

---

Curso Sistemas de Informação  
Universidade Estadual de Mato Grosso do Sul

---

**Implementação de uma heurística gulosa para o problema de otimização de portfólios**

João Pedro Vieira Vilela

Prof. Dr. Cleber Valgas Gomes Mira (Orientador)

Dourados-MS  
Novembro de 2017



**Implementação de uma heurística gulosa para o problema de otimização de portfólios**

João Pedro Vieira Vilela

Novembro de 2017

**Banca Examinadora:**

Prof.º Dr. Cleber Valgas Gomes Mira (Orientador)  
Área de computação – UEMS

Prof.º Dr. Fabrício Sérgio de Paula  
Área de computação – UEMS

Prof.º Msc. André Chastel Lima  
Área de computação – UEMS



# **Implementação de uma heurística gulosa para o problema de otimização de portfólios**

João Pedro Vieira Vilela

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por João Pedro Vieira Vilela e aprovada pela Banca Examinadora, como parte dos requisitos a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 04 de novembro de 2017.

Prof.º. Dr. Cleber Valgas Gomes Mira  
(Orientador)



*“Ser sábio é melhor do que ser forte; o conhecimento é mais importante do que a força. Afinal, antes de entrar numa batalha, é preciso planejar bem, e, quando há muitos conselheiros, é mais fácil vencer. ”*

(Provérbios 24:5, 6)

## AGRADECIMENTOS

Primeiramente agradeço a Deus, pela proteção em minha vida, pelo amor que tem sobre mim, a ajuda para que possa continuar minha jornada, com sabedoria, clareza e discernimento perante as situações que a vida nos revela e pela saúde que me permite a vida.

À minha mãe, Tânia Jucilene Vieira Vilela, por ser exemplo de vida, pelo carinho e amor oferecidos a mim em toda minha trajetória até aqui, me apoiando e corrigindo desde o início, por sempre estar disposta a me ajudar em qualquer situação, e em qualquer momento, pelo apoio e orações mostrando-me a dedicação verdadeira e o amor incondicional.

Ao meu pai, Francisco Ferreira Vilela, que me mostrou o real propósito de ser pai, me apoiando e ajudando a realizar meus sonhos, me aconselhando e acolhendo nas horas certas e que humildade e honestidade são os principais valores que formam o caráter de um bom homem e um excelente pai. E ao meu irmão Samuel Alexandre Vieira Vilela, por fazer jus a palavra irmão.

À minha tia, Renata Consuelo Vieira Kovacs, por ser minha melhor amiga e companheira de todas as horas, por ensinar que se deve sempre estar com um sorriso no rosto independente das situações. Ao Giancarlo Kovacs pelo companheirismo e ajuda em vários momentos.

Agradecimento especial aos meus avôs, Ebe Bittencourt e Francisco Mariano Vieira, por todo amor que recebo, e ensinamentos que fizeram minhas bases mais sólidas.

À minha amada Patrícia Vargas Vieira, que sempre acreditou na minha capacidade e competência com todo amor e compreensão, me apoiando, incentivando e confiando que a realização desse trabalho seria possível, e por estar sempre presente em minha vida.

Ao meu orientador Cleber Valgas Gomes Mira, pela orientação competente e dedicada, pelo respeito e paciência, pelos conselhos didáticos, a fim de realizar esse trabalho da melhor forma possível, garantindo o sucesso do mesmo.

Por fim, a todos que de alguma forma contribuíram para a realização deste trabalho.



## RESUMO

Neste trabalho apresentamos e discutimos uma versão particular do problema de otimização de portfólios encontrado na indústria de geração de energia elétrica. Apresentamos também uma solução proposta em um trabalho anterior baseado em uma heurística gulosa. Esta heurística é fundamentada na meta-heurística GRASP. Também discutimos alguns exemplos de portfólios e comparamos seus valores de função objetivo. O objetivo deste trabalho é o desenvolvimento de uma aplicação que implemente a heurística fundamentada no algoritmo GRASP, realizar experimentos computacionais e analisar as soluções obtidas.

**Palavra-chave:** PPS, GRASP, heurísticas, implementação, agendamento de projetos

## **ABSTRACT**

In this paper we present and discuss a particular version of a portfolio optimization problem found in the power generation industry. We also present a solution proposed in a previous work based on a greedy heuristics. This heuristics is based on the GRASP meta-heuristic. We also discussed some examples of portfolios and compare their objective function values. The goal of this work is to develop an application that implements a heuristics based on the GRASP algorithm, conduct computational experiments, and analyze the obtained solutions.

**Keywords:** PPS, GRASP, heuristics, implementation, project scheduling

# Sumário

<b>1. Introdução</b> .....	19
<b>1.1. Motivação</b> .....	21
<b>1.2. Objetivo</b> .....	21
<b>1.3. Metodologia</b> .....	21
<b>1.4. Organização do texto</b> .....	21
<b>2. Descrição do Problema</b> .....	23
<b>2.1. Projetos</b> .....	24
<b>2.2. Pontos de atenção</b> .....	24
<b>2.3. Usinas de energia</b> .....	25
<b>2.4. Entradas do problema</b> .....	26
<b>3. Restrições</b> .....	29
<b>4. Função objetivo</b> .....	33
<b>5. Exemplo</b> .....	35
<b>6. Algoritmo GRASP</b> .....	39
<b>6.1. Fase de Construção</b> .....	40
<b>6.2. Busca local</b> .....	44
<b>7. Comparação entre portfólios</b> .....	47
<b>8. Experimentos</b> .....	51
<b>8.1. Escolha de Parâmetros</b> .....	51
<b>8.2. Definição das instâncias de entrada do problema</b> .....	52
<b>8.3. Resultados e Análises</b> .....	53
<b>9. Conclusão</b> .....	57
<b>9.1. Trabalhos futuros</b> .....	57
<b>Referências</b> .....	59
<b>10. Apêndice</b> .....	61



## Lista de Siglas

$b_{p,m}$	benefício do par $(p,m)$
C	projeto de manutenção de duração curta
CAPEX	recursos anuais para despesas de capital
c.c	caso contrario
$C_{P,t,q}$	recurso consumido na classe $q$ em um mês $t$ por um portfólio $P$
$C_{P,y,q}$	recurso consumido na classe $q$ em um ano $y$ por um portfólio $P$
D	conjunto de divisões em uma companhia de geração de energia
$d_p$	duração de um projeto
$D_w$	data limite associada a um ponto de atenção de alto risco
$E$	um exemplo de entrada do problema PPS
$e_p$	último mês de execução de um projeto
$f_w$	último mês do último projeto que controla um ponto de atenção
$G_w$	grupo de projetos que controlam um ponto de atenção
$g(u,m)$	número de unidades geradoras paradas em uma usina $u$ durante em mês $m$
$gL(u,m)$	número de unidades geradoras paradas em uma usina $u$ durante em mês $m$ para projetos de duração longa
HRSP	lista com os pares ordenados de alto risco
$I$	conjunto de projetos
$I_p$	conjunto de projetos agendados em um portfólio $P$
L	projeto de manutenção de tempo de duração longo
m	mês de agendamento de um projeto
$MD_w$	mínimo de todas as datas limites associadas a todos os pontos de atenção controlados por um projeto
$m_p$	mês inicial de um projeto em um portfólio
$m_y$	sequencia de meses do ano $y$
N	projeto que não envolve manutenção
[n]	intervalor 1, 2, 3, ..., n
OPEX	valores anuais para despesas operacionais
$P$	um portfólio
$p$	um projeto

PH	Plano de horizonte
$p_m$	mês prescrito de um projeto mandatório
$P_{t,q}$	conjunto de projetos da classe $q$ ativos em um mês $t$
$Q$	conjunto {CAPEX, OPEX}
$q$	classe de ativos em um mês
$R(P)$	área sob a curva do risco de um portfólio $P$
$R_p$	porção de risco que o projeto $p$ controla
$r_{p,m}$	recursos consumidos por um projeto durante sua execução
$R_w$	risco de um ponto de atenção
SP	lista de pares ordenados
$S_{y,q}$	recursos avaliadas por classe $q$ em um ano $y$
T	tamanho do horizonte de planejamento em meses
$t$	um mês qualquer
$U_d$	conjunto de usinas de energia contidas em $d$
$u$	usina de energia
$V_E$	conjunto de portfólios viáveis em uma entrada
$W$	conjunto de todos os pontos de atenção
$w$	um ponto de atenção
$W_P$	conjunto de pontos de atenção controlados por um portfólio $P$
$W_{i(p)}$	pontos de atenção intoleráveis resolvidos por um projeto $p$
$y$	um ano

## Lista de Figuras

Figura 1. Agendamento do portfólio P1.....	37
Figura 2. Curva de risco ao longo do horizonte de planejamento (MIRA et al., 2015). .....	38
Figura 3. Algoritmo do GRASP (MIRA et al., 2015). .....	40
Figura 4. Algoritmo de construção de portfólio (MIRA et al., 2015). .....	44
Figura 5. Algoritmo de busca local (MIRA et al., 2015). .....	45
Figura 6. Agendamento do portfólio P2.....	48
Figura 7. Curva de risco do horizonte de planejamento do portfólio P2 (MIRA et al., 2015). .....	49
Figura 8. Curva de risco do horizonte de planejamento da entrada 5. ....	54





## Lista de Tabelas

Tabela 1. Usinas X Divisão de Áreas X Unidades Geradoras. ....	28
Tabela 2. Projetos, Custos Mensais e Duração. ....	35
Tabela 3. Valores de Riscos dos Pontos de Atenção.....	36
Tabela 4. Valores de Recursos Anuais.....	36
Tabela 5. Grupo de Projetos de Controle de Pontos de Atenção.....	41
Tabela 6. Parâmetros da heurística.....	51
Tabela 7. Entradas dos experimentos.....	53
Tabela 8. Resultados dos experimentos. ....	53



## 1. Introdução

A evolução das tecnologias da informação causou um crescente fluxo de informações, devido ao aumento da capacidade de processamento e armazenamento de dados. Várias organizações vêm gerando mudanças nas suas gestões, para responder a esse montante de informações. Uma dessas mudanças envolve o processo de tomada de decisões dos gestores.

Qualquer organização tem a necessidade de tomar decisões. Há decisões que são mais simples, podendo ser executadas à mão e outras mais complexas, que necessitam de uma quantidade de dados maior e de um processo automatizado. Uma das áreas na qual a tomada de decisão é relevante consiste na escolha e agendamento de projetos a serem executados. Por exemplo, na indústria farmacêutica, mais especificamente na pesquisa de novos produtos, um desafio dos gestores é a escolha e alocação de projetos que produzam novos medicamentos segundo determinados critérios.

A dificuldade na gestão de planejamentos de projetos deu origem ao campo de investigação de otimização de portfólios de projetos. Decisões erradas na escolha do portfólio podem ter duas consequências negativas. A primeira, é gastar recursos com projetos inadequados, e a segunda, a organização perde os benefícios que ela poderia ter ganho se esses recursos tivessem sido gastos em projetos adequados (CARAZO et al., 2010).

O problema de construção de portfólio é chamado de Seleção de Portfólio de Projetos, ou PPS (Project Portfolio Selection). Um portfólio é uma seleção e um escalonamento de projetos sobre um determinado período de tempo para sua execução. A construção do portfólio envolve questões como possíveis conflitos, restrições ou montante de recursos (MIRA et al., 2015).

Embora exista uma variedade considerável de definições do PPS de acordo com o uso de restrições específicas, dependências entre projetos e suposições sobre o tempo, o problema PPS pode ser generalizado da seguinte forma: encontre uma instância de portfólio de projetos que maximiza uma dada função-objetivo sobre todos os portfólios viáveis. Se forem consideradas mais de uma função-objetivo, o problema torna-se conhecido como Problema de Seleção de Projetos Multi-objetivo (MPPS).

A versão PPS de única função-objetivo é um problema clássico de otimização. Exemplos onde esse problema é encontrado são aplicações financeiras (CAMPBELL J;

CAMPBELL R, 2001), pesquisa e desenvolvimento (HENRIKSEN; TRAYNOR, 1999), desenvolvimento de softwares (LEE; KIM, 2000) e inúmeros outros campos.

No caso de uma função de único objetivo, procuramos solucionar o problema PPS usando um único critério principal, seja ele economia de recursos, execução da maior quantidade de projetos em um período, ou realização dos projetos considerados de maior importância. As técnicas de resolução deste tipo de problema são, na maioria das vezes, fundamentadas em algoritmos de natureza gulosa, como por exemplo o Greedy Randomized Adaptive Search Procedure - GRASP, ou o Analytic Hierarchy Process, que é utilizado para decisões mais complexas.

Recentemente, vários autores vêm dando maior atenção a versão multi-objetivo do problema. Essa abordagem em questão foca em alguns fatores relevantes para o PPS como por exemplo: a maximização de valores e a minimização de risco (MIRA et al., 2015).

Há uma ampla variedade de métodos de resolução para a seleção de projetos. Dentre os mais utilizados estão o método de pontuação, a teoria da utilidade de multiatributo e o processo analítico e hierárquico. Estes métodos destinam-se a classificação do conjunto de projetos. Esta classificação indica quais projetos serão executados e em que ordem. Os recursos disponíveis para execução dos projetos serão distribuídos de acordo com as prioridades estabelecidas na classificação.

Esses métodos não são aplicáveis em situações com múltiplas restrições (por exemplo, recursos, restrições estratégicas ou políticas). Além disso, estas abordagens assumem que os projetos a serem escolhidos são independentes. Projetos independentes são aqueles que não precisam que outros sejam executados e finalizados para serem iniciados.

Mesmo que consideremos uma formulação do problema na qual os projetos são independentes, um ranqueamento dos melhores projetos segundo um critério específico não necessariamente produzirá o melhor portfólio possível.

Muitos trabalhos relacionados ao problema PPS fornecem soluções para simulações e dados gerados aleatoriamente. Em um caso real do problema PPS de única função-objetivo, as entradas podem conter até milhares de projetos em um grande período tempo. No nosso trabalho serão utilizadas entradas que foram geradas manualmente e com poucos projetos.

## **1.1. Motivação**

O tema de pesquisa surgiu por meio de um problema PPS para a gestão de riscos em uma empresa de geração de energia elétrica, onde os engenheiros dessa empresa necessitam da automatização do processo de tomada de decisões, pois ainda que envolvessem grandes entradas de dados, as escolhas dos projetos eram feitas à mão por uma equipe de especialistas.

Para encontrar soluções de qualidade razoável para o problema PPS usaremos uma heurística gulosa baseada no algoritmo GRASP, desenvolvida em um trabalho anterior (MIRA et al., 2015).

## **1.2. Objetivo**

O objetivo desse estudo em específico é implementar a heurística baseada do GRASP utilizando a linguagem de programação Python e realizar experimentos computacionais para analisar a qualidade das soluções encontradas.

O motivo da escolha da linguagem Python se deve ao fato de ela ser uma linguagem adequada para a prototipação de aplicativos e por focarmos neste trabalho na análise das soluções encontradas pela heurística, ao invés de focarmos na eficiência do algoritmo.

## **1.3. Metodologia**

O trabalho será desenvolvido por meio das seguintes etapas:

1. Revisão bibliográfica com foco no conteúdo de seleção de portfólios e no algoritmo GRASP;
2. Escrita e desenvolvimento do texto;
3. Desenvolvimento do protótipo com base na literatura proposta;
4. Realização de testes do protótipo considerando um cenário de uso;
5. Avaliação dos resultados obtidos.

## **1.4. Organização do texto**

Na seção 2 realizaremos uma explicação do problema PPS estudado neste trabalho. Na seção 3 apresentaremos as restrições. Na seção 4 discutimos a função objetivo. Na seção 5 apresentamos um exemplo de um portfólio para uma entrada do problema. Na seção 6 apresentamos um algoritmo guloso para resolver o problema PPS deste trabalho. Na seção 7 apresentamos um novo exemplo de um portfólio melhor, comparando-o ao apresentado na seção 5. Na seção 8 são apresentados os resultados e as análises dos experimentos realizados. Na seção 9 é dada a conclusão deste trabalho.



## 2. Descrição do Problema

Existem inúmeras formulações variantes para o problema PPS com uma função específica, por exemplo, envolvendo diferentes restrições, variáveis de decisão, políticas de disponibilidade de recursos e suposições de tempo. Entretanto, essas variações normalmente compartilham algumas características em comum. Um problema PPS pode ser descrito da seguinte maneira: selecionar e criar um cronograma de projetos ao longo de períodos de tempo, que possa maximizar uma função-objetivo sobre todos os portfólios viáveis (CARAZO et al., 2010). As condições gerais envolvidas no problema PPS são (MIRA et al., 2012):

- Uma sequência de períodos de tempo;
- Uma atribuição de valores de recursos disponíveis para cada período de tempo;
- Um conjunto de projetos;
- Uma função que mede o quão benéfico é um projeto executado;
- Uma função de custo que descreve uma demanda do projeto para recursos ao longo de sua duração;
- Um conjunto de restrições que representam diferentes propriedades do problema, tal como limitação de recursos, as interdependências entre projetos e restrições de agendamento; e
- Uma função objetivo que mede o quão benéfico é um portfólio.

Neste trabalho descrevemos uma solução para uma versão específica de PPS que ocorre em uma empresa geradora de energia brasileira. Essa organização é constantemente confrontada com desafios a seleção e programação de projetos que auxiliam a operação da empresa. Geralmente estes desafios são:

- Cumprir regulamentos governamentais;
- Minimizar exposição de riscos;
- Aumentar o retorno sobre os investimentos;
- Reduzir os custos de operação.

## 2.1. Projetos

Nesta empresa existem dois diferentes tipos de projetos:

- Projetos de gestão de Riscos: são responsáveis por prever, evitar e controlar possíveis problemas que possam surgir durante o processo de geração de energia. Esses projetos influenciam em atividades como a aquisição de novos equipamentos, o reparo de equipamentos e até mesmo melhoria no processo de manutenção. Projetos desta categoria podem ser escalonados com uma certa liberdade, porém devem seguir algumas restrições.
- Projetos de gestão de Não-Risco: não previnem e nem controlam riscos. Eles estão relacionados à, por exemplo, compra de suprimentos de escritório, equipamentos de segurança, uniformes de empregados, e outros. Esse tipo de projeto também auxilia no crescimento estratégico da empresa, como a construção de uma nova unidade geradora de energia, ou atendem exigências tais como, legislação ambiental, capacidade de produção e disponibilidade de energia. Devido a essa natureza, projetos desta categoria não podem ser escalonados, mas sim, têm seu agendamento pré-definido pela gestão estratégica da empresa (MIRA et al., 2015).

Os projetos que serão selecionados para construção do portfólio devem iniciar em meses específicos em um período de tempo denominado Plano de Horizonte (PH). A extensão do Plano de Horizonte, em nosso caso de estudo é de 60 meses. Este período de tempo é representado pela letra  $T$  (MIRA et al., 2015).

Projetos de gestão de Não-Risco são considerados obrigatórios, ou seja, os projetos devem ser executados exatamente no mês inicial já pré-definido no PH. Já os projetos que gerenciam riscos passam por processo de tomada de decisão para se escolher seu mês de início, onde são analisados fatores tais como, custo, riscos em potencial e interdependências (MIRA et al., 2015).

## 2.2. Pontos de atenção

A operação de uma empresa de geração de energia é um processo complexo no qual pode ocorrer eventos imprevisíveis, como falha de equipamentos ou até mesmo acidentes. Portanto, a prevenção contra essas eventualidades é de extrema prioridade e, para que essa prevenção seja efetiva, existem os seguintes passos a serem seguidos:



- O primeiro passo é levantar os pontos de atenção. Um ponto de atenção é uma breve descrição de condições específicas que podem gerar um evento indesejado. Pontos de atenção são usados pelos especialistas para criarem medidas para prever essas ocorrências.
- O próximo passo é atribuir uma escala de risco para os pontos de atenção. A escala de risco contém a probabilidade de ocorrência dos eventos indesejados, e a gravidade do evento indesejado acontecer.
- O terceiro passo consiste em conceber projetos que controlam os pontos de atenção, a fim de evitar ou amenizar o impacto dos eventos indesejados na operação da empresa.

Existem dois tipos de categorias para pontos de atenção: crítica e não-crítica. Os riscos classificados como intoleráveis estão dentro da categoria crítica. Já aqueles que são toleráveis ou moderados são categorizados como não-críticos. As categorias de riscos auxiliam na criação de medidas de controle sobre os pontos de atenção, mas o que geralmente ocorre são grupos de projetos controlando riscos. Os pontos de atenção considerados de alto risco têm um prazo limite para serem controlados.

Em nosso problema de PPS nos concentramos no controle de risco atribuído aos pontos de atenção, pois os riscos só podem ser considerados evitados por completo somente após todos os projetos do grupo controlador serem concluídos (MIRA et al., 2015).

### **2.3. Usinas de energia**

O processo de geração de energia é organizado conforme a localização geográfica e divisões administrativas. Essa organização é refletida nas restrições do problema (MIRA et al., 2015).

As usinas são afetadas por fatores climáticos da região ou por alguma falha que exija reparo técnico. Esses processos de reparo podem afetar a geração de energia devido às restrições que impõem. Por exemplo, caso uma unidade geradora de energia entre em processo de manutenção, para não haver redução da capacidade de geração de energia, uma outra unidade da mesma usina deve continuar operante. Esse tipo de restrição limita a possibilidade de execução de projetos de manutenção simultâneos em unidades geradoras diferentes da mesma usina.

A impossibilidade de realizar determinados projetos de maneira simultânea causa uma interdependência entre projetos. Apenas os projetos de manutenção que interrompem a geração de energia possuem interdependências.

#### **2.4. Entradas do problema**

Na área de ciência da computação, o problema PPS é considerado um problema NP-difícil (DOERNER et al., 2004). Devido a esta dificuldade, abordamos neste trabalho uma heurística de fácil implementação, descrita em (MIRA et al., 2015).

Para a resolução do problema são necessárias as seguintes informações: os projetos com seus custos, duração e grupos de controle aos quais pertencem; os pontos de atenção com seus riscos; o plano de horizonte e os recursos disponíveis anualmente.

O plano de horizonte possui uma extensão de 60 meses, no entanto o tempo esperado para execução do portfólio é de 120 meses, 10 anos, no qual os primeiros cinco anos são referentes ao PH, enquanto os últimos anos são usados para acomodar os projetos que, embora tenham sido iniciados antes do prazo final do PH, não foram concluídos e precisam de acompanhamento. O tempo esperado para execução dos projetos é chamado de horizonte de execução (HE). Com a duplicação do PH pode-se medir quanto um projeto que termina após o término de PH pode contribuir para reduzir os riscos dentro do portfólio.

Os recursos que constam no PH são distribuídos entre duas classes: despesas de capital e despesas operacionais. As disponibilidades dos recursos são apresentadas em duas séries de 5 montantes anuais. A primeira série designa disponibilidade de recursos anuais para as despesas de capital (CAPEX) e a segunda representa os valores anuais para as despesas operacionais (OPEX).

Os parâmetros de entrada de um projeto são os seguintes (MIRA et al., 2015):

- Duração da manutenção: projetos para reparos são classificados de duas formas, (C) se for um tempo curto, (L) se for longo, mas se for um projeto que não envolve manutenção é indicado com (N);
- A Área: identificar a unidade administrativa ligada ao projeto;
- Identificação da Unidade Geradora: identificação da unidade geradora na qual o projeto de manutenção será executado. Um projeto de manutenção afeta apenas uma unidade geradora;

- Mês de início da parada: para projetos de manutenção, o mês de parada indica qual mês, contando a partir do primeiro mês de execução, o projeto irá parar a unidade geradora correspondente;
- Duração da parada: informa, em meses, por quanto tempo a unidade geradora estará em processo de manutenção, ou seja, parada;
- Mês inicial pré-definido: O conjunto de todos os projetos, junto com seus respectivos meses iniciais, formam o portfólio, denominado de portfólio inicial. Para projetos mandatórios, é sugerido um mês inicial na entrada do problema;
- O tipo de projeto, que pode ser de gestão de risco ou gestão de não-risco. É usado para indicar se um projeto é mandatório;
- A classificação dos recursos: CAPEX e OPEX;
- O custo do projeto, ou seja, a quantidade de recurso que será gasta em cada mês de execução do projeto.

Assumimos que os projetos sempre executam dentro do tempo de sua duração e são bem-sucedidos. Em um cenário real os problemas de otimização de portfólios são mais complexos e podem envolver incertezas. Por exemplo, os projetos poderiam terminar sua execução após o previsto devido a eventos inesperados, ou mesmo falharem completamente (LIESIÖ; MILD; SALO, 2008).

Pontos de atenção são definidos pelos seguintes parâmetros, (MIRA et al., 2015):

- Um identificador: um número inteiro que não se repete;
- Medida de risco: um número racional que indica o grau do risco associado ao ponto de atenção;
- Um grupo de projetos: é uma lista de projetos identificados que define grupo de projetos que controla o ponto de atenção. Os projetos de não gestão de risco não pertencem a nenhum grupo, logo não contribuem para controle de nenhum risco;
- Categoria de riscos: classificados em dois grupos: crítico e não-crítico;
- Data limite: é o último mês no qual um ponto de atenção deve ser controlado.

O sistema de geração de energia dessa empresa consiste de 12 usinas, na qual cada usina deve seguir normas específicas, de acordo com a localização geográfica e divisões administrativas, para um bom funcionamento.

Cada usina fornece os seguintes parâmetros, (MIRA et al., 2015):

- Um identificador da usina: é um número inteiro distinto;
- Unidades geradoras: a quantidade de unidades geradoras na usina;
- Divisão: a divisão responsável pela gestão da usina;
- Localização geográfica: local residente da usina.

Tabela 1. Usinas X Divisão de Áreas X Unidades Geradoras.

<b>Divisão</b>	<b>Área/Usina</b>	<b>Localização</b>	<b>Unidade geradoras</b>
COG	COG	-	-
IT	IT	-	-
AGV	AGV	AV	6
PRO	NAV	NAPI	3
	PRO	NAPI	3
	BAB	BBB	4
BAR	BAR	BBB	3
	IBI	NAPI	3
	CAC	RP	2
	EUC	RP	4
LMO	LMO	RP	2
	MOG	RMG	2
	SJS	RJM	2
	SJQ	RJM	1

### 3. Restrições

Seja  $I$  o conjunto de projetos da entrada do problema,  $W$  como o conjunto de pontos de atenção do problema e temos  $T$  indicando, em meses, o plano de horizonte. Para cada projeto  $p \in I$ , será indicado a duração do projeto, em meses, por  $d_p$ .

Um portfólio  $P$  é um conjunto de pares  $(p, m)$ , onde  $p$  é um projeto que pertence a  $I$  e  $m$  é o mês no qual o projeto é escalonado para execução, isto é,  $P \subseteq I \times [T]$ . Seja  $I_p \subseteq I$  o conjunto de projetos escalonados no portfólio, ou seja,  $m \in [T]$ .

Um portfólio só é válido se segue as seguintes restrições (MIRA et al., 2015):

**Agendamento consistente.** Um projeto não pode ser agendado para começar em dois meses diferentes.

$$\text{Se } (p, m_1) \in P \text{ e } (p, m_2) \in P, \text{ então } m_1 = m_2, \text{ para todo } p \in I_p. \quad (1)$$

Isto significa que um portfólio válido é na verdade uma função  $P: I_p \rightarrow [T]$ .

**Projetos obrigatórios.** Um projeto dessa categoria sempre deve iniciar no mês prescrito,  $p_m$ . Logo,

$$(p, p_m) \in P, \text{ para todo projeto obrigatório } p \in I. \quad (2)$$

**Pontos de atenção intoleráveis.** Um ponto de atenção de alto risco é considerado como intolerável e deve ser totalmente controlado até uma data limite. Isso significa que todo projeto no grupo de controle,  $G_w \subseteq I$  deve ser completado até a data limite, onde  $w$  representa o ponto de atenção intolerável. Seja  $W$  o conjunto de pontos de atenção, então:

$$G_w \subseteq I_p, \text{ para todo ponto de atenção intolerável } w \in W. \quad (3)$$

Definimos  $f_w$  como o mês mais tardio que um projeto  $p \in G_w$  termina sua execução. Temos  $m_p$  como o mês inicial do projeto e  $d_p$  como a duração do projeto.

$$f_w = \max_{p \in G_w} (m_p + d_p - 1) \quad (4)$$

Logo temos a seguinte restrição:

$$f_w \leq D_w, \text{ para todo ponto de atenção intolerável } w \in W, \quad (5)$$

onde  $D_w$  significa o prazo da data limite. Ou seja, o ponto de atenção  $w$  deve ser controlado antes de sua data limite.

**Recursos limitados.** Cada projeto tem um valor de custo associado desde o início da execução até o último mês. O total de recursos consumidos de uma das categorias, CAPEX e OPEX, contanto todos os projetos que estão em execução dentro de um ano, não devem ultrapassar o montante anual de recursos disponíveis para aquele mesmo ano.

O conjunto de recursos das classificações CAPEX e OPEX é dado por  $Q$ . Para todo projeto  $p \in I$ , temos  $r_{p,m}$  o consumo de recursos de um projeto  $p$  durante o  $m$ -ésimo mês de execução ( $1 \leq m \leq d_p$ ). Para um portfólio  $P$ , temos em  $P_{t,q}$  o conjunto dos projetos das categorias  $q \in Q$  que são ativos no mês  $t \in [2T]$ , isto é,  $P_{t,q} = \{ p \in I_p / t \in [m_p, e_p] \}$ , onde  $m_p$  é o mês inicial do projeto e  $e_p$  é o último mês de execução do projeto, então

$$C_{P,t,q} = \sum_{p \in P_{t,q}} r_{p,t-m_p+1}, \quad t \in [2T], q \in Q, P \text{ um portfólio} \quad (6)$$

Representa o consumo de recursos no mês  $t$  para todos os projetos ativos de uma classe  $q$  agendados no portfólio  $P$ . É possível coletar o consumo dos recursos de uma classe  $q$  em um ano  $y$  para um portfólio  $P$  com a seguinte expressão:

$$C_{P,y,q} = \sum_{t \in M_y} C_{P,t,q}, \quad y \in \left[1, \frac{T}{12}\right], q \in Q, \quad (7)$$

$$M_y = [12y - 11, 12y], P \text{ um portfólio.}$$

Então, se  $S_{y,q}$  é o montante de recursos de uma classe  $q$  dentro de um ano  $y$ , a restrição de recursos limitados impõe que:

$$C_{P,y,q} < S_{y,q} \text{ para todo } y \in \left[1, \frac{T}{12}\right], q \in Q, P \text{ um portfólio.} \quad (8)$$

**Restrições de parada por localização geográfica.** Seja  $g(u, m)$  o número de unidades geradoras paradas em uma usina  $u$  em um mês  $m$  em um dado portfólio, enquanto  $g_L(u, m)$  é o número de unidades geradoras interrompidas em uma usina  $u$  em um mês  $m$  devido a projetos de manutenção de longa duração.

1. Na localização RP, em um dado mês, se uma usina tem duas ou mais unidades geradoras paradas, as outras usinas de energia não podem ter nenhuma unidade geradora parada, isto é:

$$g(u_1, m) \geq 2 \rightarrow g(u_2, m) = 0, \quad (9)$$

para  $m \in [2T]$  e  $u_1 u_2 \in \{CAC, EUC, LMO\}$ , com  $u_1 \neq u_2$

2. Na localização BBB tem de ser observado que:

- Em um determinado mês, se uma usina tem duas ou mais unidades geradoras paradas, uma outra usina não pode ter nenhuma unidade geradora parada. Assim,

$$g(u_1, m) \geq 2 \rightarrow g(u_2, m) = 0, \quad (10)$$

para  $m \in [2T]$  e  $u_1 u_2 \in \{BAB, BAR\}$ , com  $u_1 \neq u_2$

- Em qualquer mês, no máximo duas unidades geradoras da usina BAR podem ser interrompidas. Assim,

$$g(BAR, m) \leq 2, \text{ para } m \in [2T] \quad (11)$$

- Em um dado mês, duas unidades geradoras da usina de energia BAB podem ser desligadas simultaneamente, mas apenas se nenhuma unidade geradora de BAR está parada. Assim,

$$g(BAB, m) = 2 \rightarrow g(BAR, m) = 0, \text{ para } m \in [2T] \quad (12)$$

3. Na localidade NAPI, duas unidades geradoras podem ser interrompidas em qualquer mês. Assim,

$$g(NAV, m) + g(PRO, m) + g(ABI, m) \leq 2, \text{ para } m \in [2T]. \quad (13)$$

4. Em um dado mês, na localidade de AV, pode se ter duas unidades geradoras paradas devido a projetos de manutenção de longa duração. Logo,

$$g_L(AGV, m) \leq 2, \text{ para } m \in [2T]. \quad (14)$$

**Restrição de paradas na usina AGV.** Dado um mês, se houver duas ou mais unidades geradoras da usina de energia AGV forem interrompidas para procedimento de manutenção, então nenhuma outra unidade geradora das usinas BAB, NAV ou PRO, devem ser interrompidas, isto é,

$$(g(AGV, m) \geq 2) \rightarrow (g(BAB, m) = g(NAV, m) = g(PRO, m) = 0), m \in [2T] \quad (15)$$

**Restrições de paradas por divisões.** Em qualquer mês, não mais do que três unidades geradoras em uma mesma divisão podem ser interrompidas, seja  $D$  o conjunto de divisões em uma companhia de geração de energia. Para cada divisão  $d \in D$ , temos  $U_d$  o conjunto de usinas de energia contidas em  $d$ . Temos,

$$\sum_{u \in U_d} g(u, m) \leq 3, \text{ para } m \in [2T], d \in D \quad (16)$$

Se nenhuma restrição entrar em vigor, pode-se interromper uma unidade geradora de cada usina, BAB, BAR e IBI na divisão BAR.



#### 4. Função objetivo

Um problema PPS pode ter inúmeros objetivos de acordo com o foco do gestor de portfólios. Geralmente em uma equipe de tomada de decisão, o objetivo é reduzir os riscos que são causados pelos pontos de atenção que ainda não foram controlados. Por este motivo, na montagem do portfólio, há uma grande preferência por projetos de gestão de risco.

Ao se avançar através do horizonte de execução, os pontos de atenção vão sendo controlados na medida em que os projetos são concluídos. Assim, decresce o montante de risco até se atingir um valor mínimo.

O controle dos riscos é dado em um gráfico de risco X tempo. A área sob a curva deste gráfico apresenta dois dos mais importantes aspectos de um portfólio: a quantidade de riscos controlados e quando os riscos foram evitados. Isto significa que quanto maior for o risco evitado e quanto mais cedo um ponto de atenção for controlado melhor é o portfólio.

Seja  $W$  o conjunto de pontos de atenção. Temos  $R_w$  o risco associado ao ponto de atenção  $w \in W$ . Seja  $W_p \subseteq W$  o conjunto dos pontos de atenção controlados em  $P$ , então a área sob a curva de risco é:

$$R(P) = 2T \sum_{w \in W} R_w - \sum_{w \in W_p} R_w (2T - f_w) \quad (17)$$

Onde  $f_w$  é o primeiro mês que um ponto de atenção  $w$  é controlado.

Para uma entrada  $E$  qualquer, temos  $V_E$  o conjunto de portfólios viáveis obtidos em  $E$ . o objetivo é construir um portfólio  $P^*$  cujo  $R(P^*)$  é o **mínimo** entre  $\{R(P) | P \in V_E\}$ .



## 5. Exemplo

Dada uma instância do problema de otimização de portfólios, desejamos encontrar uma solução viável que otimize (nesse caso, minimize) o valor da função-objetivo. Nesta seção discutiremos um exemplo de uma instância desse problema de otimização e analisaremos a viabilidade e o valor de função-objetivo de duas soluções diferentes. Para melhor entendimento, a Tabela 2, abaixo, apresenta um exemplo contendo cinco projetos fictícios:

Tabela 2. Projetos, Custos Mensais e Duração.

Mês	p1	p2	p3	p4	p5
1°	R\$ 200	R\$ 110	R\$ 225	R\$ 125	R\$ 150
2°	R\$ 50	R\$ 280	R\$ 145	R\$ 135	R\$ 100
3°	R\$ 100	R\$ 120	R\$ 125	R\$ 80	R\$ 150
4°	R\$ 80	R\$ 120	R\$ 150	R\$ 80	R\$ 100
5°	R\$ 130		R\$ 50		R\$ 150
6°	R\$ 80		R\$ 15		R\$ 100
7°	R\$ 80		R\$ 30		R\$ 150
8°			R\$ 50		R\$ 100
9°					R\$ 150
10°					R\$ 100
11°					R\$ 150
12°					R\$ 100
Total	R\$ 720	R\$ 630	R\$ 790	R\$ 420	R\$ 1.500

A Tabela 2 representa o conjunto de projetos da instância de entrada. Consideramos 5 projetos:  $p_1$ ,  $p_2$ ,  $p_3$ ,  $p_4$  e  $p_5$ ; distribuídos entre as colunas 2 e 6 da tabela. A primeira coluna da tabela (Mês) indica o mês de execução do projeto. Para cada mês de execução e projeto, temos o custo de execução do projeto neste mês. Os valores de custos são apresentados em milhares de reais (R\$). Por exemplo, o projeto  $p_2$  consome 110, 280 e 120 milhares de reais em seu primeiro, segundo e terceiro meses de execução, respectivamente. A Tabela 2 também descreve a duração dos projetos. O projeto  $p_2$  dura 4 meses, enquanto o projeto  $p_1$  dura 7 meses. Recordamos que a duração de um projeto é um parâmetro de entrada do problema que não é alterado. A última linha da tabela indica os custos totais de cada projeto.

A Tabela 3 apresenta os dados do conjunto de pontos de atenção. Os projetos  $p_1$  e  $p_2$  estão contidos no grupo de controle do ponto de atenção  $w_1$ , já os projetos  $p_3$  e  $p_4$  fazem parte do grupo que controla o ponto de atenção  $w_2$  e o projeto  $p_5$  faz o controle do ponto de atenção  $w_3$ .

Tabela 3. Valores de Riscos dos Pontos de Atenção.

<b>Pontos de Atenção</b>	<b>Valor de Risco</b>	<b>Grupo de Controle</b>
<b>w<sub>1</sub></b>	50	$p_1$ e $p_2$
<b>w<sub>2</sub></b>	80	$p_3$ e $p_4$
<b>w<sub>3</sub></b>	100	$p_5$

A Tabela 3 mostra os valores dos riscos dos pontos de atenção que serão controlados pelos projetos da Tabela 2. Em um cenário real os riscos possuem um valor adimensional que geralmente é obtido por um especialista de gestão de risco. Os valores de riscos representados na Tabela 3 são fictícios.

Tabela 4. Valores de Recursos Anuais.

<b>Ano</b>	<b>Valor (mil R\$)</b>
<b>1</b>	650
<b>2</b>	700
<b>3</b>	1.400
<b>4</b>	650
<b>5</b>	850

Utilizando os dados das Tabelas 2 e 4 é possível construir alguns portfólios, alguns válidos e outros não. A Figura 1 representa um exemplo de um portfólio válido. Observe que como os projetos considerados na instância de entrada não são de manutenção, esse portfólio não viola as restrições discutidas na Seção 4. Os recursos apresentados neste exemplo são do tipo OPEX.

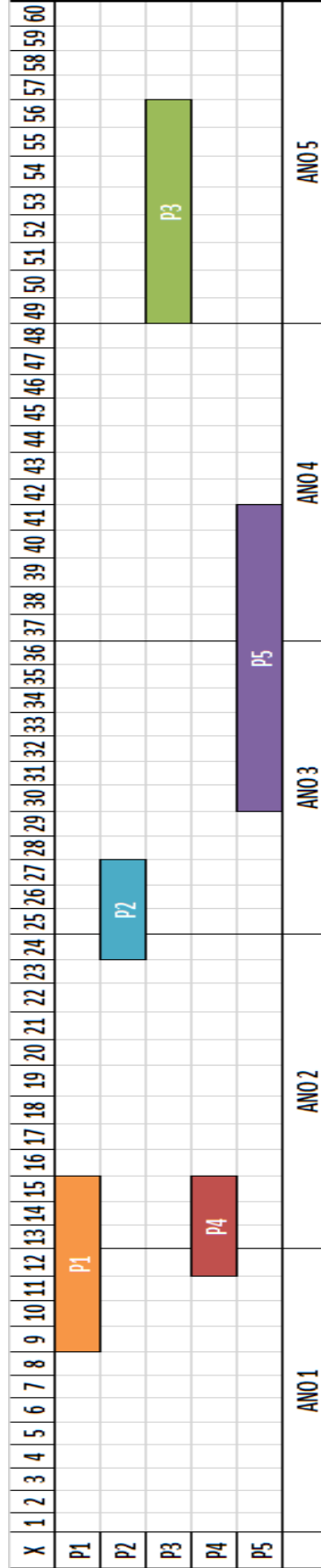


Figura 1. Agendamento do portfólio P1.

O portfólio mostrado na Figura 1 é válido, pois:

- O total (anual) de custos dos projetos não ultrapassa os valores de recursos anuais;
- O escalonamento dos projetos não viola as restrições de paradas de unidades geradoras.

A Figura 2 representa a curva do gráfico de Risco X Tempo para o portfólio P1.

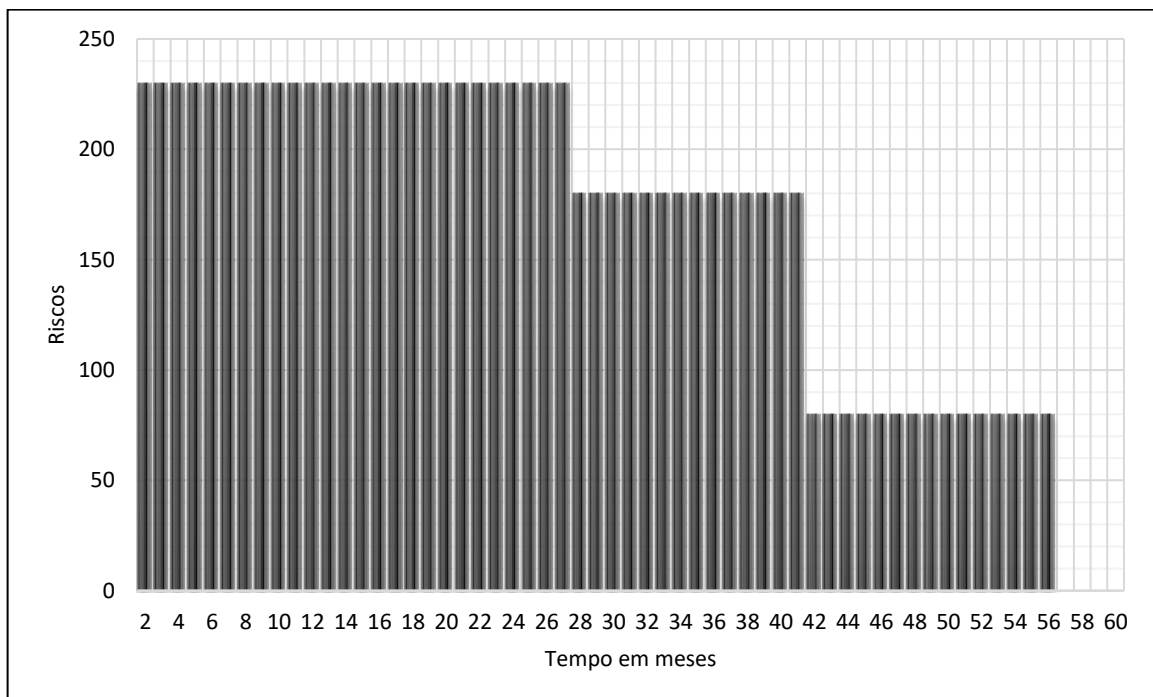


Figura 2. Curva de risco ao longo do horizonte de planejamento (MIRA et al., 2015).

Os valores dos riscos vão diminuindo conforme a execução de todos os projetos de um grupo de controle é concluída. Um ponto de atenção só pode ser considerado controlado quando todos os projetos que estão no grupo de controle associado a ele terminam sua execução. Por exemplo o ponto de atenção  $w_2$ , só foi de fato controlado no mês de execução 56, quando o último projeto,  $p_3$ , do grupo de controle foi finalizado.

## 6. Algoritmo GRASP

Apresentamos nessa seção uma variante da meta-heurística gulosa GRASP que será utilizada para encontrar soluções para o problema PPS.

O GRASP é tanto um processo multi-start quanto um processo iterativo, onde cada iteração dar-se-á por duas fases: uma fase de construção cujo, objetivo é criar uma solução possível e uma fase de busca local, onde buscas são realizadas na vizinhança da solução corrente até que uma melhor solução, potencialmente a ótima, seja encontrada no local. A melhor solução encontrada é mantida como resultado (FESTA; RESENDE, 2011). O GRASP pode gerar boas soluções para vários problemas de otimização por ser de fácil compreensão e implementação.

Inicialmente, durante a fase de construção, várias soluções são construídas e verificadas quanto sua viabilidade. As soluções construídas nesta fase são guardadas em um conjunto de portfólios viáveis. Após a fase de construção, é dado início à fase de busca local. Nesta fase o procedimento de busca local é aplicado por um repetido número de vezes, sobre as melhores soluções do conjunto de portfólios viáveis. Cada busca local realiza uma procura por soluções que sejam melhores que a solução corrente em sua vizinhança. Se uma solução vizinha tiver um valor melhor do que a solução corrente, então aquela se torna a nova solução corrente sob análise da busca local (MIRA et al., 2015).

```

1 Algorithm1 "GRASP para o problema PPS "
2
3   Inicialização de um portfólio vazio com um BEST_SOLUTION
4
5   for R iterações do
6     // Fase de Construção
7     Inicialização de um conjunto de soluções S
8
9     while (|S| < C) do
10
11        O procedimento de construção retorna uma solução P para o problema PPS
12
13        if P satisfaz a restrição de ponto de atenção intolerável then
14          P é incluído no conjunto de soluções S
15        end if
16    end while
17
18    //Fase de Busca Local
19    for cada solução P que é a melhor dentro de S do
20      Let = LOCAL_SEARCH(P)
21      if P é melhor que BEST_SOLUTION then
22        BEST_SOLUTION = P
23      end if
24    end for
25  end for
26  return BEST_SOLUTION

```

Figura 3. Algoritmo do GRASP (MIRA et al., 2015).

### 6.1. Fase de Construção

Esta primeira etapa consiste na execução repetida de um procedimento de construção. O processo de construção gera uma solução e confere se a solução é válida. As restrições que tornam a solução válida exigem que os custos dos projetos não ultrapassem os recursos disponíveis, que projetos de manutenção obedeçam às regras de parada de unidade geradora e projetos que controlam pontos de atenção intoleráveis terminem a sua execução antes da data limite.

Se um portfólio for válido então é incluído em um conjunto de soluções válidas, caso contrário, é descartado. As soluções válidas são inseridas neste conjunto até uma capacidade  $C$  ser atingida. Uma mesma solução pode ser inserida neste conjunto mais de uma vez (MIRA et al., 2015).



O comportamento desse processo de construção é baseado em uma ideia simples usando duas listas de pares  $(p, m)$  de projetos  $p \in I_p$  e potenciais meses de início  $m \in [1, \min(MD_w - d_p + 1, T)]$ , onde  $MD_w$  é o mínimo de todas as datas limites associadas a todos pontos de atenção intoleráveis controlados do projeto  $p$  (MIRA et al., 2015).

Seja  $W_{i(p)}$  os pontos de atenção intoleráveis resolvidos pelo projeto  $p$ .

$$MD_w = \begin{cases} \min_{w \in W_{i(p)}} D_w, & \text{onde } p \text{ resolve um } w \text{ que é um ponto de atenção intolerável} \\ \infty, & \text{c.c} \end{cases}$$

Observe que se um projeto  $p \in I_p$  não controla nenhum ponto de atenção intolerável, consideramos todos os pares  $(p, m)$  com  $m \in [T]$ .

Cada uma das duas listas de pares  $(p, m)$  é ordenada segundo um parâmetro chave. Esse parâmetro chave associado ao par  $(p, m)$  deve medir os ganhos obtidos pelo agendamento do projeto  $p$  no mês  $m$ , não se baseando apenas nos riscos que serão controlados pelo projeto, mas também no custo do projeto (MIRA et al., 2015).

Seja  $I$  o conjunto de projeto de uma entrada do problema com  $T$  tempo de execução e conjunto de pontos de atenção  $W$ . O custo benefício dos pares  $(p, m) \in I \times [T]$ , é denotado por  $b_{p,m}$ , denominado benefício do par  $(p, m)$ . Seja  $R_p$  a soma das porções de riscos de todos os pontos de atenção que o projeto  $p$  controla, para todos os pontos de atenção. Para esclarecer o conceito de porção de risco controlado por um projeto, suponha, por exemplo, que temos a seguinte tabela com os pontos de atenção, valores de risco e grupos de controle:

Tabela 5. Grupo de Projetos de Controle de Pontos de Atenção.

Pontos de Atenção	Valor de Risco	Grupo de Controle
w <sub>1</sub>	50	p <sub>1</sub> e p <sub>2</sub>
w <sub>2</sub>	80	p <sub>2</sub> e p <sub>3</sub>
w <sub>3</sub>	100	p <sub>4</sub>

O projeto  $p_1$  controla a seguinte porção de risco:

$$R_{P_1} = \frac{R_{W_1}}{|G_{W_1}|} = \frac{50}{2} = 25$$

Já  $p_2$  controla esta porção:

$$R_{P_2} = \frac{R_{W_1}}{|G_{W_1}|} + \frac{R_{W_2}}{|G_{W_2}|} = \frac{50}{2} + \frac{80}{2} = 25 + 40 = 65$$

O projeto  $p_3$  tem a seguinte porção:

$$R_{P_3} = \frac{R_{W_2}}{|G_{W_2}|} = \frac{80}{2} = 40$$

O projeto  $p_4$  controla esta porção:

$$R_{P_4} = \frac{R_{W_3}}{|G_{W_3}|} = \frac{100}{1} = 100$$

Sabendo que  $r_{p,t}$  retorna os recursos consumidos por um projeto  $p$  durante o  $t$ -ésimo mês de execução nós temos:

$$b_{p,m} = R_p \left( \frac{(2T - m - d_p + 1)}{\sum_{z=1}^{d_p} r_{p,m}} \right) \quad (18)$$

Então o  $b_{p,m}$  mostra o montante de riscos que o projeto  $p$  ajuda a controlar, ponderado pelo tempo que a contribuição é eficaz por unidade de custo (MIRA et al., 2015).

Tendo como base as Tabelas 2 e 3 e a Figura 1, é possível calcular o benefício de cada projeto usando a Eq. (18). Consideramos o projeto  $p_1$  para demonstração da função.

No portfólio apresentado,  $p_1$  inicia no mês 9 e tem o seguinte resultado de benefício:

$$b_{p_1,9} = \frac{25(120 - 9 - 7 + 1)}{720,000} = \frac{(25 \times 105)}{720,000} = \frac{2625}{720,000} = 0,003645833333$$

Supondo que  $P_1$  começasse no mês 24 o resultado da função benefício seria o seguinte:

$$b_{p_1,24} = \frac{25(120 - 24 - 7 + 1)}{720,000} = \frac{(25 \times 90)}{720,000} = \frac{2250}{720,000} = 0,003125$$

O valor da função de benefício de  $b_{p_1,9}$  é maior do que  $b_{p_1,24}$ , pois a porção de risco é controlada mais cedo quando  $p_1$  é agendado para ser iniciado no mês 9. Porém, note que, ao realocar o projeto  $p_1$  para que seu mês inicial seja o 24, então a restrição de montante de recursos, Eq. (8), não seria respeitada, sendo assim, seria um portfólio inválido.

O processo de construção do portfólio faz um pré-processamento da entrada para o agendamento de todos os projetos mandatórios em  $I$ . Além disso, o pré-processamento remove os custos dos projetos mandatórios dos montantes de recursos anuais. Feito isso, todos os pares  $(p, m) \in I \times [T]$ , têm seus benefícios calculados, e as duas listas de pares são construídas e ordenadas em ordem não crescente de benefício. A lista com pares ordenados de alto risco ( $HRSP$ ) contém os projetos que controlam pontos de atenção de altos riscos. A lista de pares ordenados ( $SP$ ) contém todos os pares remanescentes de  $I \times [T]$  (MIRA et al., 2015).

A construção de um portfólio é realizada através de escolhas iterativas de pares  $(p, m)$  a partir de uma das duas listas. O próximo par é escolhido da  $HRSP$  ou da  $SP$  com probabilidade de  $\eta$  e  $1 - \eta$  respectivamente. Uma vez que a lista para escolha do próximo par é determinada, o próximo par é aleatoriamente escolhido dentre os  $k$  primeiros desta lista. Então o par  $(p, m)$  é incluído no portfólio que está sendo construído, se não violar as restrições de agendamento consistente, Eq. (1); não ultrapassar os limites de recursos disponíveis, Eq. (8); e respeitar as restrições de parada, Eq. (9-16) (MIRA et al., 2015). Observe que só é possível verificar que um portfólio é viável de acordo com a restrição de pontos de atenção intoleráveis quando todos os projetos forem agendados.

```

1 Algorithm2 "Procedimento de construção para o problema PPS "
2
3 | Inicialização de um portfólio P com um conjunto vazio
4
5 | for todo projeto obrigatório do
6
7 |     Agendar cada projeto obrigatório em seu mês inicial pré definido
8 |     Subtrair os recursos dos projetos obrigatórios a partir dos recursos disponíveis.
9 | end for
10
11 | Calcular os riscos de cada projeto restante.
12 | Construir a lista dos Pares Ordenados de Alto Risco (HRSP)
13 | usando projetos de alto risco X pares de meses, ordenados por valor de benefício.
14 | Construir a lista dos Pares Ordenados (SP)
15 | usando os projetos restantes X pares de meses, ordenados por valor de benefício.
16
17 | while HRSP ou SP não é vazio do
18 |     Se uma das listas é vazia, escolha a outra, caso contrário,
19 |     escolha aleatoriamente entre HRSP e SP com probabilidade de  $\eta$  and  $1 - \eta$ , respectivamente.
20 |     Obtenha, aleatoriamente, um par (p,m) a partir dos primeiros k pares no início da lista de escolha.
21 |     Remova o par (p,m) da lista de escolha.
22
23 |     if (p não estiver agendado) and (acrescentando(p,m) para a solução não violar restrições locais) then
24 |         Let  $P = P \cup \{(p, m)\}$ .
25 |     else
26 |         Descarte o par (p,m)
27 |     end if
28 | end while
29 // Qualquer projeto restante nas listas HRSP e SP não são agendados
30 return P.

```

Figura 4. Algoritmo de construção de portfólio (MIRA et al., 2015).

## 6.2. Busca local

A segunda fase desta heurística é o processo de busca local. A busca local é executada sobre um subconjunto das soluções viáveis obtidas na fase anterior. A busca local começa na solução de entrada e examina a vizinhança checando as possíveis soluções de acordo com a função objetivo. Várias iterações de busca são realizadas nas vizinhanças de cada nova solução até a melhor solução ser encontrada. A vizinhança é o conjunto de portfólios onde cada vizinho é obtido pelo reposicionamento de um dos projetos. Por exemplo, se um projeto  $p$  é agendado para o mês  $m$  em um portfólio, os vizinhos deste são obtidos agendando o  $p$  entre  $m - \Delta$  até  $m + \Delta$ , excluindo-se  $m$  (MIRA et al., 2015).

```
1 Algorithm3 "Procedimento de busca local para o problema PPS "  
2 //  
3 Let P = solução inicial  
4 do  
5     Obter a lista de vizinhos N da solução corrente P.  
6     Buscar através de N, usando a melhor estratégia de melhoria.  
7     if uma solução melhor Q ser encontrada then  
8         Let P = Q.  
9     end if  
10 while Não haja mais soluções melhores que a solução P.  
11 return Portfólio P.
```

Figura 5. Algoritmo de busca local (MIRA et al., 2015).



## **7. Comparação entre portfólios**

Considerando as informações apresentadas nas Tabelas 2 e 4 e, os algoritmos discriminados é possível então, construir um novo portfólio melhor. A Figura 6 apresentada nesta seção, exemplifica um portfólio melhor. Como os projetos na instância de entrada não são de manutenção, esse portfólio não viola as restrições discutidas na Seção 4.





A Figura 7 representa a área sob a curva do gráfico de Risco X Tempo do portfólio P2 usando os valores apresentados nas Tabelas 2 e 3.

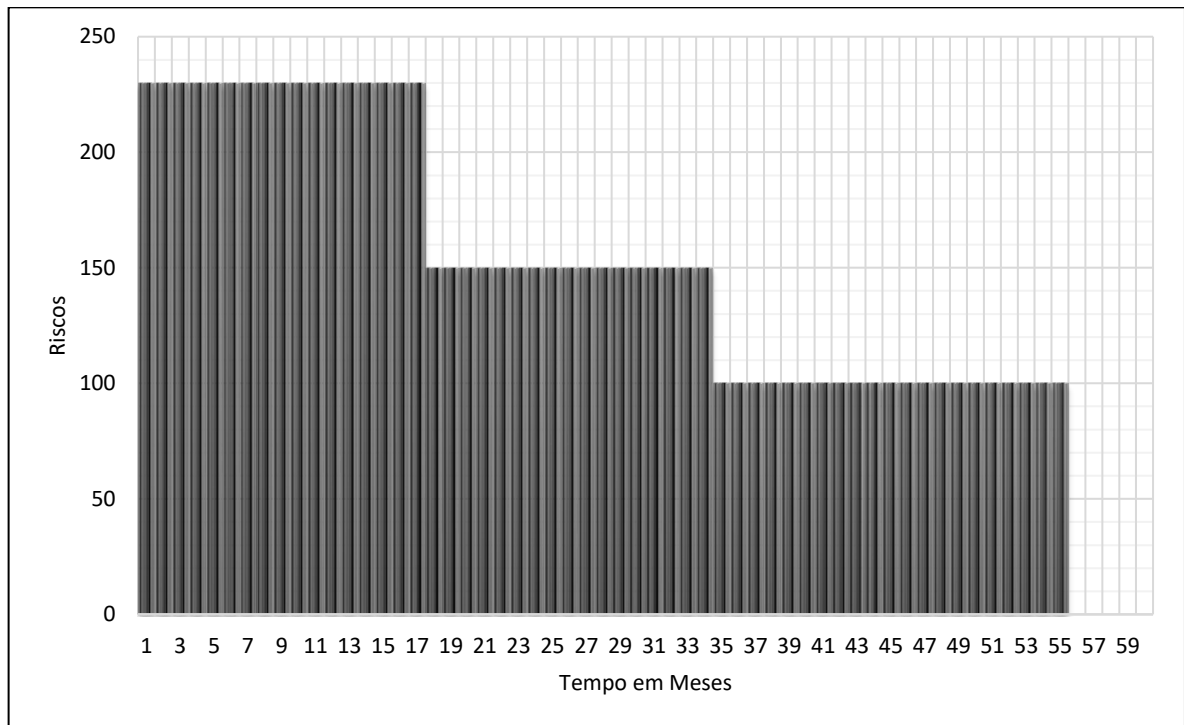


Figura 7. Curva de risco do horizonte de planejamento do portfólio P2 (MIRA et al., 2015).

No exemplo do portfólio P2 o ponto de atenção  $w_2$  foi totalmente controlado no mês 34, quando o último projeto de seu grupo de projetos foi finalizado. Observe que este ponto de atenção é controlado no mês 56 do exemplo do portfólio P1. O agendamento dos projetos que controlam  $w_2$  no portfólio P2 teve um grande impacto no gráfico da curva de risco, como observado na Figura 7. É possível utilizar os gráficos para comparar os dois portfólios e decidir qual é o melhor dentre os dois.

A função que indica se um portfólio é de fato melhor que outro é a área sob a curva do gráfico de risco ao longo do horizonte de planejamento de um portfólio. Quanto menor for a área sob a curva, melhor será o portfólio. O valor para determinar a área sob a curva é dado pela Eq. (17).

O resultado obtido ao aplicar a Eq. (17) no portfólio 1 é:

$$R(P1) = 120 \times 230 - [(50(120 - 27)) + (100(120 - 41)) + (80(120 - 56))]$$

$$R(P1) = 27600 - [(50 \times 93) + (100 \times 79) + (80 \times 64)]$$

$$R(P1) = 27600 - [4650 + 7900 + 5120]$$

$$R(P1) = 27600 - 17670$$

$$R(P1) = 9930$$

O resultado ao aplicar a Eq. (17) no portfólio 2 é:

$$R(P2) = 120 \times 230 - [(80(120 - 14)) + (50(120 - 34)) + (100(120 - 55))]$$

$$R(P2) = 27600 - [(80 \times 103) + (50 \times 86) + (100 \times 65)]$$

$$R(P2) = 27600 - [8240 + 4300 + 6500]$$

$$R(P2) = 27600 - 19040$$

$$R(P2) = 8560$$

Ao comparar os resultados é possível afirmar que a área sob a curva de risco do segundo portfólio é menor do que a do primeiro portfólio, ou seja, o montante de risco ao longo do tempo do portfólio 2 é menor do que o portfólio P1, pois um valor maior de riscos é controlado em um menor tempo pelos projetos.

## 8. Experimentos

Para comprovar a qualidade da heurística que apresentamos, realizamos experimentos para avaliar respostas algoritmo completo dadas algumas instâncias de entrada. Para indicar a qualidade das soluções encontradas, utilizamos o tempo de execução da solução e também o valor da função objetivo.

Para a realização dos experimentos, foram utilizadas algumas entradas simples de dados. Temos a soma de 41 projetos, 39 pontos de atenção distribuídos em todas as entradas. A soma dos recursos anuais para CAPEX são, em reais (R\$), 3mi, 1.2mi, 3mi, 3mi, 1.3mi; e para OPEX os valores dos recursos anuais são, em reais (R\$), 2.5mi, 600mil, 1.75mi, 3mi, 650mil.

### 8.1. Escolha de Parâmetros

Antes de executar o programa e testá-lo, definimos algumas constantes de parâmetros para a heurística.

Durante a fase de construção, o procedimento faz uma escolha entre as listas de pares HRSP e SP com a probabilidade de  $\eta$  e  $1 - \eta$  respectivamente. Definimos o valor de  $\eta = 0.7$ .

Conforme o processo de construção gera novos portfólios, o próximo par de projeto e mês a ser inserido na solução é selecionado aleatoriamente dentre os  $k$  primeiros elementos das listas HRSP ou SP. Definimos  $k = 5$ .

Foram realizados vários procedimentos de testes para definir o valor para a capacidade do conjunto de soluções,  $C$ , o qual foi definido com o tamanho de 20 soluções.

Já na fase de busca local, para encontrar vizinhos válidos em uma solução, fizemos alguns experimentos para selecionar o melhor valor para  $\Delta$  (Delta), e foi decidido que 5 é um bom valor para avaliar um conjunto de vizinhos de uma solução em um tempo razoável.

A tabela 6 demonstra os parâmetros usados na heurística.

Tabela 6. Parâmetros da heurística.

Parâmetro	Valor	Descrição
$\eta$	0.7	Probabilidade de escolha de pares (p,m) dentre as listas HRSP e SP
$k$	5	$k$ melhores elementos a serem escolhidos em uma das listas
$C$	20	Capacidade do conjunto de soluções
$\Delta$	5	$2 \times \Delta$ é o tamanho da vizinhança de uma solução

## 8.2. Definição das instâncias de entrada do problema

Para as de entradas do problema preferiu-se criar cinco tipos de instâncias simples para analisar algumas situações distintas que o algoritmo pode encontrar.

A primeira entrada é formada por um grupo de oito projetos dentre os quais, apenas três são de paradas de máquinas. Além disso, a entrada contém um conjunto de seis pontos de atenção com deadlines próximos dos meses finais do plano de horizonte, e com muita interseção entre os projetos dos grupos de controle que resolvem os pontos de atenção.

Na segunda instância de entrada, houve mais projetos com paradas de máquinas e 75% dos projetos pertencem a maioria dos grupos de controles de pontos de atenção.

A terceira entrada de dados é composta por um grupo menor de projetos, sendo que alguns destes causam paradas de máquinas, e com custos totais de 26% dos recursos totais de CAPEX do primeiro ano. A entrada possui também um conjunto de onze pontos de atenção com datas limites mais próximas do início do horizonte de planejamento e com uma menor interseção entre os grupos de controle.

Na quarta entrada, o número de projetos é igual ao número dos pontos de atenção. Os valores dos custos dos projetos são de 39% em relação aos recursos CAPEX do ano um do horizonte de planejamento. Metade dos projetos são de paradas de máquinas. Quanto ao conjunto de pontos de atenção, entorno de 36% dos projetos resolvem aproximadamente 27% dos pontos de atenção. Metade dos pontos de atenção são intoleráveis e seus deadlines estão na metade do horizonte de planejamento.

Para a última entrada, os valores utilizados são os apresentados na Seção 5 deste trabalho.

A Tabela 7 oferece mais parâmetros de características das instâncias de entrada. A primeira coluna indica a entrada. A segunda coluna apresenta a quantidade de projetos de cada instância. A próxima coluna mostra o número de pontos de atenção de cada entrada. A quarta coluna apresenta a média dos valores de custos totais de cada instância. As colunas, Razão CAPEX e Razão OPEX, representam a razão dos custos totais do conjunto de projetos de cada entrada pelo total de recursos disponíveis de uma mesma categoria. Os valores de custos são apresentados em milhares de reais (R\$).

Tabela 7. Entradas dos experimentos.

<b>Entrada</b>	<b>Nº Projetos</b>	<b>Nº Pontos de Atenção</b>	<b>Custo médio</b>	<b>Razão CAPEX</b>	<b>Razão OPEX</b>
1	08	06	R\$ 70.7	0,04	0,06
2	08	08	R\$ 70.7	0,04	0,06
3	09	11	R\$ 99.2	0,07	0,09
4	11	11	R\$ 149.7	0,10	0,14
5	05	03	R\$ 812.0	-	0,9

### 8.3. Resultados e Análises

A heurística foi executada 10 vezes para cada entrada de dados e sua melhor solução retornada. O experimento foi realizado usando um Intel® Core™ i3-4005U CPU@ 1.70GHz, com 4,00 GB de memória RAM e sistema operacional Windows 10 x64. Todo o algoritmo foi implementado usando a linguagem de programação Python versão 3.6.2.

Os resultados do experimento são apresentados na Tabela 8. A primeira coluna apresenta o identificador da entrada. A segunda e terceira coluna mostram, respectivamente, a menor função objetiva encontrada (**Min FO**) e a médias das funções objetivas para cada entrada (**Media FO**). A quarta coluna apresenta o valor da função objetivo do limitante inferior de cada entrada (**FO LI**). A próxima coluna mostra a razão entre a função objetivo da melhor solução encontrada (**Min FO**) e o valor da função objetivo do limitante inferior (**Razão FO**). E a última coluna apresenta a média de tempo de execução, em segundos, do algoritmo para cada entrada.

Tabela 8. Resultados dos experimentos.

<b>Entrada</b>	<b>Min FO</b>	<b>Media FO</b>	<b>FO LI</b>	<b>Razão FO</b>	<b>Tempo (s)</b>
1	300	300	300	1.0	4,89
2	1014	1014	528	1,92	8,78
3	1434	1434	879	1,63	9,18
4	1101	1101	879	1,25	5,96
5	2640	2640	2640	1.0	6,80

Conforme mostra a Tabela 8, podemos afirmar que a primeira instância alcançou um resultado ótimo, provavelmente devido ao pequeno tamanho da entrada.

Já nas instâncias 2 e 3 o valor da razão entre as funções objetivos foram altos em relação ao limitante inferior. Embora o valor da melhor solução encontrada seja quase o dobro da limitante inferior, não se pode afirmar se o resultado retornado é de fato o ótimo.

A instância 4 foi a que mais se aproximou do limitante inferior, com exceção da primeira entrada. Provavelmente devido as poucas interseções entre os projetos dos grupos de controle.

A última entrada é instância utilizada no exemplo dos Portfólios P1 e P2 apresentados na Seção 5 deste trabalho. No exemplo, a instância foi criada manualmente e livre das restrições de parada de máquinas. O resultado retornado pela heurística é o ótimo, já que a razão entre as funções objetivos da limitante inferior e a melhor solução encontrada é igual 1.0.

A Figura 8 demonstra o gráfico da Risco X Tempo da instância de entrada 5.

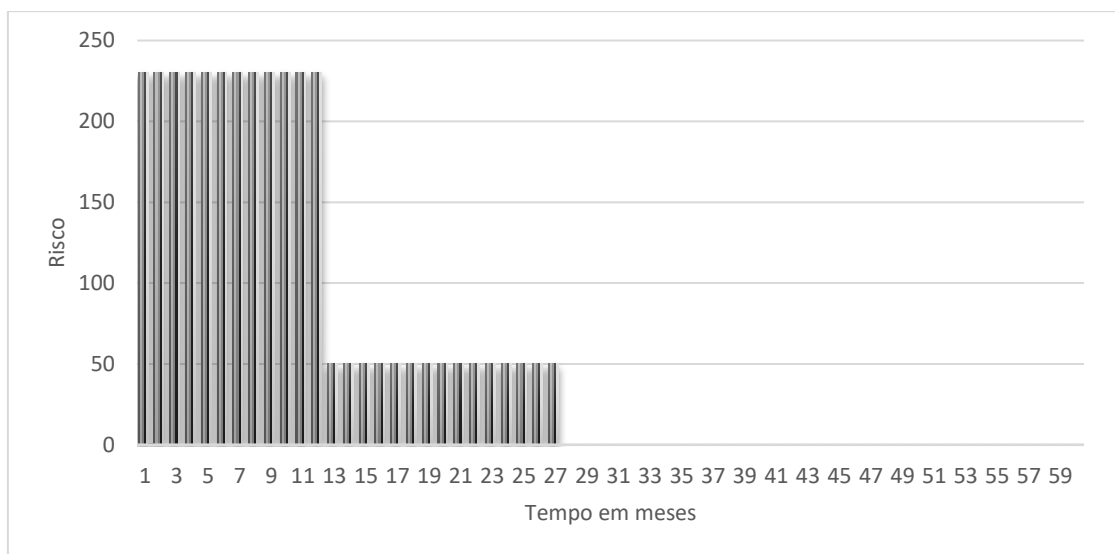


Figura 8. Curva de risco do horizonte de planejamento da entrada 5.

Durante a apresentação dos resultados dos exemplos dos portfólios P1 e P2, o ponto de atenção  $w_2$  foi resolvido, respectivamente nos meses 56 e 34 dos portfólios. Note que, no resultado retornado pela heurística implementada, este mesmo ponto de atenção foi solucionado no mês 14 do horizonte de planejamento. O escalonamento resultante do grupo de projetos que

controlam o ponto de atenção  $w_2$  gerou um valor de função objetivo menor aos dos exemplos anteriores.

Como mostra a Tabela 8, o valor da função objetivo encontrado é de 2640, enquanto os valores de função objetivo dos portfólios anteriores resultaram, respectivamente, em 9930 e 8560. O tempo médio para o algoritmo alcançar esse resultado foi de 6,8 segundos.

Nossa implementação do algoritmo GRASP apresenta uma complexidade de tempo de execução de  $O(n^2m^2)$ , onde  $n$  representa o número de projetos do portfólio e  $m$  o número de meses do horizonte de planejamento. A fase de construção possui um tempo de execução de  $O(R.C.(n^2m^2))$ , e a busca local uma complexidade de  $O(R.2\Delta.\theta.n^2m^2)$ , onde  $\theta$  estima quantas iterações que a busca local irá levar até que não encontre mais nenhuma solução melhor do que corrente. Consideramos para este cálculo apenas o algoritmo do GRASP, junto com a fase de construção e a busca local.





## **9. Conclusão**

Neste trabalho apresentamos uma implementação de uma heurística do GRASP que encontra soluções razoáveis para uma versão do problema de otimização de portfólios encontrado nas empresas de geração de energia elétrica do país. Essa versão de PPS tem por objetivo reduzir os riscos de operação de usinas e apresenta diversas restrições, como, por exemplo, as restrições de paradas de máquina, limites de utilização de recursos e datas limites para resolução de riscos.

Para o tratamento destas variadas restrições, discutimos uma estratégia proposta em um trabalho anterior (MIRA et al., 2015) baseado no algoritmo guloso GRASP.

Realizamos o desenvolvimento da heurística em uma linguagem de alto nível (Python) para estudar as estruturas de dados e algoritmos necessários para sua implementação. Além disso, estudamos o comportamento da heurística em pequenas instâncias de entradas para avaliar a qualidade das soluções. Por fim, foram feitos experimentos computacionais sobre a heurística e a análise dos resultados.

Como previsto, o desempenho desta implementação do algoritmo é inferior ao de uma linguagem como C, no entanto os experimentos dão suporte a ideia de que a heurística consegue encontrar soluções de qualidade razoável.

### **9.1. Trabalhos futuros**

Como uma continuação do presente trabalho, recomenda-se a criação de um algoritmo de força bruta afim de encontrar a melhor solução dado uma entrada base para comparar o tempo de execução com a aplicação apresentada neste trabalho.

Outro possível desenvolvimento, será de aplicações com diferentes técnicas de abordagens. Uma das prováveis abordagens são heurísticas baseadas, por exemplo, em algoritmos genéticos, que utilizada para tomadas de decisão mais complexas.



## Referências

- CAMPBELL, J; CAMPBELL, R. **Analyzing and managing risky investments**. 1st edition. Dubai: Corporate Publishing International, 2001.
- CARAZO, A. et al. Solving a comprehensive model for multiobjective project portfolio selection. **Computers & Operations Research**, v. 37, n.4, p. 630–639, 2010.
- DOERNER, K. et al. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. **Annals of Operations Research**, v. 131 n.1, p. 79–99, 2004.
- FESTA, P.; RESENDE, M. GRASP: basic components and enhancements. **Telecommunication Systems**, v. 46, n.3, p. 253–271, 2011.
- HENRIKSEN, A.; TRAYNOR, A. A practical R&D project-selection scoring tool. **Engineering Management**, IEEE Transactions on, v. 46 n.2, p. 158–170, 1999.
- LEE, J.; KIM, S. Using analytic network process and goal programming for interdependent information system project selection. **Computers & Operations Research**, v. 27 n.4, p. 367–382, 2000.
- LIESIÖ, J.; MILD, P.; SALO, A. Robust portfolio modeling with incomplete cost information and project interdependencies. **European Journal of Operational Research**, v. 190 n.3, p. 679–69, 2008.
- MIRA, C. et al. A GRASP-based heuristic for the Project Portfolio Selection Problem. **The 15th IEEE International Conference on Computational Science and Engineering** (IEEE CSE2012), v. 1, p. 36–41, 2012.
- MIRA, C. et al. Project scheduling optimization in electrical power utilities. **Brazilian Operations Research Society**, v. 35 n. 2, p. 285-310, 2015.
- RESENDE, M.G.C. Greedy Randomized Adaptive Search Procedures. **Encyclopedia of Optimization**, p. 1460–1469, 2009.



## 10. Apêndice

Para a implementação foram criados cinco módulos para resolver uma instância de entrada. Os módulos são: reader.py (leitura de dados das entradas), benefit.py (cálculo dos valores de benefício), constraints.py (validação de restrições), grasp.py (heurística grasp) e main.py (módulo principal).

Neste apêndice encontra-se apenas o código fonte do módulo grasp.py, que possui a função de construção de uma solução e a busca local.

- **Heurística do GRASP**

```
#####--GRASP--#####
def GRASP():
    global BEST_SOLUTION
    global POOL_CAPACITY
    global POOL_SOLUTIONS
    global R
    temporary_pool = {}
    #Número de vezes da execução do Laço
    for x in range(R):
        BEST_SOLUTION = {}
        POOL_SOLUTIONS = []
        #Enquanto o número de soluções for menor que a capacidade
        while(len(POOL_SOLUTIONS) < POOL_CAPACITY):
            #É dado início a fase de Construção de uma solução
            P = solution_constructor()
            #Se não violar as restrições de Pontos de Atenção Intoleráveis
            if(check_intolerable_warning_point(P)):
                #Função para salvar soluções
                save_solution(P)
                #A solução é adicionada ao POOL de soluções
                POOL_SOLUTIONS.append(P)
            #end if
        #end while
        #Para cada soluções no POOL de soluções
        for solution in range(len(POOL_SOLUTIONS)):
            #É calculada a função objetivo de cada solução
            temporary_pool[solution] = objective_function(POOL_SOLUTIONS[solution])
        #end for
        #O POOL de soluções é ordenado em ordem crescente de valores
        sorted_pool_solutions = sorted(temporary_pool.items(), key=operator.itemgetter(1))
        #Início da Fase de Busca Local
        for tupla in sorted_pool_solutions:
            solution = POOL_SOLUTIONS[tupla[0]]
            P = local_search(solution)
            #Se a função objetivo do Portfolio for menor que a BEST_SOLUTION
            if objective_function(P) < objective_function(BEST_SOLUTION):
                #BEST_SOLUTION recebe o Portfolio
                BEST_SOLUTION = P
            #end if
        #end for
    #end for
    return BEST_SOLUTION
#####--GRASP--#####

#####--CONSTRUTOR DE SOLUÇÃO--#####
def solution_constructor():
    #Mapeia projeto em mês de agendamento
    global ETA_PROBABILITY
    global K
    global mandatory_project
    global sorted_benefit_pairs
```

```

CL = []
eta = None
HRSP = []
HRSP_length = None
the_month = -1
the_project = -1
solution = {}
SP = []
SP_length = None
#Para cada projeto obrigatorio
for project in mandatory_project.keys():
    #O Projeto da solução recebe o mês pré-agendado
    solution[project] = mandatory_project[project]
#end for
#####--CONSTRUTOR DE LISTAS--#####
#Para cada par na lista de pares de benefícios
for pair in sorted_benefit_pairs:
    #p recebe um projeto
    p = pair[1]
    #Se o projeto controlar o ponto de atenção intolerável
    if p in intolerable_warning_point.keys():
        #É alocado na lista de Alto risco
        HRSP.append(pair)
    else:
        #Se não é alocado na lista dos projetos remanescentes
        SP.append(pair)
#end if
#end for
#Tamanho de cada lista de Projetos
HRSP_length = len(HRSP)
SP_length = len(SP)
#####--CONSTRUTOR DE LISTAS--#####--END--##
#Enquanto o tamanho das listas forem maior que zero
while(HRSP_length != 0 or SP_length != 0):
    if HRSP_length == 0:
        CL = SP
    elif SP_length == 0:
        CL = HRSP
    else:
        #Variavel com a probabilidade de escolha da lista
        eta = randint(1,10)
        if eta <= ETA_PROBABILITY:
            #Probabilidade de 70%
            CL = HRSP
        else:
            #Probabilidade de 30%
            CL = SP
        #end if
        #Índice aleatório dos projetos da Lista corrente (CL)
        index = (randint(1,min(K,len(CL))) - 1)
        #O par (projeto,mês) recebe o projeto e o mês conforme o index
        the_project = CL[index][0][0]
        the_month = CL[index][0][1]
        #Remove o projeto da lista
        del CL[index]
        HRSP_length = len(HRSP)
        SP_length = len(SP)
        #Lista temporaria de soluções
        temporary_solution = dict(solution)
        #Se o projeto não estiver contido na solução
        if the_project not in solution.keys():
            #Ele é adicionado a solução temporária
            temporary_solution[the_project] = the_month
            #Se o projeto for válido, é acrescentado à solução
            if is_a_valid_solution(temporary_solution):
                solution[the_project] = the_month

```

```

    #end if
    #end if
    #end while
    return solution
#end def
#####--CONSTRUTOR DE SOLUÇÃO--#####-END-##

#####--BUSCA LOCAL--#####
def local_search(P):
    global EPSYLON
    FLAG = False
    #best_current_solution recebe uma cópia da solução de entrada
    best_current_solution = dict(P)
    #Laço infinito
    while (0 == 0):
        #Para cada vizinho encontrado na best_current_solution
        for neighbor in neighborhood_constructor(best_current_solution):
            FLAG = False
            if((objective_function(neighbor) - objective_function(best_current_solution)) < EPSYLON):
                #O vizinho passa a ser a best_current_solution
                best_current_solution = neighbor
                FLAG = True
            #Se não houver mais mudanças
            if not FLAG:
                #A Busca Local é encerrada
                break
            #end if
        #end for
        #Se a best_current_solution for igual a solução de entrada
        if (best_current_solution == P or not FLAG):
            #A Busca Local é encerrada
            break
        #end if
    #end while
    return best_current_solution
#end def
#####--BUSCA LOCAL--#####-END-##
#####-- CONSTRUÇÃO DA VIZINHANÇA--#####
def neighborhood_constructor(P):
    neighbor_list = []
    temporary_solution = dict(P)
    #Para cada projeto da solução
    for project in P.keys():
        #Aloca os projetos em uma proporção de (5-month e month+5)
        for month in range_delta(P[project]):
            #O projeto é escalonado neste dado mês
            temporary_solution[project] = month
            #Se a alocação não violar nenhuma restrição
            if is_a_valid_solution(temporary_solution) and check_intolerable_warning_point(temporary_solution):
                #A solução é adicionada na vizinhança do Portfolio P
                neighbor_list.append(temporary_solution)
            #end if
        temporary_solution = dict(P)
    #end for
#end for
return neighbor_list
#end def
#####-- CONSTRUÇÃO DA VIZINHANÇA--#####-END-##

```