

PROBLEMAS DE ESCALONAMENTO DE PROJETOS COM RESTRIÇÃO DE RECURSOS: UM ESTUDO DE CASO NO SETOR DE PETRÓLEO E GÁS

Guilherme Henrique Ismael de Azevedo

Universidade Federal Fluminense

guilhermehen@gmail.com

Artur Alves Pessoa

Universidade Federal Fluminense

artur@producao.uff.br

Camila Raemy Rangel Torres

COPPEAD/Universidade Federal do Rio de Janeiro

camilaraemy@gmail.com

Resumo

O presente estudo visa aplicar um modelo de otimização de escalonamento de projetos com restrição de recurso a um estudo de caso em uma empresa do setor de Petróleo e Gás. Para isso, um Algoritmo Genético proposto na literatura é adaptado às principais especificidades da empresa. Uma formulação de programação inteira mista é utilizada para validar os resultados da heurística. Ao longo do estudo de caso, são formulados cenários que avaliam diferentes impactos na duração do projeto, auxiliam o dimensionamento das equipes e dando suporte à tomada de decisão. Como na comparação entre os cenários foram encontradas inconsistências, é proposto um método para corrigi-las. Com os resultados, é possível verificar equívocos na elaboração do contrato e a baixa sensibilidade do cronograma em relação determinados recursos. Além disso, é possível analisar diversos fatores que, no futuro, podem originar melhorias para o modelo proposto.

Palavras-chave: Otimização; Escalonamento de Projetos; Restrição de Recursos.

Abstract

This study intends to apply a project schedule with resource constraint model to an Oil and Gas Company case study. For this reason, a Genetic Algorithm proposed in the literature is adapted to the main specificities of this company. Then, we use a mixed integer programming formulation to validate the heuristic results. Throughout the case study, scenarios are created to evaluate different impacts on the project duration, to help sizing the project team and to give support to the decision-making. We propose a method to treat the inconsistencies found during the sensitivity analysis. With the results found, it is possible to check mistakes in the contract preparation and the low sensitivity of the schedules with respect to some resources. In addition, it is possible to analyze some factors that can further improve to the proposed model.

Keywords: Optimization; Project Schedule; Resource constraints.

1. INTRODUÇÃO

Os problemas de escalonamento de projetos com restrição de recursos vêm ganhando maior importância tanto para o meio acadêmico e pesquisa quanto para o mundo corporativo. Para Brucker *et al.* (1999), esse tipo de problema é atrativo para os pesquisadores porque os modelos são mais ricos e, portanto, apresentam maior complexidade na resolução.

No ambiente empresarial, são especialmente importantes para organizações voltadas para projetos, nas quais os recursos são contratados de acordo com a demanda, visando uma produção enxuta. Aplicações para esses modelos ocorrem, principalmente, no planejamento de grandes projetos tais como de investimentos governamentais, construção civil, desenvolvimento de *softwares*, lançamento de foguetes e satélites, construção de usinas hidroelétricas e plataformas de petróleo (PINEDO, 2004).

Este trabalho tem como foco otimizar a execução de um projeto de construção e implantação de um terminal marítimo e planta de armazenamento de gás liquefeito de petróleo (GLP) no Equador. Este terminal é uma instalação de armazenagem frigorífica e pressurizada para propano e butano, os formadores do GLP, com capacidade de 55.000 (cinquenta e cinco mil) toneladas métricas.

Durante o planejamento de um projeto pode-se ter como objetivo, por exemplo, minimizar a utilização de um determinado tipo de recurso, o custo total ou a duração total do projeto. Este estudo limita-se ao último caso.

Dentre as restrições de um projeto, encontram-se as precedências entre atividades, a utilização dos recursos disponíveis para o projeto, ou grupo de projetos, e os prazos de entrega com marcos contratuais. Este estudo considera os dois primeiros tipos de restrição.

Neste sentido, o presente estudo tem como objetivo o desenvolvimento de uma ferramenta que auxilie os gerentes de projeto na tomada de decisões relacionadas à alocação de recursos e ao planejamento de projetos. É utilizado um modelo que minimiza o tempo total de projetos com recursos escassos.

Através da aplicação da ferramenta proposta em um projeto na área de petróleo e gás, o presente estudo pretende mostrar a viabilidade de utilizar as técnicas de otimização na realidade empresarial e evidenciar as vantagens dessa aplicação no ambiente corporativo.

Este documento está organizado em 7 (sete) partes. Após a introdução, a segunda parte traz uma revisão da literatura nos assuntos que envolvem problemas de escalonamento de projetos, restrição de recursos e heurísticas. Em seguida, na seção três, é detalhado o algoritmo utilizado para solucionar o problema proposto de forma heurística, assim como a adaptação que se fez necessária para sua aplicação ao modelo utilizado. Na quarta seção, é apresentado o método usado para validar o procedimento heurístico. Na seção cinco, é apresentado o método para a análise de sensibilidade e correção das inconsistências encontradas. Na seção seis, é apresentado o estudo de caso em uma empresa brasileira que promove soluções de engenharia e tecnologia, através da realização de projetos no setor de petróleo e gás. Por fim, na sétima seção, são colocadas as conclusões do estudo desenvolvido.

2. REVISÃO DA LITERATURA

Segundo informações do Portal Scopus, até fevereiro de 2011, foram publicados 384 (trezentos e oitenta e quatro) artigos que contemplam o escalonamento de projetos com restrição de recursos e suas variações. Nesta seção mencionamos apenas os trabalhos mais relevantes e/ou relacionados com o presente estudo.

Brucker *et al.* (1999) apresentam um resumo das notações, classificações, modelos e métodos para problemas de escalonamento de projetos. Pinedo (2004) descreve os conceitos, as características e as formulações matemáticas para problemas de planejamento e escalonamento

produtivo. Neumann *et al.* (2003) descrevem métodos heurísticos e para o cálculo de limite inferior para estes problemas.

Mendes & Gonçalves (2003) introduzem um Algoritmo Genético que forma as soluções através do incremento do tempo de início das atividades, permitindo atrasos no escalonamento. Para o mesmo problema Hartmann (2001) apresenta um Algoritmo Genético “auto-adaptativo”, no qual o procedimento para decodificar as soluções é definido pelo próprio Algoritmo Genético.

Özdamar (1999) introduz um Algoritmo Genético Híbrido para o planejamento de múltiplos projetos com recursos renováveis e não-renováveis utilizando escalonamento progressivo e regressivo baseado em diferentes regras de prioridade para formar as soluções.

Ayala *et al.* (2010) apresentam um método baseado em Relaxação Lagrangeana para o cálculo de um limite inferior para o *makespan* do escalonamento de módulos de produção. Fleszar & Hindi (2004) apresentam um método que calcula ao mesmo tempo um limite inferior e uma solução válida, utilizando acréscimo de precedências e busca local. Hao *et al.* (2010) apresentam um método heurístico para o planejamento de múltiplos projetos baseado em redes parciais e inclui, ainda, a restrições de exclusividade, que restringe a execução de tarefas em paralelo.

Agarwal *et al.* (2011) introduzem uma abordagem Neurogenética, que é um híbrido de algoritmos genéticos e redes neurais. O algoritmo genético é utilizado na busca global e as redes neurais são utilizadas nas buscas locais de maneira que as boas soluções de um método são transmitidas para o outro. Com essa abordagem, são encontradas soluções melhores que cada um dos métodos em um mesmo número de iterações.

Chen & Wang (2011) utilizam heurísticas de escalonamento de projetos em problema de escalonamento de tarefas. São utilizados dois procedimentos heurísticos: heurísticas por recursos e pelo menor tempo de término. Seus testes demonstram que tal adaptação é possível.

Harmann (2011) discute a utilização de generalizações do problema de escalonamento de projetos com restrição de recursos. O trabalho apresenta um estudo de caso em um projeto real de pesquisa médica com disponibilidade e utilização de recursos variáveis no tempo. Este projeto tem estrutura semelhante a outros projetos de pesquisa da área médica.

Lim *et al.* (2012) introduzem uma heurística baseada no processo de recozimento que simula o esfriamento de gases ao estado sólido. Nesta heurística, o processo de diversificação é baseado na energia molecular. Para melhorar os resultados, também é utilizado um procedimento genético.

Paraskevopoulos *et al.* (2012) representam as soluções através de uma lista de eventos de início e término de atividades. É utilizada uma busca espalhada sobre a lista de eventos para melhorar as soluções. O procedimento incorpora uma busca local adaptativa como mecanismo de melhoria de soluções.

Neste estudo, o procedimento heurístico utilizado é uma adaptação do algoritmo de Mendes & Gonçalves (2003) e sua implementação é validada usando uma formulação semelhante à apresentada por Pinedo (2004).

Como os artigos apresentam diferentes notações, foram adotadas as notações apresentadas por Brucker *et al.* (1999), com poucas adaptações. Um problema de escalonamento de projetos com restrição de recursos consiste em escalonar um conjunto $N = \{1, \dots, n\}$ de atividades ao longo do tempo, respeitando as precedências e restrições de recursos. Também são dados um grafo orientado $G = (N, A)$, que representa as relações de precedência entre as atividades, e um conjunto de recursos renováveis \mathfrak{R}^p . Para cada atividade j é dado o tempo de processamento p_j e a necessidade r_{jk}^p do recurso k .

Além disso, para cada recurso k é dada sua disponibilidade R_k^p e para cada arco $a = (i, j) \in A$ é dado o *time lag* mínimo d_{ij}^{\min} representando o menor tempo entre o término da atividade i e o início da atividade j , podendo ser negativo. Em cada instante do projeto, a

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

soma dos recursos de cada tipo k necessários às atividades em execução não podem exceder sua disponibilidade R_k^ρ .

O objetivo do problema é minimizar o tempo de término da última tarefa concluída, ou *makespan*. Para facilitar, são criadas duas tarefas artificiais zero e $n + 1$ com durações iguais a zero, onde a primeira precede e a segunda sucede todas as demais tarefas, sendo $d_{0j}^{\min} = 0$ e $d_{j(n+1)}^{\min} = 0$. A Tabela 1 apresenta um resumo dos termos adotados nesse estudo.

Tabela 1 – Notações básicas

Símbolo	Definição
N	Conjunto de Atividades
n	Quantidade de atividades reais
A	Conjunto de precedências ou restrições temporais
$G = (N, A)$	Grafo orientado de precedência do tipo <i>término-início</i>
$Pred(i)$	Conjunto de predecessores da atividade i
p_j	Tempo de processamento da atividade j
\mathfrak{R}^ρ	Conjunto de recursos renováveis
R_k^ρ	Quantidade (constante) do recurso k disponível
r_{jk}^ρ	Utilização do recurso renovável k pela atividade j por período (unidade de tempo)
S_j	Tempo de início da atividade j
C_j	Tempo de término da atividade j
C_{\max}	Tempo de término da última tarefa, também chamado de <i>makespan</i>
$r_k(S, t)$	Consumo do recurso k do cronograma S no tempo t
T/t	Horizonte de tempo / índice para o período de tempo
$t = 1, 2, \dots, T$	Períodos
$[t-1, t[$	Intervalo de tempo correspondente ao período t
d_{ij}^{\min}	Menor intervalo de tempo entre os inícios das atividades i e j (<i>time lag</i>)

Fonte: Adaptado de Brucker et al. (1999)

3. O MÉTODO DE RESOLUÇÃO

3.1 REPRESENTAÇÃO CROMOSSÔMICA

Segundo Obitkio (1999), existem três tipos de representações possíveis para os cromossomos: binária, real e inteira. Em sua abordagem, Mendes & Gonçalves (2003) utilizam as chamadas “chaves aleatórias” como método de codificação de soluções. Esta metodologia foi introduzida por Bean (1994) e consiste na atribuição de números aleatórios no intervalo de 0 à 1 aos genes que constituem os cromossomos, portanto é uma representação real.

Cada cromossomo é composto por $2.n$ genes. Os primeiros n representam as prioridades de cada atividade. Os genes de $n + 1$ até $2.n$ são usados para determinar os fatores de espera de cada atividade. As equações (1) e (2) apresentam as formas de cálculo das prioridades e dos fatores de espera da atividade j , a Figura 1 apresenta o esquema de representação dos cromossomos.

$$Prioridade(j) = Gene_j \quad (1)$$

$$FatorEspera(j) = Cte.Gene_{n+j} \quad (2)$$

$$Cromossomo = \underbrace{(gene_1, \dots, gene_n)}_{Prioridade}, \underbrace{(gene_{n+1}, \dots, gene_{2n})}_{Fatores\ de\ Espera}$$

Figura 1 – Esquema de representação dos cromossomos

Fonte: (MENDES & GONÇALVES, 2003)

Em seu estudo, Mendes & Gonçalves (2003) utilizam como valor para a constante (Cte) do fator de espera uma vez e meia o tempo da maior atividade do projeto.

3.2 ESQUEMA DE GERAÇÃO DE PLANOS

Uma vez determinados os genes de um indivíduo, é necessário utilizar uma estratégia para decodificá-lo em uma solução viável para o problema. Neste estudo, é utilizada uma estratégia de formação de soluções baseada nas prioridades das atividades e nos tempos de espera definidos pelos genes.

Com esse objetivo, Mendes & Gonçalves (2003) apresentam duas formas para decodificar indivíduos, o que chamam de Esquemas de Geração de Planos (*Schedule Generation Schemes – SGS*). O primeiro esquema é chamado de SGS Série e está baseado no incremento do número de atividades programadas; o segundo, SGS Paralelo, foi utilizado neste estudo e está baseado no incremento do tempo de escalonamento.

No SGS Paralelo as atividades são sequenciadas em estágios, que têm um tempo de sequenciamento t_g associado. A cada estágio g , F_g , A_g , D_g e $RD_k(t_g)$ são atualizados. F_g é o conjunto das atividades que foram sequenciadas e terminaram seu processamento antes do tempo t_g ; A_g é o conjunto das atividades sequenciadas antes do tempo t_g , mas ainda não finalizadas; D_g é o conjunto das atividades j que podem ser sequenciadas em t_g , ou seja, cujos predecessores estarão concluídos no tempo t_g ; e $RD_k(t_g)$ é a capacidade disponível de cada recurso k no tempo t_g .

O SGS Paralelo começa no tempo $t_g = 0$, atribui-se a atividade de início ($i = 0$) aos conjuntos F_0 e A_0 e calcula-se a capacidade disponível de cada recurso k . Os estágios

consistem em determinar o próximo tempo de processamento t_g e os conjuntos F_g , A_g , D_g e $RD_k(t_g)$ associados ao estágio g . Em seguida as atividades com recursos disponíveis são sequenciadas obedecendo as prioridades.

Segundo Mendes & Gonçalves (2003) embora o procedimento descrito seja vantajoso na eliminação de soluções de má qualidade, poderá excluir o espaço que contém a solução ótima. Para reduzir a probabilidade desta ocorrência, foi incluído um fator de espera que permite considerar como disponíveis para escalonamento no tempo t_g todas as atividades que possam ser iniciadas no intervalo entre t_g e $t_g + FatorEspera(j)$. Além disso, todas as atividades disponíveis poderão ser escalonadas com seu início no mesmo intervalo.

3.3 PROCESSO EVOLUTIVO

Para iniciar o processo é gerada, aleatoriamente, uma população que será alterada pelas operações genéticas de forma que a população de soluções evolua. Mendes & Gonçalves (2003) utilizam estratégias específicas em cada uma das operações.

Para a seleção é empregada a estratégia elitista introduzida por Goldberg (1989). Essa estratégia sugere a cópia dos indivíduos mais aptos de uma geração para formar a geração seguinte. O nível de aptidão é igual ao *makespan* do escalonamento gerado.

Para o cruzamento é utilizada a estratégia dos cruzamentos uniformes parametrizados, introduzida por Spears & Dejong (1991). Neste procedimento são escolhidos, aleatoriamente, dois indivíduos da geração atual, inclusive os selecionados, para darem origem a um novo indivíduo. Mendes & Gonçalves (2003) determinam uma probabilidade de cruzamento de 70% (setenta por cento). Durante o cruzamento, para cada gene é gerado um número aleatório entre 0 (zero) e 1 (um). Se o número gerado for menor ou igual à probabilidade de cruzamento, é escolhido o gene do primeiro progenitor, caso contrário é escolhido o gene do segundo.

Depois que um cruzamento é realizado, acontece a mutação aleatória do descendente gerado como o objetivo de prevenir que todas as soluções presentes na população caiam em um ponto ótimo local.

Segundo Obitkio (1999) a técnica da mutação, assim como o cruzamento, depende da codificação dos cromossomos. Para codificações que usam valores reais nos genes, como é o caso das chaves aleatórias, a mutação é criada adicionando ou subtraindo um valor pequeno do valor contido nos genes escolhidos para mutação. Mendes & Gonçalves (2003) não apresentam detalhes sobre a estratégia de mutação usada na sua heurística, mostram apenas que as mutações são geradas sempre com a probabilidade de 20% (vinte por cento).

3.4 ADAPTAÇÃO DO ALGORITMO

Para adequar ao estudo de caso o método apresentado anteriormente e proposto por Mendes & Gonçalves (2003), 3 (três) adaptações foram necessárias.

A primeira adaptação está acerca do tipo de precedência. Mendes & Gonçalves (2003) apresentam uma formulação com relações de precedência *término-início* e sem *time lag*. O projeto deste estudo considera precedências com *time lag* mínimos, positivos ou negativos.

A segunda adaptação diz respeito às mutações. Como Mendes & Gonçalves (2003) não apresentam detalhes deste processo foi necessário adotar um novo procedimento. Após o cruzamento, cada gene pode ser substituído por um número gerado aleatoriamente com probabilidade a ser definida mais tarde.

A seguir são apresentados os procedimentos do Algoritmo Genético e do SGS Paralelo utilizados neste estudo:

Algoritmo I: Algoritmo Genético

Início

Gerar população inicial S_t aleatoriamente

Avaliar população S_t

Enquanto o número de populações geradas for menor que o pré-estabelecido

Selecionar elementos de S_t e colocar em S_{t+1}

Cruzar elementos de S_t , aplicar mutação e colocar em S_{t+1}

Avaliar nova população S_{t+1}

$S_t = S_{t+1}$

Fim-Enquanto

Fim

Algoritmo II: SGS Paralelo – Formação de Soluções

Início

Inicialização: $g = 0$; $t_g = 0$; $A_0 = \{0\}$; $F_0 = \{0\}$; $RD_k(0) = R_k^p$

Enquanto a atividade $n + 1$ não estiver alocada ou finalizada

Passo 1:

$g := g + 1$

$t_g = \min_{j \in A_g} \{C_j\}$

Calcular F_g , A_g , $RD_k(t_g)$, D_g

Passo 2:

Enquanto $D_g \neq \{ \}$

Selecionar $j \in D_g$ com maior *Prioridade*(j)

$C_j = p_j + \max\{t_g; \text{tempo mínimo para iniciar } j\}$

Atualizar A_g , $RD_k(t_g)$, D_g

Fim-Enquanto

Fim-Enquanto

Fim

4. VALIDAÇÃO DO MÉTODO

Para comprovar a validade dos resultados obtidos pelo procedimento heurístico, todas as soluções geradas são checadas quanto às restrições de precedência e de recursos. Em nenhum teste realizado foi encontrado erro nas soluções geradas.

Além de validar as soluções, foi verificada a eficiência do procedimento comparando seu resultado com os obtidos por método exato. Com esse objetivo, foi implementado um modelo de Programação Inteira Mista (*Mixed Integer Program – MIP*) e os testes foram realizados em instâncias de até 20 atividades.

A seguir é apresentada a formulação matemática introduzida por Pinedo (2004), incluindo-se as precedências do tipo *Término-Início* e os *timelags*.

Variáveis de Decisão:

$$y_{jt} \begin{cases} 1, \text{ se a atividade } j \text{ termina em } t \text{ ou depois} \\ 0, \text{ caso contrário} \end{cases}$$

$$y_{jt} \in \{0, 1\} \quad \text{para } \forall j \text{ e } EF_j \leq t \leq LF_j$$

C_j – tempo de término da atividade j

$$EF_j \leq C_j \leq LF_j \quad \text{para } \forall j$$

Onde EF_j e LF_j são, respectivamente, limite inferior e superior para o tempo de término da atividade j numa solução ótima (as siglas vêm do inglês *Early Finish Time* e *Late Finish Time*, respectivamente). O cálculo destes limites é explicado mais tarde.

Além das variáveis de decisão, são incluídas as variáveis auxiliares ω_{ijt} , que representam o quanto do recurso i a atividade j utiliza no tempo t . Esta variável é retirada do modelo substituindo-a pelo lado direito da equação (7).

Formulação 3: Modelo modificado de Programação Inteira Mista.

$$\min C_{\max} = C_{n+1} \tag{3}$$

Sujeito a:

$$\sum_{t=EF_j}^{LF_j} y_{jt} = 1 - EF_j + C_j \quad \forall j \tag{4}$$

$$C_j + d_{jk}^{\min} \leq C_k - p_k \quad \forall (j, k) \in G \tag{5}$$

$$\sum_{j=1}^n (\omega_{ijt}) \leq R_i^{\rho} \quad \forall i, t \tag{6}$$

$$\omega_{ijt} = \begin{cases} r_{ji}^{\rho} - r_{ji}^{\rho} \cdot y_{j(t+p_j)}, & \text{se } t > EF_j \text{ e } EF_j \leq t + p_j \leq LF_j \\ r_{ji}^{\rho} \cdot y_{jt} - r_{ij}^{\rho} \cdot y_{j(t+p_j)}, & \text{se } EF_j \leq t \leq LF_j \text{ e } EF_j \leq t + p_j \leq LF_j \\ r_{ji}^{\rho} \cdot y_{jt}, & \text{se } EF_j \leq t \leq LF_j \text{ e } t + p_j > LF_j \end{cases} \tag{7}$$

$$y_{jt} \leq y_{j(t-1)} \quad \forall j, t \tag{8}$$

$$y_{j(EF_j)} = 1 \quad \forall j \tag{9}$$

A equação (3) representa o tempo de término da atividade $n+1$, que é precedida por todas as demais atividades do projeto, logo representa o *makespan*. O grupo de restrições apresentado pela equação (4) define os tempos de término C_j . O grupo de restrições apresentado pelas inequação (5) garante que as restrições de precedência sejam respeitadas. O grupo de restrições apresentado pela inequação (6) e pela equação (7) garante que a demanda total para as unidades do recurso i no tempo t não ultrapasse o número total de unidades disponíveis deste recurso. O grupo de restrições apresentado pelas inequações (8) e pela equação (9) assegura que cada tarefa seja processada exatamente uma vez.

Para comparação da heurística com o MIP foram utilizadas 16 (dezesesseis) instâncias de até 20 (vinte) atividades e até 5 (cinco) *pool's* de recursos. Em todos os casos, estão disponíveis 5 (cinco) unidades de cada recurso. A duração, a quantidade de predecessores, os

predecessores, o tipo de precedência, o *time lag* entre as atividades e a necessidade recurso são gerados aleatoriamente.

Por simplificação a duração, o *time lag* e a necessidade de recursos foram considerados como números inteiros. Para se aproximar do estudo de caso, as atividades têm, no máximo, dois predecessores, que são escolhidos entre as atividades geradas anteriormente e a atividade 0 (zero), início do projeto. Desta forma, é possível garantir que todas as sequências de atividades comecem na atividade 0 (zero) e terminem na atividade $n+1$. As relações de precedência têm 70% (setenta por cento) de chance de serem do tipo *término-início* e o *time lag* é um número natural entre 0 (zero) e 10 (dez). Caso contrário elas serão do tipo *início-início* e o seu *time lag* será ajustado para convertê-la para *término-início*.

As instâncias podem ser divididas em dois grupos com características comuns, conforme a Tabela 2.

Tabela 2 – Instâncias para validação do modelo

	Grupo 1				Grupo 2			
Instâncias	A1, A2 e A3	A4, A5 e A6	A7, A8 e A9	A10, A11 e A12	B1	B2	B3	B4
n	10	10	10	10	20	20	20	20
p_j	Entre 1 e 25	Entre 1 e 50	Entre 1 e 25	Entre 1 e 25	Entre 1 e 25	Entre 1 e 50	Entre 1 e 25	Entre 1 e 25
$ \mathcal{R}^p $	2	2	3	5	2	2	3	4

4.1 COMPARAÇÃO DE RESULTADOS

A heurística foi implementada em C++ e, como este processo inclui a utilização de números aleatórios, o código é executado 50 (cinquenta) vezes para cada instância, sendo que, cada execução utiliza uma chave aleatória diferente. Em cada execução são geradas 30 (trinta) populações de 50 (cinquenta) indivíduos, sendo 15 (quinze) selecionados da população anterior e os demais indivíduos gerados por cruzamento. As probabilidades de cruzamento e mutação utilizados foram 70% (setenta por cento) e 4% (quatro por cento), respectivamente, e o maior atraso possível para as atividades é de 100.000 (cem mil) unidades de tempo, reduzindo a possibilidade de excluir a solução ótima do espaço de busca.

A formulação por MIP foi implementada e resolvida pelo *software* de otimização *ILOG CPLEX 12.1*. O cálculo de EF_j corresponde a um caminho mais longo da tarefa 0 até a tarefa j pelo grafo de precedências, sem considerar recursos. Já o cálculo de LF_j corresponde ao *makespan* da melhor solução encontrada subtraído de um caminho mais longo desde a tarefa j até a tarefa $n+1$ no mesmo grafo.

Todos os testes foram realizados em computador *Core 2 Duo 3.0 GHz* e 4 GB de memória RAM. A Tabela 3 resume os resultados obtidos para o problema e compara as soluções encontradas pela heurística e pelo MIP. A primeira coluna apresenta a identificação da instância, em seguida são apresentados os resultados obtidos com a heurística e o MIP. Para a heurística, são apresentados a média e o melhor valor encontrado nas 50 (cinquenta) execuções; a média dos tempos de execução e o tempo total para 50 (cinquenta) execuções; e a diferença percentual entre o melhor valor encontrado pela heurística e o menor valor possível para o *makespan*, ou para a solução ótima, caso disponível. Para o MIP, são apresentados a melhor solução encontrada; o limite inferior para o *makespan*; e o tempo de execução no *software* de otimização. Todos os valores de *makespan* comprovadamente ótimos estão em negrito na tabela.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Tabela 3 – Comparação de resultados

Inst	Heurística					MIP		
	Makespan Médio	Melhor Makespan	Tempo médio (s)	Tempo Total (s)	GAP %	Makespan	Melhor LB	Tempo (s)
A1	66,08	66	0,15236	7,618	0,00%	66	66	0,74
A2	87	87	0,15836	7,918	0,00%	87	87	0,16
A3	81,44	81	0,15178	7,589	0,00%	81	81	4,41
A4	192	192	0,15112	7,556	0,00%	192	192	15,45
A5	199,02	199	0,15196	7,598	0,00%	199	199	415,92
A6	243	243	0,15426	7,713	0,00%	243	243	3188,42
A7	84	84	0,15784	7,892	0,00%	84	84	30,71
A8	136	136	0,16952	8,476	0,00%	136	136	2241,27
A9	82	82	0,17546	8,773	0,00%	82	82	0,11
A10	98,06	98	0,17942	8,971	0,00%	98	98	19,17
A11	141	141	0,18092	9,046	0,00%	141	141	1358,87
A12	137,04	137	0,17694	8,847	0,00%	137	137	225,31
B1	170,26	165	0,2683	13,415	9,27%	-	151	>3600,00
B2	337,82	337	0,26718	13,359	56,80%	-	214,927	>3600,00
B3	202,34	197	0,27356	13,678	74,46%	-	112,915	>3600,00
B4	255,2	251	0,30118	15,059	52,76%	-	164,315	>3600,00

O MIP encontrou a solução ótima para 12 (dose) das 16 (dezesesseis) instâncias testadas. Para o grupo 1, todas as instâncias tiveram a solução ótima encontrada pela heurística, sendo assim, os resultados do Algoritmo Genético são de boa qualidade. Contudo, para o grupo 2, o *ILOG CPLEX 12.1* não foi capaz de encontrar a solução ótima ou qualquer solução viável em 1 (uma) hora de execução, sendo assim, não é possível ter certeza quanto a qualidade da solução heurística. Nestes casos, esta solução foi comparada com o limite inferior do *makespan*. Para a instância B1 a diferença entre estes valores é inferior a 10% (dez por cento). Para B2 e B4 a diferença foi superior a 50% (cinquenta por cento) inferior a 60% (sessenta por cento). Para a instância B3 a diferença está próxima a 75% (setenta por cento). Mesmo que a diferença seja elevada, em nenhum destes casos é possível garantir que o *GAP %* não possa ser reduzido já que a solução ótima não é conhecida.

Tendo em vista os fatores levantados com a comparação de resultados, é evidente que uma das dificuldades encontradas pelo MIP durante a resolução do problema está relacionada à quantidade de atividades. É notório que a quantidade de restrições e variáveis depende, fortemente, da quantidade de atividades. Esta relação, portanto, justifica a impossibilidade de encontrar solução.

Além disso, em contradição ao MIP, a heurística sempre foi capaz de gerar uma solução viável em tempo útil, independente do tamanho do problema. Para as instâncias A1, A2, A3 e A9, o MIP foi capaz de encontrar a solução ótima mais rápido que o tempo total de execução da heurística. Contudo, devemos levar em consideração que o MIP necessita de uma solução viável para calcular os valores de LF_j e a qualidade desta solução influencia no desempenho do MIP.

O conjunto desses resultados e análises levou a conclusão do processo de validação da heurística.

5. ANÁLISE DE SENSIBILIDADE

Análise de sensibilidade é uma etapa muito importante na metodologia de análise de decisão. De modo geral, é utilizada para: tomar melhores decisões, decidir quais dados estimados devem ser refinados antes de tomar uma decisão e concentrar-se nos elementos críticos durante a implementação (SILVA & BELDERRAIN, 2004). Neste trabalho, a análise de sensibilidade foi feita através do levantamento de cenários.

Como o presente estudo utiliza um método heurístico, os resultados encontrados para os diferentes cenários não obrigatoriamente são ótimos ou têm o mesmo afastamento da solução ótima, podendo haver inconsistências na análise de sensibilidade.

Os cenários alternativos construídos para a análise de sensibilidade foram obtidos através da retirada ou acréscimo de recursos disponíveis para a execução do projeto. Para os cenários associados às retiradas de recursos (mais restritos), se espera que tenham *makespans* maiores ou iguais uma vez que qualquer solução do cenário alternativo poderia ser utilizada no cenário original. Em contrapartida, para os cenários associados aos acréscimos de recursos (o cenário original é mais restrito do que o alternativo), se espera que tenham *makespans* menores ou iguais uma vez que qualquer solução do cenário original poderia ser utilizada no cenário alternativo. Caso uma dessas expectativas não seja alcançada, é necessário tratar a inconsistência. A seguir é proposto um algoritmo com esta finalidade.

Algoritmo III: Correção de Inconsistências na Análise de Sensibilidade

Enquanto houver inconsistências

 Testar solução dos cenários mais restritos nos menos restritos;

Fim-Enquanto

Testar a melhor solução de todos os cenários para todos os demais, se ela já não foi testada;

Rodar novamente a heurística para o Cenário Original, incluindo as soluções anteriormente testadas na primeira população;

Se a solução para o Cenário Original melhorar Então

 Testar esta solução nos demais cenários;

Fim-se

Enquanto houver inconsistências

 Testar solução dos cenários mais restritos nos menos restritos;

Fim-Enquanto

Para testar as soluções de um cenário em outro, além do cromossomo que deu origem a solução, são gerados outros 2 cromossomos. Em ambos os casos, a prioridade das atividades é dada pelo tempo de início na solução, quanto antes a atividade começar maior sua prioridade. Já os genes que determinam atraso são tais que o atraso permitido seja igual ao tempo de início da atividade, no primeiro caso, e igual a 1, no segundo caso.

6. APLICAÇÃO DO MÉTODO

6.1 CARACTERIZAÇÃO DO PROJETO

Como dito anteriormente, o escopo geral do projeto estudado é a construção e implantação de um terminal marítimo e planta de armazenamento de gás liquefeito de petróleo (GLP) para o Equador. A realização deste projeto contém 1.626 (mil seiscentos e vinte e seis) atividades que têm até 2 (dois) predecessores, a duração das atividades varia de 0,5 (meia hora) até 771,12 (setecentos e setenta e uma horas e doze centésimos) horas e os *time lags* de -80 (menos oitenta) até 580 (quinhentos e oitenta) horas.

Foram utilizados profissionais das disciplinas de civil, elétrica, telecomunicações, instrumentação, automação, processos, mecânica, tubulação, arquivo técnico, tradução, gestão e

planejamento. As equipes são compostas por gerente de projeto, *job leaders*, engenheiros (sênior, pleno e júnior), técnicos, projetistas (sênior e júnior), desenhistas, auxiliar em AutoCAD e tradutores. No total, a equipe é composta por 44 (quarenta e quatro) profissionais que estão divididos em 33 (trinta e três) tipos de recursos. Cada unidade de recurso é dividida em 10 unidades, permitindo a alocação parcial em atividades. Por exemplo, se a atividade *j* requer 1 (um) *engenheiro sênior* alocado 30% (trinta por cento) do seu tempo, cada *engenheiro sênior* representará 10 (dez) unidades do recurso sendo 3 (três) deles alocados para a atividade.

6.2 DADOS E TRATAMENTO

Para que os dados fornecidos se tornassem entrada do modelo proposto alguns tratamentos foram necessários.

A disciplina de planejamento é responsável, apenas, pela fase de iniciação do projeto e, portanto, não foi considerada em nenhuma das atividades de execução. Na unidade funcional de gestão estão alocados os gerentes de projetos que também não contabilizam suas horas nas atividades e têm disponibilidade total no tempo de projeto. As disciplinas de automação e instrumentação possuem os mesmos funcionários da empresa, logo foram agrupadas para não haver duplicidade na alocação dos recursos.

Em geral, cada disciplina possui um *job leader* que de acordo com a empresa é caracterizado, na maior parte das vezes, como engenheiro sênior com um papel de liderança. Sendo assim, por simplificação, esses recursos foram transformados em engenheiro sênior.

O planejamento do projeto inclui atividades que não são realizadas pela empresa e, sim, pelo próprio cliente ou por recurso terceirizado. Nesses casos, o modelo considera que as atividades têm um tempo de duração, mas não aloca nenhum recurso.

6.3 RESULTADOS OBTIDOS

As 50 (cinquenta) rodadas da heurística demoram 5.084,75 segundos para executar (101,69 segundos por rodada em média). A melhor solução encontrada foi de 10.554,7 horas e a média das melhores soluções de cada rodada foi de 11.056,27. O modelo considera que cada recurso está disponível por 8 (oito) horas diárias, mas na prática empresarial os recursos trabalham em média 12 (doze) horas por dia. Por esse motivo, a análise deste item pondera o resultado em função do aumento da carga horária de trabalho, e entende que um ano tem 252 (duzentos e cinquenta e dois) dias úteis.

A expectativa contratual do projeto era de 1 (um) ano (ou 2.016 horas) enquanto que a expectativa real de duração do projeto girava em torno de 2 (dois) anos (ou 4.032 horas). Assim, o resultado encontrado pelo modelo não está de acordo com estimativa da disciplina de planejamento da empresa e indica que o equívoco na elaboração do contrato. A empresa reconhece a impossibilidade de manter o mesmo escopo vinculado à mesma expectativa de prazo.

6.4 ANÁLISE DE SENSIBILIDADE

Para melhorar os resultados obtidos e compreender o comportamento do modelo quando há alterações de algumas premissas, foram estudados 6 cenários junto ao gerente do projeto.

Segundo o gerente, empresas que atuam com projetos no ramo de engenharia apresentam alto *turnover* de funcionários, principalmente quando o mercado está aquecido. Adicionalmente, empresas de pequeno porte, em geral, alocam seus recursos em um ou mais projetos ou ainda redistribuem ou substituem seus recursos em momentos delicados de algum projeto de seu portfólio.

Nesse contexto, a perda de engenheiro sênior para o mercado ou para outro projeto traz prejuízos para o andamento do projeto. Em menor escala, os projetos também estão sensíveis à

falta de projetistas em função de suas habilidades que são mais específicas. Por outro lado, a perda de um desenhista, por exemplo, pode ser facilmente corrigida por um projetista ou até mesmo engenheiro de qualquer escalão. Outro ponto crítico destacado pelo gerente de projeto é o andamento da disciplina de civil, pois ela apresenta maior dependência e sensibilidade ao trabalho do engenheiro sênior.

Durante a execução do projeto estudado, um engenheiro sênior da disciplina elétrica precisou ser alocado em outro projeto e ficou disponível apenas 50% (cinquenta por cento) do tempo planejado, acarretando atrasos e dificuldades para desempenhar as atividades.

Tendo em vista os pontos levantados pela empresa alguns cenários foram criados com a finalidade de estudar o impacto da disponibilidade dos recursos na duração do projeto e auxiliar o processo de dimensionamento da equipe.

- **Cenário 1:** Ganho de engenheiro sênior da disciplina civil durante todo o projeto;
- **Cenário 2:** Perda de engenheiro sênior da disciplina civil durante todo o projeto, sendo substituído por 1,3 engenheiro pleno na mesma especialidade;
- **Cenário 3:** Perda de engenheiro sênior da disciplina civil por 15 dias úteis, no período entre os dias 412 e 427;
- **Cenário 4:** Ganho de engenheiro sênior da disciplina civil por 15 dias úteis, no período entre os dias 412 e 427;
- **Cenário 5:** Substituição de desenhistas por projetistas e engenheiros;
- **Cenário 6:** Perda de 50% da disponibilidade do engenheiro sênior de elétrica.

6.4.1 COMPARAÇÃO ENTRE CENÁRIOS E RESULTADOS

Era esperado que os cenários 1 e 4, que envolvem ganhos de recurso, apresentassem resultados melhores, ou seja, *makespan* menor que o caso original; enquanto que as perdas de recursos dos cenários demais cenários conduzissem a um aumento na duração total do projeto. Entretanto, analisando as Tabelas 4 e 5, é possível perceber que as expectativas não foram alcançadas no cenário 3, ficando evidente que a solução para este caso não é ótima. Na Tabela 5, as inconsistências são colocadas em negrito.

Tabela 4 – Comparativo entre cenários

Cenário	Descrição	Resultado	Varição
0	Original	10.696,50	-
1	Ganho de engenheiro sênior da disciplina civil	12.212,50	14,2%
2	Perda de engenheiro sênior da disciplina civil	14.974,30	40,0%
3	Perda de engenheiro sênior da disciplina civil por 15 dias úteis	10.696,50	0,0%
4	Ganho de engenheiro sênior da disciplina civil por 15 dias úteis	10.696,50	0,0%
5	Substituição de desenhistas	11.738,10	9,7%
6	Perda de 50% da disponibilidade do engenheiro sênior de elétrica	11.563,90	8,1%

Para sanar as inconsistências, a melhor solução de cada cenário foi testada para todos os demais, conforme detalhado na seção 5. As melhores soluções obtidas com estes cromossomos podem ser observadas na terceira coluna da tabela 5. É possível observar que, para todos os cenários, exceto o terceiro, foi encontrada uma solução melhor que as anteriormente encontradas e que a solução para o cenário 5 não atende as expectativas para este caso. Assim, uma nova rodada do processo foi feita utilizando os cromossomos desta solução. Os resultados para esta rodada são apresentados na quarta coluna. Com exceção ao cenário 2,

todas as soluções passaram a ter o mesmo valor, assim não há inconsistências. Em seguida, a heurística foi rodada novamente para o cenário original, porém todos os cromossomos testados nas rodadas 1 e 2 foram incluídos na primeira geração, tendo seu resultado apresentado na coluna 5. Das 50 execuções da heurística, 9 encontraram soluções melhores que as já conhecidas, assim, os cromossomos destas 9 soluções foram testados para todos os casos. Os resultados apresentados na coluna 6 demonstra inconsistência para o resultado do cenário 3. Além dos cromossomos desta solução, foram testados os do cenário 1, que apesar de não ter uma solução incoerente, tem *makespan* menor que o caso original. A coluna 7 apresenta os resultados desta interação e a coluna 8 os melhores resultados encontrados para cada cenário. Como estes resultados não são incoerentes, o procedimento foi interrompido.

Tabela 5 – Tratamento de Inconsistências

Cenários	Heurística	Rodada 1	Rodada 2	Rodada 3	Rodada 4	Rodada 5	Melhor	Variação
0	10.554,70	10.356,40	10.348,40	10.286,40	10.286,40	10.276,40	10.276,40	-
1	10.435,50	10.356,40	10.348,40	-	10.276,40	10.276,40	10.276,40	0,0%
2	12.440,70	12.439,40	13.554,50	-	13.011,60	13.032,00	12.439,40	21,0%
3	10.356,40	10.356,40	10.348,40	-	10.276,40	10.276,40	10.276,40	0,0%
4	10.408,30	10.356,40	10.348,40	-	10.286,40	10.276,40	10.276,40	0,0%
5	10.663,90	10.348,40	10.348,40	-	10.308,40	10.817,40	10.308,40	0,3%
6	10.653,90	10.463,40	10.348,40	-	10.286,40	10.276,40	10.276,40	0,0%

A análise dos resultados reafirmou e mensurou os prejuízos que as perdas de engenheiro civil e, em menor escala, de desenhistas provocam no andamento do projeto e, como isso, o dimensionamento de equipes em projetos semelhantes poderá ser feita de forma mais precisa.

6.4.2 CONSIDERAÇÕES FINAIS DO ESTUDO DE CASO

Tendo em vista os resultados obtidos com a análise de sensibilidade e os impactos causados por perdas e ganhos de recursos foi possível verificar que:

- O cronograma do projeto é sensível aos recursos com maior especialização, como engenheiro sênior;
- Não foi possível garantir que o ganho de um recurso, sem outras alterações como consequência, possa reduzir a duração do projeto.

Em função da forma de construção do modelo, não foi possível avaliar qual seria o investimento em recurso necessário para que a empresa fosse capaz de desenvolver o projeto dentro do prazo estabelecido por contrato. Todavia, o melhor valor encontrado pela heurística é mais de 2 (duas) vezes expectativa de duração do projeto pela empresa, o que mostra que uma ferramenta como a desenvolvida neste estudo seria de grande valia durante o planejamento do projeto. Além disso, uma avaliação do projeto sem qualquer restrição de recursos indica que seriam necessárias 2.072,50 horas, ou 259 dias, de trabalho, portanto mais de um ano útil para a entrega do projeto, ratificando a impossibilidade de finalizar o empreendimento dentro da expectativa contratual. Podemos observar que o limite inferior calculado sem a limitação de recursos é, aproximadamente, 5 (cinco) vezes menor que a melhor solução encontrada pela heurística evidenciando a má qualidade deste limite.

7. CONCLUSÕES

O presente estudo mostra que é possível desenvolver ferramentas para resolver problemas de escalonamento de projeto com restrição de recursos. O método empregado no algoritmo genético pode ser melhorado ou pode fazer uso de outras heurísticas buscando novas e melhores soluções. Além disso, uma análise estatística sobre a variabilidade dos resultados gerados no caso original e nos demais cenários produziria informações mais precisas para a análise de sensibilidade.

Este estudo introduziu uma metodologia para tratar inconsistências durante a análise de sensibilidade. Além de sanar as incosistências, o método também contribuiu para a melhoria dos resultados de todos os cenários. No caso original, por exemplo, o *makespan* reduziu de 10.696,50 (dez mil seiscentos e noventa e seis e meia) horas para 10.276,4 (dez mil duzentos e setenta e seis e quatro décimos) horas, logo uma redução de 3,93% (três vírgula noventa e três por cento).

Como extensão desse estudo ainda é possível utilizar métodos exatos para resolver o problema, ou obter um limite inferior para a solução ótima, e considerar diferentes funções objetivo, mais aplicadas no mercado. Outras modificações no modelo poderiam ser incorporadas como, por exemplo, na forma de processar as atividades ou na transformação do modelo de duração fixa para trabalho fixo, possibilitando, assim, mudanças na duração das atividades em função da disponibilidade dos recursos.

Ainda há espaço para agrupar as atividades e, com isso, tratar a duração das atividades em dias e não em horas. Isso permitiria uma considerável redução do problema que, conseqüentemente, melhoraria o desempenho da heurística. Por outro lado, aglutinar atividades pode não representar a melhor forma de gerenciar recursos.

Além disso, para o melhor aproveitamento da ferramenta no ambiente corporativo é possível desenvolvê-la com interface direta ao *Microsoft Project* facilitando o uso e acesso às informações pelos gerentes de projetos e membros das equipes.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGARWAL, A.; COLAK, S.; ERENGUC, S. A Neurogenetic approach for the resource-constrained project scheduling problem. **Computers & Operations Research**, 38, n. 1, p. 44–50, 2011.
- AYALA, M.; ARTIGUES, C.; GACIAS, B. Lagrangian relaxation-based lower bound for resource-constrained modulo scheduling. **Electronic Notes in Discrete Mathematics**, n. 36, p. 191–198, 2010.
- BEAN, C. J. Genetics and Random Keys for Sequencing and Optimization. **ORSA Journal on Computing**, n. 6, p. 154–160, 1994.
- BLAZEWICZ, J.; LENSTRA, J. K.; KAN, A. H. G. Scheduling subject to resource constraints: classification and complexity. **Discrete Applied Mathematics**, n. 1, p. 11–24, 1983.
- BRUCKER, P.; DREXL, A.; MOHRING, R.; NEUMANN, K.; PESCH, E. Resource-constrained project scheduling: Notation, classification, models, and methods. **European Journal of Operational Research**, 112, n. 1, p. 3–41, 1999.
- CHEN, R. M.; WANG, C. M. Project Scheduling Heuristics-Based Standard PSO for Task-Resource Assignment in Heterogeneous Grid. **Abstract and Applied Analysis**, 20p., 2011.
- FLESZAR, K.; HINDI, K. S. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. **European Journal of Operational Research**, n. 155, p. 402–413, 2004.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1ª. ed. Boston: Addison-Wesley Professional, 1989.

HAO, Q.; SHEN, W.; XU, Y.; WANG, S. Task network-based project dynamic scheduling and schedule coordination. **Advanced Engineering Informatics**, n. 24, 4, p. 417–427, 2010.

HARTMANN, S. A Self-Adapting Genetic Algorithm for Project Scheduling under Resource Constraints. **Naval Research Logistics**, n. 49, p. 433–448, 2002.

HARTMANN, S. Project scheduling with resource capacities and requests varying with time: a case study. **Flexible Services and Manufacturing Journal**, p. 1–20, 2012.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Ann Arbor: The University of Michigan Press, 1975.

LIM, A.; MA, H.; RODRIGUES, B.; TAN, S.; XIAO, F. New meta-heuristics for the resource-constrained project scheduling problem. **Flexible Services and Manufacturing Journal**, p. 1–26, 2011.

LOURENÇO, H. R.; MARTIN, O.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**. Norwell: Kluwer Academic Publishers, v. 57, Cap. 11, p. 321–353, 2002.

MENDES, J. J. D. M.; GONÇALVES, J. F. Um algoritmo genético para o problema do sequenciamento de projectos com recursos limitados. **Investigação Operacional**, Lisboa, v. 23, n. 2, p. 179–195, 2003.

NEUMANN, K.; SCHWINDT, C.; ZIMMERMANN, J. Resource-Constrained Project Scheduling: Minimization of Project Duration. In: NEUMANN, K.; SCHWINDT, C.; ZIMMERMANN, J. **Project Scheduling with Time Windows and Scarce Resources**. 2. ed. [S.l.]: Springer, 2003.

NEUMANN, K.; SCHWINDT, C.; ZIMMERMANN, J. Temporal Project Scheduling. In: NEUMANN, K.; SCHWINDT, C.; ZIMMERMANN, J. **Project Scheduling with Time Windows and Scarce Resources**. 2. ed. [S.l.]: Springer, 2003.

OBITKO, M.; SLAVÍK, P. Visualization of Genetic Algorithms in a Learning Environment. **Spring Conference on Computer Graphics**, Bratislava, p. 101–106, 1999.

ÖZDAMAR, L. A Genetic Algorithm Approach to a General Category Project Scheduling Problem. **IEEE Transactions on Systems, Man, And Cybernetics**, v. 29, n. 1, 1999.

PARASKEVOPOULOS, D. C.; TARANTILIS, C. D.; IOANNOU, G. Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm. **Expert Systems with Applications**, 39, p. 3983–3994, 2012.

PINEDO, M. L. Project Planning and Scheduling. In: PINEDO, M. L. **Planning and Scheduling in Manufacturing and Services**. Nova York: Springer, 2004.

Portal Scopus. Disponível em: <http://www.scopus.com>. Acesso em: fevereiro 2011.

SILVA, R. M. D.; BELDERRAIN, M. C. N. **Considerações sobre Análise de Sensibilidade em Análise de Decisão**. Instituto Tecnológico de Aeronáutica. São José dos Campos, p. 1, 2004.

SPEARS, W. M.; DEJONG, K. A. On the Virtues of Parameterized Uniform Crossover. **Proceedings of the Fourth International Conference on Genetic Algorithms**, p. 230–236, 1991.

VOSS, S.; OSMAN, I. H.; ROUCAIROL, C. **Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization**. Nova York: Springer-Verlag, 1998.