

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**UMA ABORDAGEM PARA SELEÇÃO E
ALOCÇÃO DE PESSOAS BASEADA EM
PERFIS TÉCNICOS E DE PERSONALIDADES
PARA PROJETOS DE SOFTWARE**

JORCYANE ARAÚJO LIMA

João Pessoa – PB
Agosto/2019

JORCYANE ARAÚJO LIMA

**UMA ABORDAGEM PARA SELEÇÃO E
ALOCÇÃO DE PESSOAS BASEADA EM
PERFIS TÉCNICOS E DE PERSONALIDADES
PARA PROJETOS DE SOFTWARE**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Gledson Elias da Silveira

João Pessoa – PB
Agosto/2019

Catálogo na publicação
Seção de Catalogação e Classificação

L732a Lima, Jorcyane Araújo.

Uma abordagem para seleção e alocação de pessoas baseada em perfis técnicos e de personalidades para projetos de software / Jorcyane Araújo Lima. - João Pessoa, 2019.

167 f. : il.

Orientação: Gledson Elias.

Dissertação (Mestrado) - UFPB/CI.

1. Projetos de software. 2. Seleção e alocação de pessoas. 3. Tipologias de personalidades. 4. Otimização multiobjetiva. I. Elias, Gledson. II. Título.

UFPB/BC

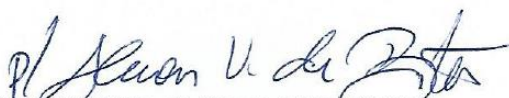


UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de
Mestrado de Jorcyane Araújo Lima, candidato ao título de
Mestre em Informática na Área de Sistemas de Computação,
realizada em 30 de agosto de 2019.

1 Aos trinta dias do mês de agosto, do ano de dois mil e dezenove, às nove horas, no Centro
2 de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os
3 membros da Banca Examinadora constituída para julgar o Trabalho Final da Sr^a. Jorcyane
4 Araújo Lima, vinculada a esta Universidade sob a matrícula nº 20171004354, candidata ao
5 grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa
6 "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade
7 Federal da Paraíba. A comissão examinadora foi composta pelos professores: Gledson Elias
8 da Silveira (PPGI-UFPB) Orientador e Presidente da Banca, Danielle Rousy Dias da Silva
9 (UFPB), Examinadora Externa ao Programa, Juliana Dantas Ribeiro Viana de Medeiros
10 (IFPB), Examinadora Externa à Instituição. Dando início aos trabalhos, o Presidente da
11 Banca cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e
12 passou a palavra à candidata para que a mesma fizesse a exposição oral do trabalho de
13 dissertação intitulado: "Uma Abordagem para Seleção e Alocação de Pessoas Baseada em
14 Perfis Técnicos e de Personalidades para Projetos de Software". Concluída a exposição, a
15 candidata foi arguida pela Banca Examinadora que emitiu o seguinte parecer: "**aprovada**".
16 Do ocorrido, eu, Clauriton de Albuquerque Siebra, Coordenador do Programa de Pós-
17 Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros
18 da banca examinadora. João Pessoa, 30 de agosto de 2019.

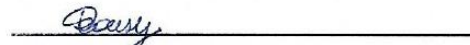

Prof. Dr. Clauriton de Albuquerque Siebra

Prof. Gledson Elias da Silveira
Orientador (PPGI-UFPB)

Prof. Danielle Rousy Dias da Silva
Examinadora Externa ao Programa (UFPB)

Prof. Juliana Dantas Ribeiro Viana de Medeiros
Examinadora Externa à Instituição (IFPB)







AGRADECIMENTOS

À Deus que sempre me deu forças e saúde para enfrentar os desafios. Testemunho a Sua ação em vários pontos do meu mestrado e de minha vida, que abriu todos os caminhos que permitiu a finalização deste trabalho e do meu fortalecimento pessoal.

Ao meu professor orientador, Dr. Gledson Elias da Silveira, pela ajuda e orientação.

A minha família pela compreensão e suporte que foram fundamentais para realização deste trabalho.

Aos professores Dr. Claurton de Albuquerque Siebra e Dra. Danielle Rousy Dias da Silva pela presença na banca examinadora da qualificação e às professoras Dra. Juliana Dantas Ribeiro Viana de Medeiros e Dra. Danielle Rousy Dias da Silva pela presença na banca examinadora da defesa final da dissertação.

Aos professores da UFPB pelo compartilhamento dos seus conhecimentos e experiências.

Enfim, agradeço a todos que, direta ou indiretamente, contribuíram com este trabalho, que é um dos passos em direção à realização de um sonho. Acredito que todos são pessoas que Deus colocou em meu caminho para me ajudar e rogo que Ele os abençoe.

RESUMO

As atividades dos projetos de desenvolvimento de software são agrupadas em diferentes papéis funcionais, cada qual executado por um tipo de engenheiro de software, que realiza diferentes atividades. Sendo assim, para cada um destes papéis são necessárias diversas habilidades técnicas e de personalidades específicas. Caso estes requisitos não sejam observados durante a seleção de membros para a equipe e alocação nos papéis, colocará o projeto de desenvolvimento de software em sério risco de fracasso, pois as pessoas selecionadas poderão não desempenhar bem as suas atribuições. Porém, o processo de seleção das pessoas e a alocação destas a papéis para a formação da equipe é um processo complexo para ser realizado pelo gerente de projetos com base apenas em sua experiência, pois, mesmo com um número relativamente pequeno de pessoas e papéis, uma grande quantidade de combinações pode ser identificada. Assim, analisar cada opção é inviável e boas opções podem não ser consideradas. A fim de fornecer suporte na tomada de decisão, explorando técnicas de Engenharia de Software Baseada em Buscas, do inglês *Search Based Software Engineering* (SBSE), este trabalho propõe uma abordagem multiobjetiva para seleção e alocação de pessoas tecnicamente qualificadas e psicologicamente adequadas para cada papel funcional do projeto de desenvolvimento de software. Para avaliação da proposta, além da comparação com a literatura existente relacionada com o tema da abordagem proposta, o trabalho apresenta resultados de três metaheurísticas distintas capazes de lidar com os múltiplos objetivos a serem otimizados. Estes algoritmos são aplicados a dados reais coletados em um estudo de caso que revelam a eficiência e aplicabilidade potencial da abordagem proposta.

Palavras-chave: Projetos de software. Seleção e alocação de pessoas. Tipologias de personalidades. Otimização multiobjetiva.

ABSTRACT

Software development project activities are grouped into functional roles, each of which is performed by a type of software engineer who carries out different activities. Therefore, for each of these roles, various technicals and personalities skills are required. Failure to meet these requirements when selecting team members and allocating roles will put the software development project at serious risk of failure as the selected people may not perform well. However, the process of selecting people and assigning them to roles is a complex process for the project manager to do based on his or her experience, because even with a relatively small number of people and roles, a large number of combinations can be identified. Thus, analyzing each option is not feasible and good options may not be considered. In order to support this decision process, exploring Search Based Software Engineering (SBSE) techniques, this paper proposes a multi-objective approach for selection and allocation of technically qualified and psychologically suitable people for each functional role of the software development project. To evaluate the proposal, besides comparing with the existing literature related to the theme of the proposed approach, the paper presents results from three distinct metaheuristics capable of dealing with the multiple objectives to be optimized. These algorithms are applied to real data collected in a case study that reveal the efficiency and potential applicability of the proposed approach.

Keywords: Software projects. Selection and allocation of people. Personality typologies. Multiobjective optimization.

LISTA DE FIGURAS

Figura 1 – Os 16 tipos de personalidade do MBTI	23
Figura 2 - Relação das dicotomias do MBTI	24
Figura 3 - Exemplo do item do teste de personalidade do MBTI	25
Figura 4 - Exemplo de resultado do teste de personalidade do MBTI	26
Figura 5 - Ten-Item Personality Inventory (TIPI).....	28
Figura 6 - Exemplo de um resultado do Personality Assessor	29
Figura 7 - Resultado de uma otimização biobjetiva de minimização.....	37
Figura 8 - Exemplos de soluções dominadas e não-dominadas	38
Figura 9 - Representação do espalhamento (a), otimalidade (b) e cenário ideal de soluções (c).....	40
Figura 10 - Cálculo do espalhamento para dois objetivos.....	40
Figura 11 - Hipervolume delimitado por um conjunto de soluções não-dominadas.....	42
Figura 12 - Ciclo de execução de um algoritmo genético	45
Figura 13 – Exemplo de cruzamento de ponto único	47
Figura 14 – Exemplo de cruzamento multipontos.....	47
Figura 15 – Exemplo de cruzamento uniforme	48
Figura 16 – Exemplo de mutação por flipping de bits	48
Figura 17 - Exemplo de mutação por reversão.....	49
Figura 18 - Cubóide do ponto i para dois objetivos	51
Figura 19 - Distribuição celular do espaço de busca.....	56
Figura 20 – Etapas da abordagem proposta.....	59
Figura 21 – Representação das dicotomias MBTI requeridas pelos papéis	63
Figura 22 – Exemplo de cálculo do nível de adequação não-técnica.....	65
Figura 23 – Representação dos domínios FFM requeridos pelos papéis	66
Figura 24 – Exemplos de representações de personalidades FFM.....	66
Figura 25 - Exemplo de cálculo de adequabilidade FFM	67
Figura 26 - Exemplo de formulário.....	71
Figura 27 - Exemplo de cromossomo	76
Figura 28 - Resultados do Experimento I com MBTI.....	89
Figura 29 - Resultados do Experimento II com MBTI	90
Figura 30 - Resultados do Experimento III com MBTI	91
Figura 31 - Resultados do Experimento IV com MBTI.....	91
Figura 32 - Resultados do Experimento V com MBTI	92
Figura 33 - Resultados do Experimento VI com MBTI.....	93
Figura 34 - Resultados do Experimento VIII	93
Figura 35 - Resultados do Experimento IX com FFM.....	95
Figura 36 - Resultados do Experimento X com FFM	95
Figura 37 - Resultados do Experimento XI.....	96
Figura 38 - Resultados do Experimento XII com FFM.....	96
Figura 39 - Resultados do Experimento XIII com FFM	97
Figura 40 - Resultados do Experimento XIV com FFM	97
Figura 41 - Resultados do Experimento XVI.....	98
Figura 42 - Versão modificada do cruzamento uniforme.....	101
Figura 43 - Resultados para o Experimento XVII com MBTI.....	105
Figura 44 - Resultados para o Experimento XVIII com MBTI.....	105
Figura 45 - Resultados para o Experimento XIX com FFM	110
Figura 46 - Resultados para o Experimento XX com FFM.....	111

Figura 47 - Resultado do Cenário IV MBTI: algoritmos x humanos.....	115
Figura 48 - Resultado do Cenário IV FFM: algoritmos x humanos.....	116
Figura 49 - Resultado do Cenário V MBTI: algoritmos x humanos.....	117
Figura 50 - Resultado do Cenário V FFM: algoritmos x humanos.....	118
Figura 51 - Profissionais candidatos e suas habilidades providas.....	129
Figura 52 - Tarefas e suas habilidades requeridas.....	130
Figura 53 - Efetividades dos funcionários para a tarefa 6 do exemplo	135
Figura 54 -Diagrama de classes do jMetal	153

LISTA DE TABELAS

Tabela 1 - Mapeamento de papéis e personalidades	34
Tabela 2 - Exemplos de tabelas de requisitos técnicos	69
Tabela 3 - Valores de referência para os níveis de complexidade	69
Tabela 4 - Papéis, requisitos e níveis de conhecimento	82
Tabela 5 - Parametrização dos experimentos com metaheurísticas (MBTI).....	103
Tabela 6 - Hipervolume normalizado, espalhamento e tempo de execução (MBTI).....	104
Tabela 7- Média e desvio padrão de quantidade de soluções presentes nas fronteiras (MBTI).....	106
Tabela 8 - Resultados do teste de Wilcoxon para o hipervolume normalizado (MBTI).....	106
Tabela 9 - Resultados do teste de Wilcoxon para o indicador de espalhamento (MBTI)	107
Tabela 10 - Parametrização dos experimentos com metaheurísticas (FFM).....	108
Tabela 11 - Hipervolume normalizado, espalhamento e tempo de execução (FFM).....	109
Tabela 12- Média e desvio padrão de quantidade de soluções presentes no conjunto ótimo ou de referência (FFM)	111
Tabela 13 - Resultados do teste de Wilcoxon para o hipervolume normalizado (FFM).....	112
Tabela 14 - Resultados do teste de Wilcoxon para o indicador de espalhamento (FFM)	112
Tabela 15 - Trabalhos relacionados	123
Tabela 16 - Habilidades consideradas por Avinash e Ramani (2014).....	125
Tabela 17 - Habilidades consideradas na plataforma MORGANA	132
Tabela 18 - Comparação dos trabalhos	137
Tabela 19 - Comparação dos trabalhos que abordam aspectos não-técnicos.....	138
Tabela 20 - Comparação dos trabalhos em relação a formulação do problema	139
Tabela 21 - Sumário de recursos disponíveis no jMetal 4.5.2.....	152

LISTA DE ABREVIATURAS E SIGLAS

ACO – *Ant Colony Optimization*

BFM – *Big Five Model*

CMMI – *Capability Maturity Model Integration*

ES – Engenharia de Software

FFM – *Five Factor Model*

KTS – *Keirsey Temperament Sorter*

HV – Hipervolume

IPIP – *International Personality Item Pool*

jMetal – *Metaheuristic Algorithms in Java*

LPS – Linha de Produto de Software

MBTI – *Myers-Briggs Type Indicator*

MOCeII – *MultiObjective Cellular genetic algorithm*

MOEA – *Multi-Objective Evolutionary Algorithm*; Algoritmo Evolucionário Multiobjetivo

MOP – *Multi-Objective Optimization Problem*; Problema de Otimização Multiobjetivo

NEO PI-R – *Revised NEO Personality Inventory*

NSGA-II – *Non-dominated Sorting Genetic Algorithm-II*

RH – Recursos Humanos

SBSE – *Search Based Software Engineering*; Engenharia de Software Baseada em Buscas

SPEA2 – *Strength Pareto Evolutionary Algorithm*

TI – Tecnologia da Informação

TIPI – *Ten-Item Personality Inventory*

TPG – *Task Precedence Graph*

TRT – Tabela dos Requisitos Técnicos

SUMÁRIO

1. Introdução.....	14
1.1. Motivação	15
1.2. Problema	17
1.3. Objetivos e metodologia	17
1.4. Estrutura do trabalho.....	19
2. Fundamentação teórica.....	20
2.1. Papéis dos engenheiros de software.....	20
2.2. Modelos de comportamento e personalidade.....	22
2.2.1. MBTI - <i>Myers-Briggs Type Indicator</i>	22
2.2.2. FFM - <i>Five Factor Model</i>	26
2.3. Gerenciamento e seleção de recursos humanos	29
2.3.1. Requisitos não-técnicos.....	30
2.3.1.1. Personalidade adequadas aos papéis da Engenharia de Software	32
2.3.2. Requisitos técnicos	34
2.4. Engenharia de software baseada em buscas.....	36
2.4.1. Problemas de otimização multiobjetivo	37
2.4.1.1. Indicadores de qualidade	39
2.5. Metaheurísticas	42
2.5.1. Algoritmos genéticos.....	43
2.5.2. NSGA-II.....	49
2.5.3. SPEA2	53
2.5.4. MOCeII.....	55
2.6. Considerações finais	57
3. Abordagem proposta	59
3.1. Visão geral da abordagem.....	59
3.2. Avaliação do Perfil de Personalidade	61
3.2.1. Formulação da avaliação não-técnica	62
3.2.1.1. MBTI	63
3.2.1.2. FFM	65
3.3. Avaliação do Perfil Técnico	68
3.3.1. Perfil Técnico dos Papéis	68
3.3.2. Perfil Técnico das Pessoas	70
3.3.3. Avaliação da qualificação técnica das pessoas para os papéis.....	74
3.4. Avaliação Integrada Pessoas-Papéis	75

3.4.1. Funções objetivos.....	77
3.5. Considerações finais	78
4. Resultados e discussão	79
4.1. Planejamento da pesquisa e coleta de dados.....	80
4.2. Experimentos	85
4.2.1. Experimentos com busca exaustiva.....	88
4.2.1.1. MBTI	89
4.2.1.2. FFM	94
4.2.2. Experimentos com metaheurísticas.....	98
4.2.2.1. MBTI	102
4.2.2.2. FFM	107
4.3. Análise empírica da competitividade dos resultados.....	112
4.4. Análise da validade.....	118
4.5. Considerações éticas	120
4.6. Considerações finais	121
5. Trabalhos relacionados.....	123
5.1. Escalonamento de projetos de software e alocação de recursos humanos	123
5.2. Abordagem multiobjetiva de alocação de pessoas em projetos de software	126
5.3. Otimização da alocação do RH de projetos de software.....	127
5.4. Otimização multiobjetivo para projetos de software	128
5.5. Algoritmo genético multiobjetivo para alocação de RH.....	131
5.6. Um assistente evolucionário baseado no conhecimento para escalonar projetos	133
5.7. Considerações finais	136
6. Conclusão	140
6.1. Trabalhos futuros	141
Referências bibliográficas	143
Apêndice A. jMetal.....	151
Apêndice B. Termo de autorização da participação da organização na pesquisa	154
Apêndice C. Termo de consentimento livre e esclarecido	155
Apêndice D. Questionário de coleta dos aspectos técnicos das pessoas	156
Apêndice E. Questionário de coleta dos aspectos não-técnicos das pessoas.....	167
Apêndice F. Perfis de personalidades das pessoas participantes.....	168

1. INTRODUÇÃO

O software é um produto de atividades humanas, tais como resolução de problemas, processamento de informações e interações sociais entre os envolvidos no seu desenvolvimento e operação (CAPRETZ; AHMED, 2010). Para desenvolvê-lo é necessário mais que manipulação de notações formais ou semiformais, pois envolve também a percepção do que o usuário deseja e realmente necessita. Portanto, as questões mais relevantes da engenharia de software (ES) envolvem as pessoas, além do fato de que as pessoas representam um custo significativo ao projeto (ROCHA, 2011).

Neste contexto, para resolver os diversos tipos de problemas relacionados ao desenvolvimento de software é necessária uma variedade de habilidades técnicas e traços de personalidades. Para atingir o sucesso do projeto de software é necessário que as pessoas trabalhem juntas em equipes, com divisão de atividades e papéis, pois um melhor software resultará dos esforços combinados de uma variedade de processos mentais, perspectivas e valores (CAPRETZ, 2003).

Entretanto, montar equipes com a combinação ideal de pessoas com personalidades e habilidades necessárias é uma tarefa complexa. Em relação a personalidade, é necessário o entendimento de estudos da área de psicologia, pois as pessoas são diferentes entre si em diversos aspectos, por exemplo, na busca de soluções para problemas, na aprendizagem, nos relacionamentos interpessoais e no processo de tomada de decisão. Estas diferenças influenciam no desempenho da equipe e do profissional, pois exercem influência na motivação individual e coletiva, consequentemente tem impacto no desempenho do projeto (CÉSAR, 2009).

Além disso, no desenvolvimento de software, as atividades podem ser agrupadas em diferentes papéis funcionais, que na maioria das vezes, correspondem ao ciclo de desenvolvimento, por exemplo: análise, projeto, desenvolvimento, testes e manutenção (CAPRETZ, 2003). Segundo César (2009), para cada papel funcional, são exigidos diferentes níveis qualitativos e quantitativos de habilidades não-técnicas (criatividade, persistência, atenção a detalhes, comunicabilidade, flexibilidade e liderança - todas são influenciadas pelas personalidades) e técnicas (ferramentas, tecnologias e conceitos).

Sendo assim, é preciso identificar as necessidades técnicas (conhecimentos, habilidades e experiências) e as formações acadêmicas necessárias para a execução das atividades dos papéis do projeto que serão atribuídas às pessoas. Segundo Barreto (2005), atribuir pessoas a

atividades que exigem mais qualificações do que possuem pode acarretar atrasos, menor qualidade ou não conclusão da atividade. Por outro lado, a atribuição de atividades que exigem qualificação menor que as possuídas pelos funcionários pode levar a frustração das pessoas ou aumento desnecessário do custo do projeto.

Dado este cenário, pode-se concluir que a gestão de pessoas é bastante importante para os projetos de desenvolvimento de software, visto que o desenvolvimento de software é uma atividade que demanda bastante esforço mental, é desenvolvida por equipes e é composta por atividades técnicas agrupadas que definem papéis funcionais distintos na equipe (CÉSAR, 2009). Portanto, nota-se que a adequada seleção dos recursos humanos (RH) que participarão de um projeto de desenvolvimento de software é fundamental para o sucesso, pois impacta diretamente no seu custo e eficiência. Sendo assim, é de grande importância a existência de uma abordagem de suporte a decisão que norteie os gerentes de projetos na seleção de RH adequado para as atividades de um projeto de software (SOUZA; ELIAS, 2019).

Apesar de relevantes contribuições encontradas na literatura, pode-se notar que as propostas, em sua maioria, mantêm um foco nos aspectos técnicos da seleção de pessoas, negligenciando outros aspectos importantes, como os relacionados à personalidade, que pode impactar profundamente de maneira positiva ou negativa no sucesso do projeto de software. Neste sentido, o objetivo principal deste trabalho é fornecer uma abordagem que considere as necessidades técnicas e as características psicológicas ideais para cada papel do desenvolvimento de software, além das habilidades técnicas e características psicológicas de cada profissional candidato a integrar a equipe do projeto. Ou seja, selecionando as pessoas aptas ao projeto, formando equipes com a divisão de papéis necessários ao desenvolvimento de todo projeto.

1.1. Motivação

De acordo com a NBR ISO 10006 (ABNT, 2006), as pessoas determinam a qualidade e o sucesso de um projeto qualquer, incluindo projetos de software. Esta norma tem a finalidade de definir diretrizes para atingir a qualidade do gerenciamento de projetos, para isso rege também os processos relacionados às pessoas, incluindo o processo de alocação da equipe, que visa selecionar pessoal suficiente e apropriado ao projeto. Para este processo, há a recomendação de que seja definida a necessária competência em termos de educação, conhecimento e experiência, considerando também os interesses pessoais, relacionamentos interpessoais, pontos fortes e fracos. Estas informações ajudarão na seleção de pessoal e na

identificação da melhor distribuição de responsabilidades entre os membros do projeto (ABNT, 2006).

Nesta direção, na maioria das organizações de desenvolvimento de software, cabe ao gerente de projetos a seleção de pessoal para os projetos, sendo um grande desafio para o tomador de decisão, pois é uma tarefa complexa e custosa (SILVA, F. 2009), já que a medida que o problema cresce, o número de soluções cresce exponencialmente, pois há uma série de variações possíveis, tornando ineficiente a tomada de decisão baseada apenas no conhecimento do gerente. Assim, é importante um apoio ferramental aos gerentes na execução desta difícil atividade, de forma a reduzir o esforço da sua realização.

A complexidade desse processo é também agravada pela natureza subjetiva dos requisitos considerados, pois para que a seleção de pessoas tecnicamente qualificadas possa acontecer, é necessário que os conhecimentos técnicos de cada pessoa sejam capturados e avaliados. Entretanto, realizar a representação e quantificação deste conhecimento não é algo trivial, não sendo suficiente a sua representação em dados *booleanos*, como possui ou não conhecimento, pois impossibilita a comparação entre candidatos que possuem pouco, médio ou muito conhecimento em determinada área, ferramenta ou conceito (SANTOS, 2014). Por outro lado, para que a seleção de pessoas em relação aos aspectos não-técnicos possa acontecer, é necessário que seja definido um instrumento de coleta, quantificação e avaliação dos aspectos psicológicos das pessoas, o que não deve ser mais uma atividade enfadonha e cansativa inserida na rotina tão atribulada do gerente de projetos.

Assim, é evidente a importância de um apoio ferramental aos gerentes de projetos na execução desta difícil atividade, de forma a reduzir o esforço da sua realização. Para isto eles devem dispor de algum processo de apoio à decisão para auxiliá-los quanto ao processo de seleção de RH. Salientando que um sistema de apoio à decisão não deve tomar a decisão pelo gerente, mas lhe prover informações que o auxiliem, deixando a seu cargo, com sua experiência, a decisão final (SANTOS, 2014).

Diante do grande espaço de busca deste problema, uma busca exaustiva, ou seja, a busca das melhores soluções dentre todas possíveis, é inviável. Por conta disso, boas soluções podem vir a ser desprezadas, acarretando perdas para o projeto. Para superar esta dificuldade, surge um novo ramo de pesquisa chamado Engenharia de Software Baseada em Buscas, em inglês *Search Based Software Engineering* (SBSE), que utiliza métodos de otimização para abordar problemas complexos da Engenharia de Software (BIBI; AHSAN; ANWAR, 2014), como é o caso do problema descrito neste trabalho. Portanto, os conceitos e ferramentas da SBSE tem enorme potencial de subsidiar a criação de uma ferramenta automatizada para auxílio aos

gerentes dos projetos de software na tarefa de seleção de quais pessoas serão alocadas aos papéis funcionais do ciclo de desenvolvimento, formando suas equipes de trabalho.

1.2. Problema

Neste contexto, diante do complexo cenário descrito nas seções anteriores, surgem as seguintes questões: Como formar a equipe de um projeto de desenvolvimento de software? Quais fatores devem ser observados no momento da seleção de um profissional para compor a equipe do projeto? Quais profissionais da equipe são mais adequados à execução das atividades referentes aos papéis do ciclo de vida do desenvolvimento de software? A investigação destas questões levou ao problema principal deste trabalho: "Como auxiliar o tomador de decisão na formação da equipe do projeto, de modo que seja composta por integrantes adequados para a execução de cada papel?".

Para este trabalho são consideradas, além dos fatores técnicos, as características de personalidade dos indivíduos, pois é importante para o sucesso do projeto, não só observar as competências técnicas dos indivíduos, mas também as suas características psicológicas (SILVA, F. 2009). Nesta direção, Martinez *et al.* (2011) concluiu que conhecer a personalidade dos engenheiros de software ajuda a construir uma equipe melhor, mais coesa e menos conflituosa. Portanto, a personalidade dos membros da equipe é um fator importante para o bom desempenho da equipe. Em suma, as equipes em que os membros foram estrategicamente atribuídos aos papéis da ES, nas quais todos os selecionados são tecnicamente qualificados e psicologicamente adequados, tem melhor desempenho em relação às demais equipes.

Vale salientar que, após a realização de uma análise dos trabalhos relacionados encontrados na literatura, pode ser concluído que a incorporação do aspecto não-técnico psicológico à seleção e alocação de pessoas representa uma importante contribuição e evolução para a área de estudo, pois são escassos os trabalhos que propõem uma abordagem de otimização de seleção e alocação de pessoas que considera os aspectos técnicos e de personalidade.

1.3. Objetivos e metodologia

Este trabalho tem como objetivo geral a concepção e implementação de uma abordagem automatizada que auxilie o gerente de projetos na escolha de quais pessoas comporão a equipe

do projeto de desenvolvimento de software, considerando que cada papel do ciclo de desenvolvimento possui necessidades técnicas e de personalidade específicas. Ou seja, montando equipes com a atribuição dos profissionais às vagas referentes aos papéis, respeitando e otimizando o atendimento dos requisitos técnicos e não-técnicos, de forma multiobjetiva. Mais especificamente, a abordagem proposta visa:

- Explorar conceitos e métodos relacionados à seleção e alocação de pessoas nos projetos de desenvolvimento de software;
- Identificar os critérios de **cunho técnico** e de **cunho psicológico** (personalidades) que influenciam no desempenho de cada papel do projeto;
- Conceber a estratégia de busca heurística para selecionar as pessoas aptas para cada papel do projeto, montando equipes formadas por profissionais que atendam às necessidades de todo ciclo de vida do desenvolvimento do software.

Para atingir os objetivos, este trabalho foi realizado em etapas seguindo os procedimentos metodológicos que orientam a realização de uma pesquisa científica. Sendo assim, a metodologia deste trabalho inclui os seguintes passos:

- Realização de uma pesquisa preliminar para definir o problema a ser discutido e o roteiro a ser seguido durante a pesquisa;
- Após a definição do tema, realização de uma pesquisa bibliográfica através de artigos científicos, teses e dissertações sobre trabalhos que possuem correlação com o tema;
- Identificação das necessidades de **cunho técnico** para cada **papel** do projeto;
- Realização de uma pesquisa bibliográfica para coletar mapeamentos das necessidades de **cunho psicológico** (personalidades) para cada **papel** do projeto;
- Enumeração e medição dos atributos **técnicos** (conhecimento) das **pessoas** candidatas;
- Caracterização dos atributos **psicológicos** (personalidade) das **pessoas** candidatas;
- Definição e adaptação de metaheurísticas de busca para o problema;
- Realização de estudo de caso para avaliar a escolha dos atributos considerados e para coletar dados reais que serão utilizados nos experimentos;

- Realização de experimentos para testar a utilização da abordagem, com o objetivo de validar o que está sendo proposto, avaliar e verificar a viabilidade da abordagem proposta.

Em suma, a partir dos atributos técnicos e não-técnicos levantados, é proposto um processo de avaliação de pessoas em relação aos papéis do projeto de desenvolvimento de software, visando a montagem de equipes formadas por pessoas tecnicamente qualificadas e psicologicamente adequadas para cada papel do projeto. Por serem objetivos conflitantes, este trabalho utilizará uma abordagem multiobjetiva, na qual, ao final será recomendado um conjunto de soluções não-dominadas, a qual almeja-se que possua boa diversidade e convergência, ou seja, que explore eficazmente o espaço de busca e seja próxima da otimalidade.

Adicionalmente, este trabalho também tem o objetivo de analisar os trabalhos que de alguma forma tratam o tema de seleção ou alocação de pessoas em projetos de software, comparando-os com a abordagem aqui proposta.

1.4. Estrutura do trabalho

O trabalho está dividido em seis capítulos, sendo o primeiro a introdução, composta por motivação, problema, objetivos e metodologia. O segundo capítulo debate a fundamentação teórica, que trata dos conceitos necessários para o entendimento da abordagem proposta neste trabalho. No terceiro capítulo é apresentada a abordagem proposta neste trabalho. Seguida de seus resultados e discussões, tratados no Capítulo 4. Posteriormente, no quinto capítulo são apresentados os trabalhos relacionados. Finalmente, o Capítulo 6 apresenta as considerações finais, onde são apresentadas as contribuições esperadas para este trabalho e seus possíveis trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados os temas inerentes ao conhecimento necessário para entender a proposta do presente trabalho. Dessa forma, inicialmente é feita uma breve introdução aos conceitos de papéis dos engenheiros de software, em seguida aos conceitos de modelos de comportamento e personalidade. Então são apresentados os conceitos de gerenciamento e seleção de recursos humanos, já que o presente trabalho trata da seleção e alocação de pessoas em papéis de desenvolvimento de software. Posteriormente, com relação à Engenharia de Software Baseada em Buscas, inicialmente é apresentado um panorama geral das áreas e problemas da engenharia de software que são tratados por técnicas de busca metaheurística, e, em seguida, é realizado um aprofundamento da discussão sobre as metaheurísticas adotadas na presente proposta.

2.1. Papéis dos engenheiros de software

A engenharia de software (ES) é uma disciplina que trata do desenvolvimento e manutenção de software, objetivando a qualidade do produto, idealmente, satisfazendo todos os requisitos, dentro do tempo e orçamento alocados (GULATI; BHARDWAJ; SURI, 2015). Já Sommerville (2011) define a ES como uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação até sua manutenção. A engenharia de software é caracterizada por um conjunto de atividades que incluem análise, projeto (design), programação, teste e manutenção do sistema, entre outras. Estas fases fazem parte do ciclo de vida do software, apesar de existirem outros modelos de processo de software que não consideram algumas ou incluem outras. Neste trabalho, será utilizado o mesmo modelo considerado por Capretz e Ahmed (2010) e Rehman *et al.* (2012), pois incluem as fases que ocorrem mais frequentemente e fazem parte do ciclo de vida mais aceito na literatura. Logicamente, as atividades dessas diferentes fases se combinam para atingir os objetivos de construção e operação de um produto de software (CAPRETZ; AHMED, 2010).

Sendo assim, os engenheiros de software também podem ser categorizados de acordo com a fase em que desenvolvem suas atividades: analistas de sistemas (também chamados de engenheiros de requisitos) desempenham suas atividades na fase de análise; projetistas de software na fase de projeto; programadores na fase de programação; engenheiros de testes na fase de testes e engenheiros de manutenção na fase de manutenção (REHMAN *et al.*, 2012).

A fim de melhorar o entendimento das atribuições de cada papel, serão explicadas as características principais de cada fase, segundo os conceitos apresentados em Capretz e Ahmed (2010) e Rehman *et al.* (2012). Destas, a fase de análise de sistemas requer que analistas de sistemas identifiquem as necessidades dos clientes, entenda as características essenciais do sistema e crie um modelo de aplicação abstrata adequada a estes requisitos. Os analistas de sistemas consideram os requisitos dos usuários e os convertem em modelos lógicos do sistema. Este modelo lógico pode ser gráfico ou descrição textual. Já a fase de projeto de software divide um projeto em subcomponentes de software. Em projetos grandes e complexos, a identificação de componentes-chave é uma tarefa árdua e demorada, as revisões são frequentes, já que um bom projeto geralmente leva várias iterações. Além disso, o número de iterações também depende do conhecimento e da experiência do projetista no domínio de aplicação. Na fase de programação, envolve traduzir o projeto gerado na fase anterior em códigos da linguagem de programação específica. Esta fase implica a identificação de estruturas de controle, variáveis relevantes e estruturas de dados. Além de necessitar que o programador tenha um detalhado entendimento de sintaxe e especificidades de linguagem de programação. Em seguida, a fase de testes envolve procurar defeitos no software. Estes defeitos podem emergir em qualquer outra fase, porém o principal foco desta fase é achar o máximo de defeitos possíveis. As estratégias de teste devem ser abordadas pelo testador de maneira metódica e sistemática, pois o teste requer atenção aos detalhes, além da pressão para cumprir prazos e entregar o produto ser enorme. Por fim, a fase de manutenção, que é necessária, pois o software normalmente está sujeito a mudanças contínuas depois de escrito e enquanto está operacional, seja por alterações nos requisitos do cliente, ou por mudanças na legislação da área de negócio do software, ou por correções de mal funcionamento. Conjuntamente, tais mudanças contínuas impõem a necessidade de manter um sistema em constante evolução (CAPRETZ; AHMED, 2010) (REHMAN *et al.*, 2012).

De acordo com Sommerville (2011), a abordagem sistemática usada na ES é chamada de processo de software, que é uma sequência de atividades que leva à produção de um produto de software. Para ele, existem quatro atividades fundamentais comuns a todos os processos de software: (i) Especificação de software: em que os clientes, em conjunto com os engenheiros definem o software a ser produzido; (ii) Projeto e implementação de software: em que o software é projetado e programado; (iii) Validação de software: em que o software é verificado para garantir que é o que o cliente deseja; e (iv) Evolução de software: em que o software é modificado para refletir as mudanças dos requisitos do cliente e do mercado. Estas atividades fundamentais, na prática, são complexas, incluindo subatividades como validação de requisitos,

projeto de arquitetura, testes unitários, entre outras. Os papéis refletem as responsabilidades das pessoas envolvidas no processo (SOMMERVILLE, 2011). Mesmo equipes ágeis, que são mais informais também adotam papéis (SRINIVASAN; LUNDQVIST, 2010).

Pode ser observado, de acordo com os conceitos e nomenclaturas, que as cinco fases enumeradas por Capretz e Ahmed (2010) e Rehman *et al.* (2012) se relacionam com as quatro fases definidas por Sommerville (2011). A fase análise de sistemas corresponde à fase especificação de software, as fases do projeto de software e programação têm correspondência com a fase projeto e implementação de software, a fase de testes corresponde a fase de validação de software e, por fim, a fase de manutenção tem correspondência com a fase evolução de software.

Vale salientar que os papéis presentes em um projeto de software dependem do processo de software adotado. Os processos de software são complexos e, como todos os processos intelectuais e criativos, dependem de pessoas para tomar decisões e fazer julgamentos. Não existindo um processo ideal, a maioria das organizações desenvolve seus próprios processos, que tem evoluído para tirarem proveito das pessoas em sua organização e as características específicas dos sistemas em desenvolvimento (SOMMERVILLE, 2011).

2.2. Modelos de comportamento e personalidade

A palavra personalidade se refere a padrões de comportamento e atitudes que são típicas de um determinado indivíduo, de forma que os traços de personalidade são diferentes de um indivíduo para outro, sendo, entretanto, relativamente constantes em cada pessoa e estáveis (REBOLLO; HARRIS, 2006). Embora existam vários métodos e teorias para a avaliação da personalidade, este trabalho tem como base dois modelos mais difundidos e aceitos (GULATI; BHARDWAJ; SURI, 2015) (CRUZ; SILVA; CAPRETZ, 2015): *Myers-Briggs Type Indicator* (MBTI) (MYERS; MCCAULLEY; MOST, 1998) e *Five Factor Model* (FFM) (DIGMAN, 1990). Segundo Barroso et al. (2017), o MBTI é o mais aplicado para identificar a personalidade dos desenvolvedores de software.

2.2.1. MBTI - *Myers-Briggs Type Indicator*

O MBTI (MYERS; MCCAULLEY; MOST, 1998) é um instrumento baseado na teoria dos tipos psicológicos criada por Jung (1991), sendo tanto um modelo quanto um instrumento

destinado a identificar o tipo psicológico de cada indivíduo (SILVA, D. 2009). O termo *type* (tipo) é usado para classificar os diferentes tipos de personalidade. Jung (1991) declarou que o comportamento humano não é aleatório, mas sim previsível e classificável. Além disso, mencionou que as diferenças do comportamento resultam das preferências individuais na utilização das diversas funções e atividades mentais básicas. Estas preferências aparecem nas fases iniciais da vida, constituindo as fundações da personalidade. Tais preferências logo se tornam o centro das atrações e repulsões das pessoas por outras pessoas, tarefas e eventos, durante toda vida.

Katharine Cook Briggs e sua filha Isabel Briggs Myers, em 1992, tendo como base psicológica a teoria sobre tipos psicológicos de Carl G. Jung, expandiram o modelo e o dotaram de uma aplicação prática dentro de ambientes organizacionais, estabelecendo quatro pares de dimensões para avaliar os tipos de personalidade: extroversão (E) e introversão (I), sensibilidade (S) e intuição (N), pensamento (T) e sentimento (F), julgamento (J) e percepção (P). Para delinear o tipo de personalidade do indivíduo são selecionados um traço de cada par, gerando dezesseis tipos distintos de personalidade, conforme ilustrado na Figura 1.

ISTJ	ISFJ	INFJ	INTJ
ISTP	ISFP	INFP	INTP
ESTP	ESFP	ENFP	ENTP
ESTJ	ESFJ	ENFJ	ENTJ

Figura 1 – Os 16 tipos de personalidade do MBTI (MYERS; MCCAULLEY; MOST, 1998)

Em suma, todas as oito preferências representadas pelas letras E/I, S/N, T/F e J/P, quando combinadas, designam os dezesseis tipos de personalidade definidos no MBTI, listados na Figura 1. Cada combinação de letras identifica o tipo psicológico de preferências do indivíduo (CAPRETZ; AHMED, 2010). A Figura 2 apresenta o relacionamento das quatro dicotomias, salientando que o MBTI pressupõe que as pessoas utilizam vários processos cognitivos, mas tendem a ter uma preferência entre as dicotomias citadas (FERREIRA; SILVA, 2008).



Figura 2 - Relação das dicotomias do MBTI (ARAÚJO; PEDRON, 2013)

Na Figura 2, o primeiro conjunto de preferências mentais, denominado extroversão-introversão, está relacionado a como os indivíduos são energizados, ou seja, qual a sua fonte de energia. Esta dimensão caracteriza os aspectos ligados a como a pessoa interage com o mundo e obtém energia. O traço Extroversão (E) caracteriza indivíduos comunicativos, falantes e iniciadores de conversas. Já o traço Introversão (I) caracteriza pessoas quietas, reservadas e que tendem a responder em conversas, ao invés de iniciar diálogos (CAPRETZ; AHMED, 2010). Os indivíduos extrovertidos, por exemplo, preferem estar em grandes grupos de pessoas, ao contrário dos introvertidos, que preferem estar sozinhos, concentrados em seu mundo interior, tendendo à introspecção (SILVA, D. 2009).

Em seguida, o segundo conjunto de preferências, sensação-intuição, identifica como os indivíduos percebem as informações (FERREIRA; SILVA, 2008), que tipo de informações notam, ou seja, como realizam o agrupamento de dados. Por um lado, o traço Sensação (S) caracteriza indivíduos que preferem perceber através de dados e informações tangíveis, utilizando os sentidos. Por outro lado, o traço Intuição (N) caracteriza pessoas interessadas em informações abstratas, conceituais, obtidas a partir de interpretação (SILVA, D. 2009). Em geral, os indivíduos S não gostam de novos problemas a menos que a experiência anterior mostre como resolvê-los e os indivíduos N gostam de novos desafios e não gostam de tarefas triviais (CAPRETZ; AHMED, 2010).

Posteriormente, o terceiro par de dicotomias, pensamento-sentimento, se refere ao processo de tomada de decisão (FERREIRA; SILVA, 2008). A dicotomia pensamento (T) representa uma preferência por tomada de decisões de maneira analítica, objetiva e lógica, além de definir uma orientação a princípios firmes. Enquanto a dicotomia sentimento (F) descreve a tendência de confiar nos valores emocionais como base para tomar decisões, além de definir pessoas com fortes habilidades interpessoais (CAPRETZ; AHMED, 2010). Segundo D. Silva

(2009), os indivíduos T tendem a dar ênfase aos resultados a serem alcançados e os indivíduos F prestam atenção ao impacto das decisões sobre outras pessoas.

Finalmente, o par julgamento-percepção, se refere ao estilo de vida (ARAÚJO; PEDRON, 2013), ou como a pessoa organiza a sua vida (FERREIRA; SILVA, 2008). Ou seja, se a pessoa vive de forma mais organizada ou mais espontânea. A pessoa J é extremamente organizada. No outro extremo, a pessoa P é procrastinadora e distraída na execução das tarefas. Em outras palavras, a aderência a prazos e pontualidade descrevem o traço J, enquanto a espontaneidade e adaptação se aplicam ao traço P.

Por exemplo, uma pessoa ISTJ tira energia do seu mundo interior e é bastante ligada a ele (I); gosta de obter informações através dos sentidos (S); prefere usar a razão para tomar decisões (T); e é organizada e pontual (J).

Em relação à maneira de coletar e classificar as personalidades, existem diversas ferramentas web gratuitas para avaliação da personalidade de acordo com o MBTI. Dentre elas, pode ser citada a disponibilizada pelo *Inspiira*¹, que consiste em um teste rápido no qual as pessoas escolhem as nuvens de palavras que melhor as descrevem em relação as quatro dicotomias MBTI. Um exemplo de uma das quatro questões do teste é ilustrado na Figura 3 e um exemplo de resultado do teste é ilustrado na Figura 4.



Figura 3 - Exemplo do item do teste de personalidade do MBTI

¹ <http://inspiira.org> – Inspiira Test

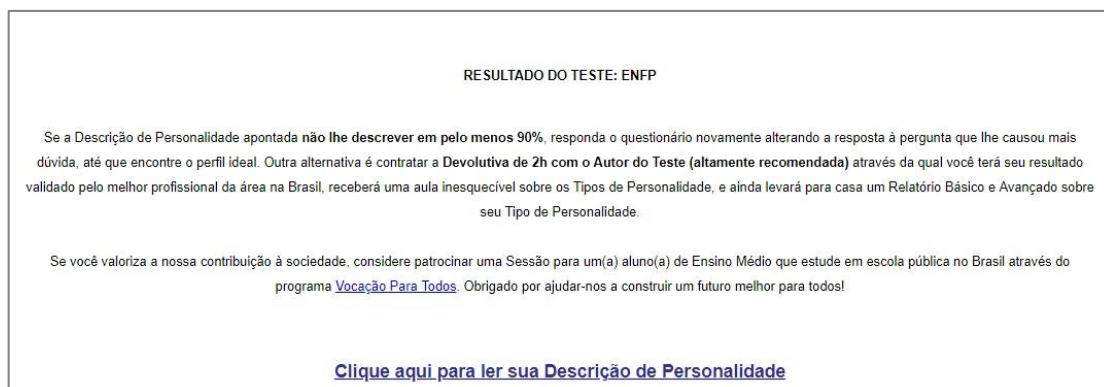


Figura 4 - Exemplo de resultado do teste de personalidade do MBTI

Outra opção de teste é disponibilizada por *Personality-tests.info*², que consiste em 28 questões e também retorna a classificação de personalidade MBTI do indivíduo. Como diferencial está o fato de que são disponibilizados alguns resultados anônimos de usuários para estudos acadêmicos.

2.2.2. FFM - *Five Factor Model*

O FFM (DIGMAN, 1990) é um modelo para descrever personalidades através de traços (*traits*), também chamado de fatores, em termos de cinco dimensões básicas: Abertura à experiência (*Openness to experience* - O), Conscienciosidade (*Conscientiousness* - C), Extroversão (*Extraversion* - E), Amabilidade (*Agreeableness* - A), e Neuroticismo (*Neuroticism* - N). As traduções utilizadas neste trabalho estão de acordo com a nomenclatura utilizada por Flores-Mendonça (2007). Cada um dos cinco fatores do modelo representa uma dimensão na qual um indivíduo se encontra. Através de procedimentos de avaliação, cada fator é medido para definir os aspectos marcantes da personalidade (COSTA; MCCRAE, 2008). O acrônimo OCEAN é utilizado para enumerar os fatores FFM (SILVA, D. 2009).

As pessoas que pontuam muito no fator abertura à experiência (O) são mais interessadas em aprender sobre coisas diferentes e ter experiências novas, são criativas, apreciam a arte, emoção, aventuras, ideias incomuns, curiosidades e variedade de experiências (REHMAN *et al.*, 2012). Este traço corresponde à percepção que o indivíduo tem da sua própria inteligência ou capacidade. Engloba, também, características como flexibilidade de pensamento, fantasia e imaginação, abertura para novas experiências (SILVA, D. 2009). Já as pessoas com baixa pontuação neste traço tendem a ter interesses mais convencionais e tradicionais (SILVA, 2015).

² <http://www.personality-tests.info> - *Personality-tests.info*

Quem possui alto nível do traço conscienciosidade (C) tende a ser mais focado em seus objetivos, disciplinado, bem organizado, trabalhador e tem uma abordagem sistemática para alcançar seus objetivos. Essas pessoas preferem planejar primeiro para depois agir em vez de agir espontaneamente (REHMAN *et al.*, 2012). Segundo D. Silva (2009), o fator C agrupa características que levam à responsabilidade e honestidade, ou, no outro extremo, à negligência e descuido. A pessoa conscienciosa tem força de vontade, é determinada, pontual, perseverante e de confiança. Entretanto, a pessoa com baixa pontuação neste fator é menos escrupulosa e menos obstinada na busca de seus objetivos.

Adicionalmente, o fator extroversão (E) mostra a tendência pela busca de estímulos no envolvimento com o mundo exterior e na companhia de outras pessoas. Pessoas que pontuam alto neste fator são sociáveis, falantes, comunicativas e enérgicas. Quando o indivíduo pontua pouco neste traço, tende a precisar de mais tempo sozinho, sendo reservado e quieto (SILVA, 2015).

Quanto à amabilidade (A), as pessoas que se enquadram nessa dimensão da personalidade são gentis, cooperativas e confiáveis. Elas também são tolerantes, simpáticas e emocionalmente estáveis. Além disso, elas também são confiantes, amigáveis, compassivas, cooperativas, complacentes, carinhosas e gentis (REHMAN *et al.*, 2012). A amabilidade indica uma tendência a ser socialmente amigável e dócil (SILVA, D. 2009). Ou seja, a pessoa com alta pontuação neste traço é altruísta, prestativa e simpática. Ao contrário, a pessoa é egocêntrica, rude, desconfiada, pouco cooperativa, vingativa, irritável e mais competitiva do que cooperativa.

Por último, a dimensão neuroticismo (N) contempla características de personalidade que envolvem aspectos positivos e negativos, ansiedade e estabilidade emocional. Um elevado nível do traço N resulta em indivíduos preocupados, nervosos, emocionalmente inseguros e respostas emocionais inadequadas. O aspecto central deste domínio é a tendência de experimentar emoções negativas, como tristeza, medo, embaraço, raiva e repulsa. Consequentemente, indivíduos com baixa pontuação nesta dimensão são emocionalmente estáveis, calmos, relaxados, seguros, satisfeitos consigo mesmo e de emoções constantes (SILVA, D. 2009).

Um reconhecido instrumento de medição da personalidade FFM é o NEO PI-R (*Revised NEO Personality Inventory*) (COSTA; MCCRAE, 2008) que possui 240 declarações nas quais os respondentes devem preencher com pontos de avaliação em uma escala de cinco pontos, que varia entre concordo fortemente e discordo fortemente. Adicionalmente, foi criada uma breve medida de personalidade chamada de Ten-Item Personality Inventory (TIPI) proposto por Gosling, Rentfrow e Swann (2003), que consiste em um questionário com 10 perguntas que

objetiva avaliar as personalidades das pessoas de acordo com o FFM. O TIPI foi escolhido como uma das opções de instrumento a ser utilizado, pois segundo Silva (2015), apesar de ser um instrumento simples, ele atingiu níveis de convergência aceitáveis quando comparados com instrumentos maiores, sendo uma excelente alternativa de medição das personalidades quando existem limites de recurso e tempo. Para responder o TIPI, os indivíduos utilizaram uma escala *Likert* (LIKERT, 1932) (cujas opções são "Discordo totalmente", "Discordo parcialmente", "Indiferente", "Concordo parcialmente" ou "Concordo totalmente"). Na Figura 5 o TIPI pode ser observado na versão em inglês.

Disagree strongly	Disagree moderately	Disagree a little	Neither agree nor disagree	Agree a little	Agree moderately	Agree strongly
1	2	3	4	5	6	7
<i>I see myself as:</i>						
1. ____	Extraverted, enthusiastic.					
2. ____	Critical, quarrelsome.					
3. ____	Dependable, self-disciplined.					
4. ____	Anxious, easily upset.					
5. ____	Open to new experiences, complex.					
6. ____	Reserved, quiet.					
7. ____	Sympathetic, warm.					
8. ____	Disorganized, careless.					
9. ____	Calm, emotionally stable.					
10. ____	Conventional, uncreative.					

Figura 5 - Ten-Item Personality Inventory (TIPI) (GOSLING; RENTFROW; SWANN, 2003)

Outra opção de instrumento para medir as personalidades retornando traços FFM chama-se International Personality Item Pool (IPIP) (GULATI; BHARDWAJ; SURI, 2015) e está disponível gratuitamente na internet, *Personality Assessor*³. Este instrumento disponibiliza uma versão do questionário completa (300 questões) e uma mais curta (44 questões). Esta última é mais simples e atende a necessidade de entrada de dados desta abordagem proposta. Muitos dos testes de personalidade do *Personality Assessor*³ são usados em pesquisas acadêmicas em andamento. Um exemplo de resultado de teste de personalidade realizado na ferramenta está ilustrado na Figura 6. Além da vantagem de que resultados anônimos são disponibilizados para pesquisas acadêmicas.

³ <http://www.personalityassessor.com/ipip300/> - *Personality Assessor*



Figura 6 - Exemplo de um resultado do *Personality Assessor*³

2.3. Gerenciamento e seleção de recursos humanos

Segundo o PMI (2014), gerenciamento de recursos humanos (RH) de um projeto é constituído pelos processos que organizam e gerenciam a equipe do projeto. Para que as organizações atinjam sucesso em seus projetos de software, elas necessitam gerenciar as pessoas alocadas de forma eficiente, pois a qualidade dos produtos a serem desenvolvidos é influenciada diretamente pela capacidade técnica e competências das pessoas envolvidas (SANTOS, 2014).

O gerenciamento de RH é composto de subprocessos, dentre eles há o planejamento da alocação de RH, que objetiva planejar a alocação das pessoas necessárias para a execução do projeto de software. Dentre as atividades pertencentes a este subprocesso, interessam para este trabalho as seguintes: *definição de perfis de competência* e *seleção de equipes*. Destes, a *definição de perfis de competência* consiste em definir perfis de competência necessários à execução de cada atividade do projeto. Um perfil de competência consiste na definição de um conjunto de atributos (que incluem conhecimentos, habilidades, formações acadêmicas, entre outros) visando à seleção e posterior alocação dos profissionais mais adequados para o projeto. Já a *seleção de equipes* consiste em selecionar quais profissionais serão alocados ao projeto. Esta seleção é realizada de acordo com os perfis de competência identificados como necessários para o projeto (SCHNAIDER, 2003).

Quanto à definição de perfis de competência, é ideal que as competências necessárias ao projeto sejam categorizadas e agrupadas, a fim de facilitar consultas, análises ou modificações (DINGSOYR; ROYRVIK, 2001). Dingsoyr e Royrvik (2001) ainda sugerem que sejam atribuídos níveis de graduação desejados e mínimos para cada competência identificada. Outras recomendações importantes são feitas por Silva (2007), uma delas preconiza que é útil guardar a lista de competências de projetos anteriores, pois pode ajudar na identificação das competências de projetos similares. A outra recomendação de Silva (2007) é que sejam

observadas as lições aprendidas em casos de sucesso ou de fracasso de alocações de projetos anteriores, a fim de tentar repeti-los ou evitá-los, respectivamente.

Quanto à seleção dos profissionais, a NBR ISO 10006 (ABNT, 2006) recomenda que o gerente de projetos se baseie nas descrições de tarefas, competências necessárias e experiências anteriores. Por ser uma tarefa complexa, é ideal que a seleção de RH tenha algum tipo de apoio à identificação das competências e tomada de decisão de quais recursos selecionar (SANTOS, 2014). No contexto de projetos de desenvolvimento de software, Herbsleb *et al.* (2006) afirmam que o apoio à tomada de decisão é um dos principais objetivos da engenharia de software, pois esta tem como objetivo melhorar as decisões técnicas que impactem positivamente no processo de desenvolvimento do software. Por isto, é necessário o apoio à tomada de decisões relacionadas à seleção de recursos humanos.

Assim, as subseções seguintes trazem quais tipos de requisitos podem ser considerados para a seleção de pessoas para formação da equipe (requisitos técnicos e não-técnicos). Ambos são importantes, pois o sucesso do projeto de desenvolvimento de software não depende apenas das técnicas, métodos e processos empregados, mas também do julgamento e decisões dos humanos envolvidos. Estes por sua vez são impactados pelas características de personalidade de cada indivíduo (FELDT *et al.*, 2010).

2.3.1. Requisitos não-técnicos

Segundo Sharp e Robinson (2005), a engenharia de software envolve vários aspectos sociais provenientes de várias interações interpessoais (com os clientes, entre os membros da equipe) necessárias para o desenvolvimento de software. Além disso, a ES é definida como sendo uma atividade social com atividades técnicas. Sendo assim, para selecionar pessoas adequadamente, os requisitos não-técnicos devem ser observados. Os requisitos não-técnicos (também conhecidos como *soft skills*) incorporam o fenômeno psicológico que incluem traços de personalidade, habilidades de interação social, comunicação e hábitos pessoais.

Para que a avaliação dos aspectos pessoais seja realizada, é necessária a utilização de instrumentos específicos. Sendo assim, na psicologia existem vários modelos e teorias que visam avaliar a personalidade das pessoas, consequentemente objetivam entender seus comportamentos e as diferenças entre elas. Cada um desses modelos utiliza diferentes perspectivas para avaliação do tipo psicológico das pessoas (CÉSAR, 2009). Dentre eles, podemos citar o *Myers-Briggs Type Indicator* (MBTI) e o *Five Factor Model* (FFM), já abordados anteriormente. De acordo com Bejanaro *et al.* (2005), estas teorias e modelos têm

sido estudados com o objetivo de melhorar o desempenho de uma equipe pelos seguintes motivos:

- Investigam as tendências naturais das pessoas, de forma a prover conhecimento de quais fatores e métodos podem ser utilizados para motivá-las;
- Investigam quais situações podem ser desmotivadoras, as quais podem prejudicar o desempenho da equipe;
- Auxiliam na formação de equipes compostas por membros com características equilibradas e necessárias para o projeto, de forma a otimizar sua produtividade;
- Auxiliam na elaboração de estratégias para a escolha de profissionais que melhor se adéquem às funções ou papéis do projeto; e
- Colaboram para o autoconhecimento dos profissionais de forma a auxiliá-los na busca de melhorias no relacionamento com os demais membros da equipe, provendo informações que podem usar para reconhecer as fraquezas e os pontos fortes, deles mesmos e dos demais.

Diante da importância dos aspectos humanos da personalidade na Engenharia de Software, existem vários trabalhos sobre o elemento humano na área da computação, especificamente no desenvolvimento de software. Alguns destes estudos visam identificar os perfis psicológicos dos profissionais da área de ES: Capretz (2003), Varona *et al.* (2012) e Sach, Petre e Sharp (2010). Outros objetivam identificar as personalidades mais adequadas para as profissões de ES, dividindo em papéis funcionais: Sodiya *et al.* (2007), Capretz e Ahmed (2010) e Rehman *et al.* (2012). Existem ainda os que buscam a relação entre os traços de personalidade dos profissionais com as estruturas eficazes de equipe de software: Yilmaz *et al.* (2017) e Gorla e Lam (2004). Além dos que visam criar técnicas de ensino customizadas para a área de computação: Rutherford (2001). Vale salientar que todos estes estudos são referentes a pesquisas de como a personalidade impacta na área de TI, diferente da proposta aqui apresentada que é o uso deste conhecimento na otimização da seleção de pessoas. Sendo assim, a proposta deste trabalho é focada no uso do mapeamento de quais personalidades são recomendadas ou não para os profissionais de TI, especificamente dos engenheiros de software e seus papéis. De acordo com o foco deste trabalho, a subseção seguinte abordará as características de personalidade mais importantes para cada tipo de engenheiro de software.

2.3.1.1. Personalidade adequadas aos papéis da Engenharia de Software

Nenhum tipo de personalidade pode desempenhar bem todos os tipos de tarefas. Assim, é importante mapear diferentes características de personalidade para diferentes tarefas a serem desempenhadas na ES (REHMAN *et al.*, 2012). Nesta direção, Capretz e Ahmed (2010) discerniram conexões entre traços de personalidade MBTI e o processo de desenvolvimento de software, identificando em qual fase do ciclo de vida do software um determinado tipo de personalidade MBTI tem o impacto mais significativo. Ou seja, Capretz e Ahmed (2010) mapearam os papéis da ES para os tipos de personalidade do MBTI mais adequados para a execução das atividades de cada papel. Para isto, mapearam os requisitos de trabalho dos papéis com as características de personalidade, baseados em profunda análise da literatura e de descrições em anúncios de emprego para engenheiros de software, relacionando empiricamente o impacto das dimensões MBTI nas diferentes fases dos processos de software. Posteriormente Rehman *et al.* (2012), baseado em Capretz e Ahmed (2010), mapearam os papéis para os tipos de personalidade do FFM.

Tanto Rehman *et al.* (2012), quanto Capretz e Ahmed (2010), mapearam os traços de personalidade para os seguintes papéis: analista de sistemas, projetista de software, programador, engenheiro de testes e engenheiro de manutenção. Visto que a definição de quais papéis são mais significantes depende das características do projeto e do processo de desenvolvimento (CAPRETZ; VARONA; RAZA, 2015), os papéis considerados foram os mais frequentes entre os modelos de ciclo de vida (CAPRETZ, 2012).

Para desempenhar o papel de analista de sistemas é requerido frequente contato com o cliente e com o time que projetará o sistema. Por esta razão, uma pessoa extrovertida é mais adequada para este papel. Além disso, o analista de sistemas deve ser cuidadoso com as necessidades do cliente a fim de entender totalmente os requisitos deles. Capretz e Ahmed (2010), usando o MBTI, mencionaram que eles devem possuir o traço extroversão (E) para comunicação e sentimento (F) para entender os requisitos dos clientes. Por sua vez, Rehman *et al.* (2012) mapearam as necessidades psicológicas deste papel para o FFM, obtendo as dimensões extroversão (E) e amabilidade (A).

No tocante ao projetista de software, o papel requer evidente criatividade humana, pois é um processo exploratório, no qual o projetista procura por componentes, experimentando uma variedade de esquemas para descobrir a mais natural e razoável forma de refinar uma solução. Consequentemente, os projetistas necessitam ter forte capacidade analítica e de resolução de problemas, e também devem ser inovadores, já que são necessárias para dividir o software

principal em componentes viáveis. Capretz e Ahmed (2010) mapearam as dimensões Pensamento (T) e Intuição (N) do MBTI como as que melhor se adequam ao papel, pois são as que trazem características psicológicas que correspondem às necessárias. Por sua vez, Rehman *et al.* (2012) mapeou para os domínios do FFM de Amabilidade (A) e Abertura à experiência (O).

Os programadores devem ser atentos a detalhes e manter um estilo de pensamento lógico e analítico. Além disso, a fase da programação exige muito menos interação com os outros. Portanto, devido a esses requisitos de trabalho, é recomendado que os programadores sejam, no MBTI: Sensitivos (S), Introversos (I) e Pensadores (T) (CAPRETZ;AHMED, 2010). E, de acordo com o FFM, os programadores devem ser abertos a experiência (O), com baixo nível de extroversão (E) (introversos), e amáveis (A) (REHMAN *et al.*, 2012).

Adicionalmente, os engenheiros de teste desempenham uma tarefa que pode ser frustrante e que requer atenção a minuciosos detalhes, que é a depuração. Também são muito pressionados a cumprir prazos. Além de que, para remover bugs, há a necessidade de mudanças na forma de pensar o problema. Portanto, prestar atenção aos detalhes minuciosos e cumprir os prazos são duas habilidades que os testadores precisam ter. Assim, os traços de personalidade Sensibilidade (S) e Julgamento (J) do MBTI são importantes para este papel (CAPRETZ;AHMED, 2010), que foram mapeados para a abertura à experiência (O) e conscienciosidade (C) do FFM (REHMAN *et al.*, 2012).

Por último, os engenheiros de manutenção precisam possuir o traço Sensibilidade (S) do MBTI, porque essas pessoas preferem trabalhar nas linhas que já foram testadas e provaram ser bem-sucedidas. Estes engenheiros também necessitam ser observadores e focados. Pessoas com o traço Percepção (P) do MBTI podem ser boas engenheiras de manutenção de software porque são abertas e adaptáveis a mudanças (CAPRETZ;AHMED, 2010). Os traços do FFM recomendados são conscienciosidade (C) e abertura à experiência (O) (REHMAN *et al.*, 2012).

Em suma, na Tabela 1 ambos os mapeamentos, MBTI e FFM, podem ser observados. Quanto ao MBTI, a dicotomia requerida de cada par é sinalizada na respectiva coluna. Observe que algumas dimensões não exercem influência no papel e são sinalizadas com o símbolo “-”. Em relação ao FFM, os domínios que são requeridos em nível mais alto são sinalizados com o símbolo “↑”. De forma contrária, os domínios que são requeridos em nível mais baixo são sinalizados com o símbolo “↓”.

Tabela 1 - Mapeamento de papéis e personalidades

Papel Funcional	Perfil MBTI				Perfil FFM				
	E/I	S/N	T/F	J/P	O	C	E	A	N
Analista de Sistemas	E	–	F	–	–	–	↑	↑	–
Projetista de Software	–	N	T	–	↑	–	–	↑	–
Programador	I	S	T	–	↑	–	↓	↑	–
Engenheiro de Testes	–	S	–	J	↑	↑	–	–	–
Engenheiro de Manutenção	–	S	–	P	↑	↑	–	–	–

2.3.2. Requisitos técnicos

Também chamados de *hard skills*, os requisitos técnicos são as habilidades técnicas e conhecimentos que uma pessoa deve possuir para desempenhar uma tarefa. Deste modo, eles incluem conhecimentos teóricos e experiências práticas que uma pessoa deve ter para executar confortavelmente uma atividade, já que a atribuição de tarefas adequadas às capacidades técnicas de cada profissional é de grande importância para que os níveis de produtividade e qualidade desejados possam ser atingidos em cada atividade do projeto (SILVA, 2007).

Assim é preciso identificar as necessidades técnicas (conhecimentos, habilidades e experiências) e as formações acadêmicas necessárias para a execução das atividades do projeto agrupadas em papéis funcionais da ES, de forma que os papéis possuam profissionais com as qualificações necessárias. No entanto, de acordo com as necessidades do projeto, diferentes políticas de seleção podem ser utilizadas, ou seja, critérios diferentes de como selecionar as pessoas para o projeto podem ser utilizados. Por exemplo, selecionar as pessoas que possuam os maiores conhecimentos sobre as tecnologias necessárias para o projeto pode ser uma boa opção quando o projeto está atrasado, já que as pessoas com maiores conhecimentos terão menos problemas técnicos para desenvolver o software, conseqüentemente, não excederão o prazo ou, até mesmo, podem diminuir o tempo estipulado para desenvolvê-lo. Porém esta estratégia pode ser considerada como desperdício de conhecimento, podendo desestimular e deixar as pessoas insatisfeitas, já que não lhes propicia tarefas desafiadoras. Além de, em geral, elevar os custos dos projetos, pois as pessoas que apresentam maiores conhecimentos, requerem maiores salários.

Por estas razões, uma política de seleção que vise aproximar os níveis de conhecimento requeridos pelo projeto e os níveis de conhecimento possuídos pelas pessoas pode ser considerada interessante. Note que esta política pode reduzir os custos, mas, em contrapartida, o projeto pode não ser concluído tão rápido quanto seria se fosse adotada a política de maior

conhecimento descrita anteriormente (SANTOS, 2014). Com a adoção desse tipo de política evita-se que pessoas com um conhecimento alto sejam selecionadas para projetos que não demandem um conhecimento profundo acerca das tecnologias utilizadas. Sendo assim, estas pessoas mais especialistas ficarão disponíveis para tarefas mais complexas e as pessoas com níveis baixos de conhecimento podem participar de projetos que demandem baixa habilidade técnica. O uso dessa política pode trazer um bom custo-benefício ao projeto, já que as pessoas são selecionadas de acordo com o necessário e não por serem as que apresentam maiores habilidades técnicas, evitando desperdício.

Entretanto, avaliar quem é e quem não é especialista em determinada área de conhecimento é uma tarefa complexa. Uma das formas de realizar este julgamento é através da elaboração de perguntas técnicas diretas, as quais são aplicadas aos avaliados e, em seguida, são conferidas as respostas dadas em comparação com as respostas corretas. Se a pessoa avaliada responder corretamente, então é um especialista. Caso contrário, não é considerada um especialista na área. Entretanto, segundo Shanteau et al. (2002), a elaboração de questões eficientes que consigam captar o nível de conhecimento geral na área é muito difícil, além de que para problemas que ocorrem frequentemente no ambiente de trabalho não existe a resposta correta, mas sim a necessidade de analisar caso a caso.

Nesta direção, para avaliar o nível de conhecimento das pessoas em determinado domínio são comumente usados como critérios de avaliação os anos de experiência, títulos e certificados possuídos pelas mesmas. Em relação à experiência, as pessoas que tenham trabalhado muitos anos na área de conhecimento desejada são consideradas especialistas, ou, caso contrário, pessoas que nunca trabalharam em determinada área ou que possuam poucos anos de experiência na área são consideradas novatas. Além disso, em relação aos títulos e certificados possuídos pelo avaliado, podem ser analisados a quantidade de atestados formais de habilidades em determinado ramo do conhecimento. Observe que estas comprovações devem ser emitidas por instituições de ensino ou órgãos certificadores confiáveis (SHANTEAU *et al*, 2002). Atualmente, muitas empresas requisitam essas comprovações como uma forma de provar que as pessoas conseguem exercer as tarefas do cargo que ocupa ou deseja ocupar.

Adicionalmente, segundo Weiss, Shanteau e Harries (2006), um fator importante que determina se alguém é um especialista em determinado domínio é o fator de capacidade de discriminar diferenças sutis em contextos similares. Esta capacidade de discriminação só pode ser vista através da observação de eventos passados, pois é um dado tipicamente empírico. Sendo assim, a capacidade de discriminação das pessoas pode ser inferida pelos números de projetos que tenham participado, pois quanto mais as pessoas participam de projetos maior é a

probabilidade de que tenham presenciado diferentes domínios, aumentando as suas capacidades de discriminação.

Porém, a avaliação da correspondência entre as necessidades técnicas que o papel requer e o nível de conhecimento e habilidade possuída pelos profissionais não é trivial, pois é necessário que primeiro os conhecimentos técnicos de cada pessoa sejam capturados para assim poderem ser avaliados. Entretanto, a representação de tal conhecimento é complexa, pois é bastante subjetiva, sendo insuficiente representar com dados booleanos, como possui ou não conhecimento (SANTOS, 2014), por isso, em geral, necessitam ser representadas de maneira que permita a atribuição de graus de competência técnica.

2.4. Engenharia de software baseada em buscas

A Engenharia de Software Baseada em Buscas (*Search-Based Software Engineering - SBSE*) é o ramo de estudo que utiliza técnicas de buscas metaheurísticas para problemas de otimização da engenharia de software (HARMAN; MANSOURI; ZHANG, 2009). A SBSE converte um problema de engenharia de software em um problema de busca computacional que pode ser resolvido com uma metaheurística. Isso envolve a definição de um espaço de busca ou o conjunto de possíveis soluções. Esse espaço é tipicamente grande demais para ser explorado exaustivamente, sugerindo uma abordagem metaheurística.

Segundo Harman *et al.* (2009), técnicas da SBSE estão sendo utilizadas em todos os aspectos da Engenharia de Software desde as fases iniciais do ciclo de vida de desenvolvimento de software, relacionadas aos requisitos, gerenciamento e planejamento, até as fases de pós-entrega de manutenção. Como exemplo, podem ser encontrados estudos para os problemas de seleção de requisitos para a próxima *release*, priorização de bugs, seleção de casos de testes, gerência de projetos (seleção de pessoas ou equipes), entre outros. Inicialmente, a SBSE tinha a meta de encontrar soluções ótimas ou próximas da ótima para os problemas tratados, no que diz respeito a um único objetivo. Recentemente surgiram diversos estudos que utilizam técnicas para lidar com múltiplos objetivos, permitindo a exploração do espaço de soluções candidatas que possuam certas propriedades, revelando a sua estrutura e os pontos potenciais que apresentam um *trade-off* atraente entre os objetivos (DURILLO *et al.*, 2010).

2.4.1. Problemas de otimização multiobjetivo

Em formulações multiobjetivo, o resultado da busca não é um único ponto (solução ótima ou quase ótima), mas sim, um conjunto de soluções candidatas, cada uma das quais não pode ser melhorada de acordo com um dos múltiplos objetivos a serem otimizados sem um impacto negativo em outro (DURILLO *et al.*, 2010). A saída de um algoritmo de otimização multiobjetivo com dois objetivos (biobjetivo) é um vetor de soluções para as funções de avaliação. A Figura 7 ilustra um exemplo deste tipo de saída. Deste modo, o tomador de decisões terá opções de escolha, além de poder visualizar facilmente o *trade-off* das soluções candidatas em relação a cada objetivo.

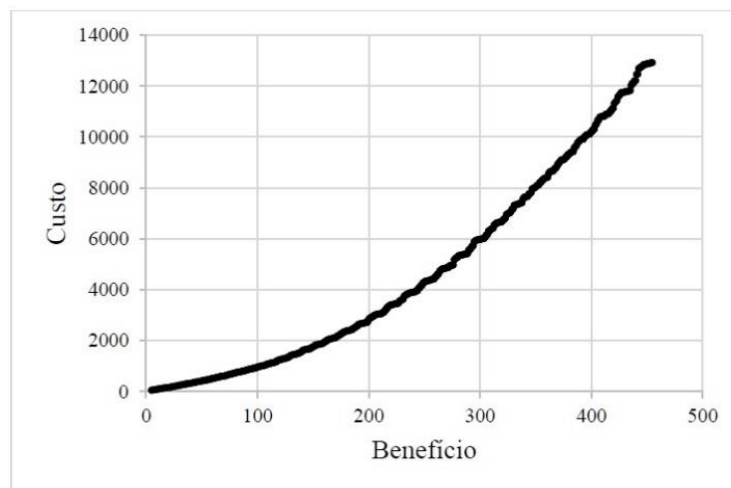


Figura 7 - Resultado de uma otimização biobjetiva de minimização (AMARAL, 2017)

Nesta direção, assim como em Durillo *et al.* (2011), a explanação a seguir assume que o problema é de minimização, sem perda de generalidade, pois a maximização pode ser considerada como minimização da função objetiva negativa. Um problema multiobjetivo (MOP - *Multi-Objective Optimization Problem*) pode ser definido pelo vetor $x = [x_1, x_2, \dots, x_n]^T$ de n variáveis de decisão que satisfaça as restrições e minimize o vetor de k funções objetivo $f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T$. O conjunto de todos os valores que satisfazem as restrições define a região de soluções factíveis Ω e $x \in \Omega$.

Neste cenário, o objetivo é encontrar um conjunto com as melhores soluções que minimizem o vetor de funções objetivo e que atendam aos parâmetros pré-determinados para o problema. Para isto, é necessário comparar as soluções encontradas e definir o quão boa cada solução é em relação às demais, que é realizado a partir do conceito de dominância de soluções. De acordo com Durillo *et al.* (2010), uma solução $x' = [x'_1, x'_2, \dots, x'_n]$ domina outra solução

$x'' = [x''_1, x''_2, \dots, x''_n]$, se e somente se $f_i(x') \leq f_i(x''), \forall i = 1, 2, \dots, k$. Além disso, deve existir pelo menos um objetivo j ($1 \leq j \leq k$) tal que $f_j(x') < f_j(x'')$. Ou seja, x' deve apresentar valores de função menores ou iguais para todos os objetivos em comparação com x'' , mas deve garantir que pelo menos uma das funções objetivo apresente uma avaliação melhor que x'' . Caso estas condições sejam satisfeitas, é dito que solução x' domina x'' , sendo representado pela expressão $x' \preceq x''$, a fim de indicar esta relação de dominância entre pares de soluções (**dominância de Pareto**). Caso contrário, as soluções são não dominantes entre si. Para os casos em que seja necessária a maximização de um vetor de funções objetivo, basta inverter os sinais de desigualdade das duas regras de dominância, de forma a comparar as soluções obtidas, para “maior ou igual que” e “maior que”, respectivamente.

A Figura 8 ilustra alguns exemplos de soluções dominadas e não-dominadas. Nesta figura, A' domina C' porque $f_1(A') < f_1(C')$ e $f_2(A') < f_2(C')$, ou seja, A' é melhor que C' em ambas funções objetivo. Em contrapartida, A' e B' são não dominadas entre si, porque $f_1(A') < f_1(B')$ e $f_2(A') > f_2(B')$, ou seja, A' é melhor que B' na primeira função objetivo, mas B' é melhor que A' na segunda função objetivo.

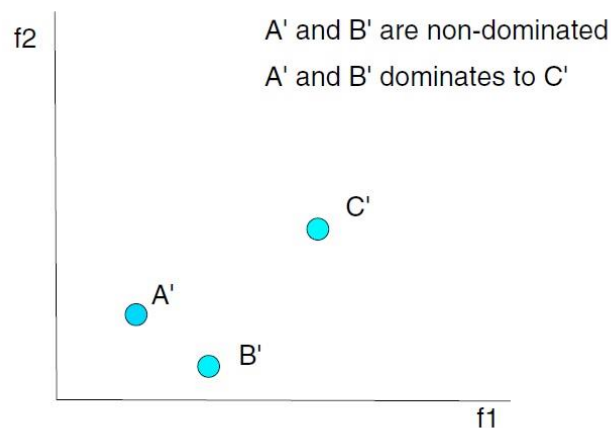


Figura 8 - Exemplos de soluções dominadas e não-dominadas (DURILLO *et al.*, 2010)

O conjunto de soluções não-dominadas por nenhuma outra solução da região Ω formam o conjunto ótimo de Pareto, conhecido como **Frenteira de Pareto**. Em suma, a Fronteira de Pareto satisfaz as seguintes condições: (i) todas as soluções no conjunto são não-dominadas; e (ii) qualquer outra solução é dominada por pelo menos uma solução no conjunto. Caso o problema de otimização contenha duas funções objetivo a serem avaliadas, a fronteira de Pareto é representada por uma curva em um espaço bidimensional.

2.4.1.1. Indicadores de qualidade

Gerar a fronteira de Pareto é o principal objetivo das técnicas de otimização multiobjetivo (AMARAL, 2017). Porém, na prática, algoritmos de busca heurística não garantem que seja encontrada, analogamente à busca pelo ótimo global em formulações mono-objetivas. Em teoria, uma frente de Pareto poderia conter muitos pontos, porém, na prática, uma solução aproximada utilizável conterá apenas um pequeno número deles. Assim, almeja-se que eles estejam os mais próximos possíveis da frente exata de Pareto e espalhados uniformemente, caso contrário, eles não seriam muito úteis para o tomador de decisão. A proximidade com a frente de Pareto garante que se lide com soluções ótimas ou bem próximas da otimalidade, enquanto uma disseminação uniforme das soluções significa que foi feita uma boa exploração do espaço de busca e nenhuma região foi deixada inexplorada (DURILLO *et al.*, 2010).

A Figura 9 ilustra exemplos de fronteiras de Pareto com diferentes propriedades qualitativas: (a) soluções se aproximam do conjunto ótimo de Pareto (boa convergência), mas possuem má diversidade (espalhamento ruim), ou seja, a fronteira contém soluções ótimas, porém existem algumas regiões não exploradas; (b) soluções não estão próximas à otimalidade (pobre convergência), mas apresentam um bom espalhamento uniforme ao longo da curva (boa diversidade), ou seja, as soluções da fronteira estão diversificadas, porém não são do conjunto ótimo; e (c) soluções se aproximam do conjunto ótimo de Pareto (boa convergência) e também apresentam um bom espalhamento uniforme ao longo da curva (boa diversidade), o que seria o cenário ideal.

Conclui-se que uma inspeção visual não é suficiente para determinar a qualidade de um conjunto de soluções obtido para um problema qualquer de múltiplos objetivos. Deste modo, é necessário algum indicador de qualidade responsável por quantificar estes conceitos e servir de demonstrativo a respeito de quão bom é o conjunto de soluções encontrado, permitindo estabelecer uma comparação. Esta comparação é de especial utilidade em situações em que é necessário avaliar diferentes metaheurísticas para um dado problema, de forma a estabelecer qual delas fornece o melhor resultado para o problema.

Dentre os indicadores de qualidade, o **Espalhamento** (Δ) é uma medida de diversidade do conjunto de soluções, responsável por demonstrar a extensão deste espalhamento do conjunto dado por uma fronteira de Pareto (DEB, 2001). O espalhamento é ilustrado na Figura 10 e seu cálculo dado na Equação 1.

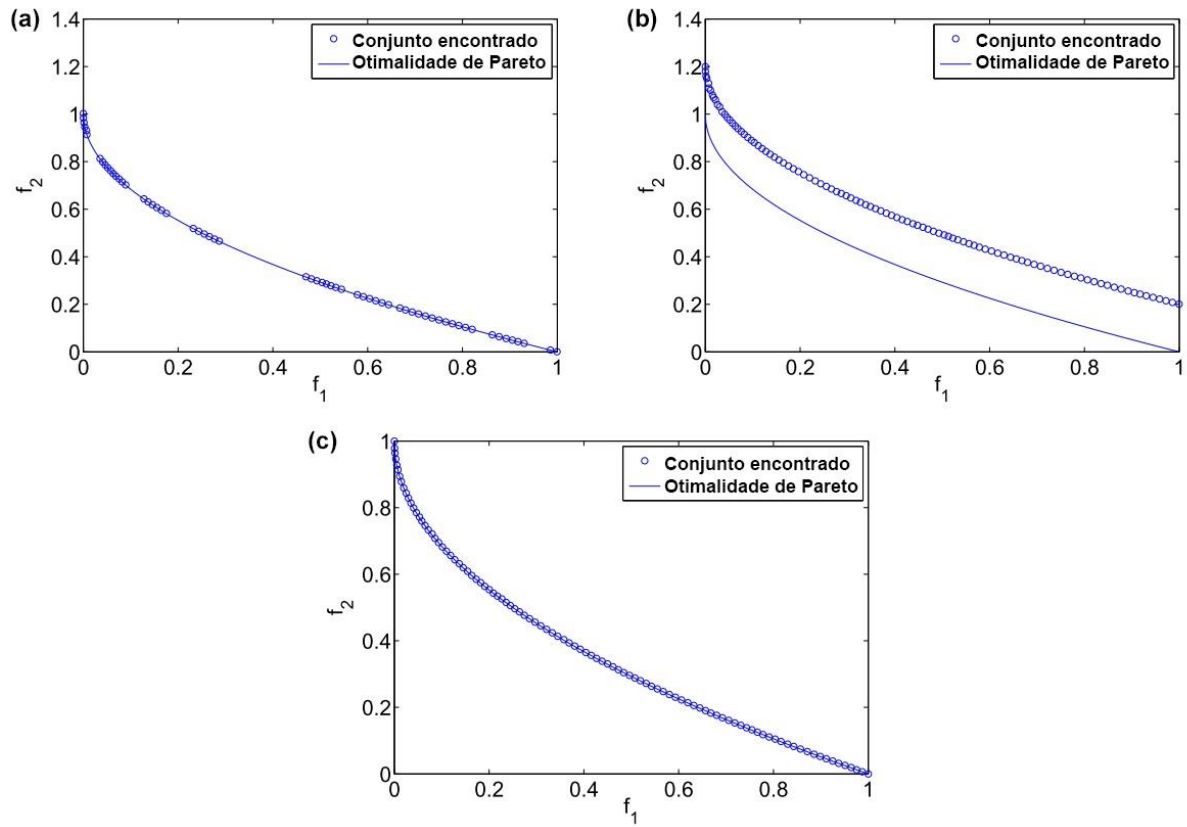


Figura 9 - Representação do espalhamento (a), otimalidade (b) e cenário ideal de soluções (c)
(DURILLO *et al.*, 2011)

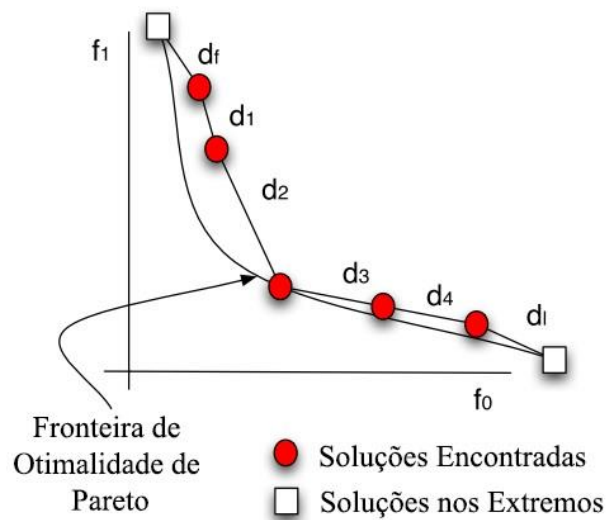


Figura 10 - Cálculo do espalhamento para dois objetivos (DURILLO *et al.*, 2009)

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N - 1) \cdot \bar{d}} \quad (1)$$

Na Equação 1, d_i é a distância Euclidiana entre pontos consecutivos da fronteira, enquanto $\bar{d}$ assume o valor médio entre estas distâncias. Finalmente, d_f e d_l são as distâncias Euclidianas para os pontos extremos do conjunto ótimo de Pareto (DURILLO *et al.*, 2009). Esta medida de desempenho recebe uma avaliação igual a zero para uma distribuição ideal, significando um espalhamento perfeito das soluções da fronteira. As fronteiras de Pareto encontradas que obtiverem um valor de espalhamento menor em relação as demais são consideradas melhores.

Para obtenção de um valor ideal de espalhamento, a curva encontrada deve satisfazer duas condições (DEB, 2001): (i) que os valores para d_f e d_l assumam um valor igual a zero, ou seja, a posição do primeiro e último pontos da curva de Pareto sejam exatamente iguais aos valores extremos da Fronteira de Pareto; e (ii) que a curva avaliada possua um espaçamento entre dois pontos consecutivos sempre com a mesma distância. Neste caso, a distância entre dois pontos consecutivos, definido por d_i , será sempre igual ao valor médio das distâncias, definido por $\bar{d}$.

Outro indicador de qualidade é chamado de **Hipervolume**, o qual considera a convergência do conjunto de soluções na fronteira atual e a diversidade das soluções definidas nesta fronteira (ZITZLER; THIELE, 1999). Esta medida calcula, dentro do espaço de objetivos, o volume dado pela cobertura dos elementos presentes na fronteira de soluções não-dominadas $Q = \{A, B, C, D, E\}$, ilustrada na Figura 11 na forma de uma linha descontínua. Para cada solução $i \in Q$, um hipercubo v_i é construído com um ponto de referência W e a solução i como as arestas diagonais do hipercubo. O ponto de referência W pode ser encontrado ao construir um vetor contendo pontos com os piores valores de avaliação para os objetivos do dado problema. Após este cálculo, a Equação 2 obtém o valor final do hipervolume através da união de todos os hipercubos.

Uma fronteira de Pareto com um valor de hipervolume maior que o de outra fronteira pode significar duas coisas: algumas soluções da fronteira com valor HV maior dominam as soluções da fronteira com HV menor, ou então as soluções de HV maior possuem uma melhor distribuição de pontos ao longo da fronteira. Assim, é desejável obter valores altos de hipervolume para qualquer problema de otimização multiobjetivo (DURILLO *et al.*, 2009).

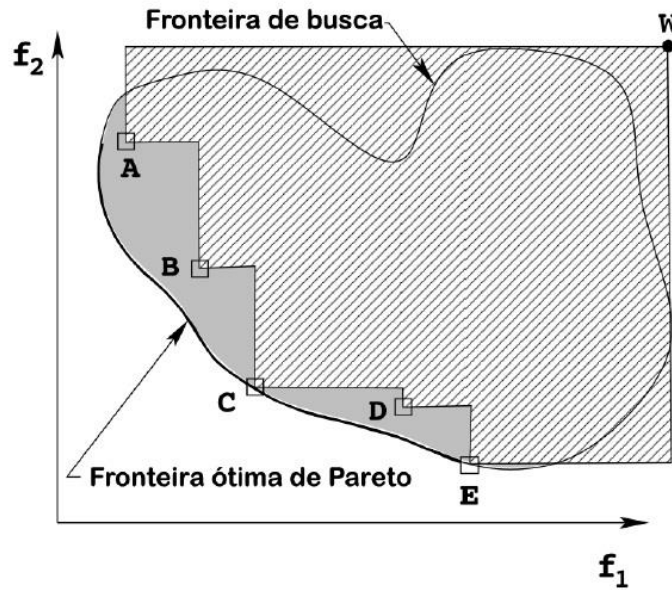


Figura 11 - Hipervolume delimitado por um conjunto de soluções não-dominadas (DEB, 2001)

$$HV = volume \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (2)$$

É possível normalizar o indicador hipervolume em valores no intervalo $[0, 1]$. Esta formulação é conhecida como hipervolume normalizado e é dada pela Equação 3, sendo resultante da divisão do valor de hipervolume obtido para a solução Q pelo valor de hipervolume da fronteira de referência P^* . Esta fronteira de referência pode, até mesmo, ser o conjunto ótimo de Pareto, caso seja conhecido (DEB, 2001).

$$HVR = \frac{HV(Q)}{HV(P^*)} \quad (3)$$

Todos os indicadores de qualidade descritos nesta seção necessitam do conhecimento a respeito da fronteira ótima de Pareto. Devido à complexidade computacional dos problemas a serem resolvidos, não é uma tarefa trivial obter esta fronteira de Pareto. Sendo assim, comumente é utilizada uma fronteira de referência, composta por todas as soluções não-dominadas obtidas por execuções anteriores de algoritmos de otimização multiobjetivo, formando um conjunto de “melhores soluções conhecidas até o momento” (DURILLO *et al.*, 2011).

2.5. Metaheurísticas

Em problemas de otimização mais simples pode ser usado um algoritmo que teste todas as possíveis soluções para o problema até encontrar o melhor resultado, o que é chamado de

busca exaustiva. Só assim, há a garantia de encontrar a melhor solução para o problema (**solução ótima**). Porém, dependendo do tamanho do problema, é computacionalmente inviável realizar a exploração de todas as possibilidades do **espaço de busca**. Sendo assim, há a necessidade do uso de técnicas de busca que encontrem boas soluções em um tempo de execução viável, apesar de não explorarem todo o espaço de busca (SIVANANDAM; DEEPA, 2008).

Nesta direção, as metaheurísticas são amplamente utilizadas, sendo definidas como estratégias que guiam o processo de busca, cujo objetivo é explorar eficientemente o espaço de busca para encontrar soluções ótimas (idealmente) ou próximas da ótima, devendo incorporar mecanismos para evitar que fiquem presas em áreas confinadas do espaço de busca (BLUM; ROLI, 2003). Para isto, aplicam heurísticas subordinadas, as quais têm que ser modeladas para um problema específico (MARTINS; RIBEIRO, 2006). Já heurísticas são técnicas inspiradas em processos intuitivos que procuram boas soluções a um custo computacional aceitável, sem garantir sua otimalidade, bem como sem garantir conhecer o quão próximo ela está da solução ótima (SOUZA, 2008).

Segundo Amaral e Elias (2019), dentre as metaheurísticas, destacam-se os Algoritmos Evolucionários Multiobjetivos (*Multi-Objective Evolutionary Algorithms* - MOEAs), como por exemplo: NSGA-II - *Non-dominated Sorting Genetic Algorithm-II* (DEB *et al.*, 2002), SPEA2 - *Strength Pareto Evolutionary Algorithm 2* (ZITZLER; LAUMANN; THIELE, 2001) e MOCeII - *MultiObjective Cellular genetic algorithm* (NEBRO *et al.*, 2009). Estes três algoritmos utilizam os conceitos de algoritmos genéticos e foram modificados para resolver problemas multiobjetivos. Por isto, inicialmente serão apresentados conceitos de algoritmos genéticos, seguidos dos conceitos destas três metaheurísticas consideradas neste trabalho.

2.5.1. Algoritmos genéticos

Os algoritmos genéticos (AG) são metaheurísticas baseadas na teoria da evolução das espécies iniciada por Charles Darwin. Segundo esta teoria, na seleção natural, os indivíduos que possuem características mais favoráveis de adaptação ao ambiente em que estão inseridos são os que possuem mais chances de sobreviver e, conseqüentemente, reproduzir. Já que no processo de reprodução, uma grande parte das características dos pais são repassadas para os filhos, indivíduos que se reproduzem mais tendem a propagar suas características para as gerações posteriores (LUCAS, 2002). Vale salientar que, originalmente, o AG foi proposto para problemas mono-objetivo.

Os AG são caracterizados por sua natureza estocástica, ou seja, não são determinísticos. Esta aleatoriedade está presente na geração da população inicial, na seleção de indivíduos para cruzamento e mutação. Além disso, os operadores de cruzamento incorporam características de busca local, enquanto os operadores de mutação incorporam características de busca global (AMARAL, 2017). Para possibilitar um melhor entendimento, os seguintes componentes do AG serão explicitados, sendo algumas definições herdadas da biologia:

- Indivíduos: codificam possíveis soluções para o problema tratado, sendo através de sua manipulação (processo evolutivo) que as respostas podem ser encontradas;
- Gene: cada gene codifica uma propriedade do indivíduo;
- Cromossomo: conjunto de genes que representa uma solução;
- População: conjunto de soluções que estão disponíveis para serem testadas;
- Representação: se referem a representação destes problemas de maneira que os AG possam trabalhar adequadamente sobre eles; e
- Grau de adaptação: representa o quão bem a solução avaliada atende ao problema proposto, chamado de função objetivo ou fitness.

O AG trabalha com um conjunto de soluções chamado população. A Figura 12 exemplifica uma execução comum do algoritmo genético. O fluxo é iniciado com a criação da população inicial, seguido pela fase de seleção, na qual são escolhidos os pais que participarão do processo de reprodução. Para a fase de seleção é necessário que os indivíduos sejam avaliados em relação aos valores obtidos nas funções objetivo do problema (*fitness*). O fluxo segue para a fase de cruzamento, seguida pela fase de mutação, que são alterações em pequenas parcelas das características destes indivíduos, pois, em alguns casos, pode incorporar características desejáveis não contidas nos pais. Após estas fases, é necessário novamente calcular o valor da função objetivo de todos os filhos. Para a etapa seguinte, conforme a Figura 12, é realizada a substituição da população anterior a partir dos indivíduos existentes e dos filhos gerados, com aplicação de uma estratégia chamada elitismo para manter a população de melhores indivíduos para próxima geração, podendo ser completada através de uma nova seleção, permitindo o desenvolvimento de indivíduos recém-criados que não estejam na população elite. Por fim, na fase de teste é verificado se o algoritmo satisfaz o critério de parada, caso não, nesse momento haverá um novo ciclo de execução, ao contrário, caso sim, a população final é retornada como resultado da execução do AG. Dentre as fases citadas na execução geral do ciclo do AG, algumas necessitam ser abordadas com mais detalhes, pois

utilizam técnicas e conceitos específicos. Estas fases são: seleção, cruzamento, mutação e substituição.

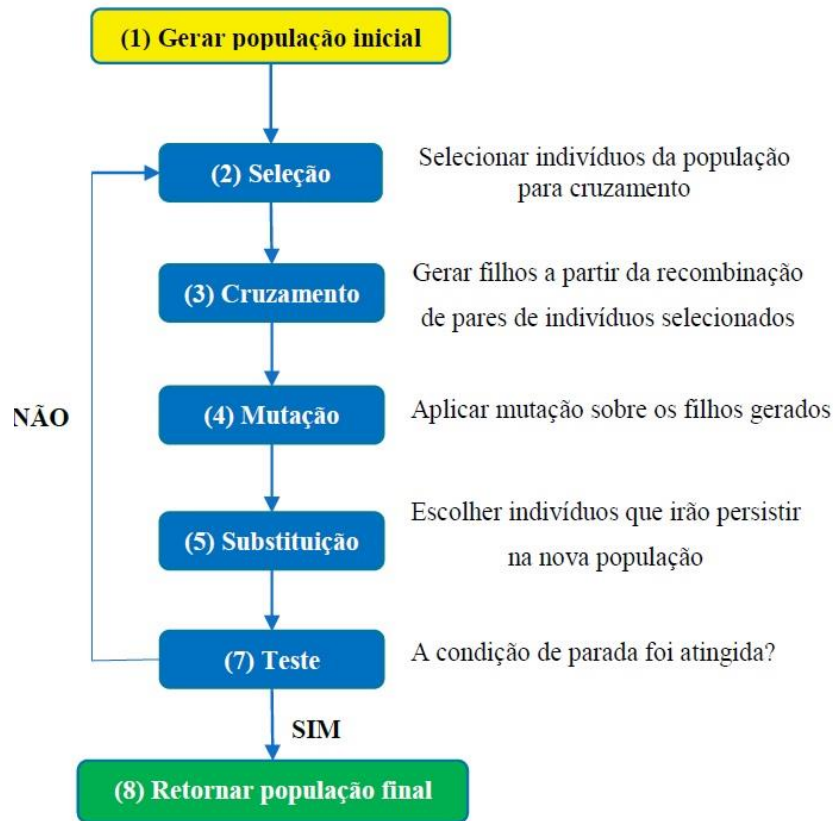


Figura 12 - Ciclo de execução de um algoritmo genético (AMARAL,2017)

No tocante à seleção, sua função consiste na estratégia para escolher os indivíduos da população atual que participarão da fase de reprodução, isto é, selecionar os indivíduos (pais) que vão gerar os novos indivíduos (filhos) para compor a próxima geração da população. Esta escolha deve ser feita de tal forma que os membros da população com maiores chances de reprodução sejam aqueles que apresentam um valor da função objetivo mais adequado (FILITTO, 2008). Dentre os vários tipos de técnicas de seleção pode-se destacar: seleção por torneio, seleção por roleta, e seleção por ordenação.

O método de seleção *por torneio* é o mais eficiente e possui maiores chances de alcançar a solução ótima (CHUDASAMA; SHAH; PANCHAL, 2011). Nesta técnica, dois ou mais indivíduos da população são aleatoriamente selecionados para participar do torneio. Neste torneio, o indivíduo que possuir o maior *fitness* será selecionado para reprodução. Estes torneios são repetidos até que o número de vencedores seja suficiente para cruzamento da nova população.

Adicionalmente, na seleção *por roleta*, a probabilidade de seleção é proporcional ao *fitness* de cada indivíduo. Inicialmente, o algoritmo computa a soma 'S' dos *fitness* de todos os indivíduos da população. Em seguida, os indivíduos são alocados em uma porção do intervalo (0, S) da roleta, proporcional aos seus respectivos valores de *fitness*. Em seguida, um número aleatório é gerado dentro deste intervalo e o indivíduo que ocupa a porção da roleta referente ao valor gerado é escolhido para posterior cruzamento. Assim, dá oportunidade também para os indivíduos com menos aptidões, de maneira que a probabilidade de seleção do indivíduo seja pela relação entre seu índice de aptidão comparado a aptidão dos outros indivíduos. Por exemplo, considerando três indivíduos, que possuem valores da função *fitness* iguais a 10, 30 e 60, as probabilidades de seleção destes indivíduos são 0.1, 0.3 e 0.6, respectivamente (LIMA, 2018).

Porém, diferentemente, a seleção *por ordenação* consiste em escolher os melhores indivíduos baseado no resultado da ordenação dos valores de suas funções de *fitness* (do maior para o menor). Nota-se que esse tipo de seleção tende a resultar em pouca diversidade, pois seleciona apenas os melhores indivíduos, e, conseqüentemente, faz com que o resultado da busca fique limitado a um ótimo local.

Após a fase de seleção no AG, a próxima etapa a ser executada é o cruzamento dos pais para geração dos filhos. Segundo Filitto (2008), o operador de cruzamento se baseia na combinação dos cromossomos dos pais para gerar os cromossomos dos filhos. Uma observação a ser feita é que o operador de cruzamento possui uma probabilidade associada, quando esta probabilidade atinge 0%, os filhos gerados serão idênticos aos pais, sem aplicar nenhuma recombinação. Em contrapartida, quando esta probabilidade atinge 100%, todos os novos indivíduos serão fruto desta recombinação. Segundo Sivanandam e Deepa (2008), é recomendável alocar uma porcentagem alta para criar uma diversidade de indivíduos, porém, ainda reservar uma pequena possibilidade de que novos indivíduos não sejam afetados por cruzamento e sejam cópias idênticas dos pais.

Existem diversos tipos de cruzamento, dos quais serão exemplificados alguns: cruzamento em um ponto, multiponto e uniforme. Para explicação dos conceitos é considerada, por simplificação, sem perda da generalidade, que a representação binária é utilizada.

O *cruzamento em um ponto* consiste em dividir os cromossomos selecionados em um ponto de sua cadeia de genes, que é escolhido aleatoriamente. Após isso, copiamos uma parte dos cromossomos de cada pai para gerar os cromossomos dos novos filhos (FILITTO, 2008), conforme ilustrado na Figura 13. Esta técnica possui a desvantagem de que, caso as características distintas de ambos os pais estejam concentradas em uma parte extrema do

cromossomo, existe uma menor probabilidade de que o ponto de corte seja localizado nesta posição, diminuindo a chance de que ambas as características distintas dos pais sejam herdadas pelos filhos.

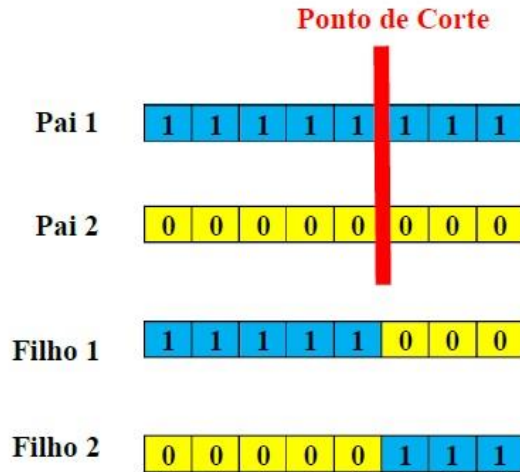


Figura 13 – Exemplo de cruzamento de ponto único (AMARAL, 2017)

Para atenuar a limitação do cruzamento de ponto único, foi concebida a técnica de *cruzamento multiponto*, ilustrado na Figura 14. Nesta técnica, o espaço de busca é investigado com uma profundidade maior, causando uma maior perturbação à solução, porém diminui sensivelmente o desempenho do algoritmo.

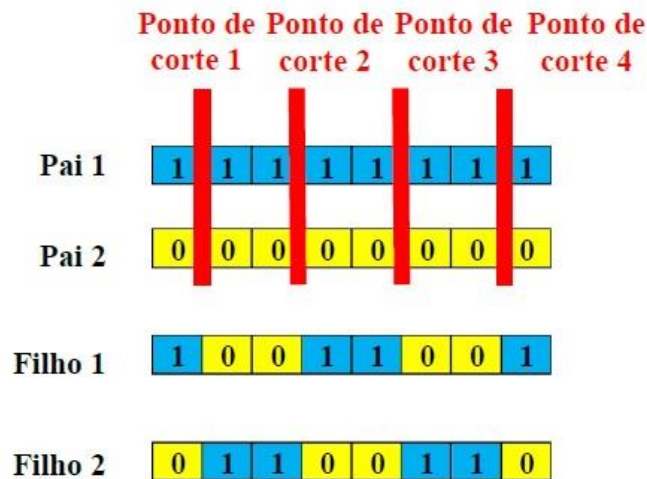


Figura 14 – Exemplo de cruzamento multipontos (AMARAL, 2017)

Na técnica de *cruzamento uniforme* (*uniform crossover*, UX), é criada uma máscara de bits do mesmo tamanho dos cromossomos selecionados para cruzamento. Esta máscara será preenchida com valores binários aleatórios. No processo de criação do cromossomo filho, todas as posições da máscara são analisadas. Quando o valor da posição é 1, o gene do primeiro pai referente a mesma posição da máscara é copiado para o filho. Caso o valor da máscara seja 0,

será copiado o gene do segundo pai. Esta lógica é então invertida para a geração do segundo filho. Esta técnica é ilustrada na Figura 15.

Pai 1	1	0	0	1	1	1	1	0
Pai 2	0	0	1	1	0	0	1	1
Máscara	0	1	0	1	0	1	1	1
Filho 1	0	0	1	1	0	1	1	0
Filho 2	1	0	0	1	1	0	1	1

Figura 15 – Exemplo de cruzamento uniforme (AMARAL, 2017)

Após o cruzamento, ocorre a mutação, com a finalidade de trazer diversidade à população ao longo das sucessivas gerações. Este operador é responsável por evitar que a solução se restrinja aos valores ótimos locais, auxiliando na perturbação da população, acrescentando características inexistentes na população presente.

Além da taxa de cruzamento, o operador de mutação também deve ser configurado com uma taxa de mutação, que sinaliza a probabilidade de atuação do operador sobre um dado cromossomo, a depender da técnica de mutação utilizada. Em relação à taxa de mutação, não existe um consenso. Para Sivanandam e Deepa (2008), este valor deve ser fixado em $1/L$, sendo ‘L’ o comprimento da *string* do cromossomo. Werner (WERNER, 2011), entretanto, defende que os melhores resultados são obtidos quando a taxa de mutação é fixada entre 1% a 10%.

Existem várias técnicas de mutação, entretanto, serão abordadas neste trabalho as seguintes técnicas: *flipping*, intercambiação e reversão. Destas, a técnica de *flipping*, consiste em testar a taxa de mutação a cada posição, e, caso o operador de mutação seja acionado, o bit será invertido. Esta técnica é ilustrada na Figura 16.

Cromossomo original	1	0	1	1	0	1	0	1
Ativação da mutação	1	0	0	0	1	0	0	0
Cromossomo modificado	0	0	1	1	1	1	0	1

Figura 16 – Exemplo de mutação por *flipping* de bits (AMARAL, 2017)

A *intercambiação* (*swap*) é a técnica mais adequada para problemas com codificação que admite valores não-binários. Ela consiste em selecionar duas posições aleatórias do cromossomo e inverter os valores destas posições. Já a *reversão* consiste em, caso o operador de mutação seja acionado, os valores das posições vizinhas são invertidos em conjunto com a posição escolhida. Isto é ilustrado na Figura 17. Uma possível desvantagem é a proximidade dos valores afetados sempre ser separada por apenas um bit intercalado.

Cromossomo original	1	0	1	1	0	1	0	1
Ativação da mutação	0	0	0	0	1	0	0	0
Cromossomo modificado	1	0	1	0	1	0	0	1

Figura 17 - Exemplo de mutação por reversão (AMARAL, 2017)

Após o cruzamento e a mutação, deve ser empregada alguma técnica responsável por inserir estes novos indivíduos na população, os quais serão armazenados para a próxima iteração do algoritmo (AMARAL, 2017). Dentre várias técnicas, uma estratégia muito utilizada nesta fase é memorizar os melhores indivíduos para não perder o valor máximo ou mínimo da função objetivo já alcançado. Esta estratégia é chamada de *elitismo*. Nela, o restante dos indivíduos é substituído usando qualquer outra técnica existente, por exemplo adotando a seleção por torneio. Outro exemplo de técnica de substituição é chamado de geracional, sendo a mais simples, na qual N pais geram N filhos e todos estes filhos irão compor a nova população. Isto ocasiona a perda de boas soluções encontradas em iterações passadas. Esta técnica reproduz fielmente o comportamento biológico das espécies, mas é raramente usada.

Todo o processo do AG é repetido até que uma condição de parada seja verdadeira e, no final da execução do algoritmo, deve ser gerado um bom conjunto de soluções. Em geral, a condição de parada pode ser definida por um determinado número total de iterações. Mas, opcionalmente, outra condição de parada pode ser um dado número de iterações sucessivas sem melhoria significativa das soluções (LIMA, 2018).

2.5.2. NSGA-II

A formulação clássica do algoritmo genético não é adequada para resolver problemas com mais de um objetivo. Portanto o AG precisou ser modificado para atender problemas multiobjetivos. Dentre os algoritmos que são baseados nos conceitos dos algoritmos genéticos e que foram modificados para lidar com múltiplos objetivos, um dos primeiros algoritmos

evolucionários foi o NSGA (*Non-dominated Sorting Genetic Algorithm*). O algoritmo NSGA-II (DEB *et al.*, 2002) surgiu da necessidade de melhorar algumas limitações presentes no NSGA original (SRINIVAS e DEB, 1995). Existe ainda a terceira versão do NSGA, o NSGA-III (DEB; JAIN, 2014), especialmente desenvolvida para lidar com quatro ou mais objetivos. Em relação aos problemas biobjetivos, o NSGA-III não foi adequadamente avaliado, tornando seu desempenho questionável e discutível (WANGSOM; BOUVRY; LAVANGNANANDA, 2018) (SEADA; DEB, 2014). Sendo assim, este trabalho abordará os conceitos referentes ao NSGA-II, pois atende as especificidades do problema avaliado, que é biobjetivo, e é o principal algoritmo utilizado nesta classe de problemas (COLANZI, 2013).

O ciclo do NSGA-II possui os seguintes passos: (i) Geração da população inicial; (ii) atribuição de dominância dos cromossomos conforme a regra de dominância de Pareto; (iii) separação dos cromossomos em fronteiras, nas quais as melhores soluções são alocadas no primeiro *front*, as soluções que são dominadas pelas soluções do primeiro *front* são classificadas no segundo *front* e assim sucessivamente; (iv) classificação de espalhamento: sendo atribuídos valores de distância para cada cromossomo em relação aos seus vizinhos, a fim de que os cromossomos com maior valor de distância possuam a melhor avaliação. Devido a simplicidade do passo inicial, já que a geração da população é um passo trivial, somente os passos seguintes serão abordados a seguir.

No primeiro passo, que é de atribuição de dominância, chamado de “*fast non-dominated sort*”, para cada indivíduo p da população P , calcula-se um valor n_p , que representa o número de soluções que dominam este elemento p . Também deve ser formado um subconjunto S_p , que indica quais elementos da população são dominados pelo indivíduo p . Este passo inicial tem complexidade $O(MN^2)$, o que define a maior complexidade do algoritmo. Conclui-se que, ao final deste passo de atribuição de dominância, os indivíduos que não são dominados por nenhuma outra solução possuirão o valor n_p igual a zero. Portanto, no passo de separação em fronteiras, este conjunto de valores são classificados no primeiro *front* da solução. Posteriormente, os valores n_p de cada elemento de S_p para todos os indivíduos do primeiro *front* devem ser decrementados em 1, e, caso algum deles atinja o valor zero, fará parte do segundo *front* da solução. Este procedimento é repetido até que todos os valores sejam classificados em algum *front*. O Algoritmo 1 apresenta este procedimento.

Em seguida, o segundo passo é a classificação do espalhamento, que é calculado para qualquer ponto em relação aos pontos vizinhos, sendo a distância dada pela média normalizada com base em todos os m objetivos do problema. A região final que representará a soma total das distâncias entre pontos vizinhos é denominada *cubóide*. Uma observação a ser feita é que

os indivíduos que se encontram em pontos extremos da região de busca recebem um valor de distância infinito, de forma a tornar desejável uma otimização do espalhamento, mesmo que estas soluções nos extremos não possuam um valor de *fitness* razoável em comparação com os demais pontos encontrados. A Figura 18 exemplifica um caso de dois objetivos f_1 e f_2 e o Algoritmo 2 descreve o cálculo do espalhamento para cada cromossomo dentro de um conjunto não-dominado L .

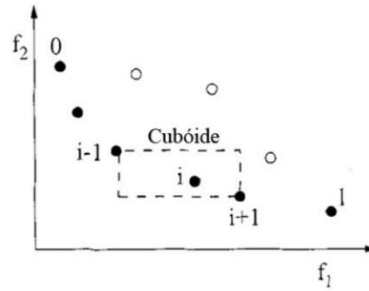


Figura 18 - Cubóide do ponto i para dois objetivos (AMARAL, 2017)

Algoritmo 1 - Atribuição de dominância e classificação em fronts do NSGA-II (AMARAL, 2017)

01	Fast-non-dominated-sort(P)
02	for each $p \in P$
03	$S_p = \emptyset, n_p = 0$
04	for each $q \in P$
05	if $(p < q)$ then //se p domina q
06	$S_p = S_p \cup \{q\}$ //adicionar q ao conjunto de soluções dominadas por p (S_p)
07	else if $(q < p)$ then //se p for dominado por q , então o contador n_p deve ser incrementado
08	$n_p = n_p + 1$
09	if $n_p = 0$ then //ao final do loop for, todos os indivíduos p que possuírem valor $n_p = 0$ pertencem ao primeiro front
10	$p_{rank} = 1$
11	$F_1 = F_1 \cup \{p\}$
12	$i = 1$ //inicializar o contador de fronts
13	while $F_i \neq \emptyset$
14	$Q = \emptyset$ //usado para armazenar elementos do próximo front
15	for each $p \in F_i$
16	for each $q \in S_p$
17	$n_q = n_q - 1$
18	if $n_q = 0$ then //q pertence ao próximo front
19	$q_{rank} = i + 1$
20	$Q = Q \cup \{q\}$
21	$i = i + 1$
22	$F_i = Q$

Na linha 08 do Algoritmo 2, o valor $L[i].m$ é referente ao m -ésimo valor da função objetivo do i -ésimo cromossomo no conjunto L . Então, a equação calcula a diferença entre os valores de função de cromossomos vizinhos. Este valor é dividido pela diferença dos valores máximo e mínimo de função para o objetivo m , que correspondem aos valores presentes nos cromossomos 1 e l para o conjunto L . Por fim, este valor calculado é acrescido ao valor de distância do cromossomo i em relação aos demais objetivos já calculados, a fim de obter o valor de distância final. Segundo DEB *et al.* (2002), este procedimento possui uma complexidade na ordem de $O(MN\log N)$, novamente admitindo M como sendo o número de objetivos, enquanto N é o tamanho da população.

Algoritmo 2 - Cálculo da distância do NSGA-II (AMARAL, 2017)

01	Crowding-distance-assignment(L)	
02	$l = L $	//o número de soluções em L
03	for each i , set $L[i]_{distance} = 0$	//inicializar o cálculo de distância
04	for each objective m	
05	$L = \text{sort}(L, m)$	//ordenar os elementos de L de acordo com o valor do objetivo m
06	$L[1]_{distance} = L[l]_{distance} = \infty$	//atribuir valor infinito aos pontos extremos
07	for $i = 2$ to $(l - 2)$	
08	$L[i]_{distance} = L[i]_{distance} + \frac{L[i+1].m - L[i-1].m}{f_m^{max} - f_m^{min}}$	

Nota-se que o operador de distância não é calculado para todos os elementos da população, mas apenas para elementos de um mesmo *front*, classificados na etapa anterior. Isto ocorre, pois, dada a característica elitista do algoritmo, os elementos classificados nos primeiros *fronts* são naturalmente aceitos na população futura. Após sucessivas iterações, a solução vai convergindo e todos os elementos irão progressivamente participar dos primeiros *fronts*. Por isso o operador de distância será utilizado como critério de desempate para gerar o novo conjunto de pais para criar o conjunto de filhos da próxima geração.

Adicionalmente, para substituir a população, o operador de comparação de densidade ($<_n$) é definido a fim de selecionar os elementos presentes para a próxima iteração, recebendo

como entrada o valor do *front* e a distância de espalhamento. Uma dada solução i é melhor que outra solução j ($i \prec_{nj}$) se: (i) $\text{front}_i < \text{front}_j$; ou (ii) $(\text{front}_i = \text{front}_j) \wedge (\text{distância}_i > \text{distância}_j)$.

Após todos os conceitos necessários definidos, o Algoritmo 3 descreve o ciclo de execução do NSGA-II, composto dos seguintes passos: (i) união dos conjuntos de pais e filhos; (ii) atribuição da dominância; (iii) execução do cálculo da distância e inclusão dos fronts na nova população de pais até que o tamanho dos *fronts* exceda o tamanho da população (linhas 04 a 07); (iv) ordena os elementos do último *front*, a fim de preencher a população de pais; (v) escolhe os primeiros elementos do último *front* para preencher o restante da solução; e (vi) utiliza os operadores de seleção, cruzamento e mutação para gerar a nova população.

Algoritmo 3 - Pseudocódigo do ciclo principal do NSGA-II (AMARAL, 2017)

01	$R_t = P_t \cup Q_t$	//criar um conjunto único de pais e filhos de tamanho 2N
02	$F = \text{fast-non-dominated-sort}(R_t)$	//atribuição de dominância
03	$P_{t+1} = \emptyset$ and $i = 1$	
04	until $ P_{t+1} + F_i \leq N$	//incluir fronts na nova população de pais até que o tamanho do fronts selecionados até o momento exceda o tamanho N da população
05	$\text{crowding-distance-assignment}(F_i)$	//cálculo de distância
06	$P_{t+1} = P_{t+1} \cup F_i$	//incluir o front na nova população de pais
07	$i = i + 1$	//incrementar e testar o próximo front i+1
08	$\text{sort}(F_i, \prec_n)$	//ordenar os elementos do último front i a fim de preencher exatos N elementos na população de pais
09	$P_{t+1} = P_{t+1} \cup F_i[1:(N - P_{t+1})]$	//escolha os primeiros (N - P _{t+1}) elementos de F _i para preencher o restante da solução
10	$Q_{t+1} = \text{make-new-pop}(P_{t+1})$	//utilizar os operadores de seleção, cruzamento e mutação para gerar a nova população Q _{t+1}
11	$t = t + 1$	//incrementar o contador de geração

2.5.3. SPEA2

Assim como o NSGA-II, o SPEA2 (ZITZLER, LAUMANN e THIELE, 2001) representa uma modificação em relação ao algoritmo original proposto em 1999 (ZITZLER e THIELE, 1999). O SPEA2 - *Strength Pareto Evolutionary Algorithm* (ZITZLER, LAUMANN e THIELE, 2001) é considerado similar ao NSGA-II (COLANZI, 2013), sendo a sua principal característica o uso de um **arquivo** para implementação do elitismo. Neste arquivo será armazenado um conjunto de soluções, separado da população, que contém todos

os cromossomos não-dominados em relação aos demais. Para manter o arquivo, o algoritmo elimina soluções repetidas e indivíduos dominados por algum outro membro da população. Então, ao final da execução, o arquivo provê todas as soluções não-dominadas que foram encontradas.

O fluxo principal do SPEA2 é definido, em linhas gerais, no Algoritmo 4. Nele, após a definição das variáveis, é realizada a inicialização da população inicial, do arquivo e da variável de contagem das iterações, seguida do cálculo dos valores de *fitness* dos indivíduos da população e do arquivo. Posteriormente, é feita a seleção ambiental, copiando os elementos não dominados da população e arquivo correntes para uma nova versão do arquivo. Este fluxo é repetido até atender algum critério de parada.

Algoritmo 4 - Etapas da execução do SPEA 2 (AMARAL, 2017)

01	Dados de Entrada	N , tamanho da população
02		$\bar{N}$, tamanho do arquivo
03		T , número de iterações
04	Dados de Saída	A , conjunto de soluções não-dominadas
05	Inicialização: Gere uma população inicial $P_0 = \emptyset$, um arquivo $\bar{P}_0 = \emptyset$, e defina a iteração inicial $t = 0$	
06	Atribuição de fitness: o valor de fitness dos indivíduos do arquivo $\bar{P}_t$ e da população regular P_t são calculados	
07	seleção ambiental: copie todos os elementos não dominados em P_t e $\bar{P}_t$ para $\bar{P}_{t+1}$. Se $\bar{P}_{t+1}$ exceder $\bar{N}$, então reduza $\bar{P}_{t+1}$ por meio de um operador de truncamento. Caso contrário, selecione elementos dominados da população P_t e do arquivo $\bar{P}_t$ para preencher o arquivo.	
08	Encerramento: se $t \geq T$, ou se algum outro critério de parada for atingido, então retorne os valores de $\bar{P}_{t+1}$ como o arquivo final A e encerre a execução do algoritmo.	

Adicionalmente aos passos descritos no fluxo principal do SPEA2, o operador de truncamento dá uma atenção especial para o valor de densidade. Caso o tamanho do novo arquivo ultrapasse o tamanho máximo definido na entrada, os elementos com as menores distâncias para os vizinhos são removidos. Caso os elementos excedentes possuam uma distância similar, o valor de distância é alterado para se relacionar apenas com o vizinho mais próximo. Caso o empate persista, a comparação é realizada em relação ao segundo vizinho mais próximo, e assim sucessivamente até que um valor diferente seja encontrado.

2.5.4. MOCeII

Os algoritmos genéticos abordados até o momento consideram a população como uma entidade singular, na qual os operadores estocásticos são aplicados de maneira uniforme sobre este conjunto completo. Esta técnica é denominada de *panmixia*. Porém, uma das vantagens dos GAs é a capacidade de paralelização, o que deu origem aos algoritmos genéticos distribuídos, capazes de trabalhar em diferentes regiões do espaço de busca, compartilhando soluções que possuem alguma relação com o espaço de busca investigado por outra instância do algoritmo. Esta classe de GA é chamada de *estratégias evolucionárias estruturadas* e é capaz de fornecer uma melhor amostragem do espaço de busca, já que o espaço é dividido em uma quantidade considerável de sub-regiões em comparação com um algoritmo genético clássico que trata o espaço de busca completo como uma só entidade no modo de *panmixia* (NEBRO *et al.*, 2009).

No caso mais extremo de segmentação do espaço de busca, são utilizados os *algoritmos genéticos celulares* (cGAs). Neste cenário, os cromossomos estão restritos a interagir apenas com os indivíduos existentes na vizinhança durante o processo de reprodução. Esta sobreposição de microrregiões vizinhas favorece a diversificação das soluções, que, no contexto dos algoritmos multiobjetivos, é representada pelo espalhamento das soluções, por conta de uma difusão vagarosa de boas soluções entre os membros da população. Em contrapartida, a intensa atividade entre cromossomos vizinhos auxilia nas buscas locais, aproximando-se do valor ótimo local para cada vizinhança estabelecida. Uma vantagem apontada por NEBRO *et al.* (2009) no uso de algoritmos genéticos celulares é o mecanismo implícito de migração, no qual, caso seja encontrado um conjunto reduzido de soluções que possuam um *fitness* muito mais elevado em comparação com o restante da população atual, isto não irá influenciar significativamente o comportamento geral do algoritmo, evitando uma convergência prematura. Já que as vizinhanças são sobrepostas, em um processo de seleção de pais para reprodução, uma solução muito melhor que as outras pode afetar, no pior caso, apenas sua própria vizinhança e as vizinhanças adjacentes, não afetando então o processo de seleção de indivíduos que não façam parte destas vizinhanças afetadas.

Os algoritmos genéticos celulares foram propostos inicialmente para serem empregados em máquinas com um alto grau de paralelismo, apesar de que o modelo em si é perfeitamente adaptável para o uso de máquinas com apenas um único núcleo de processamento. Além disso, a vizinhança é definida em razão da proximidade das soluções candidatas encontradas até o momento, o que não quer dizer necessariamente que um problema de otimização deve ser

quebrado em partes distintas que são executadas por processadores diferentes. Neste contexto, o MOCeII - *MultiObjective Cellular genetic algorithm* (NEBRO *et al.*, 2009) é um algoritmo multiobjetivo que emprega a distribuição celular da região de busca, ilustrada na Figura 19. Deste modo, o conceito de vizinhança é intensivamente usado, representando que o indivíduo deve cooperar com a vizinhança mais próxima em um *loop* de reprodução.

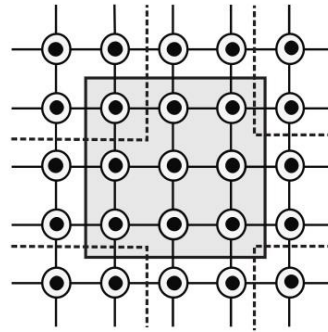


Figura 19 - Distribuição celular do espaço de busca (NEBRO *et al.*, 2009)

De maneira semelhante ao SPEA2, o MOCeII também emprega um **arquivo** com objetivo de guardar as melhores soluções encontradas de maneira a implementar o elitismo, enquanto a população regular é inteiramente renovada a cada ciclo de execução. Porém, diferentemente do SPEA2, o MOCeII possui um mecanismo de feedback, em que algumas soluções existentes na população são trocadas aleatoriamente por soluções do arquivo.

Um algoritmo genético celular é definido de maneira que sua população seja estruturada em grade com d dimensões e a vizinhança definida sobre esta população. O algoritmo avalia iterativamente cada indivíduo na grade de vizinhança, então, os pais são escolhidos entre estes vizinhos, de acordo com a técnica de seleção utilizada. Após a seleção, os operadores de cruzamento e mutação são acionados de acordo com as duas respectivas probabilidades. Em seguida, o algoritmo computa o *fitness* de cada indivíduo gerado e os coloca na posição da população em que os pais se encontram em uma população auxiliar. A população auxiliar se torna a nova população que será utilizada na próxima geração. Finalmente, a condição de parada é testada e, caso não seja satisfeita, uma nova etapa de seleção de indivíduos em cada grade para compor um novo conjunto de pais é reiniciado. Caso a condição de parada seja satisfeita, de maneira similar ao SPEA2, o arquivo que armazena as melhores soluções não-dominadas conhecidas até o momento é considerado a saída do algoritmo.

O Algoritmo 5 mostra o fluxo de execução principal do MOCeII. Este fluxo é executado até atingir a condição de parada, consistindo em percorrer toda a população, iniciando pela obtenção da vizinhança de cada indivíduo. Em seguida, os pais são selecionados. Posteriormente, nas linhas 07 e 08, os operadores de cruzamento e mutação são aplicados,

respectivamente. Após o cálculo do *fitness* de cada filho, ambos serão inseridos na mesma posição dos pais na “população auxiliar” (linha 10), que representa a população que será utilizada na próxima iteração do algoritmo, compondo a população regular (linha 12). Para decidir se os cromossomos devem ser inseridos no arquivo (linha 11), o MOCell utiliza um operador de comparação de densidade ($\prec_n$) semelhante ao descrito no NSGA-II.

Adicionalmente, o procedimento de *feedback* é utilizado com objetivo de selecionar aleatoriamente um número fixo de indivíduos dentre o arquivo e inseri-los na nova população (linha 13). Este procedimento é uma inclusão importante para melhorar a exploração do espaço de busca, pois evita uma convergência prematura da solução, já que, na próxima iteração, é possível que as soluções consideradas não-dominadas passem a ser dominadas por uma nova solução e não retornem ao arquivo na iteração seguinte (DORRONSORO; ALBA, 2008).

Algoritmo 5 – Loop principal do MOCell (AMARAL, 2017)

01	Proc Inicio(mocell)	//Inicializar parâmetros do algoritmo
02	Front-Pareto = Criar-Front()	//Criar front vazio
03	while not CondicaoParada() do	
04	for each indivíduo, 1 to mocell.tamanho_população do	
05	n_lista := Retornar_vizinhança(mocell,posição(indivíduo))	
06	pais := Selecao(n_lista)	
07	filhos := Crossover(propCrossover,pais)	
08	filhos := Mutação(probMutacao,filhos)	
09	Avaliar_fitness(filhos)	
10	Inserir(posição(indivíduo),filhos,mocell,pop_aux)	
11	Inserir_Front-Pareto(indivíduo)	//inserir no arquivo
12	mocell.população := pop_aux;	
13	Mocell.população := Feedback(mocell,Front_Pareto)	

2.6. Considerações finais

Neste capítulo foram apresentados os conceitos fundamentais sobre os processos de software e os profissionais envolvidos no seu desenvolvimento. Em seguida, foram debatidos conceitos da área de psicologia referentes a alguns modelos de personalidade comumente utilizados para entender e categorizar o comportamento das pessoas: MBTI e FFM.

Posteriormente a área de gerenciamento e seleção de recursos humanos, bem como os conceitos necessários para o entendimento, foram discutidos. Nesta área, foi apresentado como é geralmente realizada a identificação de competências dos profissionais e a seleção dos

mesmos, bem como as dificuldades e problemas inseridos neste contexto, assim demonstrando a motivação para a criação da abordagem proposta. Neste sentido, destaca-se que foram listados os requisitos técnicos e não-técnicos envolvidos na seleção de pessoas para atividades do processo de desenvolvimento de software.

Adicionalmente, os fundamentos relacionados à SBSE e às principais metaheurísticas adotadas para resolver problemas de otimização foram introduzidas. No caso particular deste trabalho, o problema de otimização é a seleção e alocação de pessoas. Inicialmente foram definidos os principais conceitos relativos aos algoritmos genéticos e a sua relação com a teoria da evolução das espécies. Em seguida, foi descrito como soluções são geradas e sucessivamente evoluem com a aplicação dos operadores de seleção, cruzamento, mutação e substituição. Na sequência, foi abordado como um algoritmo genético pode ser adaptado para resolução de problemas multiobjetivos, concluindo com uma breve descrição dos algoritmos NSGA-II, SPEA2 e MOCell.

3. ABORDAGEM PROPOSTA

Este capítulo apresenta a abordagem para formação de equipes tecnicamente qualificadas e psicologicamente adequadas para cada papel funcional do projeto de desenvolvimento de software proposta neste trabalho. Esta abordagem proporciona ao gerente de projetos uma ferramenta que o apoia na atividade de tomada de decisão de quais pessoas serão atribuídas aos papéis funcionais do ciclo de desenvolvimento de software, de modo a atender aos requisitos técnicos e não-técnicos dos papéis. O objetivo é montar equipes melhores que as escolhidas aleatoriamente ou com base apenas na experiência do tomador da decisão.

Por lidar com objetivos conflitantes, este trabalho propõe a utilização de uma abordagem multiobjetivo, na qual, ao final será gerada uma fronteira de soluções não-dominadas. Contudo, assim como em Santos (2014), a experiência do gerente deve ser levada em consideração, fornecendo meios para que o mesmo possa realizar os devidos ajustes para que a solução atenda as especificidades do projeto. Por se tratar de um problema de busca, serão utilizados conceitos, técnicas, algoritmos e estratégias de busca e métodos de inteligência computacional adotados na área de SBSE.

3.1. Visão geral da abordagem

Com o objetivo de gerar as recomendações de equipes, a abordagem proposta é organizada em três etapas, que podem ser vistas na Figura 20. As etapas são: avaliação do perfil técnico, avaliação do perfil de personalidade e avaliação integrada pessoas-papéis. Cada etapa possui entradas e saídas específicas. Como resultado final, a saída da abordagem é o conjunto de sugestões otimizadas de equipes, considerando os aspectos técnicos e não técnicos.

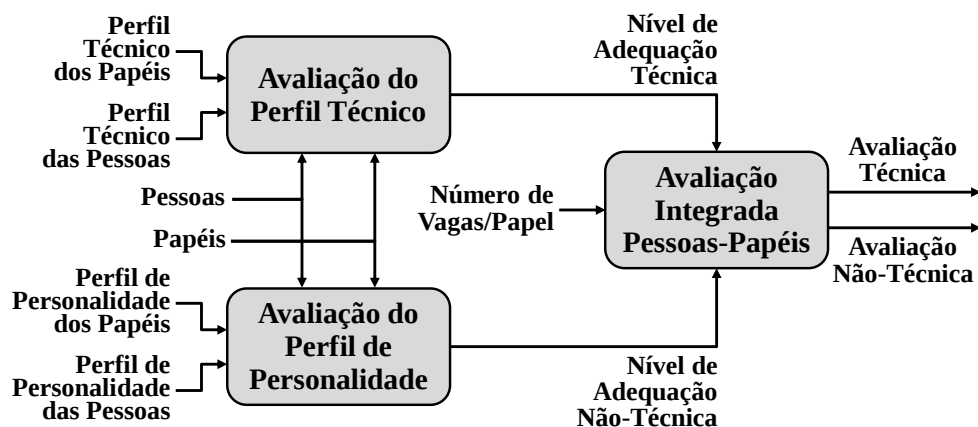


Figura 20 – Etapas da abordagem proposta

A etapa *Avaliação do Perfil de Personalidade* gera como saída o **nível de adequação não-técnica** das pessoas para os papéis. Para isto recebe como entrada os seguintes dados: o **perfil de personalidade das pessoas** e o **perfil de personalidade dos papéis**. Ambos são representados utilizando um dos modelos abordados neste trabalho (MBTI ou FFM), pois são os modelos mais usados para avaliar a personalidade dos profissionais de engenharia de software (GULATI; BHARDWAJ; SURI, 2015). Por um lado, os perfis de personalidade das pessoas representam os traços de personalidade dos profissionais candidatos. Por outro lado, os perfis de personalidade dos papéis representam as necessidades de traços de personalidade de cada papel do projeto. Os modelos MBTI e FFM são suportados pela abordagem proposta. O modelo a ser adotado é uma opção do gerente de projeto, desde que todas as pessoas e todos os papéis sejam representados usando a mesma tipologia.

Já a etapa *Avaliação do Perfil Técnico* gera como saída o **nível de adequação técnica** das pessoas para os papéis. Para isto considera como entrada o **perfil técnico dos papéis** e o **perfil técnico das pessoas**. Por um lado, os perfis técnicos dos papéis são os requisitos técnicos de cada papel do projeto necessários para o desempenho das atividades correspondentes. Por outro lado, os perfis técnicos das pessoas são as habilidades técnicas possuídas por cada profissional candidato, as quais são coletadas através da análise de um questionário identificado nesta abordagem. É importante salientar que são considerados como requisitos técnicos os conhecimentos sobre sistemas operacionais, plataformas, ferramentas, linguagens de programação, processos, entre outros domínios técnicos.

Neste ponto, as pessoas candidatas ao projeto de software estão avaliadas quanto ao nível de qualificação técnica e quanto à adequação psicológica necessária para atender as especificidades de cada papel. Portanto já foram coletadas as informações necessárias à próxima etapa que é a *Avaliação Integrada Pessoas-Papéis*. Esta etapa utiliza metaheurísticas para recomendar soluções de seleção e alocação de pessoas a papéis. Para isto, recebe como entrada as saídas das etapas anteriores (*Avaliação do Perfil Técnico* e *Avaliação do Perfil de Personalidade*), além do **número de vagas/papel**, que sinaliza a quantidade de pessoas necessárias para o desempenho de cada papel no projeto. Este último dado de entrada é fornecido pelo gerente de projeto, que dimensiona a quantidade necessária para cada projeto, dependendo das especificidades de tamanho, complexidade e orçamento do projeto.

Entretanto, esta combinação entre as avaliações técnicas e não-técnicas pode ser conflitante, pois um profissional pode ser tecnicamente mais capaz e não possuir os traços de personalidade apropriados que o papel requer, ou vice-versa (STYLIANOU; ANDREOU,

2012). Portanto, a natureza conflitante das adequações técnicas e não-técnicas deve ser tratada de maneira adequada para encontrar soluções apropriadas. Portanto, a abordagem proposta adota estratégias de problemas de otimização multiobjetivo, do inglês *Multiobjective Optimization Problems* (MOPs).

O artefato final da abordagem proposta será um conjunto de soluções não-dominadas, que será representada por meio de um gráfico bidimensional. Almeja-se que este conjunto de soluções possua uma boa aproximação da fronteira ótima de Pareto e com espalhamento uniforme, conforme explicado na seção 2.4.1.1. Em suma, na formulação multiobjetivo, várias soluções são recomendadas para o problema, cabendo ao tomador de decisão selecionar a que melhor se adeque às necessidades do projeto (AMARAL, 2017). Ou seja, a abordagem permite ao gerente de projetos escolher uma das equipes exibidas como opções para atender de forma otimizada às necessidades do projeto, ao invés de selecionar apenas a partir da sua percepção subjetiva e sua experiência, como é feito na maioria dos casos (ACUNA; JURISTO, 2003).

Na abordagem proposta, o conjunto de n pessoas candidatas ao projeto é representado por $PE = \{pe_1, pe_2, \dots, pe_n\}$, onde cada termo pe_i representa uma pessoa. Já o conjunto dos m papéis do projeto é representado por $PA = \{pa_1, pa_2, \dots, pa_m\}$, onde cada termo pa_j representa um papel funcional. Mais detalhes sobre cada uma das três etapas que compõe a abordagem proposta serão abordados nas seções seguintes.

3.2. Avaliação do Perfil de Personalidade

Esta etapa possui a finalidade de avaliar a adequabilidade não-técnica das pessoas candidatas em relação aos papéis. Na abordagem proposta, os aspectos não-técnicos considerados são os traços de personalidade das pessoas e dos papéis funcionais. Para isto, é necessário fornecer como entrada a esta etapa a classificação psicológica das pessoas candidatas e dos papéis funcionais adotando um dos modelos suportados pela abordagem (MBTI e FFM). Existem instrumentos disponíveis na literatura, amplamente utilizados para a realização destas classificações, alguns encontrados gratuitamente na Internet. As seções 2.2.1 e 2.2.2 exemplificam alguns deles. Entretanto, estes instrumentos não são o foco da abordagem aqui proposta, pois é considerado que os candidatos ao projeto serão submetidos a um destes instrumentos e as classificações de suas personalidades resultantes serão inseridas como informações de entrada para esta etapa.

Como já tratado em seções anteriores, cada papel do projeto possui traços de personalidade recomendados para a sua melhor execução, sendo chamados de requisitos não-técnicos. Na literatura, existem trabalhos (CAPRETZ; AHMED, 2010) (ACUNA; JURISTO; MORENO, 2006) (CUNHA; GREATHEAD, 2007) (REHMAN *et al.*, 2012) que propõem mapeamentos de profissões ou papéis funcionais da engenharia de software para traços de personalidades mais requeridos ou marcantes para desempenhar as respectivas atividades associadas. Com base nestes trabalhos, a abordagem aqui proposta adota como mapeamento padrão a classificação de Capretz e Ahmed (2010) e Rehman *et al.* (2012), debatidos na seção 2.3.1.1. Considerando que a presença e a importância dos papéis funcionais dependem das características do projeto de software e do processo de desenvolvimento (CAPRETZ; VARONA; RAZA, 2015), os referidos mapeamentos foram adotados por contemplar aqueles mais citados nos diversos modelos de processos de desenvolvimento (CAPRETZ, 2012).

É importante destacar que o gerente de projeto pode adotar diferentes mapeamentos de papéis funcionais a perfis MBTI ou FFM, podendo ser algum outro encontrado na literatura ou mesmo qualquer outro adaptado ou criado de acordo com as necessidades da organização ou projeto de software. Além disso, o gerente de projeto também pode adotar diferentes testes de personalidade, caso julgue mais adequado à organização ou projeto de software. Consequentemente, está fora do escopo da abordagem proposta a concepção de mapeamentos, bem como a elaboração de testes de personalidade.

3.2.1. Formulação da avaliação não-técnica

A abordagem proposta aceita como entrada tanto a personalidade dos indivíduos classificada utilizando o MBTI quanto o FFM. Porém o instrumento de análise deverá ser igual para todos os indivíduos candidatos, ou seja, uma vez que a organização escolha utilizar o MBTI, o sistema deverá considerar para todos os indivíduos candidatos os tipos de personalidade do MBTI, ou vice-versa. A análise de correspondência entre os requisitos requeridos pelos papéis e possuídos pelos indivíduos ocorrerá de formas distintas de acordo com o instrumento considerado, ambas as formulações serão detalhadas nas seções seguintes.

3.2.1.1. MBTI

Lembrando que o MBTI possui 4 dicotomias opostas (abordado na seção 2.2.1), o indivíduo que possuir as dicotomias correspondentes às requeridas pelo papel, pontuará positivamente. Ao contrário, o indivíduo que possuir as dicotomias opostas ao papel, pontuará negativamente. Para pares de dicotomia que não possuem relação com o papel, a relação entre possuídas e requeridas não ocorrerá. O conjunto de traços das dicotomias MBTI é representado por $TR_{MBTI} = \{E, I, S, N, T, F, J, P\}$.

Por um lado, conforme especificado na Equação 4, o *perfil de personalidade dos papéis* é denotado pelo conjunto P_{PA} , onde cada termo DR_j é um vetor de quatro pares de dicotomias MBTI, que conjuntamente indicam as dimensões bipolares requeridas pelo papel pa_j . Observe que cada polo dr_j^X pode assumir os valores 1, -1 ou 0, representando ser requerido, não requerido e indiferente, respectivamente. Adicionalmente, cada dimensão bipolar dr_j^X/dr_j^Y , deve sempre possuir valores complementares ou iguais a zero.

$$P_{PA} = \left\{ DR_j \left| \begin{array}{l} \exists pa_j \in PA \\ DR_j = [dr_j^E/dr_j^I \ dr_j^S/dr_j^N \ dr_j^T/dr_j^F \ dr_j^J/dr_j^P] \\ dr_j^X/dr_j^Y = \{1/-1, -1/1, 0/0\} \end{array} \right. \right\} \quad (4)$$

Por exemplo, considerando o papel de analista de sistemas (Tabela 1), cujo perfil MBTI é “E*F*”, o perfil de personalidade do referido papel é definido pelo vetor de dicotomias $[1/-1 \ 0/0 \ -1/1 \ 0/0]$. A representação de todos os papéis funcionais definidos na Tabela 1 é ilustrada na Figura 21.

	E	I	S	N	T	F	J	P
Analista de Sistemas	1	-1	0	0	-1	1	0	0
Projetista de <i>Software</i>	0	0	-1	1	1	-1	0	0
Programador	-1	1	1	-1	1	-1	0	0
Testador	0	0	1	-1	0	0	1	-1
Manutenedor	0	0	1	-1	0	0	-1	1

Figura 21 – Representação das dicotomias MBTI requeridas pelos papéis

Por outro lado, de acordo com a Equação 5, o *perfil de personalidade das pessoas* é representado pelo conjunto P_{PE} , onde cada termo DP_i é um vetor de quatro pares de dicotomias MBTI, que conjuntamente indicam as dimensões bipolares que caracterizam a pessoa pe_i . Veja

que cada polo dp_i^X pode assumir os valores 1 ou -1, caracterizando ou não o polo dominante, respectivamente. Além disso, cada dimensão bipolar dp_i^X/dp_i^Y , deve sempre ter valores complementares. Por exemplo, considerando uma determinada pessoa, cujo perfil MBTI é “ENTP”, o perfil de personalidade da pessoa é definido pelo vetor de dicotomias $[1/-1 \ -1/1 \ 1/-1 \ -1/1]$.

$$P_{PE} = \left\{ DP_i \left| \begin{array}{l} \exists pe_i \in PE \\ DP_i = [dp_i^E/dp_i^I \ dp_i^S/dp_i^N \ dp_i^T/dp_i^F \ dp_i^J/dp_i^P] \\ dp_i^X/dp_i^Y = \{1/-1, \ -1/1\} \end{array} \right. \right\} \quad (5)$$

Adotando como entrada os perfis de personalidades de papéis P_{PA} e pessoas P_{PE} , a etapa de *Avaliação do Perfil de Personalidade* da abordagem proposta mensura o *nível de adequação não-técnica*, definido na Equação 6 pelo relacionamento $ANT_{PE \times PA}$, onde cada termo $ant_{i,j}$ define o mapeamento do nível de habilidade psicológica de cada pessoa pe_i para cada papel pa_j , comparando as dimensões MBTI requeridas pelo papel com aquelas que caracterizam a pessoa em questão. Nesta comparação, conforme definido na Equação 7, em decorrência da multiplicação, a pessoa é positivamente pontuada com o valor 1 nas seguintes situações: (i) a pessoa apresenta a dimensão $dp_i^X = 1$ correspondente à requerida pelo papel $dr_j^X = 1$; (ii) a pessoa não apresenta a dimensão $dp_i^X = -1$ que também não é requerida pelo papel $dr_j^X = -1$. De forma inversa, a pessoa é negativamente pontuada com o valor -1 nas seguintes condições: (i) a pessoa possui a dimensão $dp_i^X = 1$ que não é requerida pelo papel $dr_j^X = -1$; (ii) a pessoa não apresenta a dimensão $dp_i^X = -1$ que é requerida pelo papel $dr_j^X = 1$. No caso de dimensões que não exercem qualquer influência no papel $dr_j^X = 0$, nenhuma pontuação é atribuída à pessoa.

$$ANT_{PE \times PA} = \{ant_{i,j} \mid \exists pe_i \in PE, \exists pa_j \in PA, ant_{i,j} \in \mathbb{Z}\} \quad (6)$$

$$ant_{i,j} = \sum_{X \in TR_{MBTI}} dp_i^X \cdot dr_j^X \quad (7)$$

A Figura 22 ilustra o cálculo da adequação não-técnica de uma determinada pessoa, cujo perfil MBTI é “ENFP”, em relação ao papel de analista de sistemas, cujo perfil MBTI é “E*F”.

	E	I	S	N	T	F	J	P	
Pessoa pe_i	1	-1	-1	1	-1	1	-1	1	Nível de Adequação Não-Técnica $ant_{i,j} = 4$
Papel pa_j	1	-1	0	0	-1	1	0	0	
$dp_i^X \times dr_j^X$	1	1	0	0	1	1	0	0	

Figura 22 – Exemplo de cálculo do nível de adequação não-técnica

Pode ser observado que o nível de adequação não-técnica $ANT_{PE \times PA}$ é o resultado final gerado pela etapa *Avaliação do Perfil de Personalidade* e adotado como entrada da etapa seguinte *Avaliação Integrada Papéis-Pessoas*, que é responsável por recomendar as soluções candidatas com seus respectivos níveis de adequação técnica e não-técnica, conforme será detalhado mais adiante.

3.2.1.2. FFM

Conforme explanado na seção 2.2.2, o FFM classifica a personalidade dos indivíduos em relação aos cinco fatores, nos quais os indivíduos pontuam para cada um deles. O conjunto de traços dos fatores FFM é representado por $TR_{FFM} = \{O, C, E, A, N\}$. As ferramentas de análise das personalidades das pessoas, em geral, retornam o percentual obtido em cada um destes fatores (Figura 6). Porém Rehman *et al.* (2012), em busca das personalidades FFM ideais para o desempenho dos papéis de desenvolvimento de software, adotaram os níveis alto ou baixo, conforme explicado na seção 2.3.1.1. Diante disto, em suma, esta abordagem modela matematicamente este comportamento da seguinte forma: para os fatores que forem indicados em nível alto, as pessoas que pontuarem acima de 50 serão mais indicadas, enquanto que para os fatores que forem indicados em nível baixo, as pessoas que pontuarem abaixo de 50 serão mais indicadas. Desta forma, na abordagem proposta, o valor 50 é considerado um valor neutro.

Por um lado, conforme representado na Equação 8, o *perfil de personalidade dos papéis* é denotado pelo conjunto P_{PA} , onde cada termo FR_j é um vetor de cinco fatores FFM, que conjuntamente indicam os traços de personalidade requeridos pelo papel p_j . Observe que cada fator pode assumir os valores 1, -1 ou 0, representando ser requerido em alto nível, requerido em baixo nível ou indiferente, respectivamente. Em cada termo FR_j , as posições do vetor correspondem aos seguintes fatores FFM: Abertura a experiência (O), Conscienciosidade (C),

Extroversão (E), Amabilidade (A), e, Neuroticismo (N). A Figura 23 ilustra vetores (FR_j) gerados para cada papel (pa_j) da Tabela 1.

$$P_{PA} = \left\{ FR_j \left| \begin{array}{l} \exists pa_j \in PA \\ FR_j = [fr_j^O \ fr_j^C \ fr_j^E \ fr_j^A \ fr_j^N] \\ fr_j^X = \{-1, 0, 1\} \end{array} \right. \right\} \quad (8)$$

	O	C	E	A	N
Analista de Sistemas	0	0	1	1	0
Projetista de <i>Software</i>	1	0	0	1	0
Programador	1	0	-1	1	0
Testador	1	1	0	0	0
Manutenedor	1	1	0	0	0

Figura 23 – Representação dos domínios FFM requeridos pelos papéis

Por outro lado, de acordo com a Equação 9, o *perfil de personalidade das pessoas* é representado pelo conjunto P_{PE} , onde cada termo FP_i é um vetor de cinco fatores FFM, que conjuntamente indicam os traços de personalidade que caracterizam a pessoa pe_i . Veja que cada fator fp_i^X pode assumir valores inteiros no intervalo $[0, 100]$, representando o nível percentual que a pessoa pe_i possui no fator X . A Figura 24 exemplifica a classificação de personalidade de três pessoas fictícias.

$$P_{PE} = \left\{ FP_i \left| \begin{array}{l} \exists pe_i \in PE \\ FP_i = [fp_i^O \ fp_i^C \ fp_i^E \ fp_i^A \ fp_i^N] \\ fp_i^X \in \mathbb{Z} \wedge 0 \leq fp_i^X \leq 100 \end{array} \right. \right\} \quad (9)$$

	O	C	E	A	N
Pessoa 1	95	92	61	65	5
Pessoa 2	10	51	98	70	60
Pessoa 3	70	80	85	20	10

Figura 24 – Exemplos de representações de personalidades FFM

Adotando como entrada os perfis de personalidade dos papéis P_{PA} e das pessoas P_{PE} , a etapa de *Avaliação do Perfil de Personalidade* da abordagem mensura o *nível de adequação*

não-técnica, definido na Equação 10 pelo relacionamento $ANT_{PE \times PA}$, onde cada termo $ant_{i,j}$ define o mapeamento do nível de habilidade psicológica de cada pessoa pe_i para cada papel pa_j , comparando os fatores FFM requeridos pelo papel com aqueles que caracterizam a pessoa em questão. Nesta comparação, conforme definido na Equação 11, em decorrência da subtração e da multiplicação, a pessoa é positivamente pontuada nas seguintes situações: (i) a pessoa apresenta alto nível no fator em questão $fp_i^X > 50$ e o papel também requer alto nível naquele fator $fr_j^X = 1$; (ii) a pessoa apresenta baixo nível no fator avaliado $fp_i^X < 50$ e o papel também requer baixo nível naquele fator $fr_j^X = -1$. De forma inversa, a pessoa é negativamente pontuada nas seguintes situações: (i) a pessoa apresenta alto nível no fator em questão $fp_i^X > 50$, mas o papel requer baixo nível naquele fator $fr_j^X = -1$; (ii) a pessoa apresenta baixo nível no fator avaliado $fp_i^X < 50$, mas o papel requer alto nível naquele fator $fr_j^X = 1$. No caso de fatores que não exercem qualquer influência no papel $fr_j^X = 0$, nenhuma pontuação é atribuída à pessoa. Portanto, para fatores requeridos em nível alto pelo papel, quanto mais alto for o valor possuído pela pessoa, mais adequado é considerado o candidato. Em contrapartida, para fatores requeridos em nível baixo pelo papel, quanto mais baixo for o valor possuído pela pessoa, mais adequado é considerado o candidato. Para normalizar a influência de cada fator FFM no intervalo $[-1, 1]$, primeiro, a multiplicação por 2 leva o resultado parcial do fator para o intervalo $[-100, 100]$, e, em seguida, a divisão por 100 traz o resultado final do fator para o intervalo normalizado.

$$ANT_{PE \times PA} = \{ant_{i,j} \mid \exists pe_i \in PE, \exists pa_j \in PA, ant_{i,j} \in \mathbb{R}\} \quad (10)$$

$$ant_{i,j} = \sum_{X \in TR_{FFM}} \frac{2 * fr_j^X * (fp_i^X - 50)}{100} \quad (11)$$

Para exemplificar, a Figura 25 ilustra o cálculo da adequabilidade de uma pessoa i para o papel j .

	O	C	E	A	N	
Pessoa pe_i	95	92	61	65	5	Nível de Adequação Não-Técnica $ant_{i,j} = 0.98$
Papel pa_j	1	0	-1	1	0	
$\frac{2 * fr_j^X * (fp_i^X - 50)}{100}$	0.90	0.00	-0.22	0.30	0.00	

Figura 25 - Exemplo de cálculo de adequabilidade FFM

Como já mencionado, o nível de adequação não-técnica $ANT_{PE \times PA}$ é resultante da etapa *Avaliação do Perfil de Personalidade* e adotado como entrada da etapa seguinte *Avaliação Integrada Papéis-Pessoas*, a qual é responsável pela recomendação das soluções candidatas, com seus respectivos níveis de adequação técnica e não-técnica, conforme será explicado a seguir.

3.3. Avaliação do Perfil Técnico

Para a etapa de caracterização técnica das pessoas serão levados em consideração atributos que tenham relevância com o conhecimento técnico que as pessoas possuem, como habilidade em determinada tecnologia e conhecimento acerca de um dado domínio. Quanto aos papéis de software, serão caracterizados pelos requisitos técnicos esperados que uma pessoa deve apresentar para ser capaz de desempenhá-los.

A etapa denominada *Avaliação do Perfil Técnico* realiza a avaliação dos diferentes perfis técnicos requeridos pelos diversos papéis funcionais e os perfis técnicos dominados pelas pessoas candidatas, contrastando assim as habilidades, conhecimentos e experiências das pessoas em relação aos papéis funcionais. Nesta etapa, as entradas são representadas pelo **perfil técnico dos papéis** e **perfil técnico das pessoas**, gerando como saída o mapeamento do **nível de adequação técnica** de cada pessoa candidata a cada papel funcional. Na abordagem proposta, os perfis técnicos de papéis e pessoas são caracterizados com base em adaptações da proposta de Souza e Elias (2018), que originalmente propõem mecanismos e estratégias para caracterizar aspectos técnicos de módulos de software e equipes de desenvolvimento, em relação aos requisitos e conhecimentos, como, por exemplo, sistemas operacionais, plataformas, ferramentas, linguagens de programação e processos.

3.3.1. Perfil Técnico dos Papéis

Do ponto de vista dos papéis, é necessário identificar o conjunto das tecnologias (sistema operacional, linguagens de programação, plataformas, ferramentas, processos, entre outras) e domínios de conhecimento necessários à execução do papel, representando os requisitos técnicos que as pessoas devem possuir para conseguir desempenhá-lo. Além disso, é necessário obter qual o nível de conhecimento exigido em cada requisito técnico. Na abordagem

proposta, os requisitos técnicos de cada papel funcional e seus respectivos níveis de conhecimento são representados na forma de uma **tabela de requisitos técnicos** (TRT).

Como em Santos (2014), as colunas da tabela de requisitos técnicos representam os níveis de conhecimento requeridos e as células contêm os requisitos técnicos (as tecnologias e conhecimentos de domínio) correspondentes ao papel em questão. A Tabela 2 exemplifica duas TRT para dois papéis do projeto. Vale salientar que, para cada papel funcional do projeto, uma TRT deve ser construída. É entendido que o gerente de software, juntamente com o arquiteto de software ou líder técnico do projeto, definem os requisitos técnicos e os respectivos níveis de conhecimento, pois, na maioria dos projetos, o arquiteto ou líder técnico é o responsável pela tomada de decisões sobre as tecnologias a serem adotadas no desenvolvimento dos vários componentes arquiteturais. Dessa forma, o nível de conhecimento não é calculado pela abordagem aqui proposta, mas apenas fornecido como entrada.

Tabela 2 - Exemplos de tabelas de requisitos técnicos

Programador		
Baixo	Médio	Alto
Scrum	Eclipse	Python
-	MySQL	-

Engenheiro de Testes		
Baixo	Médio	Alto
		JUnit

Dada a complexidade de quantificar níveis de conhecimento, Souza e Elias (2018) adotam termos *fuzzy* para representá-los, a saber: *nenhum*, *baixo*, *médio* e *alto*. Por analogia, na abordagem aqui proposta, o *perfil técnico dos papéis* também pode ser caracterizado de forma similar. Assim, os valores de referência para cada um dos níveis de conhecimento considerados nesta abordagem adotam como padrão os mesmos termos fuzzy adotados por Souza e Elias (2018). Porém, o número de níveis e os termos usados para descrever os níveis de conhecimento podem ser redefinidos pelo gerente de projeto, caso julgue necessário. A configuração padrão dos níveis de conhecimento possíveis para cada requisito técnico de cada papel e os seus valores de referência é representada na Tabela 3.

Tabela 3 - Valores de referência para os níveis de complexidade

Nenhum	Baixo	Médio	Alto
0	0.2	0.5	0.8

Logo, na abordagem proposta, para cada papel funcional, o gerente de projeto deve elaborar uma tabela para representar os requisitos técnicos e níveis de conhecimento. Posteriormente, tais tabelas são matematicamente mapeadas, de acordo com a Equação 12, para modelar o *perfil técnico dos papéis*, que é caracterizado pelo conjunto T_{PA} , onde cada termo CR_j representa o conjunto de níveis de conhecimento $cr_{j,k}$ requeridos pelo papel pa_j em relação aos requisitos técnicos rt_k . Note que os requisitos técnicos requeridos pelos vários papéis são representados pelo conjunto $RT = \{rt_1, rt_2, \dots, rt_p\}$. Por outro lado, os níveis de conhecimento requeridos são denotados pelo conjunto $CX = \{0.0, 0.2, 0.5, 0.8\}$, que são os quantificadores dos termos *fuzzy* adotados por padrão (Tabela 3).

$$T_{PA} = \left\{ CR_j \left| \begin{array}{l} \exists pa_j \in PA \\ CR_j = \{cr_{j,k} \mid \exists rt_k \in RT, cr_{j,k} \in CX\} \end{array} \right. \right\} \quad (12)$$

Por exemplo, considere um projeto que possui apenas os papéis de *Programador* (pa_1) e *Engenheiro de Testes* (pa_2). Neste projeto, como ilustrado na Tabela 2, o papel de *Programador* tem como requisitos técnicos *Scrum* (rt_1), *Eclipse* (rt_2), *Python* (rt_3) e *MySQL* (rt_4) com níveis de conhecimento *baixo*, *médio*, *alto* e *médio*, respectivamente. Já o papel de *Engenheiro de Testes* tem apenas o requisito técnico *JUnit* (rt_5) com nível de conhecimento *alto*, conforme indicado na Tabela 2. Neste caso, o *perfil técnico dos papéis* é representado pelo conjunto $T_{PA} = \{ CR_1, CR_2 \}$, onde os papéis de *Programador* e *Engenheiro de Testes* possuem os seguintes níveis de conhecimento $CR_1 = \{0.2, 0.5, 0.8, 0.5, 0.0\}$ e $CR_2 = \{0.0, 0.0, 0.0, 0.0, 0.8\}$, respectivamente.

3.3.2. Perfil Técnico das Pessoas

Em relação às habilidades técnicas das pessoas, Souza e Elias (2018) adotam um esquema baseado em formulários para estimar o nível de habilidades das pessoas em cada requisito técnico, onde três eixos são avaliados: *anos de experiência*, *número de projetos desenvolvidos* e *número de certificações*. Este formulário é elaborado com base nos requisitos técnicos (tecnologias e domínios), visando evitar o desperdício de tempo e recursos que poderia ocorrer caso fossem utilizados questionários demasiadamente longos ou não proveitosos. Por

exemplo, a Figura 26 ilustra um formulário para avaliar o nível de habilidade no requisito técnico *Python*.

Requisito Técnico: PYTHON					
Há quantos anos você desenvolve projetos em PYTHON?					
☐ none	☐ 1-3	☐ 3-5	☐ 5-7	☒ 7-9	☐ +9
Quantos projetos você desenvolveu usando a linguagem PYTHON?					
☐ none	☐ 1-5	☐ 5-10	☒ 10-15	☐ 15-20	☐ +20
Quais certificações você possui em PYTHON?					
☒ PCEP-30	☐ PCPP-32-1				
☐ PCAP-31	☐ PCPP-32-2				

Figura 26 - Exemplo de formulário

Por meio dos dados coletados nos formulários, a experiência de cada pessoa em cada requisito técnico pode ser inferida, gerando um número entre zero e um, que expressa a competência ou habilidade técnica estimada (conhecimento que uma pessoa apresenta em algum requisito técnico). O exemplo apresentado na Figura 26 contempla a informação em relação a linguagem de programação *Python*. Observe que o formulário completo contempla informações acerca de todos requisitos técnicos do projeto, representados pelo conjunto *RT*. Para cada requisito técnico, até três dados devem ser coletados: anos de experiência, número de projetos desenvolvidos e número de títulos (títulos podem ser certificações ou especializações). Todas as respostas serão dadas pelas pessoas de maneira objetiva. Para as tecnologias que não possuem certificações ou especializações, esta informação pode ser omitida.

Originalmente, Souza e Elias (2018) propuseram o formulário através de avaliação da literatura da área. Por exemplo, como já mencionado, Shanteau *et al.* (2002) afirma que anos de experiência, títulos e certificados que alguém possui são comumente utilizados para avaliar se a pessoa é um especialista em determinado domínio. Segundo Weiss, Shanteau e Harries (2006), também citado por Souza e Elias (2018), um fator importante que determina se alguém é um especialista em um determinado domínio é o fator de capacidade de discriminação. Contudo essa capacidade de discriminação só pode ser vista reunindo informações através da observação de eventos passados, já que é tipicamente um dado empírico. Sendo assim, é possível inferir, até certo grau, a capacidade de discriminação de uma pessoa através do número de projetos que ela tenha participado. Com base nessa análise da literatura, Souza e Elias (2018) adotaram os três eixos de avaliação (anos de experiência, número de projetos desenvolvidos e número de títulos)

para compor o formulário, que também atendem de forma satisfatória as necessidades específicas da abordagem aqui proposta.

Neste caso, o gerente de projeto deve montar um formulário para cada requisito técnico dos vários papéis funcionais, de modo que o nível de adequação técnica de cada pessoa pode ser inferido para cada requisito técnico. A abordagem proposta também adota como padrão as mesmas opções de valores sugeridos em Souza e Elias (2018) para anos de experiência e número de projetos. Para número de certificações, assim como em Souza e Elias (2018), a abordagem proposta não adota opções pré-definidas, permitindo o gerente de projeto enumerar certificações relacionadas ao requisito técnico em questão. Logo, nos três eixos, as opções podem ser customizadas, caso desejado.

Nota-se que, no formulário exemplo (Figura 26), os dados coletados para cada requisito técnico (anos de experiência, número de projetos desenvolvidos e número de títulos) podem possuir diferentes representações: valores máximos, quantidade de níveis e variação entre níveis. Após a aplicação do questionário para as pessoas candidatas e a realização da coleta, o cálculo do valor de habilidade técnica que as pessoas apresentam para cada requisito técnico do projeto pode ser realizado.

Neste sentido, tais formulários são matematicamente mapeados pela Equação 13 para modelar o *perfil técnico das pessoas*, que é caracterizado pelo conjunto T_{PE} , onde cada termo CP_i representa o conjunto de níveis de conhecimento $cp_{i,k}$ possuídos pela pessoa pe_i em relação aos requisitos técnicos $rt_k \in RT$. Note que os níveis de conhecimento são denotados como números reais no intervalo $[0, 1]$.

$$T_{PE} = \left\{ CP_i \mid \begin{array}{l} \exists pe_i \in PE \\ CP_i = \{ cp_{i,k} \mid \exists rt_k \in RT, cp_{i,k} \in [0, 1] \} \end{array} \right\} \quad (13)$$

Baseado em Souza e Elias (2018), o nível de conhecimento $cp_{i,k}$ de cada pessoa pe_i em cada requisito técnico rt_k é inferido dos formulários, conforme indicado na Equação 14, realizando uma média ponderada das pontuações obtidas nos eixos avaliados. Nesta equação, os termos $Y_{i,k}$, $P_{i,k}$ e $C_{i,k}$ representam as pontuações obtidas em anos de experiência, projetos e certificações, respectivamente. Já os termos w_Y , w_P e w_C representam seus respectivos pesos de importância, que por padrão assumem os valores 4, 4, e 2, mas que podem ser ajustados, caso seja de interesse. Observe que o eixo de certificações adota menor peso, pois segundo Shanteau *et al.* (2002), as certificações configuram o eixo de menor representatividade.

$$cp_{i,k} = \frac{Y_{i,k} \cdot w_Y + P_{i,k} \cdot w_P + C_{i,k} \cdot w_C}{w_Y + w_P + w_C} \quad (14)$$

Para inferir as pontuações $Y_{i,k}$ e $P_{i,k}$ nos eixos de anos de experiência e projetos, o cálculo é realizado conforme indicado nas Equações 15 e 16. Nestas equações, por um lado, os termos ny_k e np_k representam as quantidades de opções relacionadas aos respectivos eixos no formulário do requisito técnico rt_k . Por outro lado, os termos $oy_{i,k}$ e $op_{i,k}$ representam as posições das opções selecionadas pela pessoa pe_i relacionadas aos respectivos eixos no formulário do requisito técnico rt_k . No exemplo da Figura 26, os eixos de anos de experiência e projetos possuem seis opções $ny_k = np_k = 6$, cujas posições das opções selecionadas são $oy_{i,k} = 5$ e $op_{i,k} = 4$. Portanto, neste caso, as pontuações nos eixos são $Y_{i,k} = 0.8$ e $P_{i,k} = 0.6$.

$$Y_{i,k} = \frac{oy_{i,k} - 1}{ny_k - 1} \quad (15)$$

$$P_{i,k} = \frac{op_{i,k} - 1}{np_k - 1} \quad (16)$$

Diferentemente, a pontuação $C_{i,k}$ no eixo de certificações é inferida conforme indicado na Equação 17. Nesta equação, o termo nc_k representa a quantidade de certificações enumeradas no formulário do requisito técnico rt_k . Já o termo $oc_{i,k}$ representa a quantidade de certificações da pessoa pe_i no requisito técnico rt_k . Na Figura 26, o eixo de certificações possui quatro opções $nc_k = 4$. Considerando que um único certificado foi selecionado então $oc_{i,k} = 1$. Portanto, neste caso, a pontuação obtida no eixo de certificações é $C_{i,k} = 0.25$.

$$C_{i,k} = \frac{oc_{i,k}}{nc_k} \quad (17)$$

Considerando o formulário da Figura 26, com base na Equação 14, o nível de conhecimento $cp_{i,k}$ da pessoa pe_i no requisito técnico rt_k (*Python*) é calculado da seguinte forma: $cp_{i,k} = (0.8 \cdot 4 + 0.6 \cdot 4 + 0.25 \cdot 2) / (4 + 4 + 2) = 0.61$.

3.3.3. Avaliação da qualificação técnica das pessoas para os papéis

Neste ponto, adotando como entrada os perfis técnicos de papéis T_{PA} e pessoas T_{PE} , a etapa de *Avaliação do Perfil Técnico* da abordagem proposta mensura o *nível de adequação técnica*, definido na Equação 18 pelo relacionamento $AT_{PE \times PA}$, onde cada termo $at_{i,j}$ define o mapeamento do nível de adequabilidade técnica de cada pessoa pe_i para cada papel pa_j , comparando o conjunto de níveis de conhecimento necessários nos requisitos técnicos do papel com aqueles da pessoa em questão. A abordagem proposta adota uma política de seleção baseada em *qualificação equivalente*, cujo objetivo é nivelar os níveis de conhecimentos técnicos dos papéis e das pessoas. Como expresso na Equação 19, a adequabilidade técnica $at_{i,j}$ de uma pessoa pe_i para o papel pa_j é definida pela média da adequabilidade dessa pessoa nos diversos requisitos técnicos rt_k do papel. Neste caso, para caracterizar a política de qualificação equivalente, para cada requisito técnico rt_k , a distância absoluta $|cp_{i,k} - cr_{j,k}|$ entre o nível de conhecimento possuído pela pessoa $cp_{i,k}$ e requerido pelo papel $cr_{j,k}$ é calculada, e, em seguida, subtraída do valor 1, de modo que quanto menor é a distância, maior é a pontuação. Em outras palavras, quanto mais distantes os níveis, seja para mais ou para menos, menor se torna a adequabilidade. Observe que a função teto $\lceil cr_{j,k} \rceil$ é adotada para desprezar requisitos técnicos rt_k não demandados pelo papel pa_j . Ou seja, se o requisito técnico rt_k não é demandado pelo papel pa_j , então $cr_{j,k} = 0$, e, portanto, a função teto mantém o valor nulo, eliminando assim qualquer impacto do requisito técnico em questão no resultado. Caso contrário, se o requisito técnico rt_k é demandado pelo papel pa_j , então $0 < cr_{j,k} \leq 1$, e, portanto, a função teto assume o valor 1, incluindo assim o impacto do requisito técnico em questão no resultado.

$$AT_{PE \times PA} = \{at_{i,j} \mid \exists pe_i \in PE, \exists pa_j \in PA, at_{i,j} \in [0, 1]\} \quad (18)$$

$$at_{i,j} = \frac{\sum_{rt_k \in RT} (1 - |cp_{i,k} - cr_{j,k}|) \cdot \lceil cr_{j,k} \rceil}{\sum_{rt_k \in RT} \lceil cr_{j,k} \rceil} \quad (19)$$

Como exemplo do cálculo de adequabilidade, por um lado, considere a pessoa pe_i , que possui o seguinte conjunto de níveis de conhecimento $CP_i = \{0.0, 0.2, 0.6, 0.7, 0.5\}$ nos requisitos técnicos $rt_1, \dots, rt_5$, respectivamente. Por outro lado, considere o papel pa_j , que demanda o seguinte conjunto de níveis de conhecimento $CR_j = \{0.2, 0.5, 0.8, 0.5, 0.0\}$ nos

referidos requisitos técnicos, respectivamente. Neste caso, com base na Equação 19, o nível de adequação técnica $at_{i,j}$ da pessoa pe_i ao papel pa_j é 0.775.

Vale ressaltar que a política de qualificação equivalente é adotada na abordagem proposta por dois motivos (BARRETO, 2005): (i) evitar a frustração e o aumento de custo decorrente da seleção de pessoas com níveis de conhecimentos superiores aos requeridos pelos papéis; e (ii) evitar atrasos e perda de qualidade em função da seleção de pessoas com níveis de conhecimentos inferiores aos requeridos pelos papéis.

3.4. Avaliação Integrada Pessoas-Papéis

Nesta terceira etapa, denominada *Avaliação Integrada Pessoas-Papéis*, a abordagem proposta utiliza metaheurísticas capazes de lidar com múltiplos objetivos, cujos conceitos estão apresentados no capítulo 2. Deste modo, são necessárias implementações destas metaheurísticas multiobjetivo. Para isto, foi utilizado o jMetal (DURILLO; NEBRO, 2011) que é um *framework* capaz de executá-las, codificar o problema em questão, e apresentar resultados com base em indicadores qualitativos (DURILLO; NEBRO, 2011). Este *framework* é apresentado no Apêndice A deste trabalho.

Neste ponto, a abordagem já está de posse do **nível de adequação não-técnica** $ANT_{PE \times PA}$ e do **nível de adequação técnica** $AT_{PE \times PA}$ das pessoas em relação aos papéis funcionais, definidos nas Equações 10 e 18 das duas etapas anteriores. Portanto, neste momento, esta etapa necessita que seja fornecida como entrada o **número de vagas/papel**, representado na Equação 20 pelo conjunto VG , que é o vetor que contém os números de vagas vg_j para cada papel funcional pa_j do projeto de software em questão.

$$VG = \{ vg_j \mid \exists pa_j \in PA, vg_j \in \mathbb{N}^* \} \quad (20)$$

Com este dado, é possível ilustrar a representação das soluções do problema (cromossomo), que será codificada na forma de uma cadeia de inteiros (genes), segmentados por papéis funcionais de acordo com o número de vagas por papel, cujos possíveis valores são os identificadores das pessoas candidatas (alelos). Cada vaga deve ser alocada para uma única pessoa, e, como restrições, cada pessoa somente pode ser alocada para uma única vaga e todas as vagas devem ser preenchidas.

Um ponto importante a ser observado é que, para cada projeto de software, é responsabilidade do gerente de projetos identificar a quantidade requerida de pessoas para cada papel funcional, tomando por base as especificidades do projeto em questão e as suas experiências adquiridas em projetos anteriores. Vale ressaltar que está fora do escopo deste trabalho apresentar e discutir propostas para dimensionar as vagas nos diferentes papéis do projeto de software.

Por exemplo, a Figura 27 ilustra uma solução possível de seleção e alocação de pessoas a papéis em um projeto de software que contempla 50 pessoas candidatas $PE = \{pe_1, \dots, pe_{50}\}$ e 5 papéis funcionais $PA = \{pa_1, \dots, pa_5\}$, cada um deles possuindo as seguintes quantidades de vagas $VG = \{4, 5, 3, 2, 2\}$, respectivamente. Note que cada vaga é preenchida por uma única pessoa candidata. Além disso, cada pessoa somente pode ser alocada para uma única vaga.

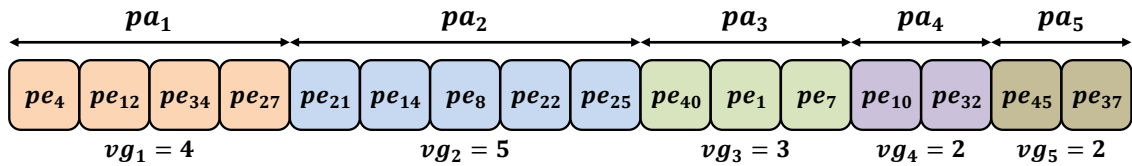


Figura 27 - Exemplo de cromossomo

Embora a representação gráfica da Figura 27 seja intuitiva, do ponto de vista da modelagem matemática, a abordagem proposta representa cada solução candidata pelo conjunto X , conforme indicado na Equação 21, onde cada termo $x_{i,j}$ pode assumir os valores 1 ou 0, para representar a alocação ou não da pessoa pe_i para uma vaga do papel pa_j , respectivamente.

$$X = \{x_{i,j} \mid \exists pe_i \in PE, \exists pa_j \in PA, x_{i,j} \in \{0,1\}\} \quad (21)$$

É importante ressaltar que uma solução candidata somente é válida quanto atende duas restrições. Na primeira restrição, expressa na Equação 22, cada pessoa pe_i pode no máximo ser alocada a uma única vaga de algum dos papéis pa_j . Na prática, a alocação de uma pessoa a apenas um papel pode ser muito restritiva. Assim, em futuras versões da proposta, esta restrição será aliviada permitindo a alocação parcial de pessoas a diversos papéis. Na segunda restrição, definida na Equação 23, para cada papel pa_j , devem ser alocadas pessoas pe_i para completar todas as vagas disponíveis vg_j naquele papel.

$$\forall pe_i \in PE \Rightarrow \sum_{pa_j \in PA} x_{i,j} \leq 1 \quad (22)$$

$$\forall pa_j \in PA \Rightarrow \sum_{pe_i \in PE} x_{i,j} = vg_j \quad (23)$$

3.4.1. Funções objetivos

Para finalizar, a etapa de *Avaliação Integrada de Pessoas-Papéis* da abordagem proposta, baseada em algoritmos evolucionários multiobjetivo, do inglês *Multiobjective Evolutionary Algorithms* (MOEAs), recomenda um conjunto de soluções válidas não-dominadas na perspectiva da **avaliação não-técnica** e da **avaliação técnica**, representadas pelas funções objetivo de maximização ANT_X e AT_X , definidas nas Equações 24 e 25, respectivamente. Em ambos os casos, a avaliação consiste basicamente no somatório dos níveis de adequação não-técnico $ant_{i,j}$ e técnico $at_{i,j}$ das pessoas pe_i alocadas aos papéis p_j na solução em questão, respectivamente.

$$\text{maximizar } ANT_X = \sum_{pe_i \in PE} \sum_{pa_j \in PA} ant_{i,j} \cdot x_{i,j} \quad (24)$$

$$\text{maximizar } AT_X = \sum_{pe_i \in PE} \sum_{pa_j \in PA} at_{i,j} \cdot x_{i,j} \quad (25)$$

Em conclusão, uma vez que a adequação não-técnica ANT_X e a adequação técnica AT_X estão definidas e modeladas adequadamente no jMetal, juntamente com a representação do cromossomo e as restrições, as metaheurísticas multiobjetivo podem ser executadas e retornam como saída a fronteira de soluções não-dominadas encontradas. Neste momento, o gerente de projeto pode avaliar a fronteira e deve optar por aquela solução que apresenta os valores de adequação não-técnica e técnica mais apropriados para atender as especificidades do projeto em questão. Porém, por serem objetivos conflitantes, provavelmente, não há uma solução vencedora em ambos objetivos simultaneamente. Assim, o gerente de projetos deve selecionar uma solução boa o suficiente para atender as demandas do projeto e da organização, por um lado, favorecendo

o objetivo de maior interesse, mas, por outro lado, penalizando o outro objetivo de menor interesse.

3.5. Considerações finais

Foi apresentada neste capítulo a abordagem para formação de equipes tecnicamente qualificadas e psicologicamente adequadas para os papéis de software. Para isto, esta abordagem possui três etapas com artefatos de entrada e saída definidos: *Avaliação do Perfil Técnico*, *Avaliação do Perfil de Personalidade* e *Avaliação Integrada Pessoas-Papéis*.

Destas etapas, a avaliação do perfil de personalidade gera como saída o nível de adequabilidade não-técnica das pessoas para os papéis, utilizando como critério de cálculo a correspondência entre os traços de personalidade desejados para o desempenho das tarefas que compõe o papel e os possuídos pelas pessoas que são candidatas a compor o projeto.

Já a etapa da avaliação do perfil técnico avalia o nível de aptidão técnica das pessoas para os papéis. Para isto, receberá como entrada quais são os conhecimentos técnicos necessários para cada papel, juntamente com o nível desejado deles. É considerado que esta informação é informada pelo gerente de software auxiliado pelo arquiteto de software ou líder técnico. Além disso, esta etapa necessita também conhecer qual é o nível de conhecimento possuído pelas pessoas candidatas, sendo assim, é sugerida a aplicação de formulários visando capturar esta informação.

Finalmente, a terceira etapa utiliza as saídas das etapas anteriores, e, baseada em metaheurísticas multiobjetivas, recomenda um conjunto de equipes, que representam soluções válidas não-dominadas. De posse de tais soluções, o gerente de projetos pode selecionar aquela mais adequada às necessidades do projeto ou organização. Vale ressaltar que, na formulação multiobjetiva, dois objetivos são considerados, ambos de maximização, permitindo a avaliação de uma dada solução candidata em relação às qualificações técnica e não-técnica.

4. RESULTADOS E DISCUSSÃO

No capítulo anterior, a abordagem proposta para o problema de seleção e alocação de pessoas para projetos de desenvolvimento de software incorporando a análise dos perfis técnicos e de personalidades foi descrita em detalhes. Concluída esta etapa, é necessária a validação da proposta com base em experimentos. Apesar do foco deste trabalho ser uma modelagem multiobjetiva para o problema avaliado, inicialmente foram verificados os resultados da proposta observando se a modelagem matemática descrita no Capítulo 3 apresenta resultados numéricos consoantes com o sentido designado pelas respectivas equações. Deste modo, antes da realização de um experimento para uma proposta multiobjetiva utilizando alguma metaheurística anteriormente apresentada, foi realizado um experimento reduzido com um algoritmo de busca exaustiva de modo a facilitar a interpretação dos resultados obtidos e a comparação com o que é esperado do modelo proposto.

Nesta direção, são apresentados resultados práticos em experimentos com base em dados reais utilizando as metaheurísticas descritas no Capítulo 2. Deste modo, também é possível validar o modelo de acordo com o que é esperado de um problema multiobjetivo para a seleção e alocação de pessoas. Os dados reais foram coletados por meio de estudo de caso e experimentos conduzidos em uma instituição pública no Brasil, denominada Instituto Federal da Paraíba (IFPB), mais especificamente na Coordenação de Desenvolvimento e Gerenciamento de Sistemas, pois tem um ambiente e uma equipe de desenvolvimento que mostrou-se bastante adequada, favorável e acessível à execução do estudo de caso, contemplando um portfólio de projetos de software que possibilitam a condução de experimentos, e, ao mesmo tempo, podem ser beneficiados pela abordagem proposta.

Dentre o portfólio de projetos de software desenvolvidos ou mantidos pela organização analisada está o Sistema Unificado de Administração Pública (SUAP), que foi desenvolvido inicialmente pelo Instituto Federal do Rio Grande do Norte (IFRN) e cedido de forma colaborativa para uso em outros institutos federais, incluindo o IFPB. Atualmente, o sistema se encontra em pleno desenvolvimento e, portanto, continua recebendo atualizações constantes para resolução de *bugs*, adição de novas funcionalidades, melhorias de funcionalidades existentes e criação de novos módulos. Estas atualizações, no âmbito do IFPB, são realizadas por equipe própria local. Outro importante projeto constante do portfólio do IFPB é o desenvolvimento e atualização do Portal do Estudante do IFPB, que é uma plataforma

desenvolvida pela equipe com o objetivo de apresentar as informações e gerenciar os processos relacionados aos cursos e processos seletivos para os estudantes que queiram estudar no IFPB.

Na seção seguinte são apresentados os procedimentos de planejamento e coleta de dados necessários para validação e avaliação da abordagem proposta.

4.1. Planejamento da pesquisa e coleta de dados

Após a definição do local para realização da avaliação da abordagem, desejava-se entender como é o fluxo de trabalho da equipe de desenvolvimento de software desta organização, a maneira como é feita a alocação de RH, qual é a percepção do gerente sobre o tema e quais são as estruturas das equipes. Esta atividade de contextualização serviu para identificar que tipo de estrutura seguem as equipes de software desta organização e para guiar o restante do estudo de caso.

O estudo de caso teve o propósito exploratório, cujo objetivo é prover o pesquisador de maior conhecimento sobre o tema da pesquisa (WOHLIN *et al.*, 2012), visando identificar e entender alguns aspectos da equipe de desenvolvimento de software em análise, por exemplo: (i) a organização do fluxo de trabalho da equipe; (ii) a estratégia de seleção e alocação de pessoas aos projetos de software; (iii) a percepção do gerente de projetos sobre as características e os desafios da seleção e alocação de pessoas; (iv) os papéis funcionais do fluxo de trabalho adotado; e (v) a quantidade típica de pessoas por papel.

Por isso, inicialmente, foi feito um trabalho de reconhecimento do ambiente, realizado por meio de observação diária, na qual a imersão da pesquisadora no ambiente foi de grande importância neste momento, pois a observação busca compreender um fenômeno por meio do comportamento explícito, observável de forma natural e no ambiente que ocorre (ARAUJO, 2014). Ao longo das observações, a pesquisadora assumiu a postura de participante ativa, visto que ela é uma membra nata do grupo estudado. A vantagem é que normalmente não provoca uma ruptura no cotidiano do grupo, porém uma das desvantagens é que o pesquisador pode ser visto como um julgador ou avaliador dos indivíduos (MERRIAM, 2009). Para que isso fosse evitado, foi esclarecido a todos os envolvidos que o objetivo do estudo era buscar respostas a perguntas sobre um fenômeno, e não avaliar ou julgar o comportamento dos envolvidos.

Em estudos de caso, a coleta de dados por meio de entrevistas é a principal fonte de evidências (YIN, 2014) e, ao contrário da técnica de observação, busca conhecer as características implícitas do fenômeno (ARAUJO, 2014). Desta forma, entrevistas também foram conduzidas no estudo de caso. Foram realizadas entrevistas abertas, também ditas não

estruturadas, nas quais o entrevistador vai explorando os conceitos no decorrer da conversa (MERRIAM, 2009). As entrevistas foram presenciais e individuais, visando coletar as seguintes informações: (i) a forma de seleção e alocação de pessoas aos projetos; (ii) os papéis funcionais do fluxo de trabalho adotado; e (iii) a quantidade típica de pessoas por papel. As entrevistas foram realizadas com o gerente de projetos da organização, que possui bastante experiência na função e mais de dez anos de atuação em projetos de software.

Baseado em Srinivasan e Lundqvist (2010) e Lui e Chan (2006), pode-se identificar equipes de desenvolvimento estruturadas por papéis e por organização. Na estrutura por papéis, os membros ocupam papéis bem definidos dentro das equipes, por exemplo analista de requisitos e programador. Mesmo equipes ágeis, que são mais informais, também adotam papéis (SRINIVASAN; LUNDQVIST, 2010). Na estrutura por organização, os membros podem ser alocados a módulos dos sistemas. Neste contexto, a equipe de desenvolvimento avaliada no estudo de caso adota uma estrutura mista, onde subgrupos são alocados a módulos do sistema e dentro dos mesmos pode-se identificar atribuições por papéis. Ou seja, após a divisão dos profissionais em equipes atribuídas a módulos do sistema, as pessoas tendem a realizar continuamente o mesmo tipo de tarefa.

Na equipe de desenvolvimento alvo do estudo de caso, a partir das observações dos fluxos de trabalho e das entrevistas com o gerente de projetos, cinco papéis funcionais foram identificados, representados respectivamente pelo conjunto $PA = \{pa_1, \dots, pa_5\}$, a saber: Analista de Sistemas (pa_1), Projetista de Software (pa_2), Programador (pa_3), Engenheiro de Teste (pa_4) e Engenheiro de Manutenção (pa_5). Vale ressaltar que Capretz e Ahmed (2010) também identificam os mesmos papéis a partir da análise da literatura e de descrições em anúncios de emprego para engenheiros de software.

No que se refere aos critérios de seleção e alocação de equipes, bem como às atribuições de tarefas, a equipe avaliada não adota um procedimento formalizado, porém os seguintes aspectos técnicos estão intuitivamente presentes: conhecimentos e experiências; históricos de desempenho; e, em menor importância, preferências dos próprios profissionais. O gerente de projetos reconhece a importância dos aspectos não-técnicos, porém menciona que são difíceis de avaliar e medir. Mas, ainda que de forma *ad-hoc*, o gerente direciona pessoas com facilidades de comunicação e expressão para análise, pessoas minuciosas e detalhistas para testes, e pessoas cuidadosas no trabalho para programação.

Conforme o gerente de projetos, nenhum critério técnico ou não-técnico é eliminatório, e, por vezes, prevalece a necessidade e disponibilidade do momento. O próprio gerente considera prejudicial ao projeto a alocação intempestiva, mas, devido a quantidade de

atividades diárias, torna-se inviável a alocação mais criteriosa. Também é importante ressaltar que nenhum instrumento formal é adotado para avaliar os aspectos técnicos e não-técnicos dos membros da equipe, sendo, portanto, uma mera visão particular do gerente de projetos de acordo com sua vivência no dia-a-dia. Mais importante, ao ser introduzido aos conceitos, estratégias e posteriores experimentos, o gerente de projetos reconheceu abertamente a utilidade e aplicabilidade da abordagem proposta.

De acordo com o gerente de projetos, em geral os projetos de médio porte são mais frequentes na organização, requerendo em média uma equipe com 20 membros, distribuídos em três analistas de sistemas, três projetistas de software, sete programadores, quatro engenheiros de teste e três engenheiros de manutenção ($VG = \{3, 3, 7, 4, 3\}$). Considerando tais papéis, foram identificados 16 requisitos técnicos, representados respectivamente pelo conjunto $RT = \{rt_1, \dots, rt_{16}\}$, a saber: *Análise Orientada a Objetos* (rt_1), *Projeto Orientado a Objetos* (rt_2), *Django* (rt_3), *Git* (rt_4), *HTML* (rt_5), *JavaScript* (rt_6), *Rastreabilidade de Requisitos* (rt_7), *PostgreSQL* (rt_8), *Prototipagem* (rt_9), *Python* (rt_{10}), *PyCharm* (rt_{11}), *UML* (rt_{12}), *Selenium* (rt_{13}), *Suíte de Testes Django* (rt_{14}), *Técnicas de Criatividade com Brainstorming, Delphi e Mapa Mental* (rt_{15}), e *Técnicas e Ferramentas de Elicitação de Requisitos* (rt_{16}). Baseado nos papéis funcionais e seus requisitos técnicos, a Tabela 4 mostra o mapeamento de papéis, requisitos técnicos e níveis de conhecimento requeridos.

Tabela 4 - Papéis, requisitos e níveis de conhecimento

Papel	Nível de Conhecimento		
	Baixo	Médio	Alto
pa_1	rt_1	—	$rt_7, rt_9, rt_{12}, rt_{15}, rt_{16}$
pa_2	—	rt_3, rt_7, rt_{10}	rt_1, rt_2, rt_{12}
pa_3	—	$rt_4, rt_8, rt_{11}, rt_{12}$	$rt_1, rt_3, rt_5, rt_6, rt_{10}$
pa_4	rt_8	rt_4, rt_{11}, rt_{12}	$rt_3, rt_{10}, rt_{13}, rt_{14}$
pa_5	rt_8	rt_4, rt_{11}, rt_{12}	rt_3, rt_5, rt_{10}

A partir do mapeamento de papéis, requisitos técnicos e níveis de conhecimento, a implementação da abordagem proposta faz o mapeamento matemático automático para o perfil técnico dos papéis, caracterizado na Equação 12 pelo conjunto $T_{PA} = \{CR_1, \dots, CR_5\}$, onde cada termo CR_j representa os níveis de conhecimento $cr_{j,k}$ requeridos pelo papel pa_j para cada requisito técnico rt_k . Por exemplo, um *analista de sistemas* (pa_1), conforme a Tabela 4, possui os níveis de conhecimento $CR_1 = \{0.2, 0.0, \dots, 0.0, 0.8, 0.0, 0.8, 0.0, 0.0, 0.8, 0.0, 0.0, 0.8, 0.8\}$.

Posteriormente, como ponto de partida para coletar o perfil técnico das pessoas, de forma similar à Figura 26, um formulário específico foi elaborado para cada requisito técnico $rt_k \in RT$, contemplando os três eixos avaliados: *anos de experiência*, *projetos* e *certificações*. Os formulários foram respondidos pelos 30 membros da equipe de desenvolvimento alvo do estudo de caso, representados por $PE = \{pe_1, \dots, pe_{30}\}$, sendo a numeração atribuída na ordem de conclusão do preenchimento dos formulários. A aplicação destes formulários foi realizada de forma *online*.

A partir das respostas dos formulários, a implementação da abordagem proposta mapeia de forma automática e matemática o *perfil técnico das pessoas*, caracterizado na Equação 13 pelo conjunto $T_{PE} = \{CP_1, \dots, CP_{22}\}$, onde cada termo CP_i representa os níveis de conhecimento $cp_{i,k}$ da pessoa pe_i em cada requisito técnico rt_k . Note que os níveis de conhecimento são denotados como números reais no intervalo $[0, 1]$. Dado que o projeto contempla 16 requisitos técnicos, para não dificultar a leitura do texto, não é apresentado um exemplo de níveis de conhecimento de uma dada pessoa.

Neste ponto, com um nível de esforço relativamente baixo do gerente de projetos e da equipe de desenvolvimento alvo, o estudo de caso já obteve a caracterização dos perfis técnicos de papéis e pessoas. Agora, o estudo de caso deve se concentrar na caracterização dos perfis de personalidades de papéis e pessoas, que, respectivamente, identificam os traços de personalidade requeridos pelos papéis e marcantes nas pessoas, permitindo avaliar a adequabilidade psicológica das pessoas em relação aos papéis, que é considerado um fator de impacto no sucesso de um projeto de software (SILVA, F. 2009).

Nesta direção, adotando a tipologia MBTI, a abordagem proposta requer o mapeamento dos papéis funcionais às dimensões MBTI, caracterizando assim o *perfil de personalidade dos papéis*. Tendo como ponto de partida o mapeamento de Capretz e Ahmed (2010) (Tabela I), diversas apresentações e discussões foram realizadas com o gerente de projetos, primeiro, para colocá-lo a par dos conceitos da tipologia MBTI, e, segundo, para tentar adaptar o mapeamento sugerido à realidade da organização alvo. No entanto, ao final das discussões, o gerente de projetos julgou o mapeamento sugerido bastante adequado e não propôs adaptações. Surpreendentemente, o gerente de projetos mencionou que o mapeamento sugerido de fato se assemelha à forma intuitiva adotada no dia a dia, ainda que não claramente organizada na forma da tipologia MBTI.

Com base no mapeamento de papéis e dimensões MBTI, a implementação da abordagem proposta faz o mapeamento matemático automático para o *perfil de personalidade dos papéis*,

definido na Equação 4 pelo conjunto $P_{PA} = \{DR_1, \dots, DR_5\}$, onde cada termo DR_j representa um vetor de quatro pares de dicotomias MBTI, que conjuntamente sinalizam as dimensões bipolares requeridas pelo papel pa_j . Por exemplo, um *analista de sistemas* (pa_1), com perfil MBTI "E-F-" (Tabela 1), possui o vetor de personalidade $DR_1 = [1/-1 \ 0/0 \ -1/1 \ 0/0]$.

Em seguida, para coletar o perfil de personalidade das pessoas, o estudo de caso adota o teste de personalidade MBTI, disponibilizado de forma *online* e gratuita pela *Inspiira*⁴, que é fundamentado no conceito de nuvens de palavras para caracterizar as diferentes dimensões MBTI das pessoas. O teste de personalidade foi realizado pelos 30 membros da equipe de desenvolvimento alvo do estudo de caso, representados por $PE = \{pe_1, \dots, pe_{30}\}$. Vale ressaltar que o teste de personalidade da *Inspiira* também é adotado por outros trabalhos (SOUZA, 2016) (BRANCO; PRIKLADNICKI; CONTE, 2012), relacionados à avaliação de personalidades de engenheiros de software.

A partir dos resultados dos testes de personalidade, a implementação da abordagem proposta mapeia de forma automática e matemática o *perfil de personalidade das pessoas*, definido na Equação 5 pelo conjunto $P_{PE} = \{DP_1, \dots, DP_{30}\}$, onde cada termo DP_i representa um vetor de quadro pares de dicotomias MBTI, que conjuntamente sinalizam as dimensões bipolares presentes na pessoa pe_i . Por exemplo, no estudo de caso, a pessoa 1 (pe_1), listada no Apêndice E, com perfil MBTI "ENTJ", possui o vetor de personalidade $DP_1 = [1/-1 \ -1/1 \ 1/-1 \ 1/-1]$.

Em relação ao modelo FFM, a abordagem proposta também necessita que seja fornecida como entrada o mapeamento dos papéis funcionais aos fatores FFM, caracterizando assim o *perfil de personalidade dos papéis*. Para isto, após busca na literatura realizada durante a elaboração desta abordagem, foi adotado por base o mapeamento proposto por Rehman et al. (2012), ilustrado anteriormente na Tabela 1. Vale lembrar, que este mapeamento foi elaborado através da conversão do modelo proposto pelo Capretz e Ahmed (2010) do MBTI para o FFM, utilizando o relacionamento que existe entre ambos.

De maneira similar ao procedimento realizado para a tipologia MBTI, diversas apresentações e discussões foram realizadas com o gerente de projetos para colocá-lo a par dos conceitos da tipologia FFM, e, em seguida, para tentar adaptar o mapeamento do Rehman et al. (2012) à realidade da referida organização. Todavia, ao final das discussões, novamente o gerente de projetos considerou o mapeamento sugerido adequado e não propôs adaptações. Baseado no mapeamento de papéis e dimensões FFM, a implementação da abordagem proposta

⁴ <http://inspiira.org> – *Inspiira Test*

faz o mapeamento matemático automático para o *perfil de personalidade dos papéis*, definido na Equação 8 pelo conjunto $P_{PA} = \{FR_1, \dots, FR_5\}$, onde cada termo FR_j representa um vetor de cinco fatores FFM, que conjuntamente sinalizam os traços de personalidade requeridos pelo papel pa_j . Por exemplo, para o papel de *analista de sistemas* (pa_1) são indicadas pessoas que possuam os traços Extroversão (E) e Amabilidade (A) em nível mais alto (Tabela 1), ou seja, que possuam o vetor de personalidade $FR_1 = [0, 0, 1, 1, 0]$.

Posteriormente, para realizar a coleta do perfil de personalidade das pessoas, o estudo de caso também utiliza o teste de personalidade FFM, disponibilizado de forma *online* e gratuita pelo *Personality Assessor*⁵, que, em sua versão mais curta, consiste de 44 questões. Esta versão é utilizada, em detrimento da versão maior, para não utilizar demasiado tempo dos participantes. O teste de personalidade foi realizado pelos 30 membros da equipe de desenvolvimento alvo do estudo de caso, representados por $PE = \{pe_1, \dots, pe_{30}\}$. De posse dos resultados dos testes de personalidade, a implementação da abordagem proposta realiza um mapeamento matemático de forma automática do *perfil de personalidade das pessoas*, definido na Equação 9 pelo conjunto $P_{PE} = \{FP_1, \dots, FP_{30}\}$, onde cada termo FP_i representa um vetor de cinco fatores FFM, que conjuntamente sinalizam os traços de personalidade possuídos pela pessoa pe_i . Por exemplo, no estudo de caso, a classificação de personalidade da pessoa 1 (pe_1), listada no Apêndice E, é o vetor de personalidade $FP_1 = [38, 35, 67, 45, 32]$.

4.2. Experimentos

Para avaliar a aplicabilidade da abordagem proposta em relação a acuracidade da modelagem matemática, ferramentas protótipos foram desenvolvidas em Java especificamente para o problema avaliado neste trabalho, adotando um algoritmo de busca exaustiva, também conhecida como “Força Bruta”, para identificar todas as soluções possíveis e válidas, definidas pelas diferentes seleções e alocações de pessoas candidatas aos papéis funcionais. Embora sua implementação seja simples, o inconveniente da busca exaustiva é que, para o problema em análise, o espaço de soluções cresce de forma exponencial. Então, um pequeno acréscimo de pessoas ou papéis pode tornar seu uso proibitivo, pois seu custo computacional é proporcional ao número de possíveis soluções.

Diante desta limitação, a abordagem proposta adota metaheurísticas evolucionárias multiobjetivas, que, guiadas por estratégias de busca e operadores genéticos avaliam apenas um

⁵ <http://www.personalityassessor.com/ipip300/> - *Personality Assessor*

pequeno percentual do espaço de busca, recomendando um conjunto de soluções não-dominadas de boa qualidade, ao invés de avaliar todas as soluções possíveis e em seguida recomendar as melhores soluções, reduzindo drasticamente o tempo de processamento e facilitando o processo de tomada de decisão.

No tocante ao cálculo do tamanho do espaço de busca deste problema, considerando um conjunto de n pessoas candidatas ao projeto (PE), as combinações de k elementos de PE para cada papel (pa_j) formam subconjuntos não-ordenados de k elementos diferentes de PE , chamado de coeficiente binomial (MONTGOMERY; RUNGER, 2009), sendo o total de combinações possíveis por papel calculado pela Equação 26 e o espaço de busca total (S) calculado pela Equação 27.

$$C(n, k_{pa_j}) = \binom{n}{k_{pa_j}} = \frac{n!}{(n - k_{pa_j})! k_{pa_j}!} \quad (26)$$

$$S = \prod_{pa_j \in PA} C(n, k_{pa_j}) \quad (27)$$

Originalmente, o estudo de caso contempla 30 candidatos e 5 papéis ($VG = \{3, 3, 7, 4, 3\}$), o que resulta em um gigantesco espaço de busca, que tem cerca de 3.7333731×10^{21} soluções candidatas (calculado pelas Equações 26 e 27). Consequentemente, as execuções dos protótipos do algoritmo de busca exaustiva levariam aproximadamente 14.9×10^9 anos para avaliar completamente todas essas soluções. Por esta razão, as ferramentas protótipos foram adotadas para conduzir uma avaliação com experimentos pilotos, definidos por subconjuntos reduzidos de pessoas e papéis da organização avaliada no estudo de caso. Além disto, foram adotadas metaheurísticas multiobjetivas para avaliação de conjuntos de dados médios e grandes, incluindo o conjunto total de dados da organização do estudo de caso. Os cenários elaborados são definidos da seguinte forma:

- *Cenário I*: representa um projeto cuja implementação, teste e manutenção será contratada via edital público, e, portanto, requer apenas que a organização apresente a análise e projeto do referido sistema, necessitando assim apenas dos papéis de analista de sistemas e projetista de software. Neste caso, o projeto adota cinco pessoas $PE = \{pe_1, \dots, pe_5\}$, que devem ser alocadas nos dois papéis $PA = \{pa_1, pa_2\}$, sendo duas vagas para cada papel $VG = \{2, 2\}$. Considerando as combinações de pessoas, papéis e vagas, o *Cenário I* tem 100 soluções possíveis.

- *Cenário II*: referente a um projeto cujo teste e manutenção será contratada via edital público, requerendo da organização a análise, projeto e implementação do sistema. Neste caso, o projeto possui dez pessoas candidatas $PE = \{pe_1, \dots, pe_{10}\}$, que devem ser alocadas nos três papéis $PA = \{pa_1, pa_2, pa_3\}$ (analista de sistemas, projetista de software e programador) e $VG = \{2, 3, 2\}$. Este conjunto de dados gera 243×10^3 soluções possíveis.
- *Cenário III*: descreve um projeto completo de um sistema de pequeno porte, que adota seis pessoas $PE = \{pe_1, \dots, pe_6\}$, que devem ser alocadas nos cinco papéis $PA = \{pa_1, \dots, pa_5\}$, sendo apenas uma vaga para cada papel $VG = \{1, 1, 1, 1, 1\}$. Considerando as combinações de pessoas, papéis e vagas, o *Cenário III* possui 7.776×10^3 soluções possíveis.
- *Cenário IV*: corresponde a um projeto completo de um sistema de médio porte, que possui 12 pessoas candidatas $PE = \{pe_1, \dots, pe_{12}\}$, que devem ser alocadas nos cinco papéis $PA = \{pa_1, \dots, pa_5\}$, sendo duas vagas para cada papel $VG = \{2, 2, 2, 2, 2\}$. Este experimento lida com cerca de 1.25233×10^9 soluções possíveis.
- *Cenário V*: relativo ao conjunto de dados total da organização, que é constituído de 30 candidatos e 5 papéis ($VG = \{3, 3, 7, 4, 3\}$), conforme já descrito.

Vale lembrar que a numeração identificadora única de cada candidato foi atribuída de acordo com a ordem em que foram preenchidos os formulários que coletaram as informações técnicas e não-técnicas das pessoas. Sendo assim, na composição dos dados dos cenários, as pessoas de identificadores 1 ao número de candidatos definidos nos cenários são as que farão parte do respectivo experimento. É importante destacar que os Cenários de I a III são cenários fictícios criados com o objetivo de testar a abordagem proposta, diferentemente dos cenários IV e V que são configurações de equipes comumente formadas na organização do estudo de caso.

Coletados os dados de entrada dos cenários, as ferramentas protótipos e as metaheurísticas foram executadas nos experimentos, tornando possível a análise multiobjetiva das soluções recomendadas na perspectiva da **avaliação não-técnica** e da **avaliação técnica**. Todos os experimentos foram executados em um microcomputador equipado com um processador de dois núcleos Intel Core i7 7500U de 3.50 GHz com 4MB de memória cache e memória RAM DDR3 com de 8GB de frequência de 667MHz.

4.2.1. Experimentos com busca exaustiva

Nesta direção, uma das técnicas exaustivas de resolução de problemas implementada (*Algoritmo I*) gera, de forma sistemática, todas as S soluções possíveis e, para cada solução gerada, verifica a validade em relação as restrições modeladas pelas Equações 22 e 23. Caso não viole nenhuma das restrições, calcula e imprime os valores *fitness* obtidos através das funções objetivo modeladas pelas Equações 24 e 25. Em suma, o Algoritmo I é modelado da seguinte forma:

Soluções possíveis: Todos os S subconjuntos de k elementos de PE para cada papel (pa_j).

Verificação: Viola alguma das restrições?

Ação: Caso não viole alguma restrição, imprime o subconjunto.

Adicionalmente, para avaliar o comportamento biobjetivo da abordagem proposta, o Algoritmo I foi modificado gerando uma segunda versão de algoritmo de busca exaustiva (*Algoritmo II*). Similarmente ao Algoritmo I, a técnica de resolução de problemas implementada gera, de forma sistemática, todas as S soluções possíveis e, para cada solução gerada, verifica a validade em relação as restrições modeladas pelas Equações 22 e 23 e calcula os valores *fitness* obtidos através das funções objetivo modeladas pelas Equações 24 e 25. Porém, só retorna como saída a fronteira das soluções não-dominadas, por se tratar de uma busca exaustiva, sabe-se que esta fronteira é a fronteira de Pareto. Em suma, o Algoritmo II teve como plataforma de execução o jMetal e é modelado da seguinte forma:

Soluções possíveis: Todos os S subconjuntos de k elementos de PE para cada papel (pa_j).

Verificação: Pertence à fronteira de soluções não-dominadas?

Ação: Caso pertença à fronteira de soluções não-dominadas, imprime o subconjunto.

Vale lembrar que a abordagem proposta suporta os modelos de comportamento MBTI e FFM, sendo o utilizado dependente da classificação das personalidades dos candidatos fornecida como entrada e da escolha do gerente de projetos, o qual deve ser único por execução para todos os avaliados. Sendo assim, serão realizados experimentos para ambos os modelos, os quais serão explicitados a seguir.

4.2.1.1. MBTI

Em relação ao Algoritmo I aplicado ao Cenário I (*Experimento I*), com os dados de entrada das personalidades das pessoas classificados em relação as dicotomias do MBTI, a Figura 28 mostra os resultados da avaliação técnica e da avaliação não-técnica das 30 soluções válidas obtidas. Neste experimento, pode ser observado que a solução válida (verde) que obteve a melhor pontuação na avaliação técnica $AT_X = 2.2267$, também obteve a melhor pontuação na avaliação não-técnica $ANT_X = 8$. É importante frisar que este padrão não é garantido para todos os cenários, pois os objetivos são conflitantes, ou seja, uma solução pode ser boa na perspectiva técnica, mas pode ser ruim na perspectiva não-técnica, ou vice-versa. Por exemplo, na Figura 28 existe uma solução (laranja) que obteve uma pontuação boa na avaliação técnica $AT_X = 2.0024$, no entanto obteve uma pontuação ruim na avaliação não-técnica $ANT_X = -4$.

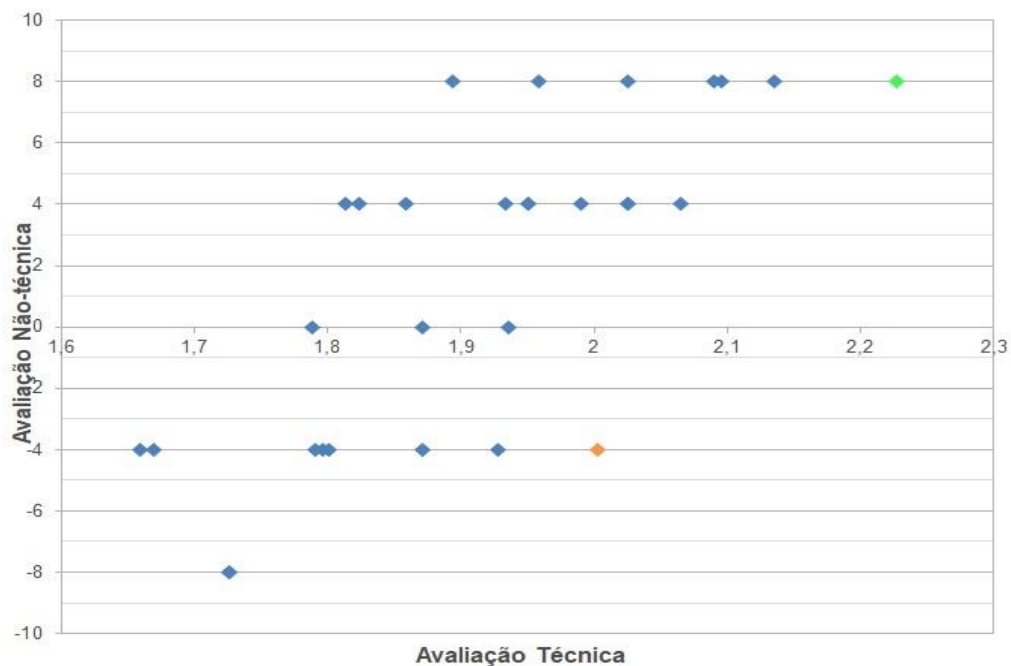


Figura 28 - Resultados do Experimento I com MBTI

Após análise do resultado do Experimento I, utilizando os conceitos de dominância apresentados na seção 2.4.1, pode-se concluir que a fronteira de Pareto possuirá apenas uma única solução (cor verde da Figura 28), pois todas as outras soluções são dominadas por uma ou mais soluções. Buscando confirmar esta constatação foi executado o Algoritmo II com os dados do Cenário I, com o objetivo de visualizar a fronteira de Pareto. Este experimento foi chamado

de *Experimento II* e seu resultado é visualizado na Figura 29. Este resultado confirma a constatação anterior, pois retorna como fronteira de Pareto exatamente a solução esperada. O que mostra que a modelagem adotada e a representação da solução concebidas são capazes de encontrar as soluções não-dominadas do problema.

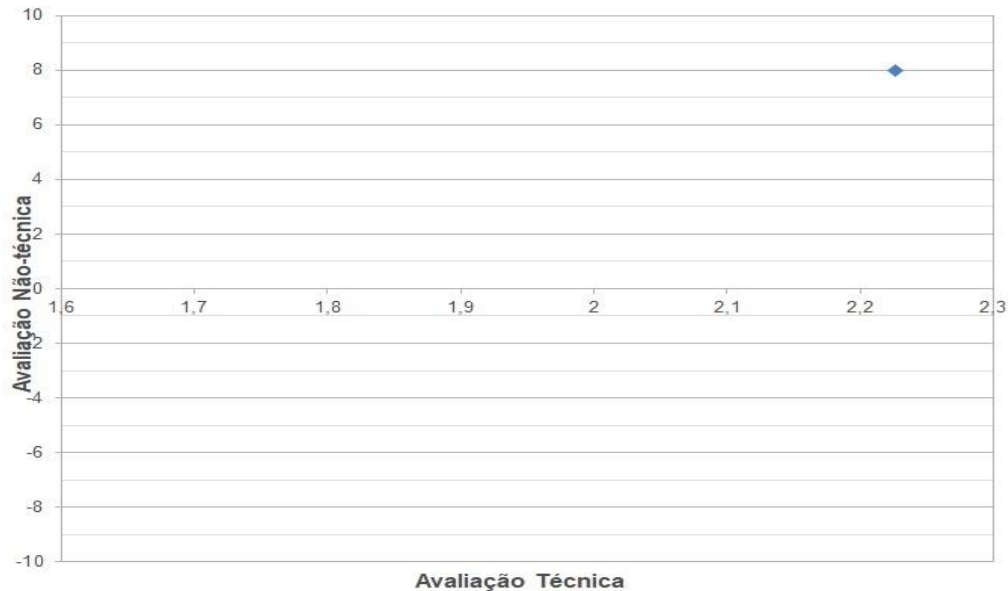


Figura 29 - Resultados do Experimento II com MBTI

Em seguida, desejou-se aumentar o espaço de busca do problema e observar o comportamento apresentado pela abordagem. Para isto, o Algoritmo I foi executado no Cenário II (*Experimento III*), obtendo 25.2×10^3 soluções válidas, presentes na Figura 30. Neste experimento, como pode ser observado, não existe uma solução válida que obteve a melhor pontuação tanto na avaliação técnica quanto na avaliação não-técnica. Por um lado, a solução válida (laranja inferior) que tem a melhor pontuação na avaliação técnica $AT_x = 4.508253969$, não tem a melhor pontuação na avaliação não-técnica $ANT_x = 20$, uma vez que a melhor pontuação não-técnica é $ANT_x = 28$. Por outro lado, a solução válida (laranja superior) que tem a melhor pontuação na avaliação não-técnica $ANT_x = 28$, não tem a melhor pontuação na avaliação técnica $AT_x = 4.32206349$, uma vez que a melhor pontuação técnica é $AT_x = 4.508253969$. Neste caso, não há uma solução vencedora em ambos objetivos simultaneamente. Portanto, cabe ao gerente avaliar e selecionar uma solução boa o suficiente que atenda as demandas e se adeque da melhor forma possível ao projeto e à organização, priorizando um objetivo em detrimento do outro, provavelmente considerando outros aspectos da solução, como, por exemplo, custo, disponibilidade e prazo.

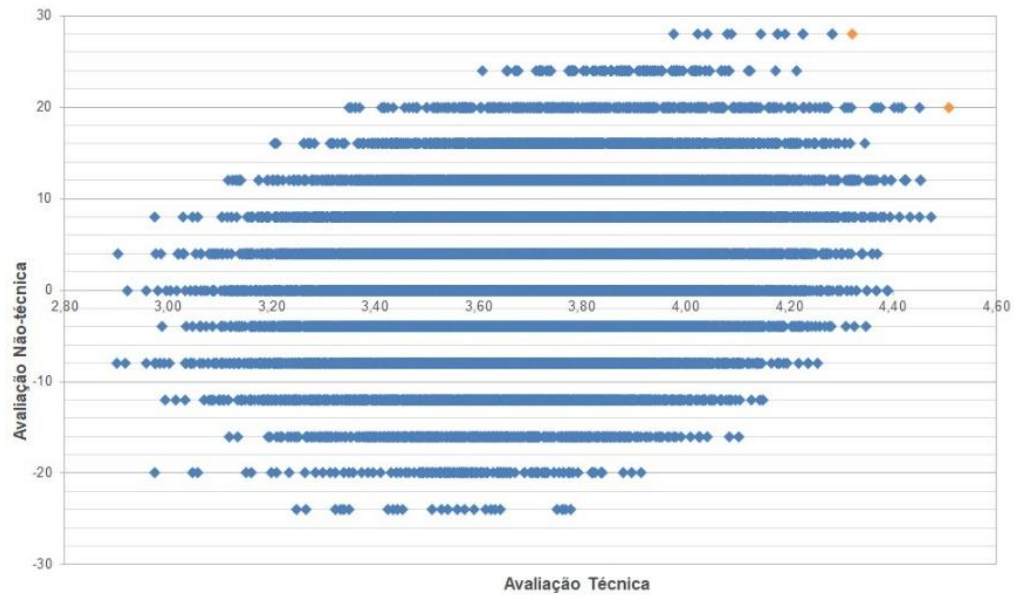


Figura 30 - Resultados do Experimento III com MBTI

Novamente, com o objetivo de verificar a fronteira de Pareto do Cenário II, através de análise da Figura 30, foi executado o Algoritmo II no Cenário II (*Experimento IV*). É esperado que a fronteira de Pareto possua as duas soluções sinalizadas na cor laranja, o que é confirmado no resultado dado como saída da execução do algoritmo, ilustrado na Figura 31.

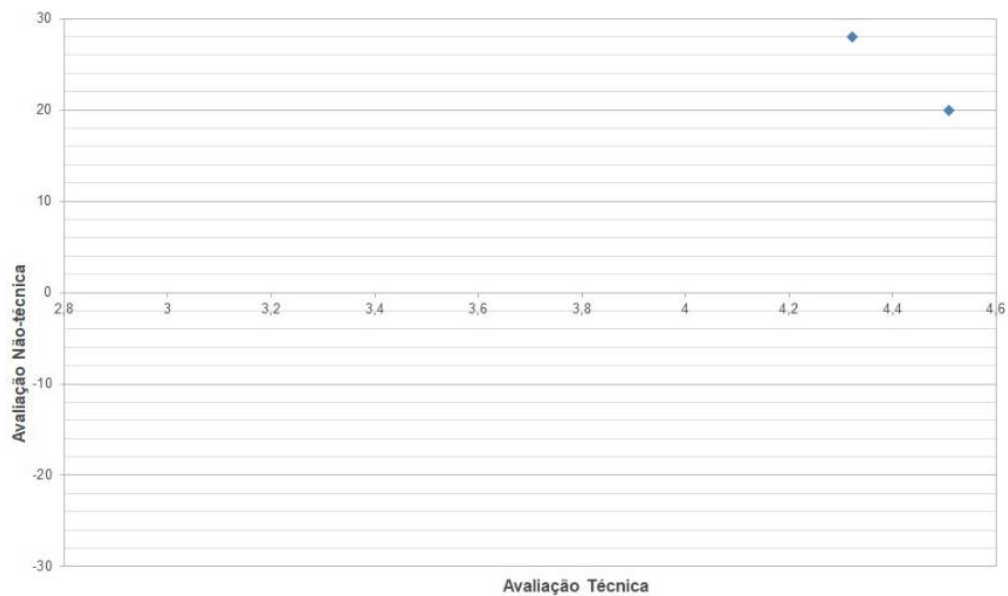


Figura 31 - Resultados do Experimento IV com MBTI

Adicionalmente, a Figura 32 mostra os resultados da avaliação técnica e da avaliação não-técnica do *Experimento V*, que é o Algoritmo I aplicado ao Cenário III, onde foram obtidas 720 soluções válidas. Neste experimento objetivou-se analisar o comportamento dos dados em relação a modelagem de todos os cinco papéis coletados no estudo de caso da organização. Como pode ser observado, similarmente ao Experimento III, não existe uma solução válida que obteve a melhor pontuação tanto na avaliação técnica quanto na avaliação não-técnica. Por um lado, a solução válida (laranja inferior) que tem a melhor pontuação na avaliação técnica $AT_X = 2.8634$, não tem a melhor pontuação na avaliação não-técnica $ANT_X = 2$. Por outro lado, a solução válida (laranja superior) que tem a melhor pontuação na avaliação não-técnica $ANT_X = 10$, não tem a melhor pontuação na avaliação técnica, pois obteve $AT_X = 2.7548$. Ambas soluções também foram apontadas como pertencentes à frente de Pareto, conforme ilustrado na Figura 33, que representa o resultado do *Experimento VI* (Algoritmo II executado no Cenário III).

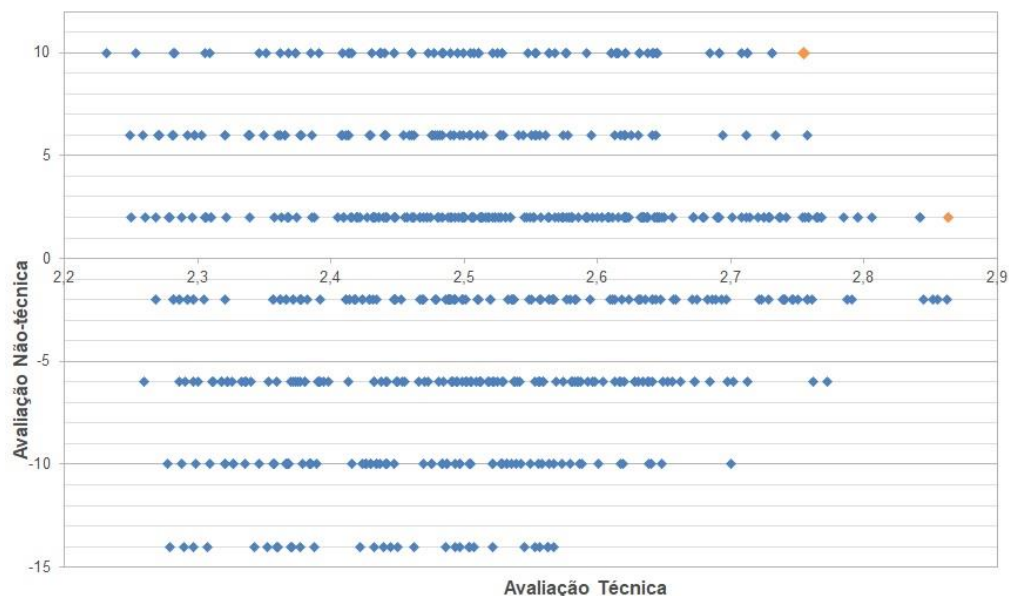


Figura 32 - Resultados do Experimento V com MBTI

Finalmente, em relação ao Cenário IV, o Algoritmo I foi executado (*Experimento VII*), retornando 7.4844×10^6 soluções válidas. Devido a quantidade destas soluções retornadas, não foi possível a elaboração de um gráfico que visualizasse todas elas, de forma similar ao realizado nos cenários anteriores. Entretanto, através da execução do Algoritmo II (*Experimento VIII*) foi possível obter o conjunto ótimo de Pareto, formado por 9 soluções não-dominadas, ilustrado na Figura 34. Vale ressaltar que esta fronteira de Pareto será usada para comparação, em seções posteriores, com as fronteiras retornadas pelas metaheurísticas multi-

objetivas. Vale lembrar que não é computacionalmente viável executar o algoritmo de busca exaustiva no Cenário V.

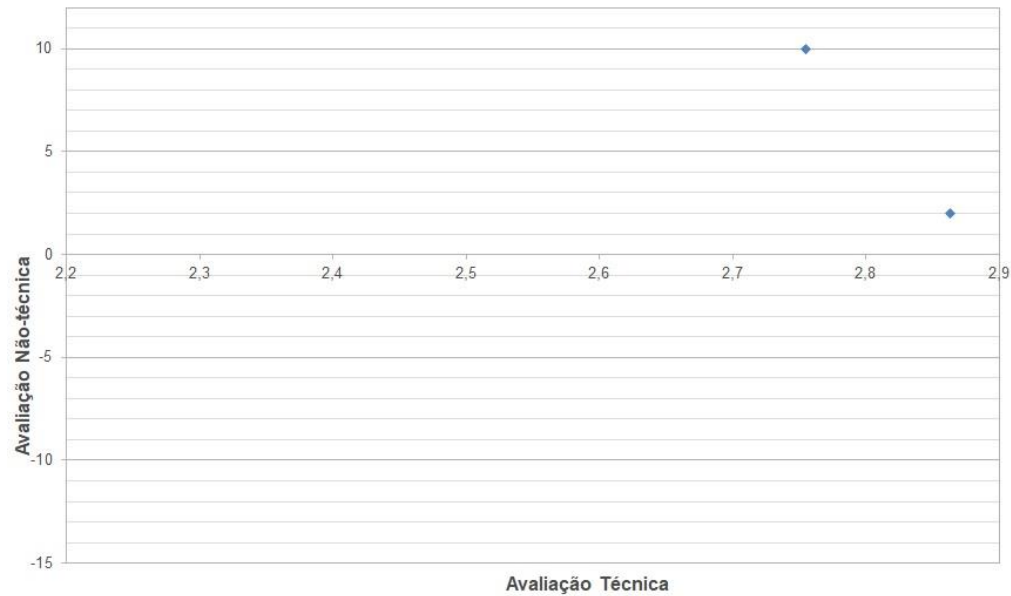


Figura 33 - Resultados do Experimento VI com MBTI

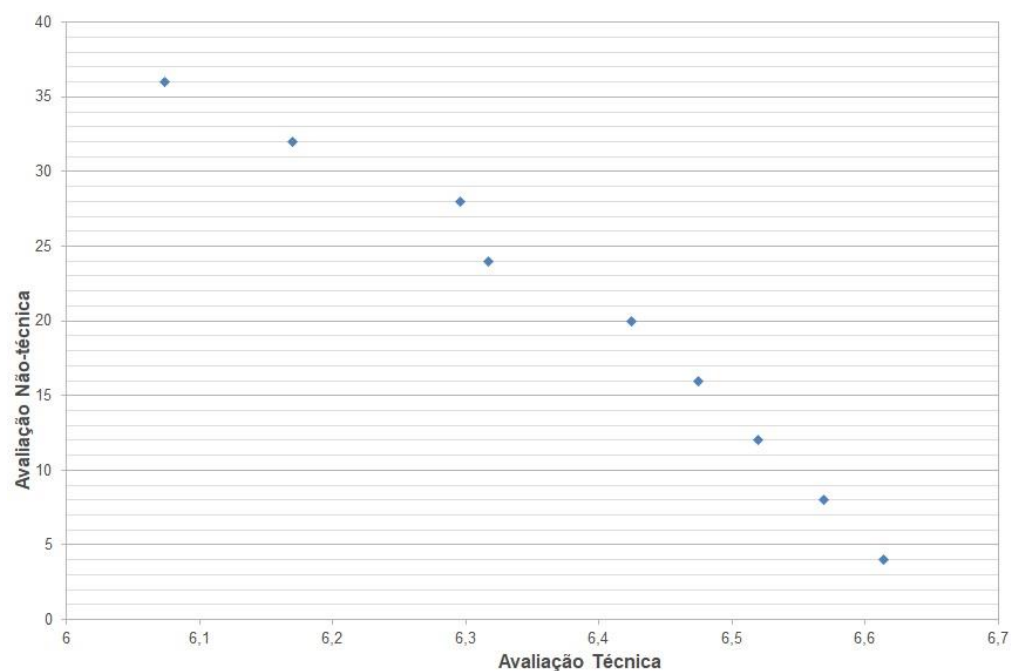


Figura 34 - Resultados do Experimento VIII

Em suma, após as análises de todos estes experimentos, as seguintes observações podem ser destacadas: (i) os resultados apresentam características típicas de problemas multiobjetivos, neste caso com objetivos conflitantes em relação aos aspectos técnicos e não-técnicos da seleção e alocação de pessoas a papéis; e (ii) a modelagem mostra ser capaz de mensurar de

forma intuitiva a qualidade das soluções em relação aos objetivos avaliados, pois o tomador de decisão pode, através de inspeção visual do gráfico, comparar as soluções não-dominadas apresentadas como resultado.

4.2.1.2. FFM

De forma bastante similar ao apresentado na seção anterior, esta seção detalha as execuções dos experimentos da abordagem proposta utilizando os dados das classificações de personalidades dos candidatos realizadas com o modelo FFM. Sendo assim, os cenários, os algoritmos exaustivos utilizados, os candidatos, os valores *fitness* da função objetivo técnica (AT_x) e as quantidades de soluções válidas obtidas são os mesmos dos experimentos realizados com o MBTI. Por esta razão não serão detalhados em demasia. Entretanto, os valores *fitness* da função objetivo não-técnica (ANT_x) são diferentes e apresentam um comportamento distinto conforme será relatado a seguir.

Em relação ao Algoritmo I aplicado ao Cenário I (*Experimento IX*), com os dados de entrada das personalidades das pessoas classificados em relação aos fatores FFM, a Figura 35 mostra os resultados da avaliação técnica e da avaliação não-técnica das 30 soluções válidas obtidas. Neste experimento, não existe uma solução válida que obteve a melhor pontuação tanto na avaliação técnica quanto na avaliação não-técnica. Por um lado, a solução válida (laranja inferior) que tem a melhor pontuação na avaliação técnica $AT_x = 2.226666667$, não tem a melhor pontuação na avaliação não-técnica, pois obteve $ANT_x = 0.20$. Por outro lado, a solução válida (laranja superior) que tem a melhor pontuação na avaliação não-técnica $ANT_x = 1.06$, não tem a melhor pontuação na avaliação técnica, já que pontuou $AT_x = 1.796666667$. Porém, diferentemente dos casos avaliados anteriormente, estas duas soluções pontuam muito bem em um aspecto, mas muito mal no outro aspecto, permitindo que outras soluções se mostrem atraentes, que é o caso das soluções sinalizadas na cor preta.

Nesta direção, analisando o resultado do Experimento IX e considerando a definição de dominância, pode-se concluir que a fronteira de Pareto possuirá as soluções não-dominadas marcadas nas cores preta e laranja na Figura 35. Para constatar esta afirmação, foi executado o Algoritmo II com os dados do Cenário I. Este novo experimento foi chamado de *Experimento X* e seu resultado é visualizado na Figura 36. Este resultado contém as soluções esperadas, pois são as soluções não-dominadas, o que comprova que a implementação algorítmica de busca pelas

soluções não-dominadas, aplicada a abordagem concebida para o FFM se comportou como esperado.

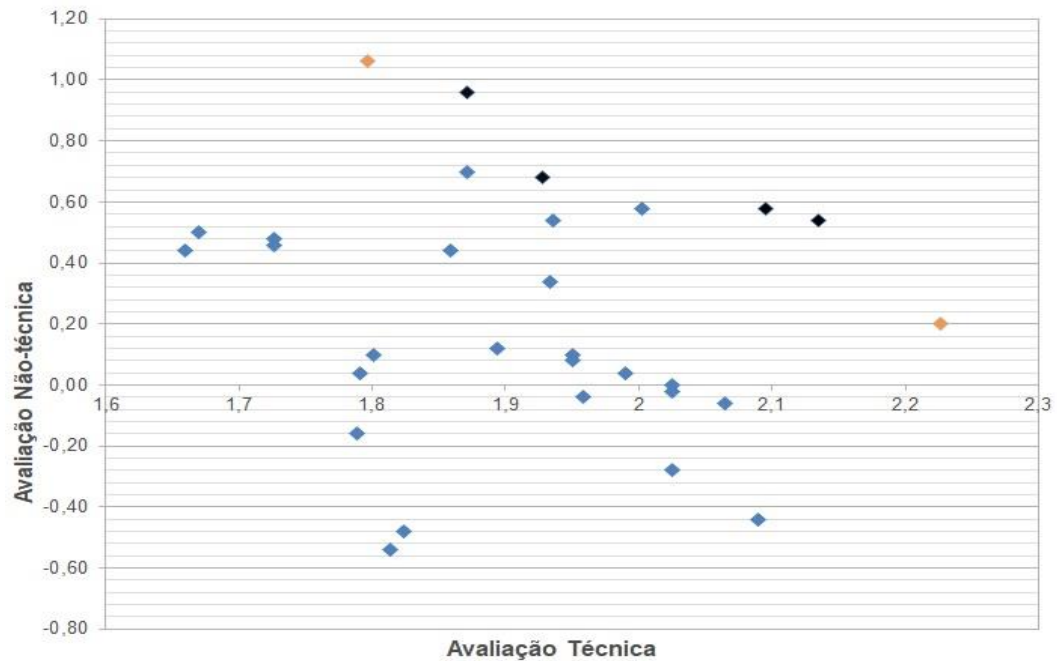


Figura 35 - Resultados do Experimento IX com FFM

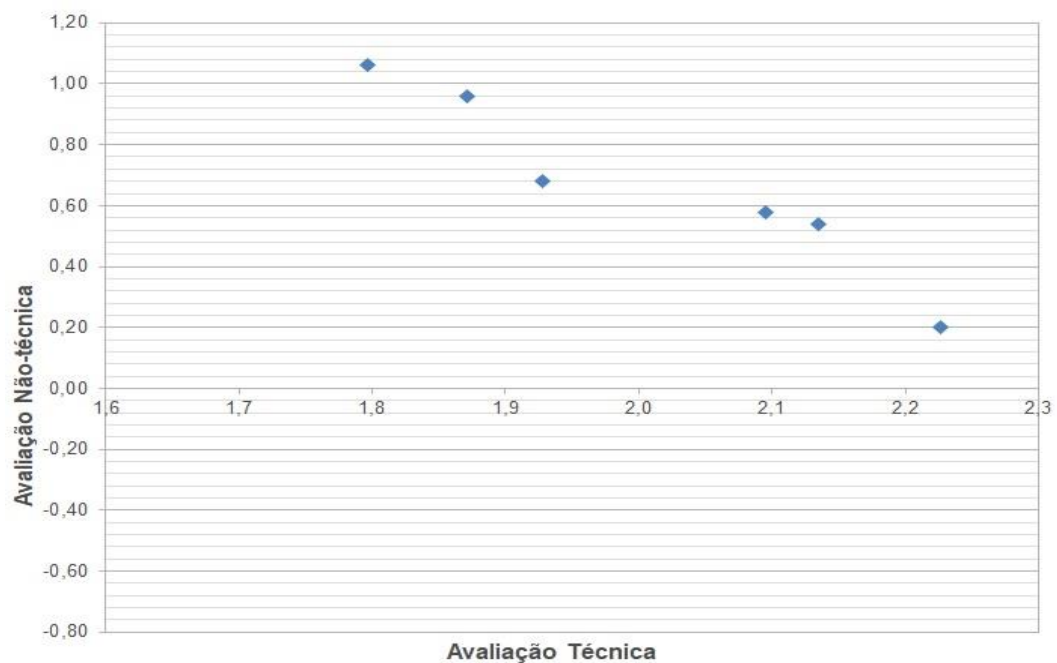


Figura 36 - Resultados do Experimento X com FFM

Em seguida, o Algoritmo I foi executado no Cenário II (*Experimento XI*), obtendo 25.2×10^3 soluções válidas, apresentadas na Figura 37. Entretanto, para este experimento, a quantidade de soluções válidas e o comportamento dos dados retornados pela modelagem

dificulta a determinação de quais soluções são não-dominadas apenas pela inspeção visual do gráfico. Porém, neste ponto das execuções dos experimentos, o Algoritmo II já provou a sua acuracidade na busca pela frente de Pareto, sendo os resultados da sua execução (*Experimento XII*) ilustrados na Figura 38.

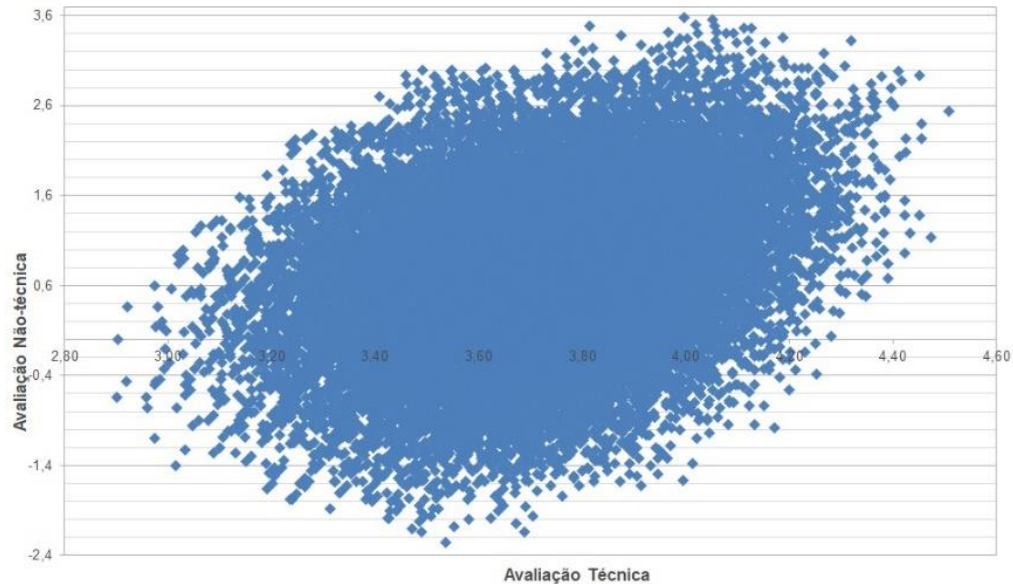


Figura 37 - Resultados do Experimento XI

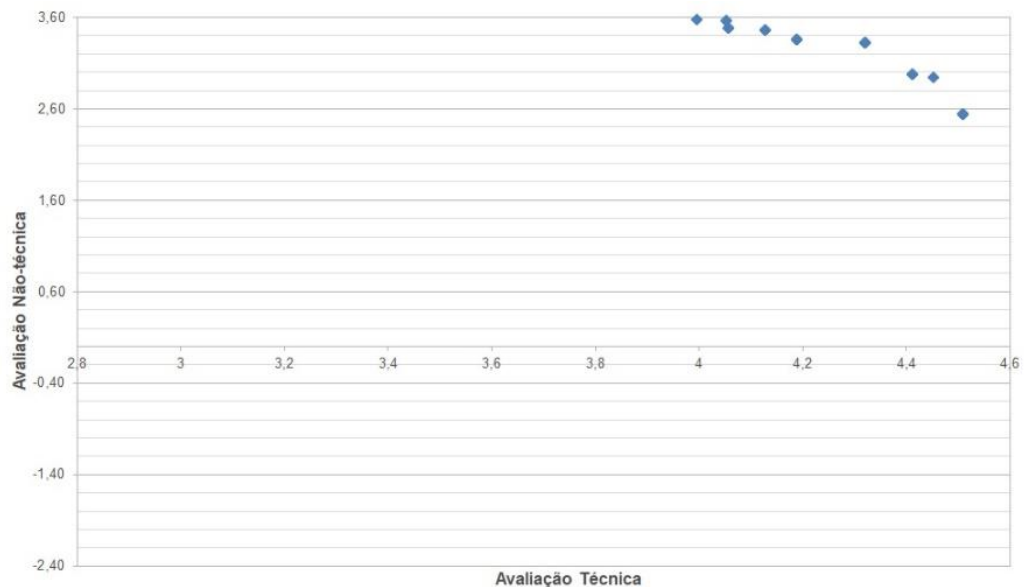


Figura 38 - Resultados do Experimento XII com FFM

Outro experimento (*Experimento XIII*) executado mostra os resultados da avaliação técnica e da avaliação não-técnica da execução do Algoritmo I aplicado ao Cenário III, ilustrado na Figura 39. Visto que este experimento obteve 720 soluções válidas e objetiva analisar o

comportamento dos dados em relação a modelagem de todos os cinco papéis coletados no estudo de caso da organização. A execução do Algoritmo II neste cenário retorna o resultado ilustrado na Figura 40.

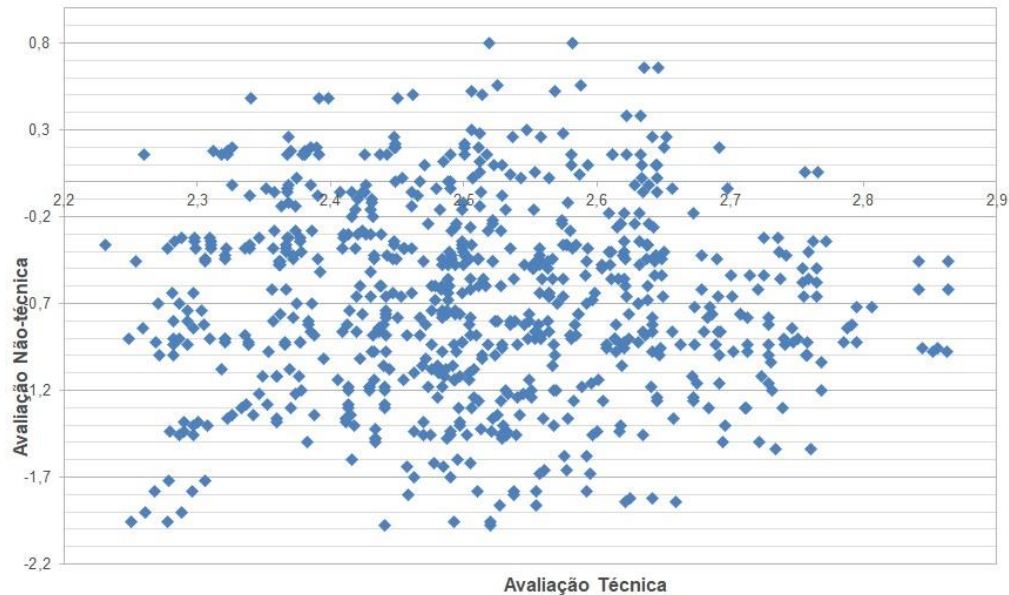


Figura 39 - Resultados do Experimento XIII com FFM

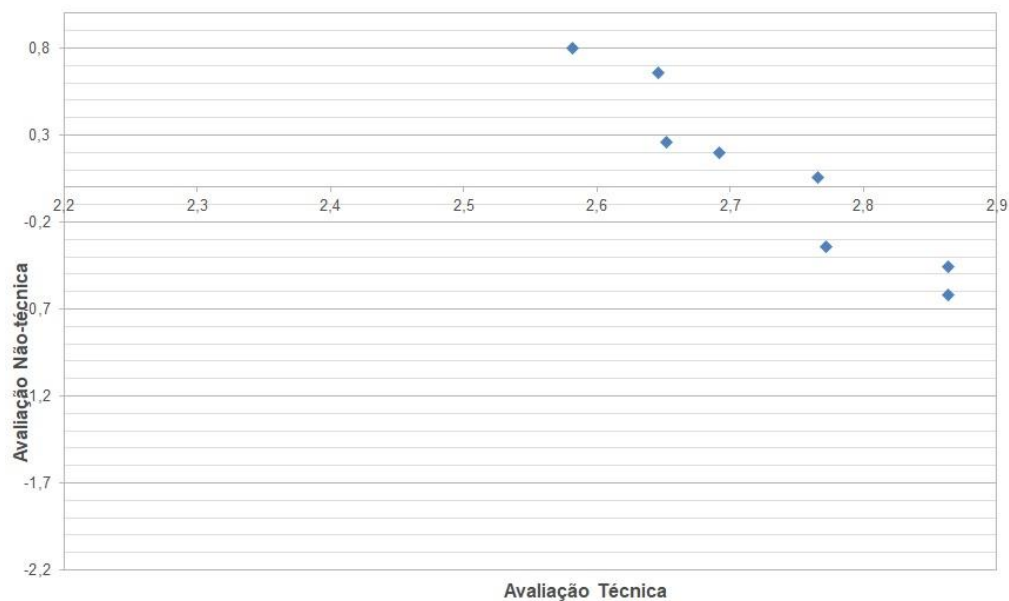


Figura 40 - Resultados do Experimento XIV com FFM

Por último, em relação ao Cenário IV, o Algoritmo I foi executado (*Experimento XV*), retornando 7.4844×10^6 soluções válidas, não sendo possível a elaboração de um gráfico que visualizasse todas elas. Porém, através da execução do Algoritmo II (*Experimento XVI*) foi possível obter a frente ótima de Pareto, formada por 19 soluções não-dominadas, ilustrada na Figura 41. Vale ressaltar que esta fronteira de Pareto será usada para comparação, em seções

posteriores, com as fronteiras retornadas pelas metaheurísticas multiobjetivas. Também é importante destacar que não é computacionalmente viável executar o algoritmo de busca exaustiva no Cenário V.

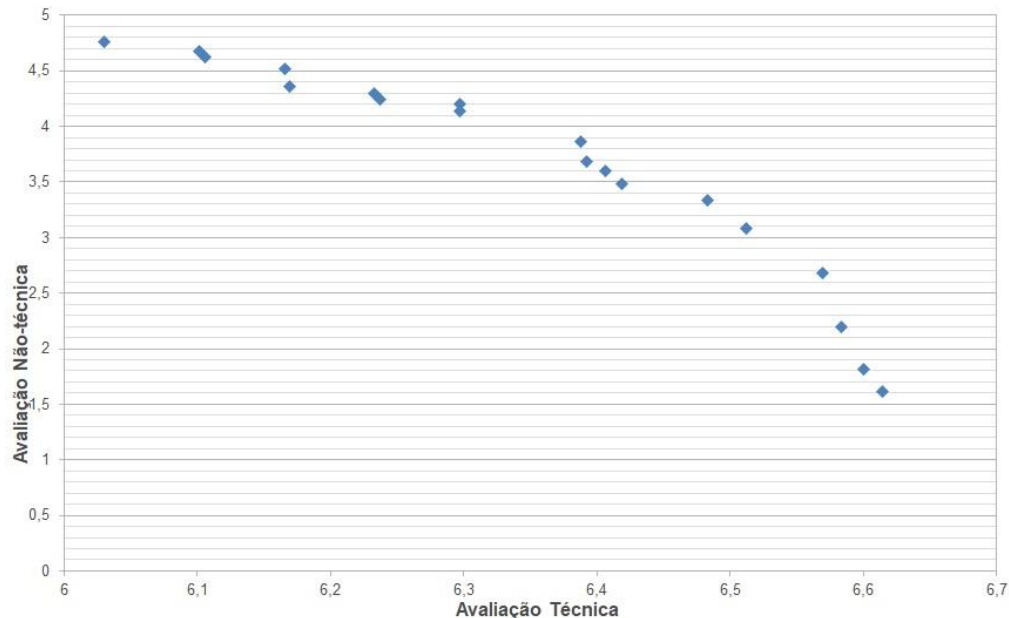


Figura 41 - Resultados do Experimento XVI

Após as análises destes experimentos, de forma similar ao constatado pela modelagem MBTI, pode ser observado que os resultados se apresentam de maneira típica de problemas multiobjetivos, neste caso com objetivos conflitantes quanto aos aspectos técnicos e não-técnicos da seleção e alocação de pessoas a papéis. Além disso, a modelagem FFM também provou ser capaz de mensurar de forma intuitiva a qualidade das soluções em relação aos objetivos avaliados. Conclui-se que o modelo matemático proposto é implementado de acordo com o que é esperado de suas equações formuladas.

4.2.2. Experimentos com metaheurísticas

Os experimentos com os algoritmos de busca exaustiva relatados na seção anterior se mostraram bastante promissores, pois demonstraram que o modelo apresentado se comporta conforme descrito no Capítulo 3. Entretanto, como já mencionado, não são aplicáveis a problemas com espaços de busca maiores. Para sanar esta limitação, a abordagem proposta adota, de maneira customizável, uma das seguintes metaheurísticas evolucionárias multiobjetivas: MOCell, NSGA-II e SPEA2. Deste modo, nesta seção, são realizados experimentos com estas metaheurísticas e os seus desempenhos são avaliados e comparados entre si e também com o algoritmo de busca aleatória. Esta comparação com os resultados

obtidos randomicamente é chamada de “teste de sanidade” (HARMAN *et al.*, 2012) e consiste em demonstrar que a estratégia de busca de uma metaheurística é capaz de fornecer um resultado suficientemente superior. Para confirmar que os algoritmos superaram satisfatoriamente os resultados obtidos pelo algoritmo de busca aleatória, é utilizado o teste estatístico de Wilcoxon-Mann-Whitney (FAY; PROSCHAN, 2009) com um valor-*p* igual a 0.05, denotando um grau de confiança de 95%.

Vale salientar que a plataforma de execução de todos estes algoritmos (busca aleatória, busca exaustiva, NSGA-II, SPEA2 e MOCeII) foi o jMetal e todos foram codificados em Java. Cada algoritmo, inclusive o aleatório, foi executado 100 vezes em cada experimento. Estas execuções foram alocadas em 4 threads concorrentes, correspondente ao limite do processador.

Como já relatado no Capítulo 3, a primeira etapa dos algoritmos evolucionários é a geração da população inicial. Após experimentos utilizando a implementação padrão do procedimento desta fase, foi constatado que o mesmo não demonstrou desempenho satisfatório para a modelagem desta proposta, que lida com um espaço de busca com muitas soluções inválidas, o que acarretava maior tempo para a identificação de boas soluções válidas. Por esta razão, este algoritmo padrão disponível no jMetal foi modificado para gerar apenas indivíduos válidos para a população inicial. É importante ressaltar que uma solução somente é válida quando atende as restrições do problema.

Além disso, os algoritmos evolucionários utilizam operadores de seleção, cruzamento, mutação e substituição. Sendo assim, as metaheurísticas utilizadas neste trabalho necessitam da implementação e parametrização destes operadores. Nesta direção, o operador de seleção não utiliza nenhum parâmetro e a implementação disponível no jMetal foi utilizada sem modificações. Para isto, as três metaheurísticas foram configuradas para utilizarem a técnica de seleção por torneio, por ser considerada como padrão definido na implementação do NSGA-II (AMARAL, 2017), facilitando uma comparação entre os demais algoritmos, além de apresentar vantagens em relação as demais técnicas. Por exemplo, a técnica de seleção da roleta possui a desvantagem de alocar um espaço muito grande para os indivíduos de maior *fitness*, fazendo com que o algoritmo adote um viés muito elitista especialmente nos casos em que existe uma grande variância no *fitness* da população. Já a seleção por *ranking*, apesar de amenizar este comportamento, ainda não o resolve, pois os cromossomos de maior *fitness* possuem grandes probabilidades de seleção. Em contrapartida, a seleção por torneio realiza inicialmente uma seleção aleatória de pares de cromossomos, e, em seguida, cada par selecionado é comparado em relação ao seu *fitness*, sendo então escolhido o indivíduo melhor avaliado. Portanto, existe

uma possibilidade de que este par comparado contenha dois pais de *fitness* baixo, devido à aleatoriedade da primeira etapa da seleção.

Os operadores de cruzamento disponíveis do jMetal não demonstraram desempenho satisfatório em relação a modelagem concebida nesta abordagem, devido as especificidades do espaço de busca. Desta forma, modificações na implementação padrão do *crossover* uniforme foram necessárias. Em suma, esta versão modificada realiza o cruzamento assegurando gerar apenas cromossomos válidos, com pessoas selecionadas uma única vez e papéis com todas as vagas preenchidas. De maneira similar ao *crossover* padrão, uma máscara binária do tamanho dos cromossomos é gerada, e, em seguida, cada bit é verificado. Caso o bit seja 0, o gene da respectiva posição do filho 1 recebe uma cópia do valor do gene do pai 1 e o filho 2 recebe uma cópia do gene do pai 2, ou vice-versa. Porém este procedimento só ocorre desta forma padrão caso não gere filhos inválidos, caso contrário, a posição que receberia o identificador de uma pessoa já alocada, recebe um alelo ainda não presente neste cromossomo (identificador de uma pessoa livre), conforme ilustrado na Figura 42.

Para possibilitar um melhor entendimento do cruzamento, o exemplo explanado na Figura 42 deve ser analisado em detalhes. Nesta figura, após a geração aleatória da máscara binária, cada gene dos cromossomos pais escolhidos para reprodução (pai 1 e pai 2) é avaliado em relação a respectiva posição da máscara. A primeira posição da máscara é 1, o que determina que o gene 1 do filho 1 receberá o gene do pai 2 e vice-versa. O contrário acontece em relação a segunda posição da máscara que é o bit 0, ou seja, posição 2 do filho 1 receberá o gene do pai 1 e vice-versa. Até este ponto ocorreu o comportamento padrão, contudo um comportamento diferente ocorre para a posição 3 em relação ao filho 1, nesta posição a máscara é o bit 1, porém a pessoa que está representada no gene 3 do pai 2 já foi alocada neste cromossomo, o que impede novamente a sua atribuição. Por isto, esta posição do cromossomo do filho 1 recebe um alelo escolhido aleatoriamente pertencente ao conjunto de alelos não utilizados. Em seguida, este conjunto de não utilizados é atualizado removendo o alelo escolhido, de forma a não permitir uma atribuição duplicada em outro gene posterior. Entretanto, o comportamento padrão do cruzamento ocorre para o gene 3 do filho 2.

A probabilidade de 90% foi fixada para o operador de cruzamento após a realização de testes, onde a probabilidade de 100% apresentou piores resultados, pois, caso o operador de cruzamento não seja acionado, os filhos gerados serão cópias exatas dos pais, sendo em seguida submetidos ao operador de mutação, que pode melhorar uma solução pré-selecionada.

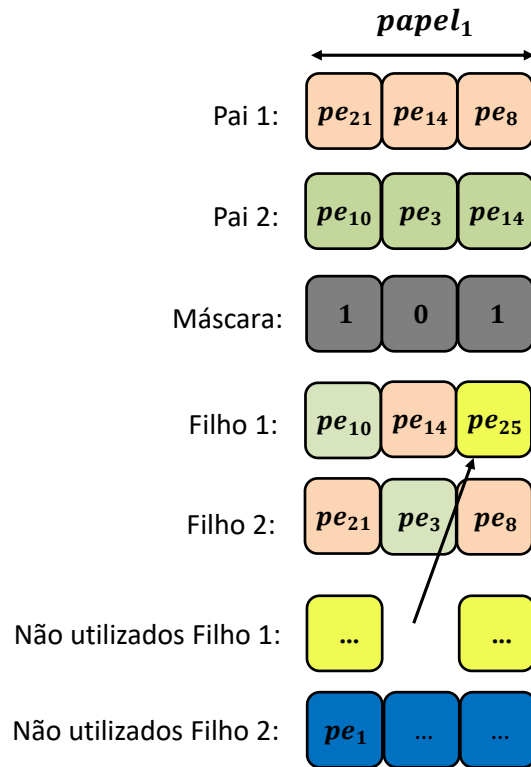


Figura 42 - Versão modificada do cruzamento uniforme

Após o cruzamento, ocorre a mutação com a finalidade de trazer mais diversidade à população ao longo das gerações futuras, através da inserção de indivíduos mutantes. Neste contexto, o operador de mutação foi implementado de forma a manter a validade da solução mutante, ou seja, não gerar soluções que violem uma das restrições. Com uma probabilidade de $1/n$, em que n corresponde ao total de vagas em todo o projeto, somadas todas as quantidades de vagas dos papéis, cada gene é testado, e, caso seja selecionado para sofrer a mutação, é escolhido aleatoriamente outro identificador de pessoa entre os valores dos genes de outros papéis ou outro identificador ainda não utilizado no cromossomo. Para as situações que outro gene de outro papel é selecionado para sofrer a mutação, os valores são trocados entre os papéis envolvidos, comportamento semelhante à técnica *swap* (EIBEN; SMITH, 2013).

Neste ponto é necessária a validação da abordagem do ponto de vista da otimização multiobjetiva. Para isto, foram realizados experimentos com os cenários anteriormente descritos. Lembrando que os cenários I a III foram elaborados de maneira a possibilitar a avaliação de todo o espaço de busca. Enquanto que os cenários IV e V correspondem a configurações de projetos que ocorrem frequentemente na organização. Por isto, esta seção focou no detalhamento da realização de experimentos com os cenários IV e V, porém na prática foram executados experimentos utilizando metaheurísticas para outros cenários elaborados,

obtendo em todos eles as fronteiras de Pareto em tempo muito menor que o gasto da forma exaustiva.

Os experimentos foram avaliados utilizando os indicadores de qualidade de hipervolume normalizado (HVR) e espalhamento (Δ), descritos anteriormente no Capítulo 3. Vale lembrar que estes indicadores de qualidade exigem algum conhecimento a respeito da Fronteira de Pareto ou, pelo menos, de uma fronteira de referência com as melhores soluções conhecidas. Sendo assim, nos cenários em que foi possível a execução do algoritmo exaustivo, a Fronteira de Pareto é conhecida. Porém, caso contrário, foi possível apenas a obtenção de uma fronteira referência. Ou seja, para o cenário IV, foi possível a execução do algoritmo de busca exaustiva (Algoritmo II), portanto o conjunto de soluções não-dominadas da fronteira de Pareto se tornou conhecida. Diferentemente, para o cenário V, não foi possível a execução de um algoritmo exaustivo, sendo assim, as metaheurísticas foram executadas inúmeras vezes de forma a encontrar uma fronteira de referência representativa.

4.2.2.1. MBTI

Primeiramente, foram executados experimentos utilizando a classificação de personalidade MBTI para as entradas da etapa Avaliação do Perfil de Personalidade da abordagem deste trabalho (apresentada no Capítulo 3). Destes, o *Experimento XVII* se refere à execução das três metaheurísticas no Cenário IV, enquanto o *Experimento XVIII* é relativo à execução destas no Cenário V.

Os algoritmos evolucionários foram parametrizados conforme descrito na Tabela 5, cujos valores foram obtidos através de sucessivas calibrações. Lembrando que as saídas dos algoritmos SPEA2 e MOCell não são as populações, mas sim os arquivos de soluções não-dominadas. Por isto eles possuem um parâmetro adicional chamado *arquivo*, o qual deve ser preenchido com as soluções não-dominadas encontradas. Para prover ao NSGA-II condições justas de comparação com os demais algoritmos, a configuração deste parâmetro do SPEA2 e do MOCell será definida com o tamanho igual ao da população regular. O *feedback*, por sua vez, é um parâmetro exclusivo do MOCell, que configura a substituição aleatória de soluções da população por soluções do arquivo. O algoritmo aleatório é parametrizado apenas com a quantidade de soluções a serem avaliadas, que é equivalente ao tamanho da população multiplicada pela quantidade de gerações dos demais algoritmos.

Tabela 5 - Parametrização dos experimentos com metaheurísticas (MBTI)

	Experimento XVII	Experimento XVIII
População	20	40
Arquivo	20	40
Feedback	4	4
Gerações	1250	5000
Soluções avaliadas	25000	200000

Vale salientar que o MOCell não foi capaz de executar o Experimento XVIII com uma população igual a 40 cromossomos, ocasionando um erro de posição inválida de *array* ao buscar as soluções na vizinhança. Por esta razão, a população do MOCell para este experimento foi aumentada para 42 indivíduos, o que não prejudica a saída do algoritmo, nem a comparação com as demais metaheurísticas, pois todos os algoritmos exploram a mesma quantidade de soluções no espaço de busca. Após inspeção no código, feita por Amaral (2017), foi verificado um erro no cálculo de vizinhança ao gerar a população inicial. De forma análoga à Amaral e Elias (2018), decidiu-se não modificar a implementação do algoritmo pré-existente no jMetal, evitando introduzir bugs ou omitir possíveis resultados de comparações e validações.

Após a execução dos experimentos, os resultados obtidos foram compilados na Tabela 6. Pode ser observado que todos os três algoritmos obtiveram uma performance significativamente melhor que o algoritmo aleatório, em todos os aspectos, exceto no tempo de execução. Mas, tomando como exemplo a execução do NSGA-II no Experimento XVII, que durou apenas 5 segundos a mais que a busca aleatória e obteve o melhor desempenho para os parâmetros de hipervolume normalizado (HVR) e espalhamento (Δ), esta diferença de tempo quase imperceptível não justificaria o uso de um algoritmo com performance muito pior. Ademais, lembrando do Experimento VIII, que executou o algoritmo exaustivo no mesmo cenário do Experimento XVII e encontrou a Fronteira de Pareto com todos os 9 valores ótimos em 52 minutos, uma metaheurística capaz de se aproximar destes valores em cerca de 8 segundos é um resultado bastante interessante.

Comparando as três metaheurísticas avaliadas em relação ao tempo de execução, pode ser notado que o SPEA2 obteve os piores resultados em ambos os experimentos, com um desempenho muito mais lento em contraste com o NSGA-II e MOCell. A justificativa para este comportamento, também observado em Amaral (2017), é a complexidade de $O(M^2 \log M)$, em que M é a soma da população regular e do arquivo (ZITZLER, LAUMANN e THIELE, 2001). Já o NSGA-II possui uma complexidade $O(MN^2)$, onde M é a quantidade de gerações e N é o

tamanho da população (DEB et al., 2002). No caso do MOCcell, os autores não especificaram a complexidade do algoritmo (NEBRO et al., 2009), mas visto que o operador de dominância é semelhante ao utilizado pelo NSGA-II, Amaral (2017) inferiu que a complexidade deste algoritmo se aproxima do NSGA-II, conforme demonstrado nos resultados.

Segundo Durillo *et. al.* (2009), fronteiras cujo valor de hipervolume normalizado se aproximam de 1 (valor máximo para o indicador) são mais desejadas, pois indicam que são iguais ou muito próximas da fronteira dada como referência. Ao contrário do espalhamento, em que se almeja valores mais próximos de 0 (valor mínimo para o indicador), indicando uma distribuição mais uniforme. Sendo assim, no Experimento XVII, o NSGAII obteve os melhores valores nos indicadores HVR e tempo de execução, seguido do MOCcell e SPEA2, respectivamente. Entretanto, em relação ao indicador espalhamento, o MOCcell apresentou melhor valor, seguido do NSGA-II e SPEA2. Em contrapartida, no Experimento XVIII, os algoritmos listados em ordem decrescente em relação aos desempenhos nos indicadores HVR e espalhamento são: NSGA-II, SPEA2 e MOCcell. Porém em relação ao indicador tempo de execução, o NSGAII apresentou melhor desempenho, seguido do MOCcell e SPEA2.

Tabela 6 - Hipervolume normalizado, espalhamento e tempo de execução (MBTI)

	NSGA-II	SPEA2	MOCcell	Busca aleatória
HIPERVOLUME NORMALIZADO. Média e desvio padrão				
Experimento XVII	1.00e+00 _{1.4e-03}	9.99e-01 _{2.2e-03}	9.99e-01 _{1.9e-03}	2.63e-01 _{8.5e-02}
Experimento XVIII	9.91e-01 _{8.7e-03}	9.91e-01 _{9.1e-03}	9.91e-01 _{1.0e-02}	7.81e-02 _{4.3e-02}
ESPALHAMENTO. Média e desvio padrão				
Experimento XVII	1.68e-01 _{1.1e-02}	1.88e-01 _{1.6e-02}	1.61e-01 _{1.8e-02}	6.48e-01 _{1.1e-01}
Experimento XVIII	3.45e-01 _{3.6e-02}	3.50e-01 _{4.2e-02}	3.73e-01 _{4.8e-02}	7.97e-01 _{9.6e-02}
TEMPO DE EXECUÇÃO				
Experimento XVII	0'08"	0'13"	0'12"	0'03"
Experimento XVIII	1'45"	3'09"	2'54"	0'44"

Os gráficos com as saídas dos Experimentos XVII e XVIII são ilustrados nas Figuras 43 e 44, respectivamente. Estas saídas representam as fronteiras com melhores valores de HVR dentre as 100 resultantes das 100 execuções. Observa-se a dificuldade em distinguir visualmente os resultados das três metaheurísticas em relação ao Pareto Ótimo (Experimento

XVII) ou em relação à fronteira de referência (Experimento XVIII). Já que, por um lado, as fronteiras retornadas por todas as metaheurísticas do Experimento XVII possuem todas as 9 soluções da fronteira de Pareto. Por outro lado, as fronteiras das metaheurísticas do Experimento XVIII possuem todas as 17 soluções da fronteira de referência. Diferentemente das fronteiras resultantes do algoritmo aleatório, que, em ambos experimentos, estão distantes das fronteiras tomadas como referências.

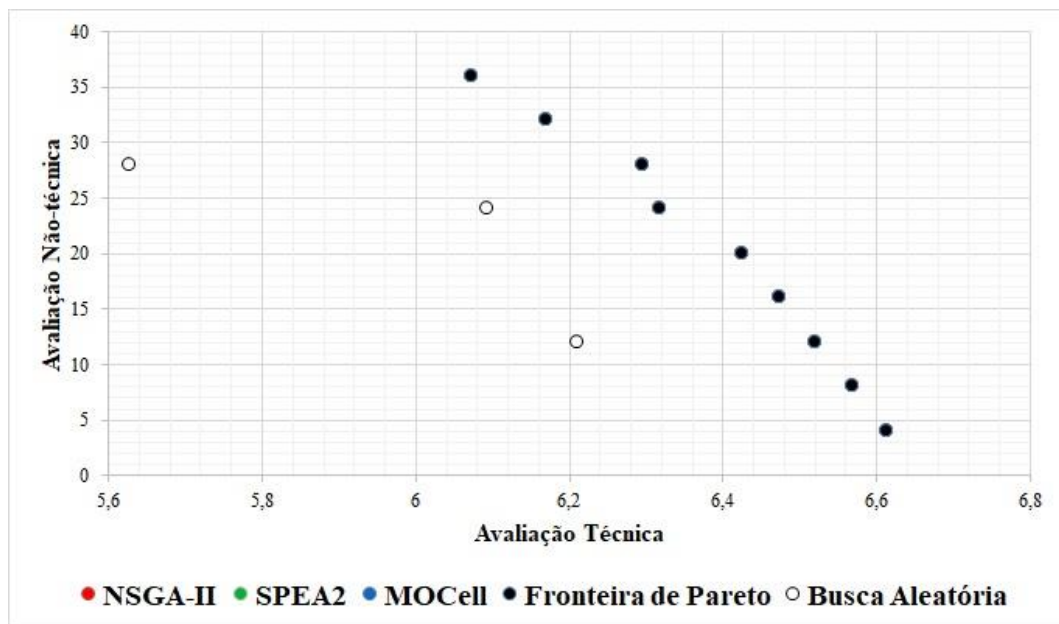


Figura 43 - Resultados para o Experimento XVII com MBTI

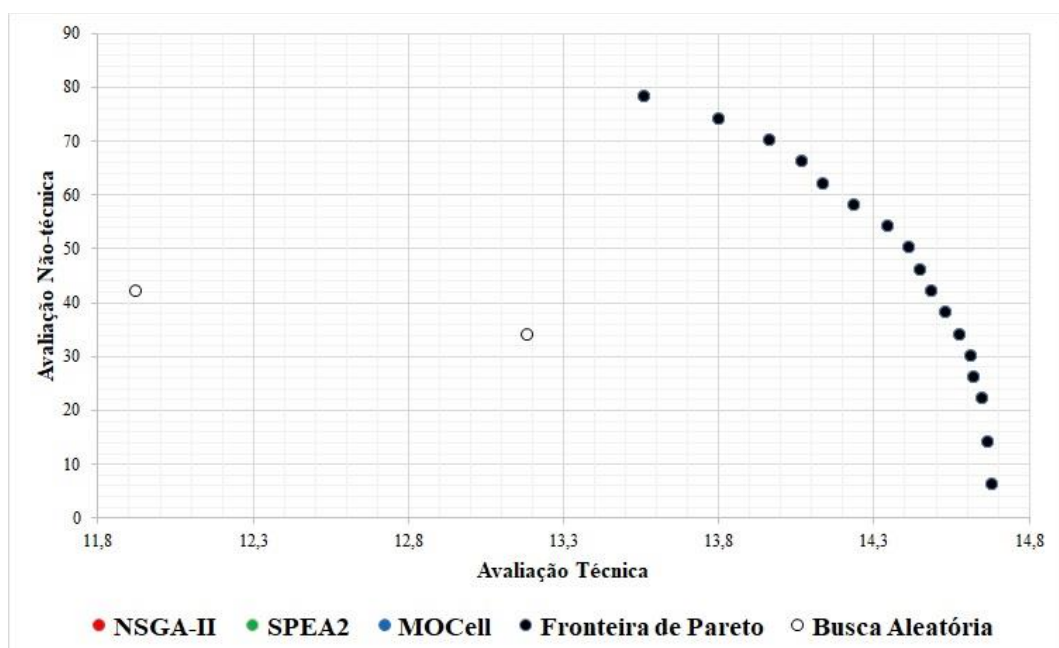


Figura 44 - Resultados para o Experimento XVIII com MBTI

Adicionalmente, a Tabela 7 apresenta o valor médio e o desvio padrão da quantidade de soluções encontradas por cada algoritmo nas 100 execuções, pertencentes à fronteira de Pareto (9 soluções) no Experimento XVII ou à fronteira de referência (17 soluções) no Experimento XVIII. A busca aleatória apresentou um resultado claramente insatisfatório, ao não encontrar uma única solução pertencente ao conjunto de referência em ambos experimentos, ratificando mais uma vez a pobre performance já visualizada na Tabela 6 em relação aos indicadores HVR e espalhamento.

Tabela 7- Média e desvio padrão de quantidade de soluções presentes nas fronteiras (MBTI)

	EXPERIMENTO XVII		EXPERIMENTO XVIII	
NSGA-II	8.90E+00	4.1E-01	1.38E+01	2.1E+00
SPEA2	8.78E+00	6.3E-01	1.39E+01	2.6E+00
MOCeII	8.83E+00	5.5E-01	1.40E+01	2.9E+00
Random	0.00E+00	0.0E+00	0.00E+00	0.00E+00

Finalmente, os resultados para o teste estatístico de Wilcoxon-Mann-Whitney são ilustrados nas Tabelas 8 e 9, considerando o HVR e o espalhamento, respectivamente. Elas utilizam a seguinte simbologia: (i) o símbolo ▲ sinaliza que o algoritmo da linha obteve uma performance melhor que o algoritmo da coluna; (ii) o símbolo ▼ sinaliza que o algoritmo da coluna obteve uma performance melhor que o algoritmo da linha; e (iii) o símbolo – confirma a hipótese nula, em que os algoritmos, para o dado indicador de qualidade avaliado, não obtiveram diferenças significativas nas suas distribuições de valores para este indicador. É importante destacar que os resultados de ambos os experimentos XVII e XVIII são apresentados nas tabelas, de maneira sequencial para cada par de algoritmos comparados. Em outras palavras, em cada célula das Tabelas 8 e 9, o primeiro e o segundo símbolos representam os resultados para o Experimento XVII e Experimento XVIII, respectivamente.

Tabela 8 - Resultados do teste de Wilcoxon para o hipervolume normalizado (MBTI)

	SPEA2	MOCeII	Busca aleatória
NSGA-II	--	--	▲▲
SPEA2		--	▲▲
MOCeII			▲▲

Tabela 9 - Resultados do teste de Wilcoxon para o indicador de espalhamento (MBTI)

	SPEA2	MOCcell	Busca aleatória
NSGA-II	▲ –	▽ ▲	▲ ▲
SPEA2		▽ ▲	▲ ▲
MOCcell			▲ ▲

Em ambas as tabelas (8 e 9), pode ser visto que todos os algoritmos obtiveram uma performance melhor que a busca aleatória sob qualquer aspecto avaliado, excetuando o tempo de execução, o que mostra, incontestavelmente, que não deve ser considerado uma boa escolha, como já demonstrado nos indicadores avaliados. Quanto a comparação entre as metaheurísticas, em relação ao HVR, não foram encontradas diferenças significativas nas suas distribuições de valores para este indicador, o que pode ser constatado na Tabela 6, na qual percebe-se que as metaheurísticas obtiveram valores de HVR muito próximos. Diferentemente do teste Wilcoxon para o espalhamento, que, para o Experimento XVII, o NSGA-II obteve melhores resultados que o SPEA2, mas o MOCcell obteve melhores resultados que os outros dois algoritmos. Porém, no Experimento XVIII, o NSGA-II e SPEA2 tiveram desempenhos equivalentes e o MOCcell obteve piores resultados em comparação com os outros dois algoritmos.

4.2.2.2. FFM

A seção anterior analisa a abordagem proposta através da execução das metaheurísticas utilizando as modelagens concebidas e apresentadas no Capítulo 3 para a tipologia MBTI, juntamente com os dados reais coletados através de um estudo de caso apresentado na Seção 4.1. Entretanto, lembrando que esta abordagem suporta duas tipologias de personalidade, que são modeladas de maneiras distintas, o resultado obtido com o MBTI foi apresentado na subseção anterior e esta subseção analisa o resultado das metaheurísticas aplicadas a dados em relação ao modelo FFM. Portanto, os conceitos, indicadores, cenários e procedimentos dos experimentos são os mesmos aos utilizados nos experimentos com MBTI.

Nesta direção, foram executados dois experimentos utilizando a classificação de personalidade FFM para as entradas da etapa Avaliação do Perfil de Personalidade da abordagem apresentada neste trabalho. Destes, o *Experimento XIX* se refere a execução das três metaheurísticas no Cenário IV, enquanto o *Experimento XX* é relativo à execução destas no Cenário V. Para isto, os algoritmos evolucionários foram parametrizados conforme descrito na Tabela 10. Novamente salienta-se que os valores dos parâmetros foram obtidos através de

sucessivas calibrações. Contrastando estes parâmetros com os utilizados nos experimentos com MBTI (Tabela 5), pode ser percebido um acréscimo no tamanho das populações e, consequentemente, no tamanho dos arquivos, além do acréscimo no número de gerações e, portanto, no número de soluções avaliadas. Estes acréscimos foram necessários devido ao maior número de soluções não-dominadas nas fronteiras de Pareto e de referência dos experimentos com FFM, pois a respectiva modelagem matemática gera resultados menos concentrados no espaço de busca.

Tabela 10 - Parametrização dos experimentos com metaheurísticas (FFM)

	Experimento XIX	Experimento XX
População	25	200
Arquivo	25	200
Feedback	4	4
Gerações	2000	2500
Soluções avaliadas	50000	500000

Devido a especificidade já mencionada do MOCcell, ele não foi capaz de executar o Experimento XX com uma população igual a 200 cromossomos, ocasionando um erro de posição inválida de *array* ao buscar as soluções na vizinhança. Por esta razão, a população do MOCcell para o este experimento foi diminuída para 196 indivíduos, o que não prejudica a saída do algoritmo, nem a comparação com as demais metaheurísticas, pois todos algoritmos exploram a mesma quantidade de soluções no espaço de busca.

Após a execução dos experimentos, os resultados obtidos foram compilados na Tabela 11. Pode ser observado que todos os três algoritmos obtiveram uma performance significativamente melhor que o algoritmo aleatório, em todos os aspectos, exceto no tempo de execução. Mas, tomando como exemplo a execução do NSGA-II no Experimento XIX, que durou apenas 8 segundos a mais que a busca aleatória e obteve o melhor desempenho em todos os outros indicadores, esta pequena diferença de tempo não justificaria o uso de um algoritmo com performance tão ruim. Novamente, lembrando do Experimento XVI, que executou o algoritmo exaustivo no mesmo cenário do Experimento XIX e encontrou a Fronteira de Pareto com todos os 19 valores ótimos em 52 minutos, uma metaheurística capaz de se aproximar destes valores em cerca de 15 segundos é um excelente resultado.

No Experimento XIX, o MOCcell obteve o melhor valor no indicador HVR, seguido pelos algoritmos SPEA2 e NSGA-II, respectivamente. Este mesmo padrão de desempenho em

relação ao HVR foi repetido no Experimento XX, em que o MOCell foi melhor que SPEA2, que, por sua vez, foi melhor que o NSGA-II. Em relação ao tempo de execução, em ambos os experimentos, o mais rápido algoritmo foi o NSGA-II, seguido pelo MOCell e, novamente, o SPEA2 obteve os piores resultados, com um desempenho muito mais lento em contraste com os demais. Como já mencionado, este comportamento é atribuído às complexidades computacionais dos referidos algoritmos. Ainda na Tabela 11, pode ser observado os desempenhos dos algoritmos em relação ao espalhamento, no qual, no Experimento XIX, o NSGA-II teve uma melhor avaliação, seguido do SPEA2 e MOCell. Diferentemente, o MOCell obteve melhor índice no espalhamento no Experimento XX, seguido pelo NSGA-II e SPEA2, respectivamente.

Tabela 11 - Hipervolume normalizado, espalhamento e tempo de execução (FFM)

	NSGA-II	SPEA2	MOCell	Busca aleatória
	HIPERVOLUME NORMALIZADO. Média e desvio padrão			
Experimento XIX	9.99e-01 _{6,3e-04}	1.00e+00 _{7,2e-04}	1.00e+00 _{5,8e-04}	2.65e-01 _{8,2e-02}
Experimento XX	9.88e-01 _{3,9e-03}	9.89e-01 _{4,0e-03}	9.92e-01 _{3,5e-03}	1.04e-01 _{4,2e-02}
	ESPALHAMENTO. Média e desvio padrão			
Experimento XIX	5.07e-01 _{3,5e-02}	5.16e-01 _{3,3e-02}	5.72e-01 _{3,1e-02}	6.88e-01 _{1,6e-01}
Experimento XX	6.62e-01 _{4,9e-02}	6.73e-01 _{5,2e-02}	6.62e-01 _{2,9e-02}	8.44e-01 _{8,9e-02}
	TEMPO DE EXECUÇÃO			
Experimento XIX	0'15"	0'25"	0'24"	0'07"
Experimento XX	5'18"	25'54"	7'08"	1'50"

O gráfico com as saídas do Experimento XIX é ilustrado na Figura 45, por sua vez, o Experimento XX tem a sua saída ilustrada na Figura 46. Estas saídas representam as fronteiras com melhores valores de HVR dentre as 100 execuções. Por um lado, na Figura 45, é difícil distinguir visualmente os resultados das três metaheurísticas em relação ao Pareto ótimo, pois todas as metaheurísticas do Experimento XIX encontraram as 19 soluções do Pareto ótimo. Por outro lado, na Figura 46, com certo grau de dificuldade, podem ser notados valores levemente díspares entre as metaheurísticas avaliadas, pois as metaheurísticas não foram capazes de encontrar todas as soluções não-dominadas da fronteira de referência, que possui 119 soluções não-dominadas. No Experimento XX, o NSGA-II retornou uma fronteira com 108 soluções

não-dominadas, destas 95 pertencem à fronteira de referência. Além disso, MOCell encontrou 108 soluções não-dominadas, destas 106 pertencem à fronteira de referência. Já o SPEA2 encontrou 106 soluções não-dominadas, destas 102 pertencem à fronteira de referência. Diferentemente, as fronteiras resultantes do algoritmo aleatório, em ambos experimentos, estão distantes das fronteiras de Pareto e de referência, encontrando apenas 6 e 2 soluções não-dominadas (Experimentos XIX e XX, respectivamente), e, destas, nenhuma pertence às respectivas fronteiras.

Além disso, os algoritmos ainda podem ser avaliados em relação ao número de soluções descobertas em todas as 100 execuções de cada um deles. Segundo Amaral e Elias (2018), o NSGA-II tende a encontrar menos soluções em cada rodada, mas diferentes soluções entre todas as rodadas em relação à fronteira de Pareto ou de referência. Já o SPEA2 e o MOCell revelam uma tendência de encontrar mais soluções não dominadas em cada execução, porém mais recorrentes entre todas as execuções (AMARAL; ELIAS, 2018). Esta constatação pode ser observada no comportamento do NSGA-II no Experimento XX, que encontrou 95 soluções pertencentes à fronteira de referência em uma execução (a de maior HVR), porém nas 100 rodadas, descobriu 116 soluções da fronteira referência que possui 119 soluções. Em relação às 100 rodadas, o NSGA-II obteve desempenho similar aos demais algoritmos, que descobriram 118 e 116 soluções (MOCell e SPEA2, respectivamente). Quanto ao Experimento XIX, todas metaheurísticas, considerando as 100 execuções, descobriram todas as 19 soluções da fronteira de Pareto. Já o algoritmo aleatório, em todos os experimentos, não descobriu boas soluções, mesmo quando consideradas as 100 execuções.

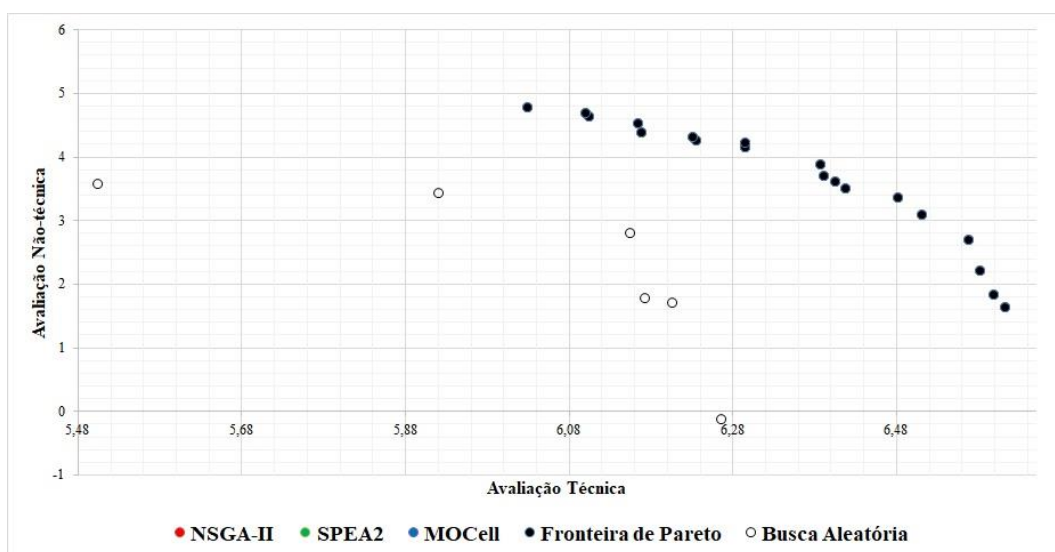


Figura 45 - Resultados para o Experimento XIX com FFM

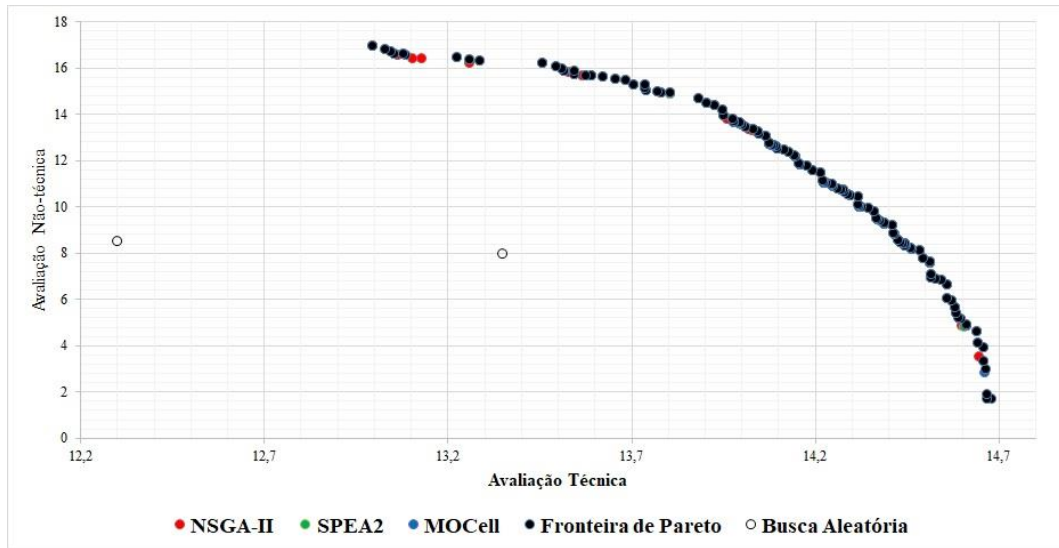


Figura 46 - Resultados para o Experimento XX com FFM

Adicionalmente, a Tabela 12 apresenta o valor médio e o desvio padrão da quantidade de soluções encontradas por cada algoritmo nas 100 execuções, que são pertencentes ao conjunto ótimo almejado (ótimo ou referência, Experimento XIX ou Experimento XX, respectivamente). A busca aleatória apresentou um resultado claramente insatisfatório, ao não encontrar uma única solução pertencente ao conjunto de referência em ambos os experimentos.

Tabela 12- Média e desvio padrão de quantidade de soluções presentes no conjunto ótimo ou de referência (FFM)

	EXPERIMENTO XIX		EXPERIMENTO XX	
NSGA-II	1.82E+01	4.4E-01	9.03E+01	6.0E+00
SPEA2	1.89E+01	4.9E-01	8.83E+01	7.5E+00
MOCeII	1.89E+01	1.9E+01	9.42E+01	6.1E+00
Random	0.00E+00	0.0E+00	0.00E+00	0.00E+00

Finalmente, os resultados para o teste estatístico de Wilcoxon-Mann-Whitney são ilustrados nas Tabelas 13 e 14, considerando o HVR e o espalhamento, respectivamente. Estas tabelas utilizam simbologia própria, já apresentada na subseção anterior. Porém, para melhor entendimento, em suma, a simbologia é novamente apresentada: (i) o símbolo ▲ sinaliza que o algoritmo da linha obteve uma performance melhor que o algoritmo da coluna; (ii) o símbolo ∇ sinaliza que o algoritmo da coluna obteve uma performance melhor que o algoritmo da linha; e (iii) o símbolo – sinaliza que não há diferenças significativas nas distribuições de valores para este indicador. É importante lembrar que os resultados dos experimentos são apresentados nas

tabelas de maneira sequencial para cada par de algoritmos comparado. Em outras palavras, em cada célula das Tabelas 13 e 14, o primeiro símbolo e o segundo representam os resultados para o Experimento XIX e Experimento XX, respectivamente.

Tabela 13 - Resultados do teste de Wilcoxon para o hipervolume normalizado (FFM)

	SPEA2	MOCcell	Busca aleatória
NSGA-II	∇–	∇∇	▲ ▲
SPEA2		–∇	▲ ▲
MOCcell			▲ ▲

Tabela 14 - Resultados do teste de Wilcoxon para o indicador de espalhamento (FFM)

	SPEA2	MOCcell	Busca aleatória
NSGA-II	▲ ▲	▲–	▲ ▲
SPEA2		▲–	▲ ▲
MOCcell			▲ ▲

Em ambas as tabelas (13 e 14), novamente, constata-se que todos os algoritmos obtiveram uma performance melhor que a busca aleatória em relação a todos aspectos avaliados, o que mostra, sem dúvidas, que não deve ser considerado uma boa escolha, como já demonstrado nos outros indicadores avaliados. Quanto a comparação entre os demais algoritmos, no Experimento XIX, por um lado, em relação ao HVR, o SPEA2 obteve melhores índices em comparação com o NSGA-II, o MOCcell foi melhor que o NSGA-II, já entre o SPEA2 e MOCcell não houve significante diferença. Por outro lado, em relação ao espalhamento, o NSGA-II teve melhor desempenho que o SPEA2 e MOCcell, além disso, o SPEA2 teve melhor desempenho que o MOCcell. Em relação ao Experimento XX, não foram encontradas diferenças significativas nos desempenhos, em relação ao HVR, do SPEA2 em comparação com o NSGA-II, nem nos desempenhos, em relação ao espalhamento, do MOCcell em comparação com o NSGA-II e MOCcell. Além disso, o espalhamento do NSGA-II foi melhor que o SPEA2 e o MOCcell foi melhor que as outras metaheurísticas em relação ao HVR.

4.3. Análise empírica da competitividade dos resultados

A seção anterior analisa a abordagem proposta sob o ponto de vista da aplicabilidade em termos da modelagem matemática, coleta de dados aplicando os formulários sugeridos, e performance das metaheurísticas em termos dos clássicos indicadores de qualidade relativos a

problemas multiobjetivos. Dando continuidade, esta seção se concentra na análise da qualidade das soluções obtidas, para isso, os resultados encontrados pelos algoritmos foram avaliados em relação à sua competitividade para com resultados de seres humanos.

A comparação de resultados da SBSE com resultados obtidos através do julgamento humano é uma forma viável e interessante de quantificar e julgar a competitividade dos resultados de uma abordagem automatizada (TORRES, 2010). Este conceito foi primeiramente sugerido por Harman (2007), que discute os problemas abertos e desafios na área de otimização para ES, citando a potencialidade da avaliação envolvendo a comparação com soluções propostas por humanos, caracterizando a competitividade humana das abordagens automáticas. Posteriormente, este conceito é utilizado no trabalho de Souza et al. (2010), que avalia bem conhecidos problemas da SBSE (próximo release multiobjetivo, formação de grupo de trabalho e seleção multiobjetiva de casos de testes), considerando duas instâncias, sinteticamente geradas, de tamanhos crescentes para cada um destes problemas. Estas instâncias foram resolvidas por metaheurísticas e humanos, e, em seguida, os resultados gerados foram comparados, chegando à conclusão que apresentaram fortes indícios de que SBSE é capaz de produzir soluções competitivas, ditas em inglês como *human competitive*. Mais recentemente, à luz de um mapeamento sistemático realizado com o interesse de prover uma visão geral do estado da arte do tópico *human competitive* da SBSE, Souza et al. (2018) constatarem que o interesse da comunidade científica por este tema tem se apresentado crescente, especialmente após 2010. Nesta direção, esta forma de validação foi utilizada por Maia et al. (2011), Torres (2010) e Colares et al. (2009).

Dada a importância da análise de competitividade humana, os resultados obtidos pela abordagem apresentada neste trabalho são avaliados, quanto à eficácia e eficiência, do ponto de vista da gestão de projetos no contexto da seleção e alocação de pessoas à papéis em projetos de software. O objetivo é analisar possíveis ganhos que a abordagem pode trazer a gestão de projetos, especificamente em relação à gestão de recursos humanos. Estes ganhos serão analisados em relação aos seguintes aspectos: (i) *competitividade*: a abordagem proposta produz resultados melhores ou equiparáveis aos de seres humanos especialistas; e (ii) *produtividade*: a abordagem proposta produz resultados em menor tempo que seres humanos especialistas.

Para isto, inicialmente, foi solicitado ao gerente de projetos que realizasse duas seleções e alocações de pessoas a papéis de dois projetos criados de acordo com os cenários IV e V descritos na seção 4.2. Estas atividades de tomada de decisão foram feitas com base em seu próprio julgamento e apoiado pelas informações conceituais levantadas pela pesquisa e apresentadas ao mesmo. Além disso, as soluções apresentadas deveriam ser válidas. De posse

dos resultados, deparou-se com a seguinte limitação: comparar uma fronteira com uma única solução com outra fronteira com diversas soluções, o que eliminaria a possibilidade de análise do hipervolume normalizado (HVR). Para lidar com esta limitação, foi solicitado que o gerente substituto também realizasse a mesma tarefa de tomada de decisão. Sendo assim, para cada cenário avaliado, se tornou possível a comparação entre as soluções obtidas por humanos e as soluções não-dominadas encontradas pelas metaheurísticas. Para esta comparação, o HVR dos algoritmos é a média das 100 execuções e o HVR dos humanos é calculado em relação as duas soluções apresentadas. Vale ressaltar que o indicador espalhamento não agrega valor à esta validação, pois mediria a distribuição entre somente dois pontos.

Por questões de clareza, as definições dos cenários utilizados serão reapresentadas:

- *Cenário IV*: corresponde a um projeto completo de um sistema de médio porte, que possui 12 pessoas candidatas $PE = \{pe_1, \dots, pe_{12}\}$, que devem ser alocadas nos cinco papéis $PA = \{pa_1, \dots, pa_5\}$, sendo duas vagas para cada papel $VG = \{2, 2, 2, 2, 2\}$. Este experimento lida com cerca de 1.25233×10^9 soluções possíveis.
- *Cenário V*: relativo ao conjunto de dados total da organização, que é constituído de 30 candidatos e 5 papéis ($VG = \{3, 3, 7, 4, 3\}$), o que resulta em um gigantesco espaço de busca, que tem cerca de 3.7333731×10^{21} soluções candidatas.

Primeiramente, em relação ao Cenário IV, utilizando os dados de personalidade MBTI, a Figura 47 apresenta um gráfico confrontando os resultados obtidos pelas metaheurísticas e pelo algoritmo aleatório no Experimento XVII com os resultados obtidos pelos tomadores de decisão humanos. Nesta figura podem ser observadas as duas soluções obtidas pelos dois humanos especialistas (uma pelo gerente de projetos e outra por seu substituto), onde ambas estão simbolizadas pelos quadrados da cor laranja. Nota-se que um dos humanos obteve resultado até mesmo pior do que as soluções do algoritmo aleatório. Entretanto, o outro humano obteve uma solução próxima à fronteira de Pareto. Racionalizando este comportamento, a distância em qualidade das soluções humanas pode ser atribuída a diferença de experiência profissional possuída por ambos, o que não é o foco desta validação. Vale salientar que não foi avaliada a não dominância entre as soluções obtidas por humanas, por não agregar valor à análise pretendida nesta seção.

As soluções obtidas por humanos em relação à fronteira de Pareto obtiveram o valor 2.89E-01 no indicador HVR. Confrontando este valor com os valores de HVR dos algoritmos no Experimento XVII, apresentados na Tabela 6, pode ser notado que os humanos obtiveram desempenho muito pior que as metaheurísticas, cujos valores de HVR são muito próximos ou iguais ao ideal (valor 1.0), porém ligeiramente melhor que o algoritmo aleatório (2.63E-01).

Em relação ao tempo de execução, facultou-se aos humanos a livre execução da atividade, podendo inclusive ser feito em horários não consecutivos, desde que os tempos dedicados a atividade fossem devidamente registrados e anotados, e que os profissionais se dedicassem exclusivamente à atividade nos momentos escolhidos para sua execução. Sendo assim, somados os tempos dedicados à tomada de decisão, o gerente de projetos gastou 45 minutos e o gerente substituto gastou 58 minutos para executar a seleção e alocação de pessoas no cenário IV. Confrontando este resultado com o obtido pelos algoritmos, percebe-se que os humanos gastaram muito mais tempo que o algoritmo mais demorado (SPEA2), que levou apenas 13 segundos nas 100 execuções.

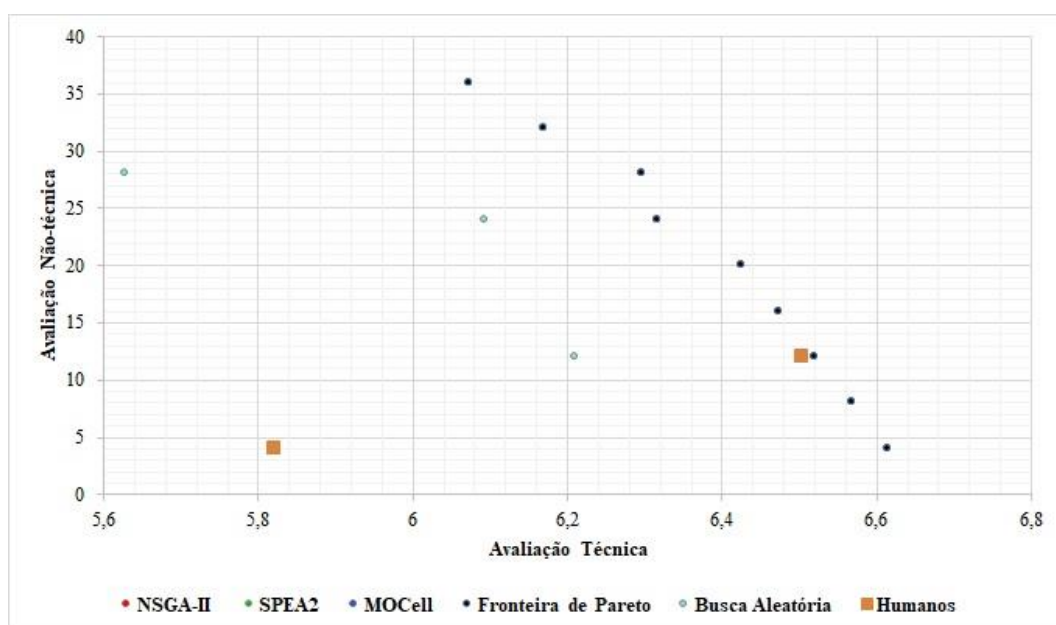


Figura 47 - Resultado do Cenário IV MBTI: algoritmos x humanos

Em seguida, ainda em relação ao Cenário IV, agora utilizando os dados de personalidade FFM, a Figura 48 apresenta um gráfico comparando os resultados obtidos pelas metaheurísticas e pelo algoritmo aleatório no Experimento XIX com os resultados obtidos pelos humanos. Nesta figura podem ser observadas as duas soluções obtidas pelos dois humanos especialistas, onde ambas estão simbolizadas pelos quadrados da cor laranja. Novamente, observa-se que um dos humanos obteve resultado pior que as soluções do algoritmo aleatório. Entretanto, o outro obteve uma solução próxima à fronteira de Pareto. As soluções obtidas por humanos em relação à fronteira de Pareto obtiveram o valor $3.68E-01$ no indicador HVR. Confrontando este valor com os valores de HVR dos algoritmos no Experimento XIX, apresentados na Tabela 11, pode ser notado que os humanos obtiveram desempenho muito pior que as metaheurísticas (acima de $9.88E-01$), porém melhor que o apresentado pelo algoritmo aleatório ($1.04E-01$). Além

disso, somados os tempos dedicados à tomada de decisão, o gerente de projetos gastou 1 hora e 10 minutos e o gerente substituto gastou 1 hora e 34 minutos para executar a seleção e alocação de pessoas neste cenário. Comparando este resultado com o obtido pelos algoritmos percebe-se que os humanos, novamente, gastaram muito mais tempo que o algoritmo mais demorado (SPEA2), que levou apenas 25 segundos em suas 100 execuções.

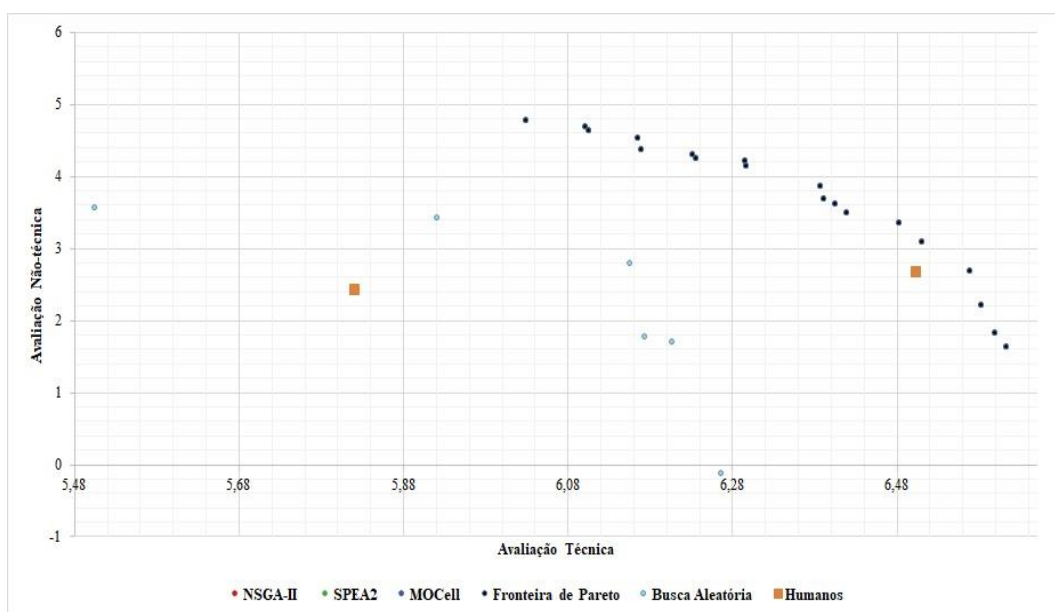


Figura 48 - Resultado do Cenário IV FFM: algoritmos x humanos

Adicionalmente, em relação ao Cenário V, com classificação MBTI, a Figura 49 apresenta a comparação entre os resultados obtidos pelos algoritmos (metaheurísticas e aleatória) no Experimento XVIII com os resultados obtidos pelos humanos. Nesta figura podem ser observadas as duas soluções obtidas pelos dois humanos especialistas (quadrados da cor laranja). Diferentemente das análises anteriores, ambas soluções obtidas por humanos estão distantes da fronteira de referência, obtendo o valor $8.32E-02$ no indicador HVR. Confrontando este valor com os valores de HVR dos algoritmos no Experimento XVIII, apresentados na Tabela 6, pode ser notado que os humanos obtiveram desempenho muito pior que todas as metaheurísticas ($9.91E-01$), porém levemente melhor do que o algoritmo de busca aleatória ($7.81E-02$). Além disso, somados os tempos dedicados à tomada de decisão, o gerente de projetos gastou 2 horas e 20 minutos e o gerente substituto gastou 2 horas e 45 minutos para executar a seleção e alocação de pessoas neste cenário. Comparando este resultado com o obtido pelos algoritmos, percebe-se que os humanos, novamente, gastaram muito mais tempo que o algoritmo mais demorado (SPEA2), que levou apenas 3 minutos e 9 segundos em suas 100 execuções. Esta piora no desempenho dos humanos pode ser atribuída ao aumento da

complexidade do problema, o que torna muito mais difícil e cansativa a análise dos perfis técnicos e de personalidades requeridos pelos papéis funcionais e possuídos pelas pessoas.

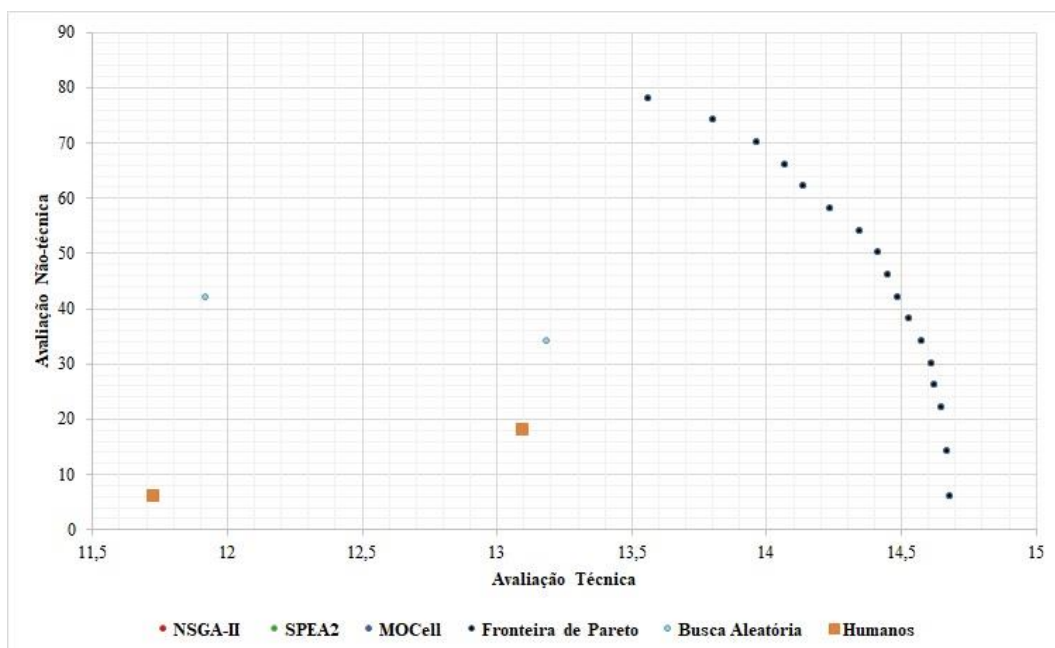


Figura 49 - Resultado do Cenário V MBTI: algoritmos x humanos

Finalmente, em relação ao Cenário V, com classificação FFM, a Figura 50 apresenta a comparação entre os resultados dos algoritmos (metaheurísticas e algoritmo de busca aleatória) no Experimento XX com os resultados obtidos pelos humanos. Nesta figura podem ser observadas as duas soluções obtidas pelos dois humanos especialistas (quadrados da cor laranja). Ambas as soluções obtidas pelos humanos estão distantes da fronteira de referência, obtendo o valor $6,38E-02$ no indicador HVR. Comparando este valor com os valores de HVR dos algoritmos no Experimento XX, apresentados na Tabela 11, pode ser observado que os humanos obtiveram desempenho muito pior que todos as metaheurísticas (acima de $9,88E-01$) e também do algoritmo de busca aleatória ($1,04E-01$). Em relação ao tempo de execução, somados os tempos dedicados à tomada de decisão, o gerente de projetos gastou 2 horas e 52 minutos e o gerente substituto gastou 3 horas e 05 minutos para executar a seleção e alocação de pessoas neste cenário. Comparando este resultado com o obtido pelos algoritmos percebe-se que os humanos, novamente, gastaram muito mais tempo que o algoritmo mais lento, que levou apenas 25 minutos e 54 segundos em suas 100 execuções. De forma similar ao que ocorreu nos resultados deste cenário V com o modelo MBTI, esta piora no desempenho dos humanos certamente pode ser atribuída ao aumento da complexidade do problema.

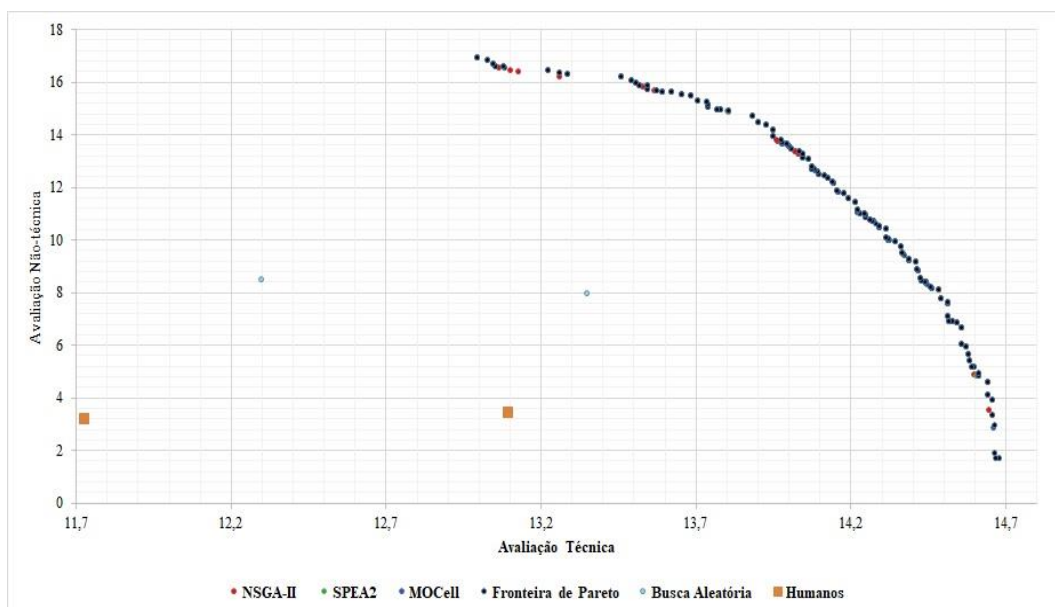


Figura 50 - Resultado do Cenário V FFM: algoritmos x humanos

Em suma, após as análises dos resultados dos cenários IV e V, para ambas as modelagens de personalidade, os resultados obtidos pela abordagem apresentada neste trabalho podem ser avaliados quanto à eficácia e eficiência, do ponto de vista da gestão de projetos no contexto da seleção e alocação de pessoas a papéis em projetos de software. Neste estudo de caso, a abordagem proposta trouxe ganhos à gestão de recursos humanos dos projetos. Estes ganhos foram analisados em relação aos seguintes aspectos: (i) *competitividade*: a abordagem proposta trouxe resultados melhores aos de seres humanos especialistas; e (ii) *produtividade*: a abordagem proposta produz resultados em menor tempo que seres humanos especialistas. Respeitando as ameaças à validade que serão apresentadas a seguir, pode ser concluído que a abordagem proposta apresenta fortes indícios de possuir competitividade humana (*human competitive*).

4.4. Análise da validade

Após análise dos experimentos em relação à validação da modelagem, ao desempenho dos algoritmos e à competitividade das soluções comparadas às obtidas por humanos, conclui-se que a abordagem proposta apresenta resultados promissores. Entretanto, é necessário avaliar as ameaças à validade dos resultados obtidos, analisando os fatores que podem introduzir vieses nos resultados.

Dentre as principais ameaças à validade destes experimentos, podemos elencar como principal ameaça externa, o fato que o estudo de caso foi desenvolvido numa organização

pública, pois os critérios e o processo de montagem das equipes podem variar numa organização privada, por exemplo. Segundo Wohlin *et al.* (2012), os resultados devem ser válidos em primeiro lugar para a população das quais a amostra é tirada e em segundo plano pode haver algum interesse em generalizar tais resultados de uma população maior. Sendo assim, em trabalhos futuros, uma organização privada pode ser avaliada para incrementar a confiança nos resultados obtidos.

Adicionalmente, outra ameaça externa, que se relaciona com a ameaça anterior, é o número e diversidade dos humanos participantes, que é essencial para generalização das conclusões. Neste sentido, os 30 participantes candidatos ao projeto (incluindo o gerente substituto) e o gerente de projetos representam uma pequena amostra, agravado pelo fato de serem da mesma organização, cidade, tem similares formações educacionais e experiências profissionais. Uma mais diversa amostra de profissionais seria benéfica para uma maior validade das conclusões.

Existem ainda as ameaças internas, que são relacionadas a fontes de equívocos no projeto de pesquisa, que afetariam o processo causa-efeito que está sendo estudado, introduzindo variáveis imprevistas (WOHLIN *et al.*, 2012). Dentre as ameaças internas, a expectativa do experimentador pode ser um possível problema de validade do estudo, pois há uma possibilidade de que as suas ações possam ter contaminado os resultados, já que a projetista da pesquisa faz parte do ambiente do estudo de caso sendo também uma das avaliadoras dos resultados. Além disso, outra ameaça interna está relacionada à subjetividade de alguns dados de entrada devido ao envolvimento de um aspecto complexo que é a psique humana, no entanto, tal viés é comum em projetos reais. Por isso, apesar da contribuição relevante e dos resultados promissores, a abordagem proposta deve ser avaliada de forma mais intensiva, incluindo sua aplicação em outras organizações e projetos de modo a evitar vieses.

Finalmente, em relação às ameaças de conclusão, como uma forma de minimizar os efeitos da natureza estocástica dos algoritmos de busca e garantir uma comparação justa, NSGA-II, SPEA2, MOCell e a busca aleatória foram executados múltiplas vezes em relação a cada um dos experimentos. Isto foi feito com objetivo de superar a aleatoriedade inerente em tais técnicas. Já em relação às ameaças de construção, por não ter havido de fato a seleção e alocação das pessoas nos papéis de acordo com as atribuições sugeridas pela abordagem, não foi possível verificar na prática o desempenho das pessoas selecionadas e das equipes formadas. Sendo assim, em trabalhos futuros, a solução escolhida pelo tomador de decisão deve ser utilizada em uma alocação real.

4.5. Considerações éticas

Uma vez que esta pesquisa estuda equipes de software em relação aos seus integrantes, seres humanos estão envolvidos, ainda que de forma não invasiva. A realidade, o cotidiano e a intimidade das pessoas e o seu ambiente de trabalho são analisados quando se avalia os aspectos técnicos e não técnicos envolvidos no processo de seleção e alocação de pessoas que atuarão nos projetos de software da organização. Desse modo, é necessário assegurar a todos os envolvidos, inclusive a organização participante, que todos os procedimentos são realizados em observância aos princípios éticos.

Nesta direção, no Brasil, a Resolução 466/12 do Conselho Nacional de Saúde – CNS incorpora, sob a ótica do indivíduo e das coletividades, referenciais da bioética, tais como, autonomia, não maleficência, beneficência, justiça e equidade, objetivando prezar pelo respeito à dignidade humana e à proteção devida aos participantes das pesquisas científicas envolvendo seres humanos. A resolução visa assegurar os direitos e deveres que dizem respeito aos participantes da pesquisa, à comunidade científica e ao Estado.

Em relação aos conceitos da bioética considerados pela resolução, a *autonomia* assegura ao indivíduo o respeito a sua vontade de contribuir e permanecer, ou não, na pesquisa, por intermédio de manifestação expressa, livre e esclarecida, manifestado através do consentimento livre e esclarecido dos indivíduos-alvo. Já a *beneficência* é a ponderação entre riscos e benefícios, tanto conhecidos como potenciais, individuais ou coletivos, comprometendo-se com o máximo de benefícios e o mínimo de danos e riscos. Em complemento, a *não maleficência* é garantia de que danos previsíveis serão evitados. Finalmente, a “*justiça e equidade*” se refere à relevância social da pesquisa, o que garante a igual consideração dos interesses envolvidos, não perdendo o sentido de sua destinação sócio humanitária.

Assim, observando o preconizado pela Resolução 466/12 do CNS, foram desenvolvidos e formalizados os seguintes documentos:

- Termo de Autorização para Pesquisa: descreve os objetivos, benefícios, procedimentos e eventuais riscos para a organização participante, além de obter sua autorização para desenvolver a pesquisa. Este termo pode ser visto integralmente no Apêndice B.
- Termo de Consentimento Livre e Esclarecido: informa ao participante sobre os objetivos, benefícios, procedimentos, e eventuais riscos para a organização e para si, além de formalizar a participação de forma voluntária. Este termo pode ser visto integralmente no Apêndice C.

Outra consideração ética a ser debatida é em relação ao risco aos direitos dos trabalhadores submetidos a processos seletivos guiados por decisões automatizadas. Neste sentido é necessário prover meios de proteção contra práticas discriminatórias (REIS; GRAMINHO, 2019). Por isto é importante equacionar a liberdade de escolha do empregador e a preservação do direito fundamental do trabalhador de não ser discriminado ante as oportunidades de emprego (NASCIMENTO, 2011). Este risco é atenuado pelo fato de que a abordagem proposta nesta dissertação não aponta uma única solução de seleção e alocação de pessoas a papéis, mas sim um conjunto destas soluções, as quais servirão de auxílio ao tomador de decisão, que pode escolher uma delas, podendo, inclusive adequá-la. Além disso, recomenda-se aos utilizadores da abordagem proposta que forneça explicações claras acerca dos critérios e procedimentos utilizados no processo de decisão, de forma que a tomada de decisão seja a mais transparente possível, permitindo ao interessado expressar seu ponto de vista.

4.6. Considerações finais

Este capítulo apresentou um estudo de caso realizado em uma instituição pública de ensino brasileira, chamada de Instituto Federal da Paraíba. Nesta organização foram coletados os dados relativos à atividade de seleção e alocação de pessoas a papéis dos projetos de desenvolvimento de software, que é o foco deste trabalho. De posse dos dados coletados, experimentos com as metaheurísticas MOCell, NSGA-II e SPEA2 foram realizados, juntamente com o algoritmo aleatório, que foi utilizado a fim de realizar um “teste de sanidade”. Quando o espaço de busca do cenário do problema possibilitava uma busca exaustiva, esta era executada para prover informações a respeito da fronteira de Pareto.

De posse dos indicadores e soluções obtidas pelos algoritmos utilizados, os resultados foram comparados entre si e com os resultados obtidos pelos humanos especialistas da área de gerência de projetos de software. Este experimento agregou grande valor ao trabalho, demonstrando de forma fundamentada a eficácia da abordagem em atingir os objetivos propostos, no que se refere à qualidade dos resultados e aos ganhos gerenciais obtidos com a sua utilização. Também foram avaliadas às ameaças à validade deste trabalho, algumas das quais abrem possibilidades de trabalhos futuros. Além disso, na condução dos estudos de caso, foram asseguradas as questões relativas à ética envolvida em pesquisas que envolvem seres humanos.

Em suma, com base no estudo de caso e experimentos, foi possível avaliar a utilidade e aplicabilidade, bem como a modelagem matemática da abordagem proposta. Em relação à utilidade e aplicabilidade, o gerente de projetos do estudo de caso reconheceu abertamente que a abordagem proposta atende tais propriedades, concluindo que os seus parâmetros de entrada não são difíceis de coletar em processos clássicos de gerenciamento e planejamento de projetos, em especial nas atividades relativas à gestão de pessoas. Vale ressaltar que tais propriedades foram alcançadas com estratégias fáceis, simples e de baixo esforço, principalmente quando se vislumbra bases históricas de projetos com seus papéis, equipes e requisitos técnicos previamente caracterizados.

5. TRABALHOS RELACIONADOS

O presente capítulo tem como objetivo apresentar alguns trabalhos relacionados ao tema deste trabalho, compará-los e evidenciar seus pontos fracos e fortes. Para isto, foram considerados os trabalhos que consideram abordagens de otimização para o gerenciamento de RH em projetos de software, mais especificamente na alocação de pessoas a tarefas ou montagem de equipes de desenvolvimento. A Tabela 15 lista os trabalhos selecionados, que serão descritos nas seções a seguir.

Tabela 15 - Trabalhos relacionados

Título	Referência
<i>A Hybrid Technique for Software Project Scheduling and Human Resource Allocation</i>	(AVINASH; RAMANI, 2014)
<i>A Multi-objective Genetic Algorithm for Software Development Team Staffing Based on Personality Types</i>	(STYLIANOU; ANDREOU, 2012)
<i>Project Resource Allocation Optimization using Search Based Software Engineering – A Framework</i>	(BIBI; AHSAN; ANWAR, 2014)
<i>Multi-Objective Optimization for Software Development Projects</i>	(GONSALVES; ITOH, 2010)
<i>A Multi-Objective Genetic Algorithm for Software Personnel Staffing for HCIM Solutions</i>	(JIMÉNEZ-DOMINGO; COLOMO-PALACIOS; GÓMEZ-BERBÍS, 2014)
<i>A Knowledge-Based Evolutionary Assistant to Software Development Project Scheduling</i>	(YANNIBELLI; AMANDI, 2011)

5.1. Escalonamento de projetos de software e alocação de recursos humanos

Avinash e Ramani (2014) propõem uma abordagem de planejamento de projetos que envolve a geração de cronogramas e alocação de pessoas. Neste trabalho, o objetivo é gerar o escalonamento das tarefas do projeto e a alocação de pessoas para estas tarefas minimizando o custo. Para atingir tal objetivo, o modelo proposto utiliza dois componentes. O primeiro componente gera o grafo de precedência entre as tarefas, no qual cada tarefa só poderá ser iniciada após o término das antecessoras. Além disso, a proposta também considera o esforço estimado para cada tarefa, as habilidades requeridas, o prazo para término de cada tarefa e as prioridades das tarefas. O segundo componente utiliza o algoritmo de otimização colônia de formigas (*Ant Colony Optimization* - ACO) para gerar a alocação de profissionais para cada

tarefa. Para isto, considera o custo salarial, a quantidade de horas disponíveis da força de trabalho do funcionário, período de disponibilidade e adequabilidade de cada candidato para cada tarefa.

Porém, diferentemente da abordagem aqui proposta, Avinash e Ramani (2014) consideram que a organização possui um banco de dados dos profissionais de seu quadro e das tarefas com todos os atributos necessários para a execução da abordagem. Os atributos dos profissionais que são considerados são referentes ao salário (valor da hora normal e hora extra), jornada de trabalho (máximo de horas de trabalho por mês), janela de tempo que está disponível e a lista de habilidades possuídas. Para cada tarefa é considerada a ordem de precedência entre elas, além dos atributos esforço estimado, conjunto de habilidades requeridas, número máximo de pessoas e o prazo.

Nota-se que, na abordagem de Avinash e Ramani (2014), o fato de que é necessário que a organização gere, mantenha e atualize todas as informações necessárias para a abordagem, sem nenhuma especificação de como pode ser realizado, é um ponto que pode ser melhorado pois as definições dos mecanismos de coleta poderiam estar incorporadas a uma abordagem que proponha auxiliar no gerenciamento do RH, de forma a evitar que pessoas avaliadas de maneiras distintas possam ser comparadas em relação a critérios diferentes. Em contraponto, a abordagem proposta neste trabalho define as formas de coleta e cálculo de cada atributo necessário para a sua utilização, sugerindo formas sistemáticas de obter estas informações através de questionários e ferramentas automatizadas.

Em Avinash e Ramani (2014), de forma similar a abordagem aqui descrita, a lista de tarefas considerada é dada como entrada pela organização em cada projeto, entretanto, as granularidades consideradas são diferentes, pois Avinash e Ramani (2014) consideram as tarefas e a abordagem deste trabalho considera papéis que agrupam várias tarefas típicas. Assim, por um lado, pode ser notado que a utilização de uma lista de microtarefas do projeto é mais trabalhosa e pouco reaproveitável entre projetos distintos, porém úteis para projetos que possuam uma lista bem definida de pequenas tarefas e necessitem de maior controle em sua alocação. Por outro lado, utilizar uma lista de papéis é mais prático, reutilizável e útil para projetos com atividades não previamente definidas.

Embora Avinash e Ramani (2014) não tenham classificado as habilidades em relação aos aspectos técnicos ou não-técnicos, a Tabela 16 lista as habilidades consideradas em seu trabalho, separadas em colunas que definem a classificação atribuída pela autora desta dissertação para facilitar uma análise comparativa. Como pode ser observado, as habilidades consideradas como técnicas não se referem diretamente a conhecimentos em tecnologias e

domínios, mas sim conceitos amplos relacionados a utilização de tecnologias. Embora possam ser obtidos através de estudo e práticas profissionais e acadêmicas, tais habilidades técnicas são bem mais difíceis de mensurar e subjetivas. Observe que similar dificuldade e subjetividade também se aplicam às habilidades não-técnicas. Após análise das habilidades consideradas em Avinash e Ramani (2014), listadas na Tabela 16, percebe-se a complexidade que a organização utilizadora da abordagem enfrentará para medir quão bom ou ruim o funcionário é em cada uma delas.

Tabela 16 - Habilidades consideradas por Avinash e Ramani (2014)

Não-técnicos	Técnicos
Apto a organizar tarefas na ordem correta	Apto a entender o processo e ferramentas
Apto a encontrar as pessoas adequadas e disponíveis para o projeto	Deve auxiliar no desenvolvimento de procedimentos
Apto a resolver as necessidades e obstáculos	Apto a prover suporte técnico para tecnologias
Apto a testar as pessoas do time	Apto a testar o produto em todos os aspectos
Apto a lidar com mudanças no projeto	Aptos a implementar tecnologias e testá-las
Apto a motivar e inspirar outras pessoas do time	Experiência
Deve ser bom em gerenciar o tempo	
Apto a gerenciar conflitos	
Aptos a treinar outros membros do time	

Em suma, do trabalho de Avinash e Ramani (2014), foram identificadas as seguintes desvantagens: (i) as habilidades técnicas são analisadas de forma subjetiva e genérica ao invés de serem analisadas de forma objetiva e pontual por tecnologia, processo ou domínio de aplicação - a natureza subjetiva e genérica pode erroneamente levar a problemas em projetos reais, uma vez que um profissional pode ser muito experiente e habilidoso em determinada técnica e não ter conhecimento em outra; (ii) as habilidades não-técnicas consideradas são difíceis de medir e controlar pois são aspectos micro representantes das personalidades. Estas desvantagens podem ser sanadas com a proposta desta dissertação, pois avaliará as habilidades técnicas necessárias a cada papel do projeto em cada tecnologia e considerará as habilidades não-técnicas de acordo com as personalidades que são facilmente medidas por diversos instrumentos da psicologia.

5.2. Abordagem multiobjetiva de alocação de pessoas em projetos de software

O artigo de Stylianou e Andreou (2012) propõem um algoritmo genético multiobjetivo para selecionar desenvolvedores para projetos de desenvolvimento de software baseados em habilidades técnicas e traços de personalidade. Portanto, o objetivo da abordagem proposta por Stylianou e Andreou (2012) é apoiar os gerentes de projeto de software das pequenas e médias empresas na seleção e alocação dos desenvolvedores mais adequados às tarefas, tentando otimizar as atribuições com base em habilidades técnicas e traços de personalidade dos desenvolvedores usando um algoritmo genético multiobjetivo, que retornará a menor quantidade de desenvolvedores suficiente para o projeto.

Para atingir o objetivo proposto, Stylianou e Andreou (2012) relatam superficialmente que utilizam a taxonomia sistemática das profissões no catálogo *Occupational Information Network* (O*NET) *Resource Center*⁶ para obter as características e requisitos das profissões de software. Com estas informações, os traços de personalidade mais adequados exigidos pelos profissionais para realizar as atividades são definidos e expressos usando o FFM. Sendo assim, o nível desejado de cada fator é definido de modo a determinar se uma profissão requer um nível baixo, médio ou alto daquele domínio particular. Lembrando que, as personalidades do FFM possuem cinco fatores, cada um dos quais assumem valores entre 0 e 100 de acordo com o nível apresentado pela pessoa. Portanto, para converter os valores destes fatores em níveis baixo, médio ou alto é necessária a definição de faixas de valores, porém esta informação não é relatada na proposta de Stylianou e Andreou (2012).

Entretanto, Stylianou e Andreou (2012) não ilustram ou exemplificam quais são os traços de personalidade mais adequados para as atividades de desenvolvimento de software mapeados do O*NET para o FFM, nem relatam como foi realizado este mapeamento. Já que o mapeamento entre o O*NET⁴ e as personalidades requeridas não é direto, pois o O*NET⁴ somente lista habilidades e características gerais das profissões. Por exemplo, para a profissão de desenvolvedor de software são listadas as características: compreensão de leitura, audição ativa, pensamento crítico, solução complexa de problemas, fala, gerenciamento de tempo, tomada de decisões, matemática e escrita. Ou seja, estas características por si só não são suficientes para derivar quais são os domínios de personalidade FFM ideais, nem quais níveis são requeridos.

⁶ U.S. Department of Labour. *Occupational Information Network, Employment and Training Administration.*

Em contraponto, a abordagem aqui proposta suporta mapeamentos entre as atividades dos papéis de software e seus requisitos psicológicos propostos por outros autores, trazendo como padrão um mapeamento MBTI e outro FFM, explicados no Capítulo 3. Entretanto, estes mapeamentos MBTI e FFM podem ser adaptados ou outros podem ser adotados, conforme a preferência do tomador de decisão.

Adicionalmente, Stylianou e Andreou (2012) consideram que será dada como entrada à abordagem a avaliação das habilidades técnicas e de conhecimento realizada pela organização do projeto. Nesta avaliação, a proposta adota valores no intervalo entre 0 e 1, onde quanto mais próximo de 1, maior a adequação do candidato aos requisitos técnicos, e vice-versa. Ou seja, os autores consideram como atribuição da organização a coleta e cálculo desta informação, restringindo-se a apenas definirem o intervalo de valores aceitos como entrada. Ao contrário, a abordagem proposta neste trabalho sugere formulários de coleta e a modelagem matemática para estimar o grau de habilidade técnica dos candidatos.

5.3. Otimização da alocação do RH de projetos de software

Bibi, Ahsan e Anwar (2014) propõem o uso da SBSE em uma abordagem multiobjetivo para otimização de alocação de recursos humanos a atividades. Sua proposta possui três objetivos modelados: aumentar a utilização de recursos humanos; diminuir a duração do projeto e diminuir o custo do projeto. Como critério de verificação da adequação de um profissional, em relação às atividades do projeto, é utilizada a habilidade técnica apresentada pelo mesmo. Entretanto Bibi, Ahsan e Anwar (2014) consideram que a habilidade é apenas o nível de especialização do profissional, classificando-a nos níveis especialista, moderado, intermediário ou iniciante. Além disso, os autores consideram que as características dos profissionais juntamente com o nível profissional apresentado pelos mesmos são dadas como entrada para a sua abordagem.

Após análise do trabalho de Bibi, Ahsan e Anwar (2014), observa-se que os autores consideram poucos atributos dos profissionais, apenas considerando o nível de especialização do profissional como um todo, o que pode não ser suficiente, pois um profissional pode ser especialista em determinada atividade e pouquíssimo efetivo em outras, seja por falta de conhecimento ou de experiência. Além disso, a organização poderá enfrentar dificuldades em relação a característica subjetiva desta classificação, pois não foram definidos quais critérios ou mesmo orientações de classificação. Em contrapartida, a abordagem aqui proposta sugere a aplicação de questionários que colem o nível de habilidade técnica dos candidatos e define a

fórmula para cálculo do nível de adequação técnica para os papéis, gerando assim uma forma de avaliação técnica não sujeita a subjetividade do avaliador.

Em relação ao modelo matemático, Bibi, Ahsan e Anwar (2014) consideram também o custo de atribuir a pessoa à atividade e o tempo para completar a atividade. Porém, também não há especificações de como estes valores de custo são obtidos, nem qual tipo de custo é considerado (financeiro, de horas de trabalho, entre outros). Pelas restrições modeladas em Bibi, Ahsan e Anwar (2014), não pode ser atribuído recursos humanos além dos disponíveis e o custo das atividades não pode exceder o custo total do projeto.

Como experimento, Bibi, Ahsan e Anwar (2014) implementaram a abordagem proposta nos algoritmos MOPSO, MOGA e ENSES, objetivando comparar as fronteiras de soluções não-dominadas geradas e verificar as qualidades das soluções para o problema. A conclusão apresentada por Bibi, Ahsan e Anwar (2014) é que a SBSE pode ser usada eficientemente para resolução de problemas de alocação de recursos humanos a atividades. Entretanto, não são avaliadas as qualidades das soluções, nem realizada uma comparação entre os algoritmos multiobjetivos utilizados. Em contrapartida, a abordagem aqui proposta, realiza estas avaliações, tanto da qualidade das soluções apresentadas, quanto da comparação dos desempenhos dos diferentes algoritmos utilizados.

5.4. Otimização multiobjetivo para projetos de software

Gonsalves e Itoh (2010) propõem um modelo de alocação de pessoas a tarefas baseado em habilidades (skills), no qual as pessoas são alocadas a várias tarefas em projeto de desenvolvimento de software de acordo com suas habilidades. A abordagem objetiva a minimização do custo e do tempo de duração do projeto, para isto considera que o tempo gasto na execução das tarefas varia de acordo com o nível de correspondência entre a habilidade possuída pela pessoa atribuída a tarefa e a habilidade necessária para a execução satisfatória da tarefa. Os autores utilizam uma abordagem multiobjetiva, pois para diminuir o tempo de duração do projeto pode ocorrer o aumento dos custos do mesmo, pelo fato de que os profissionais com mais habilidade podem ser mais caros. Sendo assim, foi utilizado o algoritmo multiobjetivo de enxame de partículas (MOPSO).

Em seu trabalho, Gonsalves e Itoh (2010) consideram que cada tarefa possui uma lista de habilidades requeridas. Cada habilidade tem certo nível que deve ser provido para que a tarefa possa ser executada no tempo estimado. O tempo de execução da tarefa varia em proporção à correspondência entre as habilidades necessárias para executar a tarefa e aquelas

possuídas pelo pessoal designado. As habilidades consideradas são relativas aos conhecimentos acerca de banco de dados, Java, negociação, trabalho em equipe, projeto orientado a objeto, telecomunicações, análise orientada a objeto, relacionamento com pessoas, testes e elicitação de requisitos. Estas habilidades são medidas para profissionais e assumem valores que variam de 1 a 5.

A Figura 51 ilustra um exemplo de instância do problema, no qual os profissionais candidatos (P1 a P7) são representados com suas respectivas habilidades e o seu custo por dia (moeda do Japão - yen). Porém, nenhuma informação foi dada em relação a forma de obter estas pontuações das habilidades dos profissionais.

Developmental skills	Software development personnel and thier skill levels						
	P1	P2	P3	P4	P5	P6	P7
Database	3	3	4	3	3	3	4
Java	3		4	3	3	4	4
Negotiation	2	2	3		2		3
Team work	2	2	3	2	3		3
Object oriented design		3	4				4
Telecommunications		3	3				4
Object oriented analysis		3		4			4
Relationship with people		2	3				3
Testing			3	3			
Requirements elicitation				3			
Skill lelvels total (q)	10	18	27	18	11	7	29
Personnel cost/day (yen)	15,000	25,000	30,000	20,000	10,000	15,000	35,000

Figura 51 - Profissionais candidatos e suas habilidades providas (GONSALVES; ITOH, 2010)

Por outro lado, cada tarefa contém uma lista de habilidades requeridas para desempenhá-las. Além disso, cada tarefa tem certo nível mínimo desejado para cada habilidade requerida a fim de permitir que seja executada a tempo. Sendo assim, as pessoas serão alocadas às tarefas de acordo com o nível de habilidade requerida pela tarefa versus o seu nível de habilidade possuída.

O trabalho de Gonsalves e Itoh (2010) utiliza duas funções objetivo de minimização. Uma delas minimiza o custo total do projeto, que é resultante da soma dos custos por unidade de tempo das pessoas alocadas na solução avaliada. Já a outra função objetivo minimiza a duração total do projeto, que é dado pela soma do tempo de realização das tarefas do projeto e o tempo de espera das tarefas pela disponibilidade de recursos.

As tarefas consideradas por Gonsalves e Itoh (2010) são coletadas do projeto em que a abordagem está sendo utilizada. Entretanto, os autores abordam em seu trabalho um exemplo de projeto no qual as tarefas estão ilustradas na Figura 52 e correspondem as tarefas do ciclo de

desenvolvimento: elicitação de requisitos, análise, projeto, codificação e testes. Cada uma delas possui uma lista de habilidades requeridas e o total de dias estimado para a sua execução.

SKILLS NEEDED FOR TASKS PROCESSING			
Tasks	Skills	Skill	Process
	needed for task processing	levels	time (days)
Requirements elicitation	Java	4	60
	Negotiation	5	
	Experience in telecommunications	5	
	Object oriented analysis	5	
	Tests techniques	3	
	Requirements elicitation techniques	8	
Analysis	Database	3	70
	Java	3	
	Team work	4	
	Object oriented design	4	
	Object oriented analysis	8	
	Relationship with people	5	
Design	Database	7	80
	Java	3	
	Team work	6	
	Object oriented design	7	
	Object oriented analysis	2	
Coding	Database	7	110
	Java	7	
	Team work	3	
	Experience in telecommunications	5	

Figura 52 - Tarefas e suas habilidades requeridas (GONSALVES; ITOH, 2010)

Após análise do trabalho de Gonsalves e Itoh (2010) foi verificado que as habilidades são listadas sem distinção de quais são técnicas e quais não são, tratando todas com igual importância e da mesma forma. Entretanto, esta característica da abordagem não considera a natureza conflitante que existe entre as funções objetivos das habilidades técnicas e das não-técnicas. Por exemplo, um profissional considerado com um nível adequado em Java em relação a uma atividade pode ser péssimo na habilidade de negociação e, quando isto acontece, não há definição que determine qual aspecto é mais relevante em relação às peculiaridades da atividade

em questão. Em contraponto, a proposta apresentada nesta dissertação avalia as habilidades dos candidatos de forma específica e separada tecnicamente e não tecnicamente, que são modelados em objetivos distintos inseridos em uma abordagem multiobjetiva. Além disso, nesta dissertação as habilidades não-técnicas necessárias são agrupadas e mapeadas para personalidades, pois facilita a compreensão das implicações práticas nas diferenças entre o desempenho dos profissionais.

Em adição, em Gonsalves e Itoh (2010), as habilidades não-técnicas requeridas tem que ser identificadas e listadas antecipadamente, o que pode ser mais um complicador inserido na complexa atividade do gerente, já que representa um ônus extra à organização a responsabilidade de definir quais são as atividades, seus requisitos e os níveis requeridos. Ao contrário da abordagem proposta neste trabalho que mapeia quais os atributos não-técnicos são mais adequados nas fases do software e calcula o nível possuído pelos candidatos, retirando esta responsabilidade da organização. Consequentemente, o gerente pode se dedicar a outras atividades mais relevantes ao sucesso do projeto.

5.5. Algoritmo genético multiobjetivo para alocação de RH

Jiménez-domingo, Colomo-Palacios e Gómez-Berbís (2014), em seu trabalho, propõem o MORGANA, que é uma plataforma para alocação de pessoal em projetos de software orientados a competência, formando equipes de trabalho, a qual utiliza algoritmo genético multiobjetivo. Para isto, a plataforma MORGANA considera os fatores humanos e as habilidades profissionais dos candidatos. Sendo assim, as características de aspecto humano consideradas são apresentadas na Tabela 17, classificadas em técnicas ou não-técnicas.

Em suma, o objetivo da plataforma MORGANA é formar equipes de trabalho que maximizem o desempenho do grupo através das interações de habilidades de seus membros, levando em conta fatores pessoais e profissionais. Sendo assim, o sistema foca em formar a melhor equipe global, ou seja, montar uma equipe para maximizar as habilidades de todos os profissionais alocados à equipe. Em outras palavras, o sistema priorizará as equipes que possuem conhecimento considerável na maioria dos membros, em vez da seleção de alguns especialistas em uma área ou tarefa.

Tabela 17 - Habilidades consideradas na plataforma MORGANA

Não-técnicos	Técnicos
Tomada de decisão	Sistemas Inteligentes (AI)
Capacidade de análise	Gerenciamento de Informações (DB)
Independência	Desenvolvimento de Sistemas de Informação
Inovação / criatividade	Análise de Requisitos de Negócios
Julgamento	Análise de Requisitos Técnicos
Tenacidade	Modelagem e Análise de software
Tolerância ao estresse	Design de software
Auto-organização	Verificação e validação de software
Gerenciamento de riscos	Evolução de software (Manutenção)
Conhecimento e orientação ambiental	Processo de software
Disciplina	Qualidade de software
Habilidades de negociação	Sistemas distribuídos
Serviço ao cliente	Segurança: Implementação e Gerenciamento
Empatia	Integração de Sistemas
Sociabilidade	
Trabalho em equipe / cooperação	
Avaliação de colega de trabalho	
Liderança do grupo	
Planejamento e organização	

O desenvolvimento da função objetivo é a parte mais complicada dos sistemas de otimização devido ao número de variáveis que devem ser levadas em consideração de modo a recomendar equipes adequadas. Após a análise das diferentes alternativas, Jiménez-domingo, Colomo-Palacios e Gómez-Berbís (2014) desenvolveram uma função de adequação que calcula uma média ponderada das aptidões dos membros selecionados para compor a equipe para cada habilidade considerada no sistema. Porém, algumas habilidades que possuem características semelhantes foram agrupadas e são analisadas e calculadas de maneira semelhante. Para algumas habilidades é desejável que o nível de cada membro seja alto, então o valor da pontuação da equipe aumenta. Exemplos deste grupo de habilidades são “trabalho em grupo”, “análise”, “design”, entre outras. Já para outras habilidades, valores moderados são considerados melhores, por exemplo, a liderança, pois, os autores consideram que, em equipes que possuam mais de um membro com este aspecto elevado, os conflitos têm maior probabilidade de ocorrer, porém se todos os membros apresentarem níveis baixos pode

desencadear uma situação de estagnação nas tomadas de decisão. Assim, deseja-se um grupo bem equilibrado em que todos os membros possuam um nível aceitável de liderança. Finalmente, os autores consideram um grupo de atributos nos quais um valor mais baixo gera mais benefícios para o grupo. Neste grupo, encontramos o “custo” de adicionar mais membros na equipe, para evitar que equipes muito grandes sejam formadas. Essas variações são consideradas no cálculo do valor de adequação. Então, o valor da função objetivo é diretamente influenciado pelo valor das capacidades de cada pessoa e pela função de transformação aplicada, que é diferente dependendo do tipo de aptidão associada a ela.

Diferentemente da abordagem aqui proposta, o objetivo geral da plataforma MORGANA é a formação da equipe como um todo, sem realizar a alocação das pessoas às atividades / papéis. Além disso a plataforma MORGANA não considera as habilidades necessárias por atividade, mas para todo o projeto como um todo. Ou seja, a plataforma considera as habilidades requisitadas para o projeto e, usando os níveis de habilidades das pessoas dados como entradas, a plataforma agrupa as pessoas em equipes. Porém, na prática, uma abordagem de otimização que apenas entrega um conjunto de pessoas para a alocação em um projeto pode não ser efetivo, pois é inevitável que algumas pessoas sejam excelentes em algumas habilidades e inadequadas em relação a outras. Após a recomendação da equipe formada ao gerente, caberá a ele realizar a separação das pessoas por habilidades distintas para que seja possível a alocação em tarefas. Em contrapartida, a abordagem aqui proposta identifica as necessidades de habilidades por papéis do ciclo de desenvolvimento e por pessoas, recomendando a alocação das pessoas aos papéis, o que facilita a atividade diária do gerente de projetos.

Semelhante aos trabalhos descritos em seções anteriores, os autores também não relatam o procedimento que gerou os níveis de habilidades possuídas pelas pessoas, apenas apresentando a forma que estes são utilizados em sua modelagem matemática. Este é um fator que pode desestimular a sua utilização por organizações na prática, pois estas teriam que prover métodos para quantificar o nível das habilidades de cada profissional.

5.6. Um assistente evolucionário baseado no conhecimento para escalonar projetos

No trabalho de Yannibelli e Amandi (2011) é proposta uma abordagem evolutiva mono-objetiva baseada no conhecimento que objetiva auxiliar os gerentes de projetos na fase de escalonamento de projetos de software. Portanto, a abordagem cria cronogramas automaticamente e os avalia de acordo com o objetivo de otimização que é atribuir o conjunto

mais efetivo de funcionários, considerando as atribuições de cada atividade do projeto. Para isto, a avaliação de cada cronograma é desenvolvida com base no conhecimento obtido das informações históricas, mantidas pela organização de software, sobre a participação dos colaboradores em projetos já executados anteriormente.

Na abordagem proposta por Yannibelli e Amandi (2011), as atividades do projeto requerem recursos humanos (funcionários) qualificados em diferentes áreas de conhecimento. Especificamente, cada atividade requer uma ou várias habilidades e um determinado número de funcionários para cada habilidade. Estas habilidades, chamadas também de especialidades, são relativas ao conhecimento em relação a subáreas da engenharia de software (por exemplo: programador, analista de requisitos, projetista, testador, entre outras). Além de que cada funcionário pode dominar uma ou várias habilidades.

Levando em consideração um determinado projeto, a abordagem trabalha com um conjunto que representa as habilidades requeridas para desenvolver o projeto e outro conjunto que representa os funcionários que possuem estas habilidades. Então, alguns funcionários possuem a habilidade requerida para uma certa atividade j do projeto. Um dos exemplos dados por Yannibelli e Amandi (2011) é a atividade "*programming of the architecture*" (identificada como número 6), que necessita de dois funcionários que possuam a habilidade "programador". Do exemplo, esta habilidade é possuída pelos funcionários E2, E4 e E5. Além disso, um funcionário não pode assumir mais de uma habilidade dentro de determinada atividade e não pode ser atribuído a mais de uma atividade ao mesmo tempo.

Então, para cada funcionário disponível, é possível definir diferentes contextos de trabalho. Cada um destes contextos é composto por quatro componentes principais. O primeiro componente é a atividade j à qual o funcionário r é atribuído, avaliada em relação à complexidade, domínio e características gerais. O segundo componente é uma habilidade s exigida por j e que deve ser possuída por r . O terceiro componente é o conjunto de funcionários g que também foram atribuídos a j , incluindo r . Finalmente, o quarto componente refere-se aos atributos de r , ou seja, suas habilidades, suas experiências, as relações de trabalho entre ele e os outros funcionários de g e seus conhecimentos.

Para Yannibelli e Amandi (2011), os quatro componentes descritos são considerados como os principais fatores que determinam o nível de efetividade de um funcionário. Por esta razão, a efetividade de um funcionário precisa ser definida em relação a todos os componentes do seu contexto de trabalho. Então, para cada funcionário, é possível definir diferentes níveis de efetividade em relação a diferentes contextos de trabalho. Sendo assim, dado um funcionário r , em relação a cada contexto possível, é definido um nível de efetividade que descreve quão

bem r pode desenvolver, dentro da atividade j , as tarefas associadas para a habilidade s , considerando que deve trabalhar em colaboração com os outros funcionários do conjunto g . O nível de efetividade é representado como um valor real na faixa $[0, 1]$.

Na abordagem de Yannibelli e Amandi (2011), o objetivo almejado é atribuir o conjunto mais efetivo de funcionários ao projeto. Para atingi-lo foi elaborada uma modelagem matemática para maximizar o nível médio de efetividade dos funcionários atribuídos às tarefas. Estes níveis de efetividade são calculados para os funcionários em relação aos contextos da solução avaliada. Para isto, é considerado que, em atividades de projeto de software, o nível de efetividade de um conjunto de funcionários depende do nível de efetividade de cada funcionário pertencente ao conjunto. Além disso, considera-se que os níveis individuais de efetividade são igualmente importantes na determinação da efetividade de um conjunto de funcionários.

A Figura 53 mostra um exemplo, no qual os funcionários que possuem a habilidade requerida pela "atividade 6" são listados juntamente com os possíveis contextos e os níveis de efetividade. Yannibelli e Amandi (2011) consideram que o conhecimento sobre a efetividade dos funcionários nos diferentes contextos é decorrente de informações históricas, mantidas pela empresa que gerencia o projeto. Ou seja, para conhecer o nível de efetividade de um funcionário em relação ao contexto, é necessário que o funcionário já tenha trabalhado nos mesmos contextos anteriormente. Neste ponto, pode ser percebido que a abordagem necessitará de várias execuções de projetos para montar sua base de dados com o conhecimento dos possíveis contextos, já que existem várias possibilidades de combinação. Além de que, dado a rotatividade das pessoas nas empresas, há a possibilidade desta base de dados nunca ser construída de forma a ser suficiente para utilização na abordagem, pois cada alteração das pessoas disponíveis gerará vários contextos possíveis.

Table 4
Effectivity levels of each employee that can be assigned to activity 6. For each mentioned employee, different effectivity levels are defined in relation to different work contexts.

Employee	Activity	Skill	Set of employees	Effectivity level
E2	6	Programmer	(E2, E4)	1
	6	Programmer	(E2, E5)	0.9
E4	6	Programmer	(E4, E2)	1
	6	Programmer	(E4, E5)	0.7
E5	6	Programmer	(E5, E2)	0.7
	6	Programmer	(E5, E4)	0.5

Figura 53 - Efetividades dos funcionários para a tarefa 6 do exemplo (YANNIBELLI; AMANDI, 2011)

Ainda da análise de Yannibelli e Amandi (2011), pode-se perceber também que ela não provê informações de como pode ser realizado o cálculo dos atributos dos funcionários, necessários a abordagem, considerando todos oriundos de informações históricas da organização, ou seja, não há menção em relação a forma de coleta ou medição destes dados.

5.7. Considerações finais

Nesta seção é realizada a análise comparativa entre os trabalhos discutidos neste capítulo, os quais possuem relação com o tema deste trabalho. Para isto, diante da diversificada nomenclatura em relação aos recursos humanos, encontrada na literatura, foi necessário, neste momento, visando facilitar a comparação entre eles, considerar equivalentes os termos atividades, papéis e tarefas. Já que todos se referem a unidades que compõe a execução do desenvolvimento de software e necessitam de profissionais para desempenhá-los. Além disso, existem alguns trabalhos que propõem alocação ao que é chamado neste trabalho de papel funcional, porém chamando-os de nomes diversos.

Igualmente diversificada é a nomenclatura encontrada em relação às ações principais propostas nas abordagens encontradas na literatura (por exemplo: escalonamento, alocação de recursos humanos, montagem de equipes, seleção de pessoas, entre diversas outras combinações de termos). A fim de unificar as nomenclaturas dos temas dos trabalhos, visando facilitar a comparação entre eles, foram considerados alguns temas principais, sendo: alocação de pessoas a tarefas/papéis, montagem de equipes e escalonamento de projetos. Destes, "alocação de pessoas a tarefas" foi atribuída a trabalhos que otimizam a seleção de pessoas para a execução de tarefas (atividades, tarefas e papéis) do projeto. O termo "montagem de equipes" agrupa os trabalhos que propõem abordagens que otimizam a formação de equipes para o projeto, ou seja, possuem o objetivo de alocar recursos humanos ao projeto de forma a montar equipes, considerando as especificidades do trabalho em grupo. E, finalmente, "escalonamento de projetos" é o termo que se refere a trabalhos que otimizam a criação de cronogramas para o projeto, ou seja, são aqueles que otimizam a definição de início e fim das atividades.

Além disso, os fatores considerados para a execução do foco das abordagens propostas nos trabalhos foram classificados em técnicos e não-técnicos. Os trabalhos considerados como técnicos são referentes aos que, de alguma maneira, consideram os aspectos referentes a conhecimentos de tecnologias ou áreas de negócios, experiências práticas e/ou formações acadêmicas. Por outro lado, os não-técnicos envolvem trabalhos que, de alguma maneira,

consideram os aspectos sociais ou pessoais provenientes de interações interpessoais, características psicológicas e/ou hábitos sociais.

Em relação aos dados dos atributos considerados para otimização, sejam eles técnicos ou não-técnicos, são classificados quanto a forma de coleta. As classificações foram chamadas de "especificados" ou "dados como entrada". Os "especificados" foram assim chamados quando existem, nos trabalhos, informações sobre a forma de coleta e análise das caracterizações das pessoas ou tarefas, ou seja, quando há informações quanto à existência de questionários, cálculos ou roteiros de como obter as informações de entrada. Em contrapartida, os "dados como entrada" são os que não especificam ou falam vagamente, não permitindo a replicação do procedimento de obtenção das informações das pessoas ou tarefas, considerando que todos os dados são obtidos e disponibilizados pelos gerentes de projetos. Os trabalhos relacionados são comparados na Tabela 18 em relação a estas classificações.

Tabela 18 - Comparação dos trabalhos

Trabalho	Foco da Abordagem			Atributos		Coleta dos atributos	
	Montagem de equipes	Alocação a tarefas /papéis	Escalonamento	Técnicos	Não-técnicos	Dados como entrada	Especificados
(BIBI; AHSAN; ANWAR, 2014)		X		X		X	
(GONSALVES; ITOH, 2010)		X		X	X	X	
(STYLIANOU; ANDREOU, 2012)		X		X	X	X	
(AVINASH; RAMANI, 2014)		X	X	X	X	X	
(JIMÉNEZ-DOMINGO; COLOMO-PALACIOS; GÓMEZ-BERBÍS, 2014)	X			X	X	X	
(YANNIBELLI; AMANDI, 2011)		X	X	X	X	X	
Proposta deste trabalho		X		X	X		X

Dando continuidade à comparação, os trabalhos que tratam os aspectos não-técnicos serão comparados entre si em relação aos aspectos não-técnicos considerados, ao instrumento de personalidade utilizado e mapeamento realizado. Destes, os aspectos não-técnicos

considerados se dividem em duas classificações: os que lidam com personalidade e os que lidam com microaspectos. Os trabalhos que lidam com personalidade são os que consideram algum dos instrumentos de medidas de personalidade consolidados na área da psicologia. Já os que lidam com microaspectos, foram assim classificados, pois consideram alguns aspectos componentes da personalidade ou influenciado por ela. Os trabalhos que consideram as personalidades serão identificados em relação aos instrumentos de medição de personalidade utilizados (FFM ou MBTI). E, em relação ao mapeamento, são considerados os trabalhos que utilizam teorias ou procedimentos de mapeamento entre as atividades/papéis e as suas necessidades não-técnicas. Ou seja, são considerados trabalhos que realizam ou propõem um mapeamento, os que evidenciam a forma que foi realizada a ligação entre as atividades/papéis e os fatores não-técnicos necessários para executá-las, e não apenas consideram estas necessidades como previamente conhecidas. A Tabela 19 traz esta comparação entre os trabalhos que abordam os aspectos não-técnicos em relação a estas características.

Tabela 19 - Comparação dos trabalhos que abordam aspectos não-técnicos

Trabalho	Aspectos não-técnicos		Instrumentos		Mapeamento
	Personalidade	Microaspectos	FFM	MBTI	
(BIBI; AHSAN; ANWAR, 2014)		X			
(GONSALVES; ITOH, 2010)		X			
(STYLIANOU; ANDREOU, 2012)	X		X		
(AVINASH; RAMANI, 2014)		X			
(JIMÉNEZ-DOMINGO; COLOMO-PALACIOS; GÓMEZ-BERBÍS, 2014)		X			
(YANNIBELLI; AMANDI, 2011)		X			
Proposta deste trabalho	X		X	X	X

Finalmente, os trabalhos são comparados em relação a sua formulação para o problema. São apresentadas na Tabela 20 as seguintes características: multiobjetivo ou mono-objetivo; quais são os objetivos correspondentes às funções objetivo modeladas; quais são as restrições modeladas, as quais definem a validade das soluções; e o algoritmo de busca utilizado.

Tabela 20 - Comparação dos trabalhos em relação a formulação do problema

Trabalho	Formulação		Objetivos			Restrições		Algoritmo
	Multi obj.	Mono obj.						
(BIBI; AHSAN; ANWAR, 2014)	X		Min. tempo	Min. custo	Max. utilização dos recursos	Custo	Tempo	MOPSO, MOGA, ENSES
(GONSALVES; ITOH, 2010)	X		Min. tempo	Min. custo		Disponib. das pessoas	Precedência entre tarefas	MOPSO
(STYLIANOU; ANDREOU, 2012)	X		Max. compatib. técnica	Max. compatib. não-técnica	Min. nº pessoas	Disponib. das pessoas	Tarefas c/ pessoas aptas	NSGAI
(AVINASH; RAMANI, 2014)		X	Max. eficiência	Min. custo		Disponib. das pessoas		ACO
(JIMÉNEZ-DOMINGO; COLOMO-PALACIOS; GÓMEZ-BERBÍS, 2014)	X		Max. eficiência					MOGA
(YANNIBELLI; AMANDI, 2011)		X	Max. efetividade			Disponib. das pessoas	Precedência entre tarefas	AG
Proposta deste trabalho	X		Max. compatib. técnica	Max. compatib. não-técnica		Limite de pessoas por papel		NSGAI

6. CONCLUSÃO

São raras as pessoas que desempenham perfeitamente todas as tarefas de um projeto (CAPRETZ; VARONA; RAZA, 2015). Algumas pessoas podem ser excelentes em um tipo de atividade, enquanto outras podem desempenhar a mesma atividade insatisfatoriamente. Na maioria dos casos, isso não se deve apenas à maneira ineficaz de realizar esta atividade em particular, o que pode ser uma das razões, mas é principalmente devido a não adequação da pessoa à atividade. Sendo assim, a pessoa certa para a atividade certa é de extrema importância para o sucesso do projeto como um todo (GONSALVES; ITOH, 2010).

Este requisito da pessoa certa para a atividade certa também é importante para a engenharia de software, pois a engenharia de software é uma atividade humana e realizada por diferentes profissionais cujos papéis que eles executam diferem uns dos outros e possuem necessidades técnicas e não-técnicas específicas. Assim, é importante saber quais pessoas serão mais adequadas a cada papel do projeto de desenvolvimento de software.

Diante deste problema, este trabalho abordou a fundamentação teórica sobre gerenciamento e seleção de pessoas, analisando principalmente os papéis funcionais da engenharia de software, os quais possuem requisitos técnicos (conhecimentos e habilidades de cunho técnico) e não-técnicos (personalidades) para o satisfatório desempenho das atividades que os compõem. Além de realizar uma extensa busca na literatura sobre o tema, buscando o estado da arte, constatando a existência de uma lacuna no que tange a consideração da personalidade para seleção de pessoas para projetos de software.

Nesta direção, este trabalho propõe uma abordagem para formação de equipes tecnicamente qualificadas e psicologicamente adequadas para cada papel funcional do projeto de desenvolvimento de software. O objetivo principal desta abordagem é proporcionar ao gerente de projetos uma ferramenta que o apoie na atividade de tomada de decisão de quais pessoas serão atribuídas a cada papel do ciclo de desenvolvimento de software no seu projeto.

Em relação aos trabalhos relacionados, é possível encontrar propostas que, de forma isolada, parcial ou pontual, contemplam alguns aspectos técnicos ou não-técnicos relativos às tarefas do projeto e pessoas candidatas. No entanto, em sua maioria, tais propostas não apresentam quaisquer proposições de técnicas, métodos ou metodologias para identificação e coleta de informações que viabilizem a medição e a valoração de parâmetros ou atributos adotados.

Assim sendo, a principal contribuição da abordagem proposta é a adoção da análise integrada de perfis técnicos e de personalidades, relacionados aos papéis funcionais e às pessoas, incluindo estratégias de identificação, coleta e medição dos dados requeridos, fornecendo informações de valor agregado a tomadores de decisões. Nos trabalhos relacionados, tais estratégias não são sinalizadas, impondo ao gerente de projeto a complexidade e o esforço de concebê-las.

A abordagem apresentada se enquadra em um campo da engenharia de software denominado Engenharia de Software Baseada em Buscas, do inglês *Search-Based Software Engineering* (SBSE), que consiste em formular o problema como um problema de busca, visando a posterior aplicação de algoritmos de otimização. Diante da complexidade do problema abordado, a abordagem utiliza metaheurísticas evolucionárias multiobjetivas. Nesta direção, estudos empíricos foram conduzidos para validar e avaliar a abordagem proposta. Baseando-se em sólidas referências de experimentação em engenharia de software, são apresentados um estudo de caso e experimentos, que mostraram fortes indícios da aplicabilidade da abordagem em situações reais. Os experimentos mostraram a eficácia da abordagem ao obter resultados melhores que aqueles produzidos por especialistas da área e indicaram a potencialidade dos ganhos que o uso da abordagem pode proporcionar à gerência de recursos humanos em projetos de software.

6.1. Trabalhos futuros

Há uma gama de possibilidades para condução de estudos e trabalhos futuros. Inicialmente, espera-se investigar junto a gerentes de projetos novos aprimoramentos à formulação, tais como a possibilidade de adicionar características dinâmicas à abordagem, como, por exemplo, parâmetros de desempenho em projetos anteriores. Outro parâmetro importante a ser otimizado em uma versão futura pode ser o custo total do projeto, de modo a oferecer uma alocação de pessoas que atenda às restrições orçamentárias do projeto.

Seria interessante, futuramente, analisar a real aplicação da solução selecionada pelo gerente de projetos entre as soluções não-dominadas recomendadas como saída da abordagem. Tal análise possibilitaria mensurar a efetividade da equipe selecionada e recomendada pela abordagem, inclusive podendo ser comparada com outra equipe formada de forma *ad-hoc*. Outro tipo de análise seria avaliar a satisfação pessoal dos indivíduos nos papéis em que forem alocados, o desempenho do grupo e a qualidade do software desenvolvido.

Outro trabalho futuro relevante é realizar a comparação das fronteiras de soluções não-dominadas resultantes da abordagem para as tipologias MBTI e FFM, permitindo tecer comentários sobre a qualidade das soluções produzidas pelos dois modelos. Por exemplo, identificando soluções iguais e, nas soluções diferentes, avaliando do ponto de vista da engenharia de software qual seria melhor (MBTI ou FFM). Além disso, a aplicação da abordagem em outros projetos reais, inclusive em organizações privadas, para a condução de novos estudos de caso é uma importante questão a ser realizada, a fim de se analisar o comportamento da abordagem em diferentes cenários.

Vale mencionar que a proposta contida neste trabalho pode integrar o framework de alocação de equipes de desenvolvimento em projetos distribuídos de linha de produto de software (PEREIRA *et al.*, 2010) (SANTOS *et al.*, 2010), como também pode ser usada separadamente. Para esta integração, a saída desta proposta (o conjunto de equipes sugeridas) pode ser utilizada como entrada do *framework*. O referido *framework* objetiva prover recomendações de alocações de equipes aos componentes de software em um projeto distribuído de linha de produto de software, considerando aspectos técnicos e não técnicos. Esta integração é possível, pois o *framework* tem como aspecto a independência entre as quatro fases que o compõe. É importante destacar que esta integração é prevista pelos próprios autores (SANTOS *et al.*, 2010), que afirmam que o *framework* possui os artefatos de entrada e saída das várias fases claramente definidas, sendo possível a evolução ou até mesmo a mudança das técnicas e algoritmos empregados em qualquer uma delas, sem introduzir qualquer impacto nas demais fases. Pereira *et al.* (2010) ainda afirmam que também é possível acrescentar novas fases. Neste caso, esta proposta sugere a inclusão da etapa de montagem de equipes ao *framework*.

Adicionalmente, dentre os possíveis desdobramentos deste trabalho, também podem ser realizados novos experimentos utilizando outras metaheurísticas disponíveis no jMetal, particularmente o NSGA-III (DEB; JAIN, 2014) apresentado anteriormente. Finalmente, reconhece-se que, na prática, a alocação de uma pessoa a apenas um papel pode ser muito restritiva. Assim, em futuras versões da proposta, esta restrição pode vir a ser aliviada, permitindo a alocação parcial de pessoas a diversos papéis.

REFERÊNCIAS BIBLIOGRÁFICAS

ACUNA, S. T.; JURISTO, N. **Modelling human competencies in the software process**. Proceedings of the International Workshop on Software Process Simulation Modelling, 2003.

ACUNA, S. T.; JURISTO, N.; MORENO, A. M. Emphasizing human capabilities in software development. **IEEE Software**. [S.l.]: Institute of Electrical and Electronics Engineers (IEEE) v. 23, n. 2, p.94-101, mar. 2006.

AMARAL, A. G. N. **Uma abordagem baseada em riscos de software para seleção de requisitos**. 2017. 159 f. Dissertação (Mestrado) - Curso de Mestrado em Informática, Universidade Federal da Paraíba, João Pessoa, 2017.

AMARAL, A. G. N.; ELIAS, G. A risk-driven multi-objective evolutionary approach for selecting software requirements. **Evolutionary Intelligence**. [S.l.]: Springer Science and Business Media LLC, p.1-24, mai. 2019.

ARAUJO, C. C. S.; PEDRON, C. D. **A influência do tipo de personalidade no desenvolvimento dos soft skills do gerente de projetos de TI**. II Simpósio Internacional de Gestão de Projetos, 2013.

ARAUJO, F. O. **Escolha e aplicação de critérios para formação de equipes de software: implicações para a composição de personalidade**. 105 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2014.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 10006:2006**: Sistemas de gestão da qualidade - Diretrizes para a gestão da qualidade em empreendimentos. Rio de Janeiro, 2006.

AVINASH, A.; RAMANI, K. A hybrid technique for software project scheduling and human resource allocation. **International Journal of Engineering Development and Research**. 2014.

BARRETO, A. S. **Apoio à alocação de recursos humanos em projetos de software: uma abordagem baseada em satisfação de restrições**. 2005. 93 f. Dissertação (Mestrado) - Curso de Mestrado em Ciências em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

BARROSO, A. S. et al. **Influence of human personality in software engineering - a systematic literature review**. Proceedings of the 19th International Conference on Enterprise Information Systems. [S.l.]: SCITEPRESS - Science and Technology Publications, v. 3, p.53-62, 2017.

BEJARANO, V. C.; PILATTI, L. A.; OLIVEIRA, C.; KOVALESKI, L. **Como formar equipes com o equilíbrio ideal de personalidades e perfis pessoais: a teoria e as ferramentas de Meredith Belbin**. XXXIII Congresso Brasileiro de Ensino de Engenharia, 2005.

BIBI, N.; AHSAN, A.; ANWAR, Z. **Project resource allocation optimization using search based software engineering - A framework**. Ninth International Conference on Digital Information Management (ICDIM 2014), [S.l.], p.226-229, set. 2014.

BRANCO, D.; PRIKLADNICKI, R.; CONTE, T. **Um estudo preliminar sobre tipos de personalidade em equipes Scrum**. XV Conferencia Iberoamericana de Software Engineering (CIbSE), pp. 304-311, 2012.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization. **ACM Computing Surveys**. [S.l.]: Association for Computing Machinery (ACM), v. 35, n. 3, p.268-308, set. 2003.

CAPRETZ, L. F. Personality types in software engineering. **International Journal of Human-computer Studies**. [S.l.]: Elsevier BV, v. 58, n. 2, p.207-214, fev. 2003.

CAPRETZ, L. F. **Software development and personality traits**. Tenth International Conference on Computers Applications (ICCA), 2012.

CAPRETZ, L. F.; AHMED, F. Making sense of software development and personality types. **IT Professional**. [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), v. 12, n. 1, p.6-13, jan. 2010.

CAPRETZ, L. F.; VARONA, D.; RAZA, A. Influence of personality types in software tasks choices. **Computers in Human Behavior**. [S.l.]: Elsevier BV, v. 52, p.373-378, nov. 2015.

CÉSAR, A. C. F. **Um estudo sobre as relações entre preferências por atividades técnicas e perfis de comportamento de engenheiros de software**. 2009. 104 f. Dissertação (Mestrado) - Curso de Mestrado em Ciências da Computação, Universidade Federal de Pernambuco, Recife, 2009.

CHUDASAMA, C.; SHAH, S. M.; PANCHAL, M. Comparison of parents selection methods of genetic algorithm for TSP. **International Journal of Computer Applications (IJCA)**. [S.l.]: [s.n.], 2011.

COLANZI, T. E. et al. Search based software engineering: review and analysis of the field in Brazil. **Journal of Systems and Software**, [S.l.]: Elsevier BV, v. 86, n. 4, p.970-984, abr. 2013.

COLARES, F. et al. **A new approach to the software release planning**. XXIII Brazilian Symposium on Software Engineering (SBES 2009), pp. 207-215. IEEE, 2009.

COSTA, A. S. et al. O uso do método estudo de caso na Ciência da Informação no Brasil. **InCID: Revista de Ciência da Informação e Documentação**, v. 4, n.1, p.49-69, 2013.

COSTA, P. T.; MCCRAE, R. R. **The revised NEO personality inventory (NEO-PI-R)**. The SAGE handbook of personality theory and assessment, p.179-198, 2008.

CRUZ, S.; SILVA, F. Q. B.; CAPRETZ, Luiz F. Forty years of research on personality in software engineering: A mapping study. **Computers in Human Behavior**, [S.l.]: Elsevier BV, v. 46, p.94-113, mai. 2015.

CUNHA, A. D.; GREATHEAD, D. Does personality matter? **Communications of the ACM**, [S.l.]: Association for Computing Machinery (ACM), v. 50, n. 5, p.109-112, mai. 2007.

DEB, K. **Multi-objective optimization using evolutionary algorithms**. 1. ed. [S.l.]: Wiley, 2001.

DEB, K. et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), v. 6, n. 2, p.182-197, abr. 2002.

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**, [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), v. 18, n. 4, p.577-601, ago. 2014.

DEB, K. Multi-objective optimisation using evolutionary algorithms: an introduction. **Multi-objective Evolutionary Optimisation for Product Design and Manufacturing**, [S.l.]: Springer London, p.3-34, 2011.

DIGMAN, J. M. Personality structure: emergence of the Five-Factor Model. **Annual Review of Psychology**, [S.l.], v. 41, n. 1, p.417-440, jan. 1990.

DINGSØYR, T.; RØYRVIK, E. Skills management as knowledge technology in a software consultancy company. **Advances in Learning Software Organizations**, [S.l.]: Springer Berlin Heidelberg, p.96-103, 2001.

DORRONSORO, B.; ALBA, E. **Cellular genetic algorithms**. [S.l.]: Springer, v. 42, 2008.

DURILLO, J. J. *et al.* **A study of the multi-objective next release problem**. 2009 1st International Symposium on Search Based Software Engineering, [S.l.]: IEEE, p.49-58, maio 2009.

DURILLO, J. J. *et al.* A study of the bi-objective next release problem. **Empirical Software Engineering**. [S.l.]: Springer Science and Business Media LLC, v. 16, n. 1, p.29-60, dez. 2010.

DURILLO, J. J.; NEBRO, A. J. jMetal: A java framework for multiobjective optimization. **Advanced Engineering Software**, v. 42, p. 760-771, out. 2011.

EIBEN, A. E.; SMITH, J. E. **Introduction to evolutionary computing**. Springer-Verlag, v.53, 2003.

FAY, M. P.; PROSCHAN, M. A. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. **Statistics Surveys**, v. 4, p. 1-39, jul. 2009. ISSN 1935-7516.

FELDT, R. *et al.* Links between the personalities, views and attitudes of software engineers. **Information and Software Technology**. [S.l.]: Elsevier BV, v. 52, n. 6, p.611-624, jun. 2010.

FERREIRA, A.; SILVA, F. Q. **Fatores humanos que influenciam a utilização de processos de software**. Proceedings of the 7th Brazilian Symposium on Software Quality, p. 123-138, 2008.

FILITTO, D. **Algoritmos genéticos**: uma visão explanatória. Revista Multidisciplinar da UNIESP. Saber Acadêmico, n. 6, p. 136-143, 2008.

FLORES-MENDONZA, C. E. **Inventário de personalidade NEO-Revisado**: Manual técnico, São Paulo – SP, Vetor Editora, 2007.

GONSALVES, T.; ITOH, K. Multi-objective optimization for software development projects. **International Multi Conference of Engineers and Computer Scientists**, v.1, 2010.

GORLA, N.; LAM, Y. W. Who should work with whom? **Communications of the ACM**. [S.l.]: Association for Computing Machinery (ACM), v. 47, n. 6, p.79-82, 1 jun. 2004.

GOSLING, S. D.; RENTFROW, P. J.; SWANN, W. B. A very brief measure of the Big-Five personality domains. **Journal of Research in Personality**. [S.l.]: Elsevier BV, v. 37, n. 6, p.504-528, dez. 2003.

GULATI, J.; BHARDWAJ, P.; SURI, B. **Comparative study of personality models in software engineering**. Proceedings of the Third International Symposium on Women in Computing and Informatics - WCI '15. [S.l.]: ACM Press, p.209-216, 2015.

HARMAN, M. **The current state and future of search based software engineering**. Future of Software Engineering (FOSE '07). [S.l.]: IEEE, p.342-357, mai. 2007.

HARMAN, M.; MANSOURI, S. A.; ZHANG, Y. **Search based software engineering**: A comprehensive analysis and review of trends techniques and applications. Department of Computer Science, King's College London, Technical Report TR-09-03, 2009.

HARMAN, M. et al. Empirical software engineering and verification. In: **Search Based Software Engineering**: Techniques, Taxonomy, Tutorial. Berlin, Heidelberg: Springer-Verlag, p. 1-59, 2012.

HERBSLEB, J. D.; MOCKUS, A.; ROBERTS, J. A. **Collaboration in software engineering projects**: a theory of coordination. Proceedings International Conference on Information Systems (ICIS), Milwaukee, 2006.

JIMÉNEZ-DOMINGO, E.; COLOMO-PALACIOS, R.; GÓMEZ-BERBÍS, J. M. **A multi-objective genetic algorithm for software personnel staffing for HCIM solutions**. International Journal of Web Portals. [S.l.]: IGI Global, v. 6, n. 2, p.26-41, abr. 2014.

JUNG, C. G. **Tipos psicológicos**. Petrópolis: Vozes. 1991.

LIKERT, R. A Technique for the Measurement of Attitudes. **Archives of Psychology**, v. 140, n. 55, 1932.

LIMA, G. A. **Uma abordagem para seleção automática de bugs em projetos de software baseados em componentes**. 2018. 103 f. Dissertação (Mestrado) - Curso de Mestrado em Informática, Universidade Federal da Paraíba, João Pessoa, 2018.

LUCAS, D. C. **Algoritmos genéticos: uma introdução**. Universidade Federal do Rio Grande do Sul–Porto Alegre, 2002.

LUI, K. M.; CHAN, K. C. C. **Programming task demands**, 5th IEEE International Conference on Cognitive Informatics (ICCI), pp. 765-770, 2006.

MAIA, C. L. B. et al. An evolutionary optimization approach to software test case allocation. **Communications in Computer and Information Science**. [S.l.]: Springer Berlin Heidelberg, p.637-641, 2011.

MARTÍNEZ, L. G. et al. **Knowing software engineer's personality to improve software development**. International Conference on Software and Data Technologies, p.99-104, 2011.

MARTINS, S. L.; RIBEIRO, C. C. Metaheuristics and applications to optimization problems in telecommunications. **Handbook of Optimization in Telecommunications**. [S.l.]: Springer US, p.103-128, 2006.

MERRIAM, S. B. **Qualitative Research: A Guide to Design and Implementation**, 2. Ed, San Francisco, CA: Jossey-Bass, 2009.

MYERS, I. B.; MCCAULLEY, M. H.; MOST, R. **Manual, a guide to the development and use of the Myers-Briggs type indicator**, v. 3. Consulting Psychologists Press, 1998.

MONTGOMERY D.; RUNGER, G. C. **Estatística aplicada e probabilidade para Engenheiros**. Rio de Janeiro: LTC, 4ª edição, 2009.

NASCIMENTO, A. M. **Curso de direito do trabalho**. 26ª ed. São Paulo: Saraiva, 2011.

NEBRO, A. J. et al. MOCeLL: A cellular genetic algorithm for multiobjective optimization. **International Journal of Intelligent Systems**, v. 24, n. 7, p. 726-746, jul. 2009.

NEBRO, A. J.; DURILLO, J. J. **jMetal 4.5 User Manual**. [S.l.], p. 88. 2014.

PEREIRA, T. A. B. et al. **A recommendation framework for allocating global software teams in software product line projects**. Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering - RSSE '10. [S.l.]: ACM Press, p.36-40, 2010.

PROJECT MANAGEMENT INSTITUTE. **Project Management Body of Knowledge - PMBOK**. 5 ed. João Pessoa, 2014.

REBOLLO, I.; HARRIS, J. R. Genes, personalidade e ambiente. In: FLORES-MENDOZA, Carmen; COLOM, Roberto. **Introdução à psicologia das diferenças individuais**. Porto Alegre: Artmed, p. 300-322, 2006.

REHMAN, M. et al. **Mapping job requirements of software engineers to Big Five Personality traits**. 2012 International Conference on Computer & Information Science (ICCIS), [S.l.]: IEEE, p.1115-1122, jun. 2012.

REIS, B. F.; GRAMINHO, V. M. C. **A inteligência artificial no recrutamento de trabalhadores: o caso Amazon analisado sob a ótica dos direitos fundamentais**. Seminário Internacional Demandas Sociais e Políticas Públicas na Sociedade Contemporânea, 2019.

ROCHA, I. M. **Uma abordagem otimizada para o problema de alocação de equipes e escalonamento de tarefas para a obtenção de cronogramas eficientes**. 2011. 120 f. Dissertação (Mestrado) - Curso de Mestrado em Ciência da Computação, Centro de Ciências e Tecnologia, Universidade Estadual do Ceará, Fortaleza, 2011.

RUTHERFOORD, R. H. **Using personality inventories to help form teams for software engineering class projects**. Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education - ITICSE '01. [S.l.]: ACM Press, p.73-76, 2001.

SACH, R.; PETRE, M.; SHARP, H. **The use of MBTI in software engineering**. 22nd Annual Psychology of Programming Interest Group, p.19-22, 2010.

SANTOS, V. S. **Uma abordagem para seleção de equipes tecnicamente qualificadas para implementação de projetos de software**. 2014. 154 f. Dissertação (Mestrado) - Curso de Mestrado em Informática, Universidade Federal da Paraíba, João Pessoa, 2014.

SANTOS, V.; PEREIRA, T. A. B.; RIBEIRO, B. L.; ELIAS, G. (2010). **Um framework de recomendação para alocação de equipes de desenvolvimento em projetos distribuídos de linhas de produto de software**. IV Workshop de Desenvolvimento Distribuído de Software (WDDS), pp. 42-49, 2010.

SCHNAIDER, L. R. C. **Planejamento da alocação de recursos humanos em ambientes de desenvolvimento de software orientados à organização**. 2003. Dissertação (Mestrado) - Curso de Mestre em Ciências da Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

SEADA, H; DEB, K. **U-NSGA-III: a unified evolutionary algorithm for single, multiple, and many-objective optimization**. COIN Report 2014022, Michigan State University, 2014.

SHANTEAU, J. *et al.* Performance-based assessment of expertise: How to decide if someone is an expert or not. **European Journal of Operational Research**. [S.l.]: Elsevier BV, v. 136, n. 2, p.253-263, jan. 2002.

SHARP, H.; ROBINSON, H. Some social factors of software engineering. **ACM Sigsoft Software Engineering Notes**. [S.l.]: Association for Computing Machinery (ACM), v. 30, n. 4, p.1-6, jul. 2005.

SILVA, D. R. D. **Atores sintéticos em jogos sérios: uma abordagem baseada em psicologia organizacional**. 2009. 259 f. Tese (Doutorado) - Curso de Doutorado em Ciência da Computação, Universidade Federal de Pernambuco, Pernambuco, 2009.

SILVA, F. E. O. **Uma abordagem de combinação social para apoiar a formação de equipes**. 2009. 167 f. Dissertação (Mestrado) - Curso de Mestrado em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

SILVA, M. A. **WebAPSEE-Planner: Auxílio à alocação de pessoas em projetos de software através de políticas**. Dissertação (Mestrado). Universidade Federal do Pará, 2007.

SILVA, M. H. L. **Soft skills do programador de software: abordagem conceitual e definição de métricas para identificação automática no contexto de um sistema de juiz online**. 2015. 68 f. Tese (Doutorado) - Curso de Mestrado em Informática, Universidade Federal de Alagoas, Maceió, 2015.

SIVANANDAM, S. N.; DEEPA, S. N. **Introduction to Genetic Algorithms**. Berlim: Springer, p.15-37, 2008.

SODIYA, A. S. et al. An improved assessment of personality traits in software engineering. **Interdisciplinary Journal of Information, Knowledge and Management**, vol. 2, 2007.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, B. P. et al., Explorando a personalidade do desenvolvedor em ecossistemas de software móvel, **I Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES)**, pp. 51-60, 2016.

SOUZA, J. T. et al. **The human competitiveness of search based software engineering**. 2nd International Symposium on Search Based Software Engineering, [S.l.], p.143-152, set. 2010. IEEE.

SOUZA, J. et al. A preliminary systematic mapping study of human competitiveness of SBSE. **Search-based Software Engineering**. [S.l.]: Springer International Publishing, p.131-146, 2018.

SOUZA, M. J. F. **Inteligência computacional para otimização**. Notas de aula, Departamento de Computação, Universidade Federal de Ouro Preto, 2008.

SOUZA, V.; ELIAS, G. **A Fuzzy-Based approach for selecting technically qualified distributed software development teams**. Mexican International Conference on Artificial Intelligence. [S.l.]: Springer, p.221-235, 2018.

SRINIVAS, N.; DEB, K. Multiobjective function optimization using nondominated sorting genetic algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221-248, 1995.

SRINIVASAN, J.; LUNDQVIST, K. **Agile in India**. Proceedings of the 3rd India Software Engineering Conference on India Software Engineering Conference - ISEC '10. [S.l.]: ACM Press, p.125-130, 2010.

STYLIANOU, C.; ANDREOU, A. S. A Multi-objective genetic algorithm for software development team staffing based on personality types. **IFIP Advances in Information and Communication Technology**. [S.l.]: Springer, p.37-47, 2012.

TORRES, F. C. **Uma nova abordagem baseada em otimização multi-objetivo para o planejamento de releases em projetos de software**. 2010. 90 f. Dissertação (Mestrado) - Curso de Mestrado em Ciências da Computação, Universidade Federal de Minas Gerais, 2010.

VARONA, D. et al. Evolution of software engineers' personality profile. **ACM Sigsoft Software Engineering Notes**. [S.l.]: Association for Computing Machinery (ACM), v. 37, n. 1, p.1-5, 27 jan. 2012.

WANGSOM, P.; BOUVRY, P.; LAVANGNANANDA, K. **Extreme Solutions NSGA-III (E-NSGA-III) for scientific workflow scheduling on cloud**. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). [S.l.]: IEEE, p.1139-1146, dez. 2018.

WEISS, D. J.; SHANTEAU, J.; HARRIES, P. People who judge people. **Journal of Behavioral Decision Making**. [S.l.]: Wiley, v. 19, n. 5, p.441-454, 2006.

WERNER, F. **Genetic algorithms for shop scheduling problems: A survey**. Universidade de Otto-von-Guericke. Magdeburg, p. 31. 2011.

WOHLIN, C. *et al.* **Experimentation in software engineering**, Springer Verlag, 2012.

YANNIBELLI, V.; AMANDI, A. A knowledge-based evolutionary assistant to software development project scheduling. **Expert Systems with Applications**. [S.l.]: Elsevier BV, v. 38, n. 7, p.8403-8413, jul. 2011.

YILMAZ, M. *et al.* An examination of personality traits and how they impact on software development teams. **Information and Software Technology**. [S.l.]: Elsevier BV, v. 86, p.101-122, jun. 2017.

YIN, R. K. **Estudo de caso: planejamento e métodos**, Bookman, 2014.

ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. **IEEE Transactions on Evolutionary Computation**. [S.l.]: IEEE, v. 3, n. 4, p.257-271, 1999.

ZITZLER, E; LAUMANN, M; THIELE, L. **SPEA2: improving the strength Pareto evolutionary algorithm**. Technical Report TIK-Report 103, Computer Engineering and Networks Laboratory, Department of Electrical Engineering, Swiss Federal Institute of Technology, Zurich, 2001.

Apêndice A. jMetal

O Capítulo 2 descreveu os conceitos relativos às metaheurísticas, com um enfoque maior nos algoritmos evolucionários, devido a fácil adaptação e aplicação em problemas com múltiplos objetivos distintos, que é o caso do problema tratado por este trabalho. Após ter o domínio conceitual destas metaheurísticas, é necessário implementá-las em alguma linguagem de programação ou encontrar implementações pré-existentes e adaptá-las para o modelo proposto.

Caso a primeira alternativa seja escolhida, ou seja, se opte pela implementação de uma versão destes algoritmos, além do maior tempo necessário para a programação dos mesmos, dois desafios ainda seriam enfrentados. O primeiro, que o algoritmo deveria ser validado em relação aos seus resultados, em geral, isto é feito a partir de um “teste de sanidade” (HARMAN *et al.*, 2012), que visa demonstrar que o algoritmo supera a qualidade demonstrada por um algoritmo aleatório. Além da necessidade da utilização de algum problema clássico de otimização para ser um *benchmarking* para atestar estes resultados. Portanto, seria necessária uma bateria de testes diretamente relacionados com o modelo proposto para os problemas multiobjetivos. Em segundo lugar, uma verificação minuciosa da implementação do algoritmo seria necessária, demonstrando que sua modelagem conceitual e seus demais procedimentos (cruzamento, mutação, seleção e substituição) foram realizados corretamente.

Em contrapartida, a segunda alternativa, que é a seleção de implementações pré-existentes, aparenta ser mais razoável, pois economizará o tempo necessário para implementação, já que foram validadas por terceiros e estão prontas para uso. Entretanto é necessária a observação de uma linguagem de programação uniforme, de modo a eliminar possíveis variáveis que possam perturbar o resultado final dos experimentos, oriundas da diferença de desempenho existente entre linguagens.

Nesta direção, o framework jMetal (DURILLO; NEBRO, 2011) oferece uma solução, em que todos os algoritmos abordados neste trabalho foram implementados e estão prontos para uso, necessitando apenas de configuração e implementação do problema a ser avaliado. O jMetal (*Metaheuristics Algorithms in Java*) é um *framework* designado a resolver problemas de otimização para múltiplos objetivos, apesar de também implementar algoritmos mono-objetivos. O jMetal possui duas versões, a versão tradicional 4.5.2 e a versão mais recente 5.2. Entretanto, a documentação da versão mais recente se mostrou deficiente, pois possui informações faltantes, exigindo um maior esforço do usuário no aprendizado do *framework*, já

a versão anterior possui um manual oficial completo que descreve seu funcionamento em detalhes (NEBRO; DURILLO, 2014). Além disso, a versão 4.5.2 inclui alguns algoritmos que foram excluídos da nova versão, além de algumas medidas de desempenho exclusivas. Vale salientar, que a semelhança entre ambas as versões não prejudica a compreensão do *framework*.

A Tabela 21 descreve os principais recursos oferecidos pelo jMetal. Nela pode ser observado que, adicionalmente às metaheurísticas descritas neste trabalho, o *framework* oferece as medidas de desempenho de Hipervolume e Espalhamento, descritas no Capítulo 2 deste trabalho e que são utilizadas como medidas de avaliação dos experimentos realizados nesta abordagem. Porém algumas implementações padrões do jMetal foram levemente modificadas para atender às necessidades da proposta, por exemplo, a codificação do problema, a geração da população inicial, o cruzamento e a mutação.

Tabela 21 - Sumário de recursos disponíveis no jMetal 4.5.2 (AMARAL, 2017)

Algoritmos multiobjetivos	NSGA-II, SPEA2, PAES, PESA-II, OMOPSO, MOCell, AbYSS, MOEA/D, Densea, CellDE, GDE3, FastPGA, IBEA, SMPSO, SMS-EMOA, dMOPSO, WASFGA, GWSFGA, MOCHC, MOMBI, MOMBI-II, NSGA-III, WASF-GA, GWASF-GA
Algoritmos mono-objetivos	Genético, Estratégia de Evolução, DE, CMA-ES, PSO, recifes de corais
Indicadores de qualidade	Distância geracional, distância geracional invertida, épsilon aditivo, espalhamento, hipervolume, R2
Codificação do problema	Binário, binário codificado como real, inteiro, inteiro codificado como real, permutação, real.
Operadores de seleção	Torneio, por melhor fitness, por evolução diferencial, aleatória e por ranking
Operadores de cruzamento	Ponto único, dois pontos, metade uniforme, mapeamento parcial, simulação binária, evolucionário diferencial e mistura alfa.
Operadores de mutação	<i>Flipping</i> , não-uniforme, polinomial, uniforme e por substituição.

O *framework* foi projetado seguindo quatro objetivos: (i) *ser simples e fácil de usar*: caso contrário, não teria grande adoção por usuários. Isto é definido na organização de componentes do *framework*, em que cada um é responsável por uma tarefa específica e não ambígua; (ii) *ser flexível*: denotando que todos os algoritmos devem ser executados de maneira uniforme, independentemente da codificação do problema ou dos operadores genéticos escolhidos; (iii) *ter portabilidade*: obtida com o uso da linguagem Java; e (iv) *ser extensível*: define que novos operadores, metaheurísticas e problemas devem ser incorporados com facilidade ao jMetal.

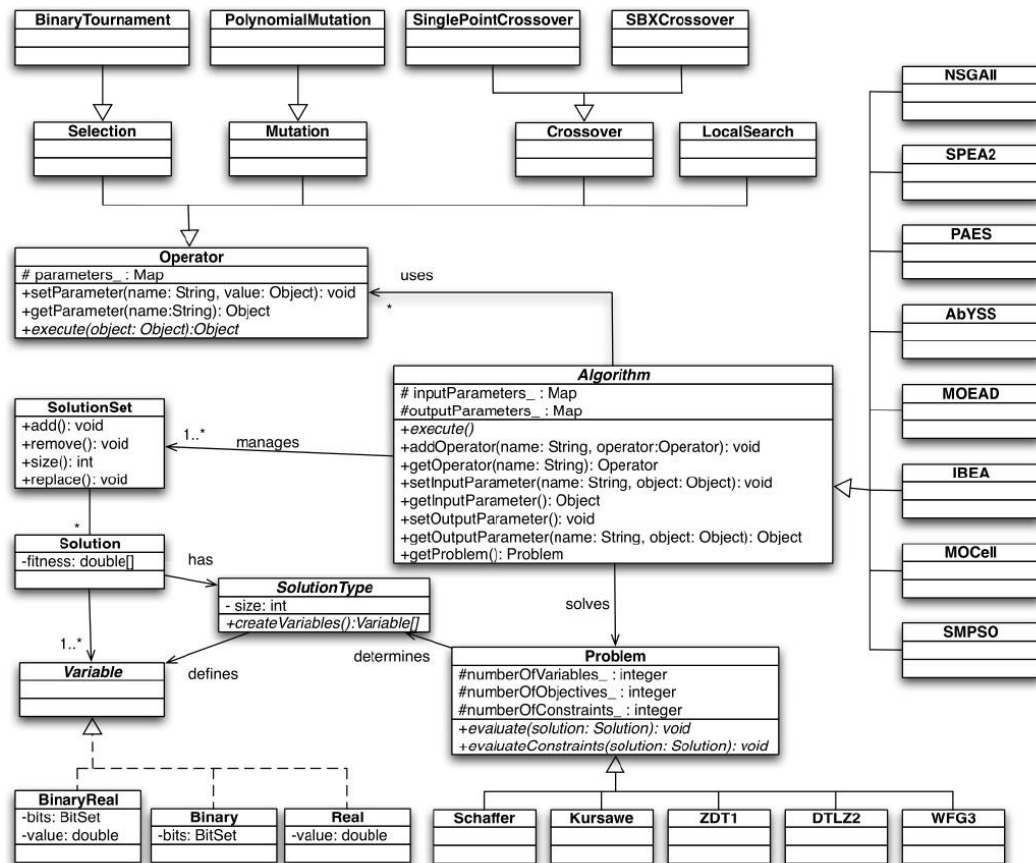


Figura 54 -Diagrama de classes do jMetal (NEBRO e DURILLO, 2014)

A Figura 54 ilustra um diagrama UML reduzido do jMetal, demonstrando as principais classes envolvidas na execução de um algoritmo. Nesta direção, a representação da solução é feita em objetos do tipo “Variable”, formando um cromossomo da população administrada pelo algoritmo na classe “SolutionSet”. A solução definida em um conjunto de variáveis possui um valor de fitness associado, assim como um tipo de codificação que deve ser compatível com o problema a ser executado. O diagrama elenca alguns problemas pré-implementados no jMetal, que podem ser usados para testar o funcionamento das metaheurísticas disponíveis, incluídos como uma forma de *benchmarking* (Schaffer, Kursawe, ZDT1, DTLZ2 e WFG3). É importante notar que os operadores de seleção, mutação e cruzamento são definidos separadamente do algoritmo. Além destes três tipos de operadores, também é disponibilizado um operador de busca local, podendo ser combinado em auxílio com o algoritmo principal. Além disso, pode ser notado que a classe “Problem” admite três critérios principais: o tamanho do cromossomo, definido pelo atributo “*numberOfVariables*”; a quantidade de objetivos a serem otimizados, definido pelo atributo “*numberOfObjectives*”; e a definição de restrições, se existirem, definidas pelo atributo “*numberOfConstraints*”.

Apêndice B. Termo de autorização da participação da organização na pesquisa

Eu, Nome Completo do Responsável abaixo assinado, como Função do Responsável, concordo que o Nome da Organização participe da pesquisa de natureza acadêmica que tem como objetivo geral fornecer uma abordagem de seleção de pessoas aptas a projetos de desenvolvimento de software, formando equipes com a divisão de papéis necessários ao desenvolvimento de todo projeto. Fui devidamente informado e esclarecido pela pesquisadora Jorcyane Araújo Lima sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de participação da organização. Estou ciente e fui esclarecido de que devo optar se o nome da organização deverá ser mantido em sigilo. Também fui devidamente informado que posso optar, neste termo, por um sigilo parcial, escolhendo onde o seu nome poderá ser divulgado. Enfim, tendo sido orientado quanto ao teor de todo o conteúdo aqui mencionado e compreendido a natureza e o objetivo do já referido estudo, como responsável, manifesto meu livre consentimento da participação da organização, estando totalmente ciente de que não há nenhum valor econômico ou material a receber, ou a pagar, nem a mim, nem à organização, pela participação.

O nome da organização poderá ser divulgado:

- ☐ Em Publicações (Por exemplo, jornais, revistas, artigos científicos)
- ☐ Nos agradecimentos dos trabalhos de dissertação de mestrado e teses de doutorado
- ☐ Divulgações

João Pessoa, __ de _____ de _____

Assinatura do Responsável

Assinatura da Pesquisadora

Apêndice C. Termo de consentimento livre e esclarecido

Pesquisa desenvolvida por Jorcyane Araújo Lima para obtenção do título de mestre pela Universidade Federal da Paraíba (UFPB), no programa de Mestrado em Informática, sob orientação do professor Dr. Glêdson Elias da Silveira. A pesquisa intitulada: “Uma abordagem multiobjetiva para formação de equipes em projeto de desenvolvimento de software baseada em aspectos técnicos e não-técnicos”, tem como objetivo geral fornecer uma abordagem de seleção de pessoas aptas a projetos de desenvolvimento de software, formando equipes com a divisão de papéis necessários ao desenvolvimento de todo projeto. Na qual sejam consideradas: Necessidades técnicas e as características psicológicas ideais para cada papel do desenvolvimento de software, habilidades técnicas e características psicológicas de cada profissional candidato a integrar a equipe do projeto. Este estudo não implica em nenhum risco para sua saúde, apenas riscos mínimos previsíveis como o desconforto para responder alguns itens do instrumento. Portanto, as perguntas que possam causar desconforto não precisarão ser respondidas, e o questionário pode ser interrompido a qualquer momento. Ressaltamos que você poderá desistir da pesquisa a qualquer momento. Para fins de pesquisa, os pesquisadores garantem que seu nome será preservado e que nenhum dado sobre sua pessoa será divulgado. Eu, _____, fui informado(a) dos objetivos da pesquisa acima de maneira clara e detalhada. Recebi informação a respeito do método que será utilizado. Sei que em qualquer momento poderei solicitar novas informações e modificar minha decisão se assim eu desejar. Fui igualmente informado da garantia de receber resposta a qualquer dúvida acerca dos procedimentos, da liberdade de tirar meu consentimento, a qualquer momento, e da garantia de que não serei identificado(a) quando da divulgação dos resultados e que as informações obtidas serão utilizadas apenas para fins científicos vinculados ao presente projeto de pesquisa e que posso entrar em contato com a pesquisadora responsável Jorcyane Araújo Lima, através do telefone pessoal (**) *****-*****, email: *@*.*.br, se assim desejar.

João Pessoa, __ de _____ de _____

Assinatura

Apêndice D. Questionário de coleta dos aspectos técnicos das pessoas

Prezado(a) colega de TI,

Solicitamos a sua colaboração para responder o presente questionário sobre sua personalidade e seus conhecimentos na área de desenvolvimento de software. A sua participação será muito importante para obtermos sucesso na consecução de nossa pesquisa de mestrado intitulada "Uma abordagem multiobjetiva para formação de equipes em projeto de desenvolvimento de software baseada em aspectos técnicos e não-técnicos". Desde já agradecemos sua colaboração.

***Obrigatório**

Termo de consentimento livre e esclarecido

Pesquisa desenvolvida por Jorcyane Araújo Lima para obtenção do título de mestre pela Universidade Federal da Paraíba (UFPB), no programa de Mestrado em Informática, sob orientação do professor Dr. Glêdson Elias da Silveira. A pesquisa intitulada: "Uma abordagem multiobjetiva para formação de equipes em projeto de desenvolvimento de software baseada em aspectos técnicos e não-técnicos", tem como objetivo geral fornecer uma abordagem de seleção de pessoas aptas a projetos de desenvolvimento de software, formando equipes com a divisão de papéis necessários ao desenvolvimento de todo projeto. Na qual sejam consideradas: Necessidades técnicas e as características psicológicas ideais para cada papel do desenvolvimento de software, habilidades técnicas e características psicológicas de cada profissional candidato a integrar a equipe do projeto. Este estudo não implica em nenhum risco para sua saúde, apenas riscos mínimos previsíveis como o desconforto para responder alguns itens do instrumento. Portanto, as perguntas que possam causar desconforto não precisarão ser respondidas e o questionário pode ser interrompido a qualquer momento. Ressaltamos que você poderá desistir da pesquisa a qualquer momento. Para fins de pesquisa, os pesquisadores garantem que seu nome será preservado e que nenhum dado sobre sua pessoa será divulgado.

Eu, _____, fui informado(a) dos objetivos da pesquisa acima de maneira clara e detalhada. Recebi informação a respeito do método que será utilizado. Sei que em qualquer momento poderei solicitar novas informações e modificar minha decisão se assim eu desejar. Fui igualmente informado da garantia de receber resposta a qualquer dúvida acerca dos procedimentos, da liberdade de tirar meu consentimento, a qualquer momento, e da garantia de que não serei identificado(a) quando da divulgação dos resultados e que as informações obtidas serão utilizadas apenas para fins científicos vinculados ao presente projeto de pesquisa e que posso entrar em contato com a pesquisadora responsável Jorcyane Araújo Lima, através do telefone pessoal (_____) _____, email: _____, se assim desejar.

1. Você concorda com o Termo de consentimento livre e esclarecido? *

Marcar apenas uma oval.

☐ Sim *Passe para a pergunta 2.*

☐ Não *Passe para "Não concorda."*

Não concorda

Ao discordar do termo de consentimento, você não poderá continuar a pesquisa. Muito obrigada pela atenção!

Pare de preencher este formulário.

Dados de Identificação

Serão utilizados somente pelo pesquisador

2. Nome *

3. Endereço de e-mail

Passe para a pergunta 4.

Habilidades técnicas**Análise orientada a objetos**

4. Há quantos anos você trabalha com análise orientada a objetos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

5. Quantos projetos você desenvolveu utilizando análise orientado a objetos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Design orientado a objetos

6. Há quantos anos você trabalha com design orientado a objetos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

7. Quantos projetos você desenvolveu utilizando design orientado a objetos? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Framework Django

8. Há quantos anos você trabalha com Framework Django? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

9. Quantos projetos você desenvolveu utilizando Django? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Git

10. Há quantos anos você trabalha com Git? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

11. Quantos projetos você desenvolveu utilizando Git? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

HTML

12. Há quantos anos você trabalha com HTML? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

13. Quantos projetos você desenvolveu utilizando HTML? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

14. Você possui a certificação em HMTL listada abaixo?*Marcar tudo o que for aplicável.*☐ Microsoft Specialist em HTML5, CSS3 e JavaScript

JavaScript

15. Há quantos anos você trabalha com JavaScript? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

16. Quantos projetos você desenvolveu utilizando JavaScript? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

17. Você possui a certificação em JavaScript listada abaixo?*Marcar tudo o que for aplicável.*☐ Microsoft Specialist em HTML5, CSS3 e JavaScript

Matriz de rastreabilidade dos requisitos

18. Há quantos anos você trabalha com Matriz de rastreabilidade dos requisitos? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

19. Quantos projetos você utilizou Matriz de rastreabilidade dos requisitos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

PostgreSQL

20. Há quantos anos você trabalha com PostgreSQL? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

21. Quantos projetos você desenvolveu utilizando PostgreSQL? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Prototipagem

22. Há quantos anos você trabalha com prototipagem? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

23. Quantos projetos você utilizou Prototipagem? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Python

24. Há quantos anos você trabalha com Python? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

25. Quantos projetos você desenvolveu utilizando Python? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

26. Quais dessas certificações você possui em Python?*Marcar tudo o que for aplicável.*

- ☐ PCEP-30
- ☐ PCAP-31
- ☐ PCPP-32-1
- ☐ PCPP-32-2

PyCharm

27. Há quantos anos você trabalha com PyCharm? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

28. Quantos projetos você desenvolveu utilizando PyCharm? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

UML

29. Há quantos anos você trabalha com UML? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

30. Quantos projetos você desenvolveu utilizando UML? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

31. Quais dessas certificações você possui em UML? **Marcar tudo o que for aplicável.*

- ☐ OMG
- ☐ IBM Rational

Selenium

32. Há quantos anos você trabalha com Selenium? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

33. Quantos projetos você utilizou Selenium? **Marcar apenas uma oval.*

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Suíte de testes Django

34. Há quantos anos você trabalha com Suíte de testes Django? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

35. Quantos projetos você utilizou Suíte de testes Django? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Técnicas de criatividade em grupo

Brainstorming, Delphi, mapa mental, entre outras

36. Há quantos anos você trabalha com Técnicas de criatividade em grupo? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

37. Quantos projetos você utilizou Técnicas de criatividade em grupo? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Técnicas e ferramentas para o processo de coleta dos requisitos

Entrevistas, grupos de discussão, workshops, entre outras

38. Há quantos anos você trabalha com Técnicas e ferramentas para o processo de coleta dos requisitos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 3 anos
- ☐ 3 a 5 anos
- ☐ 5 a 7 anos
- ☐ 7 a 9 anos
- ☐ 9 ou mais

39. Quantos projetos você utilizou Técnicas e ferramentas para o processo de coleta dos requisitos? *

Marcar apenas uma oval.

- ☐ Nenhum
- ☐ 1 a 5
- ☐ 5 a 10
- ☐ 10 a 15
- ☐ 15 a 20
- ☐ 20 a 25
- ☐ 25 a 30
- ☐ 30 ou mais

Apêndice E. Questionário de coleta dos aspectos não-técnicos das pessoas

Aspectos psicológicos

40. Classificação da sua personalidade MBTI
(Exemplo: ESTJ) *

Teste recomendado disponível no endereço:

*****.*****

Classificação da sua personalidade FFM

Teste recomendado disponível no endereço: *****.*****

41. Abertura a Experiência (O) *

Número entre 0 e 100

42. Conscienciosidade (C) *

Número entre 0 e 100

43. Extroversão (E) *

Número entre 0 e 100

44. Amabilidade (A) *

Número entre 0 e 100

45. Neuroticismo (N) *

Número entre 0 e 100

Com tecnologia



Google Forms

Apêndice F. Perfis de personalidades das pessoas participantes

Participantes	MBTI	FFM				
		Abertura a Experiência (O)	Conscienciosidade (C)	Extroversão (E)	Amabilidade (A)	Neuroticismo (N)
Pessoa 1	ENTJ	38	35	67	45	32
Pessoa 2	ENTJ	49	35	58	35	24
Pessoa 3	INFP	55	29	35	60	25
Pessoa 4	INTJ	48	52	47	67	30
Pessoa 5	INTJ	51	48	55	56	43
Pessoa 6	ISFP	20	37	35	63	21
Pessoa 7	ISTJ	71	49	40	61	35
Pessoa 8	ISTJ	56	75	32	89	52
Pessoa 9	ESTP	38	20	73	42	57
Pessoa 10	ENFJ	82	46	65	30	24
Pessoa 11	ENFJ	56	45	87	34	12
Pessoa 12	INTJ	67	76	45	32	54
Pessoa 13	ENTJ	34	56	66	24	76
Pessoa 14	INFJ	90	92	43	63	5
Pessoa 15	ISFJ	54	76	41	78	43
Pessoa 16	ENFJ	45	78	58	54	8
Pessoa 17	INFJ	43	24	45	45	34
Pessoa 18	INTJ	34	56	46	35	78
Pessoa 19	INTJ	54	34	56	76	90
Pessoa 20	INTJ	45	56	53	45	67
Pessoa 21	ISFJ	45	67	23	67	12
Pessoa 22	ISTJ	67	87	32	65	87
Pessoa 23	ISTJ	65	63	56	8	45
Pessoa 24	ESFJ	54	51	76	90	32
Pessoa 25	ISFP	43	23	56	43	4
Pessoa 26	ESFP	34	65	98	87	5
Pessoa 27	ENTP	34	23	89	12	62
Pessoa 28	INTP	54	65	7	98	23
Pessoa 29	ENTP	34	56	89	43	56
Pessoa 30	ISFP	47	14	16	60	40