

Otimização Interativa em Engenharia de Software - uma revisão sistemática

Willian Marques Freire

Maringá, Brasil

Abstract

A Engenharia de Software Baseada em Busca (SBSE) é um campo de pesquisa na computação que engloba técnicas de otimização para resolução de problemas complexos de Engenharia de Software (ES). Neste contexto, métricas de software têm sido utilizadas como funções objetivo para medir a qualidade de soluções otimizadas. Porém, as métricas nem sempre são suficientes para avaliar a qualidade das soluções, tornando necessário considerar as preferências do Tomador de Decisão (DM) durante o processo de otimização. Este trabalho apresenta uma revisão sistemática no contexto de Otimização Interativa em SBSE, também conhecida como *Interactive Search-based Software Engineering* (iSBSE) em que foi possível identificar diversos estudos em diversos cenários de ES. Além disso, foram encontradas lacunas nos estudos que permitirão o desenvolvimento de trabalhos futuros no contexto de iSBSE.

Keywords: SBSE, Preferências do tomador decisão, Algoritmos genéticos interativos, otimização interativa

1. Introdução

O processo de desenvolvimento de software é formado por diversas atividades indispensáveis para que os requisitos do cliente se tornem um produto de software, atendendo aspectos desde o planejamento inicial até a entrega do produto final. Neste sentido, a Engenharia de Software visa auxiliar o desenvolvimento de software, abordando técnicas para especificação, projeto e qualidade de software [1].

A Engenharia de Software Baseada em Busca, do inglês Search-based Software Engineering (SBSE), é um campo de pesquisa na computação, que

engloba técnicas de otimização, tais como algoritmos genéticos, para resolução de problemas complexos de Engenharia de Software (ES) [1]. Neste contexto, métricas específicas para cada área são aplicadas como funções objetivo para medir a qualidade das possíveis soluções [2].

Contudo, as métricas nem sempre são suficientes para avaliar a qualidade das soluções geradas, sendo necessário considerar aspectos subjetivos no processo de otimização [3]. Diversos pesquisadores têm estudado a área de Otimização Iterativa, também conhecida como iSBSE¹, buscando novas abordagens para incluir as preferências do engenheiro de software durante o processo de otimização. Além disso, há várias abordagens que tratam a incorporação das preferências do Tomador de Decisão (DM) durante o processo de otimização de forma interativa, seja ela por meio de atribuição de notas indicando a qualidade das soluções otimizadas, seleção de áreas de interesse (Region of Interest – ROI) ou congelamento de partes das soluções [5].

Sendo assim, o objetivo geral deste trabalho é realizar uma revisão sistemática sobre otimização interativa em SBSE. Com esta revisão será possível verificar o estado atual da literatura neste contexto, identificando e analisando trabalhos relacionados. Este trabalho está organizado da seguinte forma: a Seção 2 descreve a metodologia utilizada; a Seção 3 apresenta os resultados; e por fim a Seção 6 conclui o trabalho.

2. Metodologia

Esta seção descreve a metodologia aplicada na realização desta revisão sistemática. A ferramenta StArt (*State of the Art through Systematic Review*) foi utilizada como apoio durante todo o processo de desenvolvimento deste trabalho. O preenchimento de um protocolo descrevendo as etapas que deveriam ser realizadas foi a etapa inicial da revisão. Com a ferramenta StArt foi possível definir a string de busca, importar as referências dos artigos encontrados, excluir referências duplicadas e visualizar as principais informações dos estudos.

O processo envolveu uma pesquisa automática em sites de busca utilizando strings combinadas com palavras chaves. Os trabalhos encontrados foram selecionados com base em critérios de inclusão e exclusão, além da extração de dados correspondentes. Cada uma dessas etapas é detalhada nesta seção.

¹Do inglês *interactive search-based Software Engineering* [4]

A partir do tema de pesquisa relacionado à otimização interativa em SBSE, as seguintes questões de pesquisa foram definidas:

- QP: Quais problemas são resolvidos usando as preferências do DM durante o processo de otimização?
- QS1: Como ocorre a interação humana durante o processo de otimização?
- QS2: Qual algoritmo de otimização é usado?
- QS3: Aprendizagem de Máquina (AM) é usado? Qual algoritmo e para quê?
- QS4: Como é o tipo de instância de entrada?

2.1. Processo de Busca

O procedimento de busca foi realizado nas bibliotecas digitais ACM Library, IEEE Xplore, Science Direct e Scopus.

Para a construção das strings de busca, foram selecionadas palavras-chaves e sinônimos que definiram o escopo da pesquisa. Os termos utilizados foram: *decision maker*, *DM's preferences*, *SBSE*, *decision maker*, *genetic algorithm*, *human-in-the-loop*, *interactive evolutionary search*, *interactive optimization*, *optimization*, *preference-based multiobjective*, *search-based software engineering*, *software engineering*, *user preferences*. Operadores lógicos (AND e OR) foram combinados para vincular os termos de pesquisa.

Foi realizado um refinamento das strings de buscas e após vários ajustes a string foi definida como: ("software engineering") AND ("search based" OR "search-based" OR "sbse") AND ("decision maker" OR "interaction" OR "user preference"). A busca foi realizada considerando trabalhos de 2005 a 2021. A Figura 1 mostra o número de publicações encontradas em cada base de dados. Totalizando 1353 publicações retornadas da pesquisa, sendo que 592 foram da base de dados ACM Library, 39 da IEEE Xplore, 622 da Science Direct e 100 artigos da Scopus.

2.2. Processo de Seleção

A etapa de seleção de estudos consistiu em filtrar as publicações com auxílio da ferramenta StArt, para eliminar estudos que não correspondiam aos critérios de inclusão da revisão sistemática. A princípio foi utilizada

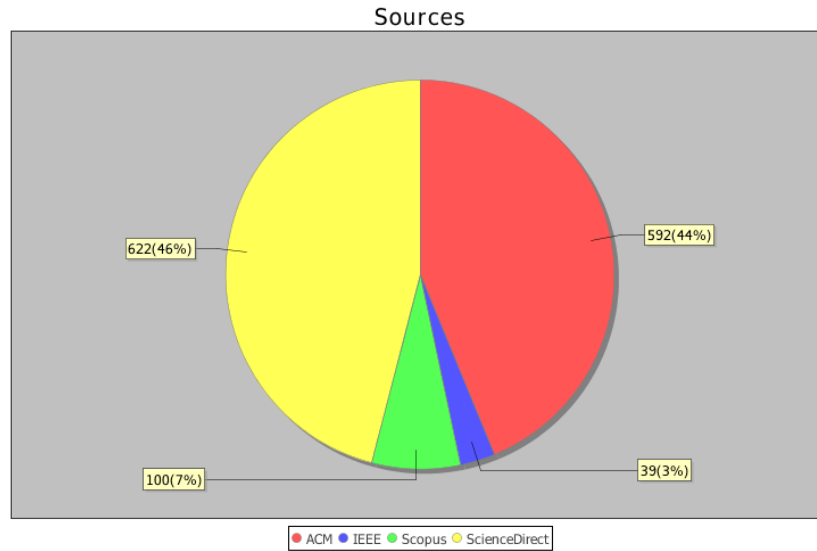


Figura 1: Total de publicações por bibliotecas digitais

uma pontuação que foi calculada para cada artigo, com base no número de palavras-chaves no título e resumo. O cálculo da pontuação foi definido da seguinte forma: 5 pontos para termos encontrado no título do estudo; 3 pontos para termos encontrado no resumo; e 1 ponto para termos encontrado nas palavras-chaves do estudo. A Tabela 1 apresenta os critérios de inclusão e exclusão utilizados para seleção de artigos.

Inclusão	Exclusão
É sobre SBSE usando algoritmos de otimização interativa	Não tem acesso ao texto completo
Apresenta análise qualitativa ou quantitativa	Outros idiomas
Apresenta uma abordagem interativa no processo de otimização	Não se trata de otimização interativa em SBSE
Está em inglês	

Tabela 1: Critérios de inclusão e exclusão

A Figura 2 apresenta a proporção de artigos aceitos (em verde), rejeitados (em vermelho) e duplicados (em azul) na etapa de seleção por meio dos critérios de inclusão e exclusão. Vale ressaltar que nesta etapa foi feita a leitura dos resumos e inspeção das palavras-chaves e do título de cada traba-

lho. Do total de artigos, 78 foram aceitos, 1209 foram rejeitados e 66 foram identificados como duplicados.

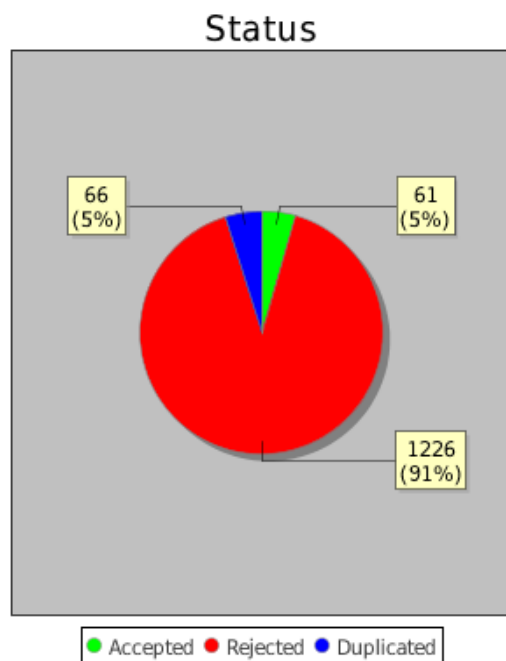


Figura 2: Total de publicações de acordo com o status

2.3. Extração de Dados

A fim de se extrair dados dos trabalhos, foi realizada a leitura dos artigos na íntegra. Divergências foram encontradas em relação à utilização das preferências do DM em diversos trabalhos e foram desconsideradas. Finalmente, 27 publicações foram rejeitadas, e 33 foram aceitas. A Tabela 2 apresenta o número de publicações por bibliotecas digitais, além da quantidade de artigos incluídos manualmente. A primeira coluna apresenta as bases de dados, a segunda coluna apresenta o total de artigos obtidos de cada busca. A terceira e quarta coluna, mostra o número de estudos obtidos por meio das etapas de seleção e extração de dados, respectivamente.

2.4. Síntese

Foi realizada uma análise sobre o número de artigos publicados em cada ano. É possível observar na Figura 3 que estes trabalhos foram publicados entre 2009 à 20021, sendo que em 2013 houve o maior número de publicações.

Base de Dados	Resultados	Seleção	Extração
ACM library	592	34	5
IEEE Xplore	39	9	8
Scopus	100	23	16
Science Direct	622	12	2
Processo Manual	2	2	2
Total	1355	80	33

Tabela 2: Número de publicações do processo busca e extração de dados.

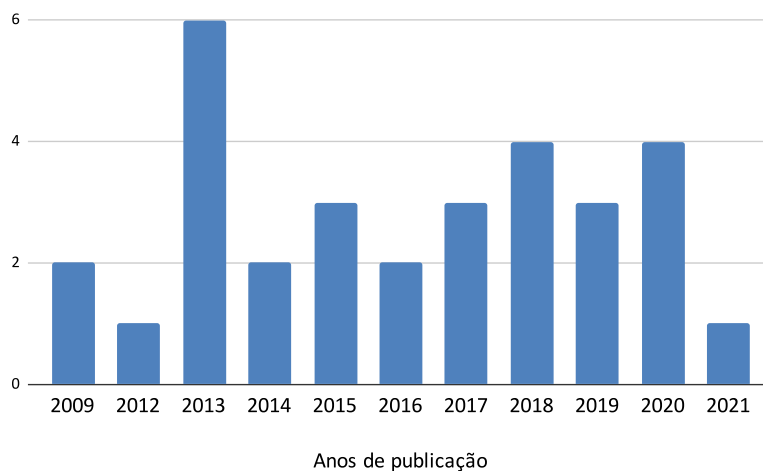


Figura 3: Total de publicações por ano

A Tabela 3 apresenta as informações principais dos artigos analisados nesse estudo. A primeira coluna apresenta o nome do primeiro autor, a segunda e terceira coluna mostra o país e o ano de publicação, respectivamente. A quarta coluna apresenta a base de dados em que foi obtido o artigo. Os trabalhos são apresentados de acordo com o nome do primeiro autor em ordem alfabética. É possível notar por meio desta tabela que o país que mais contribuiu com publicações até o momento foi o Brasil com 11 artigos, totalizando 33% dos trabalhos analisados.

Trabalho (1º Autor)	País	Ano	Base de Dados
Alizadeh [6]	Michigan	2018	IEEE
Aljawawdeh [7]	United Kingdom	2015	ACM
Ali [8]	Noruega	2020	ACM
Araújo [9]	Brasil	2014	Springer
Araújo [10]	Brasil	2017	Springer
Bindewald [11]	Brasil	2019	ACM
Bindewald [12]	Brasil	2015	Springer
Birtolo [13]	Itália	2009	IEEE
Dantas [14]	Brasil	2009	IEEE
El Yamany [15]	Egito	2014	ACM
Ferreira [16]	Brasil	2016	Elsevier
Ferreira [17]	Brasil	2017	Elsevier
Filho [18]	Brasil	2018	ACM
Freire [19]	Brasil	2020	ACM
Ganesh [20]	USA	2016	Elsevier
Jakubovski Filho [21]	Brasil	2019	Science Direct
Kalboussi [22]	Tunisia	2013	Springer
Kessentini [23]	USA	2018	ACM
Kessentini [24]	USA	2020	ACM
Marculescu [25]	Suécia	2012	Springer
Marculescu [26]	Suécia	2013	IEEE
Marculescu [27]	Suécia	2013	IEEE
Mkaouer [28]	Tunísia	2013	IEEE
Ramírez [29]	Espanha	2018	Elsevier
Ramírez [4]	Espanha	2019	IEEE
Ramírez [30]	Espanha	2021	IEEE
Rebai [31]	Michigan	2020	IEEE
Saraiva [32]	Brasil	2017	Springer
Simons [33]	Inglaterra	2013	ACM
Simons [34]	Inglaterra	2009	IEEE
Bechikh [35]	Tunisia	2015	Elsevier
Tonella [36]	Itália	2013	Elsevier
Wang [37]	China	2018	IEEE

Tabela 3: Artigos selecionados na etapa de extração de dados

3. Resultados

A extração de dados se iniciou pelos estudos secundários identificados [4] [38]. Estes trabalhos foram analisados inicialmente com o intuito de comparar os resultados desta revisão sistemática. Foi constatado que mesmo possuindo *strings* de busca diferentes, a maioria dos trabalhos obtidos nesta revisão, estão contidos estudos secundários existentes. No contexto de modelagem de software, foi encontrado o trabalho de Mkaouer et al. [28], que propõe a utilização de técnicas de otimização multiobjetivo baseado em preferências do DM, tais como qualidade, corretude, entre outras.

Em Ramírez et al. [39] foi apresentada uma abordagem no contexto de otimização multiobjetivo interativa para arquiteturas de software. Os autores do trabalho afirmam que durante a especificação de software, arquitetos de software usualmente precisam de soluções para atender determinados critérios de qualidade, que podem ser expressos por meio de métricas de software. Além disso, algumas vezes é necessário incluir a experiência e conhecimento do DM, que nem sempre podem ser modelados matematicamente. No trabalho em questão, as preferências foram implementadas em uma função de aptidão (*fitness*)² e o algoritmo NSGA-II foi adaptado a fim de incluir as preferências do DM. Por meio de experimentos com DMs, os autores observam que guiar a seleção de soluções por meio da região de interesse (Region of Interest - ROI) pode ser efetivo. O número de interações é definido pelo DM *a priori* e em tais interações é possível bloquear classes e interfaces que estejam de acordo com suas preferências, para mantê-las imutáveis durante todo o processo de otimização.

No trabalho de Birtolo et al. [13] são utilizados algoritmos genéticos iterativos para seleção de paletas de cores para interfaces de usuário. Os autores não deixam explícito qual algoritmo foi utilizado. A interação com o DM ocorre a cada N interações, definido *a priori*. Em cada interação o DM fornece uma análise subjetiva das paletas de cores.

Na literatura não foram encontradas somente aplicações de algoritmos genéticos iterativos. Há trabalhos que suportam as preferências do DM no algoritmo de otimização por colônias de formigas [41] [16]. do Nascimento Ferreira et al. [16] propõem a aplicação de um modelo interativo, obtendo 12% de perda no resultado da função de aptidão quando incluso

²Função matemática que indica a qualidade do indivíduo obtido por meio de um algoritmo de otimização [40]

as preferências do DM, porém com 80% das preferências satisfeitas. Simons and Smith [41] propõem a utilização de uma técnica conhecida como “*anti pheromone*”. Esta técnica tem como objetivo melhorar a performance do algoritmo, reduzindo o número de avaliações requeridas em até 20% e mantendo a qualidade das soluções. Ambos os trabalhos utilizam o *ranking* das soluções considerando análises qualitativas das soluções.

Nos trabalhos selecionados para extração de dados, são apresentadas estratégias, seja de forma explícita ou implícita para o tratamento da fadiga humana, decorrente do número excessivo de interações com o DM. Estas estratégias variam desde trabalhar com pequenas populações e poucas iterações, até permitir o DM decidir o número de interações e quando as mesmas ocorrerão [38]. Algumas das estratégias citadas incluem o uso de algoritmos de AM para prevenção da fadiga humana. Araújo et al. [10] propõem a utilização dos algoritmos *Least Median Square* (LMS) e *Multilayer Perceptron* (MLP), a fim de substituir o tomador de decisão após determinada quantidade de interações definida previamente pelo DM. Tais algoritmos, são utilizados para simulação do perfil de avaliação do DM, pontuando as soluções apresentadas, levando em consideração o que foi aprendido nas interações durante o processo de otimização.

No contexto de geração de testes de software foram encontrados os trabalhos de Marculescu et al. [25], Marculescu et al. [27], Marculescu et al. [26] e Kalboussi et al. [22]. Nestes trabalhos são realizados experimentos controlados na indústria utilizando um algoritmo de evolução diferencial iterativo. São utilizadas métricas específicas para o contexto dos trabalhos, tais como: consumo total de energia (TPC), número de obstáculos encontrados (NEO), quantidade de gasto coletados em determinado tempo (ACW), número de vezes que o agente coleta os gastos mais próximos (NDNW, NDND e NRSD).

Os trabalhos de Kalboussi et al. [22] e Mkaouer et al. [28] utilizam o algoritmo r-NSGA-II (*reference solution-based NSGA-II*). O uso de algoritmos de otimização iterativos para priorização de requisitos foi observado no trabalho de Tonella et al. [36]. Os resultados demonstram que tal algoritmo superou o algoritmo *Incomplete Analytic Hierarchy Process* (IAHP) em termos de efetividade, eficiência e robustez.

Foram também encontrados trabalhos no contexto de seleção de recursos em Linhas de Produto de Software (LPS) [15] e projeto de classes orientado a objetos [3]. Trabalhos com o intuito de definir estratégias de atribuição de pesos nas soluções e busca guiada pela preferência em otimização baseada em conjunto, também foram observadas nesta revisão sistemática [16] [38].

O trabalho de Ramirez et al. [29] evidencia os benefícios da otimização de muitos objetivos em SBSE. Também são apresentados alguns estudos que estão sendo realizados aplicando novos algoritmos e novas abordagens para solucionar problemas de otimização com muitos objetivos de forma mais efetiva. A autora propõe em outro trabalho [39] a combinação de critérios de avaliação qualitativos e quantitativos.

Os artigos de Araújo et al. [9] [10] e Saraiva et al. [32] apresentam uma formulação de objetivo único interativo para o problema do próximo *release*, no qual é proposto um conjunto de métricas que podem favorecer os estudos envolvendo otimização interativa, com a análise do comportamento da arquitetura ao incorporar as avaliações subjetivas do DM. Com o objetivo de evitar o problema da fadiga no DM apresentando um grande número de interações, um modelo de aprendizagem foi proposto para aprender o perfil de avaliação do DM substituindo-o após um determinado número de interações.

Utilizando a abordagem Hiper-Heurística em testes de LPS, Jakubovski Filho et al. [21] implementam as preferências do DM por meio do algoritmo r-NSGAI em que é fornecido um ponto de referência e suas preferências são incorporadas durante o processo de otimização. Esse estudo relata que foram obtidos resultados satisfatórios, reduzindo o número de soluções que não são interessantes do ponto de vista do DM.

Bindewald et al. [12] propõem alterações no algoritmo NSGA-II para permitir a interação do DM durante o processo de otimização de Arquiteturas de Linhas de Produto de Software. O número de interações assim como o intervalo e até mesmo em qual geração do algoritmo de otimização ocorre a primeira interação são especificados *a priori*. A interação é dada por meio de atribuição de notas na Escala de Likert [42] (entre 0 a 5) para as soluções otimizadas, indicando a qualidade delas do ponto de vista do DM.

O trabalho de Bindewald et al. [12] faz menção ao trabalho de Freire et al. [43], que utiliza algoritmos de agrupamento para categorizar as soluções apresentadas durante o processo de otimização. Esta categorização tem como objetivo reduzir a quantidade de soluções a serem analisadas pelo DM, prevenindo assim o problema da fadiga humana decorrente do número excessivo de soluções analisadas.

4. Síntese dos Estudos

Na Tabela 4 são apresentados os artigos por tipo de veículo de publicação. É possível observar que o simpósio SSBSE (*Symposium on Search Based*

Software Engineering) possui o maior número de publicações no contexto de otimização interativa. Além disso, um total de 24 autores principais foram identificados nos trabalhos selecionados na etapa de extração de dados. A cooperatividade no desenvolvimento de trabalhos pode ser observada, pois

Veículo de Publicação	Trabalhos
Advances in Computers	[35]
Int. Conf. on Systems, Man and Cybernetics	[34]
World Congress on Nature and Biologically Inspired Computing, NABIC	[13]
Int. Conf. on Machine Learning and Applications	[27]
Int. Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE)	[33] [28]
Int. Conf. on Automated Software Engineering (ASE)	[6]
IEEE Congress on Evolutionary Computation (CEC)	[30]
Int. Conf. Proceeding Series	[15]
Applied Soft Computing Journal	[16]
Automated Software Engineering	[10]
IEEE Transactions on Software Engineering	[4] [31]
Information and Software Technology	[17] [36]
Information Sciences	[29]
Journal of Systems and Software	[21] [20]
Journal of the Brazilian Computer Society	[32]
Int. Symposium on Search Based Software Engineering	[14] [9] [22]
	[25]
Int. Conf. on Model Driven Engineering Languages and Systems	[24]
Asia-Pacific Software Engineering Conf., APSEC	[26]
Brazilian Symposium on Software Components, Architectures, and Reuse	[12]
Int. Conf. on Model Driven Engineering Languages and Systems	[23]
Brazilian Symposium on Software Engineering	[19]
Companion Publication of the 2015 Annual Conf. on Genetic and Evolutionary Computation	[7]
Brazilian Symposium on Software Engineering	[18] [11]

Tabela 4: Artigos por tipo de veículo de publicação

aproximadamente 100% dos trabalhos possuem autores e coautores.

Resposta à QP: A Tabela 5 mostra as respostas para a QP, sendo que a primeira coluna apresenta o problema de ES e na segunda são apresentados os trabalhos relacionados.

Problema de ES	Trabalhos
Digrama de classes	[24] [24]
Seleção de cores em GUIs	[13]
Projeto conceitual de ES	[34]
Seleção de recursos em LPS	[15]
Próxima <i>release</i>	[10] [16] [14] [9] [32]
Nenhum	[33] [29] [17] [7] [20] [35]
Arquitetura de Linha de Produto de Software (PLA)	[11] [12] [19]
Refatoração	[31] [6]
Priorização de requisitos	[36]
Arquitetura de Software	[39]
Modelagem de Software	[28]
Teste de Software	[26] [25] [27] [30] [22] [21]
Teste de Variabilidade de LPS	[24]

Tabela 5: Problemas resolvidos com otimização interativa

Foi possível observar que há trabalhos que buscam integrar as preferências do DM durante o processo de otimização de diagramas de classes e até mesmo em refatoração de software. Há vários trabalhos que não especificam o contexto por serem estudos secundários ou *surveys*. A principal lacuna que foi observada é a falta de frameworks para otimização interativa que poderiam ser utilizados no desenvolvimento destes trabalhos.

Resposta à QS1: A Figura 4 apresenta as estratégias que os trabalhos utilizam para incluir as preferências do DM durante o processo de otimização. Conforme é possível observar, 38.7% dos trabalhos não especificam qual estratégia foi utilizada e 16.1% por serem estudos secundários ou *surveys* não propõem nenhuma estratégia. A estratégia que é mais utilizada nos trabalhos (25.8%) é a atribuição de notas às soluções otimizadas, indicando o nível de qualidade destas soluções do ponto de vista do DM. Os demais trabalhos apresentam estratégias diversas tais como indicar se uma solução é boa ou não, se deve ser incluída ou não nas próximas gerações, seleção de áreas de in-

interesse utilizando o algoritmo r-NSGAII, ordenação das soluções, atribuição de pesos às funções objetivo, todas de acordo com as preferências do DM.

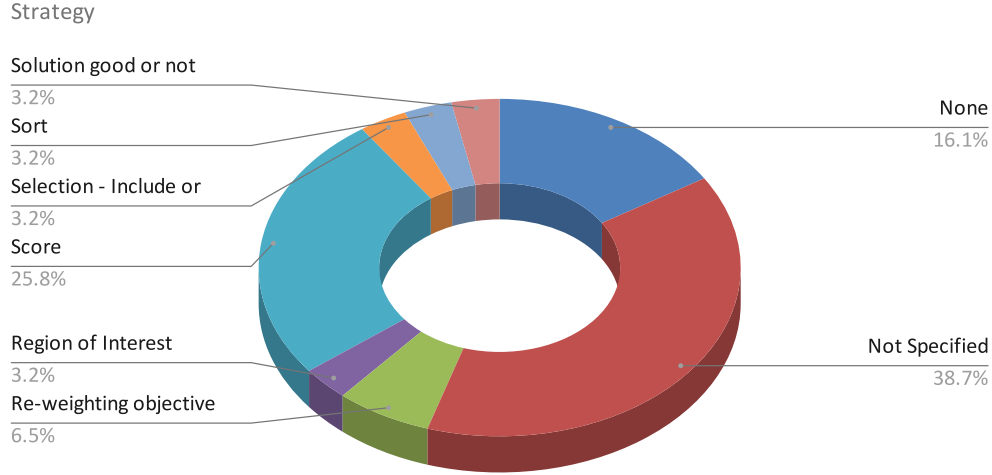


Figura 4: Estratégia interativa

Resposta à QS2: De acordo com a Figura 5, foi possível verificar que o algoritmo NSGA-II é o mais utilizado nos trabalhos. Isto se deve ao fato de que em grande parte dos trabalhos, são necessárias aproximadamente 3 funções objetivo para calcular a qualidade das soluções otimizadas. Problemas que envolvem mais de 3 funções objetivo, são comumente resolvidos utilizando o algoritmo IBEA, por possuir melhor desempenho nestes cenários (muitos objetivos). Em trabalhos de natureza mono-objetiva (com uma única função objetivo), algoritmos genéticos, inclusive interativos, são utilizados para otimização. Também é possível notar que alguns trabalhos utilizam o algoritmo r-NSGA-II por ser exclusivamente proposto para problemas de otimização interativa.

Resposta à QS3: Na Figura 6 são descritos os algoritmos de AM que são utilizados pelos trabalhos obtidos nesta revisão sistemática. É possível notar que a maioria não especifica a utilização de algum algoritmo de AM. Contudo, nos poucos poucos trabalhos que utilizam algum algoritmo de AM, tal algoritmo é aplicado para prevenir o problema da fadiga humana de forma promissora. A fadiga pode ocorrer, seja pela excessiva quantidade de in-

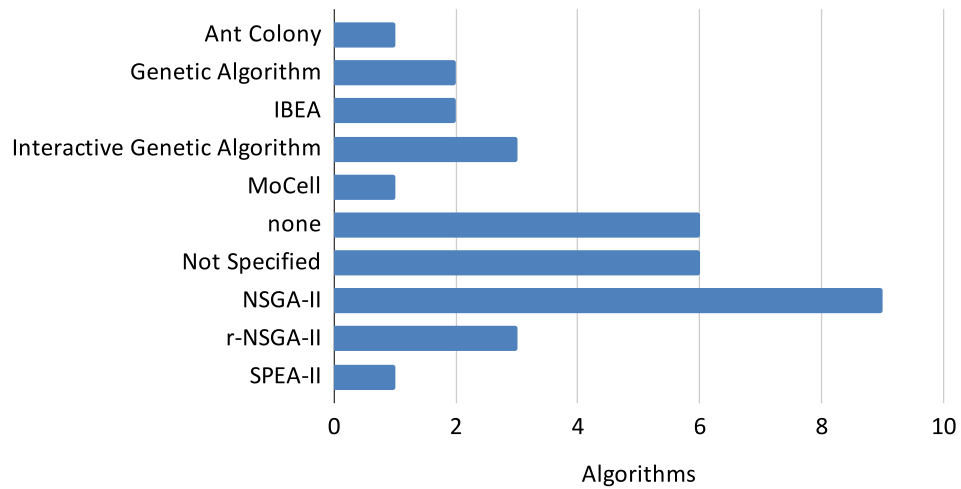


Figura 5: Algoritmos de Otimização utilizados

terações com o DM, ou pelo excessivo número de soluções avaliadas durante o processo de otimização. Desta forma, algoritmos de agrupamento têm sido aplicados para categorizar soluções otimizadas a fim de reduzir a quantidade de avaliações. Além disso, algoritmos de AM também são aplicados para aprender o perfil de avaliação do DM e substituí-lo a partir de um número de gerações especificado *a priori*.

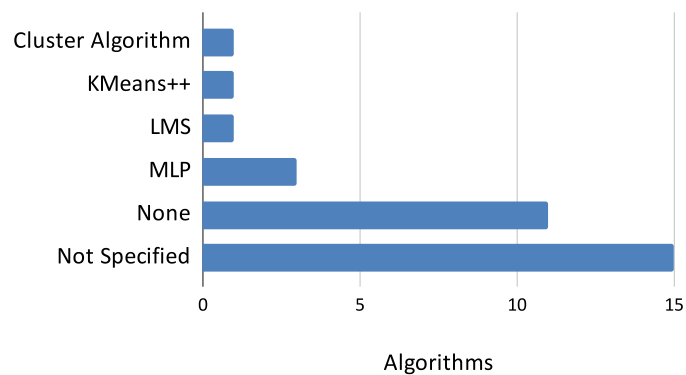


Figura 6: Algoritmos de Aprendizagem utilizados

Resposta à QS4: Em trabalhos como Bindewald et al. [12] e Freire [5] a instância de entrada é um objeto complexo que demanda grande esforço cognitivo de análise. Isto impacta diretamente na estratégia interativa, visto que analisar muitas PLAs em um curto período de tempo pode ocasionar o problema da fadiga humana. Desta forma, o tipo da instância de entrada é algo essencial na definição de abordagens que utilizam otimização interativa. Conforme é possível observar na Figura 7, grande parte dos trabalhos não especificam qual a instância de entrada. Porém, é possível observar uma grande variedade de tipos de instância de entrada, que variam desde instâncias binárias até objetos.

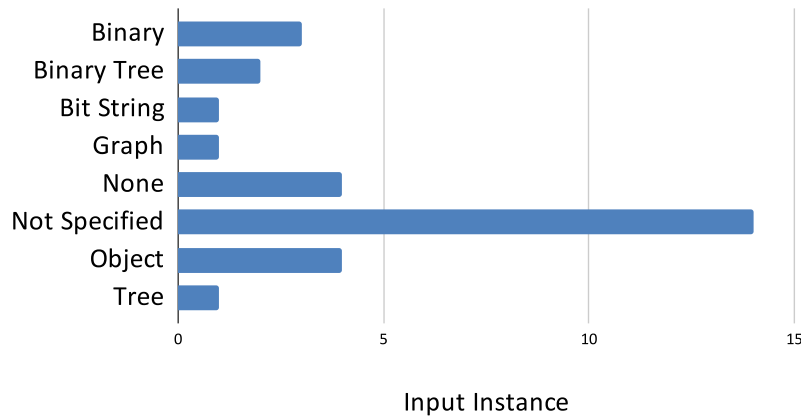


Figura 7: Tipo da instância de entrada

A quantidade de citações por trabalho é apresentada na Tabela 4. Os trabalhos mais citados são os de Tonella et al. [36] e Bechikh et al. [35]. As informações relacionadas a citações foram verificadas no veículo de publicação e na ferramenta de busca Google Scholar.

5. Ameaça a Validade

Esta seção aborda sobre os fatores que podem ameaçar a validade deste trabalho. As ameaças podem acontecer tanto na etapa de seleção dos estudos, como no processo de extração de dados.

5.1. Validade da Seleção dos Estudos

Em relação à seleção de estudos, foram criadas strings de busca para a realização da pesquisa em diferentes bases de dados. Primeiramente, foi

Trabalho (1º autor)	Citações
Alizadeh [6]	26
Aljawawdeh [7]	12
Araújo [9]	12
Araújo [10]	40
Bindewald [11]	2
Bindewald [12]	1
Birtolo [13]	13
Dantas [14]	13
El Yamany [15]	15
Ferreira [16]	23
Ferreira [17]	26
Filho [18]	1
Freire [19]	3
Ganesh [20]	14
Jakubovski Filho [21]	9
Kalboussi [22]	47
Kessentini [23]	1
Kessentini [24]	13
Marculescu [25]	7
Marculescu [26]	22
Marculescu [27]	13
Mkaouer [28]	16
Ramírez [29]	22
Ramírez [4]	1
Ramírez [30]	40
Rebai [31]	1
Saraiva [32]	5
Simons [33]	3
Simons [34]	11
Bechikh [35]	95
Tonella [36]	123

Tabela 6: Total de citações por autor

realizada uma busca piloto para validação dessas sequências de pesquisa. Além disso, este estudo restringiu a busca nas bases de dados ACM, IEEE, Scopus e Science Direct. Desta forma, algum artigo publicado em outra base

pode não ter sido incluído nesse processo. Contudo, uma busca manual foi realizada considerando referências relevantes ao tema abordado.

Um resultado de 1353 publicações retornadas nesta etapa sugere um nível de confiança ao considerar a amplitude da pesquisa realizada. Também foram criados critérios de inclusão e exclusão considerando tópicos importantes para identificar os artigos relevantes para a realização deste trabalho. Após a realização dessa etapa, 33 estudos foram separados para extração dos dados.

5.2. Validade dos Dados

No processo de extração de dados, todos os artigos selecionados foram analisados na íntegra. Para auxiliar neste processo, a ferramenta StArt foi utilizada para organizar as informações sobre: ano de publicação, veículo de publicação, tipo de veículo de publicação, algoritmos genéticos interativos, otimização interativa, preferência do tomador de decisão e engenharia de software baseada em pesquisa. Além do mais, foram considerados apenas artigos publicados em inglês.

5.3. Validade da Pesquisa

Com o intuito de garantir a validade desta revisão sistemática, foi realizada uma pesquisa com strings que incluem termos que mantêm a generalização do estudo. A realização de uma análise na íntegra dos artigos selecionados possibilitou uma visão geral embasada dos resultados obtidos.

6. Conclusão

Diversos pesquisadores têm estudado a área de Otimização Iterativa, na busca de novas abordagens para incluir as preferências do engenheiro de software durante o processo de otimização em SBSE. Sendo assim, esta revisão sistemática buscou trabalhos que têm sido realizados em diversos contextos de engenharia de software. Foi possível verificar uma grande cooperação entre autores no desenvolvimento de trabalhos neste contexto.

Além do mais, as mais diversas técnicas têm sido utilizadas para incluir as preferências do DM durante o processo de otimização. Algoritmos de AM também são empregados para prevenir a fadiga humana decorrente do número excessivo de interações e até mesmo na redução do número de soluções a serem avaliadas pelo DM nestas interações. Este trabalho servirá de base para o desenvolvimento de trabalhos futuros no contexto de iSBSE, direcionando os estudos no preenchimento das lacunas observadas na revisão sistemática.

Referências

- [1] I. Sommerville, R. Arakaki, S. S. S. Melnikoff, Engenharia de software, Pearson Prentice Hall, 2008.
- [2] M. Harman, B. F. Jones, Search-based software engineering, *Information and Software Technology* 43 (2001) 833–839.
- [3] C. L. Simons, I. C. Parmee, R. Gwynllyw, Interactive, evolutionary search in upstream object-oriented class design, *IEEE Transactions on Software Engineering* 36 (2010) 798–816.
- [4] A. Ramirez, J. R. Romero, C. Simons, A systematic review of interaction in search-based software engineering, *IEEE Transactions on Software Engineering* (2018).
- [5] W. M. Freire, Otimização interativa de arquitetura de linha de produto de software utilizando congelamento de elementos arquiteturais, Master’s thesis, Universidade Estadual de Maringá, 2020.
- [6] V. Alizadeh, M. Kessentini, Reducing interactive refactoring effort via clustering-based multi-objective search, in: 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2018, pp. 464–474.
- [7] H. J. Aljawawdeh, C. L. Simons, M. Odeh, Metaheuristic design pattern: Preference, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 1257–1260.
- [8] S. Ali, P. Arcaini, D. Pradhan, S. A. Safdar, T. Yue, Quality indicators in search-based software engineering: an empirical evaluation, *ACM Transactions on Software Engineering and Methodology (TOSEM)* 29 (2020) 1–29.
- [9] A. A. Araújo, M. Paixão, Machine learning for user modeling in an interactive genetic algorithm for the next release problem, in: International Symposium on Search Based Software Engineering, Springer, 2014, pp. 228–233.

- [10] A. A. Araújo, M. Paixao, I. Yeltsin, A. Dantas, J. Souza, An architecture based on interactive optimization and machine learning applied to the next release problem, *Automated Software Engineering* 24 (2017) 623–671.
- [11] C. V. Bindewald, W. M. Freire, A. M. M. Amaral, T. E. Colanzi, Towards the support of user preferences in search-based product line architecture design: an exploratory study, in: *Proceedings of the XX-XIII Brazilian Symposium on Software Engineering*, 2019, pp. 387–396.
- [12] C. V. Bindewald, W. M. Freire, A. M. M. Amaral, T. E. Colanzi, Supporting user preferences in search-based product line architecture design using machine learning, in: *Proceedings of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse*, 2020, pp. 11–20.
- [13] C. Birtolo, P. Pagano, L. Troiano, Evolving colors in user interfaces by interactive genetic algorithm, in: *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009, pp. 349–355.
- [14] A. Dantas, I. Yeltsin, A. A. Araújo, J. Souza, Interactive software release planning with preferences base, in: *International Symposium on Search Based Software Engineering*, Springer, 2015, pp. 341–346.
- [15] A. E. El Yamany, M. Shaheen, A. S. Sayyad, Opti-select: An interactive tool for user-in-the-loop feature selection in software product lines, in: *Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools-Volume 2*, 2014, pp. 126–129.
- [16] T. do Nascimento Ferreira, A. A. Araújo, A. D. B. Neto, J. T. de Souza, Incorporating user preferences in ant colony optimization for the next release problem, *Applied Soft Computing* 49 (2016) 1283–1296.
- [17] T. N. Ferreira, S. R. Vergilio, J. T. de Souza, Incorporating user preferences in search-based software engineering: A systematic mapping study, *Information and Software Technology* 90 (2017) 55–69.
- [18] H. L. J. Filho, T. N. Ferreira, S. R. Vergilio, Multiple objective test set selection for software product line testing: evaluating different

- preference-based algorithms, in: Proceedings of the XXXII Brazilian Symposium on Software Engineering, 2018, pp. 162–171.
- [19] W. M. Freire, M. Massago, A. C. Zavadski, A. M. Malachini, M. Amaral, T. E. Colanzi, Opla-tool v2. 0: a tool for product line architecture design optimization, in: Proceedings of the 34th Brazilian Symposium on Software Engineering, 2020, pp. 818–823.
 - [20] G. R. Santhanam, Qualitative optimization in software engineering: A short survey, *Journal of Systems and Software* 111 (2016) 149–156.
 - [21] H. L. Jakubovski Filho, T. N. Ferreira, S. R. Vergilio, Preference based multi-objective algorithms applied to the variability testing of software product lines, *Journal of Systems and Software* 151 (2019) 194–209.
 - [22] S. Kalboussi, S. Bechikh, M. Kessentini, L. B. Said, Preference-based many-objective evolutionary testing generates harder test cases for autonomous agents, in: International Symposium on Search Based Software Engineering, Springer, 2013, pp. 245–250.
 - [23] W. Kessentini, M. Wimmer, H. Sahraoui, Integrating the designer in-the-loop for metamodel/model co-evolution via interactive computational search, in: Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, 2018, pp. 101–111.
 - [24] W. Kessentini, V. Alizadeh, Interactive metamodel/model co-evolution using unsupervised learning and multi-objective search, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, 2020, pp. 68–78.
 - [25] B. Marculescu, R. Feldt, R. Torkar, A concept for an interactive search-based software testing system, in: International Symposium on Search Based Software Engineering, Springer, 2012, pp. 273–278.
 - [26] B. Marculescu, R. Feldt, R. Torkar, Practitioner-oriented visualization in an interactive search-based software test creation tool, in: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), volume 2, IEEE, 2013, pp. 87–92.

- [27] B. Marculescu, R. Feldt, R. Torkar, Objective re-weighting to guide an interactive search based software testing system, in: 2013 12th International Conference on Machine Learning and Applications, volume 2, IEEE, 2013, pp. 102–107.
- [28] M. W. Mkaouer, M. Kessentini, S. Bechikh, D. R. Tauritz, Preference-based multi-objective software modelling, in: 2013 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE), IEEE, 2013, pp. 61–66.
- [29] A. Ramirez, J. R. Romero, S. Ventura, Interactive multi-objective evolutionary optimization of software architectures, *Information Sciences* 463 (2018) 92–109.
- [30] A. Ramírez, P. Delgado-Pérez, K. J. Valle-Gómez, I. Medina-Bulo, J. R. Romero, Interactivity in the generation of test cases with evolutionary computation, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 2395–2402.
- [31] S. Rebai, V. Alizadeh, M. Kessentini, H. Fehri, R. Kazman, Enabling decision and objective space exploration for interactive multi-objective refactoring, *IEEE Transactions on Software Engineering* (2020).
- [32] R. Saraiva, A. A. Araújo, A. Dantas, I. Yeltsin, J. Souza, Incorporating decision maker’s preferences in a multi-objective approach for the software release planning, *Journal of the Brazilian Computer Society* 23 (2017) 1–19.
- [33] C. L. Simons, Whither (away) software engineers in sbse?, in: 2013 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE), IEEE, 2013, pp. 49–50.
- [34] C. L. Simons, I. C. Parmee, An empirical investigation of search-based computational support for conceptual software engineering design, in: 2009 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2009, pp. 2503–2508.
- [35] S. Bechikh, M. Kessentini, L. B. Said, K. Ghédira, Preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art, in: *Advances in Computers*, volume 98, Elsevier, 2015, pp. 141–207.

- [36] P. Tonella, A. Susi, F. Palma, Interactive requirements prioritization using a genetic algorithm, *Information and software technology* 55 (2013) 173–187.
- [37] S. Wang, S. Ali, T. Yue, M. Liaaen, Integrating weight assignment strategies with nsga-ii for supporting user preference multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 22 (2017) 378–393.
- [38] T. N. Ferreira, S. R. Vergilio, J. T. de Souza, Incorporating user preferences in search-based software engineering: A systematic mapping study, *Information and Software Technology* 90 (2017) 55 – 69. doi:<https://doi.org/10.1016/j.infsof.2017.05.003>.
- [39] A. Ramírez, J. R. Romero, S. Ventura, Interactive multi-objective evolutionary optimization of software architectures, *Information Sciences* (2018).
- [40] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: nsga-ii, *IEEE Transactions on Evolutionary Computation* 6 (2002) 182–197.
- [41] C. Simons, J. Smith, Exploiting antipheromone in ant colony optimisation for interactive search-based software design and refactoring, in: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, ACM, 2016, pp. 143–144.
- [42] R. Likert, A technique for the measurement of attitudes., *Archives of psychology* (1932).
- [43] W. M. Freire, C. V. Bindewald, A. M. M. Amaral, T. E. Colanzi, Supporting decision makers in search-based product line architecture design using clustering, in: *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, IEEE, 2019, pp. 139–148.