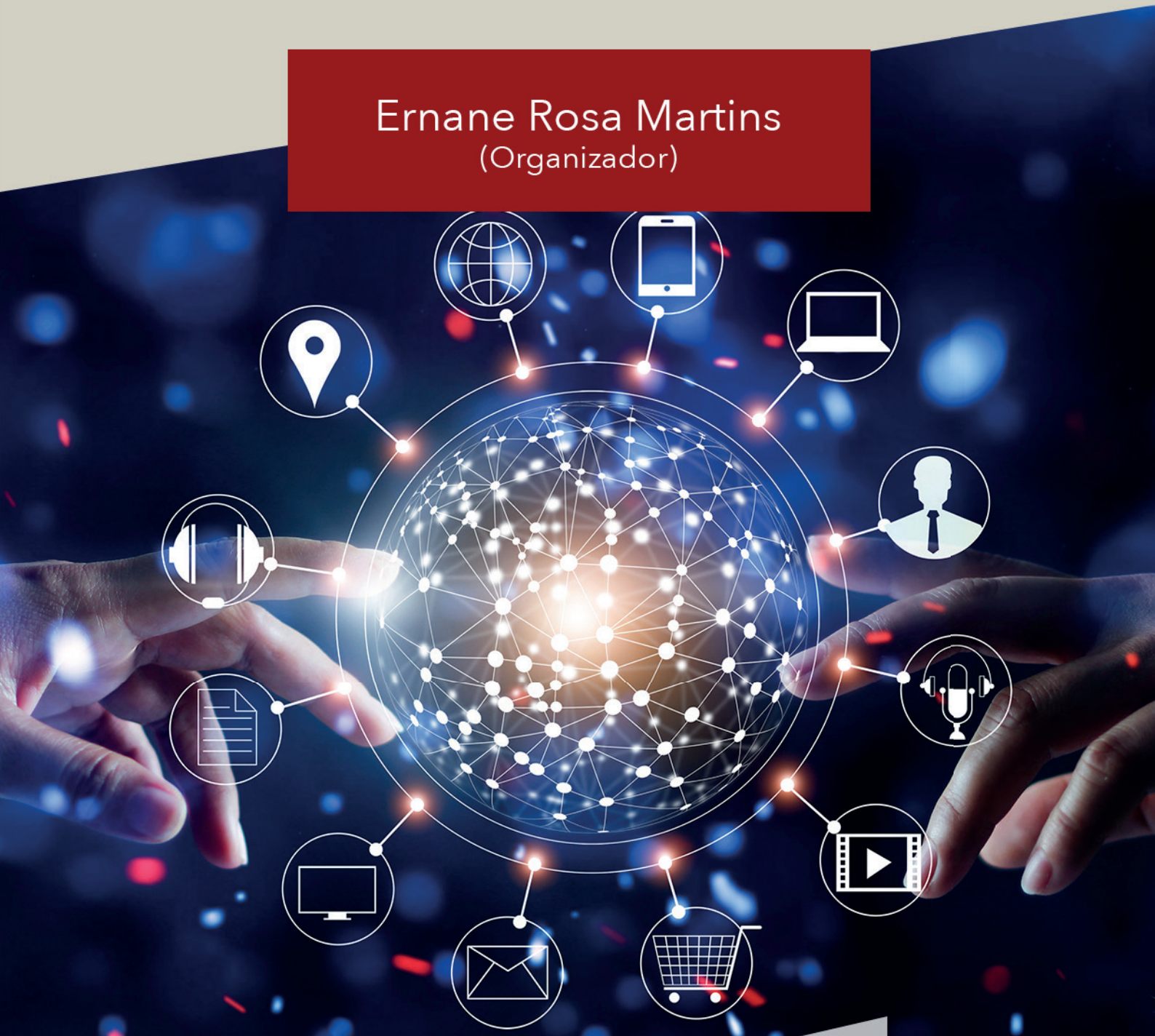
Princípios e Aplicações da Computação no Brasil 2

Ernane Rosa Martins
(Organizador)



Atena Editora
Ano 2019

Ernane Rosa Martins
(Organizador)

Princípios e Aplicações da Computação no Brasil
2

Atena Editora
2019

2019 *by Atena Editora*
Copyright © da Atena Editora
Editora Chefe: Profª Drª Antonella Carvalho de Oliveira
Diagramação e Edição de Arte: Geraldo Alves e Natália Sandrini
Revisão: Os autores

Dados Internacionais de Catalogação na Publicação (CIP)
(eDOC BRASIL, Belo Horizonte/MG)

P957 Princípios e aplicações da computação no brasil 2 [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa (PR): Atena Editora, 2019. – (Princípios e aplicações da computação no brasil; v. 2)

Formato: PDF
Requisito de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-85-7247-048-3
DOI 10.22533/at.ed.483191601

1. Computação. 2. Informática. 3. Programação de computador. I. Martins, Ernane Rosa. II. Título. III. Série.

CDD 004

Elaborado por Maurício Amormino Júnior – CRB6/2422

APRESENTAÇÃO

O volume 2 desta obra aborda mais 16 capítulos sobre o panorama atual da computação no Brasil. Tendo como alguns dos assuntos abordados nos capítulos: ensino de raciocínio lógico, desenvolvimento de sistema computacional, micromobilidade em redes sem fio, usabilidade e acessibilidade de sistemas, qualidade da informação, tecnologias de análise de aprendizagem, redes neurais artificiais, analise de vibração, algoritmos evolucionários, sistemas inteligentes e acessibilidade móvel.

Deste modo, esta obra reúne debates e análises acerca de questões relevantes, tais como: Como está o estado da arte da análise de aprendizagem preditiva, nova proposta de um framework para previsão de desempenhos em programação e quais os caminhos para avançar nessas pesquisas? É possível realizar uma modelagem computacional, analisando os parâmetros espaciais relevantes na tomada de decisão, utilizando técnicas de redes neurais artificiais? Quais são os principais desafios, no cenário nacional, a fim de estabelecer e manter um Sistema de Gestão de Segurança da Informação? Uma proposta de um agente testador que realiza busca local no espaço de estados de casos de teste orientado por utilidade e que utiliza os algoritmos evolucionários multiobjetivos, NSGAII, SPEA2, PAES e MOCell pode identificar quais deles são mais eficientes na geração de casos de testes para agentes racionais? Como realizar uma pesquisa científica que identifique os requisitos desejáveis para desenvolver uma aplicação móvel touch screen, que vise auxiliar a alfabetização de deficientes visuais?

Nesse sentido, este material tem grande relevância por constituir-se numa coletânea de referência para pesquisas e estudos da computação, tendo como objetivo reunir trabalhos acadêmicos que permitam contribuir com análises e discussões sobre assuntos pertinentes à área. Os organizadores da Atena Editora, agradecem especialmente aos autores dos diversos capítulos apresentados, parabenizam a dedicação e esforço de cada um, os quais viabilizaram a construção dessa obra no viés da temática apresentada. Por fim, desejamos aos leitores que esta obra, seja de extrema importância para todos que vierem a utilizá-la.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 1 1

ENSINO DE RACIOCÍNIO LÓGICO E COMPUTAÇÃO PARA CRIANÇAS: EXPERIÊNCIAS, DESAFIOS E POSSIBILIDADES (XXXVII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO | 25O WEI - *WORKSHOP* SOBRE EDUCAÇÃO EM COMPUTAÇÃO)

Thâmillys Marques de Oliveira
Willmara Marques Monteiro
Fábio Cristiano Souza Oliveira
Danielle Juliana Silva Martins
Alessandra da Silva Luengo Latorre

DOI 10.22533/at.ed.4831916011

CAPÍTULO 2 12

DESENVOLVIMENTO DE SISTEMA COMPUTACIONAL PARA AQUISIÇÃO E ANÁLISE DE DADOS AMBIENTAIS REMOTAMENTE.

Jucivaldo Araujo Ferreira Junior
Rardiles Branches Ferreira
Rodrigo Da Silva
Julio Tota da Silva
Samuel Alves de Souza

DOI 10.22533/at.ed.4831916012

CAPÍTULO 3 19

CARACTERIZAÇÃO DA MICROMOBILIDADE EM REDES SEM FIO INFRAESTRUTURADAS PELA VARIAÇÃO DA RELAÇÃO SINAL-RUÍDO

Kerlla Souza Luz Prates
Priscila América Solís Mendez
Barreto Henrique Domingues Garcia
Mylène Christine Queiroz de Farias

DOI 10.22533/at.ed.4831916013

CAPÍTULO 4 30

AVALIAÇÃO DE USABILIDADE E ACESSIBILIDADE DO SISTEMA DE GERENCIAMENTO DE REFEITÓRIOS DO IFPI – CAMPUS FLORIANO

Samuel de Araújo Fonseca
Antonio Rodrigues de Araújo Costa
Neto Carlos Eduardo Moreira Borges
Hugo Araújo Gonçalves
Paulo Miranda e Silva Sousa
Rennê Stephany Ferreira dos Santos

DOI 10.22533/at.ed.4831916014

CAPÍTULO 5 39

AVALIAÇÃO DA APREENSIBILIDADE E DA QUALIDADE DA INFORMAÇÃO EM SAÚDE COM O SOFTWARE SPINEFIND

Carine Geltrudes Webber
Asdrubal Falavigna
Caio Rodrigues da Silva
Marco Antonio Koff
Natália Lisboa

DOI 10.22533/at.ed.4831916015

CAPÍTULO 6 54

AS TECNOLOGIAS DE ANÁLISE DE APRENDIZAGEM E OS DESAFIOS DE PREVER DESEMPENHOS DE ESTUDANTES DE PROGRAMAÇÃO

Márcia Gonçalves de Oliveira

DOI 10.22533/at.ed.4831916016

CAPÍTULO 7 67

ANÁLISE E MODELAGEM DA RELAÇÃO INTERPESSOAL EM ESPORTES COLETIVOS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Tadeu Nogueira Costa de Andrade
Marcos Rodrigo Trindade Pinheiro
Menuchi Paulo Eduardo Ambrósio

DOI 10.22533/at.ed.4831916017

CAPÍTULO 8 75

ANÁLISE DOS DESAFIOS PARA ESTABELECER E MANTER SISTEMA DE GESTÃO DE SEGURANÇA DA INFORMAÇÃO NO CENÁRIO BRASILEIRO

Rodrigo Valle Fazenda
Leonardo Lemes Fagundes

DOI 10.22533/at.ed.4831916018

CAPÍTULO 9 87

ANALISE DE VIBRAÇÃO COM CONTROLE DE MEDIÇÃO UTILIZANDO O FILTROS ESTATÍSTICOS

Karla Melissa dos Santos Leandro
Iago Ferreira Lima
Werley Rafael da Silva
Marco Paulo Guimarães
Marcos Napoleão Rabelo

DOI 10.22533/at.ed.4831916019

CAPÍTULO 10 96

ANÁLISE DE REDE COLABORAÇÃO CIENTÍFICA COMO FERRAMENTA NA GESTÃO DE PROGRAMAS DE PÓS-GRADUAÇÃO

Aurelio R. Costa
Celia Ghedini Ralha

DOI 10.22533/at.ed.48319160110

CAPÍTULO 11 109

ALGORITMOS EVOLUCIONÁRIOS MULTIOBJETIVOS PARA A SELEÇÃO DE CASOS DE TESTE PARA SISTEMAS INTELIGENTES

Daniel Victor Saraiva
Francisca Raquel de Vasconcelos Silveira

DOI 10.22533/at.ed.48319160111

CAPÍTULO 12 124

ACESSIBILIDADE MÓVEL PARA ALFABETIZAÇÃO DE DEFICIENTES VISUAIS: PROPOSTA INICIAL DE UM PROTÓTIPO

Jenifer Melissa de Paula
José Valter Amaral de Freitas
Thatiane de Oliveira Rosa

DOI 10.22533/at.ed.48319160112

CAPÍTULO 13 129

AÇÃO PARA INCENTIVAR MENINAS DO ENSINO MÉDIO A CURSAR CARREIRAS TECNOLÓGICAS DA UNIVERSIDADE FEDERAL DE RIO GRANDE DO NORTE

Idalmis Milián Sardina
Cristiano Maciel
Midori Hijioka Camelo
Hortensia Sardina Miranda

DOI 10.22533/at.ed.48319160113

CAPÍTULO 14 137

A TÉCNICA OC2-RD2 COMO UMA PRÁTICA METODOLÓGICA PARA O ENSINO DE PROGRAMAÇÃO DE COMPUTADORES

Karina Buttignon
Ítalo Santiago Vega
Jonhson de Tarso Silva
Adriano Carlos Moraes Rosa

DOI 10.22533/at.ed.48319160114

CAPÍTULO 15 149

A DECADE OF SOFTWARE ENGINEERING BEST PRACTICES ADOPTION IN SMALL COMPANIES:
A QUASI-SYSTEMATIC MAPPING

Alex Juvêncio Costa
Juliana De Albuquerque Gonçalves
Saraiva Yuska Paola Costa Aguiar

DOI 10.22533/at.ed.48319160115

CAPÍTULO 16 162

INVENTORYIOT I²OT: UMA PLATAFORMA DE GERENCIAMENTO AUTOMATIZADO DE INVENTÁRIO

Jauberth Weyll Abijaude
Péricles de Lima Sobreira
Aprígio Augusto Lopes Bezerra
Fabíola Greve

DOI 10.22533/at.ed.48319160116

SOBRE O ORGANIZADOR 177

CAPÍTULO 11

ALGORITMOS EVOLUCIONÁRIOS MULTIOBJETIVOS PARA A SELEÇÃO DE CASOS DE TESTE PARA SISTEMAS INTELIGENTES

Daniel Victor Saraiva
Grupo de Engenharia de Software e Redes de Computadores (GERCOM)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE
Aracati – CE

Francisca Raquel de Vasconcelos Silveira
Grupo de Engenharia de Software e Redes de Computadores (GERCOM)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE
Tianguá - CE

RESUMO: Com o aumento da utilização de sistemas computacionais, agente racional surge como uma tecnologia promissora na solução desde problemas relativamente simples, bem como, problemas complexos. Devido sua autonomia, testes de agentes tem se tornado um grande desafio, pois esses agentes podem apresentar diferentes resultados em uma mesma entrada de testes. Com isso, este trabalho apresenta a proposta de um agente testador que realiza busca local no espaço de estados de casos de teste orientado por utilidade e utiliza os algoritmos evolucionários multiobjetivos, NSGAII, SPEA2, PAES e MOCell com o objetivo de identificar quais deles são mais eficientes na geração de casos de testes para agentes racionais.

PALAVRAS-CHAVE: agente racional, testes de agentes, agente testador, busca local, algoritmos evolucionários.

ABSTRACT: With the increase use of computer systems, rational agent emerges as a promising technology in solution from relatively simple problems, as well as complex problems. Due to its autonomy, agent testing has become a major challenge because these agents can present different results in the same test entry. This work presents the proposal of a tester that performs local search in the state space of utility-oriented test cases and uses multiobjective evolutionary algorithms, NSGAII, SPEA2, PAES and MOCell with the objective of identifying which ones are more efficient in the generation of test cases for rational agents.

KEYWORDS: rational agent, agent testing, agent tester, local search, evolutionary algorithms.

1 | INTRODUÇÃO

Os softwares estão cada vez mais complexos e não dependem, em última análise, do aumento do poder computacional. Desse modo, o objetivo central da Inteligência Artificial (IA) é procurar incorporar nos programas e sistemas, conhecimentos e capacidades

normalmente associadas ao ser humano, a fim de encontrar soluções viáveis (COSTA; SIMÕES, 2015).

Agentes racionais surgem, então, como uma tecnologia promissora. Um agente racional deve possuir atributos capazes de distingui-lo dos demais programas, tais como: operação sob controle autônomo; percepção do ambiente; persistência por tempo prolongado; adaptação às mudanças e capacidade de criação e perseguição de metas. Sua racionalidade pode ser definida quando o agente atua para alcançar o melhor resultado (RUSSEL; NORVIG, 2013).

Entretanto, a autonomia por um lado, ajuda os agentes de software a lidarem com ambientes complexos e abertos. Por outro lado, faz com que testes de agentes se tornem uma tarefa desafiadora. Em uma mesma entrada de teste, os resultados podem ser diferentes em diferentes execuções. Por esta razão, os testes de agentes requerem um procedimento que sirva para uma grande variedade de contextos de casos de testes e devem ser aplicados nos casos de testes mais exigentes (NGUYEN et al., 2012).

Visto que testar esses sistemas tornou-se uma premissa fundamental, este trabalho apresenta a proposta de um agente testador que realiza busca local baseado em população e orientado por uma função utilidade para encontrar conjuntos de casos de teste satisfatórios, ou seja, encontrar casos que levem os agentes testados a um maior índice de falhas. Dessa forma, são realizadas comparações entre quatro estratégias de busca local de algoritmos evolucionários multiobjetivos (MOEAs - *Multiobjective Evolutionary Algorithms*) utilizados pelo agente testador, primeiramente na geração inicial de um conjunto de casos de testes e em seguida, entre os conjuntos avaliados pela função utilidade associado ao desempenho dos agentes testados, a geração de um novo conjunto de casos de testes mais útil que o anterior.

Os algoritmos *Non-dominated Sorting Genetic Algorithm* (NSGA-II), *Strength Pareto Evolutionary Algorithm* (SPEA2), *Pareto Archived Evolution Strategy* (PAES) e *MultiObjective Cellular (MOCell)* são escolhidos por serem largamente utilizados e representam diferentes estratégias de evolução para lidar com problemas de otimização. Para avaliar a acurácia desses algoritmos a serem comparados, são desenvolvidos quatro tipos diferentes de programas de agentes que são testados com os casos de teste gerados por cada MOEA. Dessa forma, ao final do ciclo de testes, as comparações entre as aplicações de cada algoritmo têm o objetivo de identificar quais deles possuem os melhores desempenhos para que possam ser utilizados por um agente testador na geração de casos de teste que sejam capazes de apresentar informações relevantes sobre o desempenho dos agentes testados.

2 I FUNDAMENTAL TEÓRICA

2.1. Agentes Racionais

Agente Racional é toda entidade capaz de interagir com o ambiente guiada, em geral (mas não necessariamente) por objetivos. Um agente possui um mecanismo que permite recolher informações do ambiente (percepção), mecanismos que lhe permitem atuar sobre o ambiente (ação) e processos que definem qual a melhor ação a realizar (decisão) (COSTA; SIMÕES, 2015).

Cinco tipos básicos de programas de agentes que incorporam os princípios subjacentes a quase todos os sistemas inteligentes são enumerados em (RUSSEL; NORVIG, 2013): (i) agente reativo simples (seleciona suas ações com base na sua percepção atual, ignorando o restante do histórico de percepções); (ii) agente reativo baseado em modelo (que seleciona ações com base no seu histórico de percepções e assim reflete sobre alguns dos aspectos não observados do estado atual); (ii) agente baseado em objetivos (que possui como característica, selecionar ações com base nos objetivos que descreve situações desejáveis); (iv) agente baseado em utilidades (seleciona suas ações que maximizam a utilidade esperada dos resultados da ação) e (v) agente com aprendizagem (converte todos os agentes básicos citados anteriormente e pode melhorar o desempenho de seus componentes de modo a gerar melhores ações).

2.3. Algoritmos Evolucionários Multiobjetivos

Inicialmente, problemas de grande importância tinham suas soluções baseadas nos métodos matemáticos. Porém, a complexidade desses modelos levou os pesquisadores a concentrar seus esforços no desenvolvimento de *heurísticas* com soluções baseadas nos fenômenos biológicos, sociais ou físicos observados na natureza (PONSICH et al*.,* 2013).

O uso de algoritmos evolutivos para resolver problemas desta natureza, tem sido motivado principalmente por causa da natureza populacional que permite a geração de várias soluções em uma mesma execução. Além disso, a complexidade de alguns problemas de otimização multiobjetivos, como por exemplo, problemas com grande espaço de busca e de incertezas, podem impedir a utilização ou aplicação de técnicas tradicionais (COELLO et al., 2007). Neste contexto, existe uma grande motivação para o uso de algoritmos evolucionários multiobjetivos em vez de outras técnicas.

3 I TRABALHOS RELACIONADOS

Diferentes agentes autônomos foram testados usando diferentes técnicas ao longo dos anos. Por exemplo, metodologias de testes de softwares orientados a

objetivos (HOUHAMDI, 2011), algoritmos evolutivos (NGUYEN et al., 2012), agentes testadores (SILVEIRA et al., 2014; CARNEIRO et al., 2015) com o objetivo de verificar o comportamento dos agentes para atender as especificações e os desejos dos usuários.

Baseados nesses trabalhos, verificamos se as abordagens são capazes de atender aos seguintes critérios adotados em nosso trabalho: (i) noção de agentes racionais de Russell e Norvig (2013), (ii) utilização de casos de testes gerados de acordo com os objetivos, (iii) medida de avaliação de desempenho do agente testado, (iv) consideração dos planos que são necessários para que este agente alcance estes objetivos, (v) simulação da avaliação de desempenho das interações do agente testado com seu ambiente (histórias) , (vi) utilização de estratégias de busca local multiobjetivo orientada por utilidade para encontrar casos de teste e histórias correspondentes em que o agente não foi bem avaliado e (vii) comparações entre diferentes algoritmos para identificar qual deles apresenta os melhores resultados levando em consideração as mesmas configurações de testes.

Na proposta de teste orientada a objetivos de Houhamdi (2011), o autor especifica um processo estruturado de testes que explora a relação entre análise de objetivos e casos de teste. O autor ainda define um processo estruturado para a geração de testes nos agentes, fornecendo uma forma de derivar casos de teste a partir da análise de artefatos de requisitos orientado aos objetivos para a realização de dois níveis de teste: unitário e agente.

A abordagem evolucionária para a realização dos testes de agentes autônomos adotada por Nguyen et al. (2012), tem como objetivo estudar a eficácia dos testes evolutivos multiobjetivos, sob a observação de que o comportamento indesejável surge apenas quando agentes estão envolvidos em situações difíceis. A metodologia representa os objetivos dos *stakeholders* como função de qualidade e faz uso de algoritmos evolucionários guiados pela função de qualidade para gerar uma variedade de casos de teste exigentes para os agentes. A abordagem propõe aplicar um recrutamento dos melhores casos de testes para evoluir os agentes. Para cada agente, é dado um período experimental em que os testes com diferentes níveis de dificuldade são executados.

No trabalho de Silveira et al. (2014), os autores apresentaram uma abordagem para testar programas de agentes racionais, levando ao projetista, as informações relevantes sobre o desempenho do programa agente para melhorar a sua concepção e eficiência. O agente testador elaborado pelos autores, incorpora e processa as informações na formulação do problema de seleção e outras informações enviadas pelo projetista a fim de selecionar uma solução satisfatória com a finalidade de melhorar o desempenho do agente, se necessário. Este esquema de interação entre o testador deve ser contínuo até que o desempenho do agente ser considerado satisfatório. A Figura 1 ilustra a estrutura do agente testador.

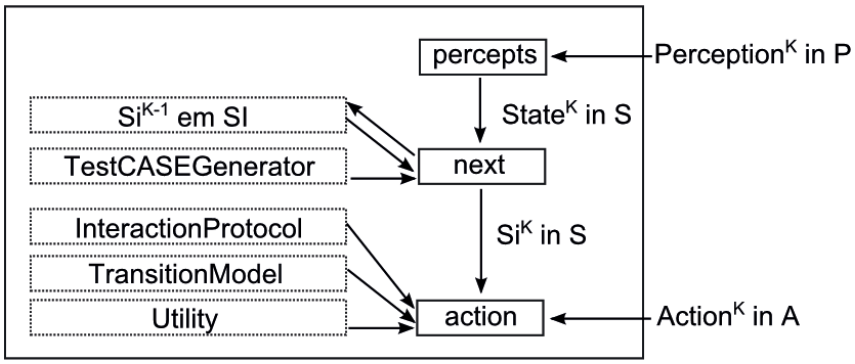


Figura 1: Estrutura do Agente Testador.

Fonte: Silveira et al. (2014).

O subsistema de percepção, *percepts*, é responsável por mapear as informações necessárias para testar o agente em uma representação computacional, *State*^{k}, útil para o processamento dos outros dois subsistemas (*next* e *action*). O subsistema *next* atualiza o estado interno em *State*^{k}, e gera um conjunto inicial de casos de testes para testar certos aspectos da estrutura interna do agente. Considerando o estado interno atualizado, a função *action* inicia um processo de busca local para encontrar uma ação satisfatória. Esta função usa informações sobre uma transição de estado para gerar novos casos de teste. Ao final dos testes, o testador envia para o projetista as informações geradas pela função *action*. Os autores ainda reforçam que, a implementação do modelo baseia-se na análise populacional, baseados em *metaheurísticas* utilizando algoritmo genético (AG).

Ainda na mesma linha de um agente testador, em Carneiro et al. (2015), os autores apresentam uma aplicação de sistemas imunológicos artificiais (AIS - *Artificial Immune Systems*), por meio do algoritmo de seleção clonal (CLONALG - *Clonal Selection Algorithm*), para o problema de otimização de seleção de casos de teste para o teste de sistemas computacionais baseados em agentes inteligentes (Figura 2).

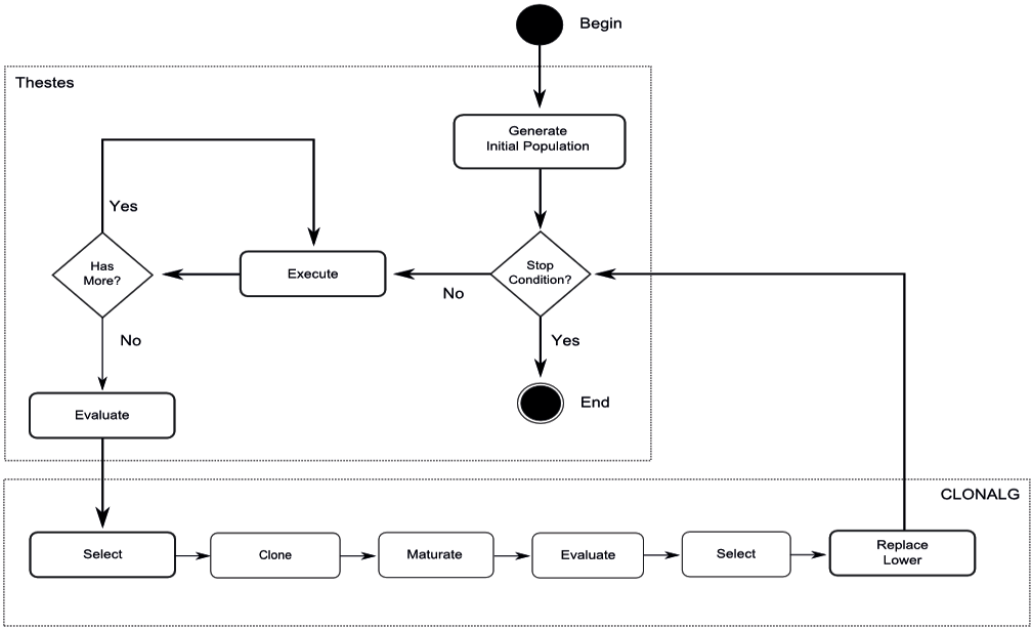


Figura 2: Fluxograma de Um Agente Testador Utilizando o Clonalg.

Fonte: Carneiro et al. (2015).

O processo começa com a geração da população inicial de casos de testes e todos os casos da geração são executados, o que significa submeter o agente testado a cada um dos cenários configurados no passo anterior. Durante a execução, as histórias são armazenadas para serem avaliadas e submetidas ao CLONALG, que seleciona os melhores casos de testes e gera clones proporcionalmente à sua avaliação. Os clones gerados são submetidos a um processo de maturação, em que cada indivíduo irá sofrer modificações com uma taxa inversamente proporcional à sua avaliação. Na fase final do CLONALG, os piores indivíduos da geração são substituídos pelos melhores clones e por novos indivíduos gerados de forma aleatória, para induzir a diversidade da população. Os autores na validação de sua proposta, realizam comparação entre as técnicas de algoritmos genéticos (AG) e algoritmos de otimização por colônia de formigas (ACO - *Ant Colony Optimization Algorithm*). Seus experimentos foram realizados com agentes inteligentes com diferentes tipos de arquitetura em tipos de ambientes diferentes.

A Tabela 1 apresenta um comparativo entre os trabalhos relacionados citados nessa seção. Podemos constatar que três das abordagens analisadas não podem atender a todos os critérios definidos e o trabalho de Carneiro *et al.* (2015) realiza comparações com somente 2 algoritmos multiobjetivos, enquanto esse trabalho realiza comparações com 4 algoritmos. Dessa forma, a necessidade da geração automatizada de testes e a aplicação de diferentes algoritmos multiobjetivos para a seleção de casos de teste, foco deste trabalho, podem gerar de forma automatizada bons casos de testes que proporcione informações relevantes sobre o desempenho insatisfatório dos componentes dos agentes testados.

	Trabalhos relacionados				
	Houhamdi (2011)	Nguyen et al. (2012)	Silveira et al. (2014)	Carneiro et al. (2015)	Nossa Abordagem
Critérios avaliados	ii, iv e vi	ii, iii, iv e vi	i, ii, iii, iv, v e vi	i, ii, iii, iv, v, vi e vii (com apenas 2 algoritmos)	i, ii, iii, iv, v, vi e vii ((com 4 algoritmos)

Tabela 1: Medida de Avaliação de Desempenho.

Fonte: Autores (2017).

4 I ABORDAGEM PROPOSTA E METODOLOGIA

4.1. Visão Geral

A abordagem fundamenta-se na noção de agentes inteligentes de Russell e Norvig (2013), Costa e Simões (2015) e na interação dos agentes abordada em Silveira et al. (2015). Assim, essa abordagem considera que além do *Projetista*, existem quatros programas agentes envolvidos na seleção e testes: (i) o programa agente a ser testado, denominado *Agente*, concebido pelo projetista; (ii) o programa ambiente

de tarefa, *Ambiente*; (iii) um programa agente para a geração e seleção de casos de testes, *Agente Testador*, foco principal desta abordagem.

A Figura 3 ilustra as interações entre os agentes. Primeiramente, o *Projetista* é o responsável pelo desenvolvimento do *Agente*, pela definição da medida de avaliação de desempenho e por outras informações necessárias para o *Agente Testador* executar o processo de teste do *Agente* em *Ambiente* e um programa agente monitorador, *Monitor.*

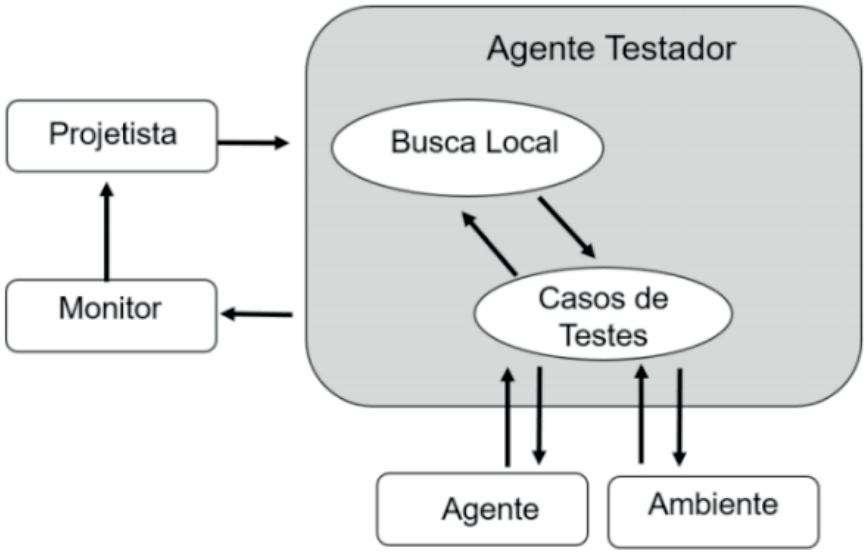


Figura 3: Visão Geral da Abordagem.

Fonte: Autores (2017).

O processo do *Agente Testador* inicia com a geração inicial de casos de testes, uma população formada por um conjunto de indivíduos (casos de testes), gerada de maneira aleatória por um algoritmo evolucionário. No próximo passo, todos os casos de testes da geração inicial são executados, o que significa submeter o *Agente* a cada um dos cenários de teste. Na avaliação das histórias, o *Agente Testador* calcula o valor de cada medida de avaliação de desempenho. As histórias avaliadas são submetidas ao um algoritmo evolucionário multiobjetivo, que selecionam os melhores casos de testes e geram novas populações ou modificam as populações existentes. Nesta fase é possível selecionar indivíduos mais capazes que possuem maior probabilidade de gerar mais descendentes, enquanto que os menos capazes poderão ainda gerar descendentes, porém em uma escala menor. O *Agente Testador*, após os ciclos de testes, envia para o agente *Monitor* os casos de teste em que *Agente* obteve o comportamento mais inadequado, um conjunto de histórias correspondentes e seus valores de utilidade contendo os objetivos não satisfeitos adequadamente pelo *Agente* e as falhas cometidas por ele em *Ambiente*.

4.2. Agente Testados

O ambiente considerado para os experimentos desses agentes possui 25 salas,

dispostas em forma de uma matriz 5 x 5, em que cada eixo (x,y) representa uma sala que pode conter ou não sujeira. Em cada iteração com o ambiente, independentemente da sala, a função do agente pode escolher uma das seguintes Ações: aspirar (*Asp*), não operar (*N-op*), ou mover-se para outra sala vizinha (*Acima, Esquerda, Direita, Abaixo*). Como o ambiente é estático, cada caso de teste é instanciado em um programa ambiente (*Ambiente*) que obedece ao modelo determinístico.

O primeiro agente desenvolvido é denominado de *RS_Parcial* do tipo reativo simples, no qual a seleção da ação a ser executada baseia-se apenas na percepção atual do ambiente parcialmente observável, de forma que o agente consegue identificar apenas a sua localização e a presença ou não de sujeira. O segundo agente desenvolvido é denominado de Agente reativo simples com alteração, *RS_Parcial_Alterado*. Esse agente possui o objetivo de verificar a sensibilidade da abordagem proposta e as possíveis falhas para que seja possível avaliar o desempenho dos algoritmos com outros agentes que realizam suas ações avaliando suas percepções e seu histórico de salas já visitadas. Esse agente, toma suas ações independentemente do estado da sala na qual o agente se encontra. O terceiro agente testado (*RS_Total*) diferencia-se dos dois primeiros agentes no seguinte aspecto: a observabilidade do ambiente. Nesse agente, é considerada a capacidade de uma visibilidade total do ambiente. Assim, o agente movimenta-se em direção à sujeira mais próxima.
O quarto e último agente testado (*RM_Parcial*) é do tipo baseado em modelo com visibilidade parcial do ambiente, no qual uma representação interna do ambiente é mantida com todas as salas já visitadas, a fim de evitar que o agente visite salas já percorridas.

Para avaliar o desempenho dos agentes testados, a função *Utilidade* avalia os conjuntos gerados, considerando valores de pesos iguais a 0,5 tanto para o atributo limpeza quanto para o atributo energia gasta em cada ação (). Além desses valores, acrescente-se a esta função um ganho por caso de teste que é igual ao número de salas sujas ao final das iterações de *Agente* com *Ambiente*. Assim, a inclusão desse valor, faz com que o *Agente Testador* busque selecionar casos de teste em que o *Agente* teve um comportamento inadequado em termos de energia e limpeza, dando privilégios aqueles casos em que o ambiente permaneceu com a maior quantidade de salas sujas ao final das interações.

4.3. Plano Experimental

Os parâmetros para utilização dos MOEAs são ajustados para que cada algoritmo execute seu processo evolutivo específico, entretanto, diante de um ambiente similar. Além disso, é considerado como critério de parada, o número máximo de 30 gerações (iterações) para cada algoritmo. Os Valores dos parâmetros utilizados pelos MOEAs são apresentados na Tabela 2.

Método de seleção torneio	NSGA-II, SPEA2; MOCell
Operador de cruzamento (*SinglePointCrossover* probabilidade = 0.9)	NSGA-II, SPEA2; MOCell
Operador e taxa de mutação (*BitFlipMutation*, probabilidade= 0.6)	NSGA-II, SPEA2; PAES; MOCell
Tamanho do Arquivo Externo (tamanho 20)	SPEA2; PAES; MOCell

Tabela 2: Valores dos Parâmetros Utilizados Pelos Moeas

Fonte: Autores (2017).

Os casos de testes iniciais, são gerados de forma aleatória pelos algoritmos multiobjetivos testados. Ao todo, são gerados 20 casos (*NCasos*), ou seja, representa o tamanho da população, e cada caso de teste é formado por 25 salas. A avaliação de desempenho final em cada caso, leva em consideração a realização de 5 simulações (*Ns*), onde cada simulação dá origem a uma história contendo 25 episódios correspondentes às iterações de *Agente* em *Ambiente* (*Nit*).

5 I APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

5.1 Experimento 1 - Resultados com RS-Parcial

A Figura 4 ilustra os resultados do primeiro experimento obtidos por *Agente Testador* considerando a função *Utilidade*, conforme os valores de pesos para os objetivos energia e limpeza ao longo de 30 gerações utilizando os quatro algoritmos multiobjetivos.

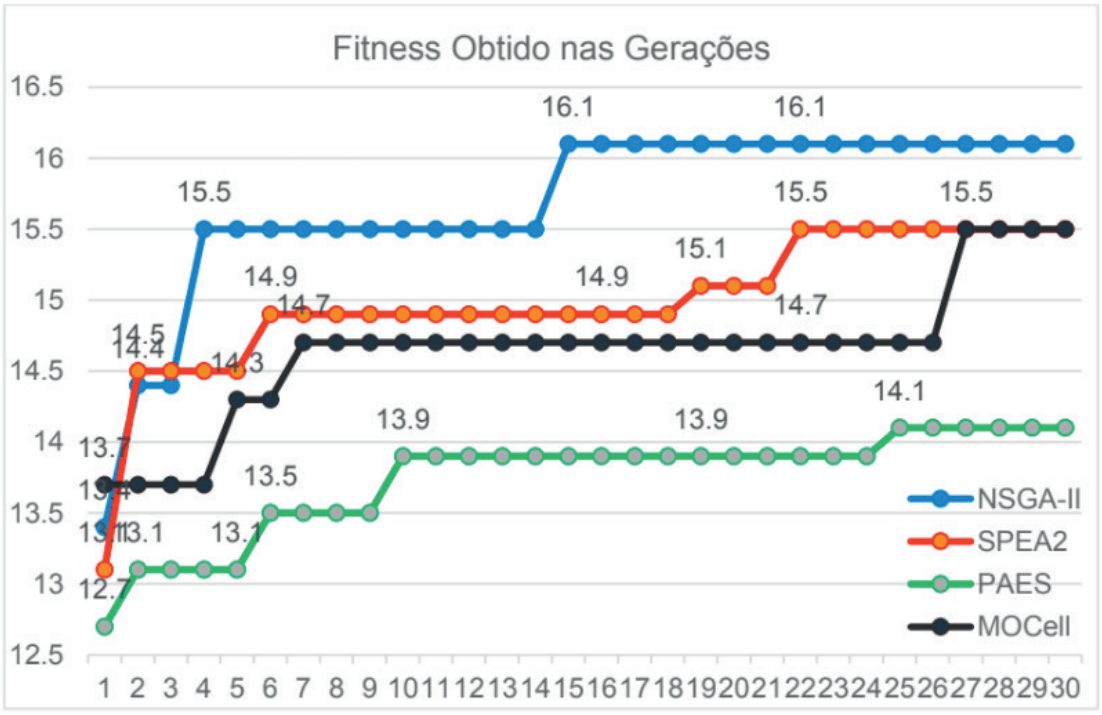


Figura 4: Resultados dos Experimentos com Rs-Parcial.

Fonte: Autores (2017).

Os pontos marcados linearmente com os rótulos, identificam o melhor caso de teste na população em sua geração, já os pontos não rotulados, significam que o melhor *fitness* daquela geração está abaixo do melhor caso de teste encontrado em uma geração anterior, ou seja, não houve melhora na geração seguinte.

No experimento 1 com *RS_Parcial* (Figura 4), o algoritmo NSGA-II apresentou o melhor valor de *fitness* em dois momentos, na 15ª e 22ª geração com utilidade = 16,1. Na utilização do algoritmo SPEA2, o melhor valor da função é obtido também na 22ª geração, entretanto, com utilidade = 15,5. Para o algoritmo PAES, o melhor valor é obtido na 25ª geração com utilidade = 14,1 e para o algoritmo MOCell, o melhor valor é obtido apenas na 27ª geração com utilidade = 15,5. A Tabela 3 destaca a geração e o melhor valor de utilidade encontrado por cada algoritmo.

Algoritmo	Geração	Utilidade
NSGA-II	15 e 22	16,1
SPEA2	22	15,5
PAES	25	14,1
MOCell	27	15,5

Tabela 3: Geração e Utilidade do Melhor Caso (Experimento 1)

Fonte: Autores (2017).

Podemos destacar também, o número de vezes (4 vezes) que os algoritmos SPEA2 e o PAES conseguiram evoluir suas populações de casos de teste. Assim, temos ao final desse primeiro experimento, considerando os melhores casos de teste, o NSGA-II com o melhor resultado, seguido pelos algoritmos SPEA2 e MOCell e, com o pior desempenho, o algoritmo PAES.

5.2 Experimento 2 - Resultados com RS-Parcial-Alterado

No experimento 2 com *RS-Parcila-Alterado* (Figura 5) podemos observar que esse agente é mais inadequado que o agente *RS_Parcial*. Isso ocorre devido as regras condição-ação do agente *RS_Parcial_Alterado* serem definidas aleatoriamente, possibilitando que o subsistema de tomada de decisão seja capaz de selecionar ações que podem produzir episódios com falhas, tornando o valor de *Utilidade* mais inadequado.

A Tabela 4 destaca a geração e o melhor valor de utilidade encontrado por cada algoritmo nesse segundo experimento. Na utilização do algoritmo NSGA-II, o melhor valor é obtido na 26ª geração com utilidade = 23,6. Nas próximas gerações, esse valor é tomado apenas como referência e não é selecionado nenhum caso de teste com valor de utilidade superior. Para o algoritmo SPEA2, o melhor valor da função é obtido na 27ª geração com utilidade = 23,4. Já para o algoritmo PAES, o melhor valor é obtido na 28ª geração com utilidade = 21,8 e para o último algoritmo, MOCell, o melhor valor

é obtido na 8ª geração com utilidade = 20,6, entretanto, o algoritmo não conseguiu mais gerar casos de testes com utilidade superior a partir dessa geração.

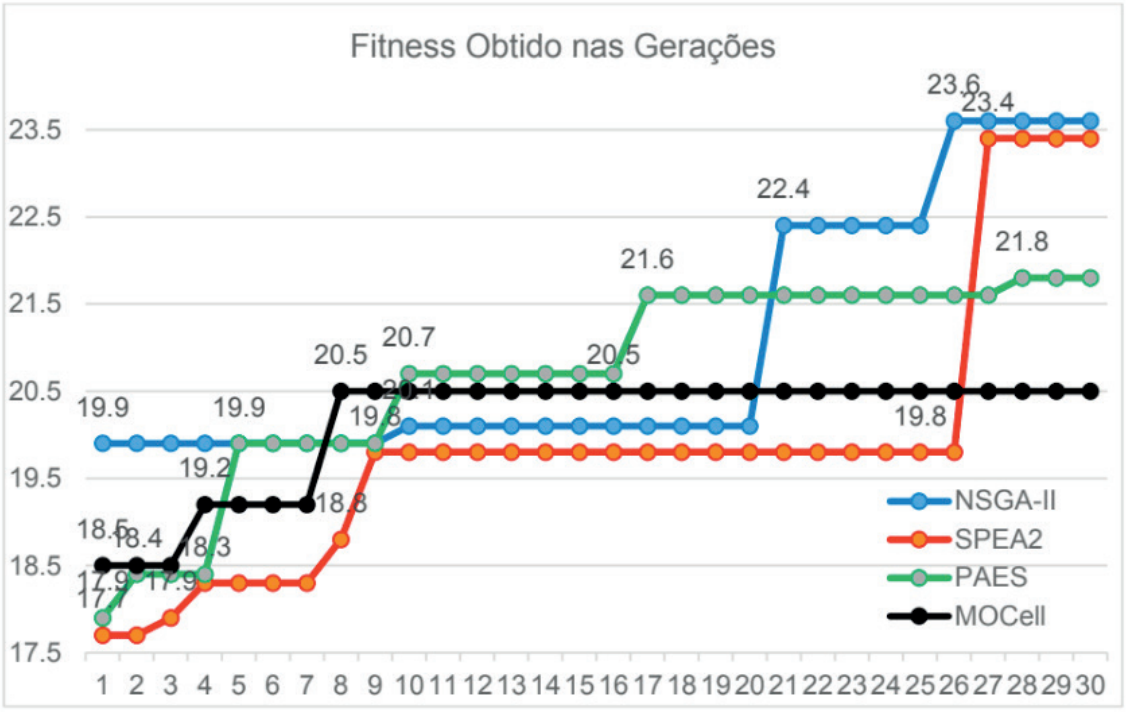


Figura 5: Resultados dos Experimentos com Rs-Parcial-Alterado.

Fonte: Autores (2017).

Algoritmo	Geração	Utilidade
NSGA-II	26	23,6
SPEA2	27	23,4
PAES	28	21,8
MOCell	8	20,5

Tabela 4: Geração e Utilidade do Melhor Caso (Experimento 2)

Fonte: Autores (2017).

Dessa forma, considerando os melhores casos de teste, o algoritmo NSGA-II e o SPEA2, foram os algoritmos que obtiveram os melhores resultados, seguidos pelo algoritmo PAES e o algoritmo MOCell apresentou o pior desempenho.

5.3 Experimento 3 - Resultados Com Rs-Total

A Figura 6 apresenta os experimentos do terceiro experimento. O valor de inadequação do agente reativo simples em um ambiente totalmente observável é bem menor que os valores dos outros dois agentes em um ambiente parcialmente observável. *RS_Total* percebe todo o ambiente e isso permite a concepção de um subsistema de tomada de decisão capaz de selecionar as ações que sejam realmente racionais em cada interação com o ambiente.

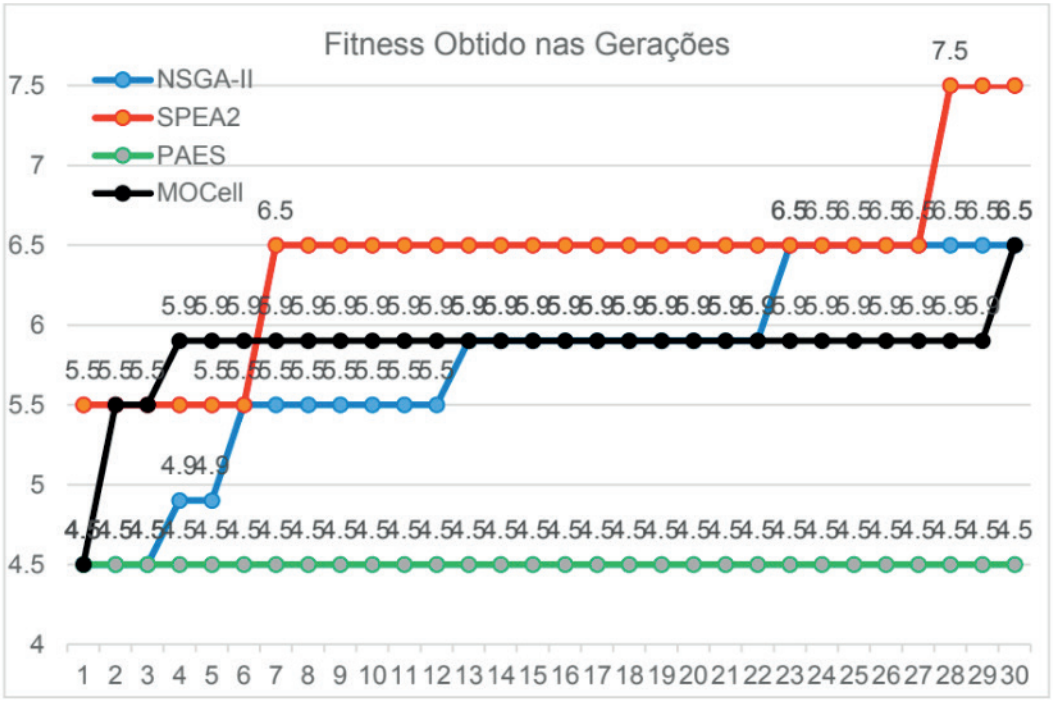


Figura 6: Resultados dos Experimentos com Rs-Total.

Fonte: Autores (2017).

Na utilização do NSGA-II, o algoritmo obteve o seu melhor valor da função *Utilidade* na 23ª geração com utilidade = 6,5. Na utilização do algoritmo SPEA2, o melhor valor da função é obtido também na 28ª geração com utilidade = 7,5. Na utilização do PAES, o algoritmo não conseguiu gerar casos de testes mais eficientes, tendo o mesmo valor de utilidade em todas as gerações. Para o algoritmo MOCell, o melhor valor é obtido apenas na 30ª geração com utilidade = 6,5.

A Tabela 5 destaca a geração e o melhor valor de utilidade encontrado no terceiro experimento pelos quatro algoritmos com o agente reativo simples em um ambiente totalmente observável. Nesse experimento, temos como maior destaque negativo, a ineficiência do algoritmo PAES em gerar casos de testes para um agente reativo simples que possua a observabilidade total do seu ambiente de tarefa. Isso pode ser justificado devido o algoritmo em seu processo evolutivo, manter apenas a melhor solução em cada geração e a estratégia na geração de novos indivíduos, consiste em utilizar somente o operador de mutação diferente das estratégias tradicionais de algoritmos evolutivos.

Algoritmo	Geração	Utilidade
NSGA-II	23	6,5
SPEA2	28	7,5
PAES	01 a 30	4,5
MOCell	30	6,5

Tabela 5: Geração e Utilidade do Melhor Caso (Experimento 3)

Fonte: Autores (2017).

Podemos destacar também ainda, o valor dos melhores *fitness* se mantém iguais ao longo das gerações na maioria dos algoritmos testados antes de encontrar uma nova melhor solução. Por exemplo, no caso do algoritmo MOCell, o segundo melhor valor alcançado na 4ª geração, é mantido nas gerações seguintes até a 29ª geração quando há uma melhora no valor da função, ou seja, o melhor indivíduo das gerações seguintes a partir a 4ª também atingiu o mesmo valor de utilidade. Isso está relacionado a observabilidade do agente testado, que mantém resultados semelhantes entre os indivíduos, ocasionando assim, pouca variação durante a geração de novos casos de testes para induzir a diversidade das populações.

Temos ao final do terceiro experimento, o algoritmo SPEA2 com sendo o melhor algoritmo em gerar casos de teste para esse tipo de agente em comparação com os demais algoritmos. O algoritmo NSGA-II foi o segundo algoritmo que obteve o melhor resultado. Porém, deve ser ressaltado, que o algoritmo conseguiu gerar casos de testes melhores que o anterior durante quatro oportunidades e iniciou com sua geração com o valor de utilidade abaixo do algoritmo SPEA2.

5.4 Experimento 4 - Resultados com Rm-Parcial

A geração e o melhor valor de utilidade encontrado pelos quatro algoritmos com agente reativo baseado em modelo em um ambiente parcialmente observável é mostrado na Tabela 6.

A Figura 7 apresenta os experimentos do agente *RM_Parcial*. Na utilização do algoritmo NSGA-II, o melhor valor da função é obtido na 26ª geração com utilidade = 13,1. Na utilização do algoritmo SPEA2, o melhor valor da função é obtido na 8ª geração com utilidade = 11,9. Na utilização do PAES, o melhor valor é obtido na 12ª geração com utilidade = 9,9. Para o algoritmo MOCell, o melhor valor é obtido na 30ª geração com utilidade = 12,3. Nesse experimento, o NSGA-II foi o algoritmo que obteve o melhor resultado, seguido pelos algoritmos MOCell e SPEA2 e com o pior desempenho, o algoritmo PAES.

Algoritmo	Geração	Utilidade
NSGA-II	26	13,1
SPEA2	8	11,9
PAES	12	9,9
MOCell	30	12,3

Tabela 6: Geração e Utilidade do Melhor Caso (Experimento 4)

Fonte: Autores (2017).

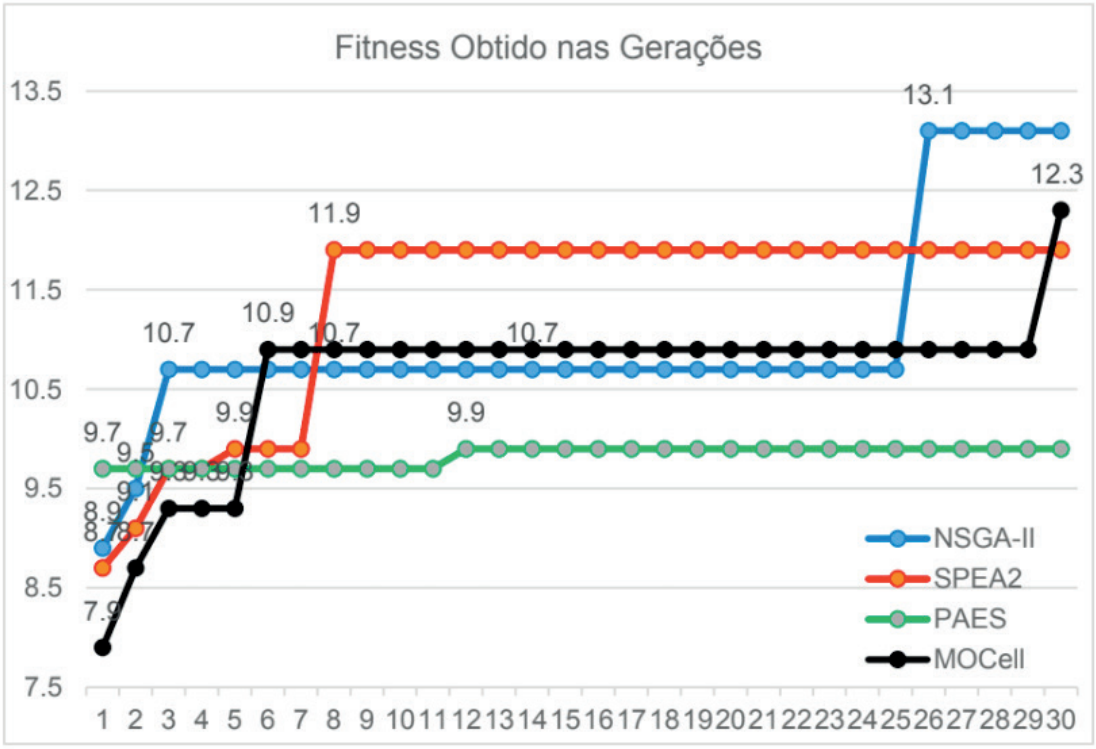


Figura 7: Resultados dos Experimentos com Rs-Total

Fonte: Autores (2017).

6 I CONCLUSÕES

Esta pesquisa apresentou a aplicação de um agente testador orientado por uma função utilidade e pelas estratégias de busca local dos algoritmos evolucionários multiobjetivos (MOEAs) baseadas em populações para encontrar conjuntos de casos de teste satisfatórios para o problema de otimização de seleção de casos de teste.

Com os resultados obtidos por meio dos quatro experimentos para cada algoritmo, a viabilidade da utilização da estratégia NSGA-II que identificou, em média, maiores situações de casos de teste em que o agente obteve um maior valor da função *Utilidade*, ou seja, piores desempenhos. Em segundo lugar ficou o algoritmo SPEA2 que conseguiu atingir um desempenho acima das demais técnicas implementadas, o que a credenciou como uma boa técnica de otimização para o problema estudado juntamente com NSGA-II. O algoritmo evolutivo MOCell obteve o terceiro melhor desempenho e o algoritmo PAES foi o algoritmo com o pior desempenho que obteve em média, o menor valor da função, se mostrando uma opção ineficiente na geração de casos de testes para essa problemática.

Assim, o agente testador selecionou um conjunto de casos de teste satisfatório em termos dos resultados gerados sobre o desempenho irregular do agente, principalmente com a utilização dos algoritmos NSGA-II e SPA2. Dessa forma, a utilização do elitismo pelos algoritmos NSGA-II e SPA2, tem como objetivo, prevenir a perda do melhor caso de teste encontrado em uma geração anterior. Com base nesses resultados,

a abordagem possibilita informações que possam realizar mudanças objetivas na estrutura interna do agente de forma a melhorar seu desempenho, ressaltando a importância dessa pesquisa, tanto na área de Inteligência Artificial (IA) como de Engenharia de Software (ES).

O aperfeiçoamento e a continuidade deste trabalho, constituem-se em oportunidades para trabalhos futuros que incluem além do emprego de agentes com maiores complexidades a serem testados, uma proposta de um agente testador que usa aprendizagem de máquina para resolução do problema de seleção de casos de testes. Para isso, o agente testador poderá utilizar algoritmos de aprendizagem de máquina para aprender o desempenho dos agentes testados a partir de um conjunto de atributos e propriedades obtidas das características do ambiente e do problema que o agente busca solucionar.

REFERENCIAS

CARNEIRO, S. M.; SILVA, T. A. R. da; RABÊLO, R. D. A. L.; SILVEIRA, F. R. V.; CAMPOS, G. A. L. de. **Artificial immune systems in intelligent agents test.** In: 2015 IEEE Congress on Evolutionary Computation (CEC), 2015. p. 536–543. ISSN 1089-778X.

COELLO GARY B. LAMONT, D. A. V. V. a. C. A. C. **Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition.** 2. ed. Springer US, 2007. (Genetic and Evolutionary Computation Series). ISBN 978-0-387-33254-3,978-0-387-36797-2. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=E14DAD9680795E083639D060B9BD8247>.

COSTA, E.; SIMÕES, A. **Inteligência Artificial: Fundamentos e Aplicações**. 3. ed. Lisboa: FCA, 2015.

HOUHAMDI, Z. **Test suite generation process for agent testing**. Indian Journal of Computer Science and Engineering IJCSE, v. 2, 2011.

NGUYEN, C. D.; MILES, S.; PERINI, A.; TONELLA, P.; HARMAN, M.; LUCK, M. **Evolutionary testing of autonomous software agents**. Autonomous Agents and Multi-Agent Systems, v. 25, n. 2, p. 260–283, 2012. ISSN 1573-7454. Disponível em: <http://dx.doi.org/10.1007/s10458-011-9175-4>.

PONSICH, A.; JAIMES, A. L.; COELLO, C. A. C. **A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications.** IEEE Transactions on Evolutionary Computation, v. 17, n. 3, p.321–344, June 2013. ISSN 1089-778X.

RUSSELL, S.; NORVIG, P**. Inteligência Artificial**. 3. Ed. Rio de Janeiro: Elsevierl, 2013.

SILVEIRA, F. R. de V.; CAMPOS, G. A. L. de; CORTéS, M. I. **A problem-solving agent to test rational agents - a case study with reactive agents.** In: Proceedings of the 16th International Conference on Enterprise Information Systems. 2014. p. 505–513. ISBN 978-989-758-027-7.

SILVEIRA, F. R. V.; CAMPOS, G. A. L. de; CORTéS, M. **Monitoring and diagnosis of faults in tests of rational agents based on condition-action rules**. In: Proceedings of the 17th International Conference on Enterprise Information Systems. 2015. p. 585–592. ISBN 978-989-758-096-3.

Agência Brasileira do ISBN
ISBN 978-85-7247-048-3