar resultado pior que a
  melhor solução
2 if  $fit_{s'} < fit_{s_{best}}$  then
3   // Se “rand” menor que 0,5, a posição do centro será
      $s_{best}$ , caso contrário será  $s'$ 
4   if  $rand < 0,5$  then
5     |  $\mu \leftarrow s_{best}$ ;
6   else
7     |  $\mu \leftarrow s'$ ;
8   end
9 else
10  // Se não, a posição do centro é definida em  $s'$ 
11  |  $s_{best} \leftarrow s'$ ;
12  |  $\mu \leftarrow s_{best}$ ;
13 end

```

Algorithm 3: Pseudo-código da estratégia do *Hybrid Vortex Search* de definição da posição do centro

2.6 Trabalhos Relacionados

Conforme visto, as MHs são bastante genéricas, podendo ser aplicadas em vários problemas. A implementação delas no ajustamento de redes geodésicas não é uma proposta inédita, porém, mostra-se recente e pouco explorada. Durante a revisão teórica, realizou-se várias buscas por

estudos que fizessem uso desta estratégia e constatou-se que a abordagem é relativamente nova, com o primeiro estudo publicado em 2007 e o mais recente em 2017. Também, é um método pouco investigado nas geociências, especialmente no ajustamento de redes geodésicas onde foram encontrados somente quatro trabalhos científicos publicados em periódicos.

Os trabalhos existentes limitam-se a trabalhar com apenas um ou dois ERs e muitas vezes em casos simples de ajustamento, sem geração de vários cenários de erros, ou sem erros aleatórios, apenas grosseiros (*outliers*). Pelo fato dos erros aleatórios serem inerentes às observações é de suma importância que se explore soluções de ajustamento que simulem eles, aproximando-se ao máximo de uma situação real.

O primeiro trabalho encontrado na revisão teórica, foi apresentado em Baselga (2007). O estudo mostrou a aplicação da otimização global por meio das MHs *Simulated Annealing* e Algoritmo Genético, fazendo uso do ER norma ℓ_1 , na identificação de *outliers* em redes. No caso específico do trabalho, se tratava de uma rede de observações no sótão de uma construção antiga para observar deformações devido a temporais. O trabalho considerou apenas um único cenário de erros aleatórios e dividiu a simulação de erros grosseiros em duas. Uma com dois erros grosseiros de grande magnitude e outra com seis erros grosseiros de grande magnitude. Isso deixa em aberto duas questões: qual seria o comportamento da solução para 1) diferentes cenários de erros aleatórios? e 2) *outliers* de magnitudes menores, próximas das magnitudes dos erros aleatórios?

O segundo trabalho em Yetkin e Berber (2012) fez uso do ER SRLS, já apresentado aqui, aplicando a MH *Shuffled Frog-Leaping Algorithm*. O estudo apresentou dois exemplos de erros: o primeiro, em uma rede de nivelamento, simulou dois casos com erros grosseiros. No caso I, com apenas um *outlier* e no caso II, com três *outliers*. O segundo exemplo, em uma rede GNSS, apresentou um único caso com três erros grosseiros. Infelizmente, este trabalho também considerou somente um único cenário de erros aleatórios, sempre usando erros grosseiros de grande magnitude. Isso levanta as mesmas questões do trabalho em Baselga (2007), citadas acima.

Outro trabalho em Yetkin e Berber (2014), fez uso dos ERs LMS e LTS, aplicando a MH da Colônia de Abelhas Artificiais. Utilizando da mesma rede GNSS do trabalho anterior, os autores simularam quatro erros grosseiros de grande magnitude, ainda deixando em aberto todas as questões levantadas aqui anteriormente.

O estudo mais recente foi conduzido em Koch et al. (2017) e fez uso da MH *Vortex Search* comparando o ER LTS com a proposta de um novo, chamado de Mínimos Quadrados Aparados com Restrição de Redundância (*Least Trimmed Squares with Redundancy Constraint* - LTS-RC) já apresentado aqui. O trabalho testou diversos cenários de magnitudes de *outliers* com ambos os ERs, porém não utilizou quaisquer erros aleatórios, deixando em aberto uma questão: qual seria o comportamento da solução para cenários que contêm erros aleatórios?

Apesar das diferentes abordagens usando ERs e MHs distintos, e além das questões levantadas acima, é possível concluir uma rotina comum para implementação do ajustamento de redes

GNSS com uso de ERs aplicados por MHs:

1. Escolha da rede a ser ajustada;
2. Obtenção dos vetores de observação;
3. Montagem do modelo de Gauss-Markov (definição das matrizes A e W e do vetor l);
4. Escolha do ER a ser aplicado;
5. Definição do algoritmo meta-heurístico para minimização da função do ER;
6. Implementação dos algoritmos;
7. Execução das rotinas de ajustamento e obtenção dos resultados.

Expostos todos os ERs considerados neste trabalho, a fundamentação e características dos trabalhos relacionados, além da rotina de implementação, apresentar-se-á, no próxima capítulo, o novo algoritmo meta-heurístico desenvolvido nesta pesquisa e seus respectivos resultados.

3 O INDEPENDENT VORTICES SEARCH

Este capítulo apresenta uma nova meta-heurística (MH), denominada *Independent Vortices Search* (Pesquisa de vórtices independentes - IVS), como primeira contribuição desta dissertação para a pesquisa acadêmica. Primeiro, serão abordadas as características e limitações do VS e ModVS, seguidas pelas estratégias adotadas no desenvolvimento do IVS, apresentando tanto a descrição do método quanto a discussão. Por último, serão apresentados os resultados de experimentos comprovando a eficiência do novo método.

3.1 Análise das características e limitações do VS e ModVS

Conforme visto no capítulo anterior, a otimização do *Vortex Search* (Pesquisa de vórtice — VS) se dá pela geração de novas soluções ao redor da melhor solução de momento, utilizando uma distribuição normal multivariada (DNM), onde o desvio-padrão é igual ao raio do vórtice. O centro do vórtice move-se ao longo do espaço de busca a medida que uma solução melhor é encontrada e, o raio, diminui ao longo dos ciclos utilizando uma função gama inversa incompleta, que faz a transição da pesquisa global à busca local.

Diferente dos Algoritmos Genéticos (AGs) e outras MHs, como a Colônia de Abelhas Artificiais (ABC), no VS não há combinações (*crossover*) nem mutações entre soluções candidatas. A cada ciclo, um número pré-definido de candidatos são gerados usando uma DNM, onde a melhor solução — seja a atual ou uma nova gerada — é salva e adotada como novo centro, sendo descartadas as demais. Isso remete um pouco a estratégia denominada de Elitismo e adotada por alguns AGs, onde o melhor indivíduo é mantido de uma geração para a outra para evitar a perda da melhor solução candidata. No VS, a melhor solução nunca é descartada. Também, pelo fato de não haver combinações nem mutações nos candidatos, não há necessidade utilizar estratégias de seleção de indivíduos. Do mesmo modo, não se encontra no VS, a necessidade de definir parâmetros extras como Taxa de Mutação, comum nos AGs, ou, Inércia e Confiança, presentes em algoritmos de Otimização por Enxame de Partículas. Essas características, além de simplificarem o algoritmo, tornam a configuração do VS mais simples, diminuindo a interferência humana na busca por uma solução adequada ao problema.

Outro aspecto interessante do VS é a transição entre a exploração e a investigação (busca global e local). Em muitas MHs, como ocorre na ABC e nos AGs, a exploração e investigação podem ocorrer simultaneamente do início ao fim da execução do algoritmo. Isso traz a possibilidade dessas MHs encontrarem uma solução suficientemente boa a qualquer momento da execução, inclusive nos primeiros ciclos, podendo interromper o processo (se assim configuradas). E, caso não encontrem, a exploração global continua ocorrendo mesmo nas últimas iterações, tentando reduzir a chance de ocorrência de soluções em mínimos locais. Especificamente na ABC, outra estratégia para evitar soluções presas em mínimos, é descartar candidatos (alimentos) que não estão mais aprimorando o valor de aptidão, dando lugar a outras soluções

(aleatórias) na memória. Isso faz com que essas MHs possuam uma estratégia **uniformemente presente** para geração de soluções por todo o processo.

No VS, como visto na seção 2.5.1, o limite para geração de soluções, é definido pelo tamanho do raio do vórtice. Este raio é reduzido ao longo dos ciclos por uma função gama inversa incompleta (Eq. 2.22), levando em consideração a proporção de ciclos já executados¹ (Eq. 2.21). Deste modo, a exploração e investigação ocorrem de forma **transitória**, não estando presentes durante toda a execução, e ainda assim proporcionando um equilíbrio adequado. A vantagem desta estratégia, é o ajuste fino sobre a melhor solução encontrada nos últimos ciclos. Ou seja, a solução será intensificada e investigada ao máximo ajustando minimamente cada parâmetro na tentativa de otimizá-la por completo, enquanto que nos primeiros ciclos, os candidatos são gerados aleatoriamente por todo o espaço de busca, configurando a exploração.

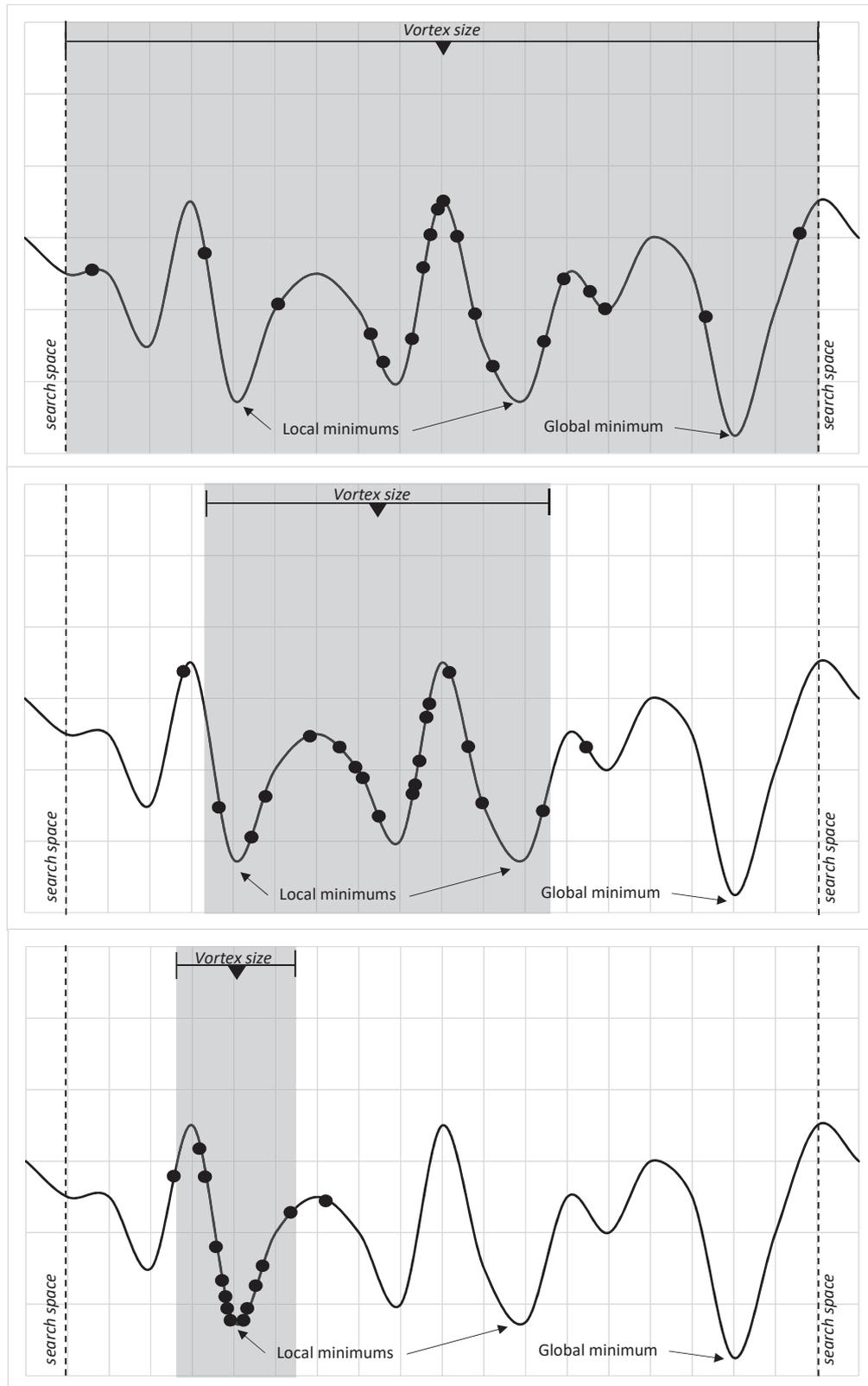
Porém, o fato de haver uma transição entre a exploração e a investigação traz duas desvantagens. A primeira, é o risco do mínimo local. Apesar de ser uma vulnerabilidade comum a todas as MHs, no VS, uma vez que o raio do vórtice for suficientemente pequeno e a solução encontrar-se presa num mínimo local, não haverá qualquer possibilidade da solução global (ou suficientemente boa) ser obtida na naquela execução. Em outras palavras, o raio do vórtice estará com uma dimensão tão pequena, que os candidatos gerados pela DNM ficarão limitados a busca local (investigação), longe da solução global. A Fig. 2 apresenta um cenário hipotético de busca pelo mínimo global, mostrando três momentos na convergência do raio do vórtice durante a execução. A figura identifica a fragilidade do VS de ficar preso em um mínimo local.

A segunda inconveniência, diz respeito a possibilidade de identificar a dispensabilidade de processamento requisitada inicialmente, durante a execução. Em outras palavras, ao definir um limite de execução com número de ciclos ou cálculos de aptidão (*Fitness Evaluations* — FE) no VS, estes, terão que ser executados em sua quase totalidade, pois conforme visto, a busca local (investigação) é realizada somente nas últimas iterações. Para ilustrar melhor, imaginemos um cenário hipotético em que 500.000 FE fossem o bastante para obter uma solução suficientemente boa para um dado problema. Uma MH como a ABC, configurada para um limite de 10.000.000 FE (20x maior), poderia interromper sua execução em torno das 500.000 FE, pois provavelmente já teria encontrado essa solução. Já o VS, acabaria por executar boa parte das 10.000.000 FE, pois apenas no último terço ou quarto dos ciclos estaria contemplando a busca local, aprimorando os valores da solução.

Para tentar contornar o problema da solução ficar presa em um mínimo local, Dogan (2016) propôs o *Modified Vortex Search* (Pesquisa de vórtices modificada — ModVS) em que o algoritmo trabalha com vários vórtices ao mesmo tempo, conforme já exposto na seção 2.5.1. Utilizando a Eq. 2.27, o ModVS ajusta a posição do centro dos vórtices, exceto o de melhor solução, de modo que estes vórtices venham a se aproximar do vórtice com melhor solução a cada ciclo.

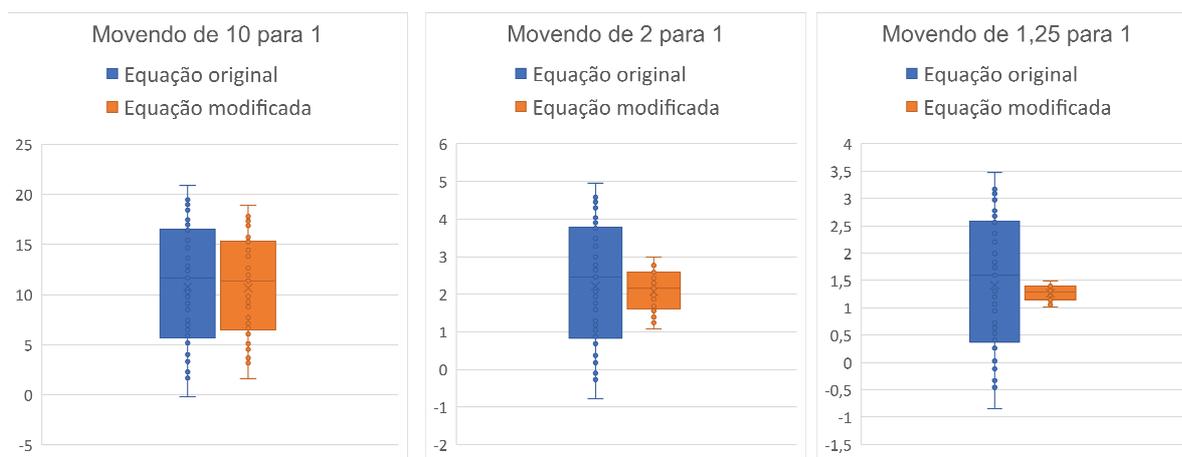
¹Em termos práticos, a dimensão do raio terá aproximadamente 10% do tamanho original ao alcançar metade dos ciclos requisitados. Ou seja, 10% do tamanho do espaço de busca.

Figura 2: Cenário hipotético de busca realizada pelo VS. São apresentados três momentos na convergência do raio do vórtex durante a execução. Ao final, vê-se a fragilidade do VS em ficar preso em um mínimo local. Os pontos indicam candidatos a soluções gerados pelo VS.



Contudo, conforme será apresentado mais adiante, os testes realizados indicam que esta estratégia não resolveu o problema. Fazendo uma análise da Eq. 2.27 e da estratégia apresentada no trabalho, pode-se concluir que o desempenho desfavorável deve-se, provavelmente, a dois problemas: 1) aplicar a Eq. 2.27 de modo que o melhor vórtice influencie no centro dos demais, não necessariamente trará as demais soluções a um mínimo global. Exemplo: caso o vórtice “A” esteja com seu centro em uma posição que, com o passar dos ciclos, possa convergir para o mínimo global, e o vórtice “B”, distante dentro do espaço de busca, está com a melhor solução geral naquele determinado ciclo, ocorrerá que o centro de “A” será “arrastado” em direção ao vórtice “B”. Isso pode fazer com que o raio de “A” deixe de conter o mínimo global em sua área de geração de soluções candidatas, piorando a busca; 2) A Eq. 2.27 apresenta uma dificuldade de balanceamento na geração de valores. Em um problema de minimização, quando o vórtice selecionado estiver muito próximo da melhor solução atual, a equação tende a extrapolar os valores. A substituição do sinal de “+” por “-” na parte final da equação, poderia reduzir o problema dando mais equilíbrio aos valores gerados, com a maioria deles ficando entre o intervalo da solução selecionada e da melhor solução. Assim, a equação passa a ser: $\mu_l = s_l + \text{rand}(s_l - s_{best})$. No caso da solução selecionada estar distante da melhor solução atual, este problema não ocorre nem a Eq. 2.27, nem com a modificação apresentada. A Fig. 3 ilustra as duas situações em três cenários distintos com um centro de pior soluções sendo ajustado ao centro de melhor solução. Enquanto a Eq. 2.27 excede muito os valores intermediários, uma vez que o vórtex já está próximo da melhor solução, a proposta de modificação deixa a maioria dos valores dentro do esperado.

Figura 3: Três situações hipotéticas em que o centro de um vórtex (10,0; 2,0; 1,25), é direcionado para a melhor solução atual (1,0). O gráfico apresenta 50 valores gerados utilizando ambas as equações nas três situações.



Fonte: Elaborado pelo autor

Expostas estas as limitações, o desenvolvimento de modificações para o VS/ModVS teve como base manter as características que interessavam — como a ideia de obter um ajuste fino sobre a solução global — além de tentar contornar o problema do mínimo local sem aumentar

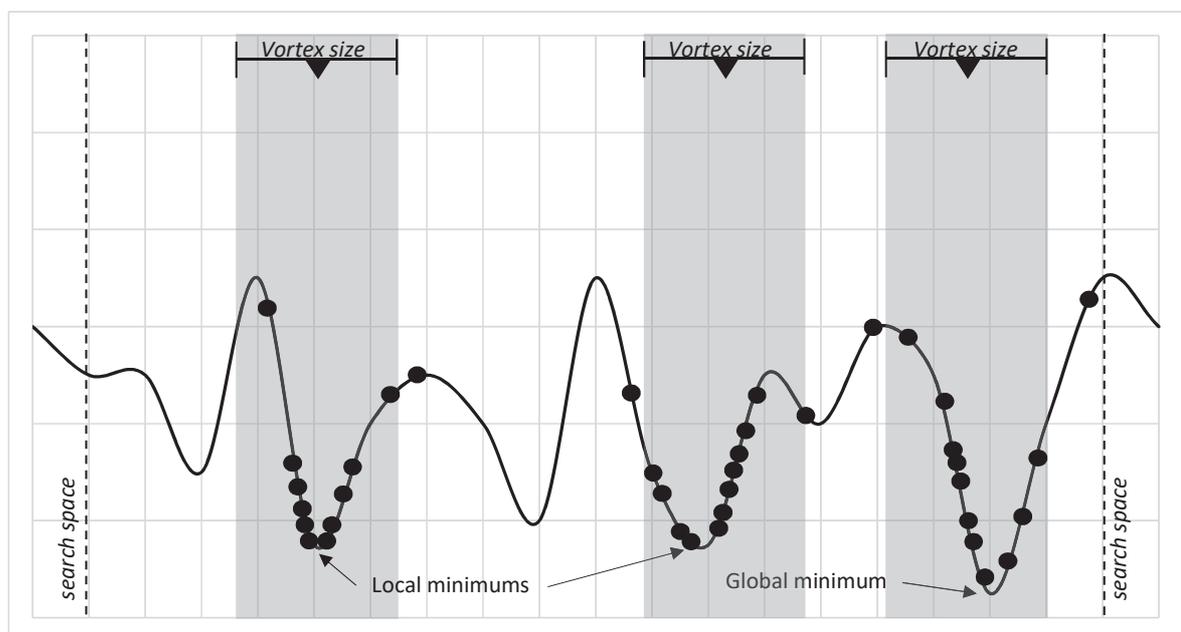
o custo computacional. A seção a seguir apresentará o IVS, explicando a estratégia adotada.

3.2 Desenvolvimento do IVS

O IVS parte da ideia do *Modified Vortex Search algorithm* (Pesquisa de vórtices modificada - ModVS) e segue uma linha de pensamento que acaba por torná-lo menos complexo, além de mais eficiente na busca da solução. Três estratégias que diferem o IVS do ModVS são propostas.

A primeira grande diferença se dá pela completa eliminação da Eq. 2.27, que controla a influência dos diversos vórtices uns em relação aos outros. No IVS, os centros passam a agir de forma independente sem influência dos vizinhos ou do centro de melhor solução. Dessa forma, é assegurado que cada vórtice prossiga seu caminho em direção ao mínimo, seja local ou global. O problema da solução mínima global *versus* local, é, em grande parte, resolvida aumentando a quantidade de vórtices de acordo com o problema a ser otimizado. A Fig. 4 apresenta uma situação conjecturada em que os vórtices agem de modo independente seguindo seus caminhos em direção aos mínimos. Graças a característica de ajuste fino das soluções, cada vórtice explorará os respectivos mínimos encontrados.

Figura 4: Cenário hipotético de busca realizada pelo IVS. É apresentado o momento na convergência do raio do vórtex em que cada vórtice seguirá sua investigação ao mínimo. Os pontos indicam candidatos a soluções gerados pelo IVS.



Fonte: Elaborado pelo autor

A segunda característica distinta do IVS, substitui a Eq. 2.20 para cada vórtice adicional. Em outras palavras, o primeiro vórtice terá a solução inicial definida com a aplicação da Eq. 2.20, enquanto as demais serão geradas de forma aleatória no espaço de busca conforme Eq.

3.1:

$$\mu_{0j}^l = \text{rand} \times (x_{\max,j} - x_{\min,j}) + x_{\min,j}. \quad (3.1)$$

Onde, μ_{0j}^l é a solução inicial para o parâmetro j do vórtice l , com $l > 0$, $x_{\min,j}$ e $x_{\max,j}$ são, respectivamente, os limites mínimos e máximos do j -ésimo parâmetro e “rand” um valor real aleatório entre [0;1].

A terceira e última modificação no IVS, considera a manutenção do número de candidatos por vórtice quando configurados mais centros. Na proposta do ModVS, o autor trabalhou com cinco centros, dividindo os candidatos entre eles, o que, conforme exposto no próprio artigo, não melhorou muito a solução encontrada, podendo o ModVS ainda ficar preso em mínimos locais (DOGAN, 2016). No IVS, entretanto, se mantém o número fixo de candidatos por vórtice, compensando o acréscimo computacional com a diminuição no número total de ciclos. Em outras palavras, quantos mais vórtices, menos iterações serão executadas proporcionalmente, mantendo o número total de FEs inicialmente previstos.

Ainda, é possível verificar que as duas primeiras estratégias, abrem a possibilidade de uma maior diversidade populacional. A eliminação da Eq. 2.27, libera cada vórtice para prosseguir com suas populações/candidatos, não arrastando-os para próximo de um vórtice com melhor resultado global de momento. A segunda estratégia (Eq. 3.1), permite uma maior diversidade já na geração das primeiras soluções. Isso porque a DNM, que utiliza o centro do vórtice como média para geração de candidatos, gera mais soluções próximas ao centro e menos longe dela. Com os centros dos vórtices mais distribuídos, os candidatos não ficarão concentrados em apenas uma parte do espaço de busca. A Tab. 1 apresenta uma tabela com as principais características das meta-heurísticas abordadas neste trabalho.

Diversos experimentos foram realizados para testes e estas estratégias foram as que se mostraram mais eficientes. Na próxima seção veremos os experimentos rodados e respectivos resultados.

3.3 Experimentos e resultados do IVS

Diversos testes foram conduzidos no desenvolvimento do IVS e seus resultados comparados com soluções obtidas por outras MHs. Para melhor análise, quatro cenários de erros em observações de uma rede geodésica foram gerados. Usando o estimador LMS, as MHs executaram vinte e cinco vezes (25x) todos os quatro cenários. Isso permitiu tanto identificar a MH com melhor resultado, assim como analisar a estabilidade das soluções nas 25 execuções.

As MHs testadas foram:

- *Artificial Bee Colony* (Colônia de Abelhas Artificiais — ABC) (KARABOGA, 2005; KARABOGA; BASTURK, 2007);
- *Vortex Search algorithm* (VS) (DOGAN; ÖLMEZ, 2015a);
- *Modified Vortex Search algorithm* (ModVS) (DOGAN, 2016);

- *Hybrid Vortex Search algorithm* (Pesquisa de vórtices híbrida — HVS) (WANG; WU; WAN, 2017); e,
- *Independent Vortices Search* (IVS).

As configurações das MHs obedeceram os seguintes parâmetros: 5.000.000 de FE como critério de parada do processamento; 50 candidatos por centro, no caso dos algoritmos dos vórtices, e 50 abelhas na ABC (25 fontes de alimentos), o que torna a quantidade de FE por ciclo idêntica no VS e na ABC. Além disso, duas variações de configuração dos vórtices nas MHs ModVs e IVS foram testadas: uma com 5 vórtices — conforme em Dogan (2016) — e outra com 40 vórtices. Testes preliminares indicaram melhores resultados de 20 a 60 vórtices, por isso um valor intermediário de 40 foi adotado.

Tabela 1: Principais características das meta-heurísticas apresentadas

	ABC	VS	ModVS	IVS
<i>Parâmetros</i>	Número de fontes de alimento (ou abelhas); limite de abandono da fonte; ciclos para execução; espaço de busca	Número de candidatos; ciclos para execução; espaço de busca	Número de candidatos; número de vórtices; ciclos para execução; espaço de busca	Número de candidatos; número de vórtices; ciclos para execução; espaço de busca
<i>Aplicação do parâmetro de candidatos/abelhas</i>	50% abelhas trabalhadoras e 50% observadoras	É igual a quantidade de candidatos gerados pela DNM a cada ciclo	É dividida pelo número de vórtices. Mais vórtices, menos candidatos gerados pela DNM por vórtice a cada ciclo	É igual a quantidade de candidatos gerados pela DNM por vórtice a cada ciclo
<i>Geração das soluções iniciais</i>	Aleatória (uniforme)	Pela DNM com a média no centro do espaço de busca	Pela DNM com a média no centro do espaço de busca	Pela DNM com a média em posição aleatória (uniforme) no espaço de busca
<i>Diversidade populacional</i>	Por mutação	Pela DNM	Pela DNM	Pela DNM
<i>Pressão seletiva</i>	De forma probabilística a qualidade da solução	Somente a melhor solução	Somente a melhor solução por vórtice	Somente a melhor solução por vórtice
<i>Refinamento/ investigação da solução</i>	Somente pela mutação	Diminuição do raio de geração dos candidatos	Diminuição do raio de geração dos candidatos em cada vórtice	Diminuição do raio de geração dos candidatos em cada vórtice
<i>Influência entre soluções</i>	Por mutação	Não há	Vórtices são reposicionados baseados no de melhor solução	Vórtices são independentes nas suas buscas
<i>Descarte de soluções por ciclo</i>	Somente se atingido o limite de abandono	Todas exceto a melhor	Todas exceto a melhor em cada vórtice	Todas exceto a melhor em cada vórtice
<i>Quantidade de processos de geração ou mutação por ciclo</i>	As três fases das abelhas	Um pela DNM	Um pela DNM em cada vórtice	Um pela DNM em cada vórtice

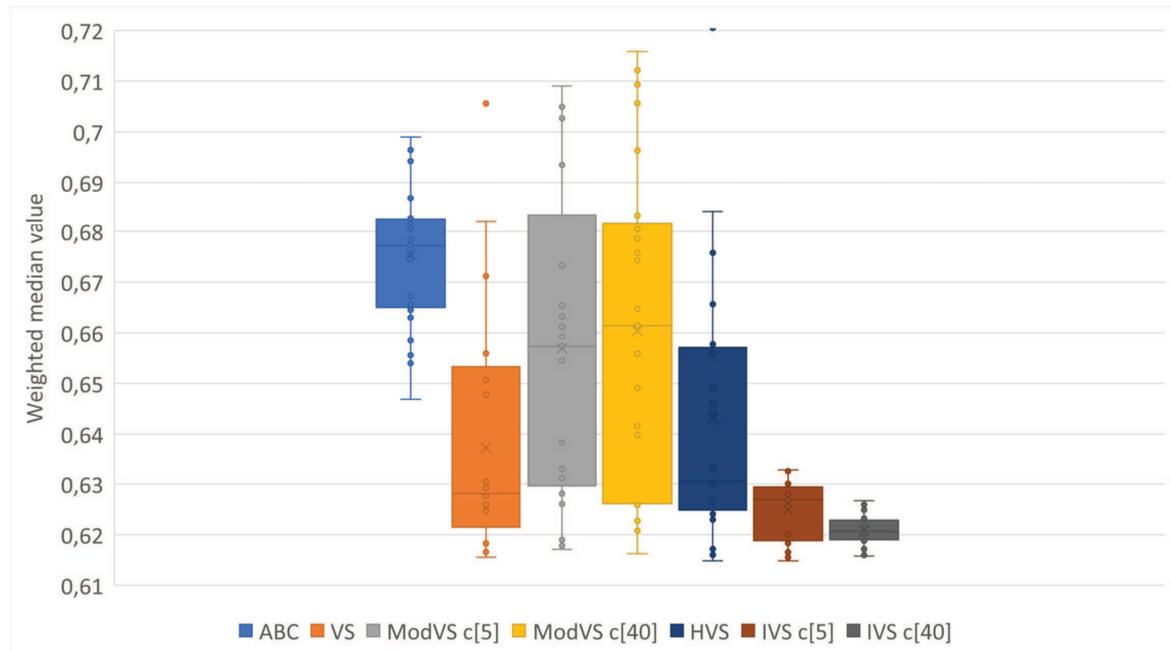
É importante salientar que a contagem de ciclos não foi utilizada como parâmetro de configuração nem como critério de parada para as execuções das MHs. Conforme Mernik et al. (2015), para as comparações entre MHs serem justas, a contagem de ciclos não deve ser usada como critério de parada, pois cada MH tem diferentes estratégias na sua execução podendo usar mais ou menos cálculos de aptidão (FE) em cada ciclo. Usar um número fixo de ciclos como base para interrupção da execução, pode causar grande variação na quantidade total de FE por cada MH, o que seria injusto, pois favoreceria as MHs com mais FE, pois estas, teriam mais oportunidades de testar/verificar suas soluções. Sendo assim, os algoritmos foram adaptados para utilizar a quantidade de FE como critério de interrupção das execuções, permitindo uma comparação e avaliação de desempenho mais adequada na busca por soluções. Apenas para situar melhor o leitor, uma execução com 5.000.000 de FE, equivale a 100.000 ciclos na ABC com 50 abelhas (25 fontes de alimento), 100.000 ciclos no VS com 50 candidatos e 20.000 ciclos no IVS de 5 vórtices e 50 candidatos por centro.

Outro ponto importante, está na implementação do ModVS. Nos experimentos realizados em Dogan (2016), o autor divide o valor total de candidatos pelo totalidade de vórtices. Ou seja, nas execuções com 50 candidatos e 5 vórtices, o ModVS gerou apenas 10 candidatos por vórtice a cada ciclo. Pelo fato deste trabalho utilizar o número de FE como limite de execução das MHs, cada vórtice gerou o número de candidatos estabelecidos pelo parâmetro inicial, sem fazer a divisão pelo número de vórtices. Isto é, para uma execução com 50 candidatos e 5 vórtices, gerou-se 50 candidatos por vórtice a cada ciclo. Esta característica é padrão no IVS — o número de candidatos não é dividido.

Para melhor representação dos resultados, são apresentadas figuras com as soluções em diagramas de caixa (*box-plots*). Cada conjunto de barra, linha e pontos representa uma MH ou uma MH com determinada configuração. Os pontos representam o resultado final de uma das 25 execuções. Quanto mais dispersos maior a variação do resultado. Quanto menor o valor no eixo y, melhor o resultado. A caixa preenchida representa o desvio-padrão das 25 soluções, enquanto a linha interna equivale a posição da mediana e a cruzeta simboliza a média. Na legenda, o valor “c” entre colchetes “c[5]” e “c[40]” representa a quantidade de vórtices, quando aplicável. As Figs. 5, 6, 7 e 8 apresentam os resultados dos experimentos para os Cenários 01, 02, 03 e 04, respectivamente.

Analisando os diagramas, fica clara a vantagem do IVS em relação às outras MHs. A configuração do IVS com 40 vórtices se saiu melhor do que com 5 vórtices, apesar desta última também ter apresentado bons resultados se compara com as demais MHs. O IVS conseguiu chegar a valores mínimos apresentando pouquíssima variação e sem recurso computacional extra. Chama a atenção o ModVS por apresentar desempenho inferior inclusive ao VS original. Isso leva a cogitar que a Eq. 2.27, já questionada neste trabalho, possa ter sido apresentada desta forma por equívoco no respectivo artigo. Porém, até a data deste trabalho, não houve qualquer errata publicada. A estratégia do HVS também não surtiu efeito, tendo soluções muito parecidas com o VS original. Por fim, a ABC apresentou variação menor nos resultados em

Figura 5: Valores encontrados no Cenário de Erros 01 para cada MH. $c[X]$ indica a quantidade de vórtices, quando aplicável.



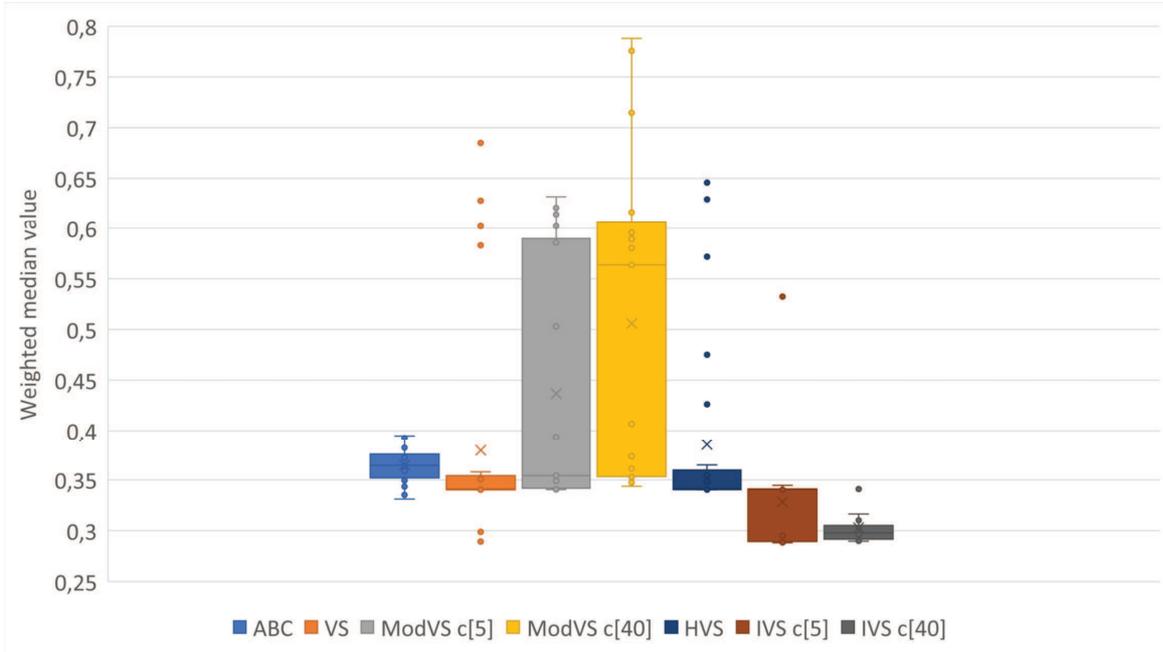
Fonte: Elaborado pelo autor

geral, se comparada as MHs já conhecidas. Isso demonstra uma certa estabilidade nas soluções, superada apenas pelo IVS. Todos esses resultados apontam para a hipótese I — “*O Independent Vortices Search* é mais eficiente que o algoritmo original e que a alteração já proposta em Dogan (2016)” — como sendo verdadeira.

Por último, é possível notar alinhamentos horizontais de pontos (soluções), comuns a várias MHs, identificando mínimos locais em que as MHs ficaram presas durante algumas execuções. Na Fig. 7 é possível notar uma alinhamento entre os valores 2, 4 e 2, 6 do eixo y, para as MHs ModVS e HVS. No mesmo diagrama há outra disposição parecida entre os valores 1, 8 e 2, 0 para todas as MHs, coincidindo com a parte inferior das caixas do diagramas das soluções do VS, ModVS e HVS.

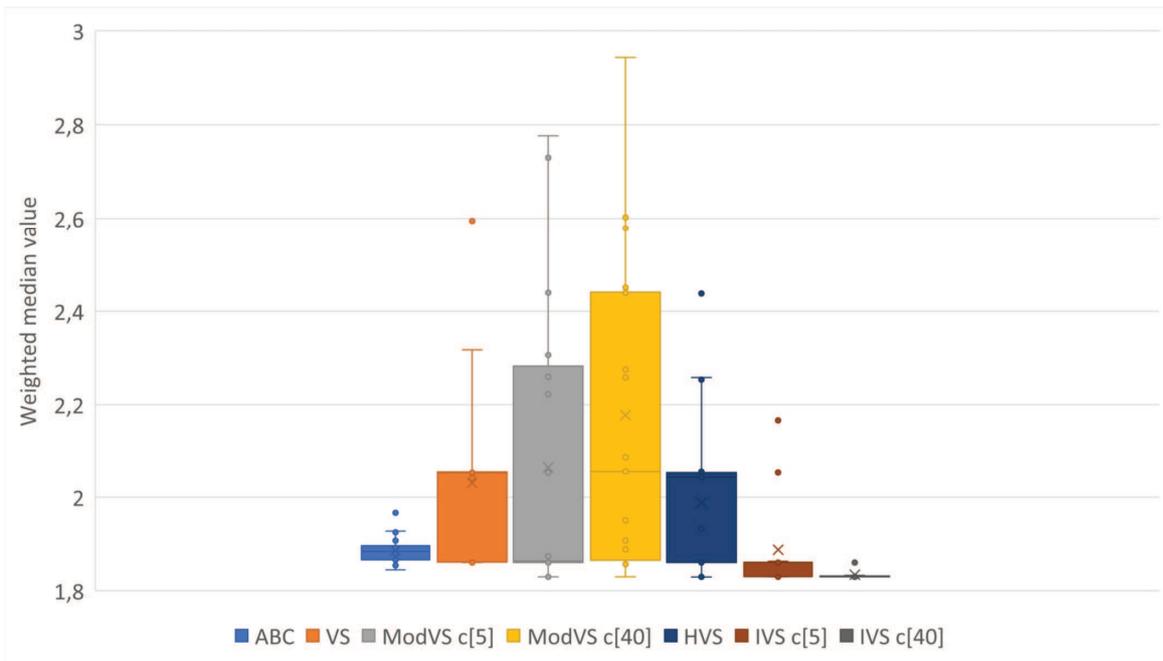
Tendo em vista que o IVS com 40 vórtices é consistente nos resultados, com as melhores soluções e menor grau de variação, ele foi adotado com esta configuração para otimização dos ERs empregados neste estudo. Maiores detalhes serão apresentados no próximo capítulo.

Figura 6: Valores encontrados no Cenário de Erros 02 para cada MH. c[X] indica a quantidade de vórtices, quando aplicável.



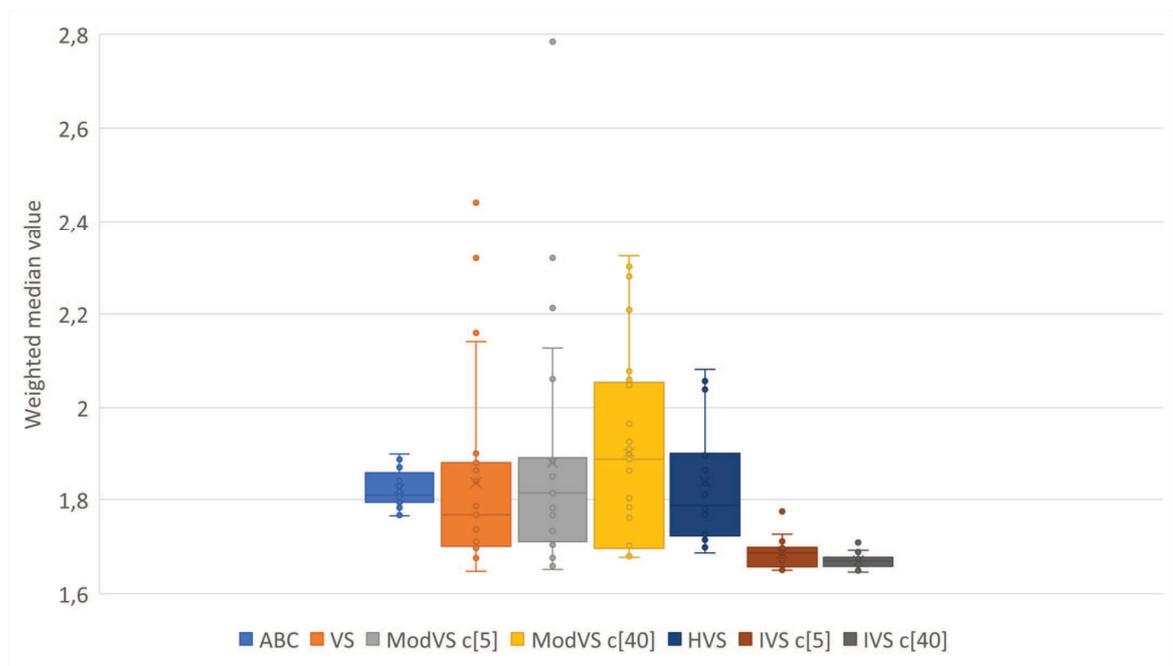
Fonte: Elaborado pelo autor

Figura 7: Valores encontrados no Cenário de Erros 03 para cada MH. c[X] indica a quantidade de vórtices, quando aplicável.



Fonte: Elaborado pelo autor

Figura 8: Valores encontrados no Cenário de Erros 04 para cada MH. c[X] indica a quantidade de vórtices, quando aplicável.



Fonte: Elaborado pelo autor

4 CONTROLE DE QUALIDADE

Neste capítulo, serão apresentados o método de avaliação dos resultados e características de cada um, assim como detalhes da rede utilizada, da simulação dos cenários de erros e do desenvolvimento e implementação para obtenção dos resultados.

4.1 Análise da qualidade da solução

Neste trabalho, a análise da qualidade das soluções de cada Estimador Robusto e do Método dos Mínimos Quadrados, se deu de três formas. Primeiro, quanto a **classificação** da solução em relação à detecção ou não dos *outliers* através da análise do vetor de resíduos. Por segundo, a **quantificação** dos *outliers* detectados, falsos positivos e não identificados. E, por último, a **solução numérica** do estimador em relação às coordenadas oficiais e conhecidas dos vértices.

O primeiro critério — a classificação — foi dividida em seis classes, representadas por letras maiúsculas de “A” a “F”. A definição se dá conforme lista abaixo:

- A: Todos os *outliers* detectados, sem falsos positivos;
- B: Todos os *outliers* detectados, com um ou mais falsos positivos;
- C: Alguns *outliers* detectados, sem falsos positivos;
- D: Alguns *outliers* detectados, com um ou mais falsos positivos;
- E: Nenhum *outlier* detectado, sem falsos positivos;
- F: Nenhum *outlier* detectado, com um ou mais falsos positivos;

Entende-se por falso positivo, a indicação de que uma observação estaria contaminada por um *outlier* quando de fato não houve qualquer contaminação.

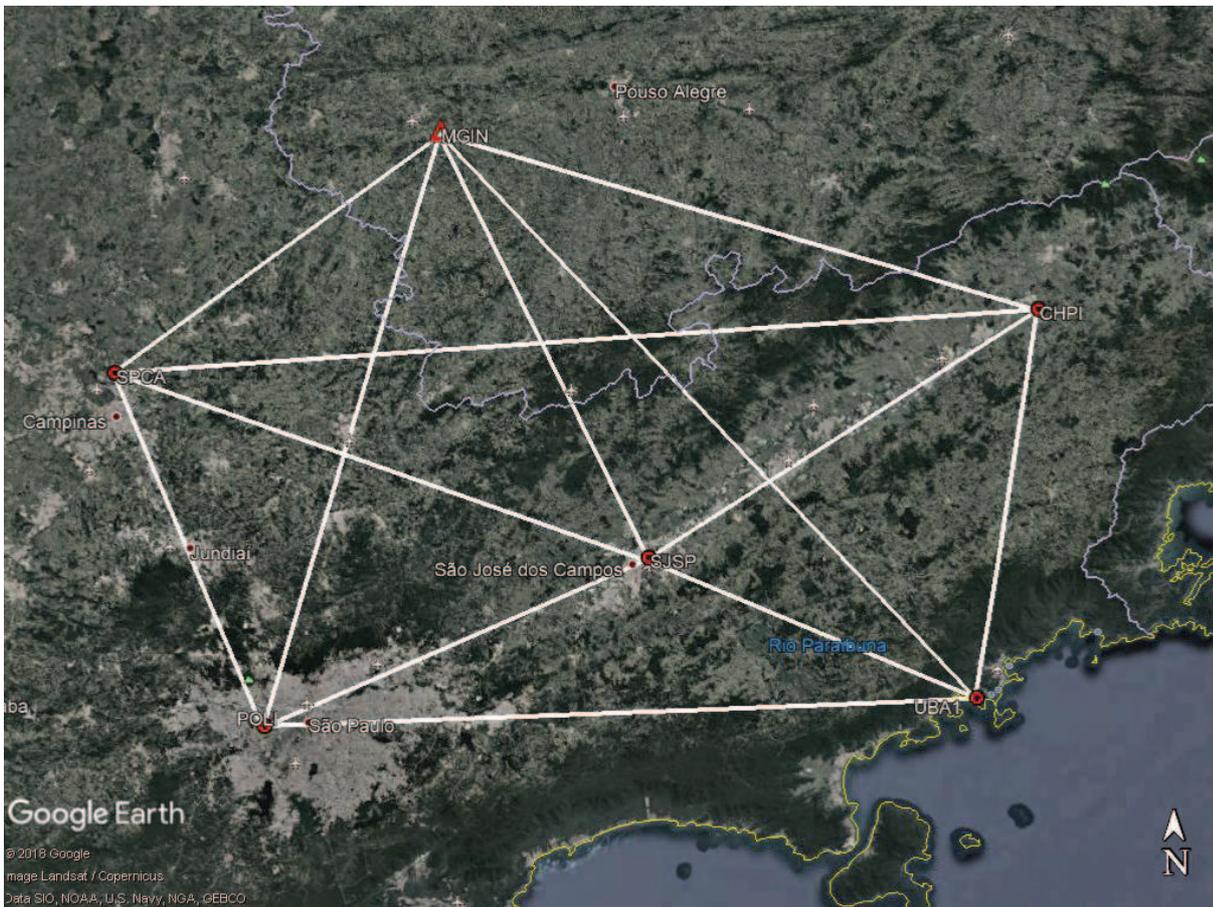
O segundo processo de avaliação — a quantificação — retorna a quantidade de erros grosseiros detectados, de falsos positivos e de *outliers* não identificados. Ela permite verificar a relação de falsos positivos pelo número de *outlier* presentes, assim como outras proporções, como, por exemplo, não identificados pelo total de *outliers* e a quantidade de falsos positivos para cada erro grosseiro detectado.

A terceira e última, é a análise e comparação numérica da solução do estimador em relação às coordenadas oficiais e conhecidas dos vértices. Esta avaliação verificará, se independente da detecção ou não dos *outliers* ou da indicação de falsos positivos, como a solução final dos parâmetros ajustados (\hat{x}) comportou-se em relação às coordenadas oficiais e conhecidas da rede. Isso permite ponderar, por exemplo, se a solução de determinado estimador pode ser utilizada diretamente como resultado final, ou se é recomendável remover as observações indicadas como contaminadas e proceder um novo ajustamento.

4.2 A rede geodésica e a simulação dos cenários de erros

Para a realização dos testes, a rede geodésica utilizada neste estudo é composta por seis vértices da Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS (RBMC), formando uma rede de menor dimensão, conforme Fig. 9. Foram adotadas as coordenadas oficiais das estações e então obtidos os vetores de observação a partir delas (vetores “livres” de erros). Os dados de rastreamento das estações foram processados em *software* de processamento de sinais GNSS para obtenção das soluções e montagem da matriz de variância e covariâncias das observações (Σ_y) permitindo, também, a obtenção da matriz peso (W). Ainda com referência na Σ_y , foram gerados os erros aleatórios, que somados aos vetores contendo os valores oficiais, formaram o vetor l .

Figura 9: Rede com seis vértices da RBMC: MGIN, CHPI, UBA1, POLI, SPCA e SJSP. Um vértice de controle e treze vetores de observação tridimensionais.



Fonte: Google earth V 7.3.2.5491. (13 de dezembro de 2015). São Paulo, Brasil. 23°00'05,70"S, 45°56'46,01"W, Altitude do ponto de visão 386,61 km. Image Landsat / Copernicus, Data SIO, NOAA, U.S. Navy, NGA, GEBCO. <http://www.earth.google.com> [10 de Setembro de 2018]

A rede foi estruturada de maneira a não possuir observações repetidas e com linhas-base independentes entre si. São ao total seis vértices, com um ponto controle e treze vetores. Isso totaliza 15 parâmetros e 39 observações, tendo em vista que os pontos e vetores são tridimensionais.

O modelo adotado para obtenção do vetor de resíduos foi o de Gauss-Markov, usando os ERs LTS, LMS, LTS-RC, SRLS, IRLS e LTA. Conforme visto no capítulo 3, para otimização dos resultados foi aplicado o algoritmo meta-heurístico IVS, com exceção para o ER IRLS que não necessita de algoritmos para otimização por possuir método iterativo próprio.

Na sequência foram conduzidos os ajustamentos em todos os cenários de erros simulados, para cada ER, além do LS (MMQ). Ao todo foram montados nove (9) pacotes com cem (100) cenários contendo erros grosseiros em algumas observações e erros aleatórios em todo o vetor l . Os erros grosseiros foram calculados através do desvio-padrão de cada observação, com magnitude entre $\pm [3\sigma; 6\sigma]$, configurando erros pequenos, e entre $\pm [6\sigma; 12\sigma]$, representando erros grandes, distribuídos aleatoriamente de maneira uniforme. Além disso, os cenários possuem de nenhum (0) até quatro (4) erros grosseiros simultâneos no vetor l . Os pacotes com quatro (4) *outliers* representam os cenários com aproximadamente 10% do total de observações contaminadas.

Os erros aleatórios também são únicos em cada cenário, variando entre $\pm [0\sigma; 3\sigma]$ gerados a partir do desvio-padrão de cada observação usando uma distribuição normal. Em resumo, foram criados os seguintes pacotes de cenários:

- Pacote 01: 100 cenários com erros aleatórios e nenhum *outlier*.
- Pacote 02: 100 cenários com erros aleatórios e 1 *outlier* em cada cenário, com magnitude de $\pm [3\sigma; 6\sigma]$;
- Pacote 03: 100 cenários com erros aleatórios e 1 *outlier* em cada cenário, com magnitude de $\pm [6\sigma; 12\sigma]$;
- Pacote 04: 100 cenários com erros aleatórios e 2 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [3\sigma; 6\sigma]$;
- Pacote 05: 100 cenários com erros aleatórios e 2 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [6\sigma; 12\sigma]$;
- Pacote 06: 100 cenários com erros aleatórios e 3 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [3\sigma; 6\sigma]$;
- Pacote 07: 100 cenários com erros aleatórios e 3 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [6\sigma; 12\sigma]$;
- Pacote 08: 100 cenários com erros aleatórios e 4 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [3\sigma; 6\sigma]$;
- Pacote 09: 100 cenários com erros aleatórios e 4 *outliers* simultâneos em cada cenário, com magnitudes de $\pm [6\sigma; 12\sigma]$;

Todos os cenários e pacotes também tiveram suas soluções estimadas pelo MMQ, seguindo a proposta de análise da qualidade das soluções exposta no início do capítulo.

4.3 Desenvolvimento e implementação

O desenvolvimento e implementação do sistema de ajustamento com uso de ERs e MH calcula as soluções conforme fluxograma indicado na Fig. 10 e foi desenvolvido em linguagem *Python*, versão 3.6. O fluxograma não contempla a configuração e escolha da rede geodésica, nem os erros inseridos no vetor de observações, que são uma etapa anterior (ver seção 4.2). O objetivo é apresentar o fluxo de trabalho do algoritmo na obtenção das soluções aplicando MH, Gauss-Markov e ER.

Por fim, os parâmetros de controle do IVS, foram definidos com os seguintes valores:

- Número de candidatos: 50 soluções por vórtice. Valor também adotado em Dogan e Ölmez (2015a,b); Dogan (2016) e no capítulo 3;
- Número de vórtices: 40 vórtices, conforme experimentos apresentados no capítulo 3;
- Limite de execução do algoritmo: o limite de execução foi definido pela quantidade de *fitness evaluations* (FE) — avaliações de aptidão — das soluções. Para cada execução, a MH realizou vinte e cinco milhões (25.000.000) de FE. Ao atingir este limite, a execução foi encerrada independentemente do resultado encontrado, para assim permitir uma melhor comparação entre os ERs;
- Espaço de busca: cada parâmetro j teve seu espaço de procura definido por $x_{min,j}$ e $x_{max,j}$. Estes limites foram computados utilizando o vetor de parâmetros \mathbf{x} obtido por MMQ, o maior resíduo absoluto $\max\{|\mathbf{v}|\}$ da solução por MMQ, a maior fração de absorção $\max\{diag(\mathbf{R})\}$ e uma margem de segurança de 15%, conforme Eqs. 4.1

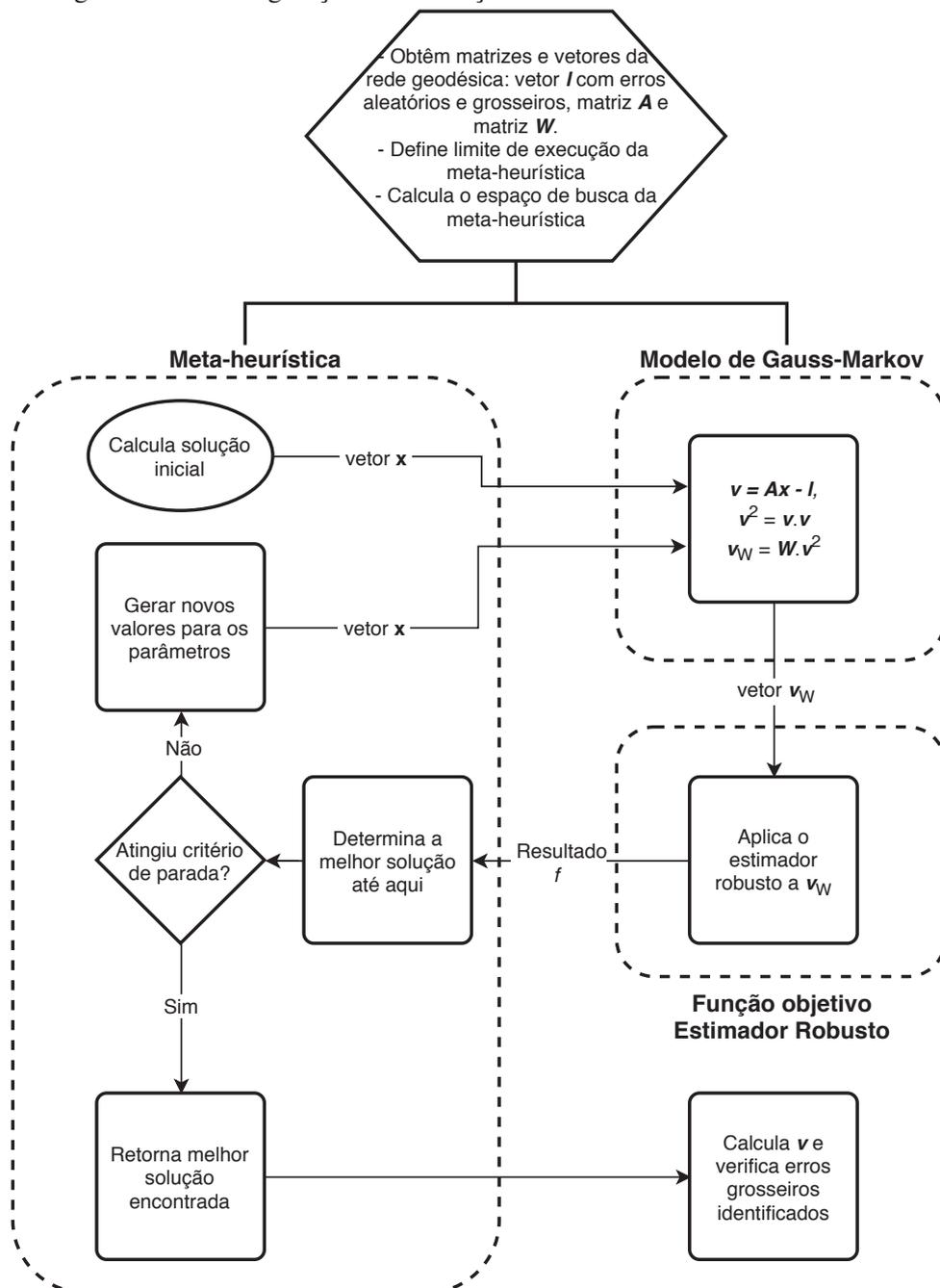
$$\begin{cases} x_{min,j} = x_j - \max\{|\mathbf{v}|\} * \max\{diag(\mathbf{R})\} * 1,15; \\ x_{max,j} = x_j + \max\{|\mathbf{v}|\} * \max\{diag(\mathbf{R})\} * 1,15. \end{cases} \quad (4.1)$$

Onde \mathbf{R} é o produto entre a matriz de covariância dos resíduos ajustados $\Sigma_{\hat{v}}$ e a matriz peso \mathbf{W} , conforme definido pela Eq. 4.2

$$\mathbf{R} = \frac{1}{\sigma^2} \Sigma_{\hat{v}} \mathbf{W} \quad (4.2)$$

A Eq. 4.2, apresenta um método inédito para definição do espaço de busca que pode ser aplicado a qualquer problema que faça uso do modelo de Gauss-Markov, não se limitando a redes geodésicas. Aplicando esta estratégia para todos os 900 cenários, nenhuma solução ficou fora do espaço de busca do IVS, ao mesmo tempo que ele foi reduzido significativamente. No capítulo seguinte, veremos os resultados obtidos.

Figura 10: Fluxograma de execução do sistema de ajustamento, uma vez tendo definidas todas as informações da rede geodésica e configurações de execução.



Fonte: Elaborado pelo autor

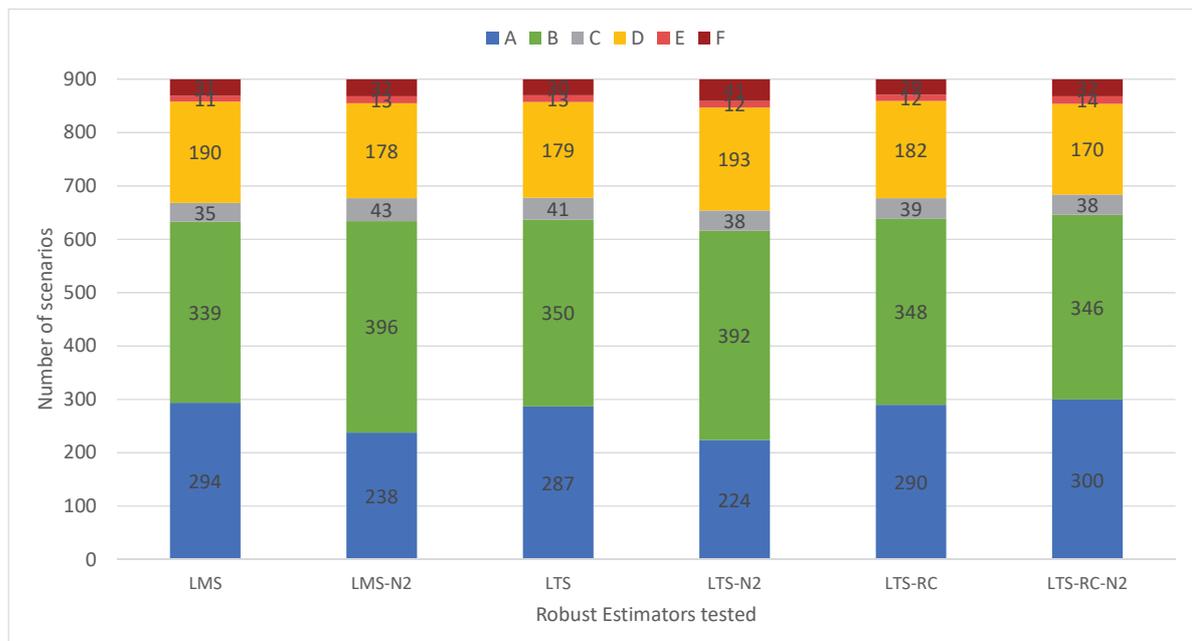
5 RESULTADOS E DISCUSSÕES

Este capítulo traz os resultados dos experimentos divididos em duas partes. Primeiro, a comparação dos resultados obtidos com dois limites de corte diferentes para o LTS e LTS-RC e duas posições distintas para a mediana no LMS. Por último, são apresentados os resultados dos ajustamentos com todos os ERs e feitas as devidas comparações e análises.

5.1 Comparação do limite de corte (LTS e LTS-RC) e posição da mediana (LMS)

Os testes com os dois valores para os limites foi realizada em todos os pacotes de cenários e a análise das soluções seguiu a abordagem descrita no capítulo 4. A partir daqui, o limite menor, $h = (n + 1) / 2$, será indicado no texto como h_{n2} e o limite maior, $h = (n + p + 1) / 2$, como h_{np2} . Para os ERs LTS e LTS-RC, um limite menor indica que menos resíduos serão levados em consideração na estimativa, ou seja, mais valores serão ignorados no vetor de resíduos. Enquanto que com um limite maior, mais valores serão somados para a estimativa, ou seja, menos resíduos serão descartados. Para o LMS, a ideia do limite define a posição da mediana. Com o limite maior, a localização do resíduo que será considerado a mediana, estará mais afastada do centro em direção aos valores maiores. Ao passo que com o limite menor, a mediana será considerada mais ao centro do vetor de resíduos.

Figura 11: Ocorrência de cada classificação nos 900 cenários para os três estimadores e sua variação na posição de corte ou mediana. O sufixo “-N2” indica o limite em h_{n2} para o respectivo estimador.



Fonte: Elaborado pelo autor

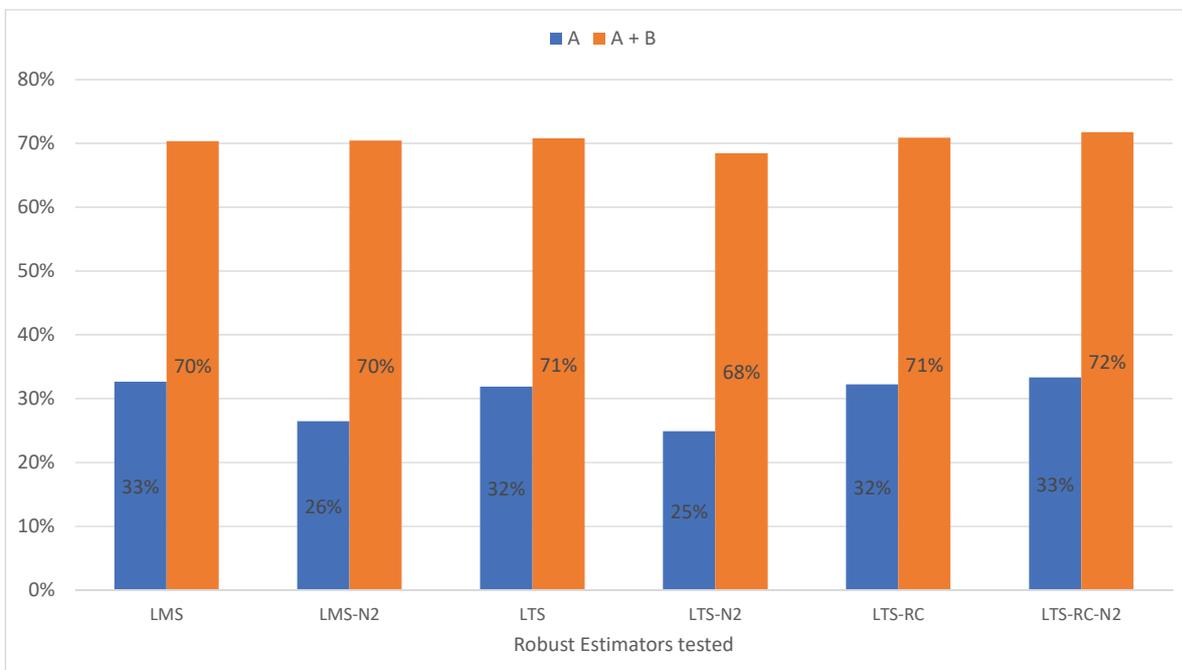
A fig. 11, apresenta a ocorrência de cada classificação nos 900 cenários para os três estimadores e sua variação na posição de corte ou mediana. O sufixo “-N2” indica o limite em h_{n2}

para o respectivo estimador e os ERs sem sufixo tem como limite h_{np2} .

No caso do LMS e do LTS, os ajustamentos com o limite em h_{n2} , tem resultados piores na classificação. Classificações do tipo “A” reduzem significativamente ao adotar o limite esperado como “mais robusto” e aumentam as classificações do tipo “B”. Ou seja, para estes estimadores, o limite menor, tende a manter identificação dos *outliers*, porém, aponta mais falsos positivos que, em verdade, seriam observações boas, não contaminadas. Por outro lado, os resultados para o LTS-RC, apresentam uma ligeira melhora para o limite em h_{n2} , analisando apenas pela classificação das soluções.

Considerando somente a porcentagem de ocorrência das classificações do tipo “A” e, “A” com “B”, conforme fig. 12, que representam a identificação de todos os *outliers*, mesmo que com falsos positivos no caso “A” com “B”, verifica-se que, somadas as duas classificações, os ERs apresentam um comportamento muito similar. A partir disso é possível concluir que a posição da mediana ou corte, seja em h_{n2} ou h_{np2} , não aumenta a quantidade de erros identificados, apenas o diminui número de falsos positivos no caso de h_{np2} . Em uma situação real, isso não é o ideal, uma vez que observações boas estariam sendo apontadas como ruins podendo levar ao descarte das mesmas e consequente piora nos resultados. Novamente, a exceção fica por conta do LTS-RC, que mantém uma solução estável em ambas as situações, com ligeira melhora para o limite em h_{n2} .

Figura 12: Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, comparando os ERs para os valores h_{n2} e h_{np2} . O sufixo “-N2” indica o limite em h_{n2} para o respectivo estimador.



Fonte: Elaborado pelo autor

Ainda referente a fig. 12, pode-se proceder uma análise entre as diferenças do LTS com e sem controle de redundância (CR), indicado pelo sufixo “-RC”. Conforme exposto no capí-

tulo 2, o CR garante que cada parâmetro continue associado a pelo menos duas observação presentes na soma, evitando perda de redundância nas observações no momento da eliminação dos resíduos. Ao comparar o LTS com o LTS-RC, vemos que os resultados apresentam uma quantidade classificações idênticas, enquanto as variações com h_{n2} , apresentam diferença significativa. A explicação para tal fenômeno foi observada já durante a execução dos testes. Para a rede testada neste trabalho, o CR raramente era ativado no caso do LTS-RC com limite em h_{np2} . Isso se deve ao fato do limite maior remover uma quantidade muito menor de resíduos, raramente quebrando a redundância. E, tendo em vista que a rede possui várias observações entre os seus pontos tridimensionais (apesar de não serem repetidas), dificilmente algum parâmetro ficava sem ao menos duas observações. Ao utilizar o limite em h_{n2} , essa redundância começou a ser quebrada pelo LTS sem CR, enquanto o LTS-RC manteve os parâmetros associados com ao menos duas observações. Deste modo, o LTS-RC conseguiu um resultado muito mais satisfatório.

A figura 13, traz a quantificação de *outliers* identificados, falsos e não detectados, com o total de erros gerados nos 900 cenários (2000 *outliers*). A figura confirma o aumento de falsos positivos com o limite em h_{n2} para o LMS e LTS, e a estabilidade com ligeira melhora no LTS-RC. Baseado nestes dados, a figura 14, traz a diferença percentual para cada estimador divididas em erros identificados, falsos e não detectados, em cada limite. Novamente é possível constatar, de forma mais clara, o aumento de falsos positivos em 22% e 18,3% para o LMS e LTS, respectivamente, quando passam a utilizar o limite em h_{n2} em substituição ao h_{np2} . Também é possível verificar que não há diferenças significativas na identificação de *outliers* entre os diferentes limites, assim como nos casos de erros não detectados, onde apenas o LTS expôs leve uma piora de 6,6%.

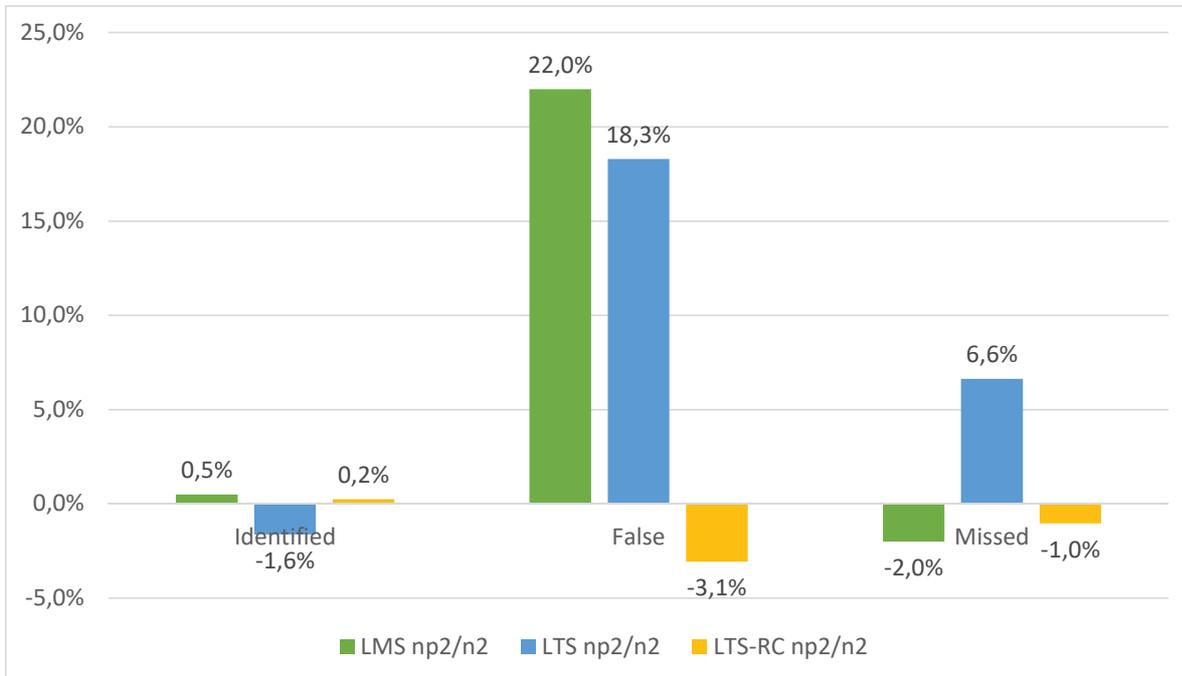
Figura 13: Quantidade de *outliers* identificados, falsos e não detectados comparando os ERs para h_{n2} e h_{np2} . O sufixo “-N2” indica o limite em h_{n2} para o respectivo estimador.



Fonte: Elaborado pelo autor

Por último, avaliou-se o afastamento das coordenadas dos valores oficiais e conhecidos das estações. A fig. 15 apresenta em gráfico e tabela, a diferença do afastamento médio das coor-

Figura 14: Diferença percentual para cada estimador divididas em erros identificados, falsos e não detectados, para os limites em h_{n2} e h_{np2} . Percentual positivo, indica maior ocorrência com h_{n2} , enquanto um percentual negativo, indicada menor ocorrência ao adotar o limite em h_{n2} .



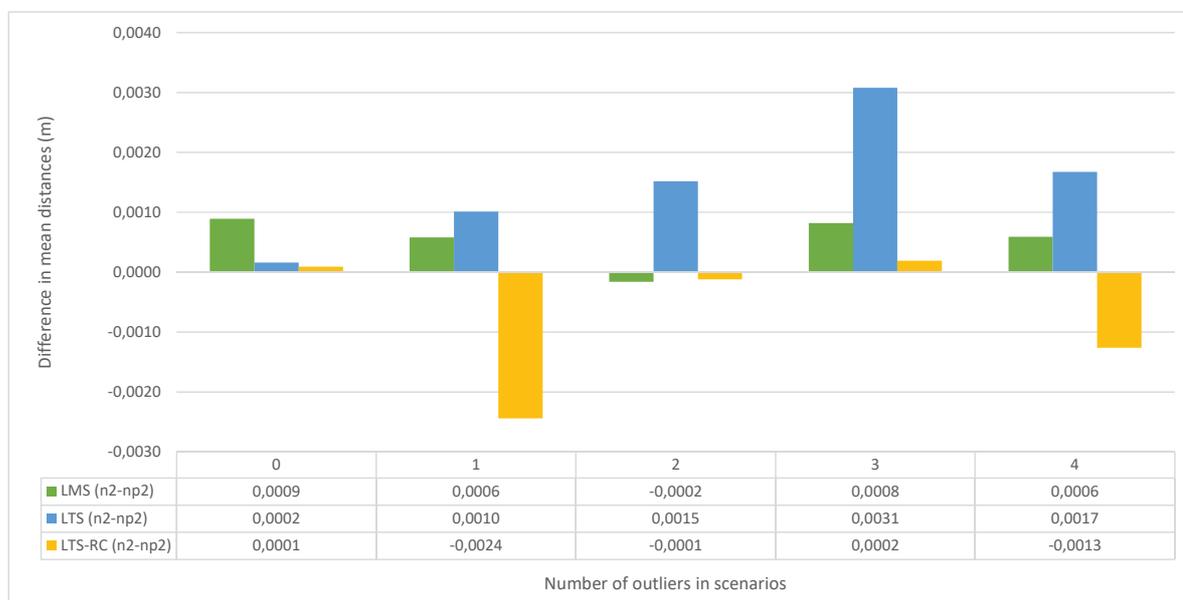
Fonte: Elaborado pelo autor

denadas em cada pacote de cenários, entre os limites h_{n2} e h_{np2} ($h_{n2} - h_{np2}$) para cada ER. As diferenças foram divididas por pacotes com quantidade de *outliers* distintos, desde nenhum (0) até quatro (4) erros simultâneos. Percebe-se que as diferenças médias são mínimas na maioria dos casos, ficando abaixo de 1 mm, seja com h_{n2} ou h_{np2} . De modo geral, o LTS apresenta piores nos afastamentos das coordenadas oficiais utilizando h_{n2} , enquanto o LTS-RC consegue diminuir a diferença em 2 casos de *outliers*: 1 e 4 erros simultâneos. O LMS ainda apresenta uma leve piora com a posição da mediana em h_{n2} , mas nada que possa ser considerado significativo.

Em suma, trabalhar com o limite maior (h_{np2}) garante que menos falsos positivos sejam apontados, sem perda na detecção dos *outliers* quando aplicados o LMS ou LTS. Caso se utilize o limite menor (h_{n2}), no entanto, é recomendável aplicar o LTS-RC devido ao algoritmo de garantia da redundância nas observações. Neste caso, conforme exposto nos experimentos, o limite em h_{n2} ainda apresentará uma pequena melhora em relação ao h_{np2} e ao LTS tradicional. Esta característica se mostra interessante para eventuais problemas onde o número de observações por parâmetro for mais limitado e/ou o limite de corte for menor.

A seção seguinte apresentará os resultados para todos ERs estudados neste trabalho, mantendo, também, a representação dos diferentes limites dos ERs abordados nesta seção.

Figura 15: Diferença do afastamento médio das coordenadas, entre os limites h_{n2} e h_{np2} , em pacotes de cenários de 0 a 4 *outliers*. Valores negativos indicam menores afastamentos com o limite em h_{n2} enquanto valores positivos correspondem a um maior deslocamento em relação as coordenadas oficiais com o limite em h_{n2} .



Fonte: Elaborado pelo autor

5.2 Resultados dos ajustamentos com ERs

Os experimentos foram conduzidos aplicando o IVS — exceto para o MMQ (LS) e IRLS — seguindo a abordagem descrita no capítulo 4. Para melhor análise dos resultados, esta seção foi dividida em três subseções: a primeira, abordará o caso de nenhum *outlier* (Pacote 01), ou seja, as observações destes 100 cenários não contiveram qualquer erro grosseiro, somente os erros aleatórios simulados. A subseção seguinte, abordará os cenários com *outliers* de pequena magnitude (Pacotes 02, 04, 06 e 08). Cada pacote apresentará quantidades de erros grosseiros diferentes, de 1 a 4 *outliers* simultâneos por cenário, além dos erros aleatórios. Por último, seguindo a mesma ideia, serão apresentados os cenários com *outliers* de grande magnitude (Pacotes 03, 05, 07 e 09). Desta forma, a segmentação dos resultados permitirá uma melhor análise do comportamento de cada ER. Também, as variações nos limites de h , expostas na seção anterior, continuarão sendo apresentadas nos resultados que seguem.

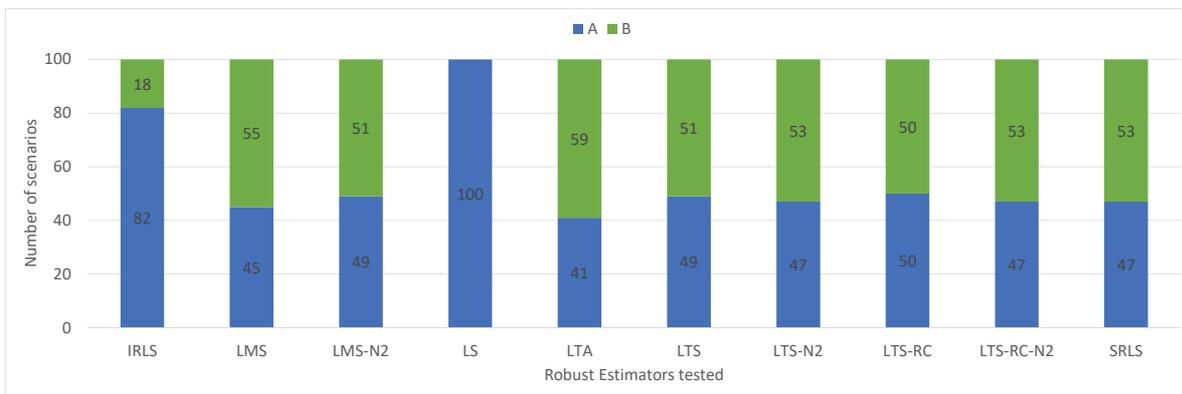
5.2.1 O caso de nenhum *outlier*

Para o pacote de cenários 01, sem qualquer *outlier*, os resultados confirmaram as afirmações presentes na literatura e já expostas neste trabalho. O Método dos Mínimos Quadrados (MMQ) (ou *Least Squares method* — LS), é o melhor estimador imparcial quando não há presença de

erros grosseiros nas observações. O MMQ é apresentado como LS nas figuras que seguem.

Conforme classificação apresentada na Fig. 16, o LS demonstra um desempenho excelente, pois não apontou qualquer *outlier*, ou seja, falso positivo, nos 100 cenários, obtendo classificação “A” em todos. Cabe destacar também os resultados do IRLS, que obteve 82 classificações do tipo “A”, restando apenas 18 para o tipo “B” — um ou mais falsos positivos por cenário. Demais ERs, apresentaram resultados semelhantes entre si, com 50% das classificações em “A” e 50% em “B”.

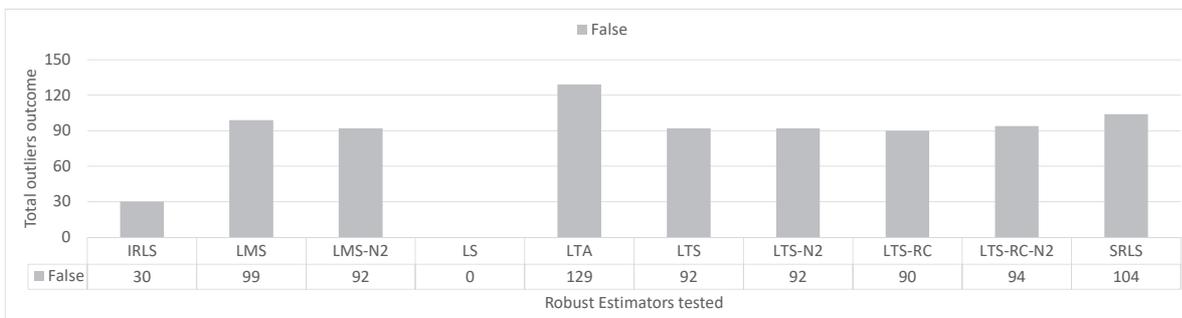
Figura 16: Contagem das classificações “A” e “B” obtidas nos 100 cenários sem *outliers* para todos os estimadores.



Fonte: Elaborado pelo autor

Na fig. 17 temos a contagem de falsos positivos para cada estimador nos 100 cenários do pacote 01. Como visto, o LS não apresentou qualquer falso, contrastando com os demais estimadores, principalmente com o LTA e o SRLS que, em média, apontaram mais de um falso positivo por cenário. Demais estimadores apresentam valores parecidos, com uma ocorrência de 9 até 10 falsos positivos a cada 10 cenários.

Figura 17: Quantidade de falsos positivos para os cenários sem *outliers* do pacote 01, agrupados por estimador.

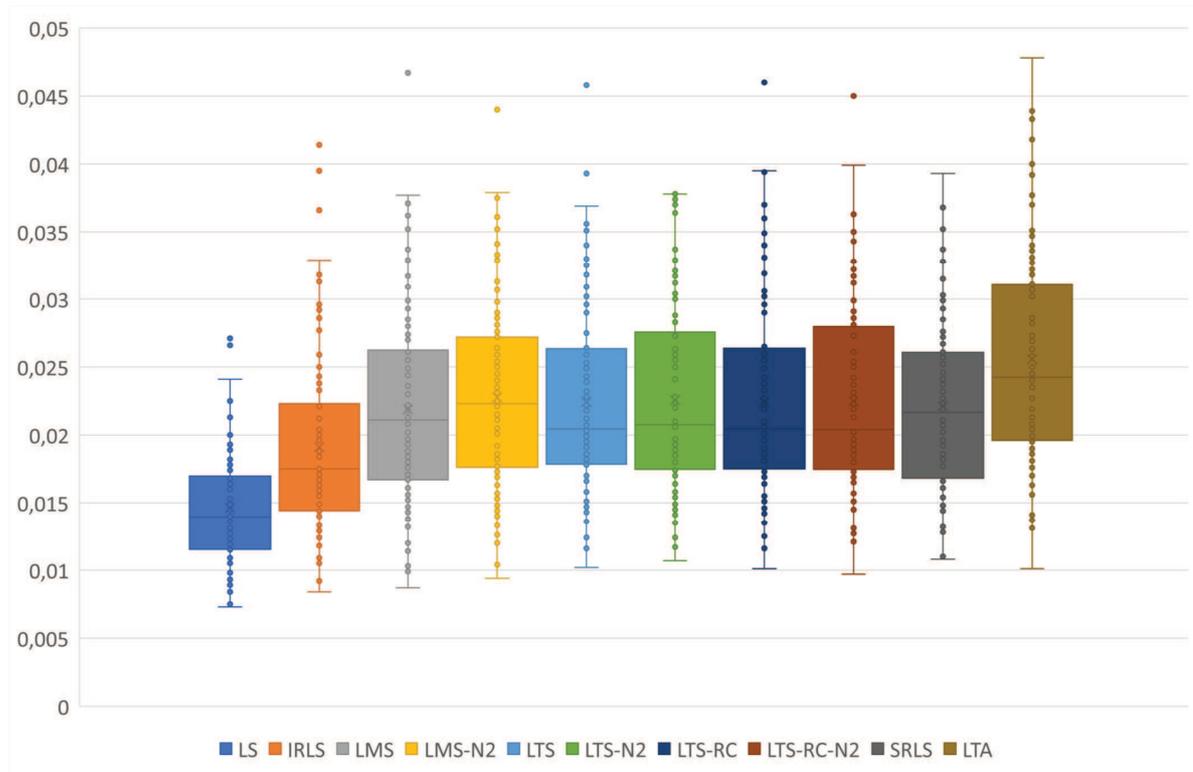


Fonte: Elaborado pelo autor

Concluindo a análise dos resultados, elaborou-se um diagrama de caixa, apresentado na Fig. 18. Cada ponto representa o afastamento médio das coordenadas estimadas de um dado cenário,

em relação aos valores oficiais das estações na rede. O limite superior da caixa indica o terceiro quartil e a borda inferior o primeiro quartil. Os fios superiores e inferiores representam os limites com base na amplitude interquartil. Pontos além dos fios são considerados discrepantes (*outliers* ou exceções). Quanto menor o tamanho da caixa e mais próxima de 0, melhor o resultado, pois indica que o afastamento dos valores foi menor e portanto a estimativa é mais segura e estável. É relevante destacar que os valores de afastamento médio não chegarão a zero, pois mesmo sem qualquer *outlier*, todas as observações tiveram erros aleatórios adicionados aos seus vetores livres de erro, objetivando assim, uma situação mais próxima da realidade.

Figura 18: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários sem *outliers*



Fonte: Elaborado pelo autor

Novamente, nota-se que o LS superou todos os demais estimadores, apresentando um afastamento consideravelmente menor e com uma variação mais baixa entre os cenários. Conforme exposto pela literatura clássica e confirmado pelos experimentos, o LS é superior a qualquer outro ER testado quando não há presença de erros grosseiros. Além desta vantagem, o LS não necessita de uma meta-heurística (MH) para estimativa dos valores, pois tem um equacionamento definido e de baixo custo computacional, se comparado com soluções que necessitam de MHs.

Levando em consideração que a aplicação da estratégia apresentada neste trabalho fosse para, apenas, a identificação de *outliers* e não a estimativa final em si, a alta quantidade de

falsos positivos nos demais estimadores não é ideal, uma vez que observações boas (não contaminadas) seriam descartadas.

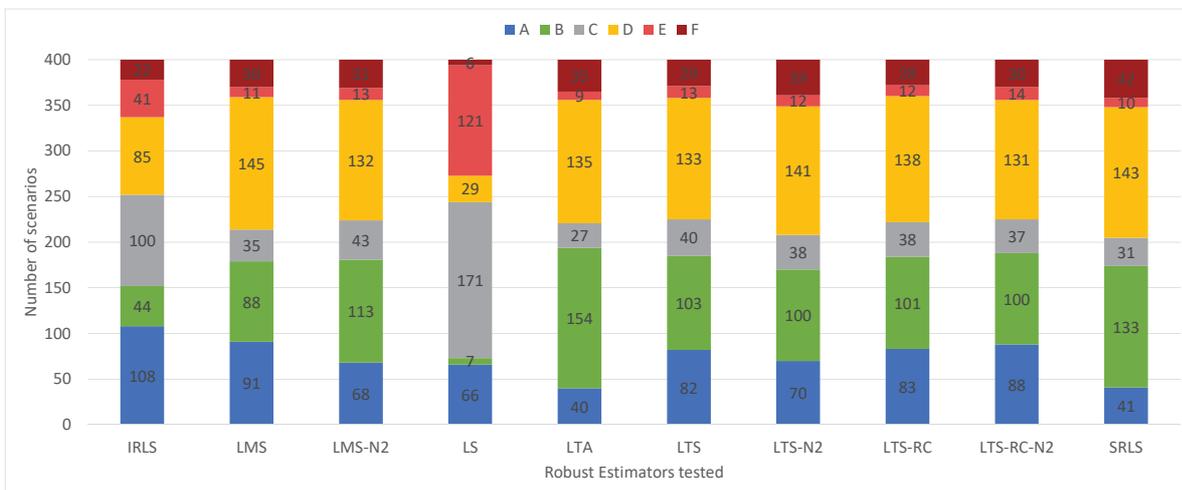
Concernindo a aplicabilidade como estimativa final das coordenadas, os ERs também deixam a desejar por apresentarem maiores afastamentos das coordenadas reais e conhecidas. Qualquer que seja o caso, o LS clássico se apresenta como a melhor alternativa para situações em que não há presença de erros grosseiros.

5.2.2 Os casos com *outliers* de pequena magnitude

Conforme visto no capítulo 2, aplicações e experimentos reais estão suscetíveis a contaminação por *outliers*. Esta subseção do trabalho abordar-se-á os casos com erros grosseiros de menor magnitude, variando desde o limite dos erros aleatórios em 3σ , até 6σ , configurando assim, o intervalo de $[3\sigma; 6\sigma)$. Ao todo foram executados 400 cenários (Pacotes 02, 04, 06 e 08) variando de 1 a 4 erros grosseiros simultâneos por cenário.

Partindo da classificações das soluções, a Fig. 19 traz a ocorrência de cada categoria nos 400 cenários. No geral, percebe-se que a maioria dos ERs não divergem muito nas classificações das soluções.

Figura 19: Contagem das classificações de “A” até “F” obtidas nos 400 cenários contendo de 1 a 4 *outliers* de menor magnitude, organizados por estimador.



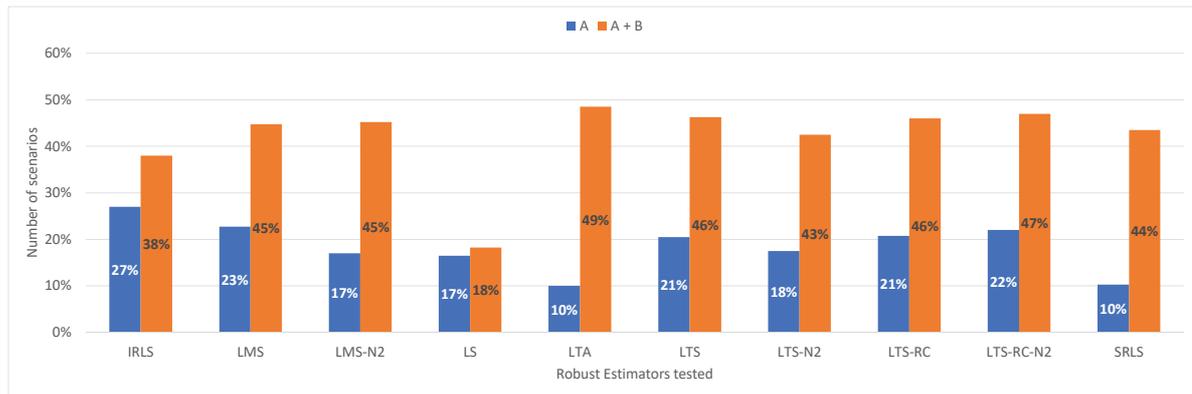
Fonte: Elaborado pelo autor

A maior discrepância fica por conta do LS, que apresenta suas soluções concentradas nas classificações tipos “A”, “C” e “E”. Estas categorias são aquelas que não apresentam qualquer falso positivo na solução do vetor de resíduos, identificando ou não o(s) erro(s) grosseiro(s) presente(s).

Outro destaque fica por conta do IRLS, que obteve o maior número de soluções classificadas como do tipo “A” — todos os *outliers* detectados, sem falsos positivos. Porém, conforme fig. 20, obteve apenas 38% das soluções com identificação de todos os erros grosseiros (“A” + “B”).

Nestes casos, os ERs com melhor desempenho na identificação ficam por conta do LMS, LTS e LTS-RC-N2 que obtiveram de 21 a 23% das classificações do tipo “A” e de 45 a 47% das soluções categorizadas como tipo “A” ou “B”.

Figura 20: Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, para cada estimador, nos cenários com *outliers* de menor magnitude.



Fonte: Elaborado pelo autor

Apesar do LTA ter apresentado 49% das classificações como tipo “A” ou “B”, este estimador obteve o menor número de soluções identificadas como tipo “A”, junto com o SRLS. Ou seja, o LTA apresentou muitos falsos positivos no vetor de resíduos das soluções. Isto é confirmado pela fig. 21 que traz a quantidade total de *outliers* gerados (1000) e a contagem de identificados, falsos e não constatados para cada estimador. Sendo assim, dependendo da aplicação, o LTA apresenta a solução menos aceitável por conter muitos falsos (1303), mesmo tendo o maior número de erros grosseiros identificados (710) entre todos os estimadores.

Figura 21: Quantidade de *outliers* identificados, falsos e não constatados para cada estimador, nos 400 cenários com erros grosseiros de menor magnitude.



Fonte: Elaborado pelo autor

Ainda na quantificação das detecções, o LS se confirma como o estimador com menor número de falsos positivos (43), conforme esperado, contrastando com o LTA que apresenta 13 falsos para cada 10 *outliers*. Porém, o LS também foi o estimador que mais deixou de detectar os erros grosseiros, perdendo 572 dos 1000 erros inseridos. O IRLS, apresentou uma solução mais intermediária, trazendo um bom número de detecções, menos falsos, porém o segundo pior em não detectar *outliers*. Entre os ERs que usam de meta-heurística, o LMS se saiu melhor identificando boa parte dos *outliers* sem tantos falsos positivos.

Para uma análise objetivando a adoção da solução final destes estimadores no cálculo dos parâmetros, e não apenas para a detecção de *outliers*, a Tab. 2, traz uma tabela com o afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos. Também foi adicionada a solução do LS para os mesmos cenários, sem a adição dos *outliers*, estabelecendo um ponto de partida para melhor comparação, na medida em que é possível verificar a influência dos erros aleatórios. Sabendo que o LS apresenta a melhor estimativa em situações sem erros grosseiros, isso permite uma melhor visualização do afastamento, não só em relação às coordenadas oficiais, mas também comparado a uma situação real com presença de erros aleatórios.

Tabela 2: Afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos de pequena magnitude. “LS (no outliers)”, indicada a solução por MMQ, sem a adição dos *outliers*.

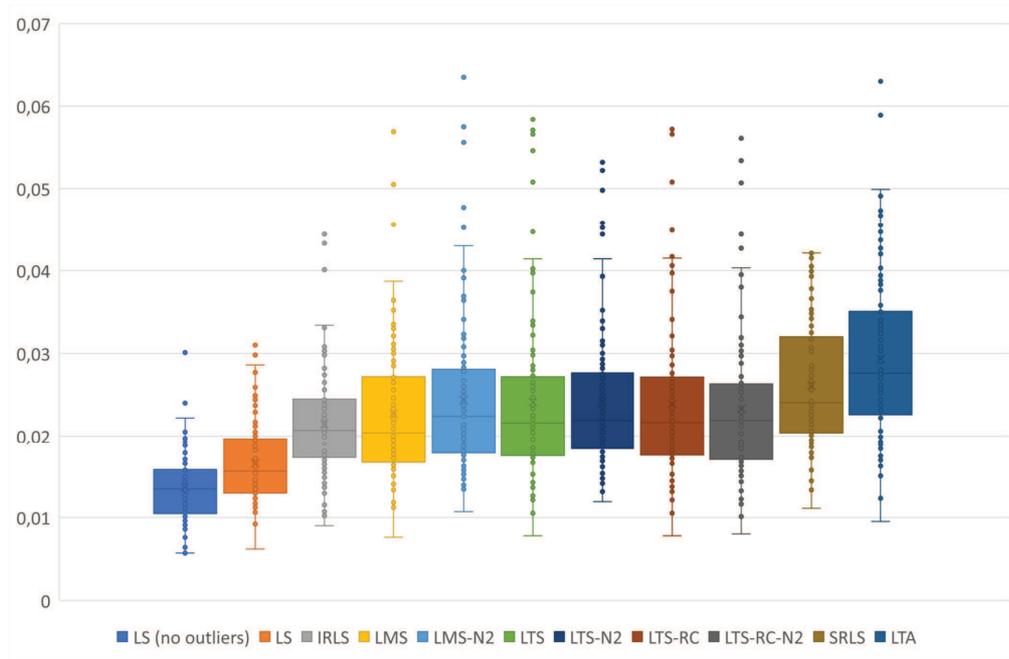
Mean distance from original coordinates for all scenarios (in meters)											
	LS (no outliers)	LS	LMS	LTS	LTS-RC	SRLS	IRLS	LTS-RC-N2	LMS-N2	LTS-N2	LTA
1 outlier	0,014	0,017	0,023	0,024	0,024	0,026	0,024	0,021	0,023	0,024	0,029
2 outliers	0,014	0,020	0,025	0,027	0,027	0,028	0,027	0,026	0,025	0,027	0,032
3 outliers	0,014	0,022	0,027	0,027	0,027	0,030	0,025	0,027	0,028	0,028	0,035
4 outliers	0,014	0,025	0,029	0,029	0,030	0,031	0,024	0,030	0,029	0,031	0,036

Fonte: Elaborado pelo autor

Conforme dados na tabela da Tab. 2, o LS apresenta os menores afastamentos, e consequentemente seria a melhor estimativa dos parâmetros para o caso de adotar a solução diretamente, sem eliminar observações contaminadas e proceder novos ajustamentos. Mesmo nos cenários com 4 *outliers* o LS tem uma solução melhor que os ERs que fazem uso de MH, ainda que nestes cenários a diferença seja menor em relação a estes métodos. Em geral, os demais estimadores apresentam soluções parecidas, com destaque para o IRLS, método que não faz uso de MH e que apresentou uma baixa variação na solução mesmo com o aumento progressivo da quantidade de *outliers*. Apesar da pequena diferença nos afastamentos entre os estimadores, o LTA contrastou com as demais soluções, se saindo o pior em todas as situações, desde 1 até 4 erros grosseiros simultâneos.

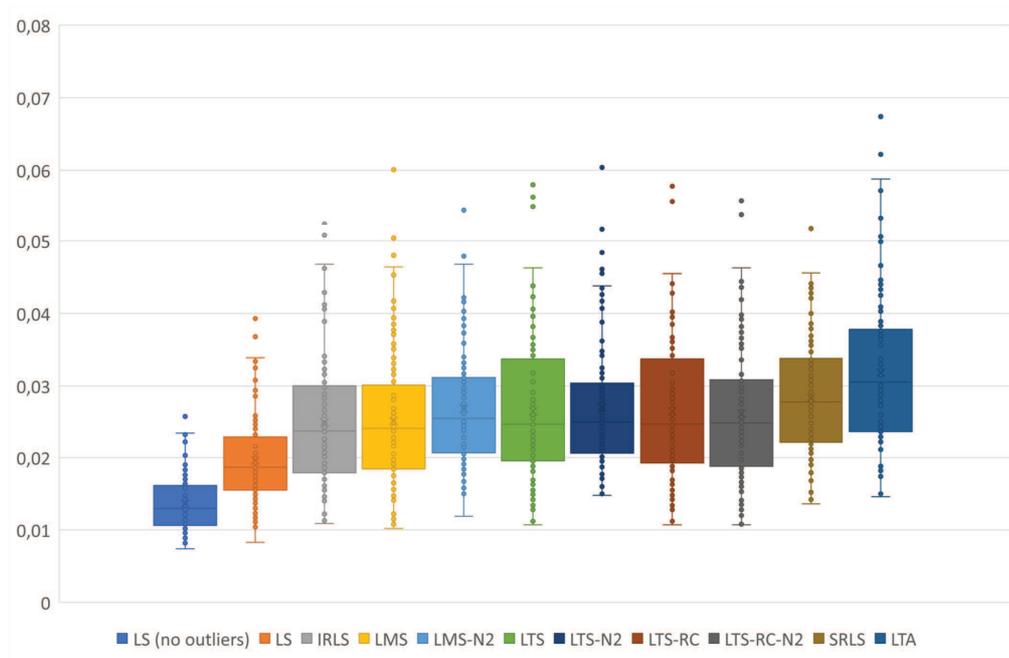
Por fim, elaborou-se quatro diagramas de caixa, apresentados nas Figs. 22, 23, 24 e 25. Cada ponto representa o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede. Também foram adicionadas as soluções do LS para os mesmos cenários sem a adição de *outliers*.

Figura 22: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 01 *outlier* de pequena magnitude.



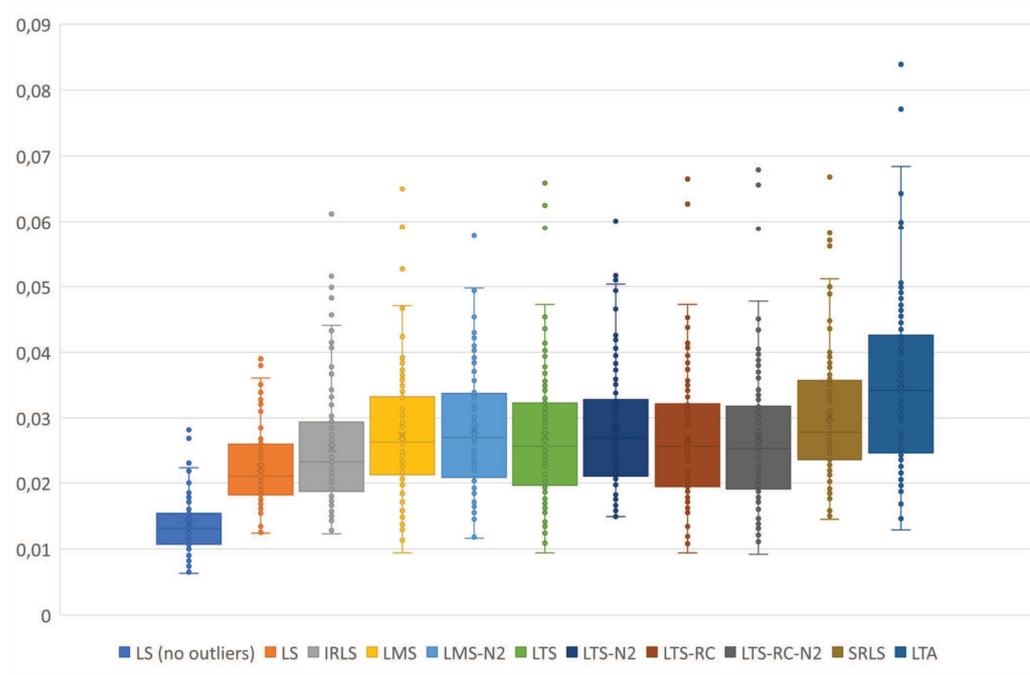
Fonte: Elaborado pelo autor

Figura 23: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 02 *outliers* de pequena magnitude.



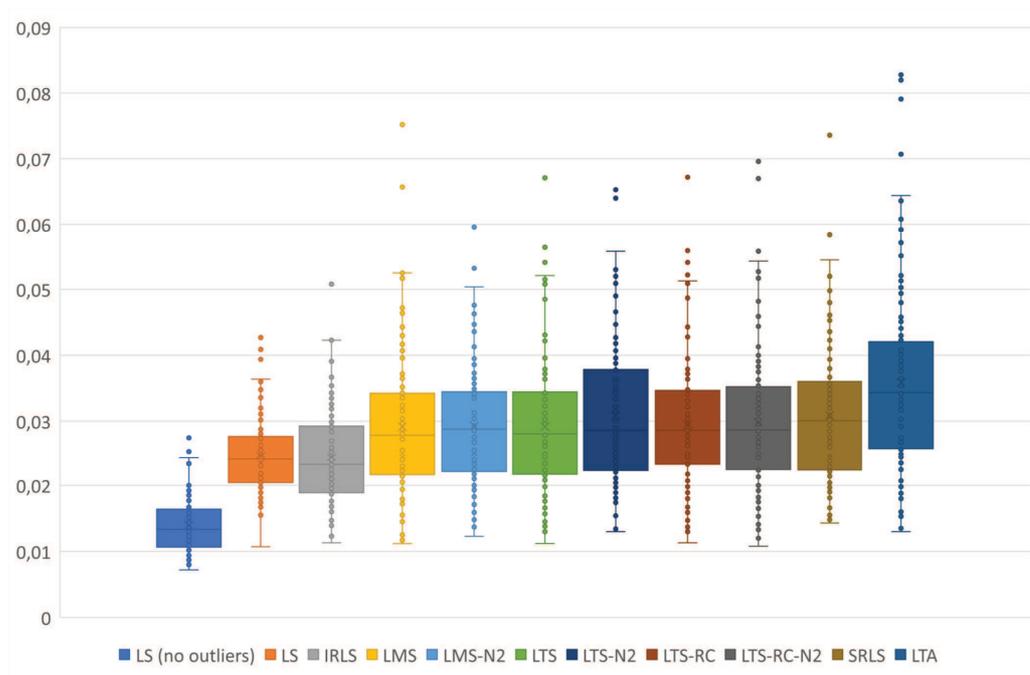
Fonte: Elaborado pelo autor

Figura 24: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 03 *outliers* de pequena magnitude.



Fonte: Elaborado pelo autor

Figura 25: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas em cada cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 04 *outliers* de pequena magnitude.



Fonte: Elaborado pelo autor

Os diagramas confirmam a melhor solução para o LS, que vai se deteriorando a medida que mais erros grosseiros são adicionados aos cenários. No último diagrama, apresentado na figura 25, as soluções já apresentam distribuições mais similares. Porém, o LS ainda se mantém como a solução com menor variação, mesmo mais afastada do resultado do LS sem *outliers*.

Em suma, para uma aplicação onde se objetiva a identificação de *outliers* e não a estimativa final em si através do método, a aplicação do LS não é a melhor. Nestes casos, conforme resultados dos experimentos, pode-se trabalhar com o IRLS, de menor custo computacional, ou para uma identificação melhor, com os ERs LMS, LTS e LTS-RC-N2 que obtiveram resultados mais satisfatórios. No geral, analisando os estimadores pela identificação dos *outliers*, destacam-se os métodos que fazem uso de meta-heurística para obtenção da solução.

Verificando a aplicabilidade dos métodos como estimativa final das coordenadas, para o caso de *outliers* de menor magnitude, o LS apresenta uma estimativa mais satisfatória, mesmo sendo o estimador com menor detecção de erros grosseiros. Isso mostra que os *outliers* não detectados pelo LS, não exercem grande distorção na estimativa final dos parâmetros, por se tratarem de erros pequenos. Ou seja, tanto para cenários sem *outliers* quanto para condições que apresentem erros grosseiros de menor magnitude, o LS persiste como melhor estimador para os parâmetros.

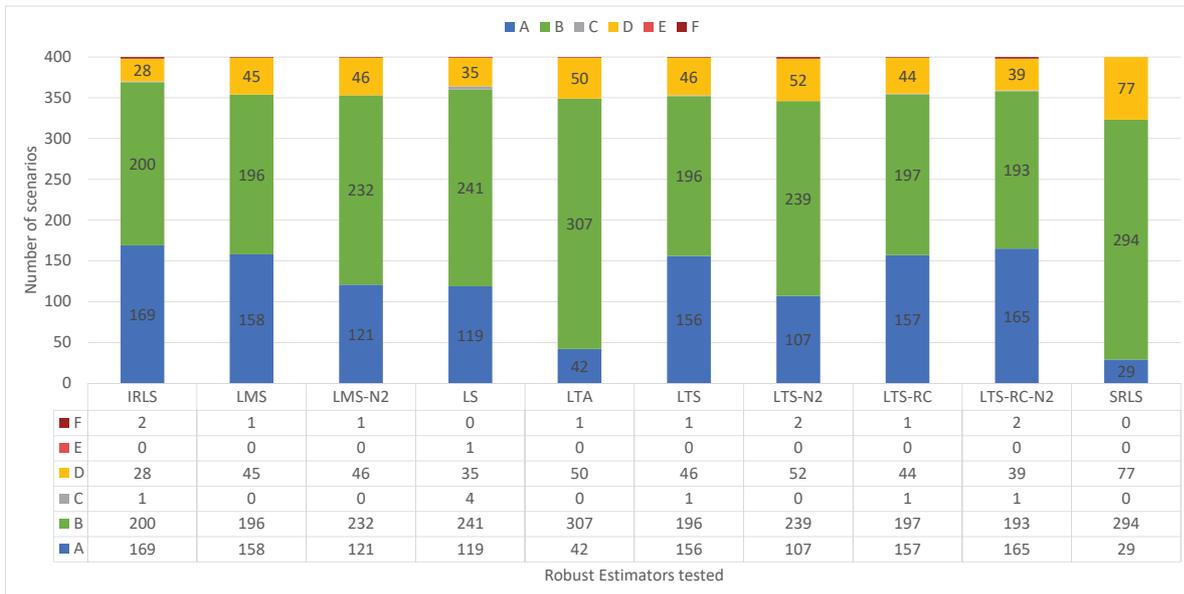
5.2.3 Os casos com *outliers* de grande magnitude

Conforme já visto neste capítulo, em cenários sem *outliers* ou de pequena magnitude, o LS apresenta ser o melhor estimador para os parâmetros, mesmo sendo o pior na detecção desses erros através do vetor de resíduos. Nesta subseção, abordar-se-á os casos com erros grosseiros de grande magnitude, variando de 6σ , até 12σ , configurando assim, o intervalo de $[6\sigma; 12\sigma]$. Ao todo foram executados 400 cenários (Pacotes 03, 05, 07 e 09) variando de 1 a 4 erros grosseiros simultâneos por cenário.

Iniciando a análise dos resultados pelas classificações das soluções, temos a Fig. 26 que traz a ocorrência de cada categoria nos 400 cenários. Ao contrário do que ocorreu com os experimentos com *outliers* de pequena magnitude, agora todos os ERs apresentam maior facilidade na detecção de *outliers* maiores, com a grande maioria resultados classificados como “A” ou “B” e muito raramente como “E” ou “F”, sendo que alguns ERs não apresentaram qualquer solução nesta categoria.

Outro ponto interessante, é que ao trabalhar com *outliers* de grande magnitude, o LS passa a apresentar, também, soluções com falsos positivos, tipos “B” e “D”, se assemelhando aos demais estimadores testados. Conforme fig. 27, é possível verificar que nas soluções classificadas como “A” combinado com “B”, ou seja, identificaram todos os *outliers*, trazendo falsos positivos, os estimadores apresentam resultados parecidos. A maioria dos experimentos apontam classificações deste tipo com resultados em torno de 90%. Já as soluções tipo “A” apresentam uma diferença maior, variando de 7% a 42%. Os melhores ERs nesta categoria foram: IRLS

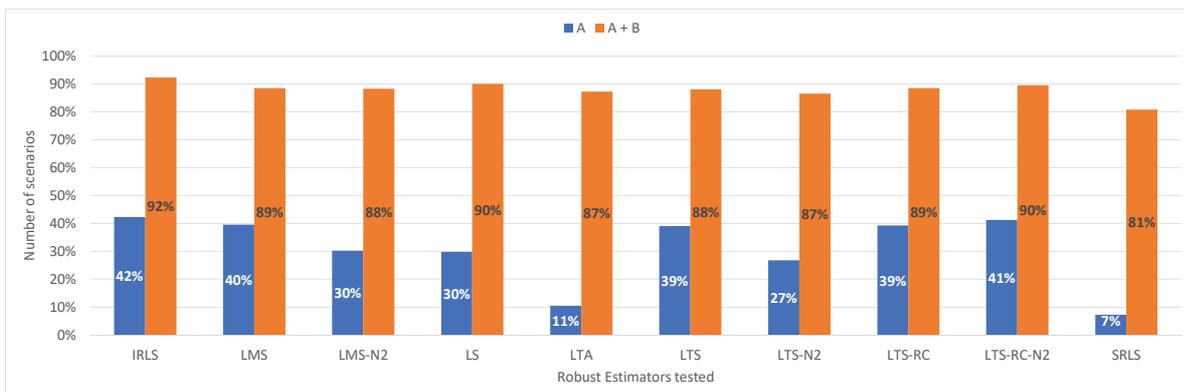
Figura 26: Contagem das classificações de “A” até “F” obtidas nos 400 cenários contendo de 1 a 4 outliers de maior magnitude, organizados por estimador.



Fonte: Elaborado pelo autor

(42%); LTS-RC-N2 (41%); LMS (40%); LTS (39%); e LTS-RC (39%).

Figura 27: Porcentagem de classificações do tipo “A” e, “A” combinado com “B”, para cada estimador, nos cenários com outliers de grande magnitude.



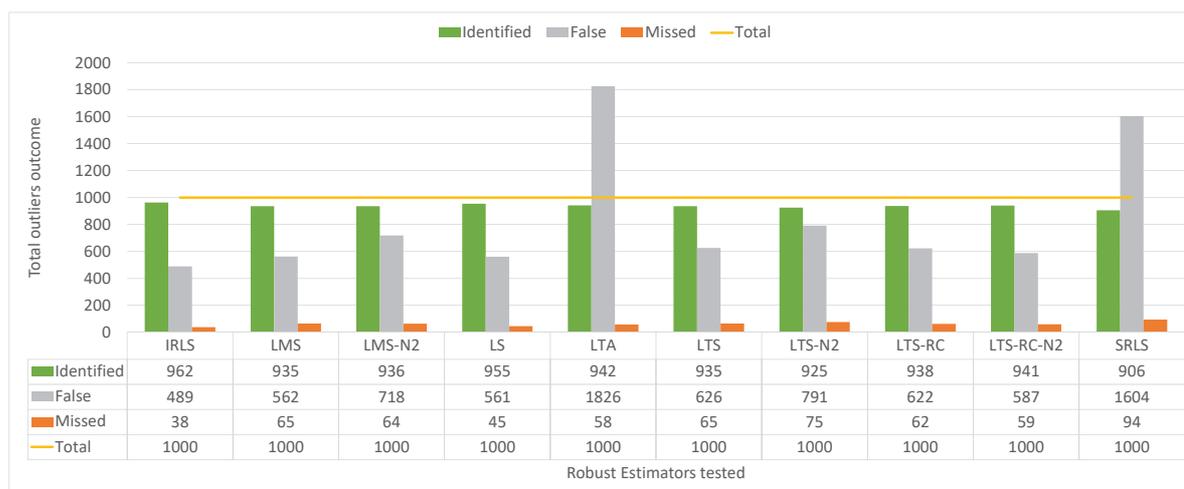
Fonte: Elaborado pelo autor

O IRLS apresenta uma leve vantagem, tanto na classificação em soluções do tipo “A” e, “A” combinado com “B”, o que é interessante do ponto de vista computacional, uma vez que não necessita de uma MH para a estimativa.

Na quantificação de detecção, falsos positivos e não-detectados, todos os estimadores detectaram a maior parte dos outliers, conforme é visto a Fig. 28. Para um total de 1000 erros grosseiros, os valores de detecção variam de 906 a 962 detectados. Neste caso, resta verificar a quantidade de falsos positivos já que a detecção dos outliers é quase ótima.

O IRLS, o LS, o LMS e o LTS-RC-N2, apresentaram os menores valores de falsos positivos, variando de 489 a 587. Em contraste, o LTA e o SRLS apresentam o pior resultado neste sentido, trazendo vários falsos positivos, com 18 falsos para cada 10 *outliers* no LTA e 16 falsos para cada 10 erros grosseiros no SRLS. Todos os demais estimadores ficaram com valores abaixo dos 800 falsos positivos.

Figura 28: Quantidade de *outliers* identificados, falsos e não constatados para cada estimador, nos 400 cenários com erros grosseiros de maior magnitude.



Fonte: Elaborado pelo autor

Entre os ERs que usam de meta-heurística, o LMS e o LTS-RC-N2 se saíram melhor, identificando boa parte dos *outliers* sem tantos falsos positivos, levando em consideração que o alto número de falsos do LTA não convém. Para uma análise objetivando a adoção da solução final destes estimadores no cálculo dos parâmetros, e não apenas para a detecção de *outliers*, a Tab. 3, traz uma tabela com o afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos. Também foi adicionada a solução do LS para os mesmos cenários, sem a adição dos *outliers*.

Ao contrário dos resultados apresentados pelo LS nos demais cenários, para o caso de *outliers* de grande magnitude, o LS apresenta uma estimativa final pouco satisfatória, sendo o terceiro pior estimador, atrás apenas do LTA e do SRLS. Com exceção destes, todos os demais estimadores apresentam resultados parecidos e mais satisfatórios, provando a maior insensibilidade aos *outliers*.

Por fim, elaborou-se quatro diagramas de caixa, apresentados nas Figs. 29, 30, 31 e 32. Cada ponto representa o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede. Também foram adicionadas as soluções do LS para os mesmos cenários sem a adição de *outliers*.

Os diagramas confirmam a deterioração da solução para o LS, comparada a estimativa com *outliers* pequenas, mesmo com apenas 1 *outlier* de grande magnitude. A medida que mais erros grosseiros são adicionados aos cenários, o LS rapidamente se torna uma das piores soluções

Tabela 3: Afastamento médio das soluções para cada estimador, separados nos cenários de 1 a 4 erros grosseiros simultâneos de grande magnitude. “LS (no outliers)”, indicada a solução por MMQ, sem a adição dos *outliers*.

Mean distance from original coordinates for all scenarios (in meters)											
	LS (no outliers)	LS	LMS	LTS	LTS-RC	SRLS	IRLS	LTS-RC-N2	LMS-N2	LTS-N2	LTA
1 outlier	0,014	0,023	0,022	0,023	0,023	0,030	0,023	0,023	0,025	0,025	0,037
2 outliers	0,014	0,031	0,025	0,027	0,027	0,036	0,026	0,027	0,025	0,028	0,039
3 outliers	0,014	0,037	0,024	0,028	0,028	0,040	0,028	0,028	0,029	0,031	0,044
4 outliers	0,014	0,045	0,036	0,031	0,031	0,043	0,032	0,031	0,034	0,036	0,051
Mean	0,014	0,034	0,027	0,027	0,027	0,037	0,027	0,027	0,028	0,030	0,043

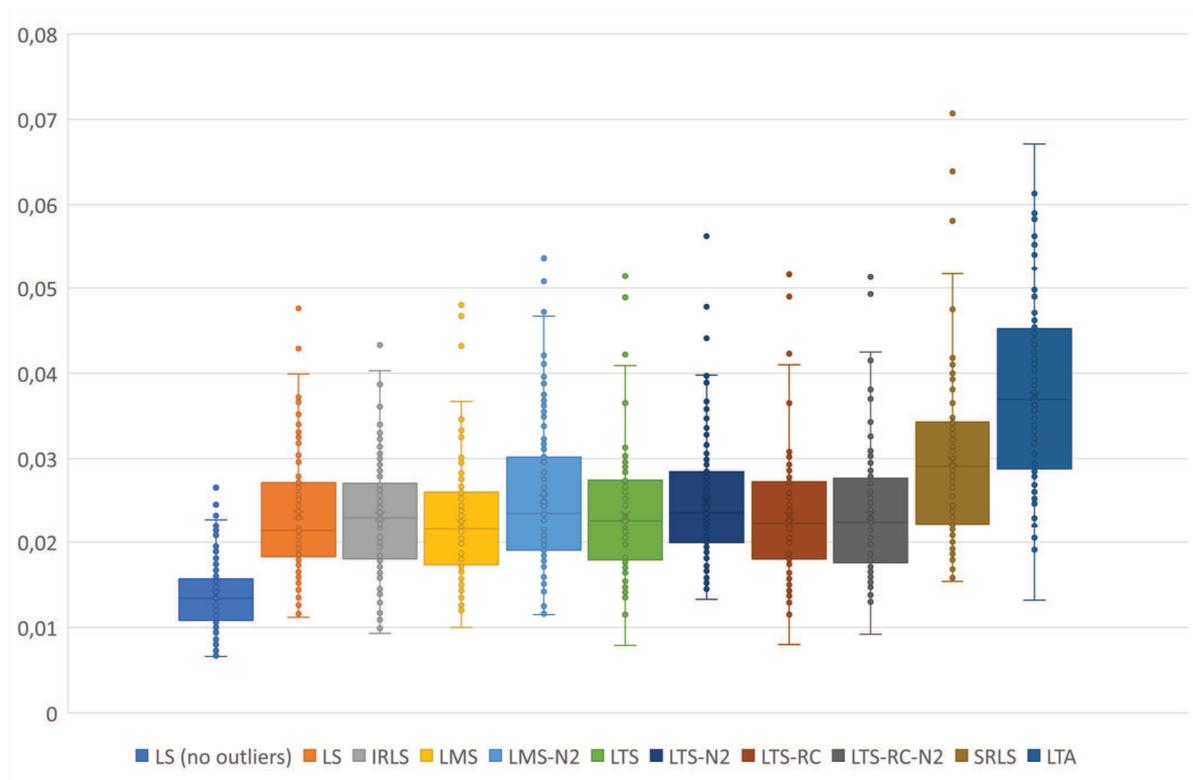
Fonte: Elaborado pelo autor

para estimativa direta dos parâmetros. No último diagrama, com a situação mais extrema (4 *outliers*), apresentado na Fig. 25, destacam-se os todos os ERs, com exceção do SRLS e LTA que sofrem maior distorção na estimativa. Apesar da piora no LS, é interessante observar que as soluções deste estimador permanecem bem distribuídas, sem muitos pontos além dos fios do diagrama.

Em suma, para uma aplicação onde se objetiva a identificação de *outliers* e não a estimativa final em si através do método, a aplicação do LS, IRLS, LMS ou LTS-RC-N2 devem apresentar melhores detecções. Em respeito ao custo computacional dessas soluções, o LS será a melhor escolha por possuir um método iterativo próprio.

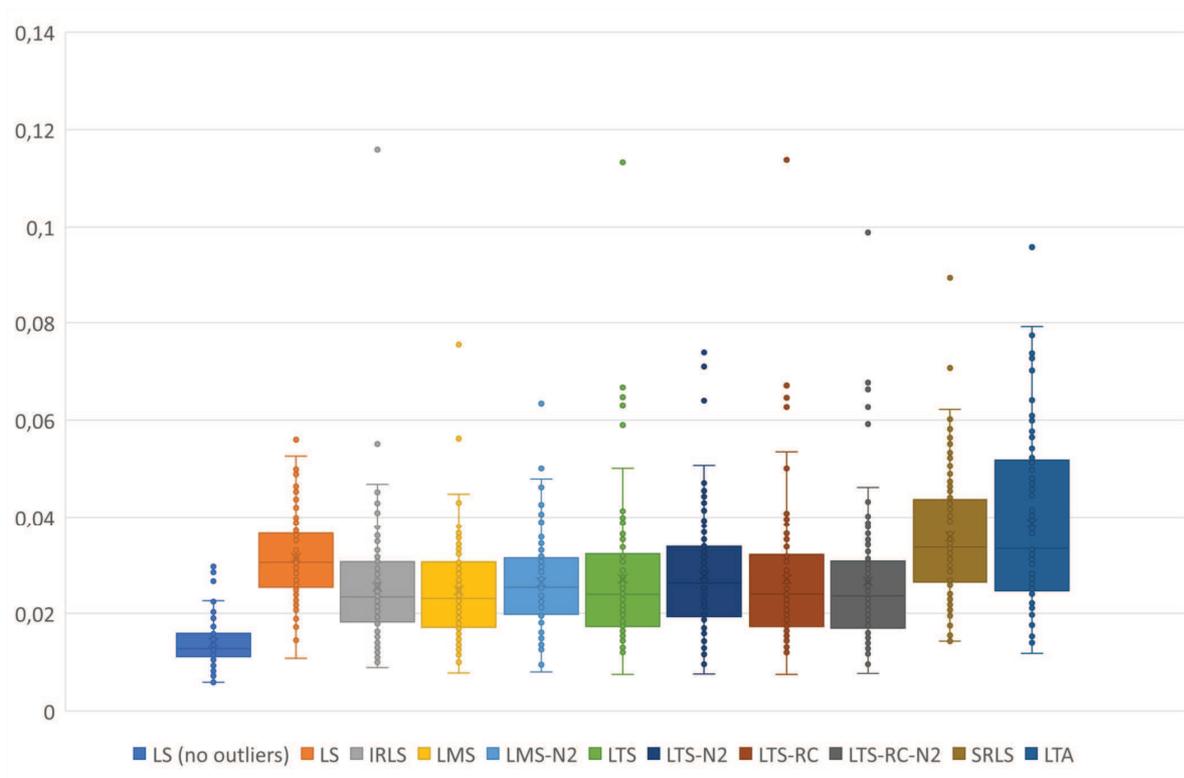
Verificando a aplicabilidade dos métodos como estimativa final das coordenadas, para o caso de *outliers* de grande magnitude, os ERs apresentam uma estimativa mais satisfatória. Isso demonstra a sensibilidade do LS aos erros grosseiros de grande magnitude e a resistência dos métodos robustos.

Figura 29: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 01 *outlier* de grande magnitude.



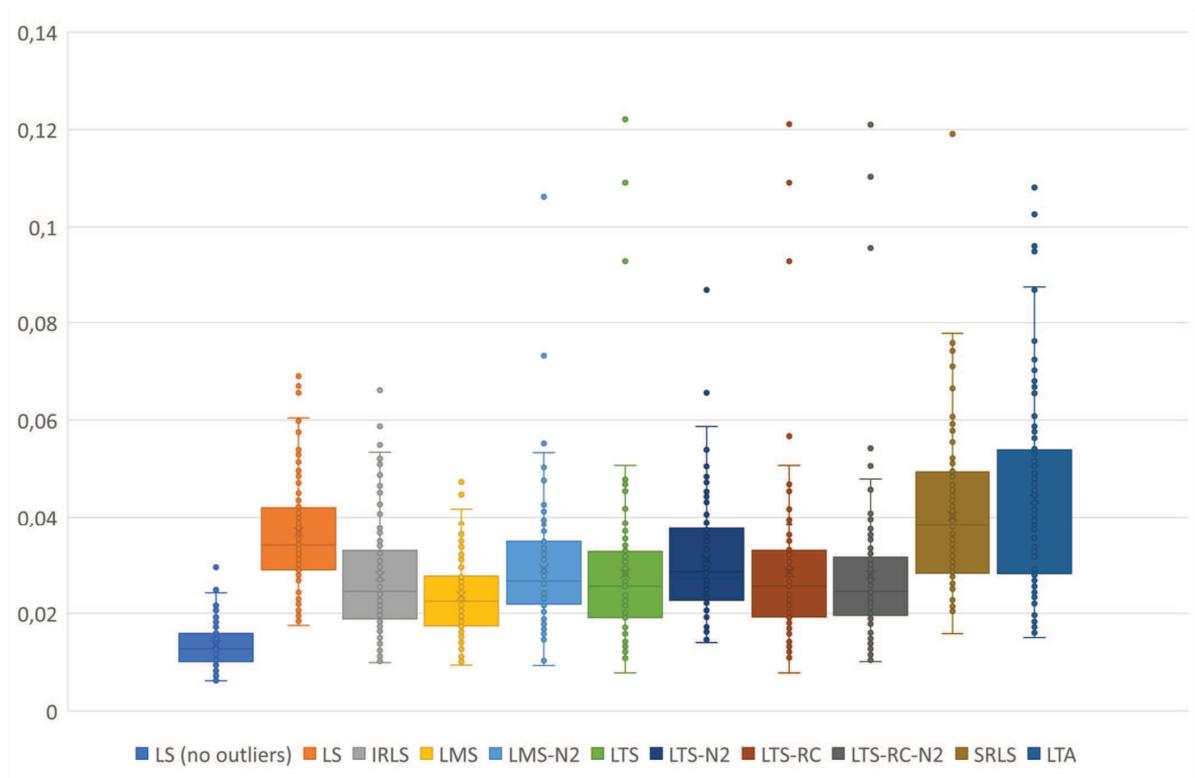
Fonte: Elaborado pelo autor

Figura 30: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 02 *outliers* de grande magnitude.



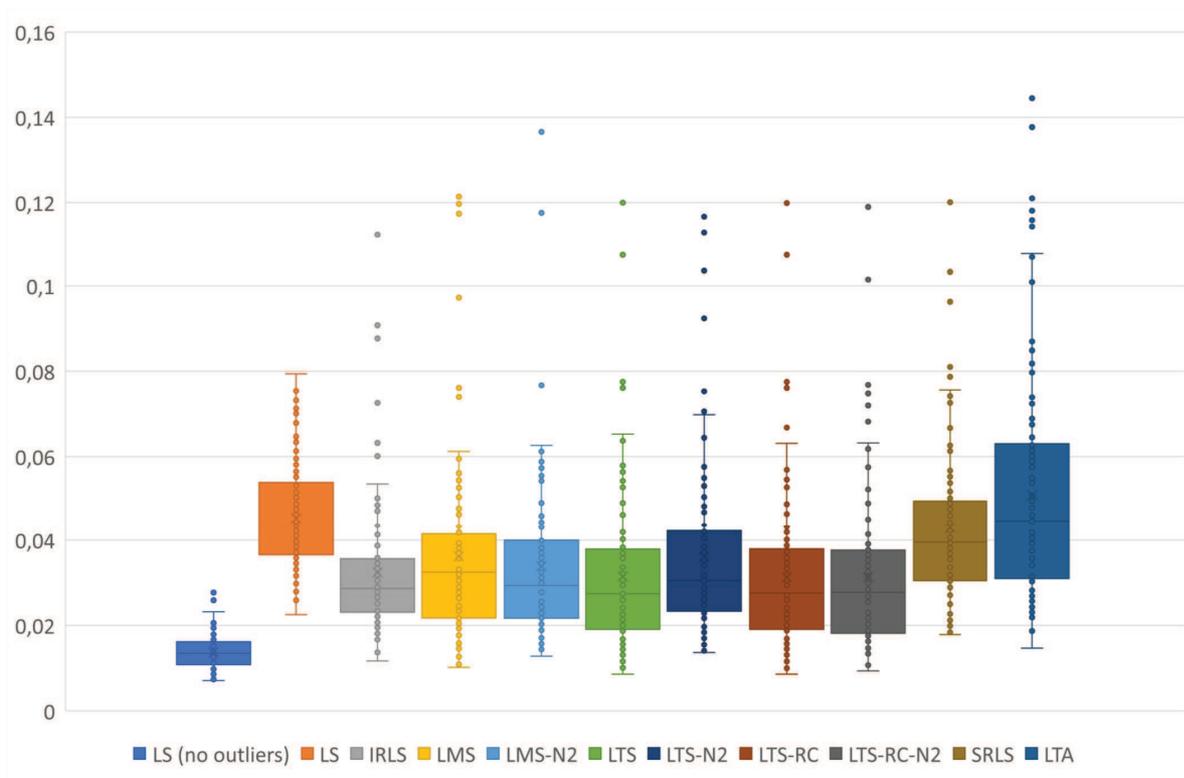
Fonte: Elaborado pelo autor

Figura 31: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 03 *outliers* de grande magnitude.



Fonte: Elaborado pelo autor

Figura 32: Diagrama de caixa apresentando o afastamento médio das coordenadas estimadas de um dado cenário, em relação aos valores oficiais das estações na rede, organizadas por estimador, para os cenários com 04 *outliers* de grande magnitude.



Fonte: Elaborado pelo autor

6 CONSIDERAÇÕES FINAIS

Conforme visto, as redes geodésicas são a base para atividades de mapeamento, geoinformação, registro de terras e outros serviços baseados em localização, além de também apresentarem um papel importante na sociedade em obras de infraestrutura que são diretamente dependentes de pontos de controle tridimensionais altamente precisos.

Este trabalho objetivou fazer uma avaliação da estratégia do uso de Estimadores Robustos (ERs) para o ajustamento de redes geodésicas e detecção de *outliers*. Os experimentos foram executados através da aplicação de uma meta-heurística nova denominada *Independent Vortices Search*. A seguir serão apresentadas de forma resumida as principais contribuições do trabalho para a comunidade científica, além de possibilidades para trabalhos futuros.

6.1 Sobre o IVS

As meta-heurísticas tem sido exploradas há vários anos pela comunidade científica e ainda há espaço para o desenvolvimento de novas técnicas. O IVS partiu do já conhecido *Modified Vortex Search* e conseguiu melhorar significativamente os resultados da MH original e superou, inclusive, as soluções de MHs já conhecidas, como a Colônia de Abelhas Artificiais.

O fato do IVS trabalhar com os centros de forma independente, facilita a implementação do algoritmo para procedimentos que exijam alto desempenho, uma vez que pode ser facilmente paralelizado. A paralelização do IVS fica em aberto para a continuidade do trabalho e a exploração de problemas mais complexos ou mesmo redes geodésicas maiores. Ainda, pode acompanhar um estudo da escalabilidade do problema, para redes grandes com dezenas de vértices.

6.2 Sobre o controle de qualidade com ERs

Nos experimentos que objetivaram o controle de qualidade, vimos que em situações com pequenos *outliers*, uma aplicação que deseje simplesmente a identificação de *outliers* — e não a estimativa final em si através do método — a aplicação do LS não é a mais recomendada. Nestes casos, conforme resultados dos experimentos, pode-se trabalhar com o IRLS, de menor custo computacional, ou para uma identificação melhor, com os ERs LMS, LTS ou LTS-RC-N2 que obtiveram resultados mais satisfatórios. No geral, analisando os estimadores pela identificação dos *outliers*, destacaram-se os métodos que fazem uso de meta-heurística para obtenção da solução.

Para os casos com nenhum, e também, pequenos *outliers*, verificando a aplicabilidade dos métodos para estimativa final das coordenadas, o LS apresentou uma estimativa mais satisfatória, mesmo sendo o estimador com menor detecção de erros grosseiros. Isso mostra que os *outliers* não detectados pelo LS, não exercem grande distorção na estimativa final dos parâme-

tros, por se tratarem de erros pequenos. Ou seja, tanto para cenários sem *outliers* quanto para condições que apresentem erros grosseiros de menor magnitude, o LS persistiu como melhor estimador para os parâmetros.

Para cenários e situações onde temos *outliers* de grande magnitude, onde se planeje a identificação de *outliers* e não a estimativa final em si através do método, a aplicação do LS, IRLS, LMS e LTS-RC-N2 apresentaram melhores detecções. Com respeito ao custo computacional dessas soluções, o LS é a melhor escolha por possuir um método de cálculo direto e simples.

Verificando a aplicabilidade dos métodos como estimativa final das coordenadas, em cenários com *outliers* de grande magnitude, constatou-se que os ERs apresentam uma estimativa muito mais satisfatória que o LS. Isto demonstrou a sensibilidade do LS aos erros grosseiros de grande magnitude e a maior resistência dos métodos robustos.

Deste modo, podemos afirmar que a hipótese II — “A maioria dos ERs apresentam resultados melhores que o MMQ na maior parte dos cenários de erros testados” — é parcialmente verdadeira. Nos casos de nenhum *outlier* ou de pequena magnitude, os resultados do ERs são inferiores ao LS, e nos casos de *outliers* de grande magnitude, são superiores.

Já a hipótese III — “É possível detectar a maior parte dos *outliers* inseridos nas observações a partir dos vetores de resíduos gerados pelos ERs” — se provou verdadeira. Independente da magnitude dos *outliers*, a maior parte deles é exposta no vetor de resíduos.

Apesar do custo computacional dos ERs que fazem uso de MHs ser maior do que os métodos clássicos, não é possível estabelecer um custo-benefício por ora. Para isso, primeiro é necessário estudar a possibilidade de estabelecer uma quantidade mínima de FE para a obtenção de bons resultados para cada ER e levando em consideração a dimensão de rede. Neste sentido, trabalhos futuros podem considerar um estudo de escalabilidade e de FE mínimos para cada ER e tamanho de rede. Caso o processamento requerido verifique ainda ser alto, a paralelização do IVS é uma opção, como já citado. No campo da estimativa robusta, também podem ser estudadas outras estratégias, como implementar novas restrições a ERs, desenvolver novos estimadores, ou ainda, testar outros ERs não contemplados por este trabalho.

REFERÊNCIAS

- AKAY, B.; KARABOGA, D. A modified Artificial Bee Colony algorithm for real-parameter optimization. **Information Sciences**, Kayseri, v. 192, p. 120–142, 2012.
- BASELGA, S. Global Optimization Solution of Robust Estimation. **Journal of Surveying Engineering**, Valencia, v. 133, n. 3, p. 123–128, 2007.
- BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, Algiers, v. 237, p. 82–117, 2013.
- DOGAN, B. A Modified Vortex Search Algorithm for Numerical Function Optimization. **International Journal of Artificial Intelligence & Applications**, [S.l.], v. 7, n. 3, p. 37–54, 2016.
- DOGAN, B.; ÖLMEZ, T. A new metaheuristic for numerical function optimization: vortex search algorithm. **Information Sciences**, Istanbul, v. 293, p. 125–145, 2015.
- DOGAN, B.; ÖLMEZ, T. Vortex search algorithm for the analog active filter component selection problem. **AEU - International Journal of Electronics and Communications**, [S.l.], v. 69, n. 9, p. 1243–1253, sep 2015.
- GHILANI, C. D. **Adjustment Computations: spatial data analysis: fifth edition**. [S.l.: s.n.], 2010.
- GUO, J.; OU, J.; WANG, H. Robust estimation for correlated observations: two local sensitivity-based downweighting strategies. **Journal of Geodesy**, [S.l.], v. 84, n. 4, p. 243–250, apr 2010.
- HAMPEL, F. R. et al. **Robust Statistics**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1986. 1–502 p.
- HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; WASLE, E. **GNSS – Global Navigation Satellite Systems GPS, GLONASS, Galileo, and more**. [S.l.: s.n.], 2008. 1–537 p.
- HUBER, P. J.; RONCHETTI, E. M. **Robust Statistics**. 2. ed. [S.l.: s.n.], 2009.
- KARABOGA, D. An idea based on Honey Bee Swarm for Numerical Optimization. **Technical Report TR06, Erciyes University**, [S.l.], n. TR06, p. 10, 2005.
- KARABOGA, D.; AKAY, B. A comparative study of Artificial Bee Colony algorithm. **Applied Mathematics and Computation**, Kayseri, v. 214, n. 1, p. 108–132, 2009.
- KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. **Journal of Global Optimization**, Kayseri, v. 39, n. 3, p. 459–471, 2007.
- KARABOGA, D.; BASTURK, B. On the performance of artificial bee colony (ABC) algorithm. **Applied Soft Computing Journal**, Kayseri, v. 8, n. 1, p. 687–697, 2008.

KARABOGA, D. et al. A comprehensive survey: artificial bee colony (abc) algorithm and applications. **Artificial Intelligence Review**, Kayseri, v. 42, p. 21–57, 2012.

KLEIN, I. **Controle de qualidade no ajustamento de observações geodésicas**. 2011. 322 p. Tese (Doutorado em Ciência da Computação) — UFRGS, 2011.

KLEIN, I. et al. **Planejamento de redes geodésicas resistentes a múltiplos outliers**. 2012. 480–507 p. v. 18, n. 3.

KOCH, I. É. et al. Least trimmed squares estimator with redundancy constraint for outlier detection in GNSS networks. **Expert Systems with Applications**, [S.l.], v. 88, p. 230–237, 2017.

KOCH, K.-R. **Parameter Estimation and Hypothesis Testing in Linear Models**. [S.l.: s.n.], 1999. 333 p.

LEICK, A.; RAPOPORT, L.; TATARNIKOV, D. **GPS Satellite Surveying: fourth edition**. [S.l.: s.n.], 2015. 1–807 p.

MERNIK, M. et al. On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. **Information Sciences**, [S.l.], v. 291, n. C, p. 115–127, 2015.

MONICO, J. F. G. **Posicionamento pelo GNSS: descrição, fundamentos e aplicações**. 2. ed. [S.l.]: Editora Unesp, 2008. 480 p.

OSMAN, I. H.; KELLY, J. P. Meta-Heuristics Theory and Applications. **Journal of the Operational Research Society**, [S.l.], v. 48, n. 6, p. 657–657, 1997.

ROUSSEEUW, P. Least Median of Squares Regression. **Journal of the American Statistical Association**, Delft, v. 79, n. 388, p. 871–880, 1984.

ROUSSEEUW, P. J.; LEROY, A. M. **Robust Regression and Outlier Detection**. New York, NY, USA: John Wiley & Sons, Inc., 1987. 352 p.

SÖRENSEN, K. Metaheuristics-the metaphor exposed. **International Transactions in Operational Research**, [S.l.], v. 22, n. 1, p. 3–18, 2015.

TABLEMAN, M. The asymptotics of the least trimmed absolute deviations (LTAD) estimator. **Statistics and Probability Letters**, [S.l.], v. 19, n. 5, p. 387–398, 1994.

TALBI, E. **Metaheuristics: from design to implementation**. 1. ed. Hoboken: John Wiley & Sons, 2009. 624 p. v. 2009.

TEUNISSEN, P. J. G. **Adjustment Theory: an introduction**. [S.l.: s.n.], 2000. 193 p.

WANG, Z.; WU, G.; WAN, Z. A novel hybrid vortex search and artificial bee colony algorithm for numerical optimization problems. **Wuhan University Journal of Natural Sciences**, [S.l.], v. 22, n. 4, p. 295–306, aug 2017.

XU, P. Sign-constrained robust least squares, subjective breakdown point and the effect of weights of observations on robustness. **Journal of Geodesy**, Kyoto, v. 79, n. 1-3, p. 146–159, 2005.

YETKIN, M.; BERBER, M. Application of the Sign Constrained Robust Least Squares Method to Surveying Networks. **Journal of Surveying Engineering**, Boca Raton, v. 139, n. February, p. 59–65, 2012.

YETKIN, M.; BERBER, M. Implementation of robust estimation in GPS networks using the Artificial Bee Colony algorithm. **Earth Science Informatics**, Izmir, v. 7, n. 1, p. 39–46, 2014.