



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Otimização de Redes de Sensores Visuais sem Fio por Algoritmos Evolutivos Multiobjetivo

Elivelton Oliveira Rangel

Feira de Santana

2018



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Elivelton Oliveira Rangel

Otimização de Redes de Sensores Visuais sem Fio por Algoritmos Evolutivos Multiobjetivo

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Angelo Conrado Loula

Coorientador: Prof. Dr. Daniel Gouveia Costa

Feira de Santana

2018

Ficha Catalográfica - Biblioteca Central Julieta Carteado - UEFS

R154 Rangel, Elivelton Oliveira

Otimização de redes de sensores visuais sem fio por algoritmos evolutivos multiobjetivo / Elivelton Oliveira Rangel. – 2018.

92 f. : il.

Orientador: Angelo Conrado Loula.

Coorientador: Daniel Gouveia Costa.

Dissertação (Mestrado) – Universidade Estadual de Feira de Santana, Programa de Pós-Graduação em Computação Aplicada, 2018.

1. Redes de sensores sem fio. 2. Redes de sensores visuais sem fio. 3. Algoritmos evolutivos. 4. Otimização multiobjetivo. I. Loula, Angelo Conrado. II. Costa, Daniel Gouveia. III. Universidade Estadual de Feira de Santana. IV. Título.

CDU: 004.722

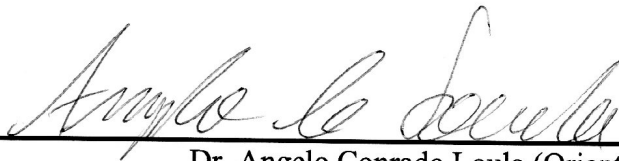
Elivelton Oliveira Rangel

Otimização de Redes de Sensores Visuais sem Fio por Algoritmos Evolutivos Multiobjetivo

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Feira de Santana, 27 de março de 2018

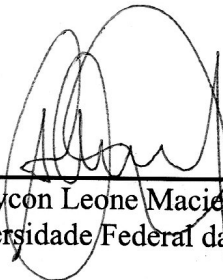
BANCA EXAMINADORA



Dr. Angelo Conrado Loula (Orientador)
Universidade Estadual de Feira de Santana



Dr. Matheus Giovanni Pires
Universidade Estadual de Feira de Santana



Dr. Maycon Leone Maciel Peixoto
Universidade Federal da Bahia

Resumo

As redes de sensores visuais sem fio podem obter, através de câmeras, informações importantes para aplicações de controle e monitoramento, e tem ganhado atenção da comunidade acadêmica nos últimos anos. Para algumas aplicações, um conjunto de alvos deve ser coberto por sensores visuais, e por vezes com demanda de redundância de cobertura, especialmente quando há requisitos de disponibilidade ou demandas de múltiplas perspectivas de cobertura para os alvos visados. Para sensores visuais rotacionáveis, as orientações de detecção podem ser ajustadas para otimizar cobertura e redundância, existindo diferentes abordagens de otimização disponíveis para solucionar esse problema. Particularmente, como diferentes parâmetros de otimização podem ser considerados, o problema de maximização de cobertura redundante pode ser tratado como um problema multiobjetivo, com algumas soluções potenciais a serem consideradas. Neste contexto, dois algoritmos evolutivos diferentes são propostos para calcular a maximização de cobertura redundante para visualização de alvos, pretendendo ser alternativas mais eficientes para algoritmos gulosos. Os resultados da simulação reforçam os benefícios de empregar algoritmos evolutivos para ajustes das orientações dos sensores, potencialmente beneficiando a implantação e o gerenciamento de redes de sensores visuais sem fio para diferentes aplicações.

Palavras-chave: Algoritmos Evolutivos, Redes de Sensores sem Fio, Redes de Sensores Visuais sem Fio, Otimização multiobjetivo.

Abstract

Wireless visual sensor networks can provide valuable information for a lot of monitoring and control applications, which has driven much attention from the academic community in last years. For some applications, a set of targets have to be covered by visual sensors and sensing redundancy may be desired in many cases, especially when applications have availability requirements or demands for multiple coverage perspectives for viewed targets. For rotatable visual sensors, the sensing orientations can be adjusted for optimized coverage and redundancy, with different optimization approaches available to address this problem. Particularly, as different optimization parameters may be considered, the redundant coverage maximization issue may be treated as a multi-objective problem, with some potential solutions to be considered. In this context, two different evolutionary algorithms are proposed to compute redundant coverage maximization for target viewing, intending to be more efficient alternatives to greedy-based algorithms. Simulation results reinforce the benefits of employing evolutionary algorithms for adjustments of sensors' orientations, potentially benefiting deployment and management of wireless visual sensor networks for different applications.

Keywords: Evolutionary Algorithms, Wireless Visual Sensor Networks, Wireless Sensor Networks, Multiobjective Optimization.

Prefácio

Esta dissertação de mestrado foi submetida a Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

A dissertação foi desenvolvido dentro do Programa de Pós-Graduação em Computação Aplicada (PGCA) tendo como orientador o Dr. **Angelo Conrado Loula** e co-orientador o Dr. **Daniel Gouveia Costa**.

Esta pesquisa foi financiada pela CAPES.

Agradecimentos

Agradeço a Deus, antes de tudo, pelo dom da vida e por construir minhas vitórias chegando aonde cheguei.

Agradeço aos meus pais Elio e Conceição pela força, apoio, incentivo, carinho e amor que me dão todos os dias e por vibrarem e comemorarem juntos a cada vitória alcançada.

A minha companheira Josi, por ter me suportado durante os períodos difíceis enfrentados até aqui, além de muito amor e incentivo.

Ao meu orientador Prof. Dr. Angelo Loula e ao meu Coorientador Prof. Dr. Daniel Costa pela dedicação, por suas grandes contribuições durante todo o processo de execução deste trabalho e por compartilhar seus conhecimentos, tornando esse trabalho uma realidade.

Agradeço também à Universidade Estadual de Feira de Santana por ter proporcionado uma experiência fantástica durante todo o curso.

Agradeço a todos os docentes, colegas e amigos que contribuíram também para a minha formação, em especial ao grupo de estudos da disciplina de Inteligência Computacional (IC), pelas tardes/noites/madrugadas de diversão, ideias e pelo enorme cabedal adquirido durante as reuniões.

Por fim, agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela bolsa de mestrado ofertada.

Muito obrigado a todos!

Sumário

Resumo	i
Abstract	ii
Prefácio	iii
Agradecimentos	iv
Sumário	vi
Lista de Tabelas	vii
Lista de Figuras	ix
Lista de Abreviações	x
Lista de Símbolos	xi
1 Introdução	1
1.1 Objetivos	3
1.1.1 Objetivos Específicos	3
1.2 Organização do Trabalho	4
2 Revisão de literatura	5
2.1 Redes de Sensores sem Fio	5
2.2 Redes de Sensores Visuais sem Fio	8
2.2.1 Sensores Visuais	9
2.3 Visualização de Alvos	11
2.4 Cobertura e redundância nas redes de sensores visuais sem fio	12
2.5 Trabalhos Relacionados	14
2.5.1 Algoritmos Gulosos para Otimização da Cobertura Redundante em Rede de Sensores Visuais sem Fio	16
2.6 Otimização Multiobjetivo	17
2.6.1 Abordagem <i>a priori</i>	20
2.6.2 Abordagem <i>a posteriori</i>	21

2.7	Algoritmos Evolutivos	22
2.7.1	Representação (Definição do Indivíduo)	23
2.7.2	População	25
2.7.3	Função de Avaliação (Função <i>Fitness</i>)	25
2.7.4	Operadores de Seleção	25
2.7.5	Operadores de Recombinação (<i>Crossover</i>)	27
2.7.6	Operadores de Mutação	30
2.8	Parâmetros do Algoritmo Evolutivo	32
2.9	Algoritmos Evolutivos para Otimização Multiobjetivo	33
2.9.1	O Algoritmo SPEA2	34
2.9.2	O Algoritmo NSGA-II	35
3	Metodologia	38
3.1	Modelagem e Formalização do Problema	38
3.2	Algoritmo Evolutivo para Abordagem Lexicográfica	41
3.3	Algoritmo Evolutivo multiobjetivo NSGA-II	43
3.4	Otimização de parâmetros (<i>Parameter Tunning</i>)	44
3.5	Método de Avaliação dos Resultados	45
4	Resultados	47
4.1	Definição dos Parâmetros	49
4.2	Algoritmo Evolutivo Lexicográfico X CPGA, ECPGA e RCMA	53
4.3	Comparação entre o NSGA-II x AE Lexicográfico, CPGA, ECPGA e RCMA	60
4.4	Custo Computacional	68
5	Considerações Finais	70
	Referências Bibliográficas	72

Lista de Tabelas

4.1	Valores do <i>grid search</i> para o AE lexicográfico	49
4.2	Valores do <i>grid search</i> para o NSGA-II	49
4.3	Resultados após execução do <i>grid search</i> no AE Lexicográfico com 20 sensores.	50
4.4	Resultados após execução do <i>grid search</i> no AE Lexicográfico com 40 sensores.	50
4.5	Resultados após execução do <i>grid search</i> no AE Lexicográfico com 60 sensores.	50
4.6	Parâmetros do algoritmo evolutivo lexicográfico	51
4.7	Resultados após execução do <i>grid search</i> no NSGA-II com 20 sensores.	51
4.8	Resultados após execução do <i>grid search</i> no NSGA-II com 40 sensores.	52
4.9	Resultados após execução do <i>grid search</i> no NSGA-II com 60 sensores.	52
4.10	Parâmetros do NSGA-II	53
4.11	Análise dos algoritmos lexicográfico e gulosos comparando cada cenário (50 alvos) individualmente.	56
4.12	Análise dos algoritmos lexicográfico e gulosos comparando cada cenário (100 alvos) individualmente.	58
4.13	Custo computacional considerando as médias dos testes com 50 alvos	68
4.14	Custo computacional considerando as médias dos testes com 100 alvos	69

Lista de Figuras

2.1	Topologia de uma Rede de Sensores sem Fio.	6
2.2	Tipos de distribuição em uma rede de sensores sem fio.	7
2.3	Câmeras com função de zoom panorâmico PTZ.	10
2.4	Representação de um nó sensor visual genérico.	10
2.5	Verificando se um alvo está no FoV do sensor visual. O alvo está dentro em a) , e fora em b)	12
2.6	Ilustração de cobertura e redundância em uma RSVSF.	13
2.7	Fluxograma de um algoritmo evolutivo genérico.	23
2.8	Seleção por roleta. As letras indicam a seção de adequação dos indivíduos, onde a região A possui maior probabilidade, e a região F possui menor probabilidade.	26
2.9	Exemplificação de cruzamento de ponto único.	27
2.10	Exemplificação de cruzamento de dois pontos.	28
2.11	Exemplificação de cruzamento através do método uniforme.	28
2.12	Distribuição da probabilidade para criar soluções filhas. As soluções pais estão marcadas como “o”. Adaptado de Deb e Beyer (2001) . . .	29
2.13	Mutação por mudança de ordem.	31
2.14	Mutação por soma de valores, onde V é o vetor e o índice V_5 foi escolhido aleatoriamente, somando 15 (também escolhido aleatoriamente) ao seu valor atual, sendo $60 + 15 = 75$	31
2.15	Mutação por substituição de valores, onde V é o vetor e o índice V_2 foi escolhido aleatoriamente, alterando para 18 o seu valor atual. . . .	31
2.16	Seleção pelo algoritmo NSGA-II.	36
2.17	Cálculo do <i>crowding-distance</i> . Os pontos preenchidos são soluções da mesma fronteira não-dominada. Adaptado de Deb et al. (2002). . . .	36
3.1	Monitoramento visual para 40 alvos e 25 sensores visuais, ambos implantados aleatoriamente. Considerando os 3 objetivos: <i>Cobertura</i> = 32.50%, <i>Redundância Mínima</i> = 2.50% e <i>Redundância Média</i> = 0.05. . . .	39
4.1	Monitoramento visual para 60 alvos e 38 câmeras, ambos implantados aleatoriamente. a) antes da otimização: $C = 56.66\%$, $R = 8.33\%$ e $RM = 0.1833$. b) após a otimização: $C = 96.66\%$, $R = 50\%$ e $RM = 1.05$	48

4.2	Resultados de cobertura para 50 alvos e diferentes números de sensores visuais.	54
4.3	Resultados de redundância mínima para 50 alvos e diferentes números de sensores visuais.	55
4.4	Resultados de redundância média para 50 alvos e diferentes números de sensores visuais.	55
4.5	Resultados de cobertura para 100 alvos e diferentes números de sensores visuais.	56
4.6	Resultados de redundância mínima para 100 alvos e diferentes números de sensores visuais.	57
4.7	Resultados de redundância média para 100 alvos e diferentes números de sensores visuais.	58
4.8	Evolução da população do algoritmo evolutivo lexicográfico na 6 ^a execução.	59
4.9	Resultados de cobertura para 50 alvos e diferentes números de sensores visuais.	61
4.10	Resultados de redundância mínima para 50 alvos e diferentes números de sensores visuais.	62
4.11	Resultados de redundância média para 50 alvos e diferentes números de sensores visuais.	62
4.12	Resultados de cobertura para 100 alvos e diferentes números de sensores visuais.	63
4.13	Resultados de redundância mínima para 100 alvos e diferentes números de sensores visuais.	64
4.14	Resultados de redundância média para 100 alvos e diferentes números de sensores visuais.	65
4.15	Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura x redundância mínima, no cenário com 60 sensores.	66
4.16	Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura x redundância média.	66
4.17	Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura e redundância média.	67
4.18	Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura e redundância média.	68

Lista de Abreviações

Abreviação	Descrição
AE	Algoritmos Evolutivos
EA	Evolutionary Algorithms
FoV	Field of View
DoF	Depth of Field
NSGA-II	<i>Non-dominate Sort Genetic Algorithm-II</i>
RSSF	Rede de Sensores Sem Fio
RSVSF	Rede de Sensores Visuais Sem Fio
SPEA	<i>Strength Pareto Evolutionary Algorithm</i>
WSN	Wireless Sensor Networks
WVSN	Wireless Visual Sensor Networks

Lista de Símbolos

Símbolo	Descrição
α	Ângulo de abertura
θ	Ângulo de orientação
A	Conjunto de alvos
P^*	Conjunto de Pareto-ótimo
S	Conjunto de sensores
S	Conjunto de soluções candidatas
\mathbb{Z}	Conjunto dos números inteiros
\prec	É dominado por
\succ	Domina
\exists	Existe
\subseteq	Igualdade entre conjuntos
η	Índice de distribuição
I	Indivíduo da população (solução candidata)
$>$	Maior que
$<$	Menor que
\geq	Maior ou igual
\leq	Menor ou igual
\forall	Para todo
\in	Pertence
w	Pesos
P	População
r	Raio de alcance
\sum	Somatório
P_{size}	Tamanho da população
\bar{P}_{size}	Tamanho da população alternativa

Capítulo 1

Introdução

Com a proliferação de sistemas microeletrônicos, sensores sem fio, mais equipados e completos, passaram a ser desenvolvidos com mais facilidade. Esses sensores possuem como objetivo capturar informações escalares do ambiente a ser monitorado, incluindo dados como temperatura, pressão atmosférica, umidade relativa do ar, entre outros (YICK; MUKHERJEE; GHOSAL, 2008). Tais dispositivos são compostos em sua maioria por um processador, memória, fonte de alimentação, transmissor e um atuador (dispositivo de sensoriamento). O conjunto desses dispositivos, posicionados de forma determinística ou aleatória, formam as redes de sensores sem fio (RSSF) que vem sendo objeto de estudo de muitos pesquisadores nos últimos anos (YAP; YEN, 2014).

As baterias portáteis são a principal fonte de alimentação dos sensores sem fio (YICK; MUKHERJEE; GHOSAL, 2008). Há também a possibilidade de captar energia através do meio ambiente, como por exemplo, energia solar e movimentação (YAP; YEN, 2014). Para isso, evidentemente, os sensores devem estar equipados com painéis solares (para energia solar) e atuador de movimentação, mas isso dependerá da disponibilidade desses recursos no ambiente da aplicação.

A gama de aplicações das redes de sensores sem fio pode ser muito extensa, podendo ter aplicações, por exemplo, em áreas militares, regiões de monitoramento remoto, institutos de clima e tempo, centros urbanos e em aplicações industriais (YICK; MUKHERJEE; GHOSAL, 2008). Nas áreas militares, o uso de sensores pode ajudar na identificação e detecção de intrusos, a exemplo de movimento de tropas e tanques coordenados e correlacionados espacialmente. Em regiões de monitoramento remoto, os sensores podem identificar movimentações suspeitas de catástrofes naturais, possibilitando a tomada de decisão com antecedência. Os institutos de clima e tempo utilizam os sensores com frequência para medir informações ambientais, possibilitando ter previsões sobre alguns fenômenos naturais como chuva, neve, tempestades, etc. Nos centros urbanos, podem existir diversos sensores para captura de diversas informações, a exemplo de poluição sonora, poluição do ar, radares de velocidade, entre outros.

Como pode ser visto, as RSSF compreendem uma vasta área a ser estudada. Pode-se monitorar diversos tipos de informação utilizando sensores sem fio, desde que estes estejam equipados com unidades de sensoriamento específicas para capturar tais informações. Com os avanços tecnológicos relativos a *hardware*, alguns processos demandam um consumo maior de energia, tráfego de dados, entre outros.

Quando os sensores são equipados com câmeras, uma grande quantidade de dados visuais pode ser capturada do ambiente monitorado, permitindo o surgimento de aplicações mais específicas. Esse novo escopo possibilita o desenvolvimento das Redes de Sensores Visuais sem Fio (RSVSF). Tais redes aprimoram a percepção do mundo físico, fornecendo dados visuais (imagens e/ou vídeos) para inúmeras aplicações (COSTA; GUEDES, 2010).

Uma das metas esperadas para as RSVSF está na cobertura de um conjunto de alvos, que podem ser estáticos ou móveis. Um sensor equipado com câmera pode ser configurado e posicionado para realizar a cobertura de um ou mais alvos, desde que estes estejam no campo de visão do sensor em questão. Para muitas aplicações, a qualidade do projeto está relacionada diretamente com o modelo de monitoramento dos alvos desejados por parte dos sensores visuais.

A forma como os alvos devem ser monitorados está intrinsecamente relacionada com o objetivo da aplicação. Para algumas, se faz necessário que todos ou a maioria dos alvos estejam cobertos. Em contrapartida, outras aplicações necessitam que os alvos estejam cobertos por mais de um sensor, tornando-os redundantes. Nesse seguimento, algumas aplicações podem necessitar de múltiplas visões do mesmo alvo, seja por questões de perspectivas de cobertura (ALAEI; BARCELO-ORDINAS, 2010) ou por questão de disponibilidade (COSTA et al., 2014c). Neste caso, quando os sensores visuais podem alterar sua orientação e são rotacionáveis, a cobertura redundante pode ser otimizada promovendo melhor qualidade nas aplicações (COSTA et al., 2014b).

O problema de cobertura redundante pode ser tratado a partir de dois objetivos distintos, a cobertura e a redundância. A cobertura de um alvo acontece quando o alvo está dentro do campo de visão de um sensor visual. Por outro lado, a redundância é obtida quando dois ou mais sensores visuais cobrem um determinado alvo (COSTA; GUEDES, 2010). Diante do problema de maximização da cobertura redundante nas redes de sensores visuais sem fios, a otimização é algo crucial para garantia da qualidade de serviço da aplicação, além de auxiliar o decisor a rotacionar corretamente os sensores implantados, visto que pode ser muito custoso realizá-los de forma manual. Outra motivação para a cobertura redundante é minimizar o consumo de energia dos nós sensores, utilizando algoritmos que provêm redundância aos alvos, fazendo com que os nós sensores redundantes possam ser desligados prolongando a vida útil da rede (JIA et al., 2009; HURUIALÁ; URZICĂ; GHEORGHE, 2010).

A maximização da cobertura redundante (MCR) é um problema que pode ser resolvido de diversas formas. Uma dessas abordagens é utilizar um algoritmo guloso para otimizar a orientação dos sensores visuais com o objetivo de maximizar a cobertura

redundante, embora alguns alvos possam ficar descobertos (COSTA et al., 2014b). Isso ocorre porque os algoritmos gulosos buscam soluções ótimas locais, neste caso, buscando a melhor orientação de cada sensor de forma individual, deixando de lado possíveis soluções consideradas ótimas globais.

O problema de maximização da cobertura redundante pode ser visto como um problema de otimização multiobjetivo, com objetivos bem definidos como cobertura e redundância. Em geral, os problemas multi-objetivos envolvem objetivos contraditórios, e no caso em questão, a cobertura redundante pode possuir contrariedade em seus objetivos, quando, por exemplo para garantir a cobertura de um alvo é preciso retirar a redundância de outro alvo. De forma contrária, caso haja a necessidade de obter maior redundância na rede, a cobertura passa a ser comprometida.

Um dos algoritmos largamente aplicados para problemas multiobjetivos são os Algoritmos Evolutivos (AE), uma meta-heurística de otimização inspirada nos processos evolutivos (ZITZLER; LAUMANN; BLEULER, 2004). O AE opera em uma população de soluções candidatas através de dois princípios básicos: seleção e variação. Enquanto o primeiro favorece as melhores soluções na competição por seleção, o segundo gera novas soluções através de recombinação e mutação das características herdadas das soluções selecionadas. Dessa forma, espera-se obter melhores resultados na resolução do problema de otimização pela aplicação de Algoritmos Evolutivos.

Os AE podem ser utilizados para tratar de problemas multiobjetivos para diversas aplicações, sendo que o desenvolvimento dessas heurísticas torna-se uma excelente opção para aproximar possíveis soluções globais, proporcionando melhoria na qualidade das aplicações, em especial na otimização da maximização da cobertura redundante das RSVSF.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver soluções para o problema de maximização de cobertura redundante dos alvos em uma Rede de Sensores Visuais sem Fio, aplicando algoritmos evolutivos para otimizar a orientação dos sensores visuais.

1.1.1 Objetivos Específicos

Formalizar o problema de maximização da cobertura redundante, definindo matematicamente os objetivos a serem otimizados.

Propor e desenvolver um algoritmo evolutivo de otimização *a-priori*, seguindo a abordagem lexicográfica para resolução do problema de maximização da cobertura redundante.

Propor e desenvolver um algoritmo evolutivo de otimização *a-posteriori*, seguindo a técnica NSGA-II para resolução do problema de maximização da cobertura redundante.

Realizar a otimização de parâmetros dos algoritmos evolutivos propostos.

Avaliar os algoritmos evolutivos desenvolvidos para otimizar a maximização da cobertura redundante das redes de sensores visuais sem fio.

Comparar o desempenho dos algoritmos desenvolvidos com outras estratégias de otimização.

1.2 Organização do Trabalho

No Capítulo 1, apresentamos uma busca na literatura sobre os temas abordados para embasamento teórico desse trabalho, além de mostrar alguns trabalhos que tratam do problema de cobertura redundante em RSVSF e outros que mostram o uso de algoritmos evolutivos para as mesmas. O capítulo 3 apresenta a metodologia utilizada, formalizando o problema de otimização proposto, além de mostrar as técnicas e procedimentos, ferramentas utilizadas e as características dos algoritmos desenvolvidos. No capítulo 4 são apresentados os resultados obtidos pelos algoritmos propostos e por fim, no capítulo 5 são apresentadas as considerações finais.

Capítulo 2

Revisão de literatura

2.1 Redes de Sensores sem Fio

Recentes avanços tecnológicos possibilitaram o desenvolvimento de pequenos sensores capazes de coletar diversas informações do meio ambiente. Esses sensores geralmente possuem um custo menor com relação aos sensores tradicionais. Sua gama de utilização para coleta de informações é bastante ampla, podendo ser utilizados em áreas como: saúde, militares, trânsito, indústria, geográficas (marítimas, terrestres) dentre outras áreas (YICK; MUKHERJEE; GHOSAL, 2008; ABBASI; YOUNIS, 2007). Geralmente, esses sensores sem fio possuem poucos recursos computacionais, sendo compostos por processador, memória, bateria, dispositivo de sensoriamento e um transmissor. Isso deve-se ao fato dos sensores possuírem um custo relativamente baixo (YICK; MUKHERJEE; GHOSAL, 2008).

Numerosas aplicações civis e militares podem fazer uso das RSSF. Um sensor sem fio pode ser implantado, por exemplo, para medir variações meteorológicas como temperatura e pressão. Essas medidas podem ser utilizadas para detectar fenômenos naturais, e até mesmo prevenir alguns desastres naturais (ABBASI; YOUNIS, 2007). Em situações que podem ocorrer desastres naturais como terremoto, as redes de sensores podem ser usadas para mapear seletivamente as regiões afetadas que dirigem unidades de resposta de emergência para sobreviventes (BOYINBODE et al., 2010). Nas aplicações militares, pode-se utilizar sensores sem fio para identificar e detectar intrusão de tropas inimigas, percebendo a movimentação de tanques e tropas coordenando e correlacionando espacialmente (YICK; MUKHERJEE; GHOSAL, 2008), e até mesmo detectar gases químicos ou presença de micros agentes (ABBASI; YOUNIS, 2007). Outras aplicações civis tais quais biomedicina, monitoramento de trânsito e indústria também podem fazer uso dos sensores sem fio (ABBASI; YOUNIS, 2007). Na biomedicina, implantes cirúrgicos podem auxiliar e monitorar o *status* da saúde do paciente. No monitoramento de trânsito, sensores de velocidade inibem os motoristas excederem o limite de velocidade permitido nas vias.

Nas indústrias, pode-se fazer uso de diversos sensores sem fio tendo como exemplo o controle de entrada dos colaboradores, detecção de incêndio e dentre outras.

Uma rede de sensores sem fio geralmente é composta por nós sensores (também conhecidos por sensores genéricos), nós *gateways* (ou *sinks*) e servidor de aplicação. Os nós sensores são utilizados para coletar grandezas físicas (temperatura, pressão, umidade, entre outros) do ambiente monitorado e os *sinks* possuem a função de coletar essas informações e transmiti-las para o servidor da aplicação, que é responsável pelo controle e monitoramento da rede (YICK; MUKHERJEE; GHOSAL, 2008). Os nós sensores são distribuídos no ambiente e os *sinks* ficam entre eles. Em geral, os *sinks* possuem maior poder de processamento, alcance e consumo de energia com relação aos nós sensores (BORGES; VELEZ; LEBRES, 2014). A topologia dessa rede pode ser representada na figura 2.1.

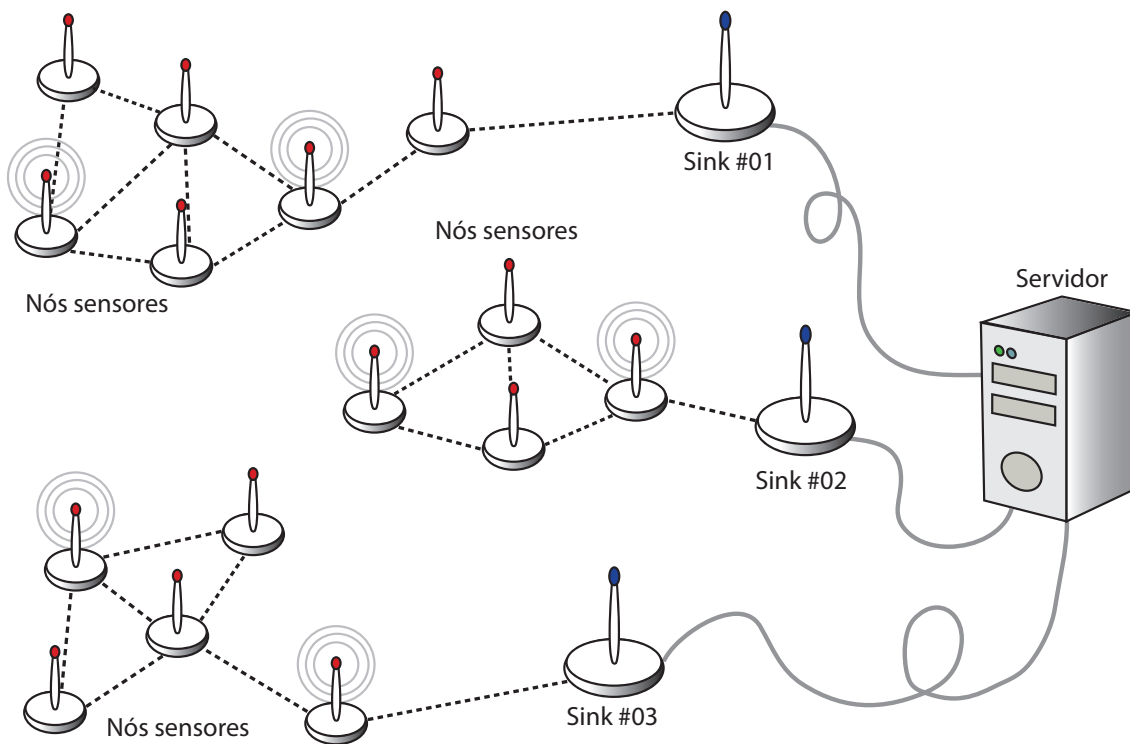


Figura 2.1: Topologia de uma Rede de Sensores sem Fio.

A distribuição dos nós sensores em uma RSSF pode ser determinística ou aleatória (figura 2.2). Essa distribuição depende exclusivamente da aplicação. Em alguns casos, os sensores precisam ser implantados em lugares específicos, onde deseja-se obter informações particulares de uma determinada região. Em outros casos, os sensores precisam ser distribuídos aleatoriamente na região de monitoramento para obter uma otimização específica, a exemplo de maximização ou minimização de cobertura.

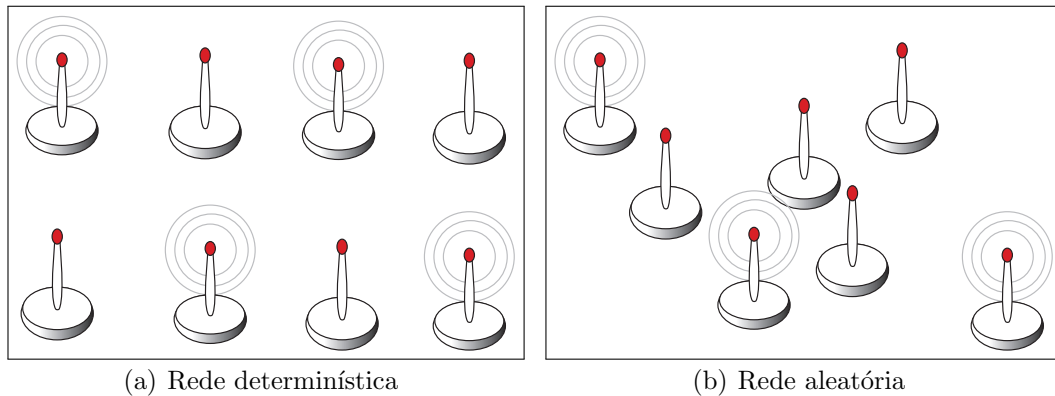


Figura 2.2: Tipos de distribuição em uma rede de sensores sem fio.

As RSSF podem ser de dois tipos: homogêneas ou heterogêneas (ABBASI; YOUNIS, 2007). Nas redes homogêneas, todos os sensores possuem a mesma capacidade em termos computacionais, comunicação e potência. Geralmente, utilizam-se sensores homogêneos quando a aplicação necessita coletar apenas um tipo de informação (YICK; MUKHERJEE; GHOSAL, 2008). Nas RSSF heterogêneas, os nós sensores podem diferenciar-se uns aos outros. Neste caso, quando as aplicações necessitam capturar diversas informações da área de monitoramento, faz-se uso de sensores variados no qual cada um pode capturar apenas um tipo de informação (mono), ou sensores que podem estar embarcados com diversos dispositivos de sensoriamento (multi) (ABBASI; YOUNIS, 2007).

O problema de projetar uma arquitetura de rede escalável é primordial. A maioria das propostas para as redes de sensores sem fio se baseiam em uma arquitetura plana e homogênea, em que cada sensor tem as mesmas capacidades físicas e só podem interagir com sensores vizinhos. Tradicionalmente, a pesquisa sobre algoritmos e protocolos para redes de sensores tem se concentrado na escalabilidade, ou seja, como projetar soluções cuja aplicabilidade não se limite ao tamanho crescente da rede. Topologias planas nem sempre podem ser adequadas para lidar com a quantidade de tráfego gerado por aplicativos que trabalham com informações visuais (AKYILDIZ; MELODIA; CHOWDHURY, 2007).

Várias tarefas precisam ser executadas para o correto funcionamento de aplicações das redes de sensores sem fio. Essas tarefas estão divididas em 3 grandes grupos: sistemas, serviços e protocolo de comunicação. Em aplicações distribuídas, os nós sensores possuem seus sistemas individuais embarcados. Nas aplicações centralizadas, o sistema é controlado pelo servidor de aplicação. Os serviços provêm de melhorias dos próprios sistemas, tais como melhoria na rede de sensores, redução de consumo da energia e otimização dos sistemas. Os protocolos são responsáveis pela comunicação e troca de informações entre os sensores (YICK; MUKHERJEE; GHOSAL, 2008). Esses protocolos são divididos em 5 camadas, sendo elas camada de Aplicação, Transporte, Rede, Enlace de dados e Física. O protocolo é desenvol-

vido utilizando as camadas para se intercomunicarem promovendo um serviço de comunicação fim a fim (AKYILDIZ; MELODIA; CHOWDHURY, 2007).

2.2 Redes de Sensores Visuais sem Fio

Algumas aplicações necessitam capturar dados visuais para realizarem tomadas de decisões, assim câmeras são incorporadas nos sensores sem fio, permitindo uma nova gama de aplicações. Formam-se Redes de Sensores Visuais sem Fio (RSVSF) (COSTA; GUEDES, 2010; CHAUDHARY; PUJARI, 2009). Essas redes aprimoram a coleta de informações sobre o mundo físico possibilitando melhorias em diversas aplicações como vigilância, monitoramento de desastres, trânsito, monitoramento de vida selvagem, assistência especializada para crianças, idosos e deficientes, controle de processos industriais e serviços de localização (COSTA; GUEDES, 2010).

Os componentes de *hardware* de uma RSVSF consistem em pequenos sensores com câmeras, embarcados com processador e transmissores sem fio (YAP; YEN, 2014). As principais diferenças das RSVSF com relação as RSSF tradicionais podem ser definidas através das seguintes características:

1. Requerimento de cobertura do campo de visão para os nós fontes: Nas redes de sensores tradicionais, quando ocorre um evento, os nós que estão dentro da faixa de detecção podem identificar o evento, tornando-se os nós fontes de dados que irão transmitir as informações capturadas para o *sink*. Entretanto, nas RSVSF, além do intervalo de detecção, outro fator que deve ser considerado é o campo de visão (*Field of View (FoV)*), a região espacial detectável pela câmera do sensor (YAP; YEN, 2014).
2. Consumo de largura de banda da rede: Os dados visuais possuem um tamanho maior em relação aos dados escalares. Com isso, o consumo de banda da rede tende a aumentar tornando a transmissão de dados desafiante em um espectro de frequência limitado (YAP; YEN, 2014).
3. Colisão na transmissão de dados: Quando ocorre um evento, os sensores vizinhos que também detectam o evento podem transmitir as informações para os *sinks* ao mesmo tempo. Como os sensores podem estar geograficamente próximos uns dos outros, as colisões podem ocorrer quando os dados são transmitidos ao mesmo tempo. Além disso, por conta do tamanho dos dados visuais, os sensores precisam transmitir uma sequência de pacotes para enviar todos os dados visuais capturados, aumentando ainda mais a probabilidade de colisão. Novos esquemas de transmissão e roteamento devem ser planejados para solucionar esse problema das RSVSF (YAP; YEN, 2014).
4. Processamento de dados multimídia: Quando comparado com o processamento de dados escalares das RSSF, os dados visuais são mais complexos e demandam mais recursos de *hardware* como poder de processamento e *buffer* de memória.

Para não consumir tantos recursos de *hardware* dos nós sensores, o tamanho dos dados visuais pode ser reduzido. Essa redução pode ser realizada através de técnicas de compressão feitas por um único nó sensor ou por métodos de processamento multimídia para eliminar as partes redundantes entre vários nós sensores. Deve-se considerar a capacidade de *hardware* dos nós sensores para projetar tais técnicas de compressão (YAP; YEN, 2014).

5. Cobertura de sensores em RSVSF: Devido ao ângulo (α) relativo dos nós sensores ou aos possíveis obstáculos na rede, um evento pode não ser capturado mesmo se estiver dentro do *FoV* dos sensores visuais. Com isso, algumas informações importantes podem não ser capturadas por conta da limitação do ângulo da câmera ou pelo obstáculo que bloqueia a visão. Para problemas com oclusão, mais sensores visuais podem ser requisitados para cobrir a área que não pode ser vista (YAP; YEN, 2014).

2.2.1 Sensores Visuais

O diferencial das RSVSF em relação as tradicionais RSSF está diretamente relacionado a área em que cada tipo de sensor pode cobrir. Nas RSSF os sensores podem capturar informações do ambiente externo que estejam dentro do seu raio de cobertura, que pode ser modelado de forma simplificada como uma circunferência, no qual o posicionamento do sensor está situado no centro (AKYILDIZ; MELODIA; CHOWDHURY, 2007), enquanto os sensores visuais sem fio possuem direção de monitoramento, dependendo das configurações das câmeras embarcadas nos sensores, e contam apenas com o campo de visão (AKYILDIZ; MELODIA; CHOWDHURY, 2007; ALMALKAWI et al., 2010; YAP; YEN, 2014). Os sensores visuais costumam ter um alcance maior com relação aos sensores tradicionais. Contudo, a qualidade da aplicação pode ser relacionada diretamente a orientação que cada sensor possa possuir.

Existem no mercado algumas opções de sensores visuais. Os sensores do tipo *pan-tilt-zoom* (PTZ) podem rotacionar tanto horizontalmente (*pan*) quanto verticalmente (*tilt*), além de modificar sua profundidade de visão *zoom* (MUNISHWAR; ABU-GHAZALEH, 2013; AKYILDIZ; MELODIA; CHOWDHURY, 2007). A rotação faz com que a região afetada pelo FoV dos sensores visuais sofram alteração, podendo ser ajustada por algum algoritmo de otimização.

As câmeras digitais de alta resolução e PTZ estão amplamente disponíveis no mercado. Entretanto, esses dispositivos sofisticados podem encontrar aplicações com alta qualidade para as redes de sensores visuais, podendo interferir diretamente no custo dessas aplicações. No entanto, dispositivos menos sofisticados que possuem baixo processamento e consumo de energia, e podem ser implantados largamente nas RSVSF devido ao baixo custo empregado neles (AKYILDIZ; MELODIA; CHOWDHURY, 2007). Por possuir um custo relativamente menor, muitos dispositivos podem ser adquiridos e implantados em diversos locais fornecendo infor-

mações visuais detalhadas de diferentes perspectivas, superando efeitos de oclusão, permitindo assim maior interação com o ambiente monitorado (YAP; YEN, 2014; AKYILDIZ; MELODIA; CHOWDHURY, 2007). A figura 2.3 apresenta alguns sensores visuais que podem ser encontrados no mercado.



Figura 2.3: Câmeras com função de zoom panorâmico PTZ.

Fonte: <https://www.axis.com/br/pt>.

Dentre os sensores apresentados, os que possuem a função *pan* (ajuste horizontal) são comumente utilizados por conta do custo de aquisição destes, modelagem da rede (geralmente bidimensional (2D)) e questões de complexidades envolvidas em algoritmos de otimização. A modelagem geométrica do campo de visão (*FoV*) de um sensor visual genérico é representado pela figura 2.4. Para simplificar, os sensores visuais são modelados em um plano 2D, tendo como perspectiva uma rede vista de cima (*top-view*).

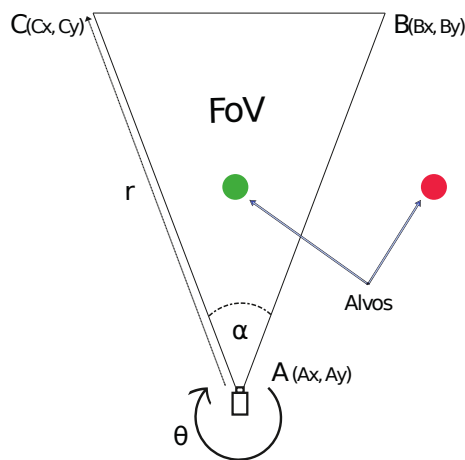


Figura 2.4: Representação de um nó sensor visual genérico.

O *FoV* do sensor visual é modelado como um triângulo isósceles, tendo o vértice A a posição geográfica (x, y) do sensor em questão. Complementando, α representa o ângulo de abertura do sensor visual, que pode variar entre os modelos, θ representa a orientação do sensor visual, sendo que pode-se assumir qualquer valor entre 0° e 360° . De modo geral, as câmeras não conseguem focalizar os objetos que estão muito próximos ou distantes. Então, torna-se comum a simplificação de modelos matemáticos que consideram a profundidade do campo partindo do vértice A até o alcance máximo (também conhecido como *Depth of Field - DoF*), no qual os objetos possam ser reconhecidos (COSTA; GUEDES, 2010). Desta forma, r representa o raio de alcance do sensor visual (*DoF*), que também pode variar entre os modelos disponíveis no mercado. Os vértices B e C são calculados utilizando funções trigonométricas (2.1), assim como a área do *FoV* (2.2), a área total do campo de visão (COSTA; VASQUES; PORTUGAL, 2017).

$$\begin{aligned} Bx_{(s)} &= Ax_{(s)} + r_{(s)} \cdot \cos(\theta_{(s)}) \\ By_{(s)} &= Ay_{(s)} + r_{(s)} \cdot \sin(\theta_{(s)}) \\ Cx_s &= Ax_{(s)} + r_{(s)} \cdot \cos((\theta_{(s)} + \alpha_{(s)}) \bmod 2\pi) \\ Cy_s &= Ay_{(s)} + r_{(s)} \cdot \sin((\theta_{(s)} + \alpha_{(s)}) \bmod 2\pi) \end{aligned} \quad (2.1)$$

$$Area_{FoV_{(s)}} = \frac{r_{(s)}^2 \cdot \sin(\alpha_{(s)})}{2} \quad (2.2)$$

Em particular, as câmeras podem captar imagens de objetos ou de partes de regiões que não estão necessariamente perto delas. No entanto, obviamente, a imagem pode ser capturada apenas quando há uma linha de visão desobstruída entre o evento e o sensor. Além disso, cada câmera percebe o ambiente ou o objeto observado a partir de um ponto de vista diferente e único, dadas as diferentes orientações (θ) e posições ($A(x, y)$) da câmera em relação ao evento ou região observada. O conceito de alcance de detecção é substituído com o campo de visão da câmera, isto é, o volume máximo visível a partir da câmera (AKYILDIZ; MELODIA; CHOWDHURY, 2007).

2.3 Visualização de Alvos

De forma geral, quando sensores visuais são implantados para visualizar alvos, podem haver diversos objetivos que irão depender da aplicação de monitoramento. Neste trabalho, é considerado que os sensores visuais devem visualizar o maior número de alvos possíveis para cumprir com os objetivos impostos. De forma geral, um alvo pode ser considerado como qualquer objeto, estático ou móvel, que deve ser monitorado por algum propósito.

Para um total \mathbf{A} de alvos, um determinado alvo a ($a = 1, \dots, \mathbf{A}$) tem uma posição (Ax_a, Ay_a) , como o centro dele, e por simplificação, um alvo a é definido como um ponto estático.

Existem diversos caminhos para verificar se um alvo a está dentro do campo de visão (FoV) de um sensor visual s , como a utilização de vetores (AI; ABOUZEID, 2006; COSTA; GUEDES, 2010) ou abordagens trigonométricas (MUNISHWAR; ABU-GHAZALEH, 2013; SILVA; COSTA, 2016). Para prover um mecanismo fácil deste tipo de verificação, este trabalho faz o uso de abordagens trigonométricas. Para isso, a distância euclidiana entre o sensor visual s considerado e o alvo a deve ser menor que o raio do sensor r_s . Depois disso, um alvo a é visto por um sensor s se e somente se o ponto (Ax_a, Ay_a) está completamente dentro do FoV do sensor s . Uma maneira prática para determinar isso é computar três triângulos utilizando os vértices do FoV do sensor (COSTA et al., 2014b), como pode ser visto na figura 2.5.

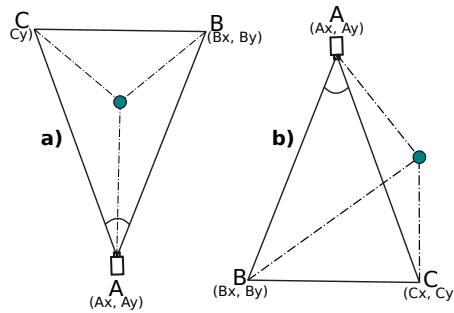


Figura 2.5: Verificando se um alvo está no FoV do sensor visual. O alvo está dentro em **a)**, e fora em **b)**.

A equação 2.3 calcula os três triângulos criados se o alvo está sendo visto.

$$\begin{aligned}
 \Delta AtB &= (Ax_{(s)} \cdot (By_{(s)} - Ty_{(t)}) + Bx_{(s)} \cdot (Ty_{(t)} - Ay_{(s)}) + Tx_{(t)} \cdot (Ay_{(s)} - By_{(s)})) \\
 \Delta AtC &= (Ax_{(s)} \cdot (Ty_{(t)} - Cy_{(s)}) + Tx_{(t)} \cdot (Cy_{(s)} - Ay_{(s)}) + Cx_{(s)} \cdot (Ay_{(s)} - Ty_{(t)})) \\
 \Delta BtC &= (Tx_{(t)} \cdot (By_{(s)} - Cy_{(s)}) + Bx_{(s)} \cdot (Cy_{(s)} - Ty_{(t)}) + Cx_{(s)} \cdot (Ty_{(t)} - By_{(s)}))
 \end{aligned}
 \tag{2.3}$$

ΔAtB calcula o vértice de A para B , ΔAtC calcula o vértice de A para C e ΔBtC calcula o vértice de B para C . Após isso, a função 2.4 retorna um valor booleano se o alvo está dentro (*true*) ou não (*false*) do FoV do sensor em questão.

$$cobre(s, a) = ((\Delta AtB + \Delta AtC + \Delta BtC) = Area_{FoV(s)})
 \tag{2.4}$$

2.4 Cobertura e redundância nas redes de sensores visuais sem fio

Em uma rede de sensores visuais sem fio, a avaliação da cobertura de alvos ou regiões por sensores é importante para avaliar a qualidade do monitoramento da mesma,

conforme a aplicação envolvida. Aplicações como rastreamento de alvos móveis utilizados por militares, podem exigir uma maior cobertura para rastrear adequadamente o alvo, enquanto aplicações como monitoramento ambiental ou de *habitat* podem tolerar uma menor cobertura (YICK; MUKHERJEE; GHOSAL, 2008). Para a obtenção de resultados mais confiáveis, deve-se considerar o aumento da cobertura nas redes, para isso, aumentando a quantidade de sensores implantados ou reposicionando-os em lugares específicos (YAP; YEN, 2014; YEN, 2013).

Pode-se classificar a cobertura de três formas: cobertura de área, cobertura de alvos e cobertura de barreira. A cobertura de área significa que cada alvo da região de interesse esteja dentro do campo de visão de pelo menos um nó sensor visual. Desta forma, é desejável implantar um número mínimo de câmeras dentro do campo de monitoramento para obtenção da cobertura total da rede (AMMARI, 2010). Cobertura de barreira refere-se a detecção de movimento através de uma barreira de sensores (AMMARI, 2010). Por fim, a cobertura de alvos refere-se a observar um conjunto fixo de alvos (COSTA et al., 2014b)

A cobertura de alvos acontece quando um alvo está dentro do campo de visão de um sensor visual. Em outras palavras, pode-se dizer que o alvo está sendo visto por pelo menos um nó sensor da rede, sobre qualquer perspectiva. Seguindo essa abordagem, o conceito de redundância pode ser definido quando um mesmo alvo está dentro do *FoV* de pelo menos dois sensores visuais, ou seja, um alvo deve ser visto por no mínimo duas câmeras (COSTA et al., 2014b). Na figura 2.6 pode-se observar de forma ilustrada uma representação de cobertura e redundância. Nesta figura, os círculos e hexágonos correspondem aos alvos, no qual os círculos vazios estão classificados como “descobertos”, os círculos com preenchimento classificam-se por “cobertos” e os hexágonos estão classificados como “redundantes”. Nota-se que, a depender do cenário, alguns alvos não serão cobertos por conta das limitações de alcance existentes nos sensores visuais (*DoF*).

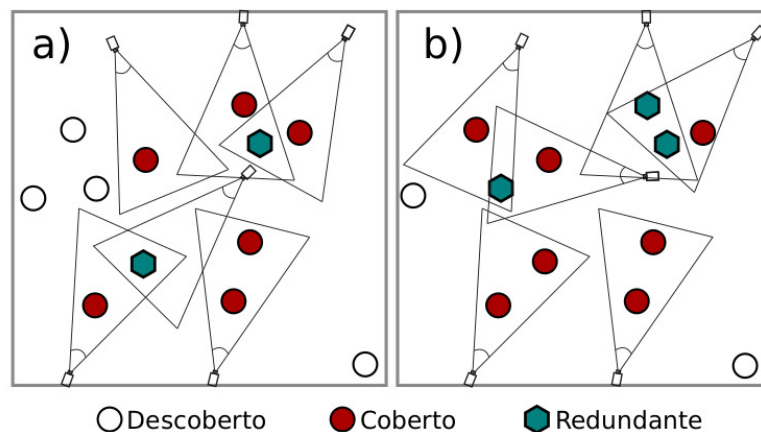


Figura 2.6: Ilustração de cobertura e redundância em uma RSVSF.

A cobertura de alvos por uma rede de sensores visuais sem fio precisa garantir que os alvos monitorados estejam do campo de visão (*FoV*) dos nós sensores visuais. A cobertura é importante pois afeta o número de sensores a serem implantados, o local na rede onde eles serão implantados, a conectividade e energia (YICK; MUKHERJEE; GHOSAL, 2008; YAP; YEN, 2014). Devido ao fato do *FoV* de um nó sensor visual possuir uma certa limitação considerando o ângulo de visão das câmeras, em alguns casos, torna-se necessário utilizar várias câmeras com orientações diferentes para cobrir completamente um conjunto de alvos, ou até mesmo um mesmo alvo (AKYILDIZ; MELODIA; CHOWDHURY, 2007).

A redundância na cobertura de alvos torna-se importante em uma rede de sensores visuais sem fio por muitos motivos, a depender da aplicação. Em aplicações que necessitam de muitas perspectivas do alvo, como em reconhecimento facial, quanto maior a redundância melhor é a qualidade dessa rede (ALAEI; BARCELO-ORDINAS, 2010). Outras aplicações que buscam minimizar o número de nós ativos na rede podem utilizar a redundância para desligar alguns nós redundantes, prolongando o tempo de vida útil da rede (CHARFI; WAKAMIYA; MURATA, 2009). Além disso, aplicar redundância em uma RSVSF pode atender a demanda de maior confiabilidade da aplicação, pois aumenta a chance da informação desejada ser capturada por um dos sensores. Outro fator importante da redundância é a tolerância a falhas. Nesse caso, quando um dos sensores falha, o outro mantém visibilidade perante os alvos monitorados desde que estes também estejam no *FoV* do sensor em questão.

Quando deseja-se obter o maior número de informações possíveis de uma RSVSF, é desejável que o seu nível de cobertura, e conseqüentemente de redundância seja alto. Desta forma, informações valiosas podem ser obtidas, mesmo que haja falhas em alguns nós-sensores (COSTA; GUEDES, 2010). Sendo assim, dois nós-sensores podem ser selecionados como redundantes quando um nível mínimo de cobertura com sobreposição entre as suas orientações, impactando de diferentes formas algumas aplicações de monitoramento visual (COSTA et al., 2014b). Aplicações de monitoramento crítico, por exemplo, consideram o maior grau de disponibilidade da rede, evitando que falhas inesperadas não prejudique na qualidade da aplicação (COSTA et al., 2014a).

2.5 Trabalhos Relacionados

Trabalhos na literatura têm discutido diversas questões relacionadas ao aumento da cobertura redundante nas RSVSF, relacionando com a temática dessa dissertação de diferentes maneiras. Além de questões de posicionamento de câmeras, geralmente modelado utilizando geometria analítica, questões sobre rotação de câmeras e cobertura de alvos são relevantes para os algoritmos propostos nesse trabalho.

Como já foi dito anteriormente, os sensores visuais em uma RSVSF podem ser

implantados de forma determinística ou aleatória. Para ambos os casos, é necessária uma abordagem de reposicionamento ou reorientação caso haja qualquer tipo de alteração nas RSVSF (aumento do número de alvos ou mudança de posição). De certa forma, serão necessários algoritmos de otimização para determinar a “melhor configuração” para as câmeras equipadas pelos nós sensores.

Uma área de pesquisa relevante está relacionada a medidas de cobertura e abordagens de avaliação em rede de sensores sem fio. Em resumo, as medidas de cobertura são desejadas ao avaliar a qualidade de detecção das RSSF (LIU; MA; ZHANG, 2008; ALAEI; BARCELO-ORDINAS, 2010), e também ao avaliar o quão bem os sensores implantados alcançam os objetivos das aplicações. Nessas circunstâncias, diversos trabalhos se concentram em abordagens diferentes para avaliar a qualidade da cobertura, considerando a área coberta, alvos vistos, cobertura de barreira, perspectivas sobre os alvos e até mesmo visualização perimetral (MUNISHWAR; ABU-GHAZALEH, 2013; COSTA; GUEDES, 2010; COSTA et al., 2015; SILVA; COSTA, 2016; COSTA; DURAN-FAUNDEZ, 2016; CHANG; HSIAO; CHANG, 2012). De certa maneira, o problema de maximização da cobertura redundante torna-se um problema multiobjetivo que consideram diferentes medidas de cobertura e redundância que precisam ser otimizadas.

Além de medidas de cobertura, a otimização de sensores para cobertura também têm sido investigadas (YEN, 2015). Quando os sensores possuem a capacidade de ajustar a área visualizada, a cobertura de detecção visual pode ser otimizada para obter o conjunto mínimo de sensores visuais que cobrem todos os alvos ou regiões requeridas. Como exemplo, o trabalho apresentado por Cai et al. (2007) calcula uma configuração ideal para sensores visuais com orientações ajustáveis, com base na ideia de que a cobertura visual é definida como conjuntos de capas não-disjuntas. Os autores Ai e Abouzeid (2006) propõem um mecanismo que calcula o número mínimo de nós que podem visualizar todos os alvos no campo monitorado, desligando os nós-sensores redundantes para economia de energia. Outros algoritmos promissores para otimizações de cobertura são discutidos por Munishwar e Abu-Ghazaleh (2013).

Numa perspectiva diferente, o trabalho de Yen (2013) propõe uma heurística para otimizar o custo da rede buscando a melhor configuração, incluindo a posição, orientação e quantidade de sensores, para cobrir todos os alvos. O algoritmo proposto realiza a implantação incremental, definindo a orientação e a posição de cada nó sensor. Posteriormente, é verificado se algum sensor pode ser removido sem ocorrer queda na taxa de cobertura. Basicamente o problema busca minimizar o número de nós sensores implantados na rede, garantindo a cobertura total.

Alguns trabalhos empregaram algoritmos evolutivos para abordar o problema da otimização da cobertura em redes de sensores sem fio. Navin et al. (2010) propuseram um algoritmo para melhorar a área visual total por sensores visuais, minimizando a sobreposição de detecção. A prioridade dos alvos foi considerada como um parâmetro de otimização em (WANG; NIU; SHEN, 2009), empregando um algoritmo evolutivo para calcular a otimização da cobertura e alcançar o número mínimo de

sensores visuais a serem ativados. Em (JIA et al., 2009), os autores utilizaram algoritmos evolutivos para a cobertura eficiente de energia nas redes de sensores sem fio. Para tanto, algoritmos evolutivos multiobjetivos foram utilizados para minimizar o número de nós sensores na rede, já que só é necessário ativar aqueles sensores que são utilizados no momento da coleta, preservando assim o consumo de energia e proporcionando o prolongamento da vida útil da rede. Eles consideraram a taxa de cobertura da rede e o número de nós ativos para realizar o cálculo de aptidão. Contrário ao nosso trabalho, a proposta deles buscam aumentar a cobertura de área da RSSF. Atualmente, o uso de algoritmos evolutivos nas RSVSF possuem diversas aplicações que podem variar de otimização da cobertura até economia de energia, mas, do melhor modo, não foram encontradas abordagens que busque tratar o problema de maximização de cobertura redundante, sendo este trabalho o pioneiro em abordagens evolutivas.

2.5.1 Algoritmos Gulosos para Otimização da Cobertura Redundante em Rede de Sensores Visuais sem Fio

Os algoritmos gulosos (*greedy algorithms*) são heurísticas que selecionam a melhor solução do momento, não levando em consideração possíveis prejuízos futuros, tornando-se um algoritmo de busca local, mas com esperança de obter ótimos globais. Seu grande problema é não explorar toda a região factível do espaço de busca, fazendo com que seja mais difícil encontrar as soluções ótimas globais (KORTE; LOVÁSZ; SCHRADER, 2012). Por possuir um certo nível de simplicidade, problemas de combinação linear ou que exigem baixa complexidade computacional podem ser resolvidos com essas abordagens. Busca em árvore de decisão (RASMUSSEN; SHIMA, 2008; LAI; LIN, 2009; BROWNE et al., 2012) e até mesmo grafos (SKIENA, 1990; PEIXOTO; COSTA, 2017) são exemplos de problemas que podem ser resolvidos por abordagens gulosas.

Outra área de pesquisa relacionada ao problema de maximização da cobertura redundante é a redundância, especialmente para avaliação e aprimoramento da disponibilidade (COSTA et al., 2014a). Neste contexto, a maximização da cobertura redundante surge como um mecanismo para calcular a redundância ativa, com benefícios para muitas aplicações de monitoramento.

O algoritmo *Centralized Prioritized Greedy Algorithm* (CPGA) foi proposto por Costa et al. (2014b) e tem como proposta aumentar a cobertura redundante das RSVSF, tornando-se o trabalho pioneiro a tratar o problema de otimização tratado nessa dissertação. De maneira semelhante ao trabalho desenvolvido nessa dissertação, a proposta de Costa et al. (2014b) considera que todos os alvos e sensores implantados na rede são fixos, podendo alterar apenas a orientação do sensor visual no processo de otimização. Para determinadas aplicações, alguns alvos possuem um nível de importância maior que outros. Neste caso o CPGA, como o próprio nome sugere, realiza o processo de otimização considerando a importância que cada alvo

possui, e quanto maior for o nível de importância, maior prioridade aquele alvo terá para ser coberto.

O CPGA considera um conjunto finito de orientações que podem ser adotadas por cada sensor visual, considerando que cada nó-sensor adotará a melhor configuração que maximize a soma das prioridades dos alvos cobertos. A quantidade máxima permitida é de $360^\circ/\alpha$, ou seja, se α for igual a 45° , o CPGA terá seis possíveis orientações para realizar a otimização. Isso pode ser um problema, pois às vezes torna-se necessário realizar um pequeno ajuste na orientação para obter resultados mais relevantes.

Proposto por Silva e Costa (2016), o algoritmo *Enhanced Centralized Prioritized Greedy Algorithm* (ECPGA) é uma adaptação da proposta de Costa et al. (2014b). A primeira adaptação realizada foi aumentar a possibilidade de possíveis regiões de monitoramento. Para isso, o ECPGA faz uma sobreposição no *FoV* do sensor visual em questão. A segunda adaptação foi ampliar a quantidade de alvos cobertos. O CPGA tenta encontrar a melhor configuração para cada sensor individualmente, podendo, em alguns casos, deixar de cobrir alguns alvos em específicos. Sendo assim, o ECPGA verifica quais alvos ainda não foram cobertos para analisar a possibilidade de cobri-los, sem que haja piora na cobertura total da rede.

Também proposto por Silva e Costa (2016), o *Redundant Coverage Maximization Algorithm* (RCMA) realiza uma abordagem diferente de seus antecessores. Tanto o CPGA quanto o ECPGA realizam a otimização considerando cada sensor de forma individual, analisando através das possibilidades de orientação que cada algoritmo possui, e qual a orientação que cobre o maior número possível de alvos. Já o RCMA busca solucionar a cobertura total dos alvos, para em seguida, realizar a otimização de redundância.

De forma geral, abordagens gulosas são fáceis de implementar, mas podem ter resultados não satisfatórios. Como alternativa, algoritmos evolutivos podem obter melhores resultados, sobretudo quando há múltiplos objetivos de otimização. Nessa perspectiva, o problema de maximização da cobertura redundante, que visa encontrar a melhor configuração quando todos os sensores estão ativos (ao contrário de (YEN, 2013)), pode ser resolvido com algoritmos evolutivos, como proposto neste trabalho.

2.6 Otimização Multiobjetivo

O processo de otimizar sistematicamente ou simultaneamente um conjunto de funções objetivo é chamado de otimização multiobjetivo (MARLER; ARORA, 2004). Muitos problemas do mundo real envolvem múltiplas medidas de desempenho, ou objetivos, que precisam ser otimizados ao mesmo tempo. Em alguns casos, os objetivos podem ser otimizados isoladamente, e informações obtidas sobre a melhor

solução podem ser alcançadas em cada perspectiva do desempenho (FONSECA; FLEMING, 1995). No entanto, em geral, os objetivos são conflitantes entre si, não permitindo que um objetivo possa ser melhorado sem comprometer os demais.

O problema de Otimização Multiobjetivo (*Multiobjective Optimization, MOO*) pode ser modelado por um vetor de variáveis de decisão que satisfaça algumas restrições, e otimiza simultaneamente uma função vetorial cujos elementos representam as funções objetivo de todos os tomadores de decisão. Essas funções formam uma descrição matemática, cujo critérios de desempenho geralmente são conflitantes entre si (maximizar e minimizar os objetivos) (COELLO, 1999).

Em geral, um problema multiobjetivo que busque maximizar os objetivos, pode ser descrito matematicamente como mostrado na equação (2.5).

$$\begin{aligned} & \underset{x}{\text{minimize (ou maximize)}} && f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ & \text{sujeito a} && x \in X \end{aligned} \tag{2.5}$$

onde $f(x)$ é um vetor n -dimensional, $f_i(x)$ é a i -ésima função objetivo a ser otimizada, x é o vetor de decisão e X é a região viável (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008).

Os problemas multiobjetivos possuem duas abordagens gerais: a priori e a posteriori. A primeira (*a priori*) transforma todas as funções objetivo individuais em uma única função composta ou em uma sequência de problemas mono-objetivos, na qual são determinadas as preferências de cada objetivo, antes de iniciar o processo de otimização. Alguns exemplos de abordagens de otimização *a priori* são a lexicográfica e a combinação de *fitness* linear ou não-linear (COELLO et al., 2007). A segunda abordagem (*a posteriori*) determina um conjunto de soluções ótimas de Pareto ou um subconjunto representativo, a partir do qual é feita a escolha da solução após o processo de otimização (KONAK; COIT; SMITH, 2006). Algumas técnicas para otimização *a posteriori* são: amostragem independente, seleção de critério, seleção de agregação, seleção baseada em Pareto, Seleção baseada em nicho e *ranking* de Pareto, seleção baseada em Pareto-tema, seleção baseada em elite do Pareto e seleção híbrida (COELLO et al., 2007).

Para entender as técnicas de otimização multiobjetivo é necessário realizar algumas definições dos conceitos para utilização em problemas multiobjetivos (QU et al., 2017).

1. Conjunto de soluções viáveis: Para $x \in X^*$, se x satisfizer todas as restrições, x é uma solução viável dentro do espaço de busca X^* , e o conjunto de todas as soluções viáveis é denotada como X_f , onde $X_f \subseteq X^*$.
2. Domínio (Pareto): Dadas duas soluções viáveis k_1 e k_2 de um problema de minimização, é dito que k_1 (Pareto) domina k_2 (denotado $k_1 \prec k_2$), se $\forall j \in \{1, 2, \dots, n\}$, $f_j(k_1) \leq f_j(k_2)$, $\exists i \in \{1, 2, \dots, n\}$, $f_i(k_1) < f_i(k_2)$.

3. Conjunto de Pareto-ótimo: Para uma solução viável $x \subseteq X$, se não houver outra solução viável $x' \subseteq X$ satisfazendo $x' \prec x$, é dito que x não é dominado com respeito a X , e essa solução viável x é definida como uma solução Pareto-ótima x^* . O conjunto de todas as soluções Pareto-ótimas é definido como o conjunto Pareto-ótimo denotado P^* , isto é, $P^* = \{x^* \mid \neg \exists x \in X : x \prec x^*\}$.
4. Fronteira de Pareto: A fronteira de Pareto é definida como P_F , onde $P_F = \{F(x^*) = [f_1(x^*), f_2(x^*), \dots, f_k(x^*)]^\top \mid x^* \in P^*\}$.

Dominância

A dominância de Pareto pode ser considerada como uma relação binária entre duas soluções (VOORNEVELD, 2003). Conforme o conceito apresentado no item 2, uma solução k_2 é dita como dominada por k_1 se somente se pelo menos um dos objetivos de k_1 for melhor que o mesmo em k_2 , além de k_1 ser melhor ou igual nos objetivos restantes.

$$\begin{aligned} & \underset{x}{\text{maximize}} && f(x) = (f_1(x), f_2(x), f_3(x)) \\ & \text{sujeito a} && x \in \mathbb{Z} \end{aligned} \tag{2.6}$$

Dada uma função qualquer que busque maximizar três funções objetivo (2.6), tem-se que:

$$\begin{aligned} a &\doteq f(x) = (3, 6, 6) \\ b &\doteq f(x) = (2, 5, 6) \\ c &\doteq f(x) = (2, 6, 5) \\ d &\doteq f(x) = (4, 8, 7) \end{aligned} \tag{2.7}$$

Na definição 2.7, $\{a, b, c, d\}$ são as soluções obtidas na maximização de $f(x)$. Neste caso, b e c são dominados por a e d , a é dominado por d , que por sua vez, domina todas as outras soluções.

Existem algumas definições complementares para dominância, a dominância fraca e a dominância forte. A dominância fraca acontece quando uma solução domina outra, mas têm valores iguais em pelo menos um dos objetivos. Já quando uma solução domina outra em todos e não há igualdade em nenhum dos objetivos, temos a chamada de dominância forte (GANGULY; SAHOO; DAS, 2013).

Para uma solução ser considerada Pareto-ótima esta deve satisfazer os requisitos de dominância, ou seja, a solução deve dominar todas as outras. No exemplo apresentado em 2.7, a solução d é considerada Pareto-ótima pois obedece os critérios apresentados. Desta forma, no conjunto de soluções $\{a, b, c, d\}$, d é a solução dominante.

2.6.1 Abordagem *a priori*

A otimização *a priori* é determinada por elementos mais simples que retornam um único objetivo (KONAK; COIT; SMITH, 2006), ou faz com que o decisor determine a importância relativa de cada objetivo e, assim, uma ou várias soluções que satisfaçam essa importância podem ser encontradas (GANDIBLEUX et al., 2004; COELLO et al., 2007). Essa abordagem articula as preferências dos objetivos individuais antes de iniciar o processo de otimização. Isto pode ser feito pela combinação dos objetivos por uma soma ponderada, ou ainda pela otimização sequencial de um objetivo por vez seguindo a ordem de prioridade destes (KONAK; COIT; SMITH, 2006), uma abordagem conhecida como lexicográfica.

Método de soma ponderada (Combinação de *fitness* Linear)

Nesse tipo de abordagem são atribuídos pesos w_i para cada função objetivo normalizada $f'_i(x)$, fazendo com que a otimização seja convertida para apenas um problema objetivo escalar, que pode ser representado como segue (KONAK; COIT; SMITH, 2006):

$$\text{maximize } f = w_1 f'_1(x) + w_2 f'_2(x) + \dots w_k f'_k(x) \quad (2.8)$$

onde $f'_i(x)$ é a função objetivo i normalizada e $\sum w_i = 1$. Esse método pode ser chamado de abordagem *a priori* desde que os pesos sejam fornecidos pelo usuário (KONAK; COIT; SMITH, 2006; COELLO et al., 2007). Geralmente, os pesos são apresentadas como um vetor $w = \{w_1, w_2, \dots, w_k\}$.

Essa técnica é utilizada geralmente quando os objetivos podem ser otimizados sem considerar os custos dos demais, fazendo assim, uma combinação linear que não envolve nenhuma sobrecarga adicional aos objetivos (COELLO et al., 2007). Uma das grandes dificuldades desse método é determinar seus pesos, a depender do objetivo, atribuir valores incorretos podem acarretar resultados não promissores (MARLER; ARORA, 2010). Por se tratar de uma abordagem não determinística, a mudança dos pesos promovem resultados diferentes.

De fato, ao utilizar o método de soma ponderada pode ser difícil distinguir entre ajustar os pesos para compensar as diferenças em magnitudes de uma função objetiva e ajustar os pesos indicando a importância relativa de cada objetivo, como é feito através dos métodos de ranqueamento (MARLER; ARORA, 2010).

Método lexicográfico

A abordagem lexicográfica organiza as funções objetivo em uma dada ordem de importância. Desta forma, os problemas de otimização são resolvidos um a um e

de forma sequencial (MARLER; ARORA, 2004). Assim, um *ranking* é estabelecido inicialmente para realizar a otimização através de uma ordem prioritária (COELLO et al., 2007).

A equação 2.9 apresenta a modelagem geral da abordagem lexicográfica.

$$\begin{aligned} & \underset{x \in X}{\text{maximize}} && F_i(x) \\ & \text{sujeito a} && F_j(x) \leq F_j(x_j^*), j = 1, 2, \dots, i-1, i > 1, i = 1, 2, \dots, k. \end{aligned} \quad (2.9)$$

onde i representa uma função em sua sequência preferencial, e $F_j(x_j^*)$ representa o valor ótimo da j -ésima função objetivo, encontrado na j -ésima iteração. Após a primeira iteração ($j = 1$), $F_j(x_j^*)$ não é necessariamente o mesmo que o valor máximo independente de $F_j(x)$, por que novas restrições podem ser obtidas (MARLER; ARORA, 2004).

Essa técnica de otimização é adequada somente quando é conhecida a importância de cada objetivo, quando comparados uns aos outros (COELLO et al., 2007).

2.6.2 Abordagem *a posteriori*

Na maioria dos casos é difícil para o decisor especificar uma preferência aproximada dos objetivos a serem otimizados. Desta forma, torna-se mais efetivo para o mesmo escolher dentre algumas soluções disponíveis, sem determinar, de início, a importância dos objetivos, e somente depois avaliar qual ou quais soluções são adequadas ao problema em questão (MARLER; ARORA, 2004).

A abordagem *a posteriori* não assume preferências entre objetivos e determina todo um conjunto de soluções ótimas de Pareto ou um subconjunto representativo, em que nenhum objetivo possui relevância maior em relação aos demais (KONAK; COIT; SMITH, 2006). Assim, a finalidade dessa abordagem é encontrar um conjunto ótimo de Pareto (KONAK; COIT; SMITH, 2006; QU et al., 2017). O decisor ou projetista escolhe a solução mais adequada ao problema em potencial a partir desse conjunto de soluções Pareto-ótimas, dessa forma, a tomada de decisão ocorre após o processo de otimização.

Para uma solução ser considerada Pareto-ótima, ela não deve ser dominada por qualquer outra solução no espaço de busca. Além disso, como é comum em problemas multiobjetivos, uma solução Pareto-ótima não consegue obter melhoras sem piorar pelo menos um dos objetivos. Esse processo é chamado de *trade-off* (ganho e perda) (KONAK; COIT; SMITH, 2006; MARLER; ARORA, 2004).

Um conjunto ótimo de Pareto é formado por todas as soluções viáveis dentro do espaço de busca (soluções não-dominadas). Quando aplicados ao mundo real, os conjuntos de Pareto-ótimo são preferíveis pelo fato de possuir mais de uma solução

(ao contrario da *a-priori*). Para um determinado conjunto ótimo de Pareto, os valores de função objetivo correspondentes no espaço de soluções são chamados de fronteira de Pareto (KONAK; COIT; SMITH, 2006; COELLO, 1999). O tamanho dos conjuntos ótimos de Pareto costumam ser variados, podendo aumentar conforme a quantidade de objetivos a serem otimizados (COELLO, 1999).

Algoritmos Evolutivos são uma das técnicas de otimização *a posteriori*, que buscam explorar as capacidades de uma população de soluções candidatas produzindo um conjunto de elementos de Pareto-ótimo (COELLO et al., 2007).

2.7 Algoritmos Evolutivos

Os Algoritmos Evolutivos (AE) representam uma classe de métodos de otimização estocásticos que simulam o processo de evolução natural (DEB, 2001). De acordo com Zuben (2000), os Algoritmos Evolutivos foram introduzidos inicialmente por John Holland a partir de 1975, com o objetivo de formalizar matematicamente e explicar rigorosamente os processos de adaptação em sistemas naturais, e desenvolver sistemas artificiais (simulados em computador) que retivessem os mecanismos originais encontrados nos sistemas naturais.

Existem muitas variantes diferentes dos algoritmos evolutivos na literatura. Todas essas variantes trazem uma ideia em comum: dada uma população de indivíduos, a pressão do meio ambiente faz com que a seleção natural (sobrevivência do mais apto) provoque um aumento de aptidão na população (EIBEN; SMITH, 2015). Como os algoritmos evolutivos operam com uma população de soluções, várias soluções candidatas podem ser obtidas em uma única execução (DEB, 1999).

Os algoritmos evolutivos operam de modo não determinístico, isto é, correspondem a um método heurístico de busca estocástica. Sendo assim, a partir de uma determinada entrada, pode-se obter resultados diferentes a cada execução, não obtendo garantias por soluções ótimas globais. No entanto, trata-se de um procedimento de otimização capaz de encontrar boas soluções globais ao explorar globalmente o espaço de busca, e, ao mesmo tempo intensificar a busca em regiões promissoras deste espaço, seguindo um processo iterativo até conseguir encontrar uma solução viável ou atingir um certo critério de parada.

Conforme Zuben (2000), os AE não podem ser vistos como algoritmos “prontos para uso”, e sim como uma heurística que pode ser adaptada para o contexto da aplicação em questão, formalizando suas restrições respectivas. Por isso, é fato que os algoritmos evolutivos podem ser aplicados em muitas áreas, como, por exemplo: Aplicações de planejamento; design; simulação e identificação; controle; classificação (BÄCK; FOGEL; MICHALEWICZ, 2000).

Um algoritmo evolutivo tradicional segue os passos representados na figura 2.7.

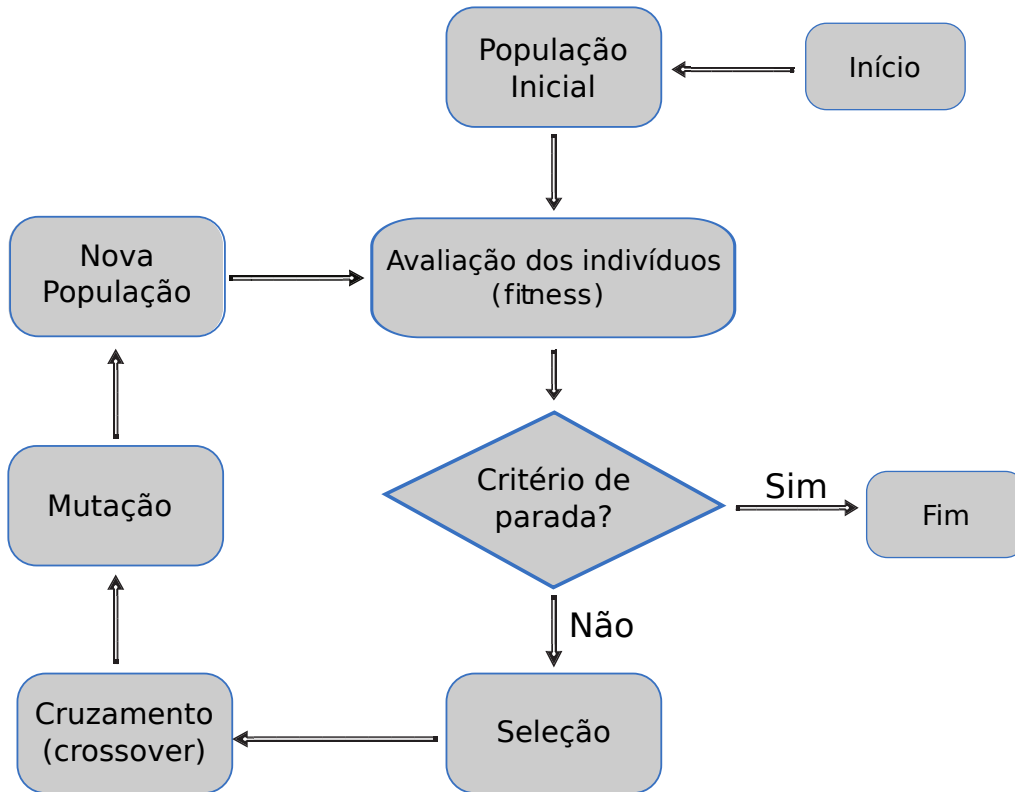


Figura 2.7: Fluxograma de um algoritmo evolutivo genérico.

2.7.1 Representação (Definição do Indivíduo)

Geralmente, o primeiro passo a ser realizado ao modelar um algoritmo evolutivo é definir sua representação. Ela possui a finalidade de realizar uma ligação entre o “mundo real” com o “mundo AE”, ou seja, realizar uma ponte entre o contexto do problema original e o espaço de resolução dos problemas em que ocorre a evolução (EIBEN; SMITH, 2015).

Como já foi dito, os algoritmos evolutivos buscam inspirações nos processos biológicos, e alguns termos da biologia genética são utilizados para denominar os componentes utilizados no seu desenvolvimento. Entre esses termos estão: fenótipo e genótipo. Nos AE’s, o fenótipo é representado por objetos que formam as possíveis soluções dentro do contexto do problema, e os genótipos têm a finalidade de codificá-los (BÄCK; FOGEL; MICHALEWICZ, 2000). Em outras palavras, um fenótipo é formado por um conjunto de especificações para uma solução do problema proposto, que é composto por suas características distintas, e genótipo é uma unidade codificada dessas especificações (EIBEN; SMITH, 2015).

Escolher a representação adequada é um dos passos mais importantes para iniciar a resolução do problema (EIBEN; SMITH, 2015). Para isso, se faz necessário escolher

o tipo de codificação que será utilizada pelo AE. A codificação do cromossomo (ou indivíduo) pode ser realizada de diversas formas, dentre elas:

1. Representação binária: Uma das primeiras propostas para resolução de problemas com uso de algoritmos evolutivos. Nessa representação, os genótipos são compostos por uma *string* ou dígitos binários (EIBEN; SMITH, 2015). Algumas aplicações não podem ser resolvidas por codificação binária. Neste caso, outras representações são propostas na literatura para se adequar ao contexto do problema.
2. Números inteiros: Para essa representação, o genótipo é composto unicamente por valores inteiros. Aplicações que não necessitam de aproximações numéricas ou até mesmo variáveis de decisão utilizam esse tipo de representação.
3. Valores reais: Em muitos casos, representar a solução candidata através de números reais torna-se o caminho mais sensato (EIBEN; SMITH, 2015). Quando se faz necessário a otimização de valores visando obter mais precisão, os números de pontos flutuantes são os mais indicados. Conforme Eiben e Smith (2015), o genótipo para uma solução com k genes é um vetor (x_1, \dots, x_k) com $x_i \in \mathbb{R}$.
4. Permutação: Consoante Bäck, Fogel e Michalewicz (2000), a permutação pode ser representada por um conjunto finito de elementos presentes em um vetor. Alguns problemas necessitam de uma ordem ou sequência no qual os eventos irão acontecer (EIBEN; SMITH, 2015). Na representação por permutação, geralmente utiliza-se valores (inteiros ou uma letra (A, B, ..., Z)) em uma sequência ordenada, sendo que esses valores não devem ser substituídos por quaisquer outros que não estejam no espaço de solução do problema.
5. Representação de máquina de estados finito: Conforme (BÄCK; FOGEL; MICHALEWICZ, 2000), uma máquina de estado finito é representada por uma sequência de instruções a ser executada. Um exemplo é a movimentação simples de um robô, podendo andar para frente, para trás, esquerda e direita. Nessa situação, pode-se imaginar que um robô precisa saber a sequência de comandos necessária para se locomover em um ambiente qualquer.

O tipo de codificação escolhida depende da aplicação proposta. A permutação é utilizada em problemas nos quais se faz necessário manter todas as características do cromossomo, como, por exemplo, o problema do caixeiro viajante (*Travel Salesman Problem - TSP* (BELLMORE; NEMHAUSER, 1968)), um problema NP-Difícil muito encontrado na literatura. Nesse problema, o caixeiro parte de um ponto de origem tendo que visitar um conjunto de cidades, visitando-as uma única vez cada, e ao final, voltar para o seu ponto inicial tendo por objetivo realizar o menor trajeto possível (BÄCK; FOGEL; MICHALEWICZ, 2000). Neste caso, o cromossomo é formado pela sequência de cidades a ser visitada, sendo que esses valores não podem ser alterados por outros valores que não estejam no vetor de soluções, mas sim, trocados de ordem.

2.7.2 População

O papel da população é gerar um conjunto de possíveis soluções candidatas (indivíduos) que passarão pelo processo de evolução. A população pode ser vista como multiconjunto de genótipos (EIBEN; SMITH, 2015; BÄCK; FOGEL; MICHALEWICZ, 2000). Em geral, a população inicial é gerada de forma aleatória. Se houver algum conhecimento inicial a respeito do problema proposto, pode ser usado como forma de inicialização da população (ZUBEN, 2000).

2.7.3 Função de Avaliação (Função *Fitness*)

O papel da função de avaliação é calcular a aptidão de todos os indivíduos da população para o problema em questão, considerando os requisitos e restrições nos quais estes terão que apresentar. O *fitness* constitui como a base para a seleção dos indivíduos facilitando as melhorias da população. Na visão da resolução de problemas, é uma função ou procedimento que atribui a qualidade aos genótipos. Geralmente, a função de avaliação é composta por uma medida de qualidade do fenótipo (EIBEN; SMITH, 2015).

2.7.4 Operadores de Seleção

Seleção é um mecanismo em computação evolutiva que possui como foco a procura de regiões promissoras no espaço de busca (EIBEN; SMITH, 2015). Esse é um dos principais operadores utilizados nos algoritmos evolutivos. O objetivo primário do operador de seleção é priorizar as melhores soluções em uma população (BÄCK; FOGEL; MICHALEWICZ, 2000). O papel da seleção é distinguir os indivíduos com base em suas qualidades, permitindo que os melhores indivíduos se tornem os genitores das próximas gerações (EIBEN; SMITH, 2015). Os operadores de seleção partem do princípio de escolha dos indivíduos para os processos de reprodução dos algoritmos evolutivos (ZUBEN, 2000). Este operador não cria uma nova solução, em vez disso, seleciona soluções relevantes tidas como “boas” em uma população e “apaga” outras soluções que não são relevantes (BÄCK; FOGEL; MICHALEWICZ, 2000).

Existem diversos métodos de seleção que podem ser utilizados pelos algoritmos evolutivos. A escolha do método de seleção depende unicamente da aplicação. Alguns operadores de seleção são apresentados nas seções seguintes.

Seleção Roleta

Neste método, os progenitores são selecionados de acordo com seu valor de *fitness*, obedecendo à ideia de que os indivíduos que possuírem melhores avaliações possuirão mais chances de serem selecionados. Esse processo é aplicado em um esquema

de roleta (figura 2.8), onde cada seção é proporcional ao valor de *fitness* de cada indivíduo, e quanto melhor for o valor de *fitness*, maior será a seção e, obviamente, mais chances o indivíduo terá de ser selecionado.

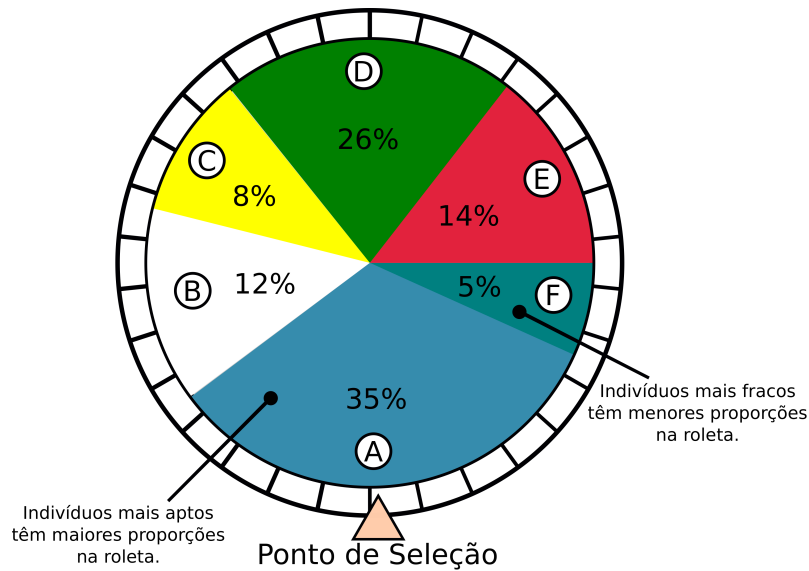


Figura 2.8: Seleção por roleta. As letras indicam a seção de adequação dos indivíduos, onde a região A possui maior probabilidade, e a região F possui menor probabilidade.

Seleção Torneio

Uma competição entre todos os indivíduos da população é realizada e deve favorecer os indivíduos com melhores avaliações (*fitness*) para gerar os indivíduos da nova população após variação por recombinação e mutação. Sendo assim, a seleção por meio de um torneio escolhe sucessivamente K indivíduos aleatoriamente, e determina dentre estes o indivíduo com melhor *fitness* como vencedor do torneio. Os vencedores são selecionados para reprodução e o procedimento é repetido até a nova população alcançar o tamanho definido.

Elitismo

O elitismo é utilizado geralmente antes de qualquer outro método de seleção (EIBEN; SMITH, 2015). Neste caso, a ideia de manter os melhores indivíduos de cada geração é predominante, garantindo assim a possibilidade de obtenção de soluções mais promissoras, não permitindo pioras no decorrer das gerações subsequentes.

2.7.5 Operadores de Recombinação (*Crossover*)

Nos Algoritmos Evolutivos, a recombinação é um processo complexo que ocorre entre um par de cromossomos. Dois cromossomos são escolhidos pelo método de seleção, e após isso, ocorre uma “quebra” em um ou mais locais correspondentes em cada cromossomo e os fragmentos são trocados e reparados. Isso resulta em uma recombinação de materiais genéticos que contribuem no surgimento de novos cromossomos, com a tendência de obter características melhores com relação aos seus progenitores (BÄCK; FOGEL; MICHALEWICZ, 2000). Este operador é considerado por muitos o processo mais importante dos algoritmos evolutivos (EIBEN; SMITH, 2015; LÜCKEN; BARÁN; BRIZUELA, 2014). Uma taxa de recombinação é utilizada para verificar se os progenitores irão atuar no processo de recombinação (EIBEN; SMITH, 2015).

Existem na literatura vários métodos de recombinação. A escolha do método de recombinação pode interferir diretamente no resultado da otimização, mas não é possível definir qual o melhor método a ser utilizado, visto que cada aplicação tem sua particularidade.

Alguns métodos de recombinação encontrados na literatura são apresentados a seguir.

Cruzamento de Ponto Único

O operador *crossover* de um ponto é um dos mais utilizados. São necessários dois indivíduos (pais) e com base em seus cromossomos são concebidos dois novos indivíduos (filhos) para a aplicação deste operador. Através de um ponto de corte definido aleatoriamente nos cromossomos dos pais, é realizada uma permuta de informações a partir do seguimento, permitindo a concepção de dois novos indivíduos (filhos) (figura 2.9) (EIBEN; SMITH, 2015; ZUBEN, 2000; BÄCK; FOGEL; MICHALEWICZ, 2000).



Figura 2.9: Exemplificação de cruzamento de ponto único.

Cruzamento de Dois Pontos

Segue-se a mesma abordagem do cruzamento de ponto único, mas com a aplicação de dois pontos de corte (figura 2.10).

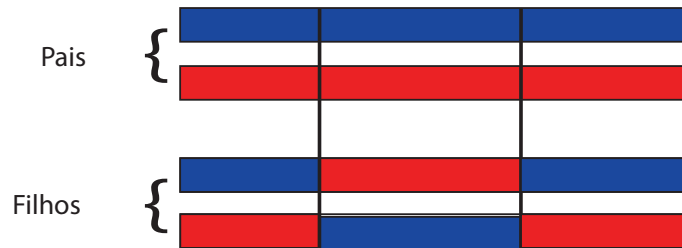


Figura 2.10: Exemplificação de cruzamento de dois pontos.

Cruzamento Uniforme

Para o cruzamento uniforme, o número de pontos de corte não é fixado (BÄCK; FOGEL; MICHALEWICZ, 2000). A cada k indivíduos selecionados (geralmente um par de genitores), são gerados novos indivíduos que recebem as informações genéticas dos progenitores de forma aleatória (figura 2.11), e tais informações são trocadas com uma taxa de probabilidade fixada (em geral, 50%) (EIBEN; SMITH, 2015).

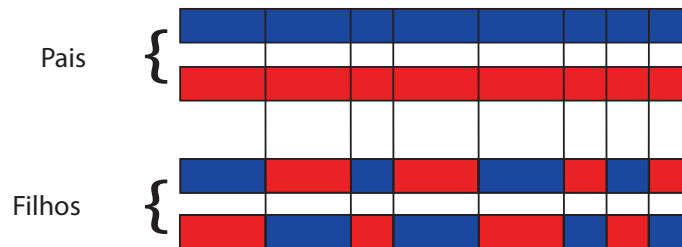


Figura 2.11: Exemplificação de cruzamento através do método uniforme.

Crossover Binário Simulado (*Simulated Binary Crossover (SBX)*)

Proposto por (AGRAWAL; DEB; AGRAWAL, 1995), esse operador simula a distribuição dos filhos através de recombinação de ponto único para codificação em binário, em um contexto de codificação em valores reais. Ele recebe como parâmetro a probabilidade de mutação e um índice de distribuição (η).

Conforme Deb e Beyer (2001), para computar as soluções filhas ($x_i^{1,t+1}$ e $x_i^{2,t+1}$) a partir de duas soluções pais ($x_i^{1,t}$ e $x_i^{2,t}$), alguns procedimentos devem ser realizados. Um fator de propagação β é definido como a diferença absoluta dos valores dos filhos

para os valores dos pais (2.10). Esse fator de propagação é calculado após a obtenção dos filhos.

$$\beta_i = \left| \frac{x_i^{2,t+1} - x_i^{1,t+1}}{x_i^{2,t} - x_i^{1,t}} \right| \quad (2.10)$$

O primeiro passo a ser realizado é criar um número aleatório $u_i \in [0, 1]$. Após isso, através de uma função de distribuição de probabilidade especificada, o valor de β_{qi} é encontrado de modo que a área sob a curva de probabilidade de 0 a β_{qi} seja igual ao número aleatório u_i sorteado. A distribuição de probabilidade usada para criar uma solução filho é derivada para ter um poder de busca similar ao *crossover* de ponto único, no qual é computada conforme a equação 2.11 (DEB; BEYER, 2001). O η controla a forma de como os filhos são distribuídos, quanto maior for o valor de η , maior será a proximidade dos filhos com os pais após processo de recombinação.

$$\rho(\beta_i) = \begin{cases} 0.5(\eta + 1)\beta_i^\eta, & \text{se } \beta_i \leq 1 \\ 0.5(\eta + 1)\frac{1}{\beta_i^{\eta+2}}, & \text{caso contrário} \end{cases} \quad (2.11)$$

A figura 2.12 mostra a importância de selecionar valores mais altos para η , pois, afeta diretamente na escolha dos filhos mais próximos aos pais. Nessa figura, os valores 2 e 5 são atribuídos a η com o intuito de mostrar as chances de escolhas dos valores dos filhos com relação aos valores dos pais. O índice de distribuição η deve ser um valor real não negativo (AGRAWAL; DEB; AGRAWAL, 1995). A equação 2.12 é utilizada para calcular β_{qi} , equiparando a área sob a curva de probabilidade igual ao número aleatório u_i .

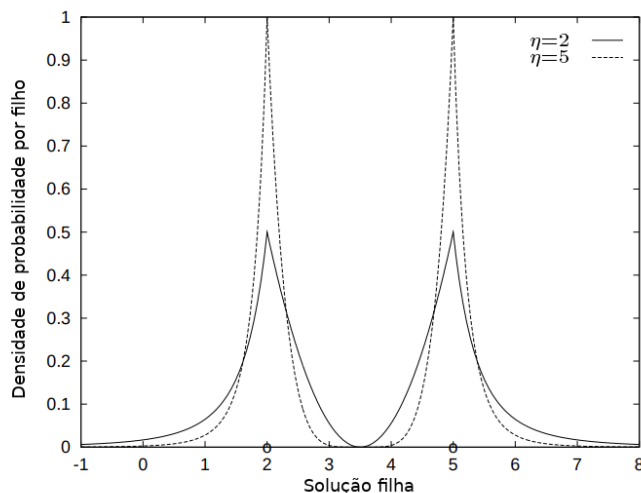


Figura 2.12: Distribuição da probabilidade para criar soluções filhas. As soluções pais estão marcadas como “o”. Adaptado de Deb e Beyer (2001)

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta+1}}, & \text{se } u_i \leq 0.5 \\ (\frac{1}{2(1-u_i)})^{\frac{1}{\eta+1}}, & \text{caso contrário} \end{cases} \quad (2.12)$$

As soluções filhas são calculadas conforme as equações 2.13 e 2.14 após a obtenção do valor de β_{qi} .

$$x_i^{1,t+1} = 0.5[(1 + \beta_{qi})x_i^{(1,t)} + (1 - \beta_{qi})x_i^{(2,t)}], \quad (2.13)$$

$$x_i^{2,t+1} = 0.5[(1 - \beta_{qi})x_i^{(1,t)} + (1 + \beta_{qi})x_i^{(2,t)}]. \quad (2.14)$$

Em resumo, para criar duas soluções filhas ($x_i^{1,t+1}$ e $x_i^{2,t+1}$) a partir de duas soluções pais ($x_i^{1,t}$ e $x_i^{2,t}$), o operador SBX deve, primeiramente, gerar um número aleatório (ui), em seguida calcular probabilidade de distribuição ($\rho(\beta_i)$) utilizando a equação 2.11, para em seguida realizar a normalização de β_{qi} (equação 2.12), e por último, computar as soluções filhas (equações 2.13 e 2.14).

2.7.6 Operadores de Mutação

Mutação é um nome dado aos operadores de variação que utilizam apenas um pai para gerar apenas um filho aplicando algum tipo de mudança aleatória em sua representação (genótipo). A forma de mutação depende do tipo de representação escolhida para a aplicação. Habitualmente, uma taxa de mutação é utilizada para definir se determinado indivíduo passará por esse processo (EIBEN; SMITH, 2015). Em geral, as taxas de mutação costumam ser relativamente baixas, pois, uma taxa muito alta pode fazer com que a população perca suas principais características de solução do problema em questão (EIBEN; SMITH, 2015). A principal ideia por trás da mutação é promover uma diversidade na população, sem que o progresso obtido seja destruído (ZUBEN, 2000).

Após o processo de recombinação, acontece uma mutação entre os genes para promover variação genética na população.

Alguns operadores de mutação são apresentados nas subseções seguintes.

Mudança de Ordem (*Exchange mutation (EM)*)

Utilizado geralmente em representação por permutação, esse processo de mutação altera o posicionamento dos genes trocando-os de lugar (figura 2.13).

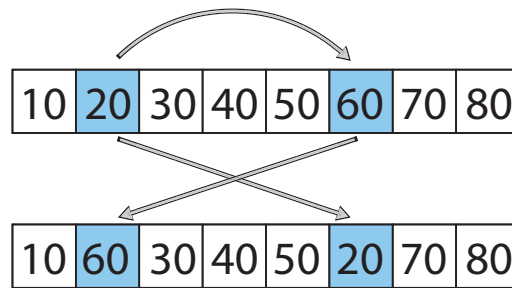


Figura 2.13: Mutaç o por mudana de ordem.

Mutaç o por Soma de Valores (*Mutation by Sum (SMV)*)

Essa mutaç o sorteia aleatoriamente $P\%$ das posi es do vetor e soma um pequeno valor aleat rio ao valor atual de cada posi o (2.14).

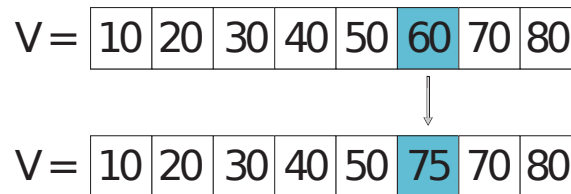


Figura 2.14: Mutaç o por soma de valores, onde V   o vetor e o  ndice V_5 foi escolhido aleatoriamente, somando 15 (tamb m escolhido aleatoriamente) ao seu valor atual, sendo $60 + 15 = 75$.

Mutaç o por Substitui o de Valores (*Mutation by Substitution (SM)*)

Nesse processo de mutaç o, seleciona-se uma posi o aleat ria do cromossomo e substitui-se o valor atual por qualquer outro valor aleat rio (figura 2.15)

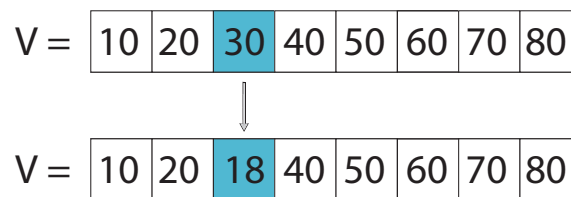


Figura 2.15: Mutaç o por substitui o de valores, onde V   o vetor e o  ndice V_2 foi escolhido aleatoriamente, alterando para 18 o seu valor atual.

Mutaç o Polinomial (*Polynomial Mutation (PM)*)

Da mesma forma que o SBX, o operador PM tenta simular a distribui o de mutaç o de *bit-flaps* codificada em bin rio, em vari veis de decis o de valores reais.

Neste operador, o η (índice de distribuição) é utilizado para fazer uma perturbação dos valores próximos ao pai. Essa perturbação é aplicada em uma distribuição de probabilidade para que os valores da direita e esquerda de uma variável seja ajustada, de modo que nenhum valor fora de um intervalo especificado $[a, b]$ seja criado. Esse intervalo (a e b) correspondem aos limites inferiores e superiores da variável (LIAGKOURAS; METAXIOTIS, 2013). Desta forma, o valor atual é trocado por outro valor que esteja próximo, não havendo mudanças drásticas no processo de mutação. É recomendável para esse operador que sua taxa de mutação seja algo em torno de $1/\text{size}(S)$ ¹, ou seja, apenas um índice da solução candidata será alterado no processo de mutação (DURILLO; NEBRO; ALBA, 2010).

Consoante (DEB; DEB, 2014), para uma solução pai $p \in [a, b]$, a solução mutada p' para uma variável particular é criada a partir de um número aleatório $u \in [0, 1]$, representado em 2.15.

$$p' = \begin{cases} p + \bar{\delta}_E(p - x_i^{(a)}), & \text{para } u \leq 0.5 \\ p + \bar{\delta}_D(x_i^{(b)} - p), & \text{para } u > 0.5 \end{cases} \quad (2.15)$$

Os parâmetros $\bar{\delta}_E$ e $\bar{\delta}_D$ são calculados em 2.16 e 2.17 na devida ordem. Eles correspondem aos valores da esquerda e direita da variável.

$$\bar{\delta}_E = (2u)^{1/(1+\eta)} - 1, \text{ para } u \leq 0.5 \quad (2.16)$$

$$\bar{\delta}_D = 1 - (2(1 - u))^{1/(1+\eta)}, \text{ para } u > 0.5 \quad (2.17)$$

É importante ressaltar que o tipo de cruzamento ou mutação que será escolhido irá depender do tipo de codificação do indivíduo, visto que alguns problemas específicos — como é o caso do TSP — só funcionam corretamente com operadores de permutação, sendo que os mesmos devem ser modificados para atender as restrições do problema.

2.8 Parâmetros do Algoritmo Evolutivo

Cada operador do algoritmo evolutivo funcionam através de parâmetros predefinidos, os quais precisam ser calibrados e definidos corretamente para o bom desempenho do AE no processo de otimização. Posto isto, alguns parâmetros popularmente utilizados são apresentados a seguir:

- Número de gerações: o número máximo de iterações que serão realizadas no processo de evolução, geralmente atribuído ao critério de parada do AE;

¹size(S) indica o tamanho do vetor de soluções

- Tamanho da população: determinando a quantidade de soluções candidatas em cada população;
- Tamanho do torneio: quantos indivíduos são selecionados para “competirem” entre si pelo método de seleção torneio;
- Taxa de recombinação: um valor que determina a probabilidade de um determinado par de indivíduos realizarem recombinação de valores;
- Taxa de mutação: um valor que determina a probabilidade de um determinado indivíduo sofrer mutação de seus valores;

A qualidade de otimização da aplicação está ligada diretamente aos parâmetros escolhidos para a execução do algoritmo evolutivo, sendo que a configuração dos valores desses parâmetros é crucial para um bom desempenho do AE (EIBEN; SMITH, 2015). Para isso, testes variando esses parâmetros devem ser realizados observando o comportamento do algoritmo evolutivo no decorrer das execuções (BÄCK; FOGEL; MICHALEWICZ, 2000). Devido ao fato dos AE serem não determinísticos, isto é, pode-se obter resultados diferentes a cada otimização, o melhor resultado nem sempre é garantido. Para isso, é recomendável efetuar uma série de testes com os mesmos parâmetros e observar se houve uma persistência de “bons” resultados nesse processo. Ademais, técnicas de otimização de parâmetros também podem ser empregadas (EIBEN; SMIT, 2011).

2.9 Algoritmos Evolutivos para Otimização Multiobjetivo

Os algoritmos evolutivos (AE) são meta-heurísticas populares que podem acomodar problemas multiobjetivos, utilizando funções de aptidão especializadas e introdução de métodos para promover uma diversidade de soluções (KONAK; COIT; SMITH, 2006).

Devido ao fato de que os Algoritmos Evolutivos Multiobjetivos (MOEAs) poderem codificar soluções individuais através de várias representações diretas (estrutura de dados cromossômicas), além de calcular de forma direta os valores de seus objetivos, eles possuem uma vantagem relevante com relação a outras técnicas tradicionais de busca e otimização (COELLO et al., 2007). Isso por que eles trabalham em um espaço de busca global, aumentando consideravelmente as hipóteses de encontrar ótimas soluções globais dentro desse espaço.

Existem na literatura diversos algoritmos que tratam problemas de otimização multiobjetivos utilizando algoritmos evolutivos, como *Pareto Archived Evolution Strategy (PAES)* (KNOWLES; CORNE, 1999), *Pareto Envelope-based Selection Algorithm (PESA2)* (CORNE; KNOWLES; OATES, 2000), *Non-dominate Sort Genetic Algorithm II (NSGA-II)* (DEB et al., 2002) e *Strength Pareto Evolutionary Algorithm 2 (SPEA2)* (ZITZLER; LAUMANN; BLEULER, 2004).

Dentre os MOEA's disponíveis na literatura, os algoritmos SPEA2 e NSGA-II serão detalhados a seguir.

2.9.1 O Algoritmo SPEA2

O SPEA2 é um algoritmo evolutivo utilizado para encontrar ou aproximar o conjunto de Pareto para problemas de Otimização multiobjetivos, utilizando *fitness* por contagem de dominância, estimação de densidade e elitismo para soluções não-dominadas. Seu objetivo é eliminar as fraquezas em potenciais do seu antecessor (SPEA) e adicionar os resultados das avaliações recentes.

Para relatar sobre as especificações do SPEA2, primeiramente é necessário abordar alguns conceitos básicos do seu antecessor, o SPEA.

Conforme Zitzler, Laumanns e Thiele (2001), o SPEA utiliza duas populações, uma população regular P e uma externa \bar{P} (armazenada em arquivo externo). Para dar início a execução do algoritmo, um arquivo vazio é criado e uma população P é iniciada. É passado por parâmetro o tamanho das duas populações, no qual se denota P_{size} e \bar{P}_{size} . Dentro de um processo de iteração, todos os membros não-dominados da população P são copiados para a população externa, preenchendo a população \bar{P} . Qualquer indivíduo dominado ou duplicado é retirado da população \bar{P} durante a operação de atualização. Caso a população externa exceda o limite predefinido, outros membros desse arquivo são excluídos através de uma técnica de agrupamento, no qual preserva as características da fronteira não-dominada (fronteira de Pareto). Logo após, os membros da população presentes na população externa recebem a atribuição do valor de *fitness*.

Ainda seguindo a abordagem de Zitzler, Laumanns e Thiele (2001), é atribuído um valor de força $S(i)$ para cada indivíduo i , que representa a quantidade de soluções dominadas por i . E o *fitness* de cada indivíduo i corresponde a soma das forças dos indivíduos que dominam i , adicionado de um fator de densidade. Assim, o SPEA irá realizar a ação de minimizar o valor da função de aptidão (*fitness*).

Segundo Zitzler, Laumanns e Thiele (2001), após o processo de avaliação, os indivíduos da população externa são selecionados por meio de um torneio binário para os processos de reprodução. O indivíduo da população com menor valor de *fitness* será o vencedor do torneio. Após o processo de recombinação e mutação, os membros da população regular antiga são substituídos pelos novos indivíduos gerados no processo.

O diferencial desse algoritmo é que ele guarda todas as suas soluções não-dominadas em uma população externa, e forma-se assim um conjunto de Pareto-ótimo que será utilizado para as tomadas de decisão.

As diferenças entre o SPEA2 e seu antecessor (SPEA) segundo Zitzler, Laumanns e Bleuler (2004) é que o SPEA2 possui:

- Um esquema melhorado de aptidão física, que leva para cada indivíduo em conta quantos indivíduos domina e é dominado por.
- Uma técnica de estimativa da densidade de vizinhos mais próxima, que permite uma orientação mais precisa do processo de busca.
- Um novo arquivo de truncamento de métodos que garante a preservação de fronteira soluções.

Os autores Zitzler, Laumanns e Bleuler (2004) afirmam que realizaram um estudo comparativo entre os dois algoritmos, e o SPEA2 oferece um bom desempenho em termos de convergência e diversidade superando assim seu antecessor.

2.9.2 O Algoritmo NSGA-II

O NSGA-II é um algoritmo evolutivo que utiliza a classificação de soluções não-dominadas, atribuindo um *ranking* as soluções da fronteira de Pareto (VACHHANI, 2016). Além disso, esse algoritmo evolutivo introduz o elitismo, sendo este um fator importante para melhorar as propriedades de convergência do algoritmo. Também é implementado um mecanismo de diversidade, aumentando a variedade de soluções da fronteira de Pareto. Esse algoritmo segue o fluxo de passos padrões (figura 2.7) de um AE tradicional, com uma diferença no operador de seleção. O processo de seleção do NSGA-II possui duas características (DEB et al., 2002):

1. É realizada uma classificação da população através do critério de dominância, isto é, dada uma população, os indivíduos são classificados como dominados ou não-dominados. Após isso, os indivíduos são categorizados em uma espécie de fronteira, onde na primeira fronteira estão todos os indivíduos não-dominados, conseguinte na segunda fronteira estão outras soluções não-dominadas com exceção dos indivíduos presentes na primeira fronteira, e seguindo até a última fronteira de soluções. Considera-se que os melhores indivíduos estão alocados na primeira fronteira. Desta forma, o conceito de elitismo é aplicado, pois, no processo de seleção, as melhores soluções são favorecidas após aplicação da dominância, o qual categoriza a população em diferentes níveis de qualidade.
2. Utiliza um mecanismo de diversidade conhecido como *crowding distance* (ou distância de aglomeração) na seleção por torneio. Essa técnica tem a função de manter uma diversidade na população, fazendo com que pontos extremos de uma solução sejam priorizadas no processo de seleção, mantendo, assim uma frente com soluções diversificadas (aproximação da fronteira de Pareto). Conforme Deb et al. (2002), para calcular o *crowding-distance* é necessário classificar a população de acordo com cada valor de função objetivo em ordem crescente. Em seguida, os extremos das soluções para cada função objetiva recebem um valor de distância infinito. Para as outras soluções intermediárias são atribuídos um valor de distância igual à diferença absoluta normalizada nos valores da função de duas soluções adjacentes. Esse cálculo também é atribuído

às outras funções objetivo. O valor global do *crowding-distance* é calculado como a soma dos valores de distância individuais correspondentes a cada objetivo. Antes de calcular o valor de *crowding-distance* é necessário normalizar cada função objetiva. A figura 2.17 apresenta o cálculo do *crowding-distance* para duas funções objetivo, sendo que este pode ser aplicado com mais funções objetivo.

A figura 2.16 mostra uma representação simplificada do processo de seleção do NSGA-II.

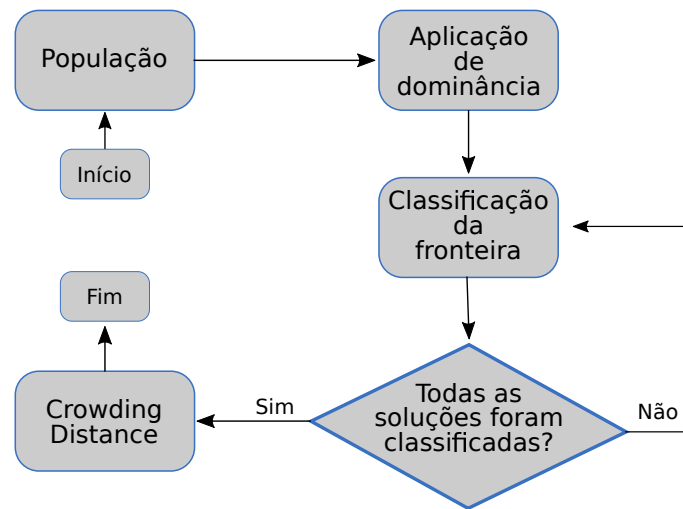


Figura 2.16: Seleção pelo algoritmo NSGA-II.

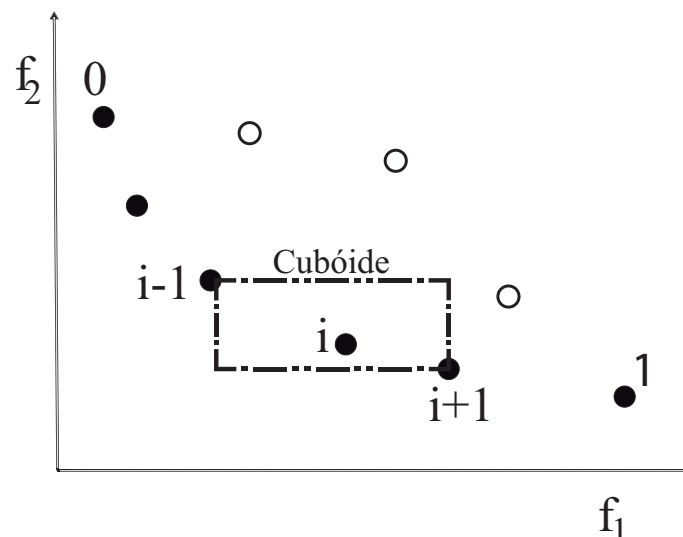


Figura 2.17: Cálculo do *crowding-distance*. Os pontos preenchidos são soluções da mesma fronteira não-dominada. Adaptado de Deb et al. (2002).

Consoante Vachhani (2016), o NSGA-II funciona da seguinte forma: É criada uma população de pais P_0 de tamanho K aleatoriamente, em seguida são aplicados os operadores de mutação e recombinação em P_0 para criar uma população de filhos Q_0 de tamanho K , tendo agora duas populações (P_0 e Q_0). Dessa forma, o conjunto de soluções candidatas é denotado $S_0 = P_0 + Q_0$ ou $S_i = P_i + Q_i$, onde i indica a iteração corrente. Em seguida calcula-se a classificação das soluções não-dominadas de todas as soluções de S_0 . Selecionam-se as melhores K soluções baseadas em classificação de dominância e *crowding distance*. Essas K soluções são repassadas para a população seguinte (S_{i+1}). Esse processo se repete a partir da classificação das soluções não dominadas, até o número máximo de gerações fornecidas em parâmetro e ao final delas, é retornado um conjunto de soluções Pareto-ótimas. Esse processo pode ser visto no pseudo-código 1 (adaptado de Deb et al. (2002) e Durillo, Nebro e Alba (2010)).

Algorithm 1 Algoritmo NSGA-II

```

1:  $P \leftarrow Populacao\_Aleatoria()$  ▷  $P = População$ 
2:  $Q \leftarrow \emptyset$  ▷  $Q = População\ auxiliar$ 
3: while  $Condicao\_Parada() \neq True$  do
4:   for  $i \leftarrow 1$  to  $n$  do
5:      $valorAleatorio \leftarrow gerarNumeroAleatorio()$  ▷ Gera um número aleatório
6:     if  $(valorAleatorio \leq 1/2)$  then
7:        $pai \leftarrow Selecao(P_i)$  ▷ Somente um pai é selecionado
8:        $filhos \leftarrow PolynomialMutation(pai)$ 
9:     else
10:       $pais \leftarrow Selecao(P_i)$  ▷ Dois pais são selecionados
11:       $filhos \leftarrow SBXCrossover(pais)$ 
12:     end if
13:      $Funcao\_Avaliacao(filhos)$ 
14:      $Insere(Q_i, filhos)$ 
15:   end for
16:    $S_i \leftarrow Q_i \cup P_i$ 
17:    $Ranking\_e\_CrowdingDistance(S_i)$ 
18:    $P_{i+1} \leftarrow Selecciona\_Melhores\_Individuos(S_i)$ 
19: end while
20: Retorna  $P$ 

```

Uma melhoria do NSGA-II é o NSGA-III, proposta por Deb e Jain (2014). Este foi elaborado para otimização com múltiplos objetivos usando a abordagem de classificação não dominada baseada em pontos de referência. Sua abordagem é parecida com o seu antecessor, possuindo algumas melhorias na classificação de indivíduos em níveis não dominados, determinação de um ponto de diferença em um hiperplano, normalização adaptativa nos membros da população, operador de associação e na operação de preservação de nicho.

Capítulo 3

Metodologia

Esse trabalho tratou do problema de maximização da cobertura redundante em redes de sensores visuais sem fio, aplicando algoritmos evolutivos neste problema de otimização multiobjetivo. Este problema foi tratado em trabalhos anteriores de Silva e Costa (2016) e Costa et al. (2014b), que utilizaram estratégias gulosas para otimizar a cobertura redundante em RSVSF. Portanto, os algoritmos aqui propostos foram comparados com ambas as abordagens.

A seguir, serão detalhadas as partes envolvendo o desenvolvimento deste trabalho, incluindo a modelagem e formalização do problema, técnicas e procedimentos, ferramentas utilizadas para o desenvolvimento do algoritmo, comparação com outras abordagens e a apresentação formal dos algoritmos evolutivos propostos, contendo suas características e operadores evolutivos utilizados.

3.1 Modelagem e Formalização do Problema

O principal objetivo deste trabalho é desenvolver heurísticas de algoritmos evolutivos multiobjetivos, que realize o processo de otimização da cobertura redundante nas RSVSF. Para isso, serão determinadas as orientações dos sensores visuais para maximizar a cobertura redundante de alvos distribuídos no cenário, tendo a cobertura e a redundância como objetivos distintos, tornando o problema multiobjetivo.

O cenário proposto é bidimensional com sensores e alvos dispersos no espaço do cenário (figura 3.1). Os alvos tratados nesse trabalho serão fixos, não podendo alterar seu posicionamento inicial no decorrer da vida útil da RSVSF. Eles são definidos como pontos com posição (x, y) , podendo, portanto, estarem apenas “dentro” ou “fora” do FoV de qualquer sensor visual considerado.

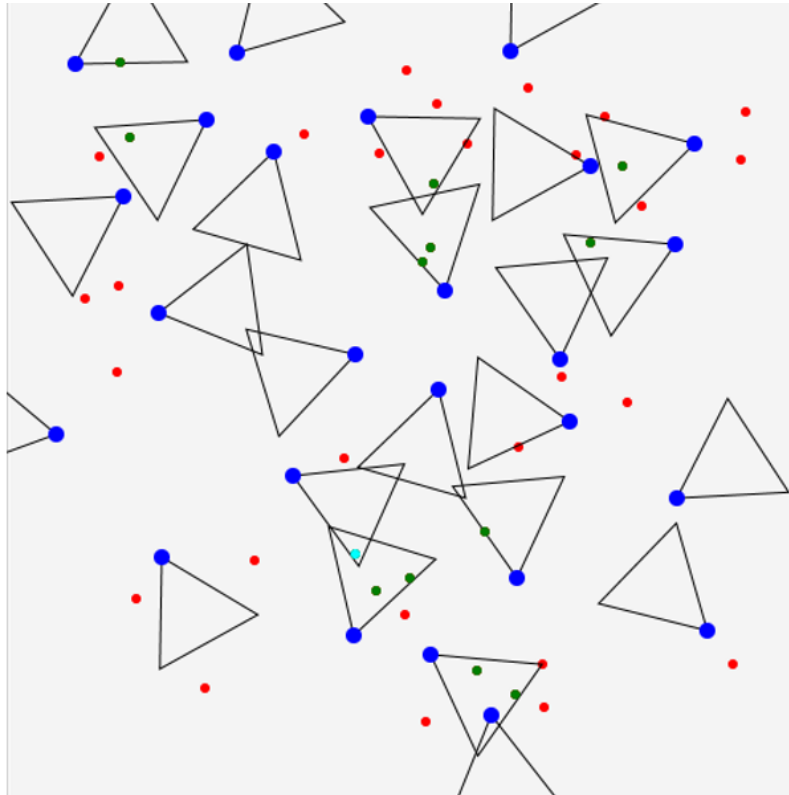


Figura 3.1: Monitoramento visual para 40 alvos e 25 sensores visuais, ambos implantados aleatoriamente. Considerando os 3 objetivos: *Cobertura* = 32.50%, *Redundância Mínima* = 2.50% e *Redundância Média* = 0.05.

Assim como os alvos, os sensores visuais também terão seu posicionamento fixo, seguindo as mesmas restrições. Para o algoritmo proposto, será considerada apenas a mudança de orientação (função *pan*) dos sensores visuais, devido sobretudo a questão do custo de aquisição das câmeras (funções de *zooming* requerem lentes mais robustas), modelagem 2D da rede e complexidades envolvidas com otimizações envolvendo outras variáveis que possam alterar o *FoV* dos sensores visuais.

Na modelagem do problema de maximização da cobertura redundante na RSVSF, é necessário formalizar os conceitos de cobertura e redundância em formulações matemáticas.

Para determinar se um alvo está dentro do *FoV* de um sensor visual, é utilizada a modelagem do campo de visão apresentada na seção 2.2.1, utilizando trigonometria para realizar a checagem (funções 2.3 e 2.4). Com relação as RSVSF, o conceito de alvo coberto é definido, conforme Costa e Guedes (2010), quando um alvo está no campo de visão (*FoV*) de pelo menos um nó-sensor visual. Sendo assim, denota-se cobertura (C) da RSVSF como a proporção de alvos cobertos conforme a equação 3.1.

$$C(\mathbf{S}, \mathbf{A}) = \frac{\sum_{j=1}^n \text{coberto}(\mathbf{S}, a_j)}{n} \quad (3.1)$$

onde n é o total de alvos, \mathbf{S} é o conjunto de sensores contendo seu posicionamento (x, y) e orientação θ , \mathbf{A} é o conjunto de alvos contendo o posicionamento (x, y) de cada alvo, $\text{coberto}(\mathbf{S}, a_j)$ é 1 se o alvo a_j está dentro do FoV de algum sensor \mathbf{S}_i e 0 caso contrário.

Da mesma forma, define-se que a redundância mínima (R) da RSVSF avalia quando os alvos estiverem dentro do FoV de pelo menos dois nós-sensores visuais. Desta forma, denota-se redundância mínima através da equação 3.2.

$$R(\mathbf{S}, \mathbf{A}) = \frac{\sum_{j=1}^n \text{redundante}(\mathbf{S}, a_j)}{n} \quad (3.2)$$

onde $\text{redundante}(\mathbf{S}, a_j)$ é 1 se o alvo a_j estiver dentro do FoV de pelo menos 2 sensores visuais, e 0 caso contrário.

Além disso, pode-se definir que a redundância média (RM) é a avaliação da quantidade média de sensores redundantes cobrindo cada alvo da RSVSF. Ela é denotada conforme a equação 3.3.

$$RM(\mathbf{S}, \mathbf{A}) = \frac{\sum_{j=1}^n (\text{hits_redundantes}(\mathbf{S}, a_j))}{n} \quad (3.3)$$

onde $\text{hits_redundantes}(\mathbf{S}, a_j)$ retorna a quantidade de alvos a_j que estejam dentro do FoV de pelo menos dois sensores.

Com os três objetivos a serem otimizados apresentados (C, R, RM), define-se a modelagem matemática da otimização multiobjetivo para o problema de cobertura redundante na equação 3.4.

$$\begin{aligned} & \underset{\theta_{s_i}}{\text{maximize}} && f(\mathbf{S}) = (C(\mathbf{S}, \mathbf{A}), R(\mathbf{S}, \mathbf{A}), RM(\mathbf{S}, \mathbf{A})) \\ & \text{sujeito a} && \theta_{s_i} \in \mathbb{Z}, \\ & && 0 \leq \theta_{s_i} \leq 360. \end{aligned} \quad (3.4)$$

onde \mathbf{A} é o conjunto de alvos a serem testados, θ_{s_i} corresponde ao valor de orientação do i -ésimo sensor visual s_i , que é um valor dentro do conjunto de números inteiros, $\theta \in \mathbb{Z}$ e deve estar entre 0 e 360.

3.2 Algoritmo Evolutivo para Abordagem Lexicográfica

O problema de maximização da cobertura redundante foi definido em três objetivos distintos: cobertura (C), redundância mínima (R) e redundância média (RM). Para utilizar a abordagem lexicográfica, foi necessário determinar a ordem de prioridade de cada objetivo.

Dada esta definição, priorizou-se, primeiro, a cobertura, pois, quanto maior a quantidade de alvos cobertos, mais informações podem ser obtidas para o controle de monitoramento da rede. A redundância mínima tem a segunda importância para garantir que o conjunto de alvos cobertos, se possível, tenham pelo menos dois sensores realizando seu monitoramento. Desta forma, caso haja falhas em alguns sensores, os nós redundantes garantirão que a rede não sofra pedras de informações valiosas. Por fim, quando os objetivos anteriores forem atingidos, a redundância média tem a finalidade de garantir que os alvos possuam o maior número de sensores apontados para eles.

Assim, o problema multiobjetivo neste caso corresponderá ao problema multiobjetivo de otimização lexicográfica como segue:

$$\begin{aligned} & \underset{\theta_{s_i}}{\text{maximize}} && f(\mathbf{S}) = C(\mathbf{S}, \mathbf{A}) \\ & \text{sujeito a} && \theta_{s_i} \in \mathbb{Z}, \\ & && 0 \leq \theta_{s_i} \leq 360. \end{aligned} \tag{3.5}$$

$$\begin{aligned} & \underset{\theta_{s_i}}{\text{maximize}} && f(\mathbf{S}) = R(\mathbf{S}, \mathbf{A}) \\ & \text{sujeito a} && \theta_{s_i} \in \mathbb{Z}, \\ & && 0 \leq \theta_{s_i} \leq 360, \\ & && C(\mathbf{S}, \mathbf{A}) \leq C(\mathbf{S}_c^*, \mathbf{A}_c^*). \end{aligned} \tag{3.6}$$

$$\begin{aligned} & \underset{\theta_{s_i}}{\text{maximize}} && f(\mathbf{S}) = RM(\mathbf{S}, \mathbf{A}) \\ & \text{sujeito a} && \theta_{s_i} \in \mathbb{Z}, \\ & && 0 \leq \theta_{s_i} \leq 360, \\ & && C(\mathbf{S}, \mathbf{A}) \leq C(\mathbf{S}_c^*, \mathbf{A}_c^*), \\ & && R(\mathbf{S}, \mathbf{A}) \leq R(\mathbf{S}_r^*, \mathbf{A}_r^*). \end{aligned} \tag{3.7}$$

onde $C(\mathbf{S}_c^*, \mathbf{A}_c^*)$ representa o valor ótimo da função objetivo $C(\mathbf{S}, \mathbf{A})$, encontrado na otimização do problema 3.5, e $R(\mathbf{S}_r^*, \mathbf{A}_r^*)$ representa o valor ótimo da função objetivo $R(\mathbf{S}, \mathbf{A})$, encontrado na otimização do problema 3.6.

Foi desenvolvido um algoritmo evolutivo (AE) lexicográfico para otimizar cobertura (C), redundância mínima (R) e redundância média (RM) de uma rede de sensores visuais sem fio, buscando as melhores orientações para o conjunto de sensores visuais implantados. Esse algoritmo funciona de forma centralizada, isto é, há um controlador que conhece o posicionamento de todos os sensores e alvos da rede, e após o processo de otimização, envia a orientação para cada sensor.

O AE proposto trabalha com um conjunto de possíveis soluções denominada População (P). Essa população é composta por indivíduos que representam soluções candidatas para o problema, assim cada indivíduo é um vetor numérico de orientações para cada sensor, ou seja, cada vetor tem tamanho fixo correspondente a quantidade de sensores presentes na RSVSF. Cada posição do vetor possui o valor da orientação que pode ser um valor inteiro entre 0° e 360° .

O algoritmo evolutivo lexicográfico proposto para otimização lexicográfica para o problema de maximização da cobertura redundante foi desenvolvido considerando as seguintes especificações:

- **População Inicial:** A população inicial de soluções é gerada pela criação de vetores com valores aleatórios inteiros limitados entre 0 e 360, correspondente as orientações dos sensores. A população inicial e das demais gerações tem tamanho fixo.
- **Função de avaliação (*fitness*):** Para cada solução, isto é, uma proposta de orientação de cada sensor, são avaliados os valores de C , R e RM , ou seja, cada solução tem associada estes três valores pela função de avaliação.
- **Método de seleção por **elitismo** e **torneio**:** Após a avaliação dos indivíduos pela função *fitness*, o processo de seleção é executado em duas etapas. Inicialmente, selecionam-se os 2 (ou 3 em caso de tamanho ímpar da população) melhores indivíduos da população corrente para colocá-los diretamente na nova população, os quais passarão pelos operadores de recombinação e mutação. Essa seleção baseia-se no elitismo, e considera os maiores valores de C , R e RM de acordo com as seguintes propriedades: C é mais relevante que R , e R é mais relevante que RM . Após isso, ocorre a seleção por torneio com K indivíduos sendo selecionados aleatoriamente, e o melhor indivíduo deles sendo selecionados para uma subpopulação temporária. O objetivo deste passo é repetir este processo de torneio até que esta subpopulação complete o tamanho da população, menos o número de indivíduos inicialmente selecionados na seleção por elitismo. Esta subpopulação, que pode até conter indivíduos repetidos, será considerada para a próxima etapa do algoritmo.
- **Método de recombinação (*crossover*) **uniforme**.** Esse operador recebe a subpopulação gerada após o processo de seleção para realizar o cruzamento entre 2 indivíduos.
- **Método de mutação por **soma de valores** (**SMV**):** Após o processo de recombinação, acontece uma mutação nos genes para promover variação genética

na população. Essa mutação sorteia aleatoriamente $M\%$ das posições do vetor e soma um pequeno valor aleatório de x° ao valor atual de cada posição. Este valor (x°) é sorteado a partir de uma distribuição normal de média zero e desvio padrão configurável em parâmetro, podendo este valor sorteado ser positivo ou negativo.

Após a execução da mutação, a subpopulação é incorporada na nova população, juntando-se aos indivíduos selecionados por elitismo. Em seguida, a nova população é avaliada e entra no ciclo evolutivo até atingir o critério de parada. Na verdade, existem diversos critérios de parada a serem considerados, mas, para o AE lexicográfico, é considerado o número máximo de gerações.

Para o funcionamento adequado do algoritmo evolutivo lexicográfico, alguns parâmetros devem ser determinados, como especificado a seguir:

- Número de gerações.
- Tamanho da população.
- Tamanho do torneio.
- Taxa de recombinação.
- Taxa de mutação.
- Taxa de posições para mutação: o percentual de posições do vetor que irá sofrer alteração por adição de um pequeno valor.
- Desvio padrão: aplicado no operador de mutação, este parâmetro é utilizado para calibrar a dispersão da distribuição normal com média zero.

A maioria destes parâmetros serão otimizados para definir qual a configuração mais adequada para solucionar o problema de maximização da cobertura redundante. O método de otimização será apresentado na seção 3.4.

3.3 Algoritmo Evolutivo multiobjetivo NSGA-II

Para a implementação do algoritmo evolutivo de otimização *a posteriori* NSGA-II, foi utilizado o *framework* JMetal desenvolvido por Durillo, Nebro e Alba (2010). O JMetal é um *framework* multiplataforma desenvolvido em JAVA, podendo ser executado em qualquer sistema operacional. A sua escolha deve-se ao fato de possuir diversos algoritmos multiobjetivos já implementados, além da sua fácil modulação e adequação a qualquer problema novo.

Cada indivíduo da população é representado da mesma forma que no algoritmo evolutivo lexicográfico, ou seja, são vetores numéricos de orientações para cada sensor. Cada indivíduo é avaliado para os três objetivos propostos, calculando os valores de C , R e RM , conforme apresentado nas funções 3.1, 3.2 e 3.3 respectivamente.

Além disso, esse algoritmo utiliza as seguintes especificações para otimizar o problema da cobertura redundante:

- Método de seleção Torneio.
- Método de recombinação SBX.
- Método de mutação PM.

Para o funcionamento adequado do algoritmo NSGA-II, alguns parâmetros devem ser determinados, como especificado a seguir:

- Número de gerações.
- Tamanho da população.
- Tamanho do torneio.
- Taxa de recombinação.
- Taxa de mutação.
- Índice de distribuição η para SBX.
- Índice de distribuição η para PM.

Assim como no AE Lexicográfico, a maioria destes parâmetros também serão otimizados.

3.4 Otimização de parâmetros (*Parameter Tuning*)

A escolha dos parâmetros é um dos principais fatores para obtenção de resultados promissores através dos algoritmos evolutivos de otimização (EIBEN; SMIT, 2011). Para este trabalho, utilizou-se a abordagem de otimização conhecida como *grid search* ou busca em grade (COROIU, 2016) .

Esta abordagem utiliza força bruta (*brutal force*) para avaliar todas as possibilidades a partir de um conjunto de valores pré-selecionados para cada parâmetro. Desta forma, primeiro são definidos alguns valores para os parâmetros a serem otimizados e o *grid search* faz uma combinação entre todos valores de parâmetros, testando-os no algoritmo de otimização. A partir disso, pode-se ordenar os resultados para avaliar e determinar qual ou quais são as combinações de parâmetros mais efetivas.

O processo de otimização de parâmetros utilizando o *grid search* foi aplicado nos dois algoritmos propostos, mas a metodologia de escolha da configuração adequada para cada algoritmo é distinta. O algoritmo lexicográfico, por ser uma abordagem *a-priori*, apresenta apenas uma solução ótima como resultado. Por outro lado, o

NSGA-II, por ser uma abordagem *a-posteriori*, dispõe como resultado um conjunto de soluções Pareto-ótimas.

Para determinar a configuração que será utilizada pelo algoritmo lexicográfico, os parâmetros são ordenados pelos melhores resultados de C , R e RM , além de ordenar os parâmetros por ordem decrescente, considerando seus menores valores e visando a redução do custo computacional do algoritmo.

Já para determinar a configuração dos parâmetros utilizados pelo NSGA-II, utilizou-se uma abordagem semelhante. Foram considerados os valores máximos de cada objetivo do conjunto de Pareto-ótimo obtido após o processo de otimização. Desta forma, foi realizada uma ordenação dos resultados, considerando primeiramente os valores de C para a máxima solução de cobertura, R para a máxima solução de redundância mínima e RM para a máxima solução de redundância média. Em seguida, a ordenação continuou por C da máxima R e RM , R da máxima C e RM , e RM da máxima C e R . Após isso, os parâmetros utilizados foram ordenados de forma decrescente, considerando os menores valores de modo a tentar reduzir o seu custo computacional.

3.5 Método de Avaliação dos Resultados

Os resultados foram gerados através de um ambiente virtual de testes, gerando diversos cenários de testes para os algoritmos evolutivos propostos e para comparação com os algoritmos da literatura. Por simplificação, neste estudo os cenários propostos para as RSVSF são definidos em um plano bidimensional (2D). Os cenários possuem um tamanho fixo já predefinido, a qual atua como a base para a colocação dos sensores e alvos.

Os cenários são gerados com posicionamento aleatório dos sensores e alvos dentro dos limites do cenário. São consideradas quantidades fixas de alvos, variando a quantidade de sensores em um intervalo especificado. De modo a obter melhores estimativas dos resultados esperados, para uma dada quantidade de alvos e sensores, varia-se o posicionamento destes para gerar diferentes cenários. Estas variações de posicionamento tem por objetivo avaliar o comportamento dos algoritmos em condições iniciais variadas, determinando assim uma estimativa por um valor médio dos resultados.

Para cada cenário gerado, todos os algoritmos realizam o processo de otimização de acordo com suas heurísticas e, após a finalização da otimização, obtêm-se os resultados finais. São avaliados os valores da cobertura, da redundância mínima e da redundância média no processo de otimização de cada algoritmo. Para o algoritmo evolutivo lexicográfico, o melhor indivíduo da população final é considerado como a melhor solução, sendo este o valor a ser comparado com o melhor valor obtido pelos algoritmos CPGA, ECPGA e RCMA. Em virtude do NSGA-II ser uma abordagem

a posteriori, foram consideradas as soluções de valores máximos de cada objetivo, ou seja, foram escolhidas 3 soluções para o NSGA-II. Também foi avaliado o custo computacional (número de testes de *FoV*) de cada algoritmo.

Capítulo 4

Resultados

Após a definição das abordagens evolutivas para resolver o problema de maximização da cobertura redundante em RSVSF, foram simulados testes extensivos para comparar as diferentes abordagens disponíveis para tratar deste problema. Considerando a implementação dos algoritmos evolutivos propostos, seguindo os métodos apresentados na seção anterior, considerando e avaliando diferentes configurações, os algoritmos foram executados no ambiente virtual para consolidar os resultados. O objetivo do ambiente virtual era automatizar os testes executados, organizando e consolidando os resultados para serem exibidos em gráficos descritivos e comparativos, a partir dos resultados de uma série de 20 execuções sucessivas dos algoritmos sobre diferentes posicionamentos dos sensores e alvos na rede.

Para uma área monitorada em um plano 2D de 500 m x 500 m e assumindo a implantação aleatória de sensores visuais e alvos, considerou-se que todos os sensores visuais têm ângulo de abertura (α) = 60° e raio de alcance (r) = 70 m. Vale destacar que os algoritmos evolutivos propostos podem realizar a otimização considerando especificações diferentes (para α e r) dos sensores visuais. Mas por questões de limitações dos algoritmos a serem comparados, foi determinado que todos os sensores possuem as mesmas especificações.

Qualquer cenário pode ser considerado para realizar a otimização dos objetivos propostos, desde que estes possuam todas as variáveis necessárias para o funcionamento adequado dos algoritmos de otimização. Na figura 4.1 é apresentado um cenário de monitoramento antes (*a*) e depois (*b*) da otimização realizada pelo algoritmo evolutivo lexicográfico.

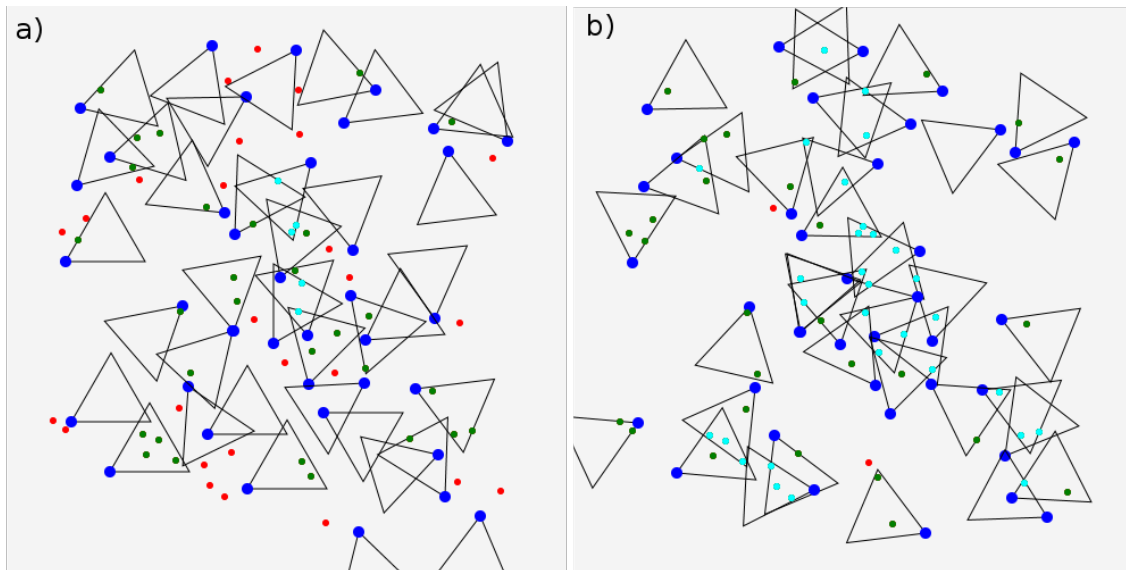


Figura 4.1: Monitoramento visual para 60 alvos e 38 câmeras, ambos implantados aleatoriamente. a) antes da otimização: $C = 56.66\%$, $R = 8.33\%$ e $RM = 0.1833$. b) após a otimização: $C = 96.66\%$, $R = 50\%$ e $RM = 1.05$

Como pode ser notado, os objetivos otimizados foram aprimorados após a execução do AE lexicográfico. Nesta figura, os pontos vermelhos são tidos como descobertos, os pontos verdes são tidos como cobertos e os pontos azuis tidos como redundantes. Observa-se que o número de pontos cobertos e redundantes aumentou consideravelmente após a otimização. Entretanto, essa análise é superficial e testes extensivos tornam-se necessários.

Foram definidas duas variações de quantidade de alvos no cenário, para o primeiro cenário a quantidade de alvos escolhida foi 50, enquanto o segundo cenário possui 100 alvos. A quantidade de sensores foi variada também, iniciando em 10 e aumentando gradativamente somando mais 10 sensores até chegar em 80. Com as configurações definidas, o cenário é montado com as posições aleatórias dos sensores e os alvos.

Os resultados serão apresentados em duas seções diferentes. Na seção 4.2, foram comparados o desempenho do algoritmo evolutivo lexicográfico proposto com os algoritmos gulosos CPGA, ECPGA e RCMA. Ainda nesta seção, foram apresentados os resultados referentes as duas configurações de cenário estabelecidas acima, com 50 e 100 alvos. Já na seção 4.3 são comparados os resultados apresentados na seção 4.2, equiparando o NSGA-II com as abordagens utilizadas.

Antes de iniciar as comparações entre os algoritmos evolutivos e gulosos, foram definidos quais parâmetros utilizados pelo AE lexicográfico e pelo NSGA-II. Os resultados obtidos após a otimização de parâmetros serão apresentados a seguir.

4.1 Definição dos Parâmetros

Os parâmetros foram definidos através da otimização de parâmetros apresentada na seção 3.4. As tabelas 4.1 e 4.2 apresentam os valores de cada possível configuração de cada parâmetro utilizadas na técnica *grid search* para os algoritmos evolutivos lexicográfico e NSGA-II.

Tabela 4.1: Valores do *grid search* para o AE lexicográfico

Parâmetros	Valores
População inicial P	50, 100, 200
Número de gerações	200, 400, 600
Taxa de <i>crossover</i> (%)	60, 80, 100
Taxa de mutação (%)	40, 60, 80
Taxa posições mutáveis (%)	10, 20, 30

Tabela 4.2: Valores do *grid search* para o NSGA-II

Parâmetros	Valores
População inicial P	50, 100, 200
Taxa de <i>crossover</i> (%)	60, 80, 100
Taxa mutação PM	$1/size(\mathbf{S})$, $3/size(\mathbf{S})$, $5/size(\mathbf{S})$
Índice de distribuição (η) SBX	15, 20, 25
Índice de distribuição (η) PM	15, 20, 25

O $size(\mathbf{S})$ indica o número de sensores, que também corresponde ao tamanho do cromossomo.

Foram realizadas três séries de testes para obter a configuração adequada pela técnica *grid search*. Para todas as séries, houve um total de 243 combinações para o AE lexicográfico e o mesmo número para o NSGA-II. Para o AE lexicográfico, foram considerados os parâmetros: população inicial, número de gerações, taxas de *crossover*, mutação e posições mutáveis, tendo um total de cinco parâmetros. Enquanto para o NSGA-II foram considerados os parâmetros: população inicial, taxas de *crossover* e mutação PM, e os índices de distribuição (η) SBX e PM. As séries de testes foram aplicadas em cenários com 20, 40 e 60 sensores respectivamente, e 50 alvos. Após as definições de quais os valores dos parâmetros seriam utilizados pelo NSGA-II, foi executado um *grid search* para definir o número de gerações.

As tabelas 4.3, 4.4 e 4.5 exibem os 5 primeiros resultados obtidos em cada cenário após a execução do *grid search* no AE lexicográfico. Os resultados foram ranqueados conforme apresentado na seção 3.4.

Tabela 4.3: Resultados após execução do *grid search* no AE Lexicográfico com 20 sensores.

Execução	C	R	RM	NG	TP	TX	TM	TMA
173	56.0	8.0	0.16	600	50	0.6	0.8	0.2
92	56.0	8.0	0.16	600	100	0.6	0.8	0.2
11	56.0	8.0	0.16	600	200	0.6	0.8	0.2
163	56.0	4.0	0.10	400	50	0.6	0.8	0.3
82	56.0	4.0	0.10	400	100	0.6	0.8	0.3

Legenda: *C*, *R* e *RM* correspondem aos valores dos objetivos. NG = número de gerações, TP = tamanho da população, TX = taxa de *crossover*, TM = taxa de mutação, TMA = taxa de posições mutáveis.

Tabela 4.4: Resultados após execução do *grid search* no AE Lexicográfico com 40 sensores.

Execução	C	R	RM	NG	TP	TX	TM	TMA
167	88.0	14.0	0.3	600	50	0.8	0.8	0.3
86	88.0	14.0	0.3	600	100	0.8	0.8	0.3
5	88.0	14.0	0.3	600	200	0.8	0.8	0.3
166	88.0	10.0	0.2	400	50	0.8	0.8	0.3
85	88.0	10.0	0.2	400	100	0.8	0.8	0.3

Legenda: vide tabela 4.3

Tabela 4.5: Resultados após execução do *grid search* no AE Lexicográfico com 60 sensores.

Execução	C	R	RM	NG	TP	TX	TM	TMA
214	86.0	54.0	1.42	400	50	1.0	0.6	0.1
133	86.0	54.0	1.42	400	100	1.0	0.6	0.1
52	86.0	54.0	1.42	400	200	1.0	0.6	0.1
215	86.0	54.0	1.42	600	50	1.0	0.6	0.1
188	86.0	54.0	1.42	600	50	1.0	0.8	0.1

Legenda: vide tabela 4.3

Na tabela 4.3, constata-se que ao diminuir o número de gerações, houve perda no valor de *R* e *RM*. Nos três primeiros resultados pode-se concluir que a variação no tamanho da população não causa sensibilidade na piora dos resultados, da mesma forma que as taxas de *crossover*, mutação e posições mutáveis.

Da mesma forma, a tabela 4.4 apresenta um comportamento parecido. Para o AE lexicográfico, o número de gerações indica que, quanto mais ele explora o espaço de busca, maiores resultados podem ser alcançados.

Já na tabela 4.5, observa-se que a variação dos parâmetros apresentados não proporcionam perdas nos resultados. Nesse caso, pode ter havido convergência prematura, no qual não foi necessário haver um grande valor de NG para chegar nos resultados apresentados. Como já foi dito, os resultados são ordenados pelos valores máximos de cada objetivo, seguindo pelos valores mínimos dos parâmetros. Diante disso, pode-se utilizar qualquer combinação de parâmetros dos quadros apresentados que, possivelmente, acarretará resultados promissores no problema de otimização tratado nesse trabalho.

Após as análises apresentadas, foi decidido então utilizar as configurações de parâmetro para o AE lexicográfico conforme a tabela 4.6. Foi realizada uma mesclagem dentre os parâmetros utilizados. Então, para o número de gerações foi escolhido o valor 400 para reduzir o custo computacional do algoritmo, sendo este parâmetro, juntamente ao tamanho da população, os principais elementos relativos ao custo considerado neste trabalho. Em relação ao tamanho da população, ela não causa sensibilidade nos resultados finais, então, foi escolhido o número intermediário dos valores disponíveis. Os demais parâmetros foram escolhidos conforme a frequência em que elas foram utilizadas.

Tabela 4.6: Parâmetros do algoritmo evolutivo lexicográfico

Parâmetros	Configuração
População inicial P	100
Número de gerações	400
Taxa de <i>crossover</i>	100%
Taxa de mutação	80%
Desvio padrão	25°
Tamanho torneio	5
Taxa posições mutáveis	30%

As tabelas 4.7, 4.8 e 4.9 apresentam os cinco primeiros resultados obtidos após a execução do *grid search* no algoritmo NSGA-II.

Tabela 4.7: Resultados após execução do *grid search* no NSGA-II com 20 sensores.

Ex.	MC-C	MC-R	MC-RM	MR-C	MR-R	MR-RM	MRM-C	MRM-R	MRM-RM	TP	TPM	η PM	TSBX	η SBX
162	48.0	6.0	0.12	40.0	14.0	0.28	40.0	14.0	0.28	50	1	15.0	0.6	15.0
171	48.0	6.0	0.12	40.0	14.0	0.28	40.0	14.0	0.28	50	1	15.0	0.6	20.0
189	48.0	6.0	0.12	40.0	14.0	0.28	40.0	14.0	0.28	50	1	15.0	0.8	15.0
198	48.0	6.0	0.12	40.0	14.0	0.28	40.0	14.0	0.28	50	1	15.0	0.8	20.0
207	48.0	6.0	0.12	40.0	14.0	0.28	40.0	14.0	0.28	50	1	15.0	0.8	25.0

Legenda: MC = solução de máxima cobertura, MR = solução de máxima redundância mínima, MRM = solução de máxima redundância média. C, R e RM correspondem aos valores dos objetivos obtidos nestas soluções. TP = tamanho da população, TPM = taxa do operador PM, η PM = índice de distribuição PM, TSBX = taxa do operador SBX, η SBX = índice de distribuição SBX.

Tabela 4.8: Resultados após execução do *grid search* no NSGA-II com 40 sensores.

Ex.	MC-C	MC-R	MC-RM	MR-C	MR-R	MR-RM	MRM-C	MRM-R	MRM-RM	TP	TPM	η PM	TSBX	η SBX
0	78.0	20.0	0.44	56.0	40.0	0.86	52.0	40.0	0.88	200	1	15.0	0.6	15.0
45	78.0	20.0	0.44	56.0	40.0	0.86	52.0	40.0	0.88	200	1	15.0	0.8	25.0
72	78.0	20.0	0.44	56.0	40.0	0.86	52.0	40.0	0.88	200	1	15.0	1.0	25.0
1	78.0	20.0	0.44	56.0	40.0	0.86	52.0	40.0	0.88	200	1	20.0	0.6	15.0
46	78.0	20.0	0.44	56.0	40.0	0.86	52.0	40.0	0.88	200	1	20.0	0.8	25.0

Legenda: vide tabela 4.7.

Tabela 4.9: Resultados após execução do *grid search* no NSGA-II com 60 sensores.

Ex.	MC-C	MC-R	MC-RM	MR-C	MR-R	MR-RM	MRM-C	MRM-R	MRM-RM	TP	TPM	η PM	TSBX	η SBX
54	96.0	66.0	1.6	90.0	72.0	1.76	86.0	72.0	1.82	200	1	15.0	1.0	15.0
55	96.0	66.0	1.6	90.0	72.0	1.76	86.0	72.0	1.82	200	1	20.0	1.0	15.0
56	96.0	66.0	1.6	90.0	72.0	1.76	86.0	72.0	1.82	200	1	25.0	1.0	15.0
36	96.0	66.0	1.6	92.0	70.0	1.66	88.0	70.0	1.78	200	1	15.0	0.8	20.0
37	96.0	66.0	1.6	92.0	70.0	1.66	88.0	70.0	1.78	200	1	20.0	0.8	20.0

Legenda: vide tabela 4.7.

Conforme resultados apresentados na tabela 4.7, percebe-se que não houve diferenças nos valores dos objetivos. Sendo assim, para essa situação, a variação dos parâmetros mostrados proporcionam os mesmos resultados.

De maneira similar, na tabela 4.8 também não houveram diferenças nos valores dos objetivos. Em comparação com a tabela 4.7, houve aumento no tamanho da população. Reforçando, o tamanho da população indica quantas soluções candidatas serão avaliadas pelo algoritmo evolutivo a cada geração, e quanto maior for o seu tamanho, maior será a variação de soluções candidatas. A variação dos parâmetros não proporcionou modificações nos resultados.

Por fim, na tabela 4.9 ocorreu o mesmo que os resultados anteriores, sem variação nos valores dos objetivos. Houve também uma semelhança entre os valores obtidos na tabela 4.4, sendo o mesmo tamanho da população e taxa da mutação polinomial em todas as situações apresentadas.

Desta forma, a tabela 4.10 exibe os parâmetros que foram utilizados pelo NSGA-II na otimização do problema proposto. Como não houve variação nos resultados, foi realizada uma mesclagem na definição de seus parâmetros, considerando a repetição das boas configurações apresentadas nos resultados acima. Assim sendo, os valores considerados foram os que apareceram com maior frequência nos ranqueamentos apresentados acima.

Após as definições de quais parâmetros seriam utilizados pelo NSGA-II (tabela 4.10), foi realizado um *grid search* para definir o número de gerações que seria utilizado no processo de otimização do problema proposto.

Tabela 4.10: Parâmetros do NSGA-II

Parâmetros	Configuração
População inicial P	200
Número de gerações	400
Tamanho torneio	2
Taxa de <i>crossover</i>	100%
Índice de distribuição SBX	15
Taxa de mutação	$1/\text{size}(\mathbf{S})$
Índice de distribuição PM	15

Com os parâmetros definidos, os algoritmos foram testados no ambiente de testes com as especificações detalhadas no início deste capítulo.

4.2 Algoritmo Evolutivo Lexicográfico X CPGA, ECPGA e RCMA

Para exibir os resultados apresentados nesta seção e na seção seguinte, gráficos de linhas foram geradas para mostrar os valores de cada objetivo dos algoritmos apresentados. O CPGA, por ter sido o primeiro algoritmo a tratar o problema de maximização da cobertura redundante em RSVSF, serviu como linha base para os demais algoritmos. Desta forma, os valores obtidos pelos objetivos dos algoritmos foi subtraído do CPGA na execução equivalente. Assim, os gráficos apresentam as médias e o erro padrão com base no CPGA, após uma série de 20 execuções para um número determinado de sensores.

A figura 4.2 mostra os resultados obtidos pelos algoritmos Lexicográfico, CPGA (linha base), ECPGA e RCMA, considerando o objetivo cobertura (C). Como pode ser constatado, o AE lexicográfico apresentou resultados superiores tanto na média quanto no erro padrão na maioria dos cenários quando comparado aos algoritmos gulosos. Nos cenários com 30 sensores, o AE Lexicográfico obteve 8.4% a mais de cobertura comparado ao CPGA, 3.5% quando comparado ao RCMA e 3.4 com relação ao ECPGA. O RCMA foi o algoritmo guloso que obteve melhor valor de cobertura na maioria dos cenários.

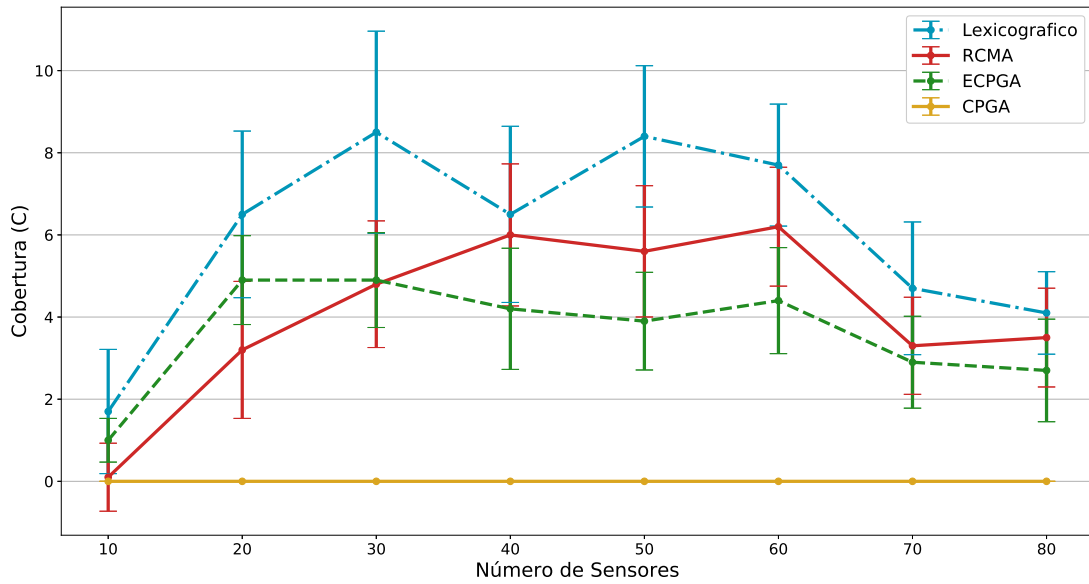


Figura 4.2: Resultados de cobertura para 50 alvos e diferentes números de sensores visuais.

Os resultados da redundância mínima considerando os mesmos cenários com 50 alvos estão representados na figura 4.3. Da mesma forma que na cobertura, o AE lexicográfico alcançou resultados mais relevantes do que as abordagens gulosas (4.11). Os resultados indicam também que a diferença no desempenho do AE Lexicográfico em comparação aos gulosos aumenta conforme a quantidade de sensores também aumenta. A redundância mínima depende fortemente da densidade espacial dos sensores, uma vez que estes precisam estar mais próximos uns dos outros para que o objetivo seja otimizado e aumentado. Com uma quantidade reduzida de sensores, a exemplo dos cenários iniciais com 10 e 20, é difícil otimizar os valores de redundância mínima, pois a distância de posicionamento dos sensores impõe uma restrição para este problema, neste caso, os valores obtidos são parecidos com os outros algoritmos. Mas à medida em que o número de sensores implantados na rede cresce, há uma tendência do AE Lexicográfico alcançar resultados mais relevantes, distanciando-se dos valores obtidos pelos algoritmos gulosos que estão sendo comparados.

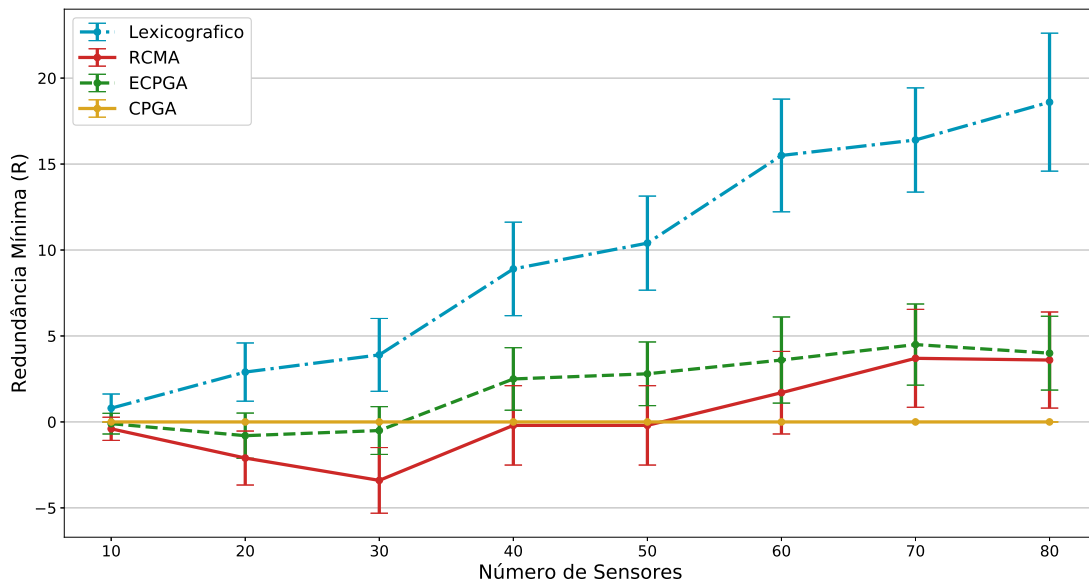


Figura 4.3: Resultados de redundância mínima para 50 alvos e diferentes números de sensores visuais.

A redundância média também foi avaliada (figura 4.4). Neste caso, pode-se observar um comportamento semelhante por parte dos algoritmos, com referência ao objetivo apresentado anteriormente (redundância mínima).

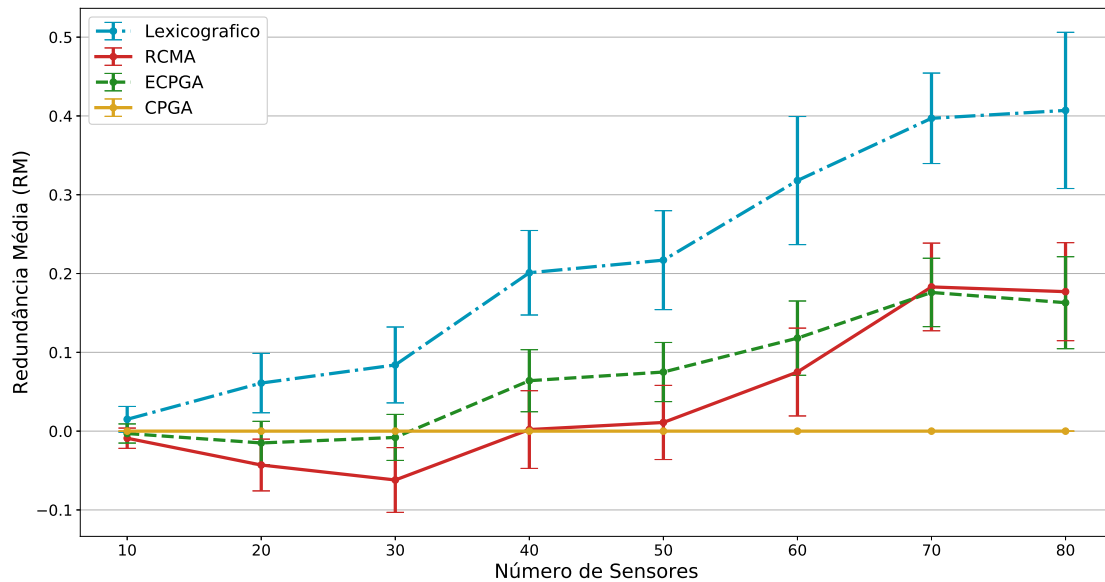


Figura 4.4: Resultados de redundância média para 50 alvos e diferentes números de sensores visuais.

A tabela 4.11 apresenta detalhadamente as comparações obtidas entre o AE lexicográfico e os algoritmos gulosos. Essa análise compara cenário por cenário e verifica se os valores de cobertura redundante (C , R ou RM) foram superados ou não. Também é apresentado quantas vezes o algoritmo evolutivo ganhou, empatou ou perdeu para os algoritmos gulosos apresentados, analisando cada um dos objetivos otimizados.

Analisando individualmente cada cenário, destaca-se que o algoritmo evolutivo obteve cobertura estritamente maior em 141 dos 160 cenários com relação ao CPGA, 105 quanto ao ECPGA e 88 tendo em consideração o RCMA.

O algoritmo evolutivo lexicográfico obteve resultados inferiores principalmente nos cenários com 40 sensores, considerando o objetivo C . Mesmo com resultados inferiores em alguns cenários, a diferença variava entre 1% e 3%.

Tabela 4.11: Análise dos algoritmos lexicográfico e gulosos comparando cada cenário (50 alvos) individualmente.

Algoritmos	GC	EC	PC	GR	ER	PR	GRM	ERM	PRM
Lexicográfico x CPGA	141	11	8	129	18	13	130	14	16
Lexicográfico x ECPGA	105	32	23	135	17	8	132	15	13
Lexicográfico x RCMA	88	38	34	143	12	5	141	10	9

Legenda: G = ganhos, E = empates e P = perdas. C , R e RM correspondem aos objetivos. Ex. GC = ganhos na cobertura, ER = ganhos na redundância, PRM = perdas na redundância média.

Com a intenção de avaliar o desempenho do algoritmo evolutivo lexicográfico proposto quando mais alvos estão disponíveis, as mesmas configurações de cenários também foram consideradas para 100 alvos, consoante as figuras 4.5, 4.6 e 4.7.

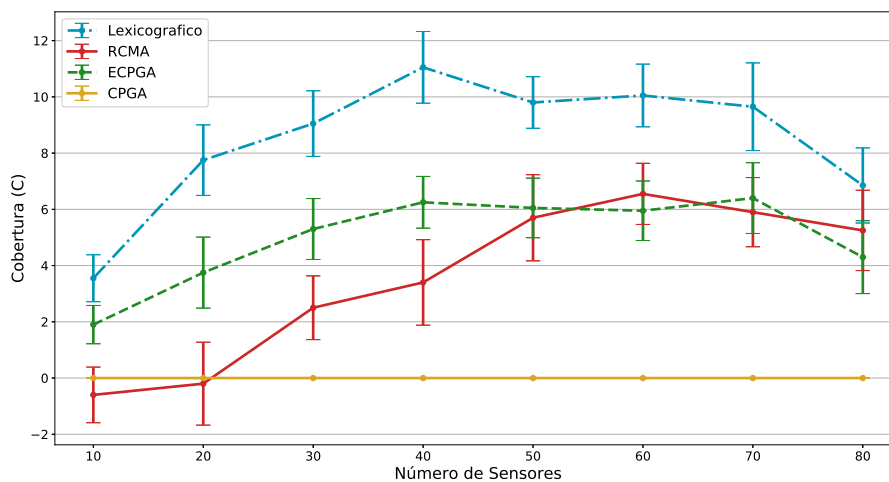


Figura 4.5: Resultados de cobertura para 100 alvos e diferentes números de sensores visuais.

Novamente, o AE lexicográfico apresentou os maiores resultados considerando os valores médios, como pode ser visto na figura 4.5. Com o aumento do número de alvos na rede, houve também uma maior diferença nos resultados obtidos pelo AE Lexicográfico, o que pode ser constatado no gráfico apresentado. Na média dos cenários com 40 sensores, o AE Lexicográfico obteve 11% a mais de cobertura com relação ao CPGA, 4,8% com relação ao ECPGA e 7,4% quando comparado ao RCMA.

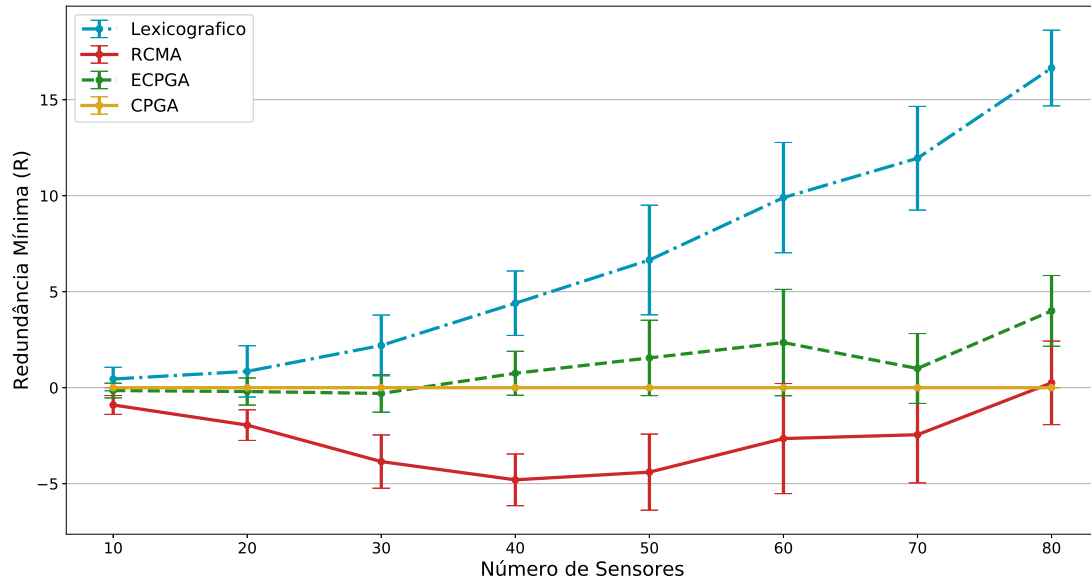


Figura 4.6: Resultados de redundância mínima para 100 alvos e diferentes números de sensores visuais.

O aumento de alvos na rede manteve o desempenho do AE lexicográfico nos objetivos Redundância Mínima e Redundância Média, como pode ser visto nas figuras 4.6 e 4.7 respectivamente. Na medida em que a quantidade de sensores aumenta, a proximidade dos sensores faz com que os algoritmos evolutivos possam otimizar melhor as redundâncias mínima e média, garantindo assim resultados mais relevantes com relação aos algoritmos gulosos.

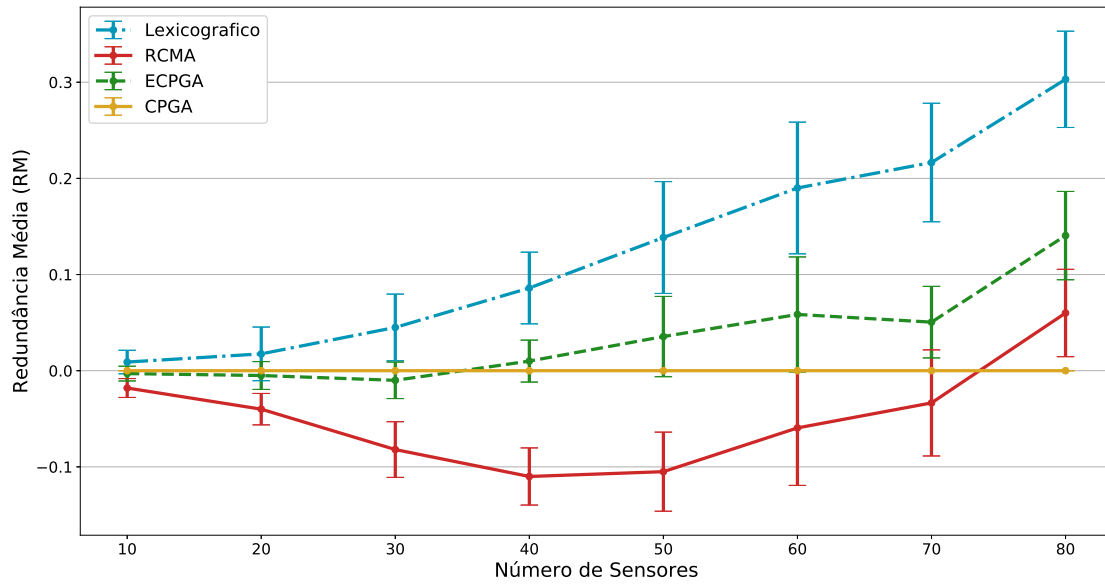


Figura 4.7: Resultados de redundância média para 100 alvos e diferentes números de sensores visuais.

A tabela 4.12 mostra a análise detalhada comparando o AE lexicográfico com as abordagens gulosas para os cenários com 100 alvos.

Pode-se notar que com o aumento de alvos na rede, o desempenho do algoritmo evolutivo lexicográfico proposto melhorou consideravelmente, tendo menos perdas que os testes realizados com 50 alvos (tabela 4.11). Isso pode ser atribuído à dificuldade de otimizar uma rede com uma quantidade reduzida de alvos, visto que estes podem estar mais dispersos na rede.

Tabela 4.12: Análise dos algoritmos lexicográfico e gulosos comparando cada cenário (100 alvos) individualmente.

Algoritmos	GC	EC	PC	GR	ER	PR	GRM	ERM	PRM
Lexicográfico x CPGA	159	1	0	129	12	19	123	11	26
Lexicográfico x ECPGA	140	12	8	125	14	21	124	13	23
Lexicográfico x RCMA	141	10	9	147	9	4	147	9	4

Legenda: G = ganhos, E = empates e P = perdas. C, R e RM correspondem aos objetivos. Ex. GC = ganhos na cobertura, ER = ganhos na redundância, PRM = perdas na redundância média.

Para analisar detalhadamente o comportamento do AE lexicográfico nos cenários que houveram perdas, a figura 4.8 apresenta a evolução da população no cenário

que possui 60 sensores e 100 alvos. Neste cenário, o AE lexicográfico obteve 79,00% de cobertura, enquanto o RCMA conseguiu 85,00%.

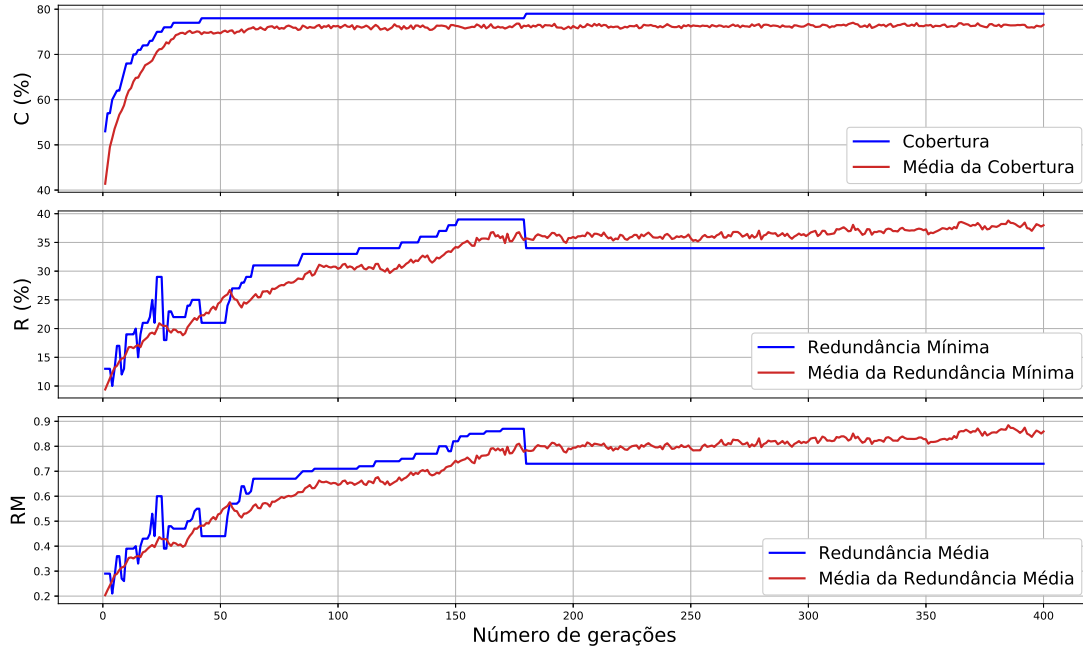


Figura 4.8: Evolução da população do algoritmo evolutivo lexicográfico na 6ª execução.

Nota-se que o AE lexicográfico obteve convergência rápida logo nas primeiras gerações, considerando o objetivo cobertura. Entre as gerações 150 e 200 ele atingiu seu ponto máximo, não conseguindo mais encontrar outras soluções de C . Destaca-se que no mesmo instante que C aumenta, os demais objetivos tiveram uma queda no seu valor. Vale destacar que, para o AE evolutivo, o melhor indivíduo da população é aquele que possui a maior cobertura, seguindo da redundância mínima como objetivo secundário e redundância média com a última prioridade. É possível observar que a média da população nos objetivos R e RM são maiores que a solução ótima, visto as considerações anteriores.

A perda de “rendimento” por parte do AE lexicográfico neste cenário pode ser atribuída ao algoritmo cair em um ótimo local, por conta da semelhança das soluções candidatas, portanto, não havendo diversidade nessas soluções. Vale destacar que as abordagens evolutivas nem sempre encontram a(s) solução(ões) ótima(s) global(is), mas por se tratar de uma meta-heurística de busca exaustiva, considerando processos robustos de otimização, a obtenção desses valores torna-se provável.

4.3 Comparação entre o NSGA-II x AE Lexicográfico, CPGA, ECPGA e RCMA

Nesta seção serão comparados as abordagens evolutivas propostas neste trabalho com as abordagens gulosas para solucionar o problema de maximização da cobertura redundante. Os cenários e testes foram os mesmos realizados na seção anterior, sendo apenas adicionados os resultados do NSGA-II para comparação com os demais algoritmos.

Conforme mencionado anteriormente, ao contrário do AE lexicográfico e as abordagens gulosas, o NSGA-II apresenta um conjunto de soluções que aproximam a fronteira de Pareto, no qual é/são escolhida(s) a(s) solução(ões) mais adequada ao problema proposto, ou a partir da preferência do projetista. Assim, para realizar a comparação com os outros algoritmos, foram escolhidas três soluções de valores máximos de cada objetivo. Desta forma, os resultados possuem um valor de cobertura, um valor de redundância mínima e um valor de redundância média para a solução de máxima C , para a solução de máxima R e para a solução de máxima RM .

A figura 4.9 apresenta os valores obtidos pelos algoritmos abordados, considerando o cenário com 50 alvos e a variação de sensores. Quando considerado a cobertura para máxima C , o NSGA-II apresenta resultado superior em todos os casos apresentados, já para as máximas dos demais objetivos (R e RM), os valores foram inferiores até mesmo com relação ao CPGA (linha base). Isso se deve a cobertura tender a ser “sacrificada” para o aumento da redundância mínima e conseqüentemente da redundância média.

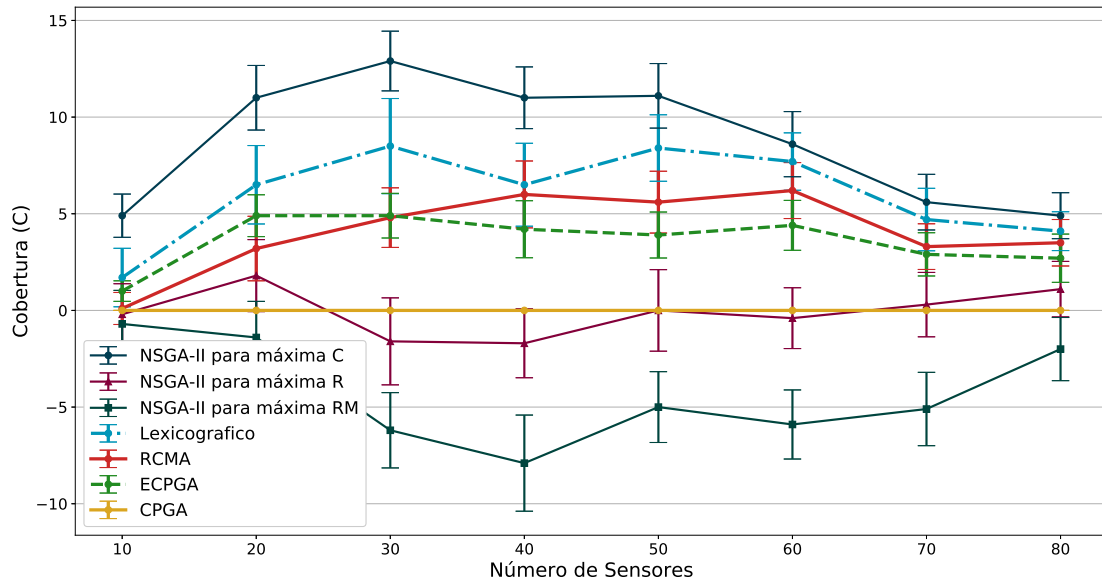


Figura 4.9: Resultados de cobertura para 50 alvos e diferentes números de sensores visuais.

Já a figura 4.10 exhibe os resultados referentes à redundância mínima. Para este objetivo, como já era esperado, a redundância mínima para máxima R e RM possuem resultados muito próximos. A redundância mínima para máxima C consegue resultados superiores em referência ao AE evolutivo e as abordagens gulosas, havendo igualdade de valores médios somente nos cenários com 10 e 20 sensores.

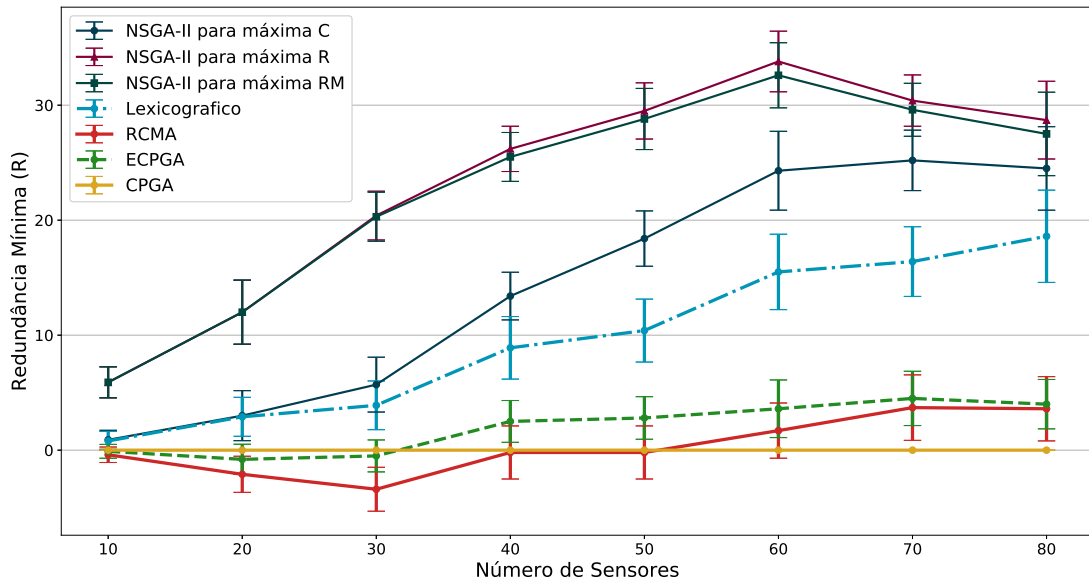


Figura 4.10: Resultados de redundância mínima para 50 alvos e diferentes números de sensores visuais.

A figura 4.11 traz os resultados referentes ao objetivo redundância média. Novamente, ao considerar as máximas de R e RM o NSGA-II apresenta os melhores resultados em referência aos demais, assim como a máxima redundância média de C .

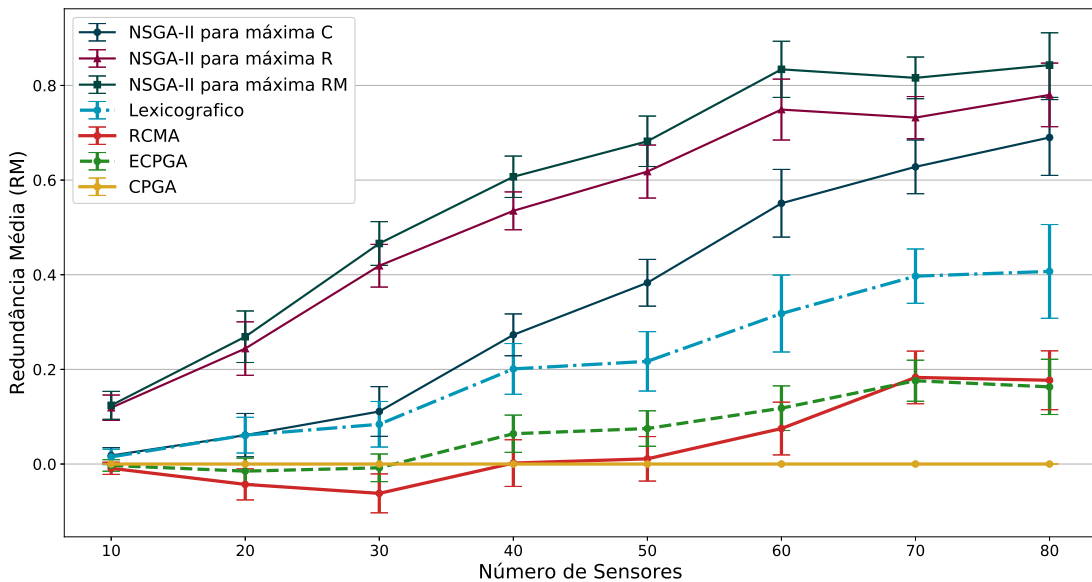


Figura 4.11: Resultados de redundância média para 50 alvos e diferentes números de sensores visuais.

Da mesma forma de como foi realizado nos cenários com 50 alvos, as figuras 4.12, 4.13 e 4.14 apresentam os resultados referentes a cada objetivo, de modo a observar o comportamento dos algoritmos aumentando a quantidade de alvos para 100.

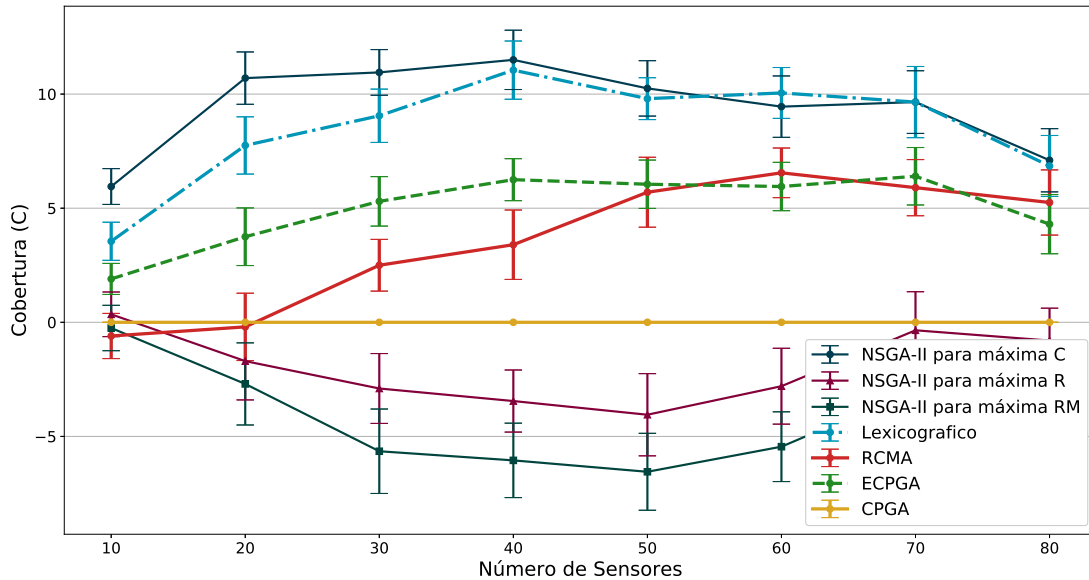


Figura 4.12: Resultados de cobertura para 100 alvos e diferentes números de sensores visuais.

Os dois algoritmos evolutivos propostos obtiveram maiores resultados para cobertura quando comparados aos algoritmos gulosos, como pode ser visto na figura 4.12. Ao analisar as abordagens evolutivas individualmente, a partir de seus valores médios, o AE lexicográfico obteve média maior nos cenários com 60 sensores, 10% para o AE lexicográfico contra 9,5% do NSGA-II para máxima C . Para a configuração do cenário com 70 sensores, os algoritmos propostos empataram, enquanto nas demais configurações, exceto nos cenários com 60 sensores, o NSGA-II para máxima C obteve os maiores resultados na maioria dos casos.

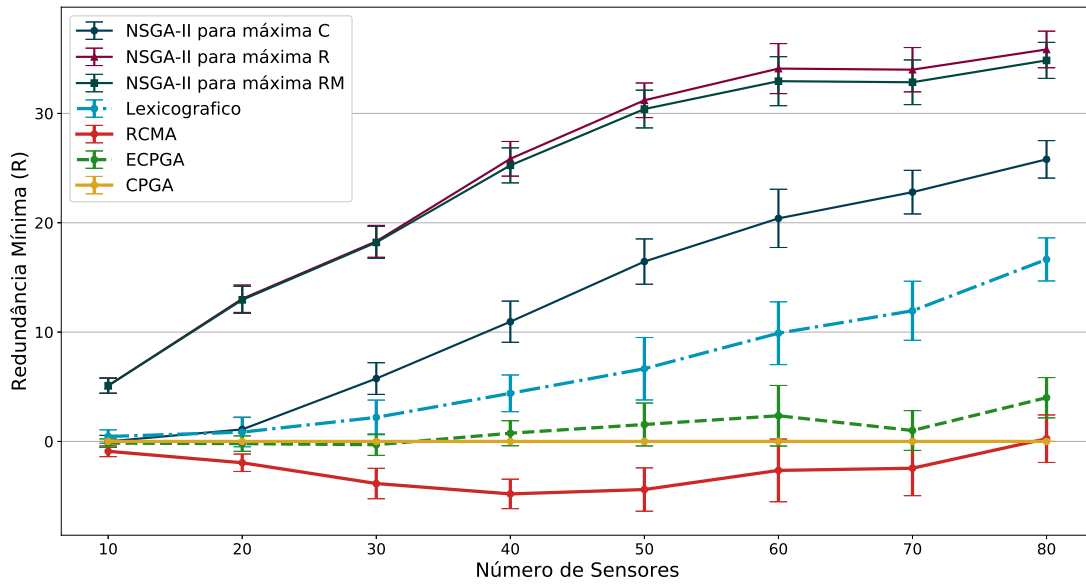


Figura 4.13: Resultados de redundância mínima para 100 alvos e diferentes números de sensores visuais.

Quando os objetivos são redundâncias mínima (figura 4.13) e média (figura 4.14), assim como nos cenários com 50 alvos, há um distanciamento dos resultados por parte do NSGA-II ao considerar os máximos de todos os objetivos. Exceto para a máxima de C , que nos cenários com 10 e 20 sensores os resultados são bem próximos ou iguais ao AE Lexicográfico. Novamente, o ECPGA foi o algoritmo guloso com melhor desempenho.

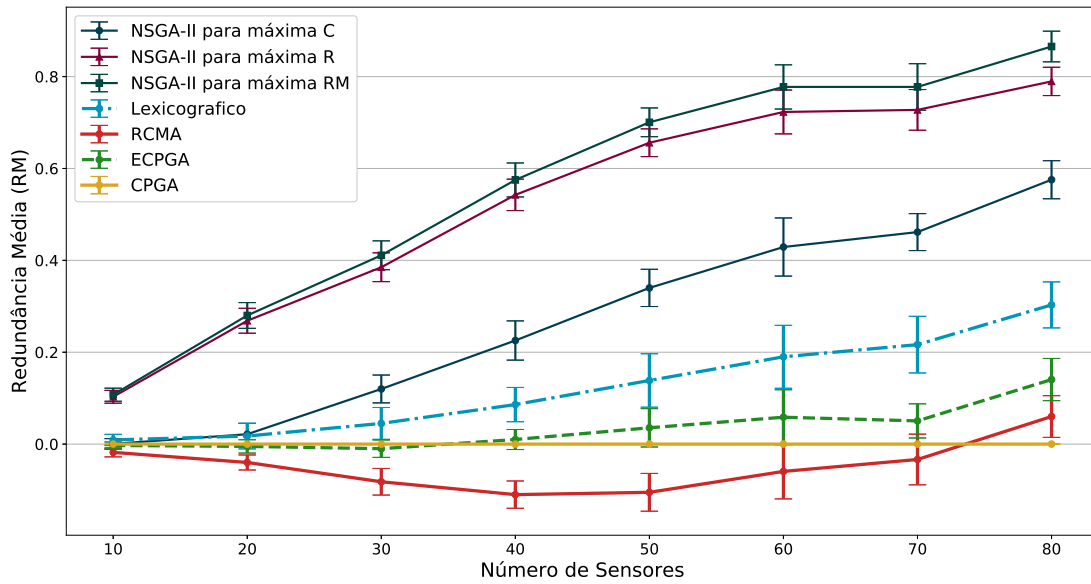


Figura 4.14: Resultados de redundância média para 100 alvos e diferentes números de sensores visuais.

Ao considerar o mesmo cenário apresentado para analisar a evolução do AE lexicográfico (figura 4.8), que possui 60 sensores e 100 alvos, as figuras 4.15 e 4.16 apresentam a frente de soluções não-dominadas do NSGA-II (aproximação da fronteira de Pareto) com o conjunto de soluções ótimas encontrado pelo NSGA-II, juntamente aos resultados dos demais algoritmos. As figuras apresentam as perspectivas de dois objetivos considerados, sendo $C \times R$ (figura 4.15) e $C \times RM$ (figura 4.16).

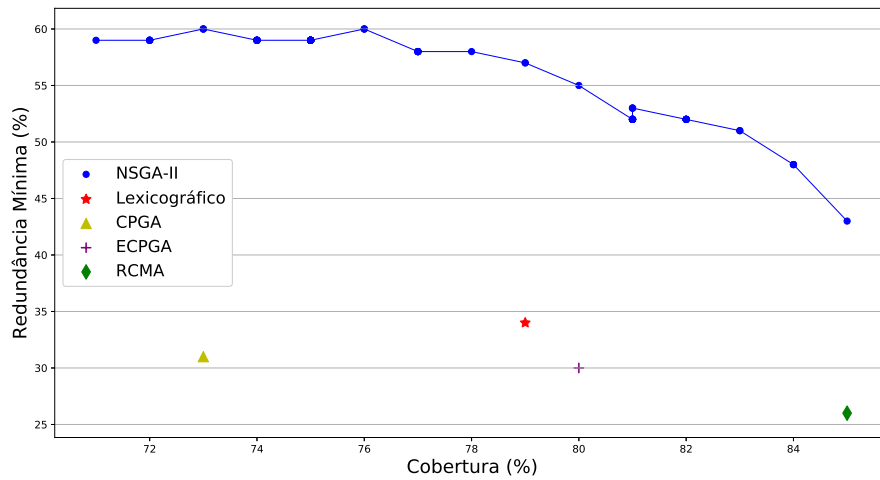


Figura 4.15: Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura x redundância mínima, no cenário com 60 sensores.

É notável que a fronteira de soluções encontrada pelo NSGA-II cobre todas as soluções que estão sendo comparadas, afirmando que abordagens *a posteriori* são excelentes propostas para otimização envolvendo múltiplos objetivos. Sendo assim, mesmo considerando o valor máximo da cobertura como a solução a ser utilizada após o processo de otimização, esta também obtém resultados maiores que as abordagens gulosas e até mesmo que a abordagem lexicográfica, quando todos os objetivos são considerados.

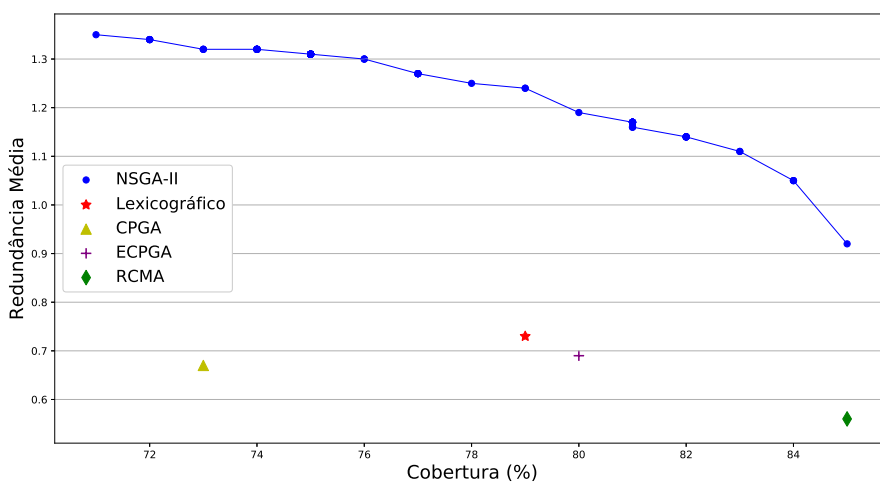


Figura 4.16: Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura x redundância média.

Como houve um certo nível de proximidade na média dos resultados considerando o objetivo cobertura, as figuras 4.17 e 4.18 apresentam as soluções de todos os algoritmos, com perspectivas de dois objetivos conforme mencionado acima. Foi considerado o cenário da 40.^a execução da série de testes, com 80 sensores. Observa-se que neste caso, o AE lexicográfico obteve maior cobertura e redundância mínima com relação aos algoritmos gulosos, mas com redundância média inferior quando comparado aos mesmos. Tendo em vista as duas situações (cenários com 60 e 80 sensores), a frente de soluções não-dominadas obtida a partir do NSGA-II supera todas as soluções que estão sendo comparadas ao considerar os três objetivos propostos.

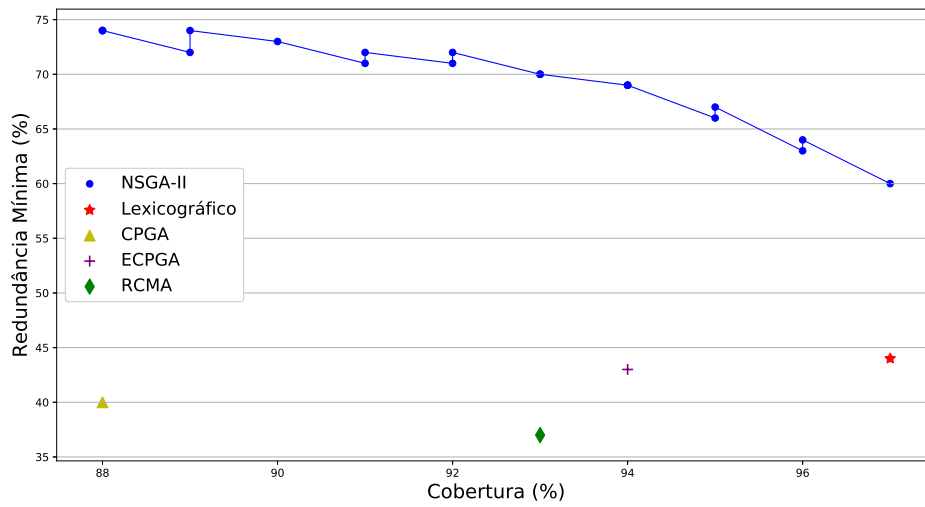


Figura 4.17: Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura e redundância média.

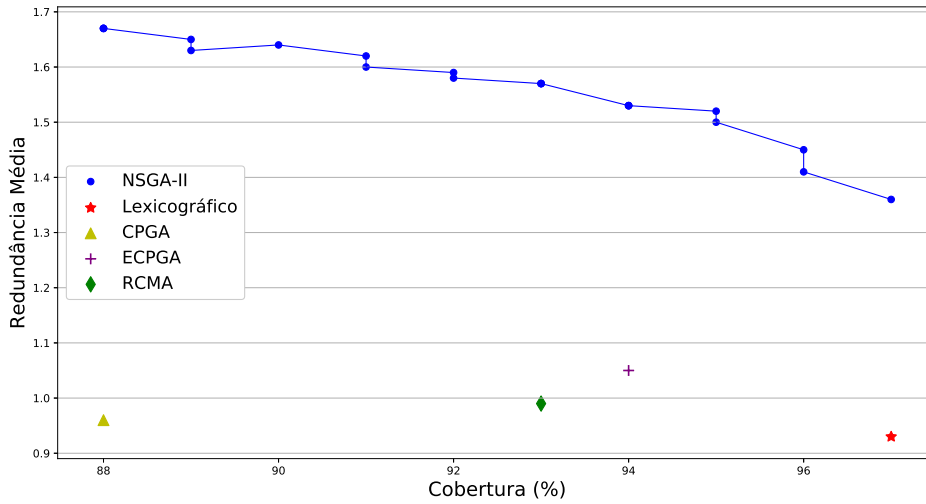


Figura 4.18: Resultado da fronteira de Pareto após otimização. Perspectiva das soluções cobertura e redundância média.

4.4 Custo Computacional

Enquanto por um lado os algoritmos gulosos apresentarem os piores resultados na maioria dos cenários, o seu custo computacional é menor quando comparados as abordagens evolutivas. Desta forma, para calcular o custo que cada algoritmo possui, foi considerado o número de vezes que o algoritmo testa se o alvo está dentro ou fora do FoV do sensor visual. A tabela 4.13 apresenta o valor médio do número de testes realizados considerando os cenários com 50 alvos, e para 100 alvos os resultados são apresentados na tabela 4.14.

Tabela 4.13: Custo computacional considerando as médias dos testes com 50 alvos

Nº de sensores	NSGA-II	Lexicográfico	RCMA	ECPGA	CPGA
10	38.869.177,70	2.930.720,05	7.490	4.815	4.500
20	77.680.732,55	11.698.860,05	15.510	9.640	9.000
30	116.467.444,05	26.316.588,45	23.590	14.560	13.500
40	155.288.790,50	46.750.225,90	32.540	19.372	18.000
50	194.379.243,25	73.172.471,05	42.107	24.185	22.500
60	233.221.844,05	105.357.620,95	51.457	29.020	27.000
70	272.121.340,65	143.348.169,00	62.210	33.847	31.500
80	311.172.804,65	187.415.551,40	72.852	38.610	36.000

Com o aumento do número de alvos para 100, o número de checagens aumentou consideravelmente, como pode ser visto na tabela 4.14. Na medida que número de alvos da rede aumenta, o número de checagens tendem a subir, pois, a base do cálculo de aptidão é verificar se um alvo está dentro ou fora dos *FoV* sensores visuais.

Tabela 4.14: Custo computacional considerando as médias dos testes com 100 alvos

Nº de sensores	NSGA-II	Lexicográfico	RCMA	ECPGA	CPGA
10	78.133.162,55	5.881.424,65	14.210	10.805	9.000
20	156.098.009,10	23.500.438,25	28.590	21.890	18.000
30	234.375.228,75	52.931.656,65	43.385	32.230	27.000
40	312.463.633,25	94.090.820,65	58.125	43.235	36.000
50	390.868.463,35	147.172.329,25	74.800	53.800	45.000
60	469.182.872,70	211.973.408,45	92.185	63.980	54.000
70	547.750.986,50	288.608.830,30	112.165	75.280	63.000
80	626.095.771,15	377.036.969,15	131.150	86.375	72.000

Por se tratar de abordagens evolutivas, eram esperados custos computacionais elevados. Para realizar a otimização, os algoritmos evolutivos requerem muito mais verificações, pois, eles tentam explorar globalmente o espaço de busca, ao contrário das abordagens gulosas. Todavia, por se tratar de algoritmos centralizados, as execuções nos *sinks* ou no servidor da aplicação podem ser facilmente efetuadas sem comprometer o funcionamento das RSVSF, além de não interferir no tempo de vida útil dos nós-sensores. Com os resultados apresentados, avaliando os ganhos de cada objetivo (C , R e RM), o uso de abordagens evolutivas podem compensar o custo computacional. Considerando os algoritmos evolutivos, a abordagem lexicográfica apresentou menor custo.

Por fim, os algoritmos propostos nesse trabalho funcionam de forma centralizada, assim como as abordagens gulosas utilizadas para comparação. Conseqüentemente, faz-se necessário conhecer previamente a rede que será otimizada, sabendo a localização dos nós-sensores e alvos presentes nela. Na verdade, isso não é um problema tratando-se de redes de sensores sem fio, posto que pode-se utilizar algoritmos para descoberta dos objetos presentes nas mesmas. Para os algoritmos evolutivos que realizam o processo de busca e otimização a partir de soluções candidatas, é necessário que o processamento seja realizado em algum servidor central, ou em algum *sink* que esteja equipado com processador e memória. Desta forma, a otimização é realizada sem comprometer a eficácia da rede em sua totalidade, deixando de consumir recursos dos nós-sensores, pois, eles só serão requisitados para receber suas respectivas orientações após o processo de otimização (no caso do AE lexicográfico), ou após o projetista escolher qual configuração mais adequada para o problema (no caso do NSGA-II).

Capítulo 5

Considerações Finais

Para aplicações como monitoramento e rastreamento, o processo de otimização de redes com sensores equipados com câmeras torna-se fundamental para obtenção de informações valiosas. Em áreas que se tornaram tendências nos últimos anos, por exemplo, a Internet das coisas (*Internet of Things* (IoT)) e sistemas de cidades inteligentes, redes que possuem câmeras implantadas, como é o caso das RSVSF, inspiram muita atenção ocasionando importantes desenvolvimentos nessa área de pesquisa.

Neste trabalho, foi tratado o problema de maximização da cobertura redundante para as RSVSF. Este problema está focado em encontrar as melhores orientações dentre um conjunto de sensores, para maximizar os objetivos propostos no monitoramento de alvos. Desta forma, ao maximizar a cobertura redundante de uma RSVSF, é proporcionado o aumento de disponibilidade das mesmas, o que o torna uma ferramenta poderosa para aplicações que necessitam desse recurso. Além disso, abordagens evolutivas para tratar do problema de maximização apresentado nesta dissertação é uma novidade, e os resultados apresentados mostram os benefícios com o uso dessas abordagens.

Primeiramente, este estudo propôs a formalização matemática do problema de maximização da cobertura redundante para RSVSF, que pode contribuir para futuros estudos sobre essa área de pesquisa. A partir desta formalização foi desenvolvido um algoritmo evolutivo baseado na abordagem de otimização lexicográfica *a priori*, e também o algoritmo evolutivo de otimização *a posteriori* baseado no NSGA-II, utilizando o *JMetal Framework*.

Após o desenvolvimento e implementação dos algoritmos propostos, um método de otimização de parâmetros foi utilizado para encontrar as configurações dos parâmetros mais eficazes para solucionar o problema proposto. Os resultados apresentados evidenciam os benefícios de utilizar estes métodos, tais como: modular os parâmetros utilizados pelos algoritmos evolutivos, adequando-os a realidade da aplicação.

Por se tratar de abordagens de otimização mais robustas, o uso dos algoritmos

evolutivos se torna mais viável do que abordagens gulosas, principalmente quando o problema envolve múltiplos objetivos. De todo modo, este trabalho mostrou a importância de realizar a otimização dos parâmetros dos algoritmos evolutivos, uma vez que estes proporcionam melhores avaliações e resultados mais promissores.

Os resultados deste trabalho demonstraram que utilizar algoritmos evolutivos para tratar de problemas multiobjetivos é uma opção interessante, podendo ser aplicados para resolver tais problemas de diversas formas e com resultados prósperos. Apesar do custo computacional ser alto, ainda assim vale apenas aplicar essas meta-heurísticas para resolução de problemas que envolvem múltiplos objetivos.

Como trabalhos futuros, será realizada uma verificação adicional, considerando o impacto da aplicação dos algoritmos propostos em redes de sensores visuais sem fio reais. Os custos de comunicação serão avaliados e o tempo de atribuição geral das orientações comparadas será avaliado. Outros trabalhos futuros relevantes estarão preocupados com uma especificação adicional dos objetivos de otimização, considerando a prioridade dos alvos. A ideia é definir diferentes níveis de prioridade para os alvos, e essa característica será considerada para orientar como a cobertura de alvos será calculada.

Referências Bibliográficas

ABBASI, A. A.; YOUNIS, M. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, Elsevier, v. 30, n. 14-15, p. 2826–2841, 2007.

AGRAWAL, R. B.; DEB, K.; AGRAWAL, R. Simulated binary crossover for continuous search space. *Complex systems*, v. 9, n. 2, p. 115–148, 1995.

AI, J.; ABOUZEID, A. A. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, Springer, v. 11, n. 1, p. 21–41, 2006.

AKYILDIZ, I. F.; MELODIA, T.; CHOWDHURY, K. R. A survey on wireless multimedia sensor networks. *Computer networks*, Elsevier, v. 51, n. 4, p. 921–960, 2007.

ALAEI, M.; BARCELO-ORDINAS, J. M. Node clustering based on overlapping FoVs for wireless multimedia sensor networks. *IEEE Wireless Communications and Networking Conference, WCNC*, 2010. ISSN 15253511.

ALMALKAWI, I. T. et al. Wireless multimedia sensor networks: Current trends and future directions. *Sensors*, v. 10, n. 7, p. 6662–6717, 2010. ISSN 14248220.

AMMARI, H. M. Coverage in Wireless Sensor Networks: A Survey. *Network Protocols and Algorithms*, v. 2, n. 2, 2010. ISSN 1943-3581.

BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. *Evolutionary computation 1: Basic algorithms and operators*. [S.l.]: CRC press, 2000. v. 1.

BELLMORE, M.; NEMHAUSER, G. L. The traveling salesman problem: a survey. *Operations Research*, INFORMS, v. 16, n. 3, p. 538–558, 1968.

BORGES, L. M.; VELEZ, F. J.; LEBRES, A. S. Survey on the characterization and classification of wireless sensor network applications. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 4, p. 1860–1890, 2014.

BOYINBODE, O. et al. A Survey on Clustering Algorithms for Wireless Sensor Networks. In: *13th International Conference on Network-Based Information Systems (NBIS), 2010*. [S.l.: s.n.], 2010. v. 8, n. 5, p. 259–268. ISBN 978-1-4244-8053-1. ISSN 2157-0418.

- BROWNE, C. B. et al. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, IEEE, v. 4, n. 1, p. 1–43, 2012.
- CAI, Y. et al. Target-oriented scheduling in directional sensor networks. In: IEEE. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE. [S.l.], 2007. p. 1550–1558.
- CHANG, C.-Y.; HSIAO, C.-Y.; CHANG, C.-T. The k-barrier coverage mechanism in wireless visual sensor networks. In: IEEE. *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. [S.l.], 2012. p. 2318–2322.
- CHARFI, Y.; WAKAMIYA, N.; MURATA, M. Challenging issues in visual sensor networks. *IEEE Wireless Communications*, v. 16, n. 2, p. 44–49, 2009. ISSN 15361284.
- CHAUDHARY, M.; PUJARI, A. K. *Q-coverage problem in wireless sensor networks*. [S.l.], 2009. v. 5408 LNCS. 325–330 p. ISSN 03029743 16113349. ISBN 3540922946 | 9783540922940.
- COELLO, C. A. C. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, Springer, v. 1, n. 3, p. 269–308, 1999.
- COELLO, C. A. C. et al. *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2007. v. 5.
- CORNE, D. W.; KNOWLES, J. D.; OATES, M. J. The pareto envelope-based selection algorithm for multiobjective optimization. In: SPRINGER. *International Conference on Parallel Problem Solving from Nature*. [S.l.], 2000. p. 839–848.
- COROIU, A. M. Tuning model parameters through a genetic algorithm approach. In: IEEE. *Intelligent Computer Communication and Processing (ICCP), 2016 IEEE 12th International Conference on*. [S.l.], 2016. p. 135–140.
- COSTA, D. G.; DURAN-FAUNDEZ, C. Assessing availability in wireless visual sensor networks based on targets perimeters coverage. *Journal of Electrical and Computer Engineering*, Hindawi Publishing Corp., v. 2016, p. 10, 2016.
- COSTA, D. G.; GUEDES, L. A. The coverage problem in video-based wireless sensor networks: A survey. *Sensors*, Molecular Diversity Preservation International, v. 10, n. 9, p. 8215–8247, 2010.
- COSTA, D. G. et al. Availability issues in wireless visual sensor networks. 2014.
- COSTA, D. G. et al. Enhancing redundancy in wireless visual sensor networks for target coverage. In: ACM. *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. [S.l.], 2014. p. 31–38.
- COSTA, D. G. et al. Selecting redundant nodes when addressing availability in wireless visual sensor networks. *Proceedings - 2014 12th IEEE International Conference on Industrial Informatics, INDIN 2014*, p. 130–135, 2014.

- COSTA, D. G. et al. Optimal sensing redundancy for multiple perspectives of targets in wireless visual sensor networks. In: IEEE. *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. [S.l.], 2015. p. 185–190.
- COSTA, D. G.; VASQUES, F.; PORTUGAL, P. Enhancing the availability of wireless visual sensor networks: Selecting redundant nodes in networks with occlusion. *Applied Mathematical Modelling*, v. 42, n. Complete, p. 223–243, 2017.
- DEB, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, MIT Press, v. 7, n. 3, p. 205–230, 1999.
- DEB, K. *Multi-objective optimization using evolutionary algorithms*. [S.l.]: John Wiley & Sons, 2001. v. 16.
- DEB, K.; BEYER, H.-G. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation*, MIT Press, v. 9, n. 2, p. 197–221, 2001.
- DEB, K.; DEB, D. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, Inderscience Publishers Ltd, v. 4, n. 1, p. 1–28, 2014.
- DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 18, n. 4, p. 577–601, 2014.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002.
- DURILLO, J.; NEBRO, A.; ALBA, E. The jmetal framework for multi-objective optimization: Design and architecture. In: *CEC 2010*. Barcelona, Spain: [s.n.], 2010. p. 4138–4325.
- EIBEN, A. E.; SMIT, S. K. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, Elsevier, v. 1, n. 1, p. 19–31, 2011.
- EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing*. [S.l.]: springer, 2015.
- FONSECA, C. M.; FLEMING, P. J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, v. 3, n. 1, p. 1–16, 1995. ISSN 1063-6560.
- GANDIBLEUX, X. et al. *Metaheuristics for multiobjective optimisation*. [S.l.: s.n.], 2004. v. 535. 251 p. ISSN 03044009. ISBN 9783642021466.
- GANGULY, S.; SAHOO, N.; DAS, D. Multi-objective particle swarm optimization based on fuzzy-pareto-dominance for possibilistic planning of electrical distribution systems incorporating distributed generation. *Fuzzy Sets and Systems*, Elsevier, v. 213, p. 47–73, 2013.

HURUIALĂ, P.-C.; URZICĂ, A.; GHEORGHE, L. Hierarchical routing protocol based on evolutionary algorithms for wireless sensor networks. In: IEEE. *Roedunet International Conference (RoEduNet), 2010 9th*. [S.l.], 2010. p. 387–392.

ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: IEEE. *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*. IEEE Congress on. [S.l.], 2008. p. 2419–2426.

JIA, J. et al. Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. *Computers & Mathematics with Applications*, Elsevier, v. 57, n. 11, p. 1756–1766, 2009.

KNOWLES, J.; CORNE, D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: IEEE. *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. [S.l.], 1999. v. 1, p. 98–105.

KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, v. 91, n. 9, p. 992–1007, 2006. ISSN 09518320.

KORTE, B.; LOVÁSZ, L.; SCHRADER, R. *Greedoids*. [S.l.]: Springer Science & Business Media, 2012. v. 4.

LAI, K.-C.; LIN, L.-W. Low-complexity adaptive tree search algorithm for mimo detection. *IEEE Transactions on Wireless Communications*, IEEE, v. 8, n. 7, 2009.

LIAGKOURAS, K.; METAXIOTIS, K. An elitist polynomial mutation operator for improved performance of moeas in computer networks. In: IEEE. *Computer communications and networks (ICCCN), 2013 22nd international conference on*. [S.l.], 2013. p. 1–5.

LIU, L.; MA, H.; ZHANG, X. On directional k-coverage analysis of randomly deployed camera sensor networks. In: IEEE. *Communications, 2008. ICC'08. IEEE International Conference on*. [S.l.], 2008. p. 2707–2711.

LÜCKEN, C. von; BARÁN, B.; BRIZUELA, C. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, Springer, v. 58, n. 3, p. 707–756, 2014.

MARLER, R. T.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, v. 26, n. 6, p. 369–395, 2004. ISSN 1615147X.

MARLER, R. T.; ARORA, J. S. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, Springer, v. 41, n. 6, p. 853–862, 2010.

MUNISHWAR, V. P.; ABU-GHAZALEH, N. B. *ACM Transactions on Sensor Networks (TOSN)*, ACM, v. 9, n. 4, p. 45, 2013.

- NAVIN, A. H. et al. Solving coverage problem in wireless camera-based sensor networks by using genetic algorithm. In: IEEE. *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*. [S.l.], 2010. p. 226–229.
- PEIXOTO, J. P. J.; COSTA, D. G. Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility. *Future Generation Computer Systems*, Elsevier, v. 76, p. 51–62, 2017.
- QU, B. et al. A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. *Swarm and Evolutionary Computation*, Elsevier, 2017. ISSN 22106502.
- RASMUSSEN, S. J.; SHIMA, T. Tree search algorithm for assigning cooperating uavs to multiple tasks. *International Journal of Robust and Nonlinear Control*, Wiley Online Library, v. 18, n. 2, p. 135–153, 2008.
- SILVA, T. F.; COSTA, D. G. Centralized algorithms for redundant coverage maximization in wireless visual sensor networks. *IEEE Latin America Transactions*, IEEE, v. 14, n. 7, p. 3378–3384, 2016.
- SKIENA, S. Dijkstra’s algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Reading, MA: Addison-Wesley, p. 225–227, 1990.
- VACHHANI, V. L. Improving NSGA-II For Solving Multi Objective Function Optimization Problems. 2016.
- VOORNEVELD, M. Characterization of pareto dominance. *Operations Research Letters*, Elsevier, v. 31, n. 1, p. 7–11, 2003.
- WANG, J.; NIU, C.; SHEN, R. Priority-based target coverage in directional sensor networks using a genetic algorithm. *Computers & Mathematics with Applications*, Elsevier, v. 57, n. 11-12, p. 1915–1922, 2009.
- YAP, F. G. H.; YEN, H. H. A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks. *Sensors (Switzerland)*, v. 14, n. 2, p. 3506–3527, 2014. ISSN 14248220.
- YEN, H. H. Efficient visual sensor coverage algorithm in Wireless Visual Sensor Networks. *2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013*, v. 2, p. 1516–1521, 2013.
- YEN, H.-H. Optimization-based visual sensor deployment algorithm in ptz wireless visual sensor networks. In: IEEE. *Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on*. [S.l.], 2015. p. 734–739.
- YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Computer Networks*, v. 52, n. 12, p. 2292–2330, 2008. ISSN 13891286.

ZITZLER, E.; LAUMANN, M.; BLEULER, S. A tutorial on evolutionary multiobjective optimization. In: *Metaheuristics for multiobjective optimisation*. [S.l.]: Springer, 2004. p. 3–37.

ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001. ISSN 03772217.

ZUBEN, F. J. V. Computação Evolutiva: Uma Abordagem Pragmática. *Anais da I Jornada de Estudos em Computação de Piracicaba e Região*, p. 25–45, 2000.