

Caio Marcelo Pereira Melo

Aplicação para automatização de agendamento de projetos e alocação de pessoal

Formiga - MG

2019

Caio Marcelo Pereira Melo

Aplicação para automatização de agendamento de projetos e alocação de pessoal

Monografia do trabalho de conclusão de curso apresentado ao Instituto Federal Minas Gerais - Campus Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

Campus Formiga

Ciência da Computação

Orientador: Diego Mello da Silva

Formiga - MG

2019

Melo, Caio Marcelo Pereira

M528a Aplicação para automatização de agendamento de projetos e
alocação de pessoal / Caio Marcelo Pereira Melo -- Formiga : IFMG,
2019.

78p. : il.

Orientador: Prof. MSc. Diego Mello da Silva

Trabalho de Conclusão de Curso – Instituto Federal de Educação,
Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Agendamento. 2. Alocação. 3. Otimização. 4. Multi-Objetivo.
5. RCPSP. I. Silva, Diego Mello da. II. Título.

CDD 004

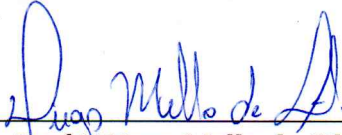
CAIO MARCELO PEREIRA MELO


**APLICAÇÃO PARA AUTOMATIZAÇÃO DE AGENDAMENTO DE
PROJETOS E ALOCAÇÃO DE PESSOAL**


Trabalho de Conclusão de Curso apresentado ao
Instituto Federal de Minas Gerais-Campus
Formiga, como Requisito parcial para obtenção do
título de Bacharel em Ciência da Computação.

Aprovado em: 28 de Novembro de 2019.

BANCA EXAMINADORA


Prof.º Diego Mello da Silva


Prof.º Wallace de Almeida Rodrigues


Prof.º Adriano Olímpio Tonelli


Prof.º Mário Luiz Rodrigues Oliveira

Agradecimentos

Primeiramente, gostaria de agradecer ao meu pai, Orlando, à minha mãe, Luzia, e à minha irmã, Caroline, por terem me dado o suporte necessário para que eu pudesse concluir essa etapa da minha vida, pela paciência e compreensão que tiveram comigo.

Agradeço também ao Diego, por ter aceito o convite para me orientar nesse trabalho, pela paciência que teve e pelos conhecimentos que me passou.

Também não poderia me esquecer de todos os meus amigos que estavam sempre ao meu lado nos momentos mais difíceis e nos mais agradáveis que tive que enfrentar durante esses anos.

Por fim deixo meus mais sinceros agradecimentos à comunidade do IFMG, por fornecerem os meios necessários para existência desse trabalho.

*“Tente me ensinar das tuas coisas
Que a vida é séria e a guerra é dura
Mas se não puder, cale essa boca, Pedro
E deixa eu viver minha loucura.”
(Raul Seixas)*

Resumo

Este trabalho apresenta o desenvolvimento do protótipo de uma aplicação que auxilie gestores de projeto na alocação de pessoal e agendamento de tarefas. Disponibilizando meios de quantificação da personalidade e nível de proficiência em determinadas habilidades a ferramenta permite que sejam encontradas soluções que melhorem o alinhamento dos membros de uma equipe com as tarefas que estão desenvolvendo. Além disso foram considerados aspectos relacionados ao tempo de desenvolvimento, custo e utilização da equipe alocada. Tratando-se de um problema de otimização de agendamento, o projeto consiste de uma adaptação do RCPSP (*Resource Constrained Project Scheduling*) multimodo, onde cada modo é composto pelo conjunto de desenvolvedores aptos à realizar uma tarefa, e multi-objetivo. O otimizador foi construído em base em um algoritmo genético multi-objetivo, o NSGA-II, e permite que o usuário final escolha dinamicamente quais aspectos deseja otimizar. A validação da aplicação foi realizada por meio de testes de convergência, garantindo que ao passar do tempo as soluções obtidas irão melhorar, escolha de parâmetros, visando otimizar a execução do projeto sem afetar a qualidade das soluções obtidas, análise dos resultados obtidos e impacto das funções objetivo. Os resultados obtidos mostraram que aplicação atingiu resultados satisfatórios para instâncias fictícias e construiu uma base para o desenvolvimento de uma aplicação que possa ser utilizada em ambientes reais, deixando para trabalhos futuros melhores tratamentos de aspectos que retratem a realidade.

Palavras-chave: Agendamento, Alocação, Otimização, Multi-Objetivo, RCPSP.

Abstract

This work presents the development of a prototype for an application that helps project managers to staff and schedule their projects. Taking means to measure the personality and proficiency in some skills this tool allows to find good solutions for the alignment of the staff and the tasks being developed. Besides that aspects related to makespan, cost, and staff use have been considered. As it is a scheduling problem, this work is an adaptation of the RCPS (Resource Constrained Project Scheduling) multi-mode, whereas each mode corresponds to a set of developers capable of working in a task, and multi-objective. The optimizer has been built based on the NSGA-II algorithm and allows that the final user chooses which objective functions to use in the evaluation of the solutions. The validation of the prototype was done with convergency tests, assuring that the solutions are getting better over time, parameter choosing, on an attempt to minimize the execution time without getting bad solutions, analysis of the solutions obtained and impact of the objective functions over each other. The results show that the application gets good solutions for fictional datasets and that a base for the development of a more robust application has been built.

Keywords: Scheduling, Staffing, Optimization, Multi-Objective, RCPSP.

Lista de ilustrações

Figura 1 – NSGA-II (DEB et al., 2002)	19
Figura 2 – Diagrama de Classes	25
Figura 3 – Cruzamento dos genes	29
Figura 4 – Mutação de um gene	30
Figura 5 – Exemplo de chamada da aplicação	31
Figura 6 – <i>Hypervolume</i> Projeto J102_2 e Equipe TEAM01	32
Figura 7 – <i>Hypervolume</i> Projeto J102_2 e Equipe TEAM02	33
Figura 8 – <i>Hypervolume</i> Projeto J125_9 e Equipe TEAM01	33
Figura 9 – <i>Hypervolume</i> Projeto J125_9 e Equipe TEAM02	34
Figura 10 – <i>Hypervolume</i> Projeto M12_8 e Equipe TEAM01	34
Figura 11 – <i>Hypervolume</i> Projeto M12_8 e Equipe TEAM02	35
Figura 12 – <i>Boxplot</i> Projeto J102_2 e Equipe TEAM01	37
Figura 13 – <i>Radviz</i> Projeto J102_2 e Equipe TEAM01	38
Figura 14 – <i>Boxplot</i> Projeto J102_2 e Equipe TEAM02	39
Figura 15 – <i>Radviz</i> Projeto J102_2 e Equipe TEAM02	39
Figura 16 – <i>Boxplot</i> Projeto J125_9 e Equipe TEAM01	40
Figura 17 – <i>Radviz</i> Projeto J125_9 e Equipe TEAM01	40
Figura 18 – <i>Boxplot</i> Projeto J125_9 e Equipe TEAM02	41
Figura 19 – <i>Radviz</i> Projeto J125_9 e Equipe TEAM02	42
Figura 20 – <i>Boxplot</i> Projeto M12_8 e Equipe TEAM01	42
Figura 21 – <i>Radviz</i> Projeto M12_8 e Equipe TEAM01	43
Figura 22 – <i>Boxplot</i> Projeto M12_8 e Equipe TEAM02	43
Figura 23 – <i>Radviz</i> Projeto M12_8 e Equipe TEAM02	44
Figura 24 – Sequenciamento do resultado 1	45
Figura 25 – Ocupações do resultado 1	45
Figura 26 – Alocações do resultado 1	45
Figura 27 – Sequenciamento do resultado 2	45
Figura 28 – Ocupações do resultado 2	45
Figura 29 – Alocações do resultado 2	46
Figura 30 – Sequenciamento do resultado 3	46
Figura 31 – Ocupações do resultado 3	46
Figura 32 – Alocações do resultado 3	46
Figura 33 – Correlações para o Projeto J102_2 e Equipe TEAM01	49
Figura 34 – Correlações para o Projeto J102_2 e Equipe TEAM02	50

Lista de tabelas

Tabela 1 – Problemas que ocorrem com mais frequência nos projetos da Organização. Fonte: PMI	13
Tabela 2 – Atributos da classe <i>Chromosome</i>	26
Tabela 3 – Métodos da classe <i>Chromosome</i>	26
Tabela 4 – Atributos da classe <i>ChromosomeFactory</i>	27
Tabela 5 – Métodos da classe <i>ChromosomeFactory</i>	27
Tabela 6 – Argumentos para o método <i>nsgaii</i>	27
Tabela 7 – Atributos da classe <i>Task</i>	28
Tabela 8 – Atributos da classe <i>Developer</i>	28
Tabela 9 – Funções objetivo	30
Tabela 10 – Parâmetros disponíveis da interface.	31
Tabela 11 – Resultados Hypervolume	36
Tabela 12 – Tabela de resultados Projeto J102_2 e Equipe TEAM01	37
Tabela 13 – Tabela de resultados Projeto J102_2 e Equipe TEAM02	38
Tabela 14 – Tabela de resultados Projeto J125_9 e Equipe TEAM01	40
Tabela 15 – Tabela de resultados Projeto J125_9 e Equipe TEAM02	41
Tabela 16 – Tabela de resultados Projeto M12_8 e Equipe TEAM01	42
Tabela 17 – Tabela de resultados Projeto M12_8 e Equipe TEAM02	43
Tabela 18 – Resultados individuais	45
Tabela 19 – Correlações entre funções objetivo	48

Sumário

1	INTRODUÇÃO	12
1.1	Justificativa	12
1.2	Objetivos	14
1.2.1	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Otimização Mono-Objetivo <i>versus</i> Multiobjetivo	15
2.2	Algoritmos Genéticos e NSGA-II	16
2.3	Problemas de Escalonamento de Projetos com Restrição de Recursos	19
2.3.1	<i>Multi Mode RCPSP</i>	19
2.3.2	<i>Multi Objective RCPSP</i>	20
2.4	<i>Five Factor Model</i>	20
2.5	Trabalhos Relacionados	21
3	MATERIAIS E MÉTODOS	22
3.1	Materiais	22
3.1.1	Python	22
3.1.2	<i>Hardware</i>	22
3.1.3	Instâncias	23
3.2	Metodologia	23
4	DESENVOLVIMENTO	25
4.1	Arquitetura da solução	25
4.2	NSGA-II	26
4.3	Problema	27
4.3.1	Elementos básicos	27
4.3.2	Gene	28
4.3.3	<i>Factory</i>	28
4.3.4	Funções Objetivo	30
4.4	<i>Command Line Interface</i>	30
5	RESULTADOS	32
5.1	Convergência	32
5.2	Seleção de parâmetros	35
5.3	Variação dos resultados	36
5.4	Agendamentos e uso de recursos	44

5.5	Funções objetivo	47
6	CONCLUSÕES E TRABALHOS FUTUROS	51
	APÊNDICE A – PROJETOS	53
	APÊNDICE B – EQUIPES	63
	REFERÊNCIAS	76

1 Introdução

De acordo com a norma ISO 10006 - Diretrizes para a Qualidade no Gerenciamento de Projetos (ISO, 2000) um projeto é um “processo que consiste de um grupo de atividades coordenadas e controladas com datas para início e término, visando alcançar um determinado objetivo com limites de tempo, recursos e custo”. Para efetivamente lidar com o desafio de atingir o objetivo de um projeto dentro dos limites de prazo e custo é essencial ter mão de obra bem alocada. De acordo com Carbno (1999), um gestor deve ponderar diversos fatores quando decide a alocação de equipe em projetos de TI, como urgência no projeto, habilidades requeridas, competência e disponibilidade da equipe, e impacto de nivelamento de recursos no cronograma de entregas.

Isto posto, o presente trabalho descreve o desenvolvimento de um protótipo de aplicação computacional que busca otimizar o agendamento e alocação de pessoal levando em consideração múltiplos critérios como minimização de custo, minimização do prazo do projeto e maximização da satisfação da equipe, dentre outros. As próximas subseções do documento irão apresentar a justificativa para o trabalho, o objetivo geral e específicos. No capítulo 2 será apresentada uma breve revisão de literatura sobre o tema, abordando técnicas e problemas correlatos em termos de agendamento de tarefas e alocação de equipe. No capítulo 3 serão apresentados os materiais utilizados e metodologia de desenvolvimento em fases. O capítulo 4 trará detalhes sobre a implementação realizada, em termos de meta-heurística escolhida e aplicação. No capítulo 5 serão reportados os resultados experimentais realizados, finalizando o trabalho com algumas considerações finais e propostas de trabalhos futuros no capítulo 6.

1.1 Justificativa

Ao planejar um projeto é desejável que os recursos disponíveis sejam distribuídos da melhor maneira possível. De acordo com Vargas (2005) um projeto pode fracassar por diversos motivos e dentre esses estão a falta de capacitação das pessoas envolvidas nas tarefas e a falta de conhecimento sobre as necessidades de pessoal, equipamentos e materiais. O *Chaos Report* (STANDISH, 2014) é um estudo realizado pelo *Standish Group* no *CHAOS Research Project* que analisa a quase um quarto de século boas práticas e taxas de sucesso em projetos de TI nos Estados Unidos, e é tratado como autoridade *de facto* quando se analisa taxas de sucesso de projetos neste segmento, onde um dos aspectos mais importantes desta pesquisa está em descobrir porque projetos falham. Segundo o relatório de 2014, no levantamento feito com 365 empresas respondentes do setor, 7.2% consideraram que a competência da equipe é um fator chave de sucesso, 6.4% considera-

ram que falta de recursos podem comprometer o sucesso de um projeto, e 10.6% acreditam que a falta de recursos podem prejudicar e eventualmente cancelar projetos em execução. No *Chaos Report* de 2015 (STANDISH, 2015) página 10, o *Standish Group* declara que projetos de sucesso necessitam de pessoas espertas e bem treinadas, de forma que um dos fatores chave para o sucesso de projetos é possuir uma equipe competente que, na visão do *Standish Group*, deve ter cinco fundamentos chave para garantir equipe competente: (i) identificar as competências requeridas e habilidades alternativas; (ii) prover um bom programa de treinamento contínuo para melhorar as habilidades da equipe; (iii) recrutamento interno e externo para prover balanço de experiências; (iv) oferecer incentivos para motivar a equipe, e (v) garantir que a equipe esteja focada no projeto. A Tabela 1 apresenta os resultados de uma pesquisa ¹ que é feita pelo PMI (Project Management Institute) ² apresentando problemas que ocorrem nos projetos que são realizados por empresas.

Problema	Frequência em que é citado pelas empresas
Não cumprimento dos prazos	60.2%
Mudanças de escopo constantes	43%
Problemas de comunicação	40.1%
Escopo não definido adequadamente	39.5%
Não cumprimento do orçamento	28.3%
Recursos humanos insuficientes	28.3%
Concorrência entre o dia-a-dia e o projeto na utilização dos recursos	27.6%
Riscos não avaliados corretamente	22.9%
Mudanças de prioridade constantes ou falta de prioridade	19.8%
Problemas com fornecedores	17.7%
Estimativas incorretas ou sem fundamento	15.6%
Retrabalho em função da falta de qualidade do produto	11.7%
Falta de definição de responsabilidades	10.2%
Falta de uma metodologia de apoio	7.5%
Falta de apoio da alta administração/sponsor (patrocinador)	7.3%
Falta de competência para gerenciar projetos	6.9%
Falta de uma ferramenta de apoio	6.7%
Falta de conhecimento técnico sobre a área de negócio da organização	2.1%

Tabela 1 – Problemas que ocorrem com mais frequência nos projetos da Organização.
Fonte: PMI

Portanto, uma ferramenta que auxilie no agendamento de tarefas de um projeto com boa alocação dos recursos humanos (equipe do projeto) pode colaborar positivamente para a redução do riscos de falha em projetos de TI (e potencialmente outros tipos de projetos). Isto posto, este trabalho justifica-se por apresentar o desenvolvimento de um protótipo de ferramenta de alocação automática de tarefas e recursos humanos para uso em organizações orientadas a projetos que queiram ser mais eficientes no uso de seus recursos. Além disso, também é válido como estudo investigativo de algoritmos de otimização que computem rapidamente resultados satisfatórios no contexto de programação de tarefas de projetos com alocação de recursos, sendo base para futuros trabalhos nesta linha.

¹ http://www.mp.go.gov.br/portalweb/hp/33/docs/benchmarking_gp_2010_geral.pdf

² <https://brasil.pmi.org/brazil/AboutUS/WhatIsPMI.aspx>

1.2 Objetivos

Este trabalho tem por objetivo construir o protótipo de uma aplicação que auxilie no agendamento de tarefas e na alocação dos recursos humanos para a realização de projetos, otimizando indicadores de interesse por meio de meta-heurística.

1.2.1 Objetivos Específicos

- Estudar problemas de otimização correlacionados com a proposta deste projeto na literatura especializada;
- Estudar modelos matemáticos existentes relacionados ao problema deste trabalho.
- Analisar métodos computacionais que foram, segundo a literatura especializada, utilizados para resolução dos modelos, com interesse especial em heurísticas e meta-heurísticas;
- Implementar o protótipo da aplicação propriamente dito;
- Levantar dados para a realização de testes em ambiente real de empresas que trabalham com projetos ou através de instâncias artificiais encontradas na literatura especializada;
- Realizar experimentos, coletar, analisar e reportar os resultados da aplicação para a análise de consistência e desempenho nas soluções calculadas.

2 Fundamentação Teórica

Neste capítulo serão abordados os aspectos teóricos sobre alguns temas tratados neste trabalho, em especial a diferenciação sobre otimização mono e multiobjetivo (ou multicritério); sobre a meta-heurística Algoritmos Genéticos em sua versão canônica e multicritério; sobre os problemas combinatórios relacionados à escalonamento de projetos com restrição de recursos e variantes, e trabalhos relacionados existentes na literatura especializada.

2.1 Otimização Mono-Objetivo *versus* Multiobjetivo

Segundo [Becceneri \(2008\)](#), problemas de otimização consistem em achar a melhor combinação dentre um conjunto de variáveis para maximizar ou minimizar uma função, geralmente chamada de função objetivo ou função custo. Tais problemas podem ser divididos em três categorias: aqueles cujas variáveis assumem valores reais (ou contínuos), aqueles cujas variáveis assumem valores discretos (ou inteiros) e aqueles em que há variáveis inteiras e contínuas, classificados, respectivamente, como problemas de Otimização Contínua, Otimização Combinatória/Discreta, e Otimização Mista.

Além dessa categorização em termos do domínio das variáveis do problema, um problema de otimização pode lidar com a maximização ou minimização de um único critério (quando é mono-objetivo), ou de vários critérios (quando é multiobjetivo). É preciso notar que existem diferenças essenciais em problemas de otimização mono-objetivo e multiobjetivo. [Deb \(2001\)](#) esclarece essa diferença dando como exemplo um problema bi-objetivo: ao decidir qual carro comprar, o cliente deseja fazer a escolha considerando como objeto da compra aquele que oferece o maior nível de conforto juntamente com o menor preço possível. Ao se tratar de uma escolha com duas funções objetivo conflitantes/concorrentes, para cada uma das funções irá existir uma escolha de veículo que é globalmente ótima para um objetivo mas não será para os demais. Por consequência não será possível determinar uma única solução para o problema, visto que existe uma escolha ótima para cada objetivo analisado que atende apenas este critério.

Para o problema anteriormente mencionado, se o comprador desejar um maior nível de conforto então ele terá que investir mais em sua compra, deixando de lado o desejo de gastar pouco com um carro confortável. Para um problema multi-objetivo é dito que existe um conjunto de soluções ótimas em que qualquer solução desse conjunto não pode ser dita como melhor do que qualquer outra pertencente ao mesmo grupo, visto que são globalmente ótimas para seus respectivos objetivos, esse conjunto caracteriza o que é conhecido como fronteira de Pareto. No seu trabalho, [Deb \(2001\)](#) ainda esclarece

que problemas que lidam com uma única função de interesse são, na verdade, um caso degenerado dos problemas multi-objetivo.

2.2 Algoritmos Genéticos e NSGA-II

O problema tratado nesse trabalho é uma variação do RCPSP (*Resource Constrained Project Scheduling*) (MERKLE; MIDDENDORF; SCHMECK, 2002). Considerando que o RCPSP trata-se de um problema NP-Difícil (CORMEN et al., 2009), e que o problema tratado neste TCC é uma variante do RCPSP, a construção de um algoritmo exato se torna inviável para resolver instâncias de tamanho real, aplicando-se apenas à instâncias de pequeno porte. Dessa forma foi necessária a busca de uma meta-heurística para a implementação do trabalho. Por se tratar de um problema multi-objetivo decidiu-se pela utilização do NSGA-II (DEB et al., 2002), por ser um dos métodos de otimização multi-objetivo mais utilizados na atualidade (em busca rápida feita no site <scholar.google.com> obtivemos como resultado aproximadamente 49.400 ocorrências do termo).

De acordo com Whitley (1994) um algoritmo genético é qualquer modelo baseado em população que utiliza seleção e operadores de recombinação para gerar novos pontos em um determinado espaço de busca. Em seu trabalho Whitley (1994) apresenta o algoritmo genético canônico cujo pseudocódigo é apresentado a seguir, no Algoritmo 1. O primeiro passo consiste em gerar uma população inicial com N indivíduos, cada um codificando uma solução para o problema. Em seguida, até que um critério de parada seja satisfeito, a população atual é avaliada de acordo com uma função de adaptabilidade ou *fitness*. Em seguida aplica-se o operador de seleção de indivíduos que seleciona os indivíduos parentais I que serão recombinados para gerar nova população. A próxima geração é construída a partir de recombinação de I e, por fim alguns indivíduos sofrem uma mutação de acordo com uma certa probabilidade, para garantir variabilidade de soluções na população.

Algorithm 1 ALGORITMO-GENETICO

```

1:  $P \leftarrow \text{populacaoInicial}$ 
2: while  $\neg \text{critérioDeParada}$  do
3:    $\text{avaliarIndividuos}(P)$ 
4:    $I \leftarrow \text{selecao}(P)$ 
5:    $P \leftarrow \text{recombinacao}(I)$ 
6:    $\text{mutacao}(P)$ 

```

Existem diversos algoritmos populacionais baseados no modelo descrito pelo Algoritmo 1. Algumas dessas variações têm foco em problemas multi-objetivo e são conhecidas como *Multi-Objective Evolutionary Algorithms* (MOEAs) (DEB, 2001). Nesse trabalho foi utilizado um MOEA definido em (DEB et al., 2002) nomeado como *Nondominated Sorting Genetic Algorithm II* (NSGA-II), que consiste basicamente em uma melhoria de seu

antecessor, o NSGA. Sendo P a população inicial, I o conjunto de indivíduos selecionados para reprodução, Q o conjunto de filhos gerados a partir de I , R um conjunto contendo todos indivíduos da população atual e a prole gerada, e N o número de indivíduos de cada população, o NSGA-II é definido pelo Algoritmo 2

Algorithm 2 NSGA-II

```

1:  $P \leftarrow \text{populacaoInicial}$ 
2: while  $\neg \text{condicaoDeParada}$  do
3:    $\text{avaliarIndividuos}(P)$ 
4:    $I \leftarrow \text{selecao}(P)$ 
5:    $Q \leftarrow \text{recombinacao}(I)$ 
6:    $\text{mutacao}(Q)$ 
7:    $R \leftarrow P \cup Q$ 
8:    $F \leftarrow \text{FAST\_NON\_DOMINATED\_SORT}(R)$ 
9:    $P \leftarrow \emptyset, i \leftarrow 1$ 
10:  while  $|F_i| + |P| \leq N$  do
11:     $\text{CROWDING\_DISTANCE\_ASSIGNMENT}(F_i)$ 
12:     $P \leftarrow P \cup F_i$ 
13:     $i \leftarrow i + 1$ 
14:   $k \leftarrow N - |P|$ 
15:  if  $k > 0$  then
16:     $\text{CROWDING\_DISTANCE\_ASSIGNMENT}(F_i)$ 
17:     $\text{ordenar}(F_i)$ 
18:     $P \leftarrow P \cup \{F_{i1}, \dots, F_{ik}\}$ 

```

Como é possível notar, o NSGA-II trata a geração de novas populações de maneira diferente do algoritmo genético canônico. No Algoritmo 2 após a recombinação e mutação os indivíduos gerados são colocados num novo conjunto R juntamente com a população anterior e então é aplicado o Algoritmo 3 nesse novo conjunto para a ordenação das fronteiras de não dominância. O conjunto R é então dividido, as melhores fronteiras são colocadas por inteiro na nova população e caso o tamanho dessa seja menor que um parâmetro N uma última fronteira é ordena utilizando um valor definido por Deb et al. (2002) chamado de *crowding_distance* e os indivíduos restantes da nova população são retirados dessa última fronteira, vide Figura 1.

O Algoritmo 3 é utilizado para separar e ordenar as fronteiras de não dominância, sendo que a primeira dessas é a fronteira de Pareto. Na primeira parte, linhas 2 à 11, para cada indivíduo p da população são atribuídas as soluções dominadas S_p e a quantidade de soluções dominantes n_p , caso não existam soluções dominantes p é inserido na fronteira de Pareto F_1 . No restante do processo as novas fronteiras são adicionadas ao conjunto F .

Algorithm 3 FAST_NON_DOMINATED_SORT (DEB et al., 2002)

```

1: for all  $p \in P$  do
2:    $S_p \leftarrow \emptyset$ 
3:    $n_p \leftarrow 0$ 
4:   for all  $q \in P$  do
5:     if  $p \prec q$  then
6:        $S_p \leftarrow S_p \cup \{q\}$ 
7:     else if  $q \prec p$  then
8:        $n_p \leftarrow n_p + 1$ 
9:   if  $n_p = 0$  then
10:     $p_{rank} \leftarrow 1$ 
11:     $F_1 \leftarrow F_1 \cup \{p\}$ 
12:  $i \leftarrow 1$ 
13: while  $F_i \neq \emptyset$  do
14:    $Q \leftarrow \emptyset$ 
15:   for all  $p \in F_i$  do
16:     for all  $q \in S_p$  do
17:        $n_q \leftarrow n_q - 1$ 
18:       if  $n_q = 0$  then
19:          $q_{rank} \leftarrow i + 1$ 
20:          $Q \leftarrow Q \cup \{q\}$ 
21:    $i \leftarrow i + 1$ 
22:    $F_i \leftarrow Q$ 

```

Ao tratar indivíduos de uma mesma fronteira não é possível utilizar os valores das funções objetivo para ordenação, para isso Deb et al. (2002) utiliza o *crowding_distance*, que é basicamente uma estimativa de densidade para cada indivíduo (quanto maior menos densa). A partir desse valor também foi definido um operador de comparação que utiliza o *rank* da fronteira de não dominância e o *crowding_distance* de cada solução, dando preferência aos menores valores de *rank* e maiores valores de *crowding_distance*.

Algorithm 4 CROWDING_DISTANCE_ASSIGNMENT (DEB et al., 2002)

```

1:  $l \leftarrow |I|$ 
2: for all  $p \in I$  do
3:    $p_{distance} \leftarrow 0$ 
4: for all  $m \in M$  do
5:    $order_nar(I, m)$ 
6:    $I_{1,distance} \leftarrow I_{l,distance} \leftarrow \infty$ 
7:   for  $i \leftarrow 2, l - 1$  do
8:      $I_{i,distance} \leftarrow I_{i,distance} + (I_{i+1,m} - I_{i-1,m}) / (f_m^{max} - f_m^{min})$ 

```

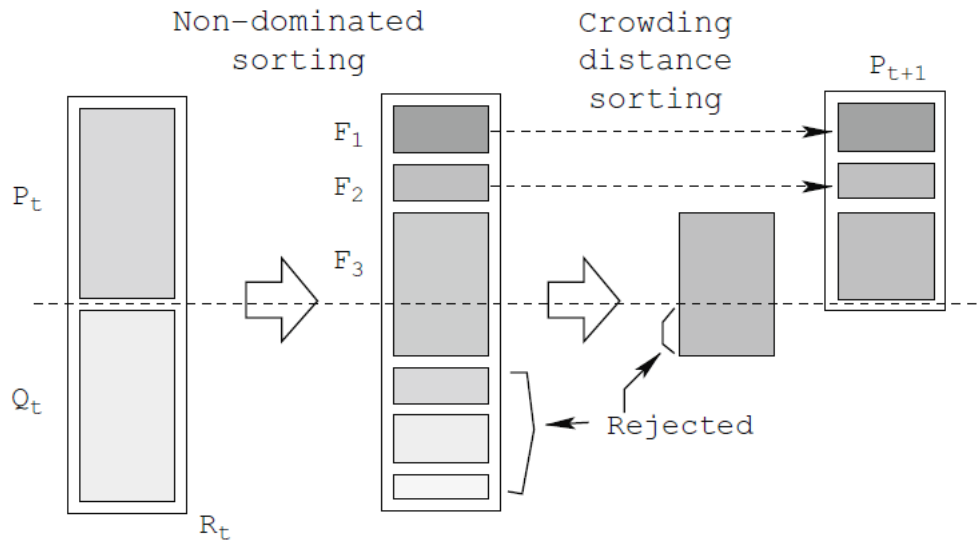


Figura 1 – NSGA-II (DEB et al., 2002)

2.3 Problemas de Escalonamento de Projetos com Restrição de Recursos

Merkle, Middendorf e Schmeck (2002) definem RCPSP (*Resource Constrained Optimization Problem*) como o problema de escalonar as tarefas de um determinado projeto de maneira que o *makespan* seja minimizado respeitando as restrições de precedência entre as atividades e que a capacidade dos recursos não seja excedida em um determinado intervalo de tempo. Além de ser um problema de pesquisa interessante por se tratar de uma forma generalizada de diversos problemas de escalonamento como o *Job Shop Scheduling Problem* também possui aplicações em diversas indústrias de engenharia e desenvolvimento de software (BRUCKER et al., 1999).

É possível alterar alguns aspectos do RCPSP à fim de se obter formas mais específicas como explorado por Hartmann e Briskorn (2010). No artigo são modificados conceitos relacionados às atividades, restrições temporais e de recursos, funções objetivo, e problemas multi-projeto. A seguir serão apresentadas algumas variações interessantes para a compreensão do problema tratado nesse trabalho.

2.3.1 Multi Mode RCPSP

De acordo com Hartmann e Briskorn (2010) RCPSP padrão assume que as atividades serão executadas de uma única maneira, com duração e requisitos de recursos fixos. No MRCPS (Multi-mode RCPSP) é permitido que uma tarefa possa ser executada de diversos modos, sendo cada modo uma maneira de combinar requisitos de duração e recursos que permitem que uma tarefa possa ser concluída.

Hartmann e Briskorn (2010) diz que o problema tratado por Bellenguez e Néron (2004) pode ser considerado como um caso especial do MRCPSP em que cada modo corresponde a um subconjunto factível de membros de uma equipe que podem executar uma tarefa.

2.3.2 *Multi Objective* RCPSP

Outra alteração possível no RCPSP é a utilização de múltiplas funções objetivo como explorado em (HAPKE; JASZKIEWICZ; SŁOWIŃSKI, 1998). Duas estratégias que podem ser utilizadas para abordar esse problema são: condensar as funções objetivos por meio de soma ponderada em uma única função objetivo e a geração de fronteiras de otimalidade de Pareto (HARTMANN; BRISKORN, 2010).

2.4 *Five Factor Model*

Devido a natureza da aplicação desenvolvida nesse trabalho, deu-se a necessidade de escolher aspectos sobre a alocação da equipe que pudessem ser otimizados. Uma das medidas escolhidas foi o nível de adequação dos funcionários às tarefas, de onde surgiu a necessidade de uma maneira de quantificar essa medida. Stylianou e Andreou (2012) utiliza em seu trabalho uma medida de personalidade estudada por Tupes e Christal (1992) conhecida como FFM (*Five Factors Model*), que divide a personalidade de uma pessoa em cinco fatores, sendo eles: estabilidade emocional, extroversão, agradabilidade, conscienciosidade e abertura à experiências. Cada um desses fatores está relacionado a algumas características como:

estabilidade emocional: Placidez, auto-suficiência e equilíbrio emocional;

extroversão: Franqueza, assertividade, sociabilidade, alegria;

agradabilidade: Maturidade emocional, brandura, cooperatividade, veracidade, adaptabilidade e auto-suficiência;

conscienciosidade: Responsabilidade, perseverança e convencionalidade;

abertura à experiências: Esteticamente metuculoso, imaginativo socialmente polido e de mente independente.

Para que esses traços de personalidade pudessem ser utilizados como medida durante o desenvolvimento da aplicação foi convencionado que para cada uma das características seriam atribuídos valores entre $[0, 1]$, indicando o quanto um determinado traço é perceptível.

2.5 Trabalhos Relacionados

Alguns artigos presentes na literatura abordam problemas relacionados ao deste trabalho e serviram também como guia para a construção tanto do problema quanto da abordagem utilizada.

Wu e Sun (2006) propôs um modelo de agendamento e alocação de pessoal levando em consideração o efeito da aprendizagem, com o objetivo de minimizar o custo com contratações externas. No modelo proposto é considerado que uma tarefa possui um intervalo disponível para o início e um outro intervalo para a finalização e outro em que a tarefa pode ser executada. Para os funcionários contratados externamente são considerados a quantidade de tempo que o funcionário estará disponível e uma constante indicando a eficiência do recurso, enquanto para os funcionários da equipe é considerado somente a eficiência sendo tratada como uma variável, já que a mesma será alterada pelo fator de aprendizagem ao longo do tempo.

Nos testes realizados os autores utilizaram um algoritmo genético, sendo que cada indivíduo da população é representado por uma matriz $M_{n,m}$ onde n é o numero de períodos de execução e m o número de desenvolvedores disponíveis. Cada posição nessa tabela indica qual tarefa um certo empregado estará realizando durante um período.

Stylianou e Andreou (2012) utiliza uma abordagem multi-objetivo levando em consideração a personalidade dos membros envolvidos no projeto, argumentando que os gerentes de projeto normalmente levam em conta somente os aspectos técnicos relativos aos funcionários, o que pode ser prejudicial ao desenvolvimento do projeto.

Para representar a solução os autores decidiram utilizar uma matriz de bits, semelhante ao utilizado por Wu e Sun (2006), porém cada linha da tabela representa uma tarefa e cada posição da tabela possui um valor binário indicando se o desenvolvedor irá participar daquela tarefa ou não. As funções objetivo utilizadas estão relacionadas aos traços de personalidade, ao nível de habilidade dos empregados e ao tamanho da equipe utilizada.

O trabalho de Kolisch e Heimerl (2012) propõe uma meta-heurística para o problema de agendamento e alocação em projetos de TI baseada em um grafo de fluxo de custo mínimo. O algoritmo proposto consiste de duas fases, na primeira utilizaram um algoritmo genético e na segunda o resultado obtido foi submetido a uma busca TABU e utiliza um grafo de fluxo de custo mínimo para representar o problema de alocação de pessoal.

3 Materiais e Métodos

Neste capítulo serão apresentados os materiais e métodos que foram utilizados durante o desenvolvimento do projeto. Especificando linguagem e bibliotecas utilizadas, além das especificações do hardware utilizado.

3.1 Materiais

3.1.1 Python

Nas palavras de [Borges \(2014\)](#) "O Python tem uma sintaxe clara e concisa que favorece a legibilidade do código-fonte, tornando a linguagem mais produtiva", além de incluir diversas estruturas como tuplas, dicionários, listas e ser multi-paradigma suportando programação funcional, modular e orientada a objetos. Sendo essas algumas das razões pela escolha da linguagem.

Existem alguns interpretadores para a linguagem como CPython ¹, o interpretador padrão, IronPython ², implementação da linguagem na plataforma .NET, PyPy ³, baseado na tecnologia RPython ⁴, e Jython ⁵, uma implementação que permite a utilização de Python na JVM. O interpretador utilizado nesse trabalho foi o CPython em sua versão 3.7.

Para facilitar a implementação de alguns aspectos do sistema proposto foram utilizadas algumas bibliotecas de terceiros. O Numpy ⁶, biblioteca utilizada para processamento numérico de alto desempenho, fornece algumas estruturas como vetores N dimensionais, matrizes e arranjos. Essa biblioteca também disponibiliza funções básicas de álgebra linear, estatística dentre outros. Também foi necessária a utilização de bibliotecas para a plotagem de gráficos que permitissem uma melhor análise dos resultados obtidos, o Matplotlib⁷ e o Pandas⁸.

3.1.2 Hardware

O computador utilizado tanto no desenvolvimento quanto nos testes foi um *notebook* Samsung X40, com processador Core i5-8250U 1.6GHz, 8Gb de RAM, placa de vídeo

¹ <https://www.python.org>

² <https://github.com/IronLanguages/ironpython2>

³ <https://www.pypy.org>

⁴ <https://rpython.readthedocs.io/en/latest>

⁵ <https://www.jython.org>

⁶ <https://numpy.org>

⁷ <https://matplotlib.org>

⁸ <https://pandas.pydata.org>

Geforce MX110, 1TB de HD. O sistema operacional utilizado foi o MANJARO sobre o kernel linux 5.2.21.

3.1.3 Instâncias

As instâncias do problema abordado nesse trabalho são compostas pela descrição de um projeto, e de uma equipe. Por não existirem instâncias prontas com todas as informações necessárias foi preciso adaptar algumas já existentes. Dessa forma os projetos foram construídos a partir de algumas instâncias do PSPLIB ⁹ (KOLISCH; SPRECHER, 1997). Segundo o site <www.om-db.wi.tum.de/psplib/main.html> o PSPLIB contém diferentes conjuntos de problemas para vários tipos de problema de agendamento de projetos com restrição de recursos bem como soluções ótimas calculadas por modelos exatos e aproximações calculadas por heurísticas, para os problemas conhecidos por RCPSP, RCPSP/max, MRCPSP, MRCPSP/max, RP/max, dentre outros. Especificamente para o MRCPSP, que foi usado como inspiração para adaptação das instâncias usadas neste trabalho, utilizou-se variações dos arquivos contidos nos conjuntos j10 files, j12 files e m1 files, sendo necessário a adição de duração das tarefas, requisitos de habilidades, e traços de personalidade esperados, os quais foram gerados aleatoriamente. Os dados das equipes também foram obtidos a partir de geradores aleatórios e foram feitas duas equipes, uma pequena, na qual cada desenvolvedor possui um maior número de habilidades disponíveis e um preço alto, e uma grande, sendo que cada membro da equipe possui um pequeno número de habilidades e um preço mais baixo. Para maiores informações consultar o Apêndice A.

3.2 Metodologia

O início do desenvolvimento desse trabalho se deu com os estudos de problemas relacionados ao proposto e quais foram as abordagens utilizadas pelos autores. Essa pesquisa deu base para a definição das estratégias utilizadas para a construção da aplicação e quais critérios seriam considerados para otimização.

A próxima etapa foi o desenvolvimento da aplicação, definindo-se as divisões dos módulos que seriam utilizados pelo sistema e visando a construção de uma aplicação minimamente funcional e de fácil manutenção.

Para que fosse possível a realização de testes e de validações da aplicação desenvolvida, foram feitas adaptações de instâncias utilizadas em outros problemas de agendamento (vide Apêndices A e B). Ao final desse processo foram realizados testes para validar a convergência dos resultados que foram obtidos pela aplicação, para decidir quais

⁹ <http://www.om-db.wi.tum.de/psplib/main.html>

os melhores parâmetros para serem utilizados, para verificar o comportamento das funções objetivo escolhidas, e para visualizar a qualidade dos resultados obtidos.

4 Desenvolvimento

Nesse capítulo serão apresentadas e discutidas as abordagens utilizadas para o desenvolvimento do projeto como um todo. Passando pela fase desenvolvimento de uma biblioteca que generaliza o NSGA-II, pela construção de um motor de otimização construído sobre essa biblioteca e finalmente o desenvolvimento de ferramenta de linha de comando para facilitar o uso do motor.

4.1 Arquitetura da solução

Para que a construção da ferramenta proposta nesse trabalho fosse possível foram tomadas algumas medidas para a organização do projeto. A Imagem 2 apresenta a estrutura básica utilizada durante o projeto.

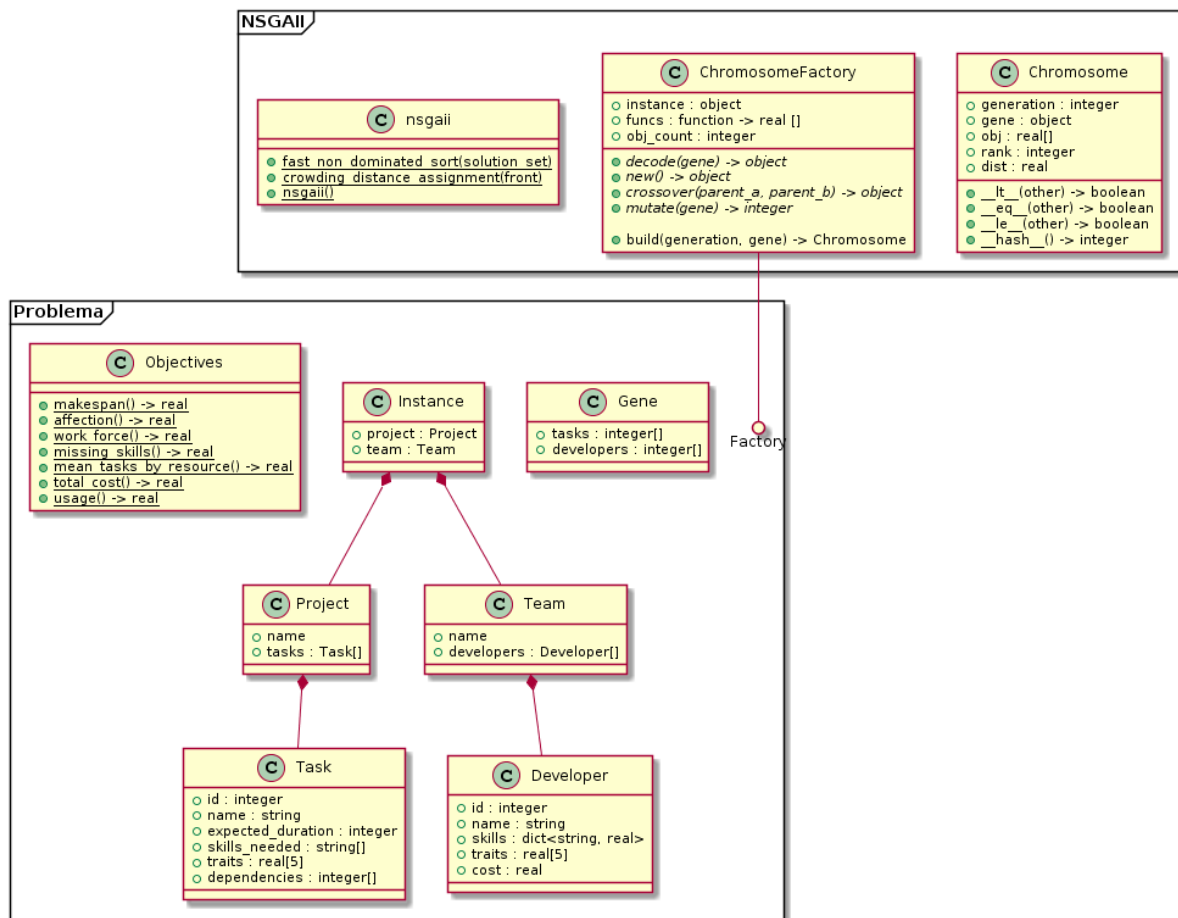


Figura 2 – Diagrama de Classes

A separação entre o desenvolvimento do NSGA-II e da implementação do pro-

blema proposto permitiu um maior grau de liberdade na experimentação de abordagens de estratégias para a construção do sistema. Os detalhes de cada um desses pacotes serão discutidos à seguir.

4.2 NSGA-II

Esse pacote foi desenvolvido com objetivo de melhorar a legibilidade do código, caso o leitor não se interesse por se aprofundar na meta-heurística utilizada ou na maneira como ela foi desenvolvida a compreensão dos detalhes de implementação do problema não será comprometida. A solução desenvolvida permite a definição generalizada de problemas multi-objetivo para o NSGA-II e foi essencial para realização de experimentações durante o desenvolvimento do projeto.

A classe *Chromosome* é responsável por armazenar informações sobre um indivíduo de uma população, além de fornecer os métodos de comparação utilizados pelo NSGA-II. Nessa classe foram utilizados métodos especiais que permitem a alteração de alguns operadores e funcionalidades do Python mostrados na Tabela 3.

O método `__hash__` é uma implementação do FNV-hash¹ utilizando os valores das funções objetivo do indivíduo no cálculo, dessa forma, em um conjunto não é possível que dois indivíduos sejam avaliados igualmente para todas as funções objetivo.

atributo	tipo	descrição
generation	inteiro	geração na qual o indivíduo nasceu
gene	objeto	objeto contendo a representação da solução em espaço de código
obj	reais	avaliação do <i>fitness</i> do indivíduo para cada uma das funções objetivo utilizadas
rank	inteiro	número da fronteira de não dominância à qual o indivíduo pertence
dist	real	medida de densidade da região do indivíduo

Tabela 2 – Atributos da classe *Chromosome*.

método	retorno	descrição
<code>__lt__</code>	boolean	operador Crowded Comparision. Substitui o operador <
<code>__eq__</code>	boolean	operador de igualdade entre dois <i>Chromosome</i> . Substitui o operador ==
<code>__le__</code>	boolean	operador de dominância. Substitui o operador <=
<code>__hash__</code>	integer	calculo de um identificador único para o indivíduo

Tabela 3 – Métodos da classe *Chromosome*.

Para que o NSGA-II pudesse ser implementado de maneira genérica foi necessária a criação de um tipo de interface que abstraísse os detalhes pertinentes ao problema que está sendo tratado. Para tal foi utilizado um padrão de projeto conhecido como *Abstract Factory* (GAMMA et al., 1995) na criação da classe *ChromosomeFactory*. A Tabela 4 apresenta os atributos armazenados, sendo que tais contém informações sobre o problema

¹ <http://www.isthe.com/chongo/tech/comp/fnv>

que está sendo tratado pelo usuário. A Tabela 5 apresenta os métodos da classe, sendo importante observar que *decode*, *new*, *crossover* e *mutate* são abstratos e é de responsabilidade do usuário retornar somente objetos do tipo escolhido para a representação das soluções em espaço de código.

atributo	tipo	descrição
instance	objeto	objeto contendo a instância que será utilizada
funcs	funções	lista com as funções objetivos que serão utilizadas para avaliar os indivíduos
obj_count	inteiro	número de funções objetivo

Tabela 4 – Atributos da classe *ChromosomeFactory*.

método	retorno	descrição
decode	objeto	transforma um objeto contendo a representação da solução em uma solução.
new	objeto	gera uma representação válida para a instancia que está sendo tratada.
crossover	objeto	realiza o cruzamento de dois Chromosome gerando um único gene
mutate		faz uma alteração em um determinado gene
build	Chromosome	cria um indivíduo válido à partir de um gene

Tabela 5 – Métodos da classe *ChromosomeFactory*.

De posse dessas classes é que se dá início a implementação dos métodos da classe *nsgaii*², sendo que *fast_non_dominated_sort* e *crowding_distance_assignment* auxiliares ao *nsgaii* e meras implementações do apresentado por Deb (2001).

argumento	tipo	descrição
factory	<i>ChromosomeFactory</i>	implementação de <i>ChromosomeFactory</i> para o problema que será tratado
selector	função	função para selecionar indivíduos para reprodução
population_size	inteiro	tamanho da população
mutation_probability	real	probabilidade de ocorrer mutação em um indivíduo
max_generations	inteiro	número de gerações que serão executadas

Tabela 6 – Argumentos para o método *nsgaii*

4.3 Problema

O outro pacote presente na arquitetura apresentada, Figura 2, é responsável pela implementação dos detalhes relacionados ao problema proposto nesse trabalho. Definindo classes para representar elementos como Tarefas e Desenvolvedores, funções objetivo, função de seleção, mutação, *crossover*, decodificação e geração de indivíduos aleatórios.

4.3.1 Elementos básicos

As instâncias utilizadas são compostas por projeto e time de desenvolvimento, sendo esses dois conjuntos de tarefas e desenvolvedores respectivamente. Essa separação

² A classe *nsgaii* da FIGURA 2 é meramente ilustrativa e só está presente na imagem para facilitar a compreensão da arquitetura do projeto.

foi feita para permitir a recombinação de funcionários e projetos sem que seja necessária a definição de novas instâncias. Tarefas são representadas na classe *Task* enquanto as informações sobre os desenvolvedores são armazenadas em *Developer*, seguindo seus atributos nas Tabelas 7 e 8. O *Affection* é medido com a média dos erros absolutos entre a personalidade dos desenvolvedores envolvidos e a esperada pela tarefa.

atributo	tipo	descrição
id	<i>inteiro</i>	identificador numérico para a tarefa
name	string	nome da tarefa
expected_duration	in	duração estimada em unidades de tempo
skills_needed	strings	conjunto de nomes das habilidades necessárias
traits	reais	medida de personalidade esperada para execução da tarefa
dependencies	inteiros	conjunto de tarefas que devem ser realizadas para que essa possa ser iniciada

Tabela 7 – Atributos da classe *Task*

atributo	tipo	descrição
id	<i>inteiro</i>	identificador numérico para a o desenvolvedor
name	string	nome do desenvolvedor
skills	dicionario<string, real>	conjunto de habilidade que esse desenvolvedor possui seguido de uma medida de proficiência
traits	reais	medida de personalidade do desenvolvedor

Tabela 8 – Atributos da classe *Developer*

4.3.2 Gene

As classes definidas na seção anterior dão fundamento para que os agendamentos possam ser feitos, para isso é necessário que se defina uma forma abstrata de se representar as soluções. Essa representação irá alimentar os objetos *Chromosome* e foi convenientemente chamada de *Gene*, possuindo dois vetores para armazenar a sequencia de tarefas e desenvolvedores que será utilizada na decodificação. A ordenação desses vetores têm impacto direto na decodificação das soluções e por isso a tarefa do algoritmo genético, nesse ponto, é encontrar um conjunto de sequencias ótimas para esses vetores. A escolha dessa representação se deu devido a facilidade em decodificar o gene numa solução factível (vide Algoritmo 5), pois as operações de cruzamento irão operar em apenas uma ocorrência de cada numero da permutação, sempre gerando indivíduos descendentes factíveis, que gera uma economia no tempo de execução ao evitar validação da soluções geradas e reconstrução no caso de infactibilidade.

4.3.3 Factory

De posse de todas as representações necessárias, segue a implementação dos métodos abstratos da classe *ChromosomeFactory* para que possam ser utilizados durante a busca do NSGA-II.

O método *decode* é responsável por transformar os vetores armazenados na classe *Gene* em agendamentos válidos e garante que não serão geradas soluções infactíveis. Sendo

T o vetor de permutação das tarefas e D o vetor de permutação dos desenvolvedores o Algoritmo 5 demonstra como uma solução é obtida a partir de um gene. Para que a condição da linha 5 seja satisfeita, todas as dependências da tarefa devem ter sido executadas em etapas anteriores e tendo em vista que um desenvolvedor só ficará alocado em uma tarefa por etapa não é necessário verificar a disponibilidade do recurso.

Algorithm 5 *DECODE*

```

1:  $etapas \leftarrow \emptyset$ 
2: while  $T \neq \emptyset$  do
3:    $etapaAtual \leftarrow \emptyset$ 
4:   for all  $tarefa \in T$  do
5:     if  $tarefa$  pode ser iniciada then
6:       adicionar tarefa em  $etapaAtual$ 
7:        $T \leftarrow T - \{tarefa\}$ 
8:       for all  $desenvolvedor \in D$  do
9:         if  $desenvolvedor$  possui habilidade não satisfeita para  $tarefa$  then
10:           Designar  $desenvolvedor$  para  $tarefa$ 
11:         if todas habilidades para  $tarefa$  foram satisfeitas then
12:           parar o loop
13:   Adicionar  $etapaAtual$  em  $etapas$ 
14: Definir o horário de início de cada tarefa em  $etapas$ 

```

O método *new* faz a criação de soluções aleatórias de maneira bem simples, obtém-se um vetor ordenado dos identificadores das tarefas e dos desenvolvedores e aplica-se uma operação de embaralhamento ³.

A Imagem 3 exemplifica o método de cruzamento utilizado para dois indivíduos. Essa operação é realizada para cada uma das listas presentes nos genes gerando-se duas novas listas que irão compor o gene do novo indivíduo. Os elementos dos vetores são pareados posicionalmente, gerando um novo vetor com o dobro de elemento dos pais, depois os elementos repetidos são retirados do resultado formando um novo gene.

$$\begin{array}{r}
 <4, 2, 0, 3, 1> \\
 + <2, 1, 4, 0, 3> \\
 \hline
 <4, 2, 2, 1, 0, 4, 3, 0, 1, 3> \\
 <4, 2, 1, 0, 3>
 \end{array}$$

Figura 3 – Cruzamento dos genes

Por fim, a mutação é feita pela troca de duas subsequências de cada uma presente

³ <https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.shuffle.html>

no gene. O tamanho e quais subsequências serão substituídas são determinadas aleatoriamente.

$\langle [4, 2], 1, [0, 3] \rangle$
 $\langle 0, 3, 1, 4, 2 \rangle$

Figura 4 – Mutação de um gene

4.3.4 Funções Objetivo

A aplicação final irá permitir que o usuário escolha os aspectos que devem ser otimizados tanto do agendamento quanto do escalonamento. Para isso foram definidas sete funções objetivos para serem utilizadas em conjunto. A Tabela 9 apresenta essas funções e da uma breve descrição sobre qual aspecto do agendamento ou do escalonamento a função está relacionada.

função objetivo	descrição
Makespan	mede a duração total do projeto em unidades de tempo
Affection	mede o quão bem os desenvolvedores se adequam às tarefas em que estão alocados
Work Force	somatório dos níveis de proficiência nas habilidades requisitadas por cada tarefa
Missing Skills	quantidade de habilidades que não foram supridas pela equipe disponível
Mean Tasks By Resource	quantidade média de tarefas que cada desenvolvedor estará encarregado
Total Cost	medida de custo do projeto em unidades monetárias
Usage	porcentagem de quanto tempo cada desenvolvedor está trabalhando

Tabela 9 – Funções objetivo

4.4 Command Line Interface

A última etapa do projeto foi a construção de uma interface que permitisse ao usuário interagir com o motor de otimização construído. A interface construída permite que sejam especificados, além da instância, os parâmetros do NSGA-II, as funções objetivo que serão utilizadas, semente para o gerador de números aleatórios, diretório de saída para os resultados, frequência para armazenar informações sobre a população gerada pelo algoritmo genético e a opção de obter somente uma solução com base na ordem em que as funções objetivos foram especificadas.

parâmetro	descrição
project_file	arquivo contendo as informações sobre o projeto
team_file	arquivo contendo as informações sobre a equipe
priority	lista de funções objetivo em ordem de prioridade
-pop_size	quantidade de indivíduos na população
-mut_prob	probabilidade de mutação
-max_gen	quantidade de gerações a serem executadas
-seed	semente do gerador de números aleatórios
-best	armazenar somente o melhor resultado (baseado na lista de prioridade)
-stats	frequência de gerações para armazenar os dados sobre a população
-out	pasta para armazenar os resultados

Tabela 10 – Parâmetros disponíveis da interface.

```
→ set-covering git:(master) ✗ python optimize.py --pop_size 110 --mut_prob
0.5 --max_gen 200 proj.json eqp.json makespan total_cost
```

Figura 5 – Exemplo de chamada da aplicação

A Figura 5 apresenta uma chamada da aplicação que será executada para o projeto descrito no arquivo *proj.json* juntamente com a equipe definida em *eqp.json* com objetivo de otimizar o *makespan* e *total_cost*. Serão executadas 200 gerações com 110 indivíduos com 50% de probabilidade de mutação. Os resultados serão armazenados em uma pasta chamada *results/solutions* em formato json, contendo o desempenho de cada solução para as funções objetivo selecionadas; o agendamento informando o início, tempo de execução e quais desenvolvedores estarão responsáveis por cada tarefa; e as alocações contendo informações sobre por quais tarefas cada funcionário estará encarregado.

5 Resultados

Neste capítulo serão apresentados e discutidos os resultados obtidos a partir dos experimentos realizados, variando os projetos as equipes e funções objetivo utilizadas na execução de cada instância.

5.1 Convergência

O primeiro teste que foi feito teve por objetivo observar a convergência da solução em relação a um único objetivo. Para avaliar a qualidade das fronteiras geradas foi utilizado um cálculo de *hypervolume* (FLEISCHER, 2003) que determina o volume de um sólido em um espaço multi-dimensional. Quanto maior for volume do sólido, mais próxima do ótimo a fronteira analisada se encontra. As Imagens 6, 7, 8, 9, 10 e 11 mostram o crescimento do *hypervolume* para cada instância durante 300 gerações utilizando todas as funções objetivo disponíveis deixando visível a melhora das soluções ao passar das gerações.

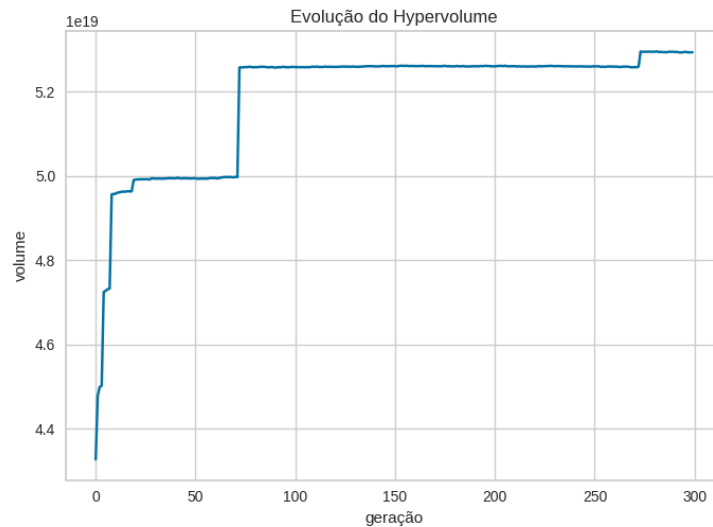
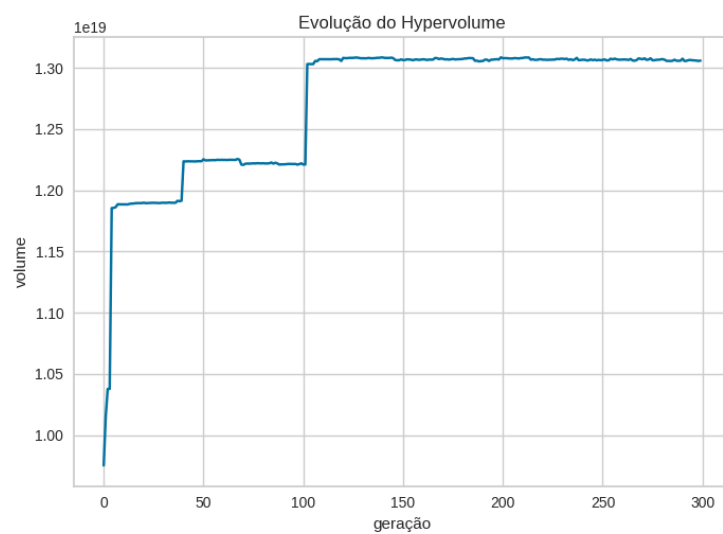
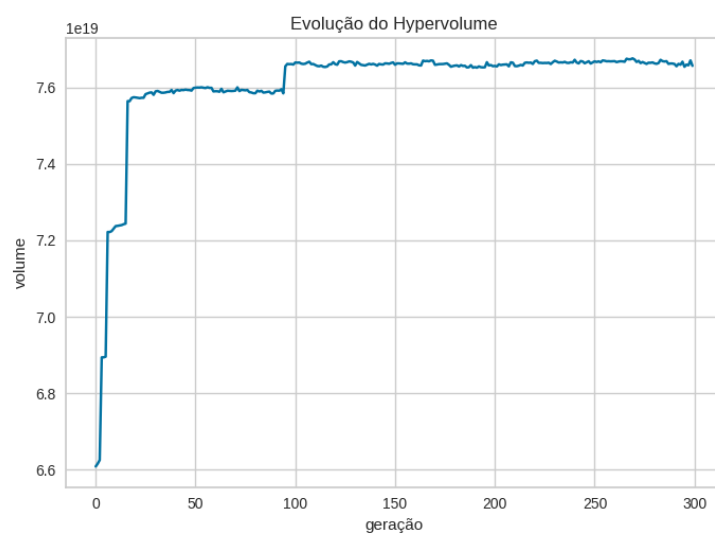
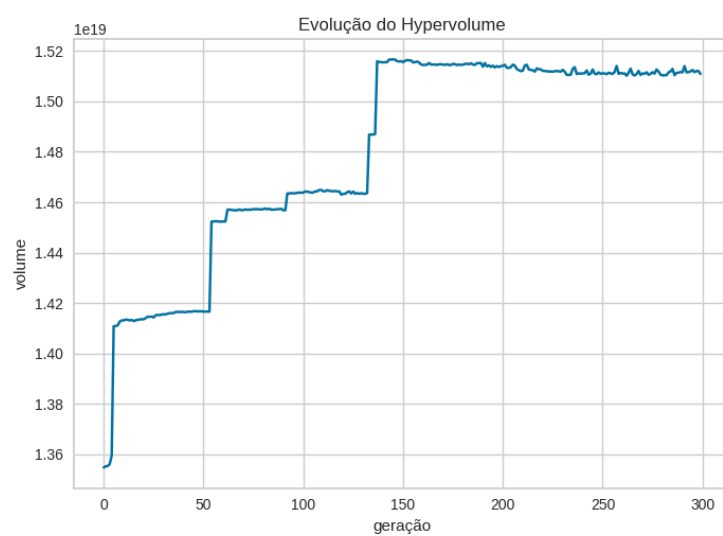
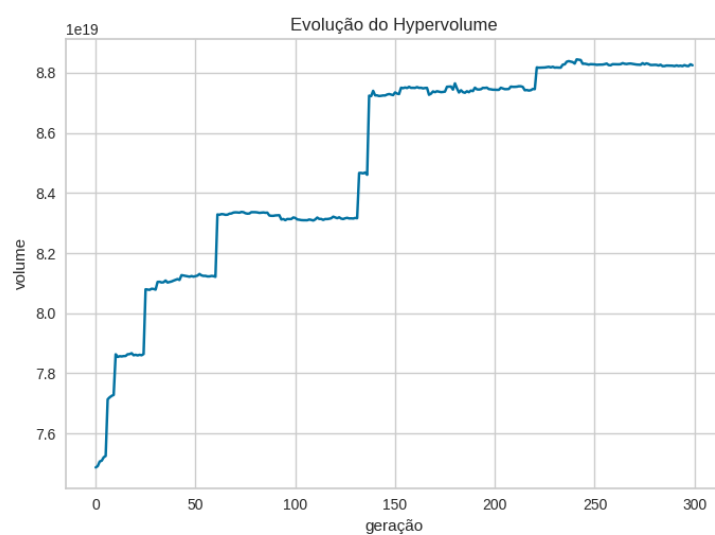


Figura 6 – *Hypervolume* Projeto J102_2 e Equipe TEAM01

Figura 7 – *Hypervolume* Projeto J102_2 e Equipe TEAM02Figura 8 – *Hypervolume* Projeto J125_9 e Equipe TEAM01

Figura 9 – *Hypervolume* Projeto J125_9 e Equipe TEAM02Figura 10 – *Hypervolume* Projeto M12_8 e Equipe TEAM01

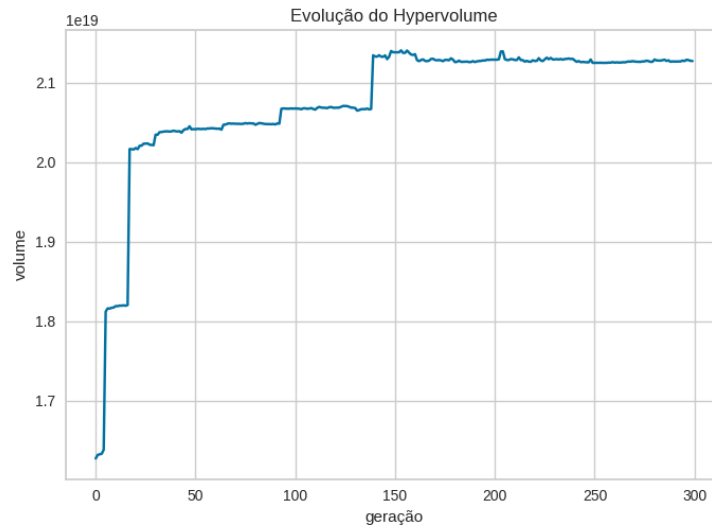


Figura 11 – *Hypervolume* Projeto M12_8 e Equipe TEAM02

5.2 Seleção de parâmetros

Para realizar a seleção dos parâmetros foram executados testes alternando os valores de *max_gen*, *mut_prob* e *pop_size*, utilizando, para cada um deles, um valor baixo, um médio e um alto. Em cada teste foi calculado o *hypervolume* da última geração sendo que esse valor foi utilizado para comparação dos resultados e escolha dos parâmetros. Para que esses valores pudessem ser comparados foi calculado um ponto de referência comum para todas as fronteiras das últimas gerações obtidas a partir das execuções da instância que utiliza o projeto J102_2 e equipe TEAM01. Esse ponto contém em cada uma de suas dimensões o pior valor para a função objetivo relacionada.

A Tabela 11 apresenta os resultados obtidos sendo que na primeira coluna estão os valores para o número máximo de gerações, na segunda, os valores para a probabilidade de mutação, na terceira, o tamanho da população e por último, o volume obtido para a última fronteira obtida. Nessa Tabela é possível perceber que os melhores valores se encontram na linha em que o valor da probabilidade de mutação é 0.4 e o número de indivíduos é 300 e o número máximo de gerações é 100, sendo essa a escolha para realização dos demais testes e definidos como valores padrão da aplicação.

max_gen	mut_prob	pop_size	volume
100	.2	100	232170605.21672878
100	.2	200	254799851.7229733
100	.2	300	267271539.09481728
100	.4	100	234696559.21271425
100	.4	200	256232631.69363922
100	.4	300	275369477.9235618
100	.6	100	228387196.18542582
100	.6	200	265773499.059349
100	.6	300	268425835.59490877
200	.2	100	231027210.13356867
200	.2	200	256978483.69031954
200	.2	300	273000990.59242463
200	.4	100	222741536.35931176
200	.4	200	259001786.239901
200	.4	300	270134742.9860277
200	.6	100	239363785.67477742
200	.6	200	259893505.66640517
200	.6	300	265858855.37307236
300	.2	100	226306678.34040606
300	.2	200	253773235.37360895
300	.2	300	272485720.605539
300	.4	100	227749424.0650496
300	.4	200	254072569.48850727
300	.4	300	271288094.5521869
300	.6	100	237029012.96039382
300	.6	200	253749729.53075004
300	.6	300	274339846.51810503

Tabela 11 – Resultados Hypervolume

5.3 Variação dos resultados

Com os parâmetros selecionados, cada instância foi executada uma vez para obtenção de resultados que possibilitassem a análise do comportamento dos resultados de

cada uma das instâncias. Cada conjunto de resultados foi disposto de três maneiras. A primeira delas é uma tabela na qual cada linha contém informações estatísticas sobre uma função objetivo, sendo elas valor mínimo, média, mediana, valor máximo, variância, desvio padrão e coeficiente de variação respectivamente. A segunda forma apresenta os dados em *boxplots*, um para cada função objetivo, evidenciando *outliers* presentes na amostra e a tendência da população em ter valores baixos, medianos ou altos para cada critério de avaliação dos resultados. A terceira forma de apresentação é um gráfico *Radviz* (SHARKO; GRINSTEIN; MARX, 2008) que tem por objetivo mostrar o impacto de cada função objetivo na disposição das soluções obtidas, nessas imagens cada ponto representa uma solução e quanto maior for o valor de uma determinada função objetivo mais próximo do eixo relacionado o ponto irá se encontrar, dessa forma os pontos irão se aproximar dos critérios para os quais se tem os piores resultados.

	min	média	mediana	max	variância	dp	cv
makespan	67	79.73	76.0	104	83.8	9.15	0.11
affection	0.21	0.67	0.64	1.26	0.04	0.21	0.31
work_force	-9.48	-5.61	-5.27	-3.44	2.04	1.43	-0.25
missing_skills	0	1.46	1.0	8	2.84	1.69	1.15
mean_tasks_by_resource	-5.0	-3.75	-3.75	-2.38	0.45	0.67	-0.18
total_cost	66030	96306.07	94804.0	129072	239603168.73	15479.12	0.16
usage	-0.7	-0.54	-0.55	-0.36	0.01	0.08	-0.15

Tabela 12 – Tabela de resultados Projeto J102_2 e Equipe TEAM01

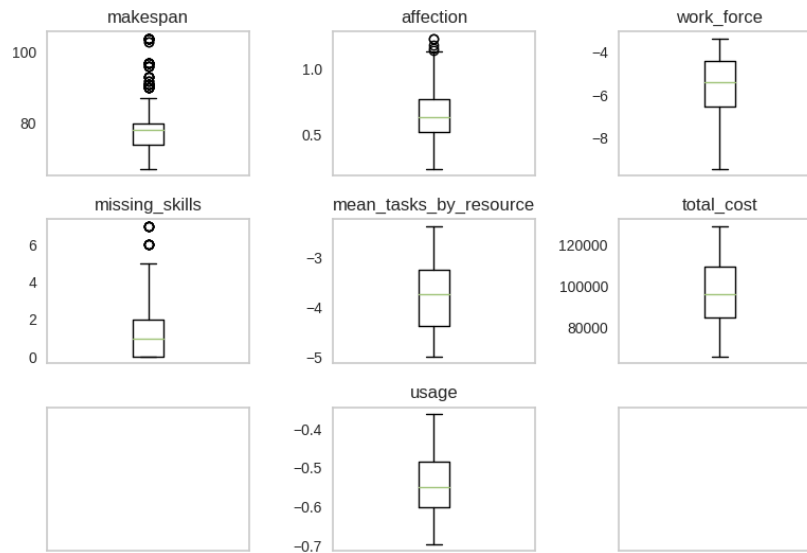
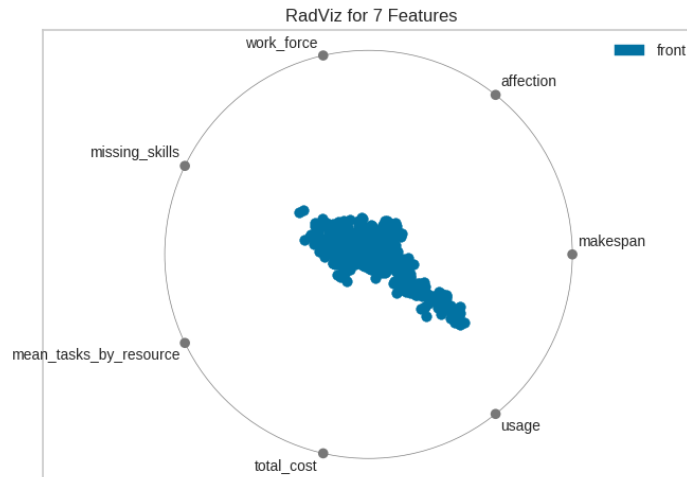


Figura 12 – *Boxplot* Projeto J102_2 e Equipe TEAM01

Figura 13 – *Radviz* Projeto J102_2 e Equipe TEAM01

Observando a Tabela 12 e a Figura 12 fica perceptível que os resultados têm uma tendência em ter melhores resultados para *makespan* e *missing_skills* e os valores de *work_force* são os que obtiveram a menor desempenho, se comparado a outras funções objetivo da mesma amostra. Se for analisado sem o auxílio de outras formas de representar os dados, o *Radviz* da Imagem 13 pode levar a entender que os valores menos interessantes estão relacionados à *missing_skills* e *usage*, enquanto o que acontece é uma disputa entre *work_force* e as demais funções objetivo.

	min	média	mediana	max	variância	dp	cv
makespan	67	76.35	76.0	80	9.77	3.13	0.04
affection	0.17	0.5	0.51	0.86	0.02	0.14	0.28
work_force	-6.23	-4.61	-4.52	-3.06	0.48	0.69	-0.15
missing_skills	2	3.36	3.0	8	1.35	1.16	0.35
mean_tasks_by_resource	-3.2	-2.57	-2.6	-1.8	0.11	0.33	-0.13
total_cost	24859	31805.16	31918.0	39363	9549235.73	3090.18	0.1
usage	-0.52	-0.4	-0.4	-0.28	0.0	0.06	-0.14

Tabela 13 – Tabela de resultados Projeto J102_2 e Equipe TEAM02

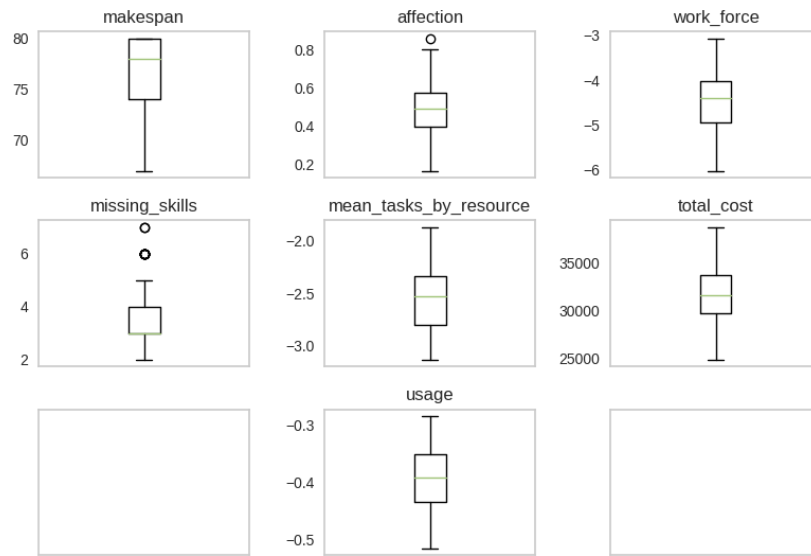


Figura 14 – *Boxplot* Projeto J102_2 e Equipe TEAM02

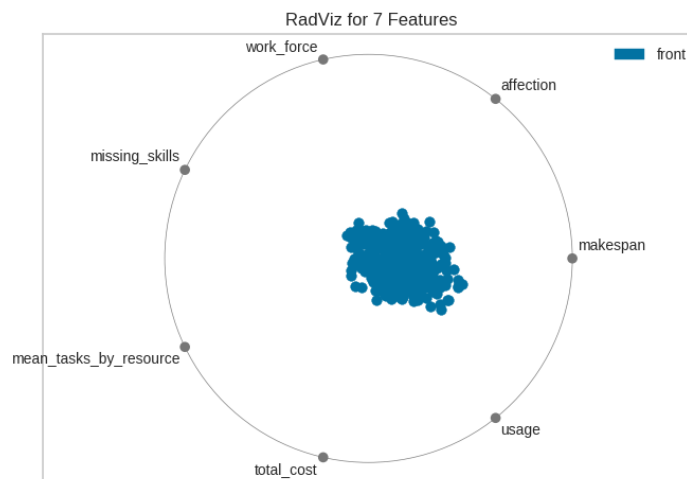
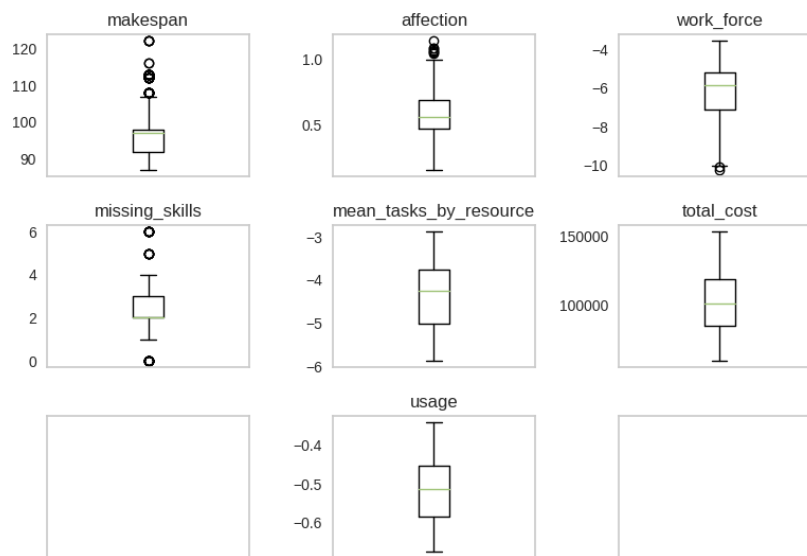
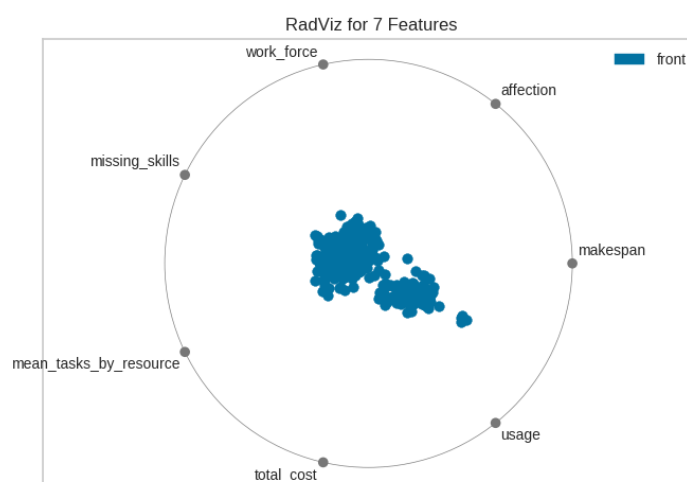


Figura 15 – *Radviz* Projeto J102_2 e Equipe TEAM02

Para essa instância, observando-se a Figura 15 é possível perceber um claro deslocamento das soluções em direção ao eixo *makespan*, o que indica fortemente que nessa amostra os resultados para esse critério não são tão interessantes se comparados ao restante das funções. Isso pode ser confirmado observando a Tabela 13 e a Imagem 14 que mostram a tendência dos resultados para valores mais altos que foram obtidos. Observe que dentre as demais soluções existe um certo equilíbrio, o que explica a concentração dos resultados em uma região do *Radviz*.

	min	média	mediana	max	variância	dp	cv
makespan	87	98.64	97.0	122	57.18	7.56	0.08
affection	0.16	0.58	0.56	1.27	0.03	0.18	0.31
work_force	-10.28	-6.28	-6.03	-3.54	2.01	1.42	-0.23
missing_skills	0	2.35	2.0	7	1.53	1.24	0.53
mean_tasks_by_resource	-6.0	-4.36	-4.25	-2.88	0.58	0.76	-0.17
total_cost	60343	103395.07	102079.5	155006	498752785.42	22332.77	0.22
usage	-0.68	-0.52	-0.52	-0.34	0.01	0.08	-0.16

Tabela 14 – Tabela de resultados Projeto J125_9 e Equipe TEAM01

Figura 16 – *Boxplot* Projeto J125_9 e Equipe TEAM01Figura 17 – *Radviz* Projeto J125_9 e Equipe TEAM01

A Figura 17 mostra claramente a formação de três grupos de solução. O menor desses grupos se aproxima bastante de *makespan* e *usage* o que indica que essas soluções

são referente a alguns dos *outliers* ruins em relação a essas funções e possivelmente aos menores valores dos critérios concorrentes como *missing_skills* e *work_force*. Os outros dois grupos mostram mais equilíbrio entre os resultados obtidos, porém cada um possui uma leve tendência para um determinado sentido.

Os resultados das demais instâncias não apresentaram resultados muito diferente dos que já foram apresentados. Sendo assim, seguem os resultados obtidos para as demais instâncias para que o leitor possa observar como cada amostra de resultados se comportou.

	min	média	mediana	max	variância	dp	cv
makespan	89	95.08	97.0	97	5.7	2.39	0.03
affection	0.26	0.59	0.56	1.12	0.04	0.19	0.33
work_force	-7.48	-5.45	-5.45	-3.77	0.71	0.84	-0.15
missing_skills	0	1.07	1.0	5	1.74	1.32	1.23
mean_tasks_by_resource	-3.87	-3.22	-3.27	-2.33	0.13	0.36	-0.11
total_cost	28968	38731.89	38126.5	50322	18766242.05	4332.0	0.11
usage	-0.48	-0.39	-0.39	-0.28	0.0	0.05	-0.12

Tabela 15 – Tabela de resultados Projeto J125_9 e Equipe TEAM02

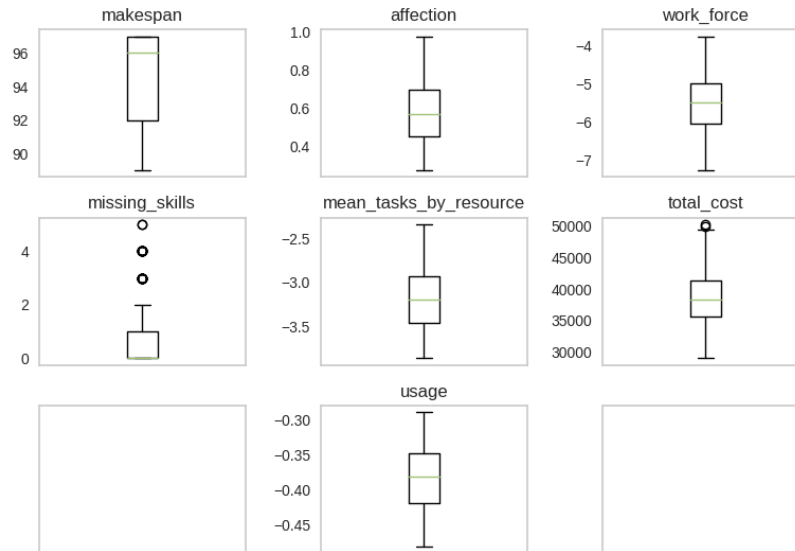


Figura 18 – *Boxplot* Projeto J125_9 e Equipe TEAM02

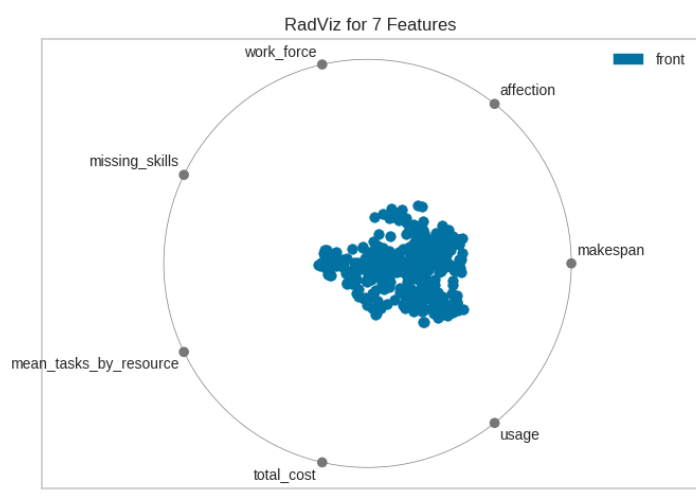


Figura 19 – Radviz Projeto J125_9 e Equipe TEAM02

	min	média	mediana	max	variância	dp	cv
makespan	94	115.63	111.0	155	160.43	12.67	0.11
affection	0.31	0.71	0.71	1.11	0.02	0.15	0.21
work_force	-10.28	-6.6	-6.45	-3.89	2.51	1.58	-0.24
missing_skills	0	4.19	4.0	13	9.63	3.1	0.74
mean_tasks_by_resource	-7.38	-5.4	-5.38	-3.62	0.97	0.98	-0.18
total_cost	98039	144061.13	144969.5	189478	476173216.2	21821.39	0.15
usage	-0.77	-0.56	-0.57	-0.36	0.01	0.09	-0.16

Tabela 16 – Tabela de resultados Projeto M12_8 e Equipe TEAM01

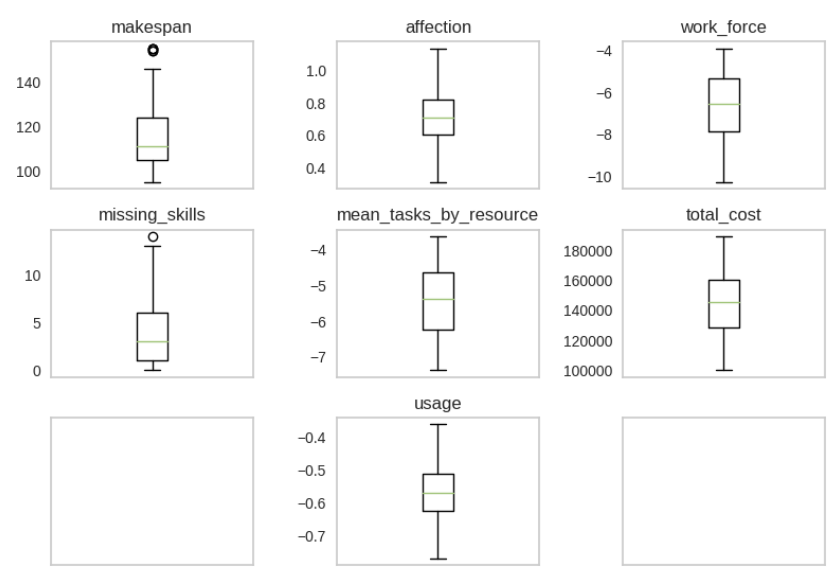


Figura 20 – Boxplot Projeto M12_8 e Equipe TEAM01

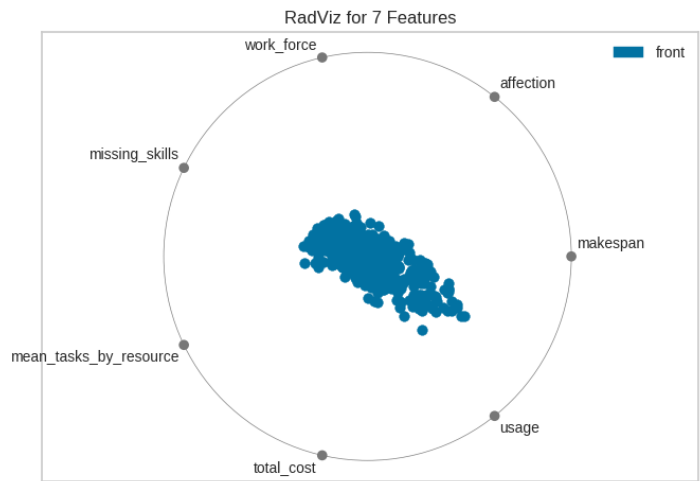


Figura 21 – *Radviz* Projeto M12_8 e Equipe TEAM01

	min	média	mediana	max	variância	dp	cv
makespan	94	107.29	105.0	127	43.42	6.59	0.06
affection	0.37	0.69	0.67	1.15	0.02	0.15	0.22
work_force	-8.41	-6.12	-6.09	-3.84	1.14	1.07	-0.17
missing_skills	2	6.06	6.0	14	9.06	3.01	0.5
mean_tasks_by_resource	-4.53	-3.73	-3.73	-2.87	0.16	0.4	-0.11
total_cost	34544	44175.37	44318.0	55294	26402357.56	5138.32	0.12
usage	-0.54	-0.42	-0.43	-0.31	0.0	0.05	-0.11

Tabela 17 – Tabela de resultados Projeto M12_8 e Equipe TEAM02

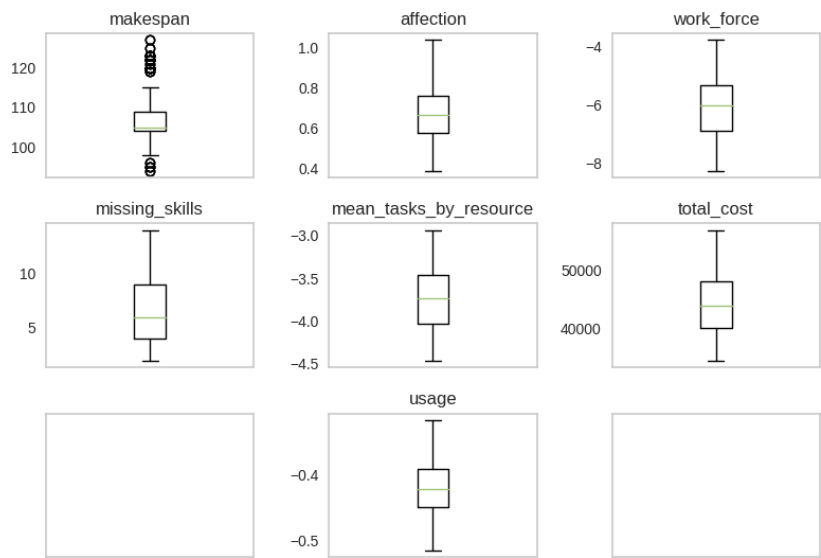


Figura 22 – *Boxplot* Projeto M12_8 e Equipe TEAM02

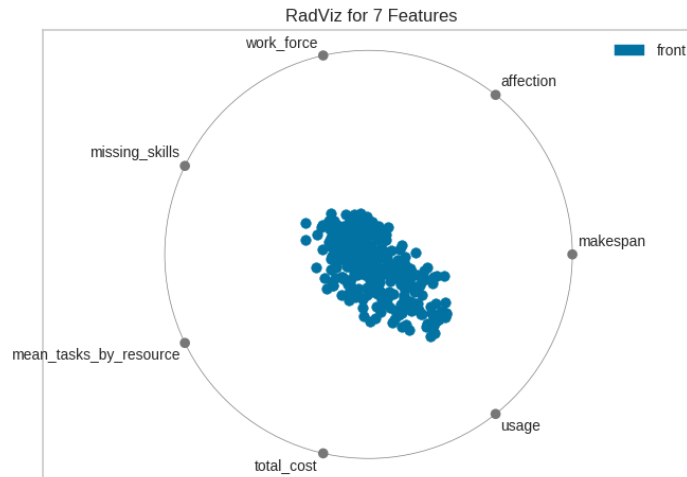


Figura 23 – *Radviz* Projeto M12_8 e Equipe TEAM02

Comparando-se as instâncias que utilizam os mesmos projetos, é possível notar que o valor mínimo obtido para o *makespan* é o mesmo, senão pelo projeto J125_9 e equipe TEAM02 que obteve um mínimo pouco maior que o da equipe TEAM01. Os demais valores obtiveram valores mais variados para cada instância, o que pode ser explicado pela dependência que essas funções têm sobre a equipe e o projeto utilizados.

5.4 Agendamentos e uso de recursos

As análises que foram feitas até agora são relativas aos conjuntos de soluções da melhor fronteira computada, demonstrando a qualidade e comportamento em relação às funções objetivo utilizadas. Essas análises são úteis para demonstrar que, ao utilizar a aplicação, o trabalho do usuário será o de ponderar entre os resultados obtidos para escolher aquele que melhor atende à suas necessidades. A seguir serão apresentadas algumas soluções, selecionadas de uma mesma população, individualmente. Cada uma delas foi selecionada por ter obtido o melhor desempenho para uma dada função objetivo, *makespan*, *total_cost* e *usage*, respectivamente.

As Imagens 24, 27 e 30 apresentam o sequenciamento das tarefas enquanto as Figuras 25, 28 e 31 mostram a ocupação dos desenvolvedores envolvidos no projeto ao longo do tempo. As figuras 26, 29 e 32 apresentam quais desenvolvedores estarão empenhados em quais tarefas. A Tabela 18 mostra o desempenho de cada uma das soluções, sendo que a primeira coluna indica à qual solução os valores se referem e as demais apresentam os valores das funções *makespan*, *affection*, *work_force*, *missing_skills*, *mean_tasks_by_resource*, *total_cost* e *usage*, nessa ordem.

	makespan	affection	work_force	missing_skills	mean_tasks_by_resource	total_cost	usage
1	67	0.48	-3.76	2	-2.75	74656	-0.46
2	80	0.81	-4.02	2	-2.75	66030	-0.37
3	72	0.81	-6.34	2	-4.37	110061	-0.69

Tabela 18 – Resultados individuais

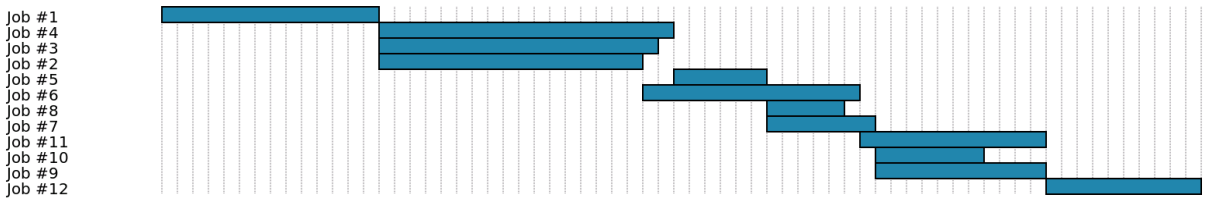


Figura 24 – Sequenciamento do resultado 1

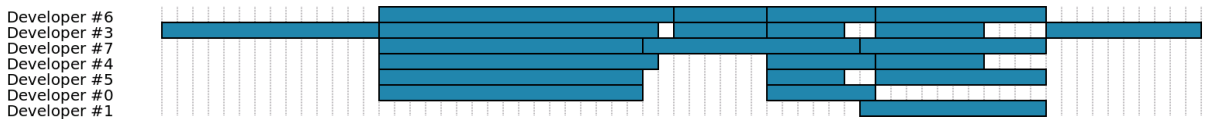


Figura 25 – Ocupações do resultado 1

	Job #1	Job #2	Job #3	Job #4	Job #5	Job #6	Job #7	Job #8	Job #9	Job #10	Job #11	Job #12
Developer #0												
Developer #1												
Developer #2												
Developer #3												
Developer #4												
Developer #5												
Developer #6												
Developer #7												

Figura 26 – Alocações do resultado 1

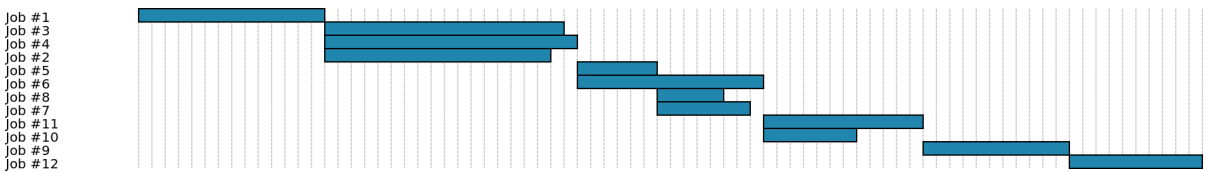


Figura 27 – Sequenciamento do resultado 2

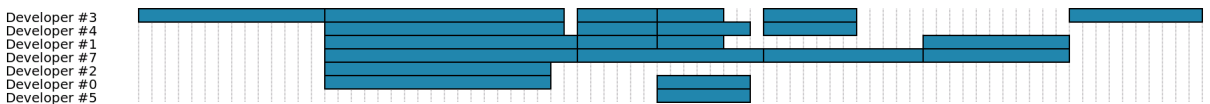


Figura 28 – Ocupações do resultado 2

	Job #1	Job #2	Job #3	Job #4	Job #5	Job #6	Job #7	Job #8	Job #9	Job #10	Job #11	Job #12
Developer #0												
Developer #1												
Developer #2												
Developer #3												
Developer #4												
Developer #5												
Developer #6												
Developer #7												

Figura 29 – Alocações do resultado 2

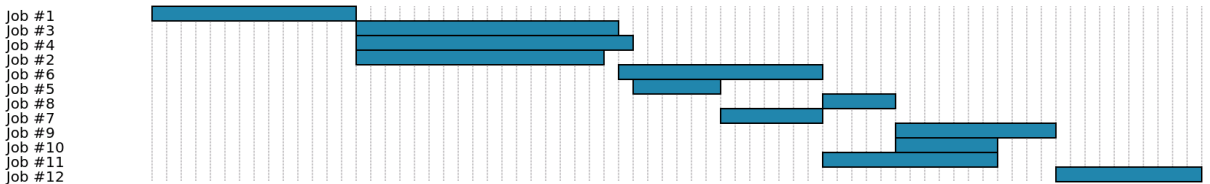


Figura 30 – Sequenciamento do resultado 3

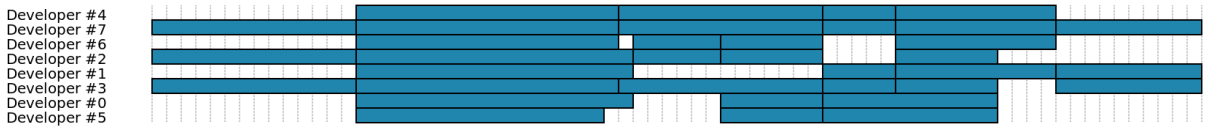


Figura 31 – Ocupações do resultado 3

	Job #1	Job #2	Job #3	Job #4	Job #5	Job #6	Job #7	Job #8	Job #9	Job #10	Job #11	Job #12
Developer #0												
Developer #1												
Developer #2												
Developer #3												
Developer #4												
Developer #5												
Developer #6												
Developer #7												

Figura 32 – Alocações do resultado 3

Observe como a alteração da equipe alocada afeta o agendamento resultante. Para a primeira solução somente o *Developer #2* não foi utilizado enquanto a segunda não

utilizou o *Developer* #6. Essa mudança resultou numa queda significativa do custo do projeto (redução de 11.5%) porém num aumento muito grande em relação ao tempo de desenvolvimento (aumento de 19.4%). No terceiro resultado todos os desenvolvedores foram utilizados, diminuindo o *makespan* em relação ao segundo porém, novamente, aumentando o custo.

5.5 Funções objetivo

Em problemas multi-objetivo é interessante que se utilize funções objetivo que não sejam equivalentes para que a partir do valor de uma das funções não seja possível inferir o outro, caso contrário o problema pode ser tratado com uma abordagem mono-objetivo.

A Tabela 19 mostra a correlação de *Pearson* entre os pares de funções objetivo. Os valores destacados indicam onde ocorreram correlações fortes, e os demais valores foram utilizados para auxiliar na análise dos valores obtidos.

Observe que, para algumas comparações, os valores mínimo e máximo mudam o sinal, dessa forma, as correlações obtidas variam de maneira que em alguns casos ao aumentar o valor de uma das funções objetivo a outra pode aumentar ou diminuir a depender da instância utilizada. Para as combinações em que, mesmo possuindo mínimo e máximo de mesmo sinal, o valor do desvio padrão é alto, as correlações mudam de forte para fraca. Em ambos os casos não é possível afirmar se existe alguma relação entre essas funções, restando analisar os valores das correlações.

Nas três últimas linhas da Tabela 19 é possível notar que as funções objetivo possuem correlação forte, em poucos casos média, indicando fortemente que possuem algum tipo de relação, que é estendida para *work_force* por meio de *mean_tasks_by_resource* e *total_cost*. As demais combinações apresentam correlações fracas ou médias, indicando que não estão relacionadas. Os valores de cada coluna da correlação foram retirados da fronteira de Pareto obtidas de cada instância na seguinte ordem: projeto J102_2 e equipe TEAM01, J102_2 e equipe TEAM02, J125_9 e equipe TEAM01, J125_9 e equipe TEAM02, M12_8 e equipe TEAM01, M12_8 e equipe TEAM02.

		correlação						min	max	média	variância	dp	cv
makespan	affection	-0.15	-0.49	0.1	0.07	0.09	-0.26	-0.49	0.1	-0.11	0.05	0.22	-2.04
makespan	work_force	-0.81	0.16	-0.79	-0.28	-0.79	-0.52	-0.81	0.16	-0.5	0.12	0.35	-0.7
makespan	missing_skills	-0.41	-0.01	-0.66	-0.32	-0.59	-0.47	-0.66	-0.01	-0.41	0.04	0.21	-0.51
makespan	mean_tasks_by_resource	-0.63	0.33	-0.52	-0.21	-0.63	-0.38	-0.63	0.33	-0.34	0.11	0.33	-0.98
makespan	total_cost	0.69	-0.37	0.54	-0.07	0.62	0.39	-0.37	0.69	0.3	0.15	0.39	1.3
makespan	usage	-0.15	0.58	-0.16	-0.0	-0.09	0.04	-0.16	0.58	0.04	0.06	0.25	6.92
affection	work_force	-0.09	-0.47	-0.26	-0.51	-0.4	0.17	-0.51	0.17	-0.26	0.06	0.24	-0.92
affection	missing_skills	-0.31	-0.18	-0.21	-0.55	-0.25	0.18	-0.55	0.18	-0.22	0.05	0.22	-0.98
affection	mean_tasks_by_resource	-0.0	-0.72	-0.46	-0.63	-0.54	0.2	-0.72	0.2	-0.36	0.11	0.34	-0.95
affection	total_cost	-0.09	0.32	0.33	-0.03	0.44	-0.48	-0.48	0.44	0.08	0.1	0.32	3.89
affection	usage	-0.08	-0.71	-0.48	-0.66	-0.61	0.17	-0.71	0.17	-0.4	0.11	0.33	-0.83
work_force	missing_skills	0.45	0.21	0.59	0.64	0.54	0.43	0.21	0.64	0.48	0.02	0.14	0.29
work_force	mean_tasks_by_resource	0.89	0.64	0.73	0.54	0.9	0.82	0.54	0.9	0.75	0.02	0.13	0.17
work_force	total_cost	-0.9	-0.59	-0.71	-0.36	-0.87	-0.71	-0.9	-0.36	-0.69	0.03	0.18	-0.26
work_force	usage	0.6	0.57	0.48	0.52	0.57	0.61	0.48	0.61	0.56	0.0	0.05	0.08
missing_skills	mean_tasks_by_resource	0.3	0.26	0.46	0.45	0.48	0.54	0.26	0.54	0.42	0.01	0.1	0.24
missing_skills	total_cost	-0.34	-0.31	-0.52	-0.09	-0.55	-0.65	-0.65	-0.09	-0.41	0.03	0.19	-0.45
missing_skills	usage	0.12	0.25	0.26	0.45	0.22	0.43	0.12	0.45	0.29	0.01	0.12	0.4
mean_tasks_by_resource	total_cost	-0.96	-0.75	-0.92	-0.59	-0.95	-0.86	-0.96	-0.59	-0.84	0.02	0.13	-0.16
mean_tasks_by_resource	usage	0.85	0.95	0.9	0.96	0.81	0.88	0.81	0.96	0.89	0.0	0.05	0.06
total_cost	usage	-0.78	-0.8	-0.85	-0.61	-0.79	-0.81	-0.85	-0.61	-0.77	0.01	0.08	-0.1

Tabela 19 – Correlações entre funções objetivo

Isso não indica que, ao utilizar essas funções objetivo, estaremos tratando um problema mono-objetivo, mas sim de que a quantidade de soluções encontradas será bem menor. Observe nas Tabelas 12 e 18 que o resultado 3 possui o menor valor de *usage* e não possui o maior valor de *total_cost*, porém o aumento desse valor é notável.

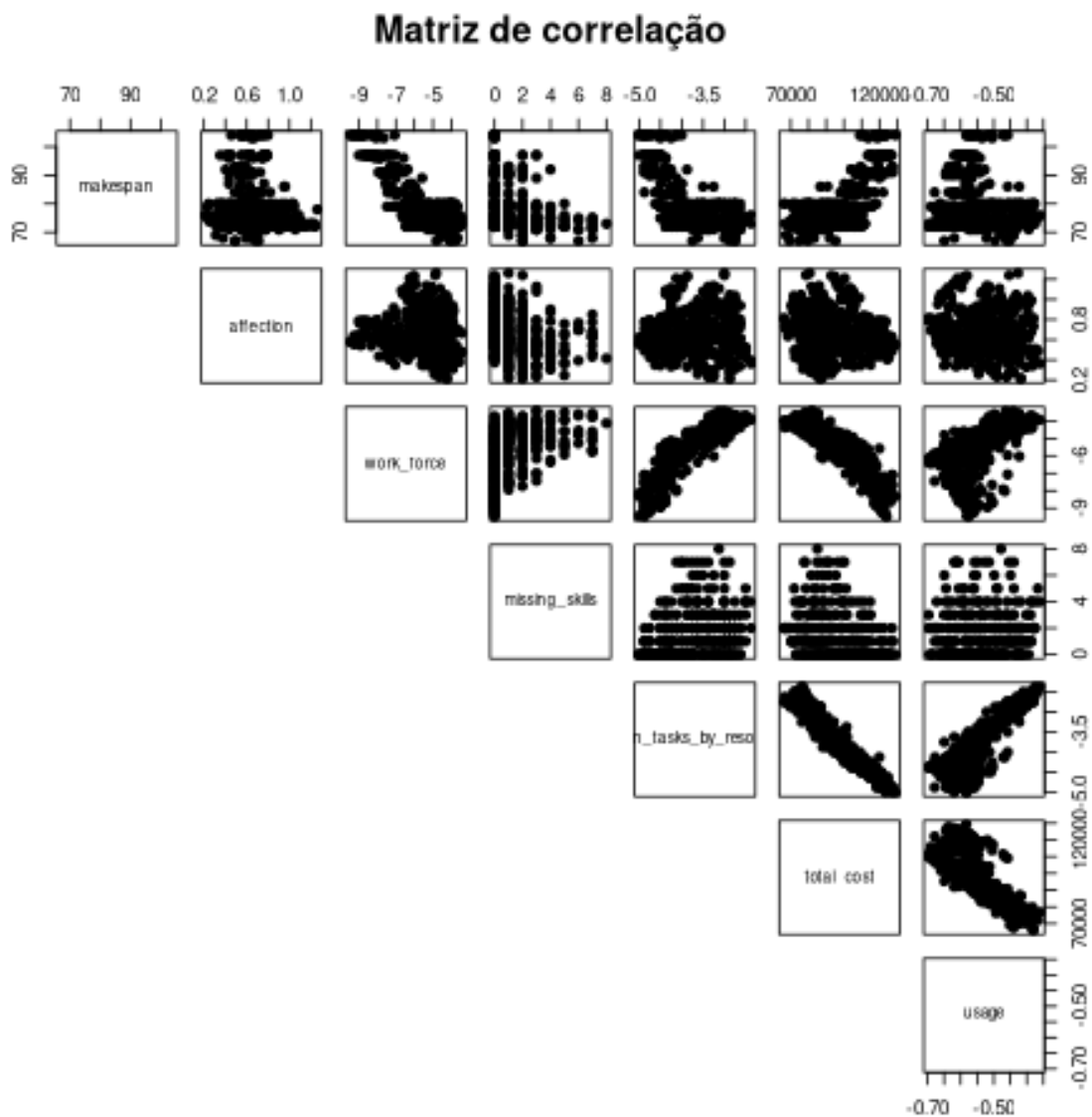


Figura 33 – Correlações para o Projeto J102_2 e Equipe TEAM01

Observe na Figura 33 a existência de correlações fortes de *work_force* com *total_cost* e *mean_tasks_by_resource*; de *mean_tasks_by_resource* com *total_cost* e *usage*; e de *total_cost* com *usage*. Aqui também é possível perceber que mesmo que a Tabela 19 tenha indicado que há uma correlação forte entre *makespan* e *work_force* não é possível afirmar que essas funções possuam algum tipo de relação.

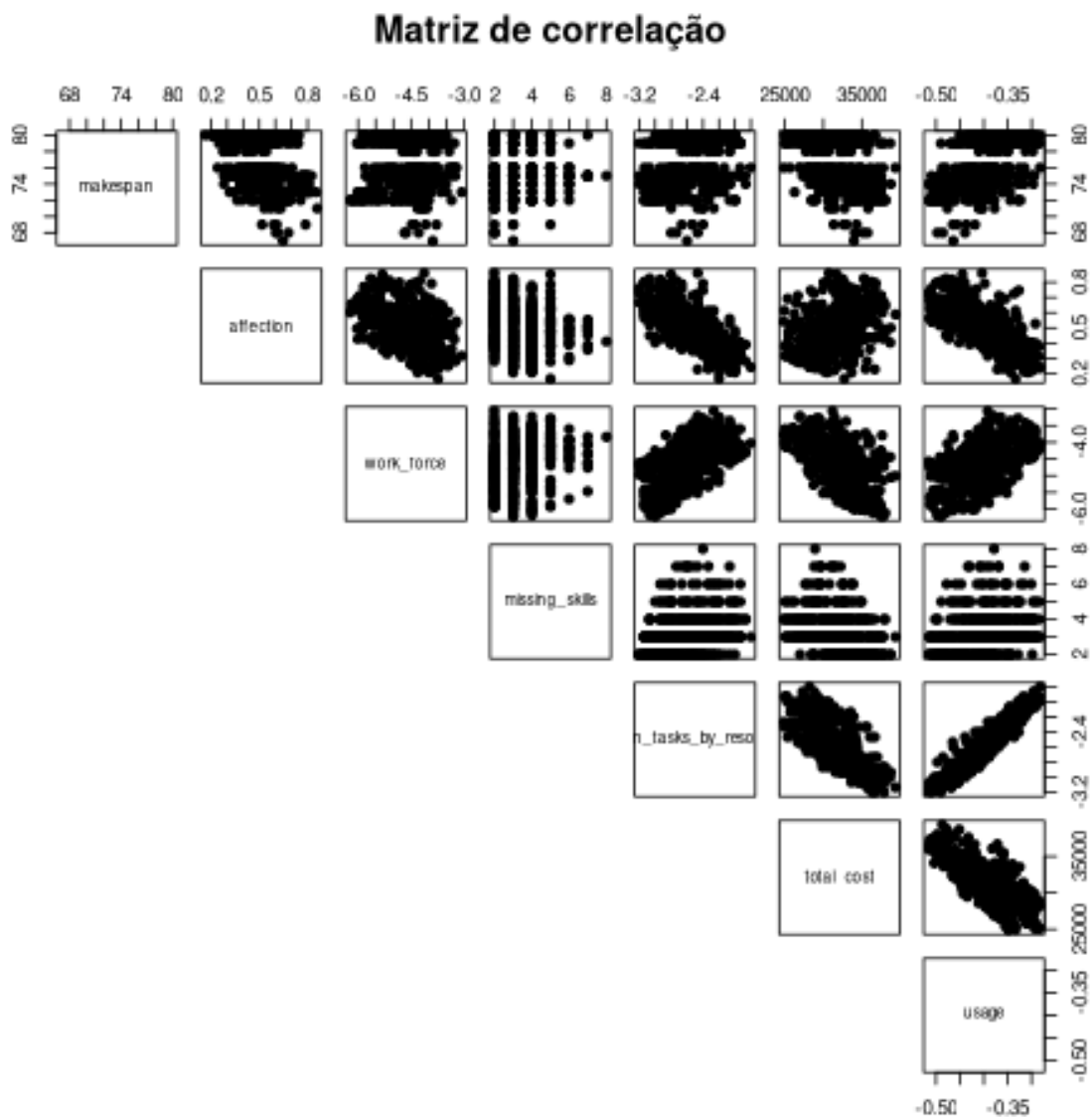


Figura 34 – Correlações para o Projeto J102_2 e Equipe TEAM02

Para a Imagem 34, comparando com a Imagem 33, as únicas combinações que se mantêm fortes são as das três últimas linhas da Tabela 19. Mas observe como a intensidade de *mean_tasks_by_resource* com *total_cost* e *total_cost* com *usage* diminuiu, porém *mean_tasks_by_resource* com *usage* aumentou. Mostrando que, mesmo que exista algum tipo de relação entre essas funções objetivo, a intensidade das mesmas ainda dependerá da instância em que são utilizadas.

6 Conclusões e Trabalhos Futuros

Este trabalho teve o objetivo de desenvolver o protótipo de uma aplicação capaz de auxiliar gerentes de projetos a agendar tarefas e escalonar a equipe disponível para a resolução de projetos. O que difere este trabalho dos demais existentes na literatura foi o fato de considerar características comportamentais da equipe no momento de alocar seus membros em atividades de projeto, visto que a literatura especializada considera que equipe competente e disponibilidade de recursos são fatores chave para o sucesso de projetos, daí a necessidade de saber alocar corretamente as pessoas em tarefas e papéis aos quais sejam adequadas. O projeto teve base em uma meta-heurística multi-objetivo e em problemas de agendamento de projetos com recursos limitados e para facilitar o uso da aplicação sem restringir a configuração da meta-heurística foram realizados testes para se definir valores padrões de execução em instâncias adaptadas do *benchmark* PS-PLIB, mais especificamente os conjuntos j10, j12 e m1 do problema MRCPSP. No trabalho foram considerados sete diferentes objetivos, a saber: *makespan*, *affection*, *work_force*, *missing_skills*, *mean_tasks_by_resource*, *total_cost* e *usage*. Os experimentos realizados demonstraram também que, independentemente da instância utilizada, os resultados obtidos tendem a melhorar ao longo das gerações do NSGA-II, indicando que o objetivo de otimizar os critérios escolhidos pelo usuário foi atingido. Também foi identificado que existe uma correlação entre três funções objetivo: *mean_tasks_by_resource*, *total_cost* e *usage*. Como esperado, com o aumento da utilização dos recursos aumenta-se a média de tarefas executadas por recurso, ou seja, estes são mais bem utilizados, reduzindo sua ociosidade. Outra relação percebida é que conforme aumenta-se a utilização dos recursos reduz-se o custo total do projeto visto que menos recursos são necessários para executar as mesmas tarefas. Para os demais objetivos considera-se que mais experimentos são necessários em quantidade maior de instâncias para confirmar correlações positivas e negativas entre eles. Independente disso os resultados observados no trabalho mostram que de fato a abordagem é efetiva em alocar pessoas às tarefas.

Pelos resultados obtidos pelo protótipo desenvolvido neste trabalho é possível afirmar que existe uma base para a construção de uma ferramenta mais avançada que poderá ser utilizada para auxiliar a tomada de decisões por gerentes de projeto. Sendo assim, esse trabalho será concluído com a indicação de algumas melhorias que podem ser feitas.

A construção de uma interface gráfica seria de grande ajuda para usuários que não possuem familiaridade com o uso de interfaces de linha de comando. Seria interessante a utilização de tecnologias utilizadas atualmente no desenvolvimento *web*, fornecendo uma solução simples e multi-plataforma para uma aplicação empresarial. Para aumentar a fidelidade do modelo utilizado neste trabalho também podem ser considerados calendários,

medidas monetárias reais, fator de aprendizagem, acréscimo de funções objetivo disponíveis para o usuário e impacto da experiência dos funcionários no tempo de execução das tarefas. Seria interessante a utilização de instâncias coletadas diretamente em empresas para realização de testes mais robustos do que os que foram apresentados. No caso específico do IFMG isso pode ser obtido em colaboração com a gestão de projetos do polo de inovação, sediado em Formiga. Também é válida a adaptação dessa aplicação para a utilização de modelos de meta conhecimento e sistemas de memória transativa. Por fim, é válida a experimentação de outros métodos de recomendação de pessoas para as tarefas, como por exemplo o algoritmo k -matching, que poderia servir de base para chegar a uma conclusão sobre quais tarefas um desenvolvedor deveria estar envolvido.

APÊNDICE A – Projetos

Projeto J102_2:

```
{
  "name": "J102_2",
  "tasks" : {
    "0": {
      "name": "Job #1",
      "expected_duration": 14,
      "skills_needed": ["H#18", "H#12", "H#13"],
      "traits": [0.76, 0.091, 0.13, 0.33, 0.46],
      "dependencies": []
    },

    "1": {
      "name": "Job #2",
      "expected_duration": 17,
      "skills_needed": ["H#06", "H#03", "H#10", "H#06", "H#12",
        "H#11", "H#11"],
      "traits": [0.11, 0.75, 0.37, 0.18, 0.86],
      "dependencies": [0]
    },

    "2": {
      "name": "Job #3",
      "expected_duration": 18,
      "skills_needed": ["H#19", "H#00", "H#03", "H#05", "H#13"],
      "traits": [0.52, 0.89, 0.25, 0.34, 0.18],
      "dependencies": [0]
    },

    "3": {
      "name": "Job #4",
      "expected_duration": 19,
      "skills_needed": ["H#07", "H#14", "H#06", "H#15", "H#03"],
      "traits": [0.15, 0.68, 0.7, 0.38, 0.76],
```

```
    "dependencies": [0]
  },

  "4": {
    "name": "Job #5",
    "expected_duration": 6,
    "skills_needed": ["H#06", "H#08", "H#08", "H#03", "H#10"],
    "traits": [0.32, 0.85, 0.58, 0.17, 0.01],
    "dependencies": [1]
  },

  "5": {
    "name": "Job #6",
    "expected_duration": 14,
    "skills_needed": ["H#02", "H#19", "H#18"],
    "traits": [0.59, 0.0011, 0.99, 0.72, 0.46],
    "dependencies": [1]
  },

  "6": {
    "name": "Job #7",
    "expected_duration": 7,
    "skills_needed": ["H#04", "H#11", "H#16", "H#14", "H#14",
    "H#13"],
    "traits": [0.52, 0.89, 0.31, 1.0, 0.54],
    "dependencies": [4]
  },

  "7": {
    "name": "Job #8",
    "expected_duration": 5,
    "skills_needed": ["H#01", "H#19", "H#11", "H#18", "H#10"],
    "traits": [0.54, 0.27, 0.81, 0.55, 0.18],
    "dependencies": [4]
  },

  "8": {
    "name": "Job #9",
    "expected_duration": 11,
```

```

    "skills_needed": ["H#00", "H#11", "H#09", "H#14", "H#11",
    "H#03"],
    "traits": [0.58, 0.64, 0.62, 0.032, 0.72],
    "dependencies": [3, 6, 7]
  },

  "9": {
    "name": "Job #10",
    "expected_duration": 7,
    "skills_needed": ["H#01", "H#04", "H#01", "H#05", "H#01"],
    "traits": [0.15, 0.65, 0.24, 0.68, 1.0],
    "dependencies": [2, 5, 6]
  },

  "10": {
    "name": "Job #11",
    "expected_duration": 12,
    "skills_needed": ["H#15", "H#11", "H#11", "H#07", "H#05"],
    "traits": [0.67, 0.76, 0.28, 0.74, 0.92],
    "dependencies": [2, 5]
  },

  "11": {
    "name": "Job #12",
    "expected_duration": 10,
    "skills_needed": ["H#19", "H#01", "H#18", "H#01"],
    "traits": [0.39, 0.016, 0.53, 0.67, 0.44],
    "dependencies": [8, 9, 10]
  }
}
}

```

Projeto J125_9:

```

{
  "name": "J125_9",
  "tasks" : {
    "0": {
      "name": "Job #1",
      "expected_duration": 16,

```



```
    "skills_needed": ["H#17", "H#02", "H#11", "H#01"],
    "traits": [0.49, 0.022, 0.61, 0.42, 0.24],
    "dependencies": []
},

"1": {
    "name": "Job #2",
    "expected_duration": 19,
    "skills_needed": ["H#18", "H#10", "H#16", "H#01"],
    "traits": [0.34, 0.063, 0.067, 0.3, 0.38],
    "dependencies": [0]
},

"2": {
    "name": "Job #3",
    "expected_duration": 10,
    "skills_needed": ["H#18", "H#18", "H#01", "H#08"],
    "traits": [0.7, 0.88, 0.2, 0.084, 0.15],
    "dependencies": [0]
},

"3": {
    "name": "Job #4",
    "expected_duration": 19,
    "skills_needed": ["H#05", "H#00", "H#19", "H#16", "H#15"],
    "traits": [0.76, 0.038, 0.21, 0.54, 0.36],
    "dependencies": [0, 1]
},

"4": {
    "name": "Job #5",
    "expected_duration": 16,
    "skills_needed": ["H#18", "H#14", "H#19", "H#09", "H#07",
    "H#17"],
    "traits": [0.22, 0.38, 0.37, 0.48, 0.77],
    "dependencies": [2]
},

"5": {
```

```
    "name": "Job #6",
    "expected_duration": 7,
    "skills_needed": ["H#18", "H#19", "H#00"],
    "traits": [0.53, 0.57, 0.39, 0.29, 0.53],
    "dependencies": [3]
  },

  "6": {
    "name": "Job #7",
    "expected_duration": 9,
    "skills_needed": ["H#06", "H#19"],
    "traits": [0.78, 0.3, 0.31, 0.066, 0.55],
    "dependencies": [1, 2, 3]
  },

  "7": {
    "name": "Job #8",
    "expected_duration": 11,
    "skills_needed": ["H#06", "H#14", "H#13", "H#15"],
    "traits": [0.3, 0.053, 0.089, 0.66, 0.77],
    "dependencies": [2]
  },

  "8": {
    "name": "Job #9",
    "expected_duration": 14,
    "skills_needed": ["H#07", "H#18", "H#04", "H#03", "H#05",
    "H#01", "H#02"],
    "traits": [0.55, 0.083, 0.66, 0.17, 0.25],
    "dependencies": [3, 7]
  },

  "9": {
    "name": "Job #10",
    "expected_duration": 5,
    "skills_needed": ["H#03", "H#00", "H#12", "H#11", "H#18"],
    "traits": [0.99, 0.35, 0.76, 0.25, 0.034],
    "dependencies": [4, 5, 7]
  },
```

```

    "10": {
      "name": "Job #11",
      "expected_duration": 6,
      "skills_needed": ["H#02", "H#15", "H#03", "H#01", "H#16",
        "H#07"],
      "traits": [0.4, 0.07, 0.84, 0.49, 0.83],
      "dependencies": [6, 8, 9]
    },

    "11": {
      "name": "Job #12",
      "expected_duration": 10,
      "skills_needed": ["H#08", "H#02", "H#06", "H#10"],
      "traits": [0.71, 0.95, 0.64, 0.063, 0.68],
      "dependencies": [1, 4, 5]
    },

    "12": {
      "name": "Job #13",
      "expected_duration": 5,
      "skills_needed": ["H#02", "H#12", "H#14", "H#09", "H#05"],
      "traits": [0.85, 0.26, 0.59, 0.94, 0.64],
      "dependencies": [1, 8, 9]
    },

    "13": {
      "name": "Job #14",
      "expected_duration": 13,
      "skills_needed": ["H#02", "H#05", "H#17", "H#17", "H#11"],
      "traits": [0.98, 0.36, 0.32, 0.15, 0.71],
      "dependencies": [10, 11, 12]
    }
  }
}

Projeto M12_8

{
  "name": "M12_8",

```

```
"tasks" : {
  "0": {
    "name": "Job #1",
    "expected_duration": 6,
    "skills_needed": ["H#06", "H#10", "H#12", "H#14"],
    "traits": [0.2, 0.73, 0.02, 0.053, 0.52],
    "dependencies": []
  },

  "1": {
    "name": "Job #2",
    "expected_duration": 16,
    "skills_needed": ["H#07", "H#06", "H#13", "H#09", "H#09",
      "H#17"],
    "traits": [0.75, 0.8, 0.23, 0.15, 0.42],
    "dependencies": [0]
  },

  "2": {
    "name": "Job #3",
    "expected_duration": 11,
    "skills_needed": ["H#09", "H#12", "H#11", "H#01", "H#13",
      "H#17", "H#10"],
    "traits": [0.036, 0.56, 0.83, 0.75, 0.32],
    "dependencies": [0]
  },

  "3": {
    "name": "Job #4",
    "expected_duration": 6,
    "skills_needed": ["H#11", "H#02", "H#10", "H#10", "H#00"],
    "traits": [0.85, 0.079, 0.68, 0.62, 0.2],
    "dependencies": [0]
  },

  "4": {
    "name": "Job #5",
    "expected_duration": 18,
    "skills_needed": ["H#03", "H#07", "H#01", "H#09", "H#01",
```

```
    "H#10"],
    "traits": [0.075, 0.17, 0.58, 0.14, 0.92],
    "dependencies": [1, 2, 3]
},

"5": {
    "name": "Job #6",
    "expected_duration": 10,
    "skills_needed": ["H#10", "H#12", "H#15", "H#14", "H#11",
    "H#04", "H#10"],
    "traits": [0.068, 0.61, 0.7, 0.36, 0.36],
    "dependencies": [3]
},

"6": {
    "name": "Job #7",
    "expected_duration": 15,
    "skills_needed": ["H#03", "H#18", "H#17", "H#03"],
    "traits": [0.24, 0.36, 0.68, 0.45, 0.39],
    "dependencies": [1]
},

"7": {
    "name": "Job #8",
    "expected_duration": 19,
    "skills_needed": ["H#04", "H#14", "H#11", "H#13", "H#13",
    "H#17", "H#12"],
    "traits": [0.21, 0.58, 0.63, 0.9, 0.41],
    "dependencies": [3]
},

"8": {
    "name": "Job #9",
    "expected_duration": 11,
    "skills_needed": ["H#19", "H#15", "H#00", "H#13"],
    "traits": [0.24, 0.64, 0.84, 0.78, 0.97],
    "dependencies": [7]
},
```

```
"9": {
  "name": "Job #10",
  "expected_duration": 16,
  "skills_needed": ["H#13", "H#15", "H#08", "H#03"],
  "traits": [0.59, 0.55, 0.18, 0.95, 0.86],
  "dependencies": [4, 8]
},

"10": {
  "name": "Job #11",
  "expected_duration": 6,
  "skills_needed": ["H#17", "H#11", "H#00", "H#05"],
  "traits": [0.6, 0.19, 0.12, 0.14, 0.82],
  "dependencies": [3]
},

"11": {
  "name": "Job #12",
  "expected_duration": 17,
  "skills_needed": ["H#09", "H#15"],
  "traits": [0.0067, 0.92, 0.38, 0.21, 0.36],
  "dependencies": [4, 6, 10]
},

"12": {
  "name": "Job #13",
  "expected_duration": 6,
  "skills_needed": ["H#07", "H#15"],
  "traits": [0.83, 0.93, 0.097, 0.41, 0.14],
  "dependencies": [5, 6, 9]
},

"13": {
  "name": "Job #14",
  "expected_duration": 16,
  "skills_needed": ["H#10", "H#09", "H#05"],
  "traits": [0.83, 0.17, 0.21, 0.89, 0.51],
  "dependencies": [7]
},
```

```
"14": {
  "name": "Job #15",
  "expected_duration": 5,
  "skills_needed": ["H#12", "H#18", "H#11"],
  "traits": [0.58, 0.94, 0.16, 0.25, 0.58],
  "dependencies": [5, 8, 11]
},

"15": {
  "name": "Job #16",
  "expected_duration": 12,
  "skills_needed": ["H#11", "H#12", "H#04", "H#06", "H#06"],
  "traits": [0.57, 0.96, 0.19, 0.96, 0.26],
  "dependencies": [1, 2, 13]
},

"16": {
  "name": "Job #17",
  "expected_duration": 17,
  "skills_needed": ["H#15", "H#19", "H#01"],
  "traits": [0.96, 0.57, 0.87, 0.38, 0.71],
  "dependencies": [10, 12, 13]
},

"17": {
  "name": "Job #18",
  "expected_duration": 13,
  "skills_needed": ["H#09", "H#01", "H#04", "H#13"],
  "traits": [0.94, 0.86, 0.23, 0.48, 0.45],
  "dependencies": [14, 15, 16]
}
}
}
```

APÊNDICE B – Equipes

Equipe TEAM#01:

```
{
  "name": "TEAM#01",
  "developers": {
    "0": {
      "name": "Developer #0",
      "skills": {
        "H#18": 0.8490965879112554,
        "H#08": 0.7915280458283853,
        "H#06": 0.859109551563512,
        "H#10": 0.02919464877348854,
        "H#11": 0.4773112377743971,
        "H#01": 0.1839444829885648,
        "H#07": 0.2037533925877243,
        "H#09": 0.6621468955707055,
        "H#15": 0.9693160590450199,
        "H#03": 0.8815277716113635,
        "H#14": 0.20149973870002102
      },
      "traits": [
        0.018103061681520538,
        0.84879655268332,
        0.8611892437905535,
        0.549464357503049,
        0.11588357086192103
      ],
      "cost": 242
    },
    "1": {
      "name": "Developer #1",
      "skills": {
        "H#17": 0.806412006478371,
        "H#16": 0.2620524971356143,
        "H#11": 0.9712925922070388,
```



```

        "H#04": 0.7209701534608635,
        "H#03": 0.9520427141292349,
        "H#06": 0.6153931964902385,
        "H#00": 0.20073499627704994,
        "H#18": 0.5765148006751553,
        "H#01": 0.8192444876014667,
        "H#02": 0.8219362541153825,
        "H#15": 0.6063665915987331,
        "H#08": 0.41701169304813357,
        "H#14": 0.9086984376613632
    },
    "traits": [
        0.5982169651852343,
        0.3443331241491301,
        0.5494870678968914,
        0.9990011251339526,
        0.9448757612864751
    ],
    "cost": 234
},
"2": {
    "name": "Developer #2",
    "skills": {
        "H#18": 0.22171021443022088,
        "H#00": 0.6845077621569702,
        "H#04": 0.4773265087065375,
        "H#02": 0.7924019349486493,
        "H#03": 0.6324876112180153,
        "H#12": 0.6777333423906998,
        "H#10": 0.9655605810052714,
        "H#09": 0.36987514863964577,
        "H#06": 0.02223514130223969,
        "H#16": 0.2844502750102008,
        "H#05": 0.08854895194768275,
        "H#08": 0.7762063166331105,
        "H#14": 0.5934574592988185
    },
    "traits": [
        0.2767788997497025,

```

```

        0.6508866725990571,
        0.5991546157831042,
        0.49633428136710545,
        0.2113773526302859
    ],
    "cost": 247
},
"3": {
    "name": "Developer #3",
    "skills": {
        "H#09": 0.004191653708202625,
        "H#13": 0.22974984984326818,
        "H#17": 0.41036049320561907,
        "H#19": 0.42019496709179993,
        "H#01": 0.5676298288858624,
        "H#00": 0.8283026769440591,
        "H#18": 0.7514162112926043,
        "H#14": 0.13363323556643913,
        "H#08": 0.9859898287122979,
        "H#05": 0.9012714697931254,
        "H#10": 0.5676450962182727,
        "H#15": 0.311300526453951,
        "H#12": 0.4214706308882292
    },
    "traits": [
        0.15995448643968824,
        0.2822374981085828,
        0.39005750050109766,
        0.234843931794124,
        0.9828038312401727
    ],
    "cost": 507
},
"4": {
    "name": "Developer #4",
    "skills": {
        "H#10": 0.0009665563565144364,
        "H#08": 0.8281415613950575,
        "H#15": 0.37222534704368093,

```

```

        "H#02": 0.7532671348192557,
        "H#14": 0.9694770086065059,
        "H#04": 0.5692966470893576,
        "H#03": 0.5351831527955814,
        "H#16": 0.8975283627639556
    },
    "traits": [
        0.19687006235817328,
        0.8355689538089912,
        0.19781857506293887,
        0.3484982211345291,
        0.10659963818823859
    ],
    "cost": 168
},
"5": {
    "name": "Developer #5",
    "skills": {
        "H#12": 0.5698339889499031,
        "H#06": 0.8126909903997203,
        "H#04": 0.48589946905711756,
        "H#09": 0.45541769046386227,
        "H#07": 0.24428639672918873,
        "H#11": 0.5478996228104482,
        "H#02": 0.1925812504984752,
        "H#13": 0.6996896292620183,
        "H#05": 0.1378403943777129,
        "H#00": 0.22309985773595287,
        "H#15": 0.852786637995478
    },
    "traits": [
        0.3456060658033092,
        0.9511718167090175,
        0.03320831850025019,
        0.19767151622074475,
        0.08068183878770951
    ],
    "cost": 407
},

```

```
"6": {
  "name": "Developer #6",
  "skills": {
    "H#08": 0.9988370578293991,
    "H#06": 0.5373143684907414,
    "H#15": 0.09685179216001405,
    "H#14": 0.33546391003098497,
    "H#09": 0.7104961340077578,
    "H#07": 0.8562608204501581,
    "H#02": 0.4730451664708828,
    "H#00": 0.39627200812159724,
    "H#03": 0.9189305406748178,
    "H#16": 0.46526617273722115
  },
  "traits": [
    6.280638893430002e-05,
    0.8710893270179884,
    0.4626162872277161,
    0.7561660604629371,
    0.37697126880510257
  ],
  "cost": 270
},
"7": {
  "name": "Developer #7",
  "skills": {
    "H#07": 0.3660132408741482,
    "H#18": 0.7764984825766891,
    "H#08": 0.13399216926995505,
    "H#05": 0.5075946102537492,
    "H#09": 0.5769584456347653,
    "H#15": 0.3488339040414724,
    "H#14": 0.007282756370344368,
    "H#17": 0.13931853506512648,
    "H#06": 0.5784444619310328,
    "H#02": 0.030026516829840144,
    "H#16": 0.30594420459568494
  },
  "traits": [
```

```

        0.24757551384971455,
        0.6544137780390832,
        0.5172844753832868,
        0.6523976390561488,
        0.05900379088301577
    ],
    "cost": 121
}
}
}

```

Equipe TEAM#02

```

{
    "name": "TEAM#02",
    "developers": {
        "0": {
            "name": "Developer #0",
            "skills": {
                "H#12": 0.6258766345852803,
                "H#01": 0.4773723199219231,
                "H#03": 0.5544999023212205,
                "H#05": 0.5024101827367098,
                "H#02": 0.6772679261043616,
                "H#16": 0.8325806950980034
            },
            "traits": [
                0.07181579369474811,
                0.10849887264206537,
                0.7016483570714678,
                0.06956476736443407,
                0.3926992776932413
            ],
            "cost": 66
        },
        "1": {
            "name": "Developer #1",
            "skills": {
                "H#10": 0.7094664776934836,
                "H#15": 0.5062593865446956,

```

```
        "H#04": 0.5174244073274987,
        "H#17": 0.9361151273793249,
        "H#16": 0.941864976509528,
        "H#05": 0.5391425899813983,
        "H#14": 0.7059409935875354
    },
    "traits": [
        0.6184615018884849,
        0.3160238080808242,
        0.37905514765597936,
        0.7172229735166847,
        0.4345531043649611
    ],
    "cost": 84
},
"2": {
    "name": "Developer #2",
    "skills": {
        "H#01": 0.9312362182341838,
        "H#16": 0.9950624934961521,
        "H#06": 0.24512476357849733,
        "H#02": 0.22738882749227318,
        "H#00": 0.19782402728431991
    },
    "traits": [
        0.3383506557947441,
        0.705337172612237,
        0.8867013651129747,
        0.518220835251711,
        0.8338456090382286
    ],
    "cost": 40
},
"3": {
    "name": "Developer #3",
    "skills": {
        "H#06": 0.7199098292132771,
        "H#13": 0.06925924760617241,
        "H#15": 0.3608919178525444,
```

```
        "H#10": 0.6925061577152762,
        "H#09": 0.135723014469479,
        "H#08": 0.02662711398995632,
        "H#16": 0.11267889212575344,
        "H#01": 0.7639886795727268
    },
    "traits": [
        0.8041063517194357,
        0.7129039578162649,
        0.03802232915277737,
        0.4906393931526992,
        0.12805408058085033
    ],
    "cost": 88
},
"4": {
    "name": "Developer #4",
    "skills": {
        "H#07": 0.48567277804618536,
        "H#19": 0.7359507489733407,
        "H#11": 0.2968336200819057,
        "H#14": 0.05737338104488776,
        "H#04": 0.26603053649172104,
        "H#02": 0.445770952132867,
        "H#10": 0.6371608555449331
    },
    "traits": [
        0.5422844051349829,
        0.6778340393880102,
        0.5199653778681921,
        0.529301995200487,
        0.7304502624811199
    ],
    "cost": 98
},
"5": {
    "name": "Developer #5",
    "skills": {
        "H#05": 0.7553044515584303,
```

```

        "H#06": 0.5781810246081951,
        "H#00": 0.7297628534371701,
        "H#08": 0.7899134837979577,
        "H#07": 0.2509094935461976,
        "H#04": 0.4534752727680976,
        "H#09": 0.709699333439267
    },
    "traits": [
        0.7967117437387634,
        0.4629938428077728,
        0.012629723290336847,
        0.5667140623642833,
        0.6620623502957143
    ],
    "cost": 70
},
"6": {
    "name": "Developer #6",
    "skills": {
        "H#12": 0.8007726218169858,
        "H#15": 0.6962736345658537,
        "H#06": 0.9772550645380702,
        "H#04": 0.12172416465195246
    },
    "traits": [
        0.3180357643430376,
        0.1814424476597245,
        0.202553275008171,
        0.7909320195418785,
        0.6881410854356742
    ],
    "cost": 32
},
"7": {
    "name": "Developer #7",
    "skills": {
        "H#18": 0.5355809642177393,
        "H#03": 0.23457903977397554,
        "H#05": 0.6545678087253819,

```



```
        "H#06": 0.5398434435967177,
        "H#16": 0.7304262596390193,
        "H#13": 0.5193303173459681,
        "H#17": 0.5560524223400927
    },
    "traits": [
        0.47537946486138283,
        0.527429320960961,
        0.09716956248256892,
        0.6049052376351035,
        0.19055870576311829
    ],
    "cost": 98
},
"8": {
    "name": "Developer #8",
    "skills": {
        "H#02": 0.03091961533178622,
        "H#00": 0.9603876808190394,
        "H#13": 0.17044629945265333,
        "H#15": 0.013754592621107675,
        "H#08": 0.5202703739312929
    },
    "traits": [
        0.3918160154674849,
        0.31961595234067564,
        0.1186640682364728,
        0.1317325298440185,
        0.63633110843196
    ],
    "cost": 40
},
"9": {
    "name": "Developer #9",
    "skills": {
        "H#01": 0.40529863078913875,
        "H#18": 0.7457938972727751,
        "H#00": 0.7629543152540277,
        "H#14": 0.9935702536333372,
```

```
        "H#06": 0.8227822910301398,
        "H#15": 0.8586226864774725,
        "H#05": 0.915537787264495
    },
    "traits": [
        0.2335779270804138,
        0.5497438060418469,
        0.8978392329330988,
        0.7709553230851708,
        0.8990609562134436
    ],
    "cost": 56
},
"10": {
    "name": "Developer #10",
    "skills": {
        "H#18": 0.8906200249826489,
        "H#01": 0.27533583096116576,
        "H#15": 0.49541819279806376,
        "H#16": 0.9144896554564954,
        "H#03": 0.5696636381971222,
        "H#06": 0.4998754072904794,
        "H#02": 0.70349172897397,
        "H#19": 0.9652876550920806
    },
    "traits": [
        0.8149959198613851,
        0.8572314221922421,
        0.5838742067884014,
        0.11765647893806486,
        0.6679600161947942
    ],
    "cost": 120
},
"11": {
    "name": "Developer #11",
    "skills": {
        "H#06": 0.28006963183633193,
        "H#15": 0.8763919048357497,
```

```
        "H#04": 0.6090108889926868,
        "H#10": 0.4500248310580385
    },
    "traits": [
        0.04192990829713927,
        0.21301574163027215,
        0.6145088605456637,
        0.0018535751619846996,
        0.6531837135908162
    ],
    "cost": 48
},
"12": {
    "name": "Developer #12",
    "skills": {
        "H#17": 0.5996071701156918,
        "H#08": 0.3694823889443355,
        "H#15": 0.9043358119229395,
        "H#02": 0.438780281245566
    },
    "traits": [
        0.9958376273936953,
        0.17515262956262256,
        0.037974804517270644,
        0.5623884768233689,
        0.36874600863723617
    ],
    "cost": 36
},
"13": {
    "name": "Developer #13",
    "skills": {
        "H#01": 0.7419201593308943,
        "H#15": 0.9841073224105562,
        "H#05": 0.3831916148252028,
        "H#06": 0.5394557112625918,
        "H#04": 0.527633356506717
    },
    "traits": [
```

```
        0.9167850224931444,  
        0.6570369781755226,  
        0.12870889619697856,  
        0.5904298276214621,  
        0.3732643465521589  
    ],  
    "cost": 35  
},  
"14": {  
    "name": "Developer #14",  
    "skills": {  
        "H#19": 0.1483325548647727,  
        "H#09": 0.46130707994093256,  
        "H#07": 0.6907863089061417,  
        "H#17": 0.4419357272629908,  
        "H#18": 0.8607702652693523,  
        "H#15": 0.18290096845485315  
    },  
    "traits": [  
        0.47493657665056876,  
        0.26143924183098055,  
        0.7680413914908024,  
        0.3150493154135131,  
        0.3270340004351291  
    ],  
    "cost": 96  
}  
}  
}
```

Referências

- BECCENERI, J. C. Meta-heurísticas e otimização combinatória: Aplicações em problemas ambientais. *INPE, Sao José dos Campos*, 2008. Citado na página 15.
- BELLENGUEZ, O.; NÉRON, E. Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. In: SPRINGER. *International Conference on the Practice and Theory of Automated Timetabling*. [S.l.], 2004. p. 229–243. Citado na página 20.
- BORGES, L. E. *Python para desenvolvedores: aborda Python 3.3*. [S.l.]: Novatec Editora, 2014. Citado na página 22.
- BRUCKER, P. et al. Resource-constrained project scheduling: Notation, classification, models, and methods. *European journal of operational research*, Elsevier, v. 112, n. 1, p. 3–41, 1999. Citado na página 19.
- CARBNO, C. Optimal resource allocation for projects. *Project Management Journal*, v. 30, n. 2, p. 22–31, 1999. Citado na página 12.
- CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Citado na página 16.
- DEB, K. *Multi-objective optimization using evolutionary algorithms*. [S.l.]: John Wiley & Sons, 2001. v. 16. Citado 3 vezes nas páginas 15, 16 e 27.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002. Citado 5 vezes nas páginas 8, 16, 17, 18 e 19.
- FLEISCHER, M. The measure of pareto optima applications to multi-objective metaheuristics. In: SPRINGER. *International Conference on Evolutionary Multi-Criterion Optimization*. [S.l.], 2003. p. 519–533. Citado na página 32.
- GAMMA, E. et al. Elements of reusable object-oriented software. *Design Patterns. massachusetts: Addison-Wesley Publishing Company*, 1995. Citado na página 26.
- HAPKE, M.; JASZKIEWICZ, A.; SŁOWIŃSKI, R. Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research*, Elsevier, v. 107, n. 2, p. 315–324, 1998. Citado na página 20.
- HARTMANN, S.; BRISKORN, D. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*, Elsevier, v. 207, n. 1, p. 1–14, 2010. Citado 2 vezes nas páginas 19 e 20.
- ISO, N. 10006. *Gestão da Qualidade–Diretrizes para a Qualidade no Gerenciamento de Projetos*, 2000. Citado na página 12.
- KOLISCH, R.; HEIMERL, C. An efficient metaheuristic for integrated scheduling and staffing it projects based on a generalized minimum cost flow network. *Naval Research Logistics (NRL)*, Wiley Online Library, v. 59, n. 2, p. 111–127, 2012. Citado na página 21.

- KOLISCH, R.; SPRECHER, A. Psplib-a project scheduling problem library: Or software-orsep operations research software exchange program. *European journal of operational research*, Elsevier, v. 96, n. 1, p. 205–216, 1997. Citado na página 23.
- MERKLE, D.; MIDDENDORF, M.; SCHMECK, H. Ant colony optimization for resource-constrained project scheduling. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 4, p. 333–346, 2002. Citado 2 vezes nas páginas 16 e 19.
- SHARKO, J.; GRINSTEIN, G.; MARX, K. A. Vectorized radviz and its application to multiple cluster datasets. *IEEE transactions on Visualization and Computer Graphics*, IEEE, v. 14, n. 6, p. 1444–1427, 2008. Citado na página 37.
- STANDISH, G. *The CHAOS report. 2015*. 2014. Citado na página 12.
- STANDISH, G. *The CHAOS report. 2015*. 2015. Citado na página 13.
- STYLIANOU, C.; ANDREOU, A. S. A multi-objective genetic algorithm for software development team staffing based on personality types. In: SPRINGER. *IFIP International Conference on Artificial Intelligence Applications and Innovations*. [S.l.], 2012. p. 37–47. Citado 2 vezes nas páginas 20 e 21.
- TUPES, E. C.; CHRISTAL, R. E. Recurrent personality factors based on trait ratings. *Journal of personality*, Wiley Online Library, v. 60, n. 2, p. 225–251, 1992. Citado na página 20.
- VARGAS, R. V. *Gerenciamento de Projetos (6a edição)*. [S.l.]: Brasport, 2005. Citado na página 12.
- WHITLEY, D. A genetic algorithm tutorial. *Statistics and computing*, Springer, v. 4, n. 2, p. 65–85, 1994. Citado na página 16.
- WU, M.-C.; SUN, S.-H. A project scheduling and staff assignment model considering learning effect. *The International Journal of Advanced Manufacturing Technology*, Springer, v. 28, n. 11-12, p. 1190–1195, 2006. Citado na página 21.