



Universidade de Brasília

Faculdade de Economia, Administração, Contabilidade e Gestão de Políticas Públicas

Departamento de Administração

Lorranna Gabriele Gonçalves Couto

**Viabilidade da Utilização de Softwares em Programação  
Matemática Inteira: Análise de um problema de grande  
porte nos Softwares AIMMS e LINGO**

Brasília/DF

2020

Lorranna Gabriele Gonçalves Couto

**Viabilidade da Utilização de Softwares em Programação  
Matemática Inteira: Análise de um problema de grande  
porte nos Softwares AIMMS e LINGO**

Monografia apresentada ao Departamento de Administração como requisito parcial à obtenção do título de Bacharel em Administração.

Professor Orientador: Doutora, Silvia Araújo dos Reis

Brasília/DF

2020

Couto, Lorranna Gabriele Gonçalves.

Viabilidade da utilização de softwares em Programação Matemática Inteira: Análise de um problema de grande porte nos softwares AIMMS e LINGO/ Lorranna Gabriele Gonçalves Couto. – Brasília, 2020.

91 f. : il.

Monografia (bacharelado) – Universidade de Brasília, Departamento de Administração, 2020.

Orientadora: Prof. Dr. Silvia Araújo dos Reis, Departamento de Administração.

1. Software. 2. AIMMS. 3. LINGO. 4. Programação Inteira. I. Viabilidade da utilização de softwares em Programação Matemática: Análise de um problema de grande porte nos softwares AIMMS e LINGO.

Lorranna Gabriele Gonçalves Couto

**Viabilidade da Utilização de Softwares em Programação  
Matemática Inteira: Análise de um problema de grande  
porte nos Softwares AIMMS e LINGO**

A Comissão Examinadora, abaixo identificada, aprova o Trabalho de  
Conclusão do Curso de Administração da Universidade de Brasília da  
aluna

**Lorranna Gabriele Gonçalves Couto**

Doutora, Sílvia Araújo dos Reis  
Professor-Orientador

Doutor, Victor Rafael Rezende Celestino  
Professor-Examinador

Doutora, Patrícia Guarnieri  
Professor-Examinador

Brasília, 26 de novembro de 2020

Dedico este trabalho aos meus pais, Jairo e Saira, que não mediram esforços para a realização deste sonho. Meus maiores incentivadores ao longo da minha jornada acadêmica.

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, que mesmo geograficamente distantes, se fizeram presentes durante a minha graduação. Obrigada por me mostrarem a importância dos estudos e investirem na minha educação desde a infância. A minha vitória também é de vocês.

Aos meus avós, meu irmão e todos os familiares que de alguma forma me apoiaram e sempre torceram pelo meu sucesso.

Aos professores do Departamento de Administração da UnB pelos conhecimentos repassados em cada aula, em especial, agradeço a minha orientadora, Doutora Silvia Araújo dos Reis, pela sua disponibilidade, confiança, dedicação e paciência, sempre empenhada em elevar a qualidade desta pesquisa.

Aos meus amigos, colegas de curso e ao meu namorado que tornaram essa caminhada mais leve, obrigada pelas palavras de encorajamento, vocês fizeram toda a diferença.

## RESUMO

O software de programação matemática é uma ferramenta fundamental no emprego da Pesquisa Operacional como estratégia para a tomada de decisão. Entre os softwares disponíveis atualmente, este trabalho analisou o desempenho do AIMMS e do LINGO na execução de um modelo de Programação Inteira de grande porte, devido à alta complexidade computacional exigida por esses problemas. . A análise foi realizada a partir de um modelo de Programação Inteira de designação, oriundo de um caso real aplicado à Agência Nacional de Aviação Civil (ANAC), com mais de um milhão de variáveis. A pesquisa realizada tem natureza aplicada, com objetivos exploratórios e abordagem mista, quantitativa e qualitativa. O problema foi modelado e executado nos softwares apresentando desempenhos satisfatórios. Assim, o estudo concluiu que ambos são softwares viáveis na execução de modelos de programação matemática inteira de grande porte. Nos testes, entretanto, ao aumentar a complexidade do modelo, observou-se que o AIMMS se sobressaiu ao LINGO. Verificou-se também que as funções extras e a interface visual do AIMMS contribuem para a usabilidade do software quando comparado ao LINGO, que apresenta uma interface mais básica. Este trabalho visa contribuir para a literatura de Pesquisa Operacional tendo em vista a escassez de estudos científicos que analisam a viabilidade e usabilidade dos softwares de programação matemática.

Palavras-chave: Software, AIMMS, LINGO, Programação Inteira.

## LISTA DE FIGURAS

Figura 1 – Processo de modelagem.....	22
Figura 2 – Interface AIMMS.....	44
Figura 3 – Interface AIMMS com a modelagem. ....	45
Figura 4 – Configurações do solver.....	46
Figura 5 – Resultado AIMMS solver CBC. ....	46
Figura 6 – Resultado AIMMS solver CPLEX. ....	47
Figura 7 – Resultado AIMMS solver ODH-CPLEX. ....	48
Figura 8 – Resultado AIMMS solver XA. ....	48
Figura 9 – Resultado AIMMS com disponibilidade 15. ....	50
Figura 10 – Resultado AIMMS com disponibilidade 10. ....	51
Figura 11 – Resultado AIMMS com disponibilidade 10 e <i>gap</i> mínimo. ....	52
Figura 12 – Interface LINGO. ....	53
Figura 13 – Interface LINGO com a modelagem.....	54
Figura 14 – Resultado LINGO. ....	55
Figura 15 – LINGO em execução.....	56
Figura 16 – Resultado LINGO com disponibilidade 15.....	57
Figura 17 – Resultado LINGO com disponibilidade 10.....	58
Figura 18 – Árvore de buscas com disponibilidade 20.....	59
Figura 19 – Árvore de buscas com disponibilidade 10.....	59
Figura 20 – Usabilidade de interface AIMMS. ....	60
Figura 21 – Comunidade de suporte AIMMS. ....	61
Figura 22 – Recursos AIMMS. ....	62
Figura 23 – Interface para usuário final.....	63



## LISTA DE QUADROS

QUADRO 1 – Características dos problemas de Programação Linear .....	20
QUADRO 2 – Protocolo da revisão sistemática .....	30
QUADRO 3 – Resultados da revisão sistemática .....	31
QUADRO 4 – Descrição dos índices.....	39
QUADRO 5 – Descrição dos parâmetros.....	39
QUADRO 6 – Descrição das variáveis.....	40
QUADRO 7 – Descrição do modelo .....	40
QUADRO 8 – Características gerais .....	64
QUADRO 9 – Comparação dos resultados.....	65

## **LISTA DE ABREVIATURAS E SIGLAS**

AIMMS – *Advanced Integrated Multidimensional Modeling Software*

ANAC – Agência Nacional de Aviação Civil

LINGO – *Language for Intective General Optimizer*

SOBRAPO – Sociedade Brasileira de Pesquisa Operacional

## SUMÁRIO

1	INTRODUÇÃO .....	12
1.1	Formulação do problema .....	13
1.2	Objetivo Geral .....	15
1.3	Objetivos Específicos .....	15
1.4	Justificativa .....	15
2	REFERENCIAL TEÓRICO .....	17
2.1	Pesquisa Operacional .....	17
2.1.1	Tomada de Decisão .....	19
2.2	Programação Matemática .....	20
2.2.1	Os Modelos .....	21
2.2.2	Programação Inteira e Binária .....	23
2.3	Softwares de Otimização Matemática .....	25
2.3.1	AIMMS .....	26
2.3.2	LINGO .....	28
2.4	Revisão Sistemática .....	29
2.4.1	Protocolo da Revisão Sistemática .....	30
2.4.2	Análise dos resultados .....	31
3	MÉTODOS E TÉCNICAS DE PESQUISA .....	34
3.1	Tipo e descrição geral da pesquisa .....	35
3.2	Procedimentos técnicos .....	36
3.3	Instrumentos de pesquisa .....	36
4	RESULTADOS E DISCUSSÃO .....	38
4.1	Descrição do modelo .....	38
4.1.1	Caracterização do modelo .....	38
4.1.2	Índices .....	39
4.1.3	Parâmetros .....	39
4.1.4	Variáveis .....	40
4.1.5	Função objetivo e restrições .....	40
4.2	Resultados Gerais .....	43
4.3	Resultados no AIMMS .....	43

4.3.1	Escolha do solver.....	45
4.3.2	Testes no AIMMS.....	49
4.4	Resultados no LINGO .....	52
4.4.1	Testes no LINGO .....	54
4.5	Árvore de valores do <i>branch-and-bound</i> .....	58
4.6	Usabilidade .....	60
4.6.1	AIMMS .....	60
4.6.2	LINGO.....	64
4.7	Compilação dos resultados.....	64
5	CONCLUSÕES E RECOMENDAÇÕES .....	66
5.1	Limitações e dificuldades da pesquisa.....	67
5.2	Sugestões para pesquisas futuras.....	68
	REFERÊNCIAS.....	69
	APÊNDICES.....	71
	Apêndice A – Modelo de Designação AIMMS.....	71
	Apêndice B – Modelo de Designação LINGO .....	72
	Apêndice C – Modelagem Completa AIMMS .....	75
	Apêndice D – Modelagem Completa LINGO.....	88

# 1 INTRODUÇÃO

A fim de entender a realidade e seus fenômenos, o homem mostra-se cada vez mais hábil na criação de metáforas e representação de problemas. Tendo em vista a dificuldade na resolução de questões complexas, Goldbarg e Luna (2005) esclarecem que os modelos matemáticos são inseridos como representações simplificadas da realidade, para que as conclusões obtidas de sua análise forneçam bases sólidas na tomada de decisão.

Mediante a Pesquisa Operacional, a formulação de modelos matemáticos permite a representação dos problemas de forma concisa, tornando sua estrutura geral mais compreensível. De maneira simplificada, a utilização de modelos matemáticos em programação matemática visa à maximização ou minimização de uma função-objetivo, a fim de encontrar uma solução ótima que respeite as restrições estabelecidas. Desse modo, as ferramentas computacionais possuem grande relevância na resolução dos problemas de programação matemática (HILLIER; LIEBERMAN, 2013).

O rápido avanço tecnológico e científico vivenciado nas últimas décadas viabilizou o desenvolvimento de softwares projetados para modelar e resolver tais problemas de otimização. Esses softwares podem se tornar importantes ferramentas de apoio à decisão quanto à gestão de operações. Belfiore e Fávero (2013) argumentam que o correto tratamento e análise dos dados propiciam ao tomador de decisão, informações precisas e confiáveis que, quando confrontadas com outras informações, oferecem o diferencial do conhecimento.

Entretanto, observa-se que o desconhecimento dessas ferramentas por parte do gestor resulta em decisões tomadas com base em experiência prévia, observação empírica e senso comum. O consenso geral sobre determinado fenômeno pode gerar conclusões corretas, porém um maior embasamento decorrente da utilização de modelos matemáticos e softwares capazes de resolvê-los pode propiciar uma melhora considerável na qualidade da decisão tomada, principalmente em ambientes competitivos (BELFIORE; FÁVERO, 2013).

Diante do exposto, o presente trabalho faz uma comparação entre os desempenhos dos softwares AIMMS e LINGO. O propósito é estudar o desempenho

de cada software na execução de um modelo matemático de grande porte, especificamente aplicado à administração pública, a fim de verificar suas potencialidades e gargalos, bem como aplicações futuras em Pesquisa Operacional.

Tanto o AIMMS quanto o LINGO podem ser obtidos gratuitamente para propósitos acadêmicos e são ensinados em diversos livros nacionais e internacionais de Pesquisa Operacional. Além disso, ambos os softwares apresentam um conjunto de interfaces de fácil utilização e não necessitam de um elevado nível de conhecimento em programação. De acordo com Ignácio e Ferreira Filho (2004), o AIMMS oferece um ambiente de desenvolvimento, no qual as pessoas experientes em modelagem podem criar aplicações funcionais, prontas para serem utilizadas por usuários finais. Baziewicz et al. (2013) esclarece que para a utilização do LINGO, é necessário que o usuário desenvolva certa afinidade com o programa e tenha noções básicas de programação.

Cabe mencionar que os softwares de programação matemática são ferramentas que proporcionam soluções aos problemas do dia a dia das organizações, com aplicações diversas na gestão logística e de projetos. Nesse sentido, os softwares de otimização matemática podem se tornar valiosas ferramentas de gestão tanto na administração pública quanto na iniciativa privada, garantindo maior eficiência e celeridade nos processos.

## **1.1 Formulação do problema**

Desde o advento da Revolução Industrial, o mundo presencia um crescimento extraordinário no tamanho e na complexidade das organizações, tornando-se cada vez mais difícil alocar os recursos disponíveis para as diversas atividades da maneira mais eficiente para a organização como um todo. A necessidade de encontrar o melhor caminho para solucionar esses problemas criou condições necessárias para o surgimento da Pesquisa Operacional (HILLIER; LIEBERMAN, 2013).

É necessário um grande volume de processamento de cálculos para o tratamento eficiente dos problemas complexos tipicamente considerados pela Pesquisa Operacional. Normalmente, fazer isso à mão seria uma hipótese fora de

cogitação. Portanto, o desenvolvimento de computadores munidos de excelentes pacotes de software com a capacidade de realizar cálculos matemáticos milhares ou até mesmo milhões de vezes mais rápido que o ser humano possibilita ao gestor empregar a programação matemática como estratégia na tomada de decisão (HILLIER; LIEBERMAN, 2013).

A natureza de muitos problemas de programação matemática sugere que eles possam ser formulados e resolvidos como um problema de Programação Linear (PL), Programação Inteira (PI) ou Programação Inteira Mista (PIM). Contudo, Longo (2004) aponta que a maioria dos softwares disponíveis até o momento, apresenta dificuldades para encontrar, em tempos computacionais aceitáveis, a solução ótima para problemas de PI e PIM de grande porte.

Um problema é classificado como Programação Inteira (PI) quando todas as variáveis de decisão são discretas (os valores representam um conjunto finito ou enumerável de números). Quando parte das variáveis de decisão é discreta e as demais são contínuas (assumem valores em um intervalo de números reais), o modelo é chamado de Programação Inteira Mista (PIM) (BELFIORE; FÁVERO, 2013).

Pode parecer que os problemas de Programação Inteira sejam relativamente fáceis de serem resolvidos, afinal, os problemas de Programação Inteira tem um número bem menor de soluções a serem consideradas. Entretanto, um número finito de soluções não garante que o problema seja prontamente solucionável. Considerando por exemplo o caso de Programação Inteira Binária, com  $n$  variáveis, há  $2^n$  soluções a serem consideradas, cada vez que  $n$  for incrementado em 1, o número de soluções dobra. Dessa maneira, até mesmo os computadores mais rápidos podem apresentar dificuldades de realizar uma enumeração exaustiva para verificar se cada uma das soluções é viável ou não e, em caso positivo calcular o valor da solução objetivo (HILLIER; LIEBERMAN, 2013).

Vale destacar, que os softwares de apoio à decisão se fundamentam fortemente nas técnicas de otimização, sua aplicação prática pode provocar dificuldades na fase de modelagem e no gerenciamento eficiente dos sistemas computacionais, o que muitas vezes inviabiliza o seu acesso por parte das organizações. Logo, faz-se necessário a utilização de programas que ofereçam um ambiente de desenvolvimento abrangente, no qual pessoas experientes em

modelagem matemática possam desenvolver aplicações práticas, prontas para serem usadas por pessoas com pouca experiência/usuários finais (IGNÁCIO; FERREIRA FILHO, 2004).

Com base nisso, este trabalho apresenta a seguinte questão de pesquisa: É viável a utilização dos Softwares AIMMS e LINGO na resolução de problemas de Programação Inteira de grande porte?

## **1.2 Objetivo Geral**

Analisar a viabilidade da utilização dos Softwares AIMMS e LINGO na resolução de um problema de Programação Inteira de grande porte e comparar os resultados.

## **1.3 Objetivos Específicos**

A fim de atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- 1) Caracterizar e descrever o problema de Programação Inteira utilizado para os testes, oriundo do projeto *Safety Oversight*;
- 2) Executar o problema no Software LINGO;
- 3) Programar e executar o problema no Software AIMMS;
- 4) Analisar diversas alternativas e métodos de resolução do modelo testado;
- 5) Construir uma tabela com os resultados.

## **1.4 Justificativa**

Entende-se que a Pesquisa Operacional (PO) é um tema em desenvolvimento, uma vez que se trata de uma área cuja evolução ocorre junto às



atualizações tecnológicas. A concepção de pesquisas acadêmicas nesse âmbito surge como solução em meio ao atual cenário econômico, político e social decorrente da necessidade de aperfeiçoamento dos processos gerenciais.

No contexto organizacional, em que muitas são as dúvidas e poucas as certezas, os modelos quantitativos permitem ao gestor a ampliação da capacidade de ação. Assim, o papel da Pesquisa Operacional é oferecer ferramentas poderosas para assistir ao tomador de decisão e protegê-lo de erros evitáveis. “Os modelos quantitativos não tomam as decisões, mas podem torná-las muito mais claras e fáceis” (GOLDBARG; LUNA, 2005).

Dado o ambiente altamente dinâmico das organizações atuais, as situações-problema vivenciadas cotidianamente, caracterizam-se como oportunidades para a aplicação da Pesquisa Operacional. Nesse sentido, o desenvolvimento de pesquisas acadêmicas com enfoque em softwares e modelos matemáticos capazes de solucionar problemas reais de grande porte colabora cientificamente na gestão organizacional.

Recursos tecnológicos como os softwares AIMMS e LINGO podem contribuir significativamente na resolução de problemas de programação matemática, uma vez que permitem que tais problemas sejam resolvidos em um tempo viável e admitem revisões nos modelos matemáticos quando houver necessidade.

A escolha dos softwares utilizados neste estudo se justifica em função dos recursos existentes em ambos, os quais são capazes de resolver modelos matemáticos de Programação Inteira e por apresentarem a possibilidade de utilização conjunta com bases de dados, como o EXCEL. A facilidade de obtenção da versão acadêmica também foi um fator considerado na escolha dos softwares. Além disso, o AIMMS e o LINGO são ensinados em bibliografias nacionais e internacionais e oferecem manuais ao usuário a fim de dirimir quaisquer dúvidas que possam surgir na sua utilização.

## 2 REFERENCIAL TEÓRICO

Denomina-se referencial teórico o capítulo do trabalho que tem por finalidade realizar a revisão de literatura, ou seja, apresentar estudos sobre o tema realizados por outros autores. É neste capítulo que se verifica o conhecimento do que já existe sobre o assunto abordado, oferecendo sustentação ao desenvolvimento do trabalho (SILVA; MENEZES, 2005).

Assim, esta sessão procura conceituar Pesquisa Operacional, programação matemática e Programação Inteira com base em livros e artigos relevantes para o problema de pesquisa a fim de elucidar o tema e contribuir com a análise e discussão dos resultados. Na sequência, são apresentados os softwares (AIMMS e LINGO) para fins de caracterização e descrição de suas particularidades visto que o conhecimento desses softwares é fundamental para a execução da pesquisa.

### 2.1 Pesquisa Operacional

O surgimento da Pesquisa Operacional como atividade formal geralmente é atribuído ao desenvolvimento das operações militares na Segunda Guerra Mundial. Em razão do empreendimento da Guerra, houve a necessidade de alocar de forma eficiente os escassos recursos para as diversas operações, utilizou-se então uma abordagem científica para lidar com os problemas de natureza tática e estratégica. Nesse contexto, o sucesso da Pesquisa Operacional despertou interesse na sua aplicação fora do ambiente militar (HILLIER; LIEBERMAN, 2013).

Desde então, a Pesquisa Operacional tem sido utilizada em uma variedade de problemas organizacionais oriundos dos setores público e privado, aplicada às mais diversas áreas de produção e logística. Assim, a aplicação da PO teve um grande impacto na melhoria da eficiência em inúmeras organizações pelo mundo gerando uma contribuição significativa no aumento da produtividade de diversos setores, desde a indústria até a prestação de serviços (ARENALES *et al.*, 2015).

A Sociedade Brasileira de Pesquisa Operacional (SOBRAPO, 2020) caracteriza a Pesquisa Operacional como uma ciência aplicada, voltada para a resolução de problemas reais. Através do uso de técnicas de modelagem matemática e algoritmos computacionais, a PO pode auxiliar o tomador de decisão na investigação dos mais variados aspectos e situações de um problema complexo a fim de encontrar as soluções que melhor servem aos objetivos dos indivíduos ou organizações. Goldberg e Luna (2005) acrescentam que as técnicas de PO podem ser utilizadas para alocar eficientemente os recursos limitados disputados por atividades alternativas.

Em termos gerais, a Pesquisa Operacional fundamenta-se na aplicação de um método científico (modelos matemáticos, estatísticos e algoritmos computacionais) para a tomada de decisão. Logo, a PO atua em um campo multidisciplinar, envolvendo áreas de administração, engenharia de produção, matemática aplicada e ciência da computação (BELFIORE; FÁVERO, 2013).

Hillier e Lieberman (2013) dividem a construção de um problema de PO em etapas. O primeiro passo constitui na observação cuidadosa do problema, incluindo a coleta de dados relevantes. A próxima etapa é a construção do modelo científico (tipicamente matemático) a fim de abstrair a essência do problema real. Assume-se então a hipótese de que esse modelo seja uma representação precisa da situação e que as conclusões obtidas também são válidas para o problema real. Uma característica adicional é que a Pesquisa Operacional tenta frequentemente encontrar uma melhor solução para o problema considerado (solução ótima).

Uma solução do modelo é “viável” se satisfizer todas as restrições – geralmente associadas à disponibilidade de recursos e requisitos específicos do problema, sendo considerada “ótima” se, além de viável, resultar no melhor valor (máximo ou mínimo) da função objetivo. Embora os modelos de PO sejam elaborados para “otimizar” um critério objetivo sujeito ao conjunto de restrições estabelecidas, a qualidade da solução resultante depende de quanto o modelo representa a realidade. Se o modelo representar bem o sistema, a sua solução vai ser ótima também para a situação real (TAHA, 2008).

A utilização das ferramentas de Pesquisa Operacional na promoção da eficiência organizacional em todos os níveis da gestão é uma realidade viabilizada pelo avanço computacional que proporciona uma melhor capacidade de cálculo,

armazenamento e recursos gráficos. Assim, o domínio das ferramentas computacionais aplicadas em PO torna-se essencial, especialmente pelos profissionais das áreas de ciências exatas e de administração (GOLDBARG; LUNA, 2005).

Arenales *et al.* (2015) explica que com o aprimoramento dos métodos de solução dos modelos e o desenvolvimento da informática, tem sido possível solucionar modelos de PO cada vez mais complexos, outrora intratáveis. No entanto, faz-se necessário certo domínio da teoria em que se baseia o método. Mesmo com a facilidade do uso dos softwares, o desconhecimento da abrangência e das suas limitações pode inibir seu uso na gestão organizacional ou conduzir a um uso equivocado.

### 2.1.1 Tomada de Decisão

Uma vez que a Pesquisa Operacional apoia-se na utilização de métodos quantitativos para corroborar a tomada de decisão, conforme definido anteriormente, é importante compreender melhor a relação entre a PO e a tomada de decisão. Para tanto, Goldberg e Luna (2005) explicam que existem outros aspectos a se considerar além do ideal quantitativo para desenvolver uma boa estratégia de gestão, principalmente relacionados ao fator humano. É exatamente reconhecendo isso que se pode melhor compreender a importância do amparo da Pesquisa Operacional na tomada de decisão.

O processo de tomada de decisão consiste em detectar um problema ou uma oportunidade e adotar uma linha de ação para solucioná-lo. Nesse âmbito, um problema ocorre quando a condição atual de uma situação é diferente da condição desejada, e a oportunidade ocorre quando as circunstâncias oferecem a chance de ultrapassar os objetivos e metas atuais (LACHTERMACHER, 2016).

Goldberg e Luna (2005) explicam que a tomada de decisão é o ato de selecionar, dentre várias decisões possíveis, a mais conveniente para o alcance de determinado objetivo. A finalidade da tomada de decisão no contexto organizacional é a maximização da utilidade, na prática traduzida pela maximização do lucro ou pela minimização do custo.

Para Belfiore e Fávero (2013), os objetivos da organização estão diretamente relacionados à tomada de decisão. Assim, o valor e a qualidade da informação que a organização dispõe tornam-se fundamentais na escolha da decisão eficiente entre as diversas alternativas disponíveis, minimizando as incertezas, os riscos e as complexidades inerentes ao processo. Lachtermacher (2016) acrescenta que outros fatores também podem afetar a tomada de decisão, tais como: importância da decisão, tempo disponível, ambiente, agentes decisores e conflito de interesses.

É com foco na tomada eficiente de decisão, levando em consideração as variadas interfaces dos sistemas e dos mercados, que a Pesquisa Operacional se estabelece como campo do conhecimento, tão indispensável em ambientes cada vez mais competitivos (BELFIORE; FÁVERO, 2013).

## 2.2 Programação Matemática

A programação matemática é muito indicada no gerenciamento de sistemas, especialmente no que diz respeito ao tratamento de variáveis quantificadas. A técnica permite a modelagem de inter-relações entre variáveis que dificilmente seriam percebidas de maneira intuitiva. Nesse caso, os softwares de programação matemática podem examinar inúmeras configurações viáveis do problema proposto pelo tomador de decisão e selecionar, dentro de certos critérios, as melhores alternativas (GOLDBARG; LUNA, 2005).

Por ser uma área muito abrangente, a programação matemática é subdividida dependendo do tipo das funções, restrições e variáveis utilizadas. Os problemas podem ser classificados como: Programação Linear (PL), Programação Inteira (PI), Programação Inteira Mista (PIM), Programação Binária (PB), Programação Binária Mista (PBM), e Programação Inteira Binária (PIB), conforme exemplifica o Quadro 1.

<b>Tipo do Modelo</b>	<b>Tipo de Variável</b>
Programação linear (PL)	Contínua
Programação linear inteira (PLI ou PI)	Discreta
Programação linear inteira mista (PLIM ou PIM)	Discreta e contínua
Programação linear binária (PLB ou PB)	Binária

Programação linear binária mista (PLBM ou PBM)	Binária e contínua
Programação linear inteira binária (PLIB ou PIB)	Discreta e binária

Fonte: Adaptado de Belfiore e Fávero (2013).

QUADRO 1 – Características dos problemas de Programação Linear

Em síntese, o adjetivo linear significa que as funções matemáticas desses modelos são necessariamente funções lineares. A programação linear é a base para a compreensão de outros modelos da programação matemática. Diversos métodos podem ser aplicados para a determinação da solução ótima em um modelo de PL, sendo o método Simplex, desenvolvido em 1947 por George B. Dantzig, um dos mais empregados (HILLIER; LIEBERMAN, 2013).

### 2.2.1 Os Modelos

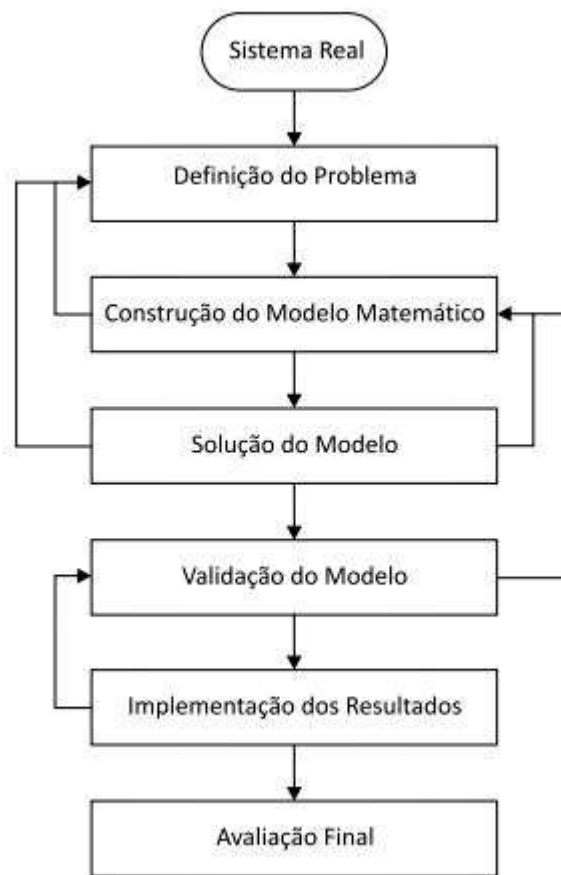
A Pesquisa Operacional reúne diversas das mais consagradas técnicas de modelagem matemática, seus modelos são estruturados de forma lógica, cujas grandezas são representadas por variáveis de decisão e as relações entre elas, por expressões matemáticas. Por estas características, os modelos matemáticos necessitam de informações quantificáveis e devem conter um conjunto suficiente de detalhes de maneira que os resultados atinjam suas necessidades. Assim, em situações nas quais um gestor precisa tomar uma decisão relativa a uma série de possibilidades conflitantes e concorrentes, uma das alternativas mais viáveis é a elaboração de modelos matemáticos da situação-problema (LACHTERMACHER, 2016).

Os principais modelos de PO são do tipo programação matemática. Nesse contexto, “programação” é entendida no sentido de planejamento. Apesar do termo geralmente ser associado ao processo de programação computacional, a expressão também é adequada para expressar o planejamento de atividades. Contudo, a Programação Matemática na maioria dos casos implica também em programação computacional (GOLDBARG; LUNA, 2005).

Hillier e Lieberman (2013) definem o modelo matemático de um problema de programação como o sistema de equações e de expressões matemáticas que

descrevem a essência do problema. Por exemplo, se houver  $n$  decisões a serem tomadas, elas serão representadas na forma de variáveis de decisão cujos valores respectivos devem ser determinados. A medida de desempenho é então expressa como uma função matemática dessas variáveis de decisão, chamada função objetivo. Quaisquer restrições nos valores que possam ser atribuídos a essas variáveis de decisão também são expressas de forma matemática, por meio de desigualdades ou equações.

Belfiore e Fávero (2013) dividem o processo de modelagem em fases. A primeira consiste na definição do problema, onde são estabelecidos os objetivos a serem alcançados. Em seguida, na fase de construção, são definidas as variáveis e as relações matemáticas. A solução consiste na aplicação de técnicas matemáticas para a resolução do modelo e o processo de validação pressupõe que as conclusões são válidas para o problema real. Decorre então a fase de implementação do modelo. Por fim, a avaliação final consiste em verificar se o objetivo foi alcançado, conforme exposto na Figura 1.



Fonte: Belfiore e Fávero (2013).

FIGURA 1 – Processo de modelagem.

Frequentemente uma análise mais detalhada do modelo mostra a necessidade de revisão. Equívocos não são incomuns, mesmo para os mais experientes modeladores. Algumas vezes a análise do modelo é suficiente para evidenciar suas inconsistências, outras vezes, somente após a operação do modelo na prática é possível detectar falhas, então o processo é repetido (ARENALES *et al.*, 2015).

De acordo com Hillier e Lieberman (2013), não existe apenas um único modelo “correto” para um problema de programação matemática, o processo de teste pode induzir a uma sucessão de modelos cada vez melhores e aprimorados. Até mesmo tipos diferentes de modelos matemáticos podem ser desenvolvidos para ajudar na análise de um mesmo problema.

### 2.2.2 Programação Inteira e Binária

Em inúmeras situações, as variáveis de decisão não poderão admitir um valor contínuo, um bom exemplo são os problemas envolvendo a alocação de pessoas e máquinas, nesses casos as soluções fracionárias perdem o sentido prático, tornando necessária a adição de uma condição de integridade para as variáveis encontradas.

A Programação Inteira é comumente empregada em problemas de gerenciamento de fluxo de transportes, como em Costa (2019) que estudou a Programação Inteira para o aprimoramento logístico no roteamento de veículos de uma transportadora. O autor propõe a utilização de modelos matemáticos de Programação Inteira com o objetivo de minimizar o tempo total de percurso e reduzir o número de veículos usados, alcançando um resultado satisfatório para a organização. Koroishi (2019) também demonstra em seu estudo que a Programação Inteira é viável no gerenciamento de transporte aéreo, obtendo como resultado um programa de otimização global apto para a aplicação real.

Hillier e Lieberman (2013) apontam que se a exigência de valores inteiros for a única maneira pela qual uma situação pode ser problematizada, então trata-se de um problema de Programação Inteira (PI). O modelo matemático para a Programação Inteira corresponde ao modelo de programação linear com uma



restrição adicional de que as variáveis sejam discretas, ou seja, assumam valores inteiros.

Quando todas as variáveis de decisão do modelo são discretas, o problema é classificado como Programação Inteira pura. Quando parte das variáveis de decisão é discreta e as demais são contínuas, o modelo é chamado Programação Inteira mista (PIM) (BELFIORE; FÁVERO, 2013).

Hillier e Lieberman (2013) esclarecem a importância dos problemas envolvendo decisões do tipo “sim ou não”. Em tais decisões, as únicas escolhas possíveis são sim e não, como por exemplo, os problemas de designação de pessoas. Com apenas duas opções, é possível representar o problema por variáveis de decisão que estão restritas apenas a dois valores, representadas por “1” se o evento ocorre e “0” se o evento não ocorre. Tais variáveis são chamadas variáveis binárias. Assim, problemas de PI contendo apenas variáveis binárias podem ser chamados de Programação Inteira Binária (PIB).

Pode parecer que os problemas de Programação Inteira sejam relativamente fáceis de serem resolvidos, afinal os problemas de PI têm um número finito de soluções a serem consideradas. Entretanto, um número finito de soluções não garante que o problema seja prontamente solucionável, uma vez que números finitos podem ser astronomicamente grandes. Considerando, por exemplo, o caso simples dos problemas de PIB com  $n$  variáveis, há  $2^n$  soluções a serem consideradas, sendo assim, cada vez que  $n$  for incrementado em 1, o número de soluções dobra. Esse padrão é conhecido como crescimento exponencial da dificuldade do problema (HILLIER E LIEBERMAN, 2013).

Uma peculiaridade da maior parte das técnicas de PO é que as soluções são determinadas por algoritmos. Um algoritmo fornece regras de cálculo fixas que são aplicadas repetidas vezes ao problema, sendo que em cada repetição (denominada iteração) a solução fica mais próxima de se tornar ótima. Como os cálculos associados a cada iteração costumam ser numerosos, o uso da computação torna-se essencial (TAHA, 2008).

A teoria da complexidade computacional distingue dois tipos de função, que representam um algoritmo “eficiente” e um “não eficiente”. Um algoritmo “eficiente” tem função de complexidade polinomial, o algoritmo “não eficiente” é caracterizado

por uma função exponencial, cujo crescimento explosivo representa uma barreira na resolução quanto ao tempo necessário para resolvê-lo (ARENALES *et al.*, 2015).

Em função disso, Belfiore e Fávero (2013) apontam que muitos problemas de PI e PIB são NP-completos, isto é, não podem ser resolvidos em tempo polinomial dado a alta complexidade computacional exigida. Em alguns casos, uma alternativa para encontrar a solução dos problemas de PI é o arredondamento da solução obtido após o relaxamento das condições de integralidade, recaindo em um problema de programação linear. Porém, o arredondamento pode resultar em uma solução inviável e não há garantia de que a nova solução arredondada seja a solução ótima do problema.

### **2.3 Softwares de Otimização Matemática**

O desenvolvimento da Pesquisa Operacional e a sua maior utilização nos últimos anos, nos mais variados campos do conhecimento, deve-se, entre outros fatores, ao avanço computacional e ao desenvolvimento dos softwares voltados para a modelagem e solução de diferentes tipos de problema.

Segundo Hillier e Lieberman (2013), com a linguagem de modelagem, os softwares auxiliam na formulação dos modelos matemáticos de grande porte de forma eficiente. Mesmo com milhares de restrições em um modelo, normalmente elas são de poucos tipos e seguem o mesmo padrão, isso ocorre de modo semelhante com as variáveis de decisão. Assim, usando bancos de dados, uma linguagem de modelagem é capaz de formular todas as restrições do mesmo tipo. Além disso, os softwares podem adiantar uma série de tarefas de gerenciamento como a transformação de dados em parâmetros e a análise de soluções a partir do modelo.

Em geral, os programas de otimização como o AIMMS e o LINGO são formados pela união de dois softwares, distintos um do outro, mas complementares nas suas funções. O primeiro é o solver, entre eles - CPLEX, LP/MIP, XPRESS e GUROBI, que recebe o problema de programação matemática como dado de entrada e aplica um ou mais métodos de otimização com o auxílio de algoritmos matemáticos, como Simplex, Branch and Bound e outros, retornando os valores

ótimos por meio de iterações em uma sequência finita de operações (D'ANDREA, 2011).

O segundo software é a interface, cuja função básica é gerar a matriz a partir de um formato, podendo ser uma linguagem de modelagem algébrica ou até mesmo uma planilha eletrônica. As interfaces geralmente possuem instrumentos para preparação de dados, formulação/alteração de modelos e análises de resultados (D'ANDREA, 2011).

Nesse sentido, alguns pontos relevantes devem ser observados na escolha de um software de otimização matemática, como a conectividade, isto é, a capacidade do software de se relacionar com outro programa e base de dados — em particular o Excel, pois este se apresenta como uma ferramenta amplamente utilizada na maioria das organizações e oferece suporte para a gestão de bases de dados.

A interface gráfica também é um fator a ser observado, uma vez que a visão mais didática e ampla do modelo pode auxiliar o usuário na análise do problema. Outro critério identificado na literatura são os manuais referentes aos softwares, que devem ser completos e de fácil entendimento. Por fim, a robustez do software, ligada a estabilidade na resolução de problemas complexos deve ser considerado, principalmente no escopo deste trabalho.

### 2.3.1 AIMMS

O software AIMMS (*Advanced Integrated Multidimensional Modeling Software*), desenvolvido pela empresa Paragon Decision Technology, é uma linguagem de modelagem algébrica de alto nível que resolve problemas complexos de programação linear, não linear e inteira (BELFIORE; FÁVERO, 2013).

A linguagem de modelagem do AIMMS oferece uma notação de índice poderosa e intuitiva que captura a complexidade dos problemas à medida que eles ocorrem. Por meio do explorador de modelos gráfico, os dados são tornados visíveis como nós em uma árvore. Além disso, sua linguagem de modelagem permite expressar cálculos complexos de maneira compacta, sem a necessidade de se

preocupar com o gerenciamento de memória e armazenamento de dados. O software também possibilita a conectividade com o EXCEL para importação e exportação de dados (AIMMS, 2020).

Ignácio e Ferreira filho (2004) adicionam que a notação de índice do AIMMS possibilita ao usuário capturar a complexidade de problemas reais através de uma linguagem que utiliza como instrução o Solver. No que diz respeito à criação de um sistema de apoio à decisão, o AIMMS fornece todo um conjunto de interfaces gráficas que proporciona, ao usuário final, uma visualização do problema de forma abrangente.

Na literatura de Pesquisa Operacional, o software AIMMS não é tão difundido quanto o LINGO. Observam-se poucas menções apenas a título de conhecimento, como no livro “Pesquisa Operacional para cursos de engenharia” de Belfiore e Fávero (2013). Entretanto, o AIMMS tem sido amplamente utilizado como software de otimização em empresas ao redor do mundo. Além disso, o AIMMS também disponibiliza licença acadêmica gratuita para alunos e professores, sem limite de capacidade.

A utilização do software AIMMS pode ser observada nos estudos científicos recentes de Pesquisa Operacional. Em 2019, no XIX Simpósio de Pesquisa Operacional e Logística da Marinha, foram apresentados trabalhos com aplicação do AIMMS em problemas diversos, dentre eles: “Rebalanceamento do sistema de bicicletas compartilhadas: experimento realizado na cidade do Rio de Janeiro” (PIERROTTI; SILVA JÚNIOR, 2019) e “Otimização da rede de transporte de suprimentos do Exército Brasileiro na região da Amazônia Oriental” (CARVALHO; SILVA JÚNIOR, 2019).

A aplicação do AIMMS também está presente no estudo de Cabral (2019): “Aplicação do algoritmo DSM a determinação do posicionamento de bases descentralizadas do SAMU/Natal com uso de simulação”. O objetivo de aplicar um modelo matemático para definir a localização ótima das bases descentralizadas do SAMU no município de Natal/RN foi viabilizado pela utilização do software AIMMS, que se mostrou bastante adequado para a resolução do problema.

### 2.3.2 LINGO

O LINGO (*Language for Intective General Optimizer*) é uma ferramenta abrangente projetada para a construção e a resolução de modelos de otimização Linear, Não Linear e Inteira. A linguagem de modelagem do LINGO permite expressar modelos de maneira intuitiva e direta, usando somatórios e variáveis subscritas (LINDO, 2020).

Trata-se de um produto da Lindo Systems, e consiste em uma poderosa linguagem de modelagem genérica. Um recurso notável do LINGO é sua flexibilidade em lidar com uma ampla gama de problemas de PO, utilizando conjuntos como seu componente básico fundamental (HILLIER; LIEBERMAN, 2013).

Sua linguagem direta e a interface “limpa” são características que podem tornar o LINGO um software atrativo na resolução de problemas de otimização. Além disso, o software pode ler um arquivo de forma interativa permitindo a entrada de dados de diferentes maneiras (BAZIEWICZ *et al.*, 2013).

Da mesma forma, o LINGO pode enviar informações da solução diretamente para um banco de dados ou planilha, facilitando a geração de relatórios no aplicativo de escolha do usuário. Outra facilidade proporcionada pelo software é que não se faz necessária a especificação do solver por parte do usuário, o próprio LINGO faz uma leitura da formulação inserida e escolhe automaticamente o algoritmo de otimização adequado (LINDO, 2020).

Rehfeldt (2009) acrescenta que o LINGO auxilia na identificação das causas dos possíveis erros. Ao encontrar inconsistências no modelo, é possível refazer o modelo matemático. No caso de erros ou *debugs*, o software também apresenta um *help*, com uma série de recursos capazes de auxiliar o usuário a encontrar a causa do erro, se tornando uma ferramenta interessante principalmente na análise de modelos de grande porte.

Ademais, é importante acrescentar que o software LINGO está presente em grandes literaturas de PO, como no livro “Introdução à Pesquisa Operacional” de Hillier e Lieberman (2013), cuja abordagem não se limita a uma simples menção de seus atributos, nele os autores ensinam a linguagem de modelagem LINGO de forma detalhada com diversos exemplos aplicados. Winston e Goldberg (2004) em

seu livro “*Operations research: applications and algorithms*” também ensinam a resolver diferentes tipos de problema através do LINGO, assim como realizar análises por meio de seus resultados. Cabe citar que o LINGO possui licença acadêmica gratuita com capacidade irrestrita para estudantes e professores do mundo inteiro.

Estudos recentes mostram a utilização do software LINGO em diversos setores. No artigo “Planejamento de Lavra: um estudo de caso” (Silva Júnior, 2019), apresentado no Simpósio Brasileiro de Automação Inteligente, foi implementado um modelo de programação matemática no LINGO a fim de minimizar os desvios de meta de produção, os desvios de qualidade e a quantidade de caminhões necessários ao processo de mineração a partir da programação linear inteira mista. Os resultados encontrados validaram o modelo, e conseqüentemente, o software como ferramenta para subsidiar o operador no processo de tomada de decisão.

## 2.4 Revisão Sistemática

A fim de verificar o estado da arte em relação às pesquisas comparativas entre softwares de programação matemática, em especial entre o AIMMS e o LINGO, foi realizada uma revisão sistemática de literatura nas seguintes bases de dados: Google Acadêmico, *Science Direct*, *Emerald* e CAPES. A escolha das bases de dados ocorreu em função da credibilidade e abrangência em nível nacional e internacional.

A revisão sistemática de literatura constitui o meio de obtenção dos subsídios para a prática baseada em evidência. É uma metodologia rigorosa, proposta para identificar os estudos sobre determinado tema, aplicando métodos explícitos e sistematizados de busca (De-la-Torre-Ugarte-Guanilo; Takahashi; Bertolozzi, 2011).

Os resultados alcançados através da investigação sistemática permitem ao pesquisador identificar lacunas e oportunidades de pesquisas futuras. É uma ferramenta potente para dar continuidade à pesquisa quantitativa e endereçar problemas complexos, além de identificar novos campos de pesquisa e áreas em mudança do estágio de maturidade, conduzindo o estudo a partir dos limites do conhecimento sobre o tema (LOUREIRO *et al.*, 2016).

### 2.4.1 Protocolo da Revisão Sistemática

A pesquisa foi realizada com base no protocolo metodológico de revisão sistemática de literatura proposto por Cronin, Ryan e Coughlan (2008). Nesta metodologia, a fim de garantir a confiabilidade da pesquisa, deve-se seguir cinco passos:

- 1) Formulação da questão de pesquisa;
- 2) Conjunto de critérios de inclusão e exclusão;
- 3) Seleção e acesso de literatura;
- 4) Avaliação da qualidade da literatura incluída na revisão;
- 5) Análise, síntese e disseminação dos resultados.

As cinco etapas supracitadas foram definidas e sintetizadas no Quadro 2.

<b>Protocolo sugerido por Cronin, Ryan e Coughlan (2008).</b>	<b>Execução</b>
Formulação da questão de pesquisa	A comparação entre os softwares AIMMS e LINGO já foi realizada em pesquisas científicas?
Critérios de inclusão e exclusão	Banco de dados: Google Acadêmico, Science Direct, Emerald e Capes Período de publicações: 2010-2020 Palavras-chave: AIMMS, LINGO Tipos de artigo: Periódicos Citações e patentes foram descartadas
Seleção e acesso de literatura	Banco de dados: Google Acadêmico, Science Direct, Emerald e Capes
Avaliação da qualidade da literatura	Número total de periódicos encontrados: Google Acadêmico (176) Science Direct (22) Emerald (01) Capes (08) Número de artigos que seguem os critérios: Google Acadêmico (2) Science Direct (0) Emerald (0) Capes (0)

Análise, síntese e disseminação dos resultados	O objetivo foi encontrar periódicos que continham as palavras-chave, excluindo citações e patentes, que fizessem uma análise comparativa dos softwares. Foram considerados artigos em português, inglês e espanhol. Os resultados estão apresentados no quadro de resultados.
--	---

Fonte: Elaborado pela autora (2020).

QUADRO 2 – Protocolo da revisão sistemática

## 2.4.2 Análise dos resultados

Como mencionado na seção anterior, foram definidas quatro bases de dados a fim de abranger o maior número de artigos e periódicos. A busca pelos termos “AIMMS” e “LINGO” no título do artigo não retornou resultados em nenhuma das bases de dados. Dessa forma, a busca se limitou a artigos com as palavras-chave presentes no corpo do texto.

Todas as pesquisas foram feitas considerando trabalhos dos últimos 10 anos (de 2010 a 2020), revisados por pares, excluindo patentes e citações, sem restrições quanto ao campo de estudo, como forma de aumentar a probabilidade de encontrar trabalhos relevantes para esta pesquisa. O Quadro 3 sintetiza os resultados encontrados:

Base de dados	Quantidade de artigos encontrados	Quantidade de artigos que seguem os critérios	Nome dos artigos
Google Acadêmico	176	02	“Utilização de softwares na Pesquisa Operacional”
			“Comparação de desempenho e usabilidade entre os softwares comerciais de otimização e o método dual para o problema clássico do transporte”
<i>Science Direct</i>	22	0	
Capes	08	0	
<i>Emerald</i>	01	0	



Fonte: Elaborado pela autora (2020).

#### QUADRO 3 – Resultados da revisão sistemática

Após a avaliação do título e resumo dos trabalhos, constatou-se que nenhum possuía conteúdo comparativo entre os softwares AIMMS e LINGO, visto que a maioria deles abordaram os softwares apenas de forma descritiva para o conhecimento do leitor a respeito de suas funcionalidades na execução de problemas de Pesquisa Operacional.

Entretanto, dois artigos encontrados no Google Acadêmico fizeram análises comparativas diversas. No artigo “Utilização de Softwares na Pesquisa Operacional”, Baron (2017) realiza uma comparação entre os softwares AIMMS e GAMS. Nesse estudo a autora executa um problema de minimização de pequeno porte denominado de “problema do incinerador”. O problema do incinerador consiste em minimizar o custo total com a coleta e destino do lixo de duas cidades.

Em seu estudo, Baron (2017) sistematiza a resolução do problema no AIMMS e no GAMS desde o início, com um passo a passo explicando as principais funcionalidades dos softwares. Posteriormente o artigo apresenta os resultados do problema, em ambos os softwares o resultado foi o mesmo.

O artigo utilizou um problema de pequeno porte para comparar o AIMMS a outro software. Em suas considerações finais, Baron (2017) destaca que o estudo serve como base para o conhecimento dos softwares e suas aplicações em Pesquisa Operacional.

O segundo artigo: “Comparação de desempenho e usabilidade entre os softwares comerciais de otimização e o método dual para o problema clássico do transporte” de Baziewicz *et al.* (2013) trata-se de um estudo comparativo mais robusto entre os softwares LINGO, CPLEX e o método *Stepping-Stone*. A análise consistiu em verificar, a partir de um problema de transporte, requisitos como: desempenho, tempo computacional, eficácia e usabilidade entre os softwares.

Para a realização da análise comparativa, foram geradas dezesseis instâncias aleatórias, variando quantidade de pontos de origens, destinos, demandas e ofertas. A análise consistiu em avaliar o tempo computacional gasto, a eficácia, a solução ótima e possíveis desvios, bem como avaliar a interface, aprendizagem e a linguagem de programação necessária para a utilização (BAZIEWICZ *et al.* 2013).

Em seus resultados, verificou-se que a eficácia dos três métodos é a mesma, tendo em vista que todos resultaram na solução ótima. Entretanto, o LINGO demonstrou o pior desempenho em relação aos demais, principalmente devido às peculiaridades do próprio software, cuja linguagem de programação se diferencia do CPLEX, que é fundamentalmente um solver e da ferramenta *Stepping-Stone* que foi desenvolvida sem a utilização de software específico. Quanto à dificuldade de utilização dos softwares, para a utilização LINGO e do CPLEX, é necessário que o usuário tenha noções básicas de programação, já o método *Stepping-Stone* demanda um maior nível de linguagem e conhecimentos específicos de Pesquisa Operacional e programação (BAZIEWICZ *et al.* 2013).

Os autores chegaram à conclusão que cada tipo de problema de programação matemática possui suas peculiaridades. Desse modo, a escolha do melhor método de resolução depende das necessidades do usuário, do conhecimento técnico em programação e das características do problema (BAZIEWICZ *et al.* 2013).

A partir da revisão sistemática, verifica-se uma lacuna existente na literatura de Pesquisa Operacional quanto à análise comparativa entre softwares, em especial entre o AIMMS e o LINGO. Pesquisas desse tipo podem se tornar instrumentos valiosos para embasar a escolha do software mais adequado a ser utilizado por gestores que desejam adotar a Pesquisa Operacional em seus processos gerenciais.

### **3 MÉTODOS E TÉCNICAS DE PESQUISA**

O conhecimento científico tem como característica fundamental a sua verificabilidade. Para que um conhecimento possa ser considerado científico, é necessário identificar as operações mentais e técnicas que possibilitam sua verificação. Surge então a necessidade de determinar quais são os procedimentos realizados que geraram as conclusões a respeito de um determinado assunto, de forma que a sua verificação seja possível com a replicação da metodologia (GIL, 2008).

Pode-se definir o método como o caminho para chegar a determinado fim, e o método científico como o conjunto de procedimentos técnicos e intelectuais adotados para se atingir o conhecimento. Método corresponde a soma de atividades racionais que permitem alcançar conhecimentos válidos e verdadeiros, os quais proporcionam as bases lógicas da investigação. São os procedimentos que deverão ser seguidos no processo de investigação científica, identificando possíveis erros e embasando a tomada de decisão (GIL, 2008; LAKATOS; MARCONI, 2003).

A metodologia deve estar presente desde o planejamento até as conclusões da pesquisa. Silva e Menezes (2005) defendem que uma mesma pesquisa pode conter diversas classificações, desde que obedeça às condições inerentes a cada tipo.

Tendo em vista a importância da metodologia para o desenvolvimento deste trabalho, o atual capítulo apresenta os métodos e técnicas empregados para a elaboração da pesquisa, bem como as suas classificações de acordo com a literatura analisada. Pretende-se, portanto, deixar claro toda a metodologia utilizada para o alcance dos objetivos.

### 3.1 Tipo e descrição geral da pesquisa

Para Gil (2008), a pesquisa tem um caráter pragmático, é o processo formal e sistemático de desenvolvimento do método científico, seu objetivo é encontrar a solução para um problema.

Nesse sentido, a pesquisa científica consiste na realização de uma investigação planejada e desenvolvida de acordo com as normas consagradas pela metodologia científica. A pesquisa desenvolvida possui natureza aplicada, a qual objetiva gerar conhecimentos para aplicação prática e conduzidos à solução de problemas específicos, envolvendo verdades e interesses locais (SILVA; MENEZES, 2005).

Do ponto de vista dos objetivos, trata-se de uma pesquisa exploratória. De acordo com Gil (2008), a pesquisa exploratória tem como finalidade desenvolver, esclarecer e modificar conceitos e ideias, considerando a formulação de questões mais precisas ou hipóteses pesquisáveis para estudos posteriores. Em geral, na pesquisa exploratória há um levantamento bibliográfico e documental, assim, o produto final desse processo é um problema mais esclarecido, passível de investigação mediante procedimentos mais sistematizados. Esse tipo de pesquisa ocorre especialmente quando o assunto do estudo é pouco explorado, muitas vezes constituindo-se a primeira etapa de uma investigação mais ampla.

Em relação à abordagem, esta pesquisa pode ser classificada como uma pesquisa qualitativa e quantitativa, concentrando-se na objetividade, a partir da análise de dados. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno e as relações entre as variáveis, para isso a pesquisa quantitativa requer o uso de recursos e técnicas estatísticas. Nesse tipo de pesquisa é considerada a possibilidade de se quantificar os dados, ou seja, transformar em números as informações que serão coletadas e analisadas posteriormente (SILVA; MENEZES, 2005).

Quanto à viabilidade dos softwares, a pesquisa comparou os resultados do AIMMS e do LINGO nos testes com vistas a ressaltar as diferenças e similaridades entre eles.

## 3.2 Procedimentos técnicos

Para o alcance dos objetivos propostos, o trabalho aplicou em um primeiro momento a revisão de literatura, com a finalidade de esclarecer os principais conceitos necessários ao entendimento do estudo. Silva e Menezes (2005) enfatizam que a revisão de literatura deverá elucidar o tema, proporcionar melhor definição do problema de pesquisa e contribuir para a análise dos resultados.

A revisão de literatura geralmente desenvolve-se a partir de livros e artigos científicos. Sua principal vantagem reside no fato de permitir o acesso a uma gama de fenômenos mais ampla acerca do tema abordado. Assim, a análise de literatura é parte indispensável de pesquisas que demandam informações dispersas e dados secundários (GIL, 2008).

Em um segundo momento, a partir da análise documental, foi selecionado um projeto de programação matemática inteira de grande porte oriundo de um estudo de caso real. As informações coletadas foram disponibilizadas a partir de dados secundários, que apesar de terem sido coletados para outra finalidade, serviram de insumo para ilustrar a problemática abordada nesta pesquisa.

## 3.3 Instrumentos de pesquisa

O modelo matemático utilizado neste trabalho trata-se de um projeto realizado em parceria entre a Universidade de Brasília (UnB) e a Agência Nacional de Aviação Civil (ANAC) para atender às demandas de fiscalização da Agência, denominado *Safety Oversight* (2019). Trata-se de um modelo de grande porte com 1000913 variáveis, sendo 24618 variáveis inteiras.

O projeto contou com um total de 156 inspetores distribuídos em 16 unidades da federação, ou seja, 16 nós de origem. Quanto aos nós de destino, totalizaram 116 aeroportos, também espalhados pelo território brasileiro. As missões foram estabelecidas considerando todos os 693 tipos de especialidades de acordo com a demanda de fiscalização da ANAC para cada aeroporto.

Os arcos origem-destino ou destino-destino são caracterizados por um custo de viagem aérea. A coleta desses custos foi realizada em 2019 através de uma rotina desenvolvida utilizando a linguagem de programação *Python3* e a biblioteca *web browser Selenium*.

Cada missão é caracterizada por uma atividade requerida, em um determinado destino, formada por uma equipe de número variado de pessoas, com um tempo de duração também variado. Desse modo, o formato da equipe e a duração para cada atividade/missão são dados de entrada que precisam ser inseridos antes do modelo ser executado.

O Excel foi utilizado para o tratamento desses dados, já que ambos os softwares possuem conectividade com a planilha eletrônica. Assim, os dados foram computados e agrupados de acordo com as necessidades encontradas no decorrer do processo de modelagem e execução de cada software. Pretendeu-se com isso, a importação e exportação automática por meio dos programas, considerando que se trata de um volume muito grande de dados.

O modelo foi implementado no AIMMS versão 4.68 e no LINGO 18.0. A *performance* computacional foi obtida em um computador com processador Intel Core i5, 2.50GHz com, 2 núcleos, 4 processadores lógicos e 8GB de memória RAM.

## 4 RESULTADOS E DISCUSSÃO

Nesta sessão foram apresentados os resultados dos desempenhos de cada software. O modelo foi executado integralmente, de acordo com os dados do projeto original, posteriormente alguns parâmetros como a disponibilidade dos servidores e a sensibilidade dos softwares foram alterados para uma análise mais completa.

### 4.1 Descrição do modelo

#### 4.1.1 Caracterização do modelo

O modelo matemático e o caso analisado nesta pesquisa são aplicados especificamente à Superintendência de Padrões Operacionais da ANAC (SPO). Buscou-se alocar eficientemente os servidores para as missões de fiscalização de modo a otimizar os custos de deslocamento, tendo em vista que os profissionais estão distribuídos pelo território brasileiro.

A ANAC é uma autarquia federal, integrante da administração pública indireta vinculada ao Ministério da Infraestrutura. Foi criada para regular e fiscalizar as atividades da aviação civil e a infraestrutura aeronáutica e aeroportuária no Brasil. Em seu programa de missões de fiscalização, atua como certificadora e é responsável pela vistoria em aeronaves, aeroportos e pela auditoria de empresas, para assim garantir níveis aceitáveis de segurança e de qualidade na prestação dos serviços aos passageiros (ANAC, 2020).

Nas missões de fiscalização, os profissionais podem ser designados a qualquer ponto de demanda espalhados pelo país. A alocação de profissionais acontece do nó de origem (UF onde o servidor reside) até o nó de destino (aeroporto) onde realizarão a fiscalização, esses pontos podem estar situados em qualquer uma das 26 unidades federativas. No âmbito da Pesquisa Operacional, esse modelo pode ser caracterizado como um Problema de Designação.

Hillier e Lieberman (2013) descrevem o Problema de Designação como um tipo especial de PL em que os designados são indicados para a realização de tarefas. O modelo matemático para o Problema de Designação apresenta-se da seguinte forma:

$$x_{ij} = \begin{cases} 1 & \text{se o designado } i \text{ realiza a tarefa } j; \\ 0 & \text{se o designado } i \text{ não realiza a tarefa } j. \end{cases}$$

Logo, o modelo utilizado caracteriza-se também como um problema de Programação Inteira binária.

#### 4.1.2 Índices

O modelo é composto pelos seguintes índices:

N	Nós	Conjunto de origens e destinos
I	Subconjunto de N	Origens em N
J	Subconjunto de N	Destinos em N
O	Origem	Unidades Federativas
D	Destino	Aeroportos
K	Missão	Atividade demandada
P	Pessoa	Servidores da ANAC
T	Período	Mensal
GR	Grupo	Capacitações necessárias

Fonte: Adaptado do Projeto *Safety Oversight*, ANAC (2019).

QUADRO 4 – Descrição dos índices

#### 4.1.3 Parâmetros

O quadro 5 mostra os parâmetros do modelo:

$Custo_{i,j}$	Custo da passagem aérea
---------------	-------------------------



$Tempo_{i,j}$	Tempo de viagem
$Duração_k$	Duração da missão
$Equipe_k$	Equipe necessária para a missão
$Disponibilidade_{p,t}$	Disponibilidade em dias do servidor
$Demanda_{t,k,gr,d}$	Demanda da missão
$Oferta_{p,gr,o}$	Local de origem do servidor

Fonte: Adaptado do Projeto *Safety Oversight*, ANAC (2019).

QUADRO 5 – Descrição dos parâmetros

#### 4.1.4 Variáveis

O modelo é composto por quatro variáveis de decisão, sendo duas inteiras binárias, e duas contínuas, como pode ser visto no Quadro 6.

$Atendida_{t,k,gr,d,p}$	Binária	Indica a pessoa (P) que atendeu a missão (K) do grupo (GR) no destino (D) no período (T)
$Alocado_{t,i,j,p,gr,k}$	Contínua	Indica o fluxo da pessoa (P) de um nó (I) para um nó (J) do grupo (GR) que atendeu a missão (K) no período (T)
$AlocadoGeral_{t,i,j,p}$	Contínua	Indica o fluxo da pessoa (P) de um nó (I) para um nó (J) no período (T)
$TempoUtilizado_{p,t}$	Binária	Indica o tempo utilizado pela pessoa (P) no período (T)

Fonte: Adaptado do Projeto *Safety Oversight*, ANAC (2019).

QUADRO 6 – Descrição das variáveis

#### 4.1.5 Função objetivo e restrições

A Função objetivo e as restrições do modelo estão esquematizadas no Quadro 7:

Função Objetivo	$\text{Minimizar } \sum_{t,i,j,p} \text{AlocadoGeral}_{t,i,j,p} * 2 * \text{Custo}_{i,j}$
Restrição 1	$\sum_k \text{Alocado}_{t,o,d,p,gr,k} \leq \text{Oferta}_{p,gr,o}$ $\forall t \in T, d \in D, p \in P, gr \in GR, k \in K$
Restrição 2	$\sum_i \text{Alocado}_{t,i,d,p,gr,k} = \text{Atendida}_{t,k,gr,d,p} + \sum_j \text{Alocado}_{t,d,j,p,gr,k}$ $\forall t \in T, d \in D, p \in P, gr \in GR, k \in K$
Restrição 3	$\sum_i \text{Alocado}_{t,i,d,p,gr,k} \geq \sum_j \text{Alocado}_{t,d,j,p,gr,k}$ $\forall t \in T, d \in D, p \in P, gr \in GR, k \in K$
Restrição 4	$\sum_p \text{Atendida}_{t,k,gr,d,p} = \text{Equipe}_k * \text{Demanda}_{t,k,gr,d}$ $\forall t \in T, d \in D, p \in P, gr \in GR, k \in K$
Restrição 5	$\sum_{k,gr,d} \text{Atendida}_{t,k,gr,d,p} * \text{Duração}_k + \sum_{i,j} \text{AlocadoGeral}_{t,i,j,p} * 2$ $* \text{Tempo}_{i,j} = \text{TempoUtilizado}_{p,t}$ $\forall t \in T, p \in P$
Restrição 6	$\text{TempoUtilizado}_{p,t} \leq \text{Disponibilidade}_{p,t}$ $\forall t \in T, p \in P$
Restrição 7	$\sum_k \text{Atendida}_{t,k,gr,d,p} \leq 1$ $\forall t \in T, gr \in GR, d \in D, p \in P$
Restrição 8	$\text{AlocadoGeral}_{t,i,j,p} \geq \text{Alocado}_{t,i,j,p,gr,k}$ $\forall t \in T, i \in N, j \in N, p \in P, gr \in GR, k \in K$
Restrição 9	$\text{AlocadoGeral}_{t,i,j,p} \in \mathbb{R}^+$ $\text{Alocado}_{t,i,j,p,gr,k} \in \mathbb{R}^+$

	$TempoUtilizado_{p,t} \in \mathbb{R}^+$ $Atendida_{t,k,gr,d,p} \in \mathbb{Z}^+\{0,1\}$
--	--

Fonte: Adaptado do Projeto *Safety Oversight*, ANAC (2019).

#### QUADRO 7 – Descrição do modelo

A modelagem do problema de alocação deve proporcionar uma solução ótima que supra a demanda de inspeções por localidade e prevê a minimização dos custos de deslocamento para cada missão. Assim, a função objetivo é formada pela soma dos custos da alocação geral dos servidores multiplicada por dois, tendo em vista que esse valor considera o preço da passagem de ida e volta dos servidores.

A primeira restrição corresponde à restrição de oferta, nela somente será alocada a pessoa que existir na origem O, capacitada no grupo GR.

As restrições 2 e 3 fazem o balanço de fluxo, assim, cada pessoa que chega de qualquer nó da rede, em cada destino D, atende uma missão K ou pode sair para atender outras missões K, garantindo também que nenhum fluxo será gerado nos nós de destinos, visto que só pode sair de um nó de destino um fluxo menor ou igual ao que chega nele (restrição 3). Essas restrições permitem o transbordo do servidor, ou seja, o aproveitamento da viagem para atender mais de uma missão naquele destino, ou em outro destino.

As restrições 4 e 7 referem-se à formação das equipes de acordo com cada demanda de missão e a formação de equipes de especialistas com generalistas.

As restrições 5 e 6 relacionam-se ao tempo, a primeira gera a variável que indica quanto tempo foi utilizado considerando a missão que o servidor atendeu e o tempo de vigem para atendê-la, enquanto a segunda indica o tempo utilizado para cada servidor dado a sua disponibilidade em dias.

A restrição 8 existe a fim de impedir que uma mesma pessoa gere várias variáveis caso ela venha a atender mais de uma missão.

A última restrição limita os conjuntos das variáveis, determinando se ela é não negativa, inteira ou binária.

## 4.2 Resultados Gerais

O projeto concluído no AIMMS apresentou 1000913 variáveis, com 24618 variáveis inteiras e as demais, contínuas. No LINGO foram 1001176 variáveis sendo 24879 inteiras. A solução foi a mesma nos dois programas: R\$ 27100,00. A diferença entre o número de variáveis é devido a forma como cada software lê as variáveis, a função objetivo e as restrições.

Ambos os softwares realizaram a importação de dados por meio do Excel. No AIMMS não foi necessário exportar os dados para a leitura das respostas, devido a facilidade de interpretação no próprio programa e no LINGO a exportação foi feita para o bloco de notas.

A partir dos resultados encontrados observou-se que o modelo possui múltiplas soluções ótimas, portanto como não era o escopo do trabalho analisar o output do modelo, o estudo dedicou-se a analisar a função objetivo e os aspectos particulares de cada software antes, durante e após a execução.

Nos testes, observou-se que o número de iterações aumentava conforme se elevava a complexidade do problema. Em todos os testes o LINGO fez mais iterações que o AIMMS, porém, isso não prejudicou sua eficácia, resultando na mesma função objetivo.

Com o intuito de contribuir com o conhecimento e aprendizagem da modelagem matemática nos softwares, a modelagem completa foi disponibilizada nos apêndices deste projeto.

## 4.3 Resultados no AIMMS

O modelo completo executado no AIMMS consta no apêndice A e foi intitulado “Modelo de Designação AIMMS”, a Figura 2 mostra a interface inicial do AIMMS.

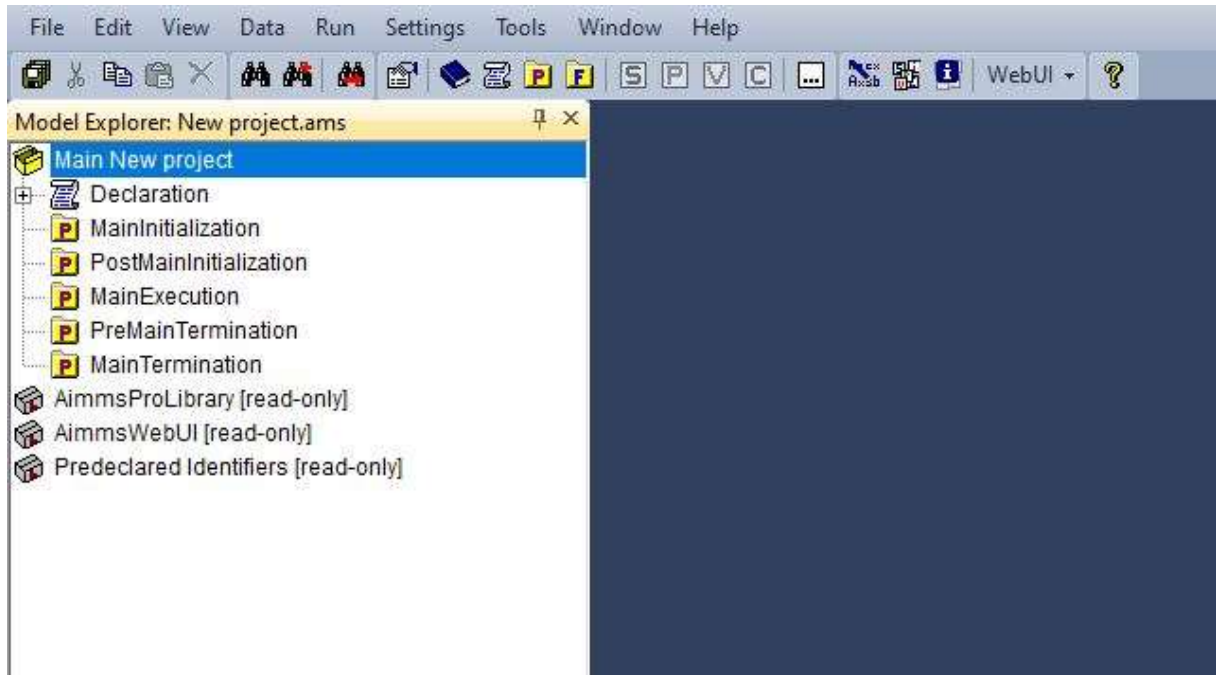
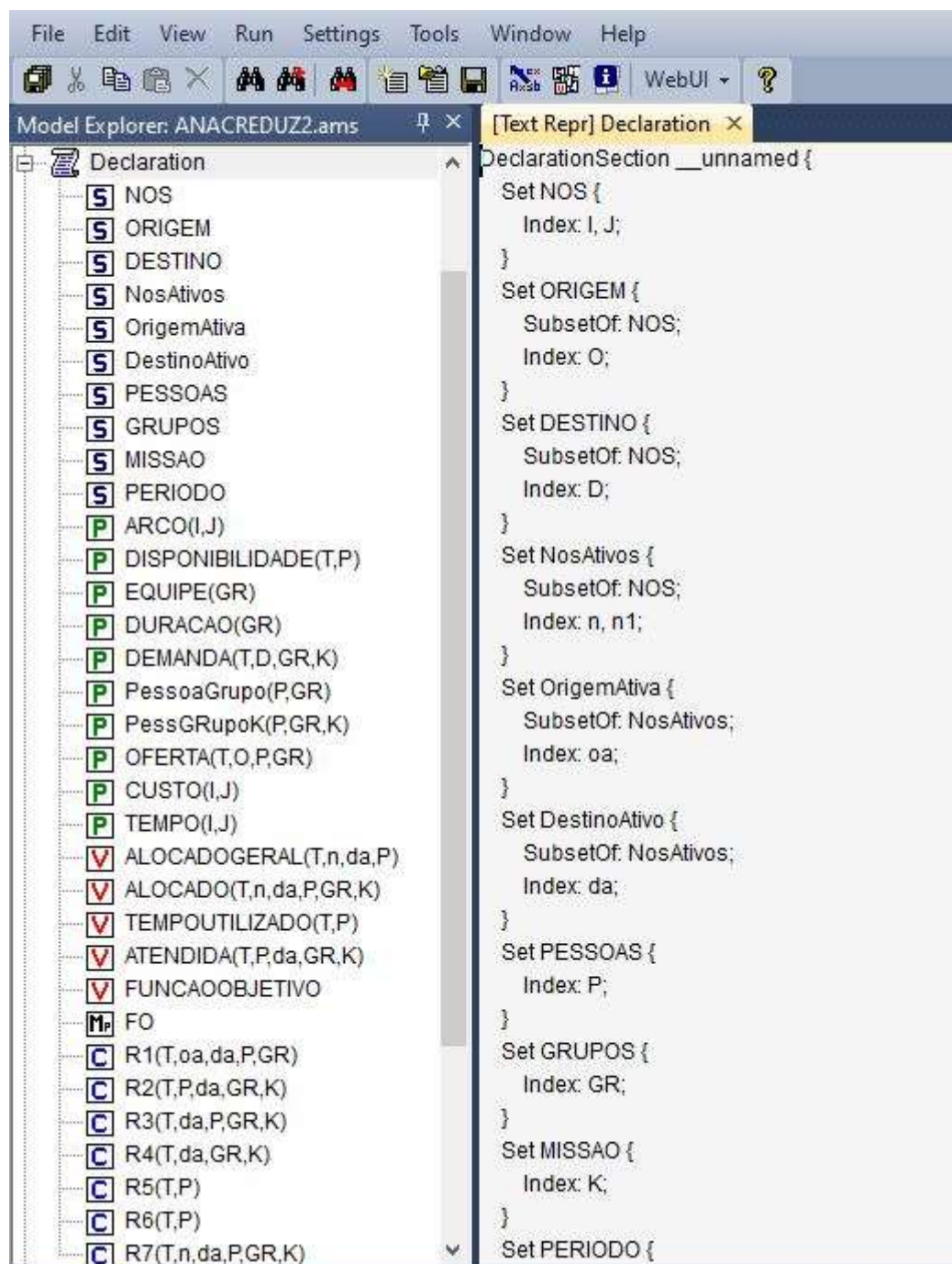


FIGURA 2 – Interface AIMMS

A Figura 3 mostra a interface do software com a modelagem do problema.



Fonte: Elaborado pela autora (2020).

FIGURA 3 – Interface AIMMS com a modelagem

#### 4.3.1 Escolha do solver

O AIMMS possibilita ao usuário escolher, dentre as opções disponíveis, qual solver utilizar na execução do modelo. A Figura 4 mostra os solvers disponíveis e os tipos de problemas que eles resolvem.

 Solver configuration

Description	LP	MIP	NLP	QP	MIQP	QCP	MIQCP	MCP	MINLP	COP
AOA					x		x		<b>X</b>	
CBC 2.9	x	x								
CONOPT 4.0	x		<b>X</b>	x		x				
CPLEX 12.9	<b>X</b>	<b>X</b>		<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>			
CPOptimizer 12.9										<b>X</b>
IPOPT 3.11	x		x	x		x				
MINOS	x		x	x		x				
ODH-CPLEX 4.0		x			x		x			
PATH 4.7								<b>X</b>		
SNOPT 7.2	x		x	x		x				
XA 16	x	x		x						

FIGURA 4 – Configurações do solver.

A configuração relevante para este trabalho é a *Mixed-Integer Programming* (MIP), referente à Programação Inteira. Conforme mostra a Figura 4, o AIMMS dispõe de quatro solvers aptos a executarem os modelos de Programação Inteira. Assim, o modelo foi testado integralmente nos quatro solvers a fim de verificar qual deles apresentava o melhor desempenho para os testes posteriores.

Inicialmente, o modelo foi executado utilizando o solver CBC 2.9:

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : CBC 2.9
Phase            : MIP
Iterations       : 13395
Nodes            : 142
Best LP Bound    : 27100.0048 (Gap: 0.00%)
Best Solution    : 27100.0048
Solving Time     : 104.72 sec (Peak Mem: 2726.3 Mb)
Program Status   : Optimal
Solver Status    : Normal completion

Total Time       : 176.66 sec
Memory Used      : 1797.6 Mb
Memory Free      : 18326.2 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 5 – Resultado AIMMS solver CBC.

Com tempo computacional de 176,66 segundos, função objetivo igual à R\$ 27100,00 e 1797.6 megabytes de memória utilizada

O modelo com as mesmas características foi executado no CPLEX 12.9, retornando também o valor ótimo de R\$ 27100,00 como resultado do custo mínimo das passagens aéreas. O tempo computacional exigido foi de 82,64 segundos e 1965.7 megabytes de memória, como pode ser visto na Figura 6.

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : CPLEX 12.9
Phase            : MIP
Iterations       : 18069
Nodes            : 0 (Left: 1)
Best LP Bound    : 27100.00377 (Gap: 0.00%)
Best Solution    : 27100.005
Solving Time    : 13.83 sec (Peak Mem: 754.4 Mb)
Program Status   : Optimal
Solver Status    : Normal completion

Total Time       : 82.64 sec
Memory Used      : 1965.7 Mb
Memory Free      : 18989.6 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 6 – Resultado AIMMS solver CPLEX.

A Figura 7 mostra os resultados no solver ODH-CPLEX 4.0, com tempo de resolução de 92,75 segundos e memória igual a 2671.7 megabytes.



```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : ODH-CPLEX 4.0
Phase            : MIP
Iterations       : 0          (Threads: 2)
Nodes            : 34
Best LP Bound    : 27100.00374 (Gap: 0.00%)
Best Solution    : 27100.0052
Solving Time     : 20.16 sec  (Peak Mem: 1897.3 Mb)
Program Status   : Optimal
Solver Status    : Normal completion

Total Time       : 92.75 sec
Memory Used      : 2671.7 Mb
Memory Free      : 8759.6 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 7 – Resultado AIMMS solver ODH-CPLEX

O solver XA 16 não conseguiu resolver o modelo, como mostra a Figura 8.

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : XA 16

Solving Time     : 0.05 sec  (Peak Mem: 0.0 Mb)
Program Status   : Error no solution
Solver Status    : Error: Solver failure (see message window)

Total Time       : 76.41 sec
Memory Used      : 3515.1 Mb
Memory Free      : 16676.3 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 8 – Resultado AIMMS solver XA.

O solver que demonstrou o melhor desempenho para o modelo trabalhado foi o CPLEX 12.9, tendo em vista que resolveu o problema no menor tempo computacional, utilizando pouca memória. Assim, ele foi escolhido para executar os demais testes.

#### 4.3.2 Testes no AIMMS

Para a comparação de desempenho entre os softwares, foram realizados os mesmos testes no AIMMS e no LINGO.

Na seção anterior, o modelo foi executado integralmente no AIMMS, com a disponibilidade de 20 dias, no CPLEX 12.9, retornando o valor ótimo de R\$ 27100,00 como resultado do custo mínimo das passagens aéreas e tempo computacional de 82,64 segundos.

Visto que o software executou o modelo com certa facilidade, optou-se por aumentar a complexidade do problema diminuindo a disponibilidade dos servidores. A figura 9 mostra o resultado do software com a disponibilidade de 15 dias. A função objetivo encontrada foi R\$ 27128,00.

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : CPLEX 12.9
Phase            : MIP
Iterations       : 17915
Nodes            : 0 (Left: 1)
Best LP Bound    : 27128.00454 (Gap: 0.00%)
Best Solution    : 27128.0064
Solving Time    : 11.72 sec (Peak Mem: 900.1 Mb)
Program Status  : Optimal
Solver Status    : Normal completion

Total Time       : 79.59 sec
Memory Used      : 2205.7 Mb
Memory Free     : 10020.6 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 9 – Resultado AIMMS com disponibilidade 15.

Em diversas tentativas de executar o modelo com disponibilidades menores, observou-se que o software apresentou dificuldades, demorando horas para resolver o problema ou simplesmente não o executava. A solução encontrada foi aumentar o *gap* entre a “*Best LP Bound*” e a “*Best Solution*”.

Com o aumento do *gap*, o software encontra uma solução ótima próxima àquela que ele poderia encontrar com o *gap* igual a 0, porém em um tempo computacional menor, e com uma distância da melhor solução de até X% de *gap*. A “*Best LP Bound*” é o melhor valor objetivo possível que o software teoricamente poderia alcançar. A “*Best Solution*” é a melhor solução inteira viável encontrada até o momento (AIMMS, 2020).

Dessa forma, a figura 10 mostra o modelo executado com a disponibilidade de 10 dias e *gap* de 5%.

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : CPLEX 12.9
Phase            : MIP
Iterations       : 198988
Nodes            : 492          (Left: 366)
Best LP Bound    : 27799.0048   (Gap: 3.01%)
Best Solution    : 28663.0082
Solving Time     : 52.31 sec    (Peak Mem: 823.3 Mb)
Program Status   : Optimal
Solver Status    : Normal completion

Total Time       : 127.84 sec
Memory Used      : 2040.1 Mb
Memory Free      : 9908.5 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 10 – Resultado AIMMS com disponibilidade 10.

O AIMMS chegou ao resultado de R\$ 28663,00. O próprio software conseguiu diminuir o *gap* para encontrar uma solução melhor. Neste caso, a “*Best Solution*” apresentou uma diferença de cerca de 3% do valor da “*Best LP Bound*”.

O AIMMS executou o modelo com *gaps* menores para essa disponibilidade, até o limite de 2%, encontrando assim um valor ótimo menor, de R\$ 28504,00, que pode ser observado na Figura 11.

```

Math.Program      : FO
# Constraints     : 1636852
# Variables      : 1000913 (24618 integer)
# Nonzeros       : 4937153
Model Type       : MIP
Direction        : minimize

SOLVER           : CPLEX 12.9
Phase            : MIP
Iterations       : 1846424
Nodes            : 6810 (Left: 2579)
Best LP Bound    : 27934.83849 (Gap: 2.00%)
Best Solution    : 28504.0068
Solving Time    : 583.20 sec (Peak Mem: 1144.2 Mb)
Program Status  : Optimal
Solver Status    : Normal completion

Total Time      : 604.52 sec
Memory Used     : 2575.3 Mb
Memory Free     : 9231.3 Mb

```

Fonte: Elaborado pela autora (2020).

FIGURA 11 – Resultado AIMMS com disponibilidade 10 e *gap* mínimo.

Nesse caso, com o *gap* menor, a *Best Solution* reduziu para R\$ 28504,00. É interessante observar que a *Best LP Solution* também aumentou. Como mencionado anteriormente, esse indicador não quer dizer necessariamente que a solução ótima global seja exatamente esse valor, ele apenas é um limite mínimo que o software poderia encontrar caso continuasse procurando soluções.

Pode-se concluir, portanto, que para um desempenho melhor, o aumento do *gap* é uma opção viável. Em contrapartida, isso significa que a melhor solução encontrada pode não ser a solução ótima global do problema, uma vez que quanto maior for o *gap*, maior será a distância da possível solução ótima.

#### 4.4 Resultados no LINGO

O modelo completo executado no LINGO consta no apêndice B e foi intitulado “Modelo de Designação LINGO”, a Figura 12 mostra a interface inicial do software.

Pode-se perceber que o software tem uma interface mais simples, como uma folha em branco, onde o modelo será inserido.

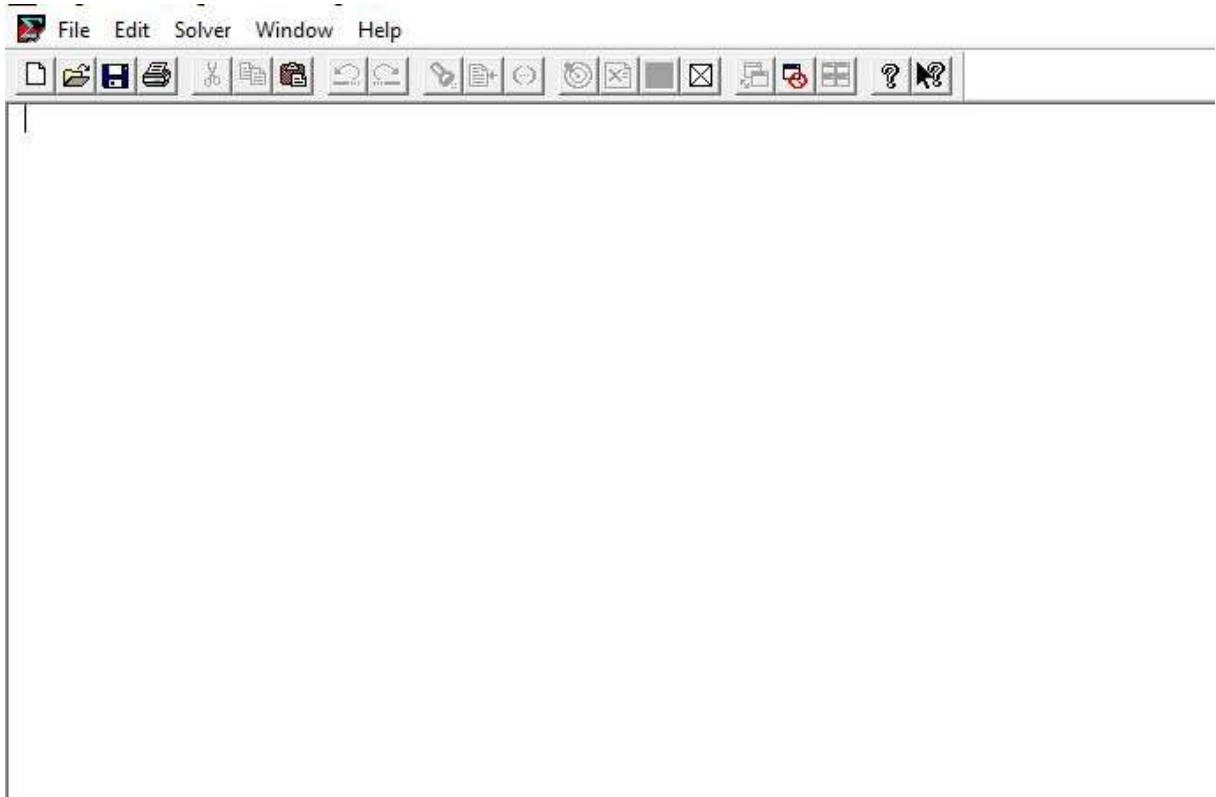



FIGURA 12 – Interface LINGO.

A interface com a modelagem pode ser observada na Figura 13.



```

Sets:
  nos/@ole()/;
  origem(nos)/@ole()/:OrigemAtiva;
  destino(nos)/@ole()/:DestinoAtivo;
  Arco(nos,nos)/@ole()/:Custo,tempo;

  Periodo/1/;
  missao/@ole()/;
  Pessoas/@ole()/;
  grupo/@ole()/:duracao,equipe,GrupoAtivo;
  PessoaAtiva(Pessoas);

  DATIVO(destino)|DestinoAtivo(&l)#GE#1;
  OATIVA(origem)|OrigemAtiva(&l)#GE#1;
  NOSATIVOS(nos)|@in(OATIVA,nos(&l))#OR#@in(DATIVO,nos(&l));
  ARCOA(nosativos,nosativos);
  ARCOATIVO(nosativos,nosativos);

  TPe(pessoas,periodo):tempoutilizado,disponibilidade;
  pessoaquebrada(pessoas,grupo)/@ole()/;
  PO(pessoas,origem)/@ole()/;
  NEPO(pessoas,origem)|#not#@in(PO,&l,&2);
  PGM(pessoas,grupo,missao);

  ExisteMissao(missao,grupo)/@ole()/;
  GRATIVO(grupo)|GrupoAtivo(&l)#GE#1;
  PGATIVO(pessoas,grupo);

  moradia(pgativo,oativa):oferta;
  dem(periodo,missao,grupo,destino)/@ole()/:demanda;

  ArcoMi(periodo,arcoativo,PGM):Alocado;
  NoPeGr(periodo,missao,grupo,destino,pessoas):Atendida;
  VarInt(periodo,arcoativo,pessoas):AlocadoGeral;

```

Fonte: Adaptado do Projeto Safety Oversight, ANAC (2019).

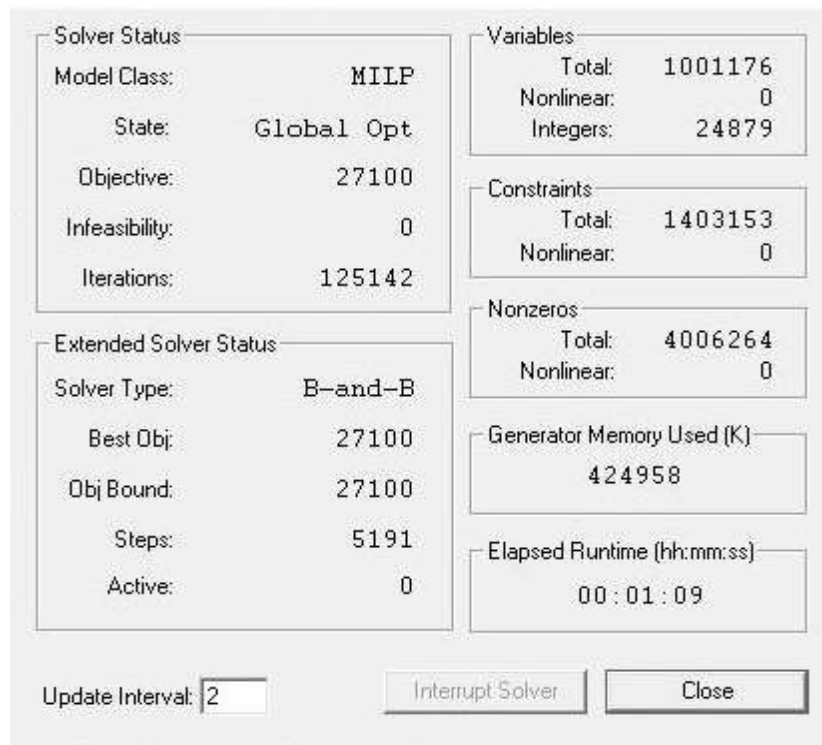
FIGURA 13 – Interface LINGO com a modelagem.

O LINGO não permite ao usuário definir qual solver utilizar na resolução dos modelos, portanto, não haverá a seção de “Escolha do solver” para os resultados do LINGO.

#### 4.4.1 Testes no LINGO

Primeiramente, o modelo foi executado com a disponibilidade dos servidores de 20 dias no mês para atender a demanda com 118 missões. A função objetivo do

modelo resultou em um gasto mínimo com passagens aéreas de R\$ 27100,00 (mesmo resultado do AIMMS), com tempo computacional de 1,9 minutos (69 segundos) e 125142 interações. O resultado pode ser visto na figura 14.

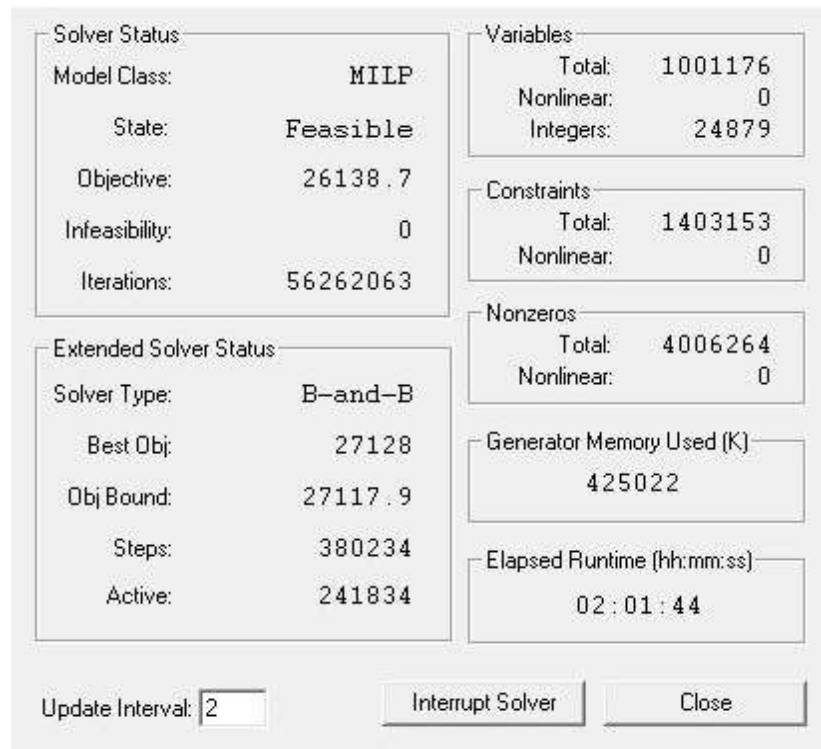


Fonte: Adaptado do Projeto Safety Oversight, ANAC (2019).

FIGURA 14 – Resultado LINGO

Ao reduzir a disponibilidade dos servidores, o software encontrou dificuldades para encontrar a solução ótima com as mesmas configurações. A Figura 15 mostra o problema ainda em execução rodando com a disponibilidade de 15 dias. Observa-se que o tempo de execução do software para esse modelo aumentou significativamente.



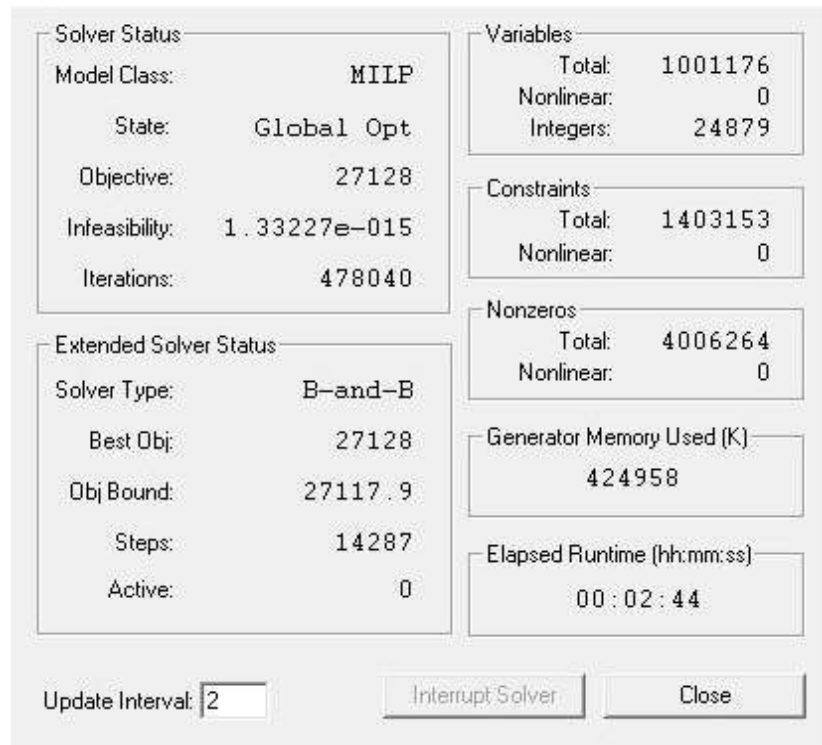


Fonte: Adaptado do Projeto Safety Oversight, ANAC (2019).

FIGURA 15 – LINGO em execução.

No caso do LINGO, a solução encontrada também foi diminuir o *gap* (*optimality tolerance*), essa redução pode ser vantajosa na medida em que reduz o tempo de solução do problema, entretanto, esse relaxamento pode gerar uma solução ótima local e não global, dependendo do *gap* selecionado. A obtenção de uma solução entre 1% e 5% do valor ótimo global em modelos inteiros de grande porte pode se tornar bastante atraente tendo em vista a grande diminuição dos tempos de execução (LINDO, 2020).

Pode ser observado na figura 16 o problema totalmente executado, com o *gap* de 0,1%.

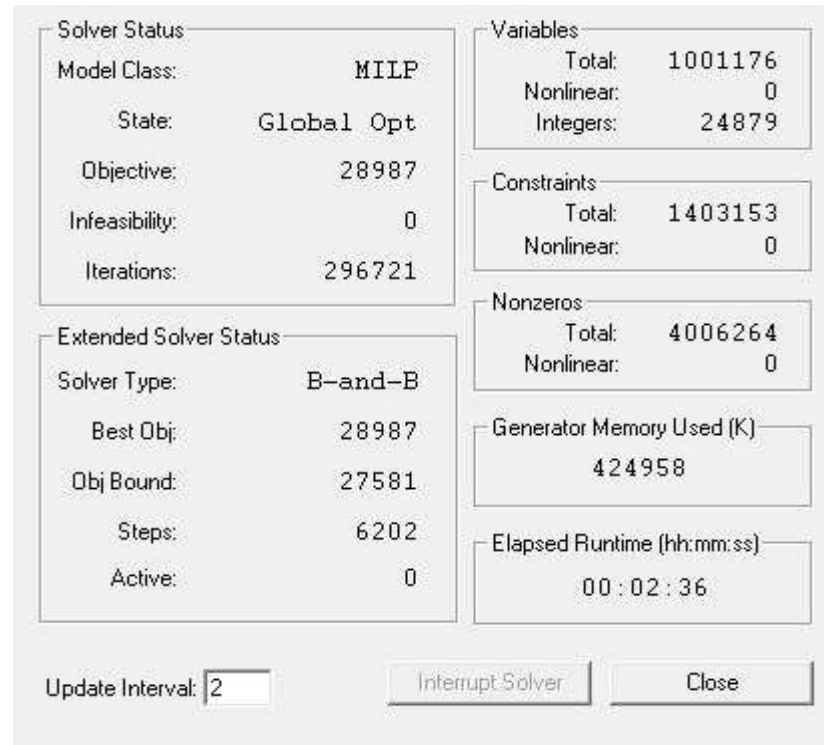


Fonte: Adaptado do Projeto Safety Oversight, ANAC (2019).

FIGURA 16 – Resultado LINGO com disponibilidade 15.

Pode-se perceber que o valor encontrado para a disponibilidade de 15 dias foi de R\$ 27128,00, valor muito próximo ao *objective bound* (R\$ 27117.9), correspondente ao limite mínimo que a solução pode alcançar. Esses dois valores podem estar muito próximos, mas a solução não pode passar desse objetivo. O fato de esses valores estarem próximos significa que a melhor solução atual do LINGO é a solução ideal ou está muito próxima a ela (LINDO, 2020).

Na execução do modelo com a oferta ainda mais reduzida, com a disponibilidade de 10 dias, o LINGO encontrou mais dificuldade para executá-lo em tempo hábil. Assim, foi necessário aumentar o *gap* para 5%. Os dados obtidos encontram-se na Figura 17.



Fonte: Adaptado do Projeto Safety Oversight, ANAC (2019).

FIGURA 17 – Resultado LINGO com disponibilidade 10.

Verifica-se que o valor obtido de R\$ 28987,00 está a 5% do limite inferior (R\$ 27581,00).

Para essa mesma oferta, o LINGO não conseguiu executar o problema em tempo hábil com o *gap* menor que 5%.

#### 4.5 Árvore de valores do *branch-and-bound*

Belfiore e Fávero (2013) explicam que o algoritmo *branch-and-bound*, utilizado na resolução do modelo, é um método de divisão e conquista em que o problema é ramificado em subproblemas menores. Resolve-se o problema para estas instâncias menores, assim, as soluções dos subproblemas são combinadas até que se obtenha a solução ótima do problema original.

Para um problema de minimização, a resolução parte de um limite superior que pode ser obtido por meio de heurísticas e de um limite inferior, obtido pela relaxação do problema, eliminando uma ou mais restrições. A figura 18 ilustra o problema com a disponibilidade de 20 dias por meio da árvore de buscas no AIMMS:

Variable Statistics		Constraint Statistics		Matrix Statistics	Matrix View
Variable Solution		Constraint Solution		Math Program Solution	MIP Search Tree
Node Number	Best LP Bound	Best Integer	Gap [%]	Time [s]	
Relaxation at root	na	na	na	0.05	
0	27100	1.15e+08	99.98	5.69	
0	27100	30184	10.22	5.83	
0	27100	28338	4.37	6.85	
0	27100	27889	2.83	12.05	
0	27100	27100	0.00	14.05	
Completed	27100	27100	0.00	14.05	

Fonte: Elaborado pela autora (2020).

FIGURA 18 – Árvore de buscas com disponibilidade 20.

Observa-se que a “*Best LP Bound*” parte de um limite inferior, enquanto a “*Best Integer*” parte do limite superior e os cálculos são interrompidos quando a solução atinge o mesmo valor, nesse caso: 27100.

A figura 19 ilustra a árvore de buscas com a disponibilidade de 10 dias e *gap* mínimo de 2%. Para esse problema, como foi definido o *gap* de 2%, o programa interrompe os cálculos quando a diferença entre os dois valores atinge o *gap*.

Variable Statistics		Constraint Statistics		Matrix Statistics	Matrix View
Variable Solution		Constraint Solution		Math Program Solution	MIP Search Tree
Node Number	Best LP Bound	Best Integer	Gap [%]	Time [s]	
Relaxation at root	27100	na	na	0.04	
0	27513.7	7.207e+07	99.96	4.39	
0	27521.5	4042144	99.32	6.31	
0	27521.5	29622	7.09	7.19	
0	27521.5	29141	5.56	12.39	
241	27521.5	29133	5.53	30.84	
341	27536	29133	5.48	35.88	
929	27825.5	28828	3.48	61.17	
1001	27842.9	28828	3.42	67.42	
1101	27853.2	28828	3.38	72.28	
1967	27899	28822	3.20	105.89	
2067	27902.1	28504	2.11	110.70	
2299	27935.8	28504	1.99	125.02	
Completed	27935.8	28504	1.99	125.02	

Fonte: Elaborado pela autora (2020).

FIGURA 19 – Árvore de buscas com disponibilidade 10.

Pode-se constatar, portanto, que ao reduzir a oferta (disponibilidade), o algoritmo precisa aprofundar as buscas para encontrar a solução, demandando mais tempo computacional.

## 4.6 Usabilidade

### 4.6.1 AIMMS

A boa usabilidade de um software pode significar ganhos em produtividade. A partir da experiência e aprendizados adquiridos na modelagem do problema no AIMMS e sua validação em diferentes etapas, foi constatado que o usuário precisa ter experiência com a modelagem matemática e conhecimento das funcionalidades do software.

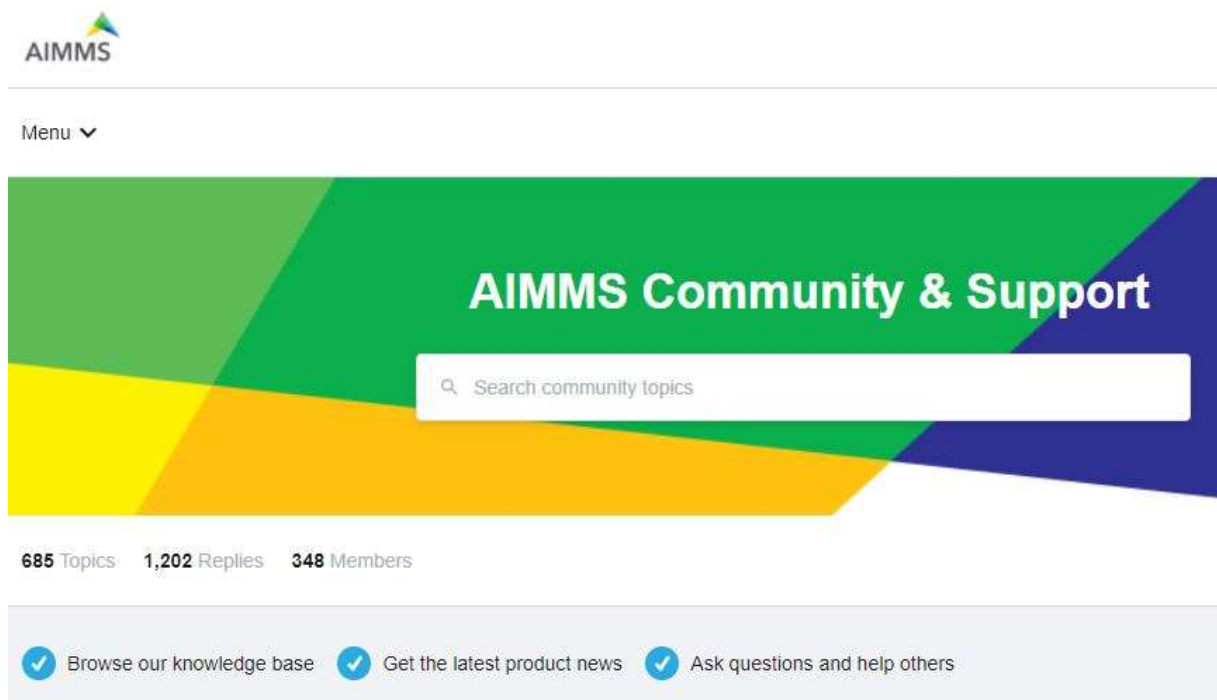
Sua interface é dinâmica e intuitiva, a separação dos índices, parâmetros, variáveis e restrições em conjuntos, permite a melhor visualização dos componentes do modelo, conforme explicitado na Figura 20.

Declaration	
<input type="checkbox"/>	NOS
<input type="checkbox"/>	ORIGEM
<input type="checkbox"/>	DESTINO
<input type="checkbox"/>	NosAtivos
<input type="checkbox"/>	OrigemAtiva
<input type="checkbox"/>	DestinoAtivo
<input type="checkbox"/>	PESSOAS
<input type="checkbox"/>	GRUPOS
<input type="checkbox"/>	MISSAO
<input type="checkbox"/>	PERIODO
<input type="checkbox"/>	ARCO(I,J)
<input type="checkbox"/>	DISPONIBILIDADE(T,P)
<input type="checkbox"/>	EQUIPE(GR)
<input type="checkbox"/>	DURACAO(GR)
<input type="checkbox"/>	DEMANDA(T,D,GR,K)
<input type="checkbox"/>	PessoaGrupo(P,GR)
<input type="checkbox"/>	PessGRupoK(P,GR,K)
<input type="checkbox"/>	OFERTA(T,O,P,GR)
<input type="checkbox"/>	CUSTO(I,J)
<input type="checkbox"/>	TEMPO(I,J)
<input type="checkbox"/>	ALOCADOGERAL(T,n,da,P)
<input type="checkbox"/>	ALOCADO(T,n,da,P,GR,K)
<input type="checkbox"/>	TEMPOUTILIZADO(T,P)
<input type="checkbox"/>	ATENDIDA(T,P,da,GR,K)
<input type="checkbox"/>	FUNCAOOBJETIVO
<input type="checkbox"/>	FO
<input type="checkbox"/>	R1(T,oa,da,P,GR)

Fonte: Elaborado pela autora (2020).

FIGURA 20 – Usabilidade da interface AIMMS.

O manual de usuário foi um suporte excelente na etapa de modelagem, grande parte dos problemas que surgiram durante o processo foi solucionada a partir da consulta no manual e da comunidade de suporte no próprio site do software onde os usuários podem interagir e obter ajuda especializada do AIMMS. A interface da comunidade consta na Figura 21.



Fonte: AIMMS (2020)

FIGURA 21 – Comunidade de suporte do AIMMS.

Ademais, no AIMMS existe a possibilidade de colocar as informações em tabelas, gráficos de barra e Diagrama de Gantt. A Figura 22 ilustra os resultados das variáveis “Alocado” e “Tempo Utilizado” em tabelas. Esses recursos podem ser valiosas ferramentas na visualização do problema e dos resultados.

Declaration									
<input checked="" type="checkbox"/>	NOS								
<input checked="" type="checkbox"/>	ORIGEM								
<input checked="" type="checkbox"/>	DESTINO								
<input checked="" type="checkbox"/>	NosAtivos								
<input checked="" type="checkbox"/>	OrigemAtiva								
<input checked="" type="checkbox"/>	DestinoAtivo								
<input checked="" type="checkbox"/>	GRUPOS								
<input checked="" type="checkbox"/>	PESSOAS								
<input checked="" type="checkbox"/>	MISSAO								
<input checked="" type="checkbox"/>	PERIODO								
<input checked="" type="checkbox"/>	ARCO(I,J)								
<input checked="" type="checkbox"/>	DISPONIBILIDADE(T,P)								
<input checked="" type="checkbox"/>	EQUIPE(GR)								
<input checked="" type="checkbox"/>	DURACAO(GR)								
<input checked="" type="checkbox"/>	DEMANDA(T,D,GR,K)								
<input checked="" type="checkbox"/>	PessoaGrupo(P,GR,K)								
<input checked="" type="checkbox"/>	PessGRupoK(P,GR,K)								
<input checked="" type="checkbox"/>	OFERTA(T,O,P,GR)								
<input checked="" type="checkbox"/>	CUSTO(I,J)								
<input checked="" type="checkbox"/>	TEMPO(I,J)								
<input checked="" type="checkbox"/>	ALOCADOGERAL(T,n,da,P)								
<input checked="" type="checkbox"/>	ALOCADO(T,n,da,P,GR,K)								

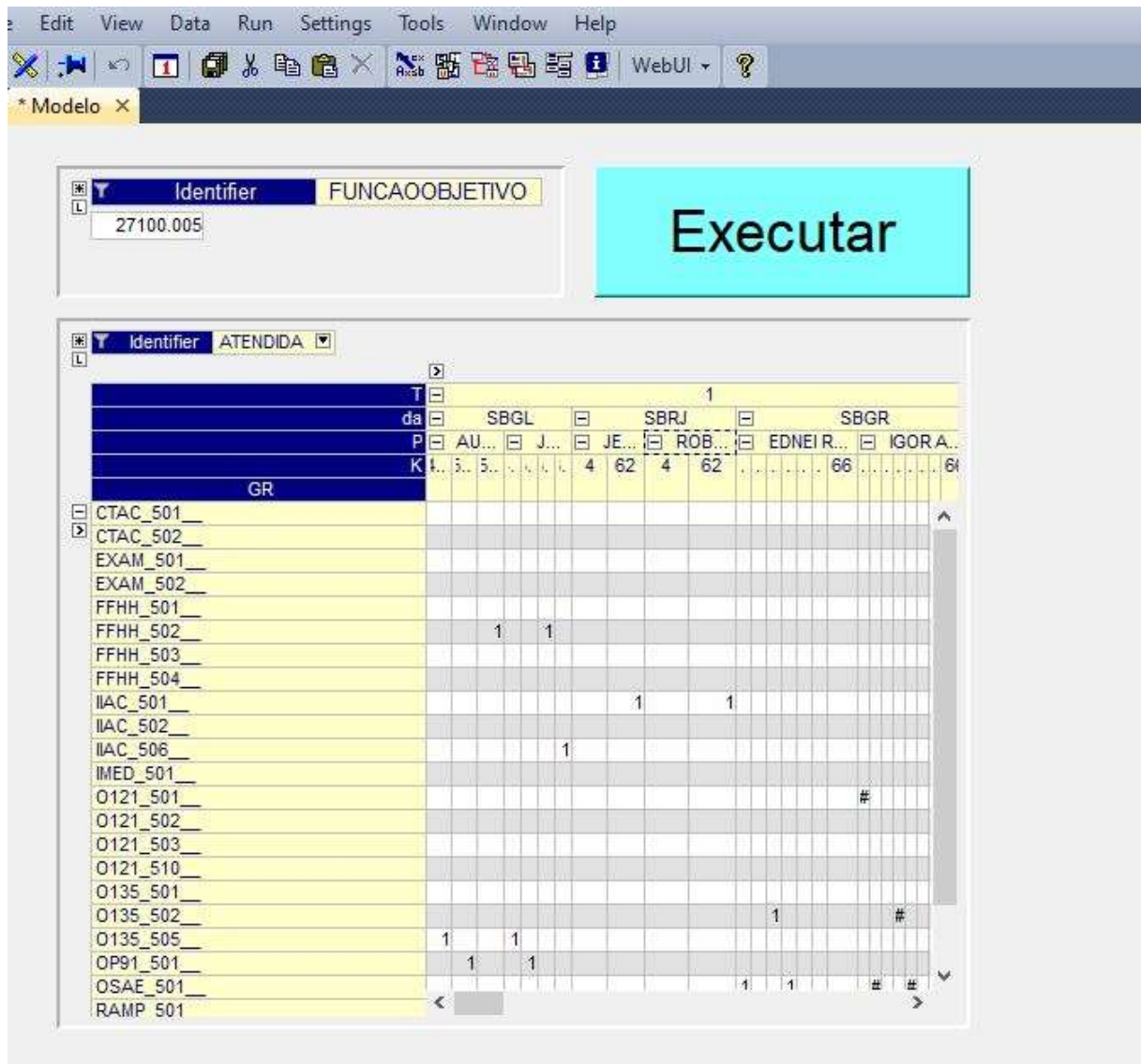
T	n	da	P	GR	K	ALOCADO	T	P	TEMPOUTILIZADO
1	SP	SBGR	SER	EXAM_502__	57	1.00	1	ADF	7.5
1	SP	SBGR	SER	0121_501__	17	1.00	1	ADF	18.0
1	SP	SBCT	SER	FFHH_501__	33	1.00	1	ALC	3.0
1	SP	SBCT	SER	IMED_501__	40	1.00	1	ALE	20.0
1	SP	SBCT	SER	0135_501__	72	1.00	1	ALE	7.5
1	SP	SBCT	SER	0135_502__	73	1.00	1	ALII	1.5
1	SP	SBCT	SER	0135_505__	74	1.00	1	ANA	3.0
1	DF	SBBR	ROC	EXAM_501__	31	1.00	1	ANC	19.5
1	DF	SBBR	ROC	FFHH_503__	85	1.00	1	ANC	7.5
1	DF	SBBR	ROC	FFHH_504__	86	1.00	1	ANT	8.5
1	DF	SBBR	ROC	IAC_501__	87	1.00	1	AUC	7.0
1	DF	SBBR	ROC	IAC_502__	88	1.00	1	CAF	4.5
1	DF	SBBR	ROC	0121_501__	92	1.00	1	DAM	8.5
1	DF	SBBR	ROC	0121_502__	18	1.00	1	EDM	7.0
1	DF	SBBR	ROC	RAMP_501__	10	1.00	1	EDN	14.0
1	SP	SBSP	ROI	EXAM_502__	32	1.00	1	EDU	16.0
1	SP	SBSP	ROI	IMED_501__	65	1.00	1	EDU	11.5
1	SP	SBSP	ROI	0135_505__	24	1.00	1	EVE	7.5
1	SP	SBCT	ROI	FFHH_501__	33	1.00	1	FRA	19.0
1	SP	SBCT	ROI	IMED_501__	40	1.00	1	FRA	6.5
1	SP	SBCT	ROI	0135_501__	72	1.00	1	HEN	6.0
1	SP	SBCT	ROI	0135_502__	73	1.00	1	IGD	14.0

Fonte: Elaborado pela autora (2020).

FIGURA 22 – Recursos AIMMS.

Assim, considerando a possibilidade de compilar os dados diretamente no software, não houve necessidade de fazer a exportação dos resultados para o Excel.

Outra função interessante no AIMMS é a possibilidade de programar uma interface para o usuário final. Para fins demonstrativos, foi criada uma interface básica do problema, conforme mostra a figura 23.



Fonte: Elaborado pela autora (2020).

FIGURA 23 – Interface para usuário final.

Ao clicar no botão “executar” o modelo é executado, assim que estiver pronto, os resultados da função objetivo e da variável “atendida” aparecem para o usuário. A ideia é deixar a interface simplificada e intuitiva. Assim, o programador pode criar a interface da forma que desejar, deixando visível apenas os resultados relevantes para o usuário final.



## 4.6.2 LINGO

Observou-se com a utilização do LINGO que além do conhecimento em programação matemática é necessário que o usuário tenha afinidade com alguns conceitos de programação computacional, uma vez que o software dispõe de uma interface limpa onde cabe ao usuário descrever todo o modelo, como mostra a Figura 12.

O LINGO é um software mais simples no tocante às funcionalidades, sua interface é enxuta, porém mostrou-se eficiente na execução do problema analisado neste estudo.

O output do modelo no LINGO foi exportado para um arquivo no bloco de notas visto que transferir os dados para um arquivo de texto foi mais simples que exportá-los para o Excel.

## 4.7 Compilação dos resultados

De acordo com as peculiaridades de cada software e a experiência obtida na modelagem, o estudo chegou às seguintes conclusões acerca das características dos softwares:

<b>Características gerais</b>	<b>AIMMS</b>	<b>LINGO</b>
Escolha do solver	Permite	Não permite
Interface	Adequada	Adequada
Usabilidade	Ótimo	Regular
Funções extras	Muitas opções	Poucas opções
Conhecimento em PO	Intermediário	Intermediário
Conhecimento em programação	Básico	Intermediário
Linguagem de modelagem	Intermediária	Complexa
Suporte/manual	Ótimo	Regular

Fonte: Elaborado pela autora (2020).

### QUADRO 8 – Características gerais

Os resultados dos testes foram compilados no Quadro 9.

<b>Testes</b>	<b>AIMMS</b>	<b>LINGO</b>
Disponibilidade de 20 dias (Modelo completo)	Resultado: 27100	Resultado: 27100
	Tempo: 82.64 segundos	Tempo: 69 segundos
	18069 iterações	125142 iterações
	<i>Gap: 0%</i>	<i>Gap: 0%</i>
Disponibilidade de 15 dias	Resultado: 27128	Resultado: 27128
	Tempo: 79.59 segundos	Tempo: 164 segundos
	17915 iterações	478040 iterações
	<i>Gap: 0%</i>	<i>Gap: 0,1%</i>
Disponibilidade de 10 dias com <i>gap = 5</i>	Resultado: 28663	Resultado: 28987
	Tempo: 127.84 segundos	Tempo: 156 segundos
	198988 iterações	296721 iterações
	<i>Gap: 3.01%</i>	<i>Gap: 5%</i>
Disponibilidade de 10 dias com <i>gap &lt; 5</i>	Resultado: 28504.00	-
	Tempo: 583.20 segundos	-
	1846424 iterações	-
	<i>Gap: 2%</i>	-

Fonte: Elaborado pela autora (2020).

### QUADRO 9 – Comparação dos resultados

## 5 CONCLUSÕES E RECOMENDAÇÕES

Diante da escassa produção científica em relação aos softwares de programação matemática, comprovada pela revisão de literatura, observou-se a necessidade de estudos que atestem a viabilidade dos softwares na execução de modelos diversos. Análises como estas podem ser relevantes na escolha do software mais adequado de acordo com as necessidades das organizações, além de contribuir com a literatura científica de Pesquisa Operacional ajudando a difundir o conhecimento.

No referencial teórico foram explanados demais fatores que motivaram a realização deste trabalho, como o avanço da utilização da Pesquisa Operacional nas organizações, a relevância da PO como forma de subsidiar a tomada de decisão, a programação matemática inteira como ferramenta eficiente para a otimização e a importância do apoio dos softwares nesse processo.

Dessa forma, o presente trabalho propôs cinco objetivos principais (listados na seção 1.3) que foram atingidos no decorrer do processo de pesquisa. O problema de programação inteira de grande porte utilizado nos testes foi oriundo do projeto *Safety Oversight*, parceria entre a ANAC e a UnB. Assim, o problema, que estava modelado no LINGO, foi modelado na linguagem AIMMS, esse processo contou com várias etapas de validação e ajustes. Depois de validado, iniciou-se a fase de testes, na qual diversas alternativas e configurações foram avaliadas, chegando aos resultados finais dispostos no Quadro 9, do capítulo 4.

A possibilidade de escolher o solver não representou muito impacto no desempenho do AIMMS, dado que os solvers obtiveram performances muito parecidas para o problema analisado. No entanto, acredita-se que em outros tipos de problemas de PO essa possibilidade possa ser um diferencial.

Em relação ao tempo computacional, foi constatado que este aumentava consideravelmente quando a complexidade do modelo foi elevada ao reduzir a disponibilidade dos servidores (fator que pode acontecer ocasionalmente). Assim, quando aumenta a complexidade do problema, os softwares podem se tornar inviáveis devido ao tempo despendido. A solução encontrada foi “relaxar” o problema aumentando o *gap* entre a solução ótima encontrada e a possível solução

ótima global, gerando um valor próximo ao ideal, porém, em um tempo computacional menor.

De acordo com os resultados, constatou-se que o AIMMS e o LINGO são opções viáveis para a execução de modelos de grande porte de Programação Inteira, já que ambos conseguiram alocar os servidores da melhor maneira possível, minimizando o custo de passagens aéreas, em um tempo aceitável de resolução.

Observou-se também que o usuário precisa ter experiência em modelagem matemática e na linguagem dos softwares. Dependendo da complexidade do modelo, a implantação da programação matemática pode ser um processo demorado e complicado, uma vez que o problema precisa ser modelado, testado e validado.

A programação matemática é capaz de reduzir significativamente os custos e/ou aumentar o lucro nas organizações, todavia, é importante lembrar que os softwares são produtos comercializados, sendo necessário avaliar também a relação custo-benefício para a organização que deseja adotar a Pesquisa Operacional e adquirir um software para otimizar seus processos de tomada de decisão.

## **5.1 Limitações e dificuldades da pesquisa**

A principal limitação da pesquisa foi a análise de somente um modelo de grande porte, dessa forma, não se pode tirar conclusões genéricas sobre os softwares.

Algumas dificuldades foram encontradas na realização deste estudo, grande parte delas relacionadas à falta de experiência na utilização dos softwares. Afinal, modelar um problema complexo com um elevado volume de dados exige um nível de conhecimento relevante tanto em modelagem matemática quanto na linguagem dos softwares.

Com o auxílio da comunidade de dúvidas do AIMMS e todo suporte fornecido pela plataforma, foi possível solucionar os problemas que apareceram no decorrer da modelagem. Em contrapartida, entender a programação do problema no LINGO foi um desafio maior devido a sua linguagem mais complexa e a falta de suporte

pelo *help* do software, tendo em vista que as informações disponíveis no suporte são mais genéricas, com poucos exemplos. A despeito das adversidades, os objetivos da pesquisa foram atingidos com êxito.

## 5.2 Sugestões para pesquisas futuras

Durante a elaboração deste trabalho, foram observadas lacunas na literatura de Pesquisa Operacional sobre os softwares. Grande parte da produção científica encontrada apenas utilizavam o AIMMS e o LINGO para resolver determinado problema de pesquisa, ou seja, não analisavam especificamente o desempenho dos softwares.

Tendo em vista os diversos âmbitos da Pesquisa Operacional (linear, não linear, estocástico, quadrático), existem oportunidades de pesquisas a serem exploradas. Sugere-se, portanto, que sejam realizados testes com diferentes tipos de problemas de PO, capazes de avaliar a viabilidade dos softwares em novos níveis, são vastas as possibilidades de incremento e remodelamento desses programas.

A realização de comparações entre outros softwares existentes no mercado também seria relevante a fim de verificar o desempenho de cada um de acordo com o tipo de problema proposto.

É importante considerar que a *performance* dos programas também está relacionada à capacidade computacional. Nesse sentido, podem ser feitas pesquisas com o objetivo de avaliar qual o impacto dos atributos de hardware no desempenho de cada software.

Espera-se com o passar dos anos, que o número de pesquisas semelhantes a esta tenha crescido e possa atestar diferentes aspectos e funcionalidades dos programas, difundindo o conhecimento sobre os softwares e mostrando-se úteis para o aprimoramento dos processos gerenciais.

## REFERÊNCIAS

- AIMMS. **Advanced Integrated Multidimensional Modeling Software**. Disponível em: <https://www.aimms.com/>. Acesso em: 30 abr. 2020.
- ANAC. **Agência Nacional de Aviação Civil**. Disponível em: <https://www.anac.gov.br/>. Acesso em: 30 abr. 2020.
- ARENALES, M. *et al.* **Pesquisa Operacional: para cursos de engenharia**. 2ed. Rio de Janeiro: Elsevier, 2015.
- BARON, C. Utilização de softwares na Pesquisa Operacional. **Revista Brasileira de Iniciação Científica**, Itapetininga, v. 4, n. 1, 2017.
- BAZIEWICZ, C. M. *et al.* Comparação de desempenho e usabilidade entre os softwares comerciais de otimização e o método dual para o problema clássico do transporte. **Simpósio de Pesquisa Operacional e Logística da Marinha**, 2013.
- BELFIORE, P.; FÁVERO, L.. **Pesquisa Operacional para cursos de engenharia**. Rio de Janeiro: Elsevier, 2013.
- CABRAL, E. L. D. S. **Aplicação do algoritmo DSM à determinação do posicionamento de bases descentralizadas do SAMU/Natal com uso de simulação**. Dissertação de Mestrado em Engenharia de Produção. Universidade Federal do Rio Grande do Norte, Natal, 2019.
- CARVALHO, Y. M.; SILVA JÚNIOR, O. S. D. Otimização da rede de transporte de suprimentos do Exército Brasileiro na região da Amazônia Oriental. **Simpósio de Pesquisa Operacional e Logística da Marinha**, Rio de Janeiro, 2019.
- COSTA, M. T. D. **Aplicação da Programação Inteira na otimização do roteamento de veículos de uma transportadora**. Dissertação (mestrado). Itajubá, Minas Gerais, 2019.
- CRONIN, P.; RYAN, F.; COUGHLAN, M. Undertaking a literature review: a step-by-step approach. **British Journal of Nursing**, 17(1), p.38-43, 2008.
- D'ANDRÉA, R. **Análise de um modelo matemático de localização na reorganização de uma rede logística estabelecida**. Dissertação (Mestrado). Universidade Federal do Espírito Santo, Vitória, 2011.
- DE-LA-TORRE-UGARTE-GUANILO, M. C.; TAKAHASHI, R. F.; BERTOLOZZI, M. R. Revisão sistemática: noções gerais. **Revista da Escola de Enfermagem da USP**. São Paulo, v. 45, n. 5, p. 1260-1266, out. 2011.
- GIL, A. C. **Métodos e técnicas de pesquisa social**. 6ª. ed. São Paulo: Atlas, 2008.
- GOLDBARG, M. C.; LUNA, H. P. L. **Programação linear e otimização combinatória: modelos e algoritmos**. 2.ed. – Rio de Janeiro: Elsevier, 2005

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à Pesquisa Operacional**. 9 ed. Porto Alegre: AMGH, 2013.

IGNÁCIO, A. A. V.; FERREIRA FILHO V. J. M.. **O uso de software de modelagem AIMMS na solução de problemas de programação matemática**. Pesquisa Operacional, vol.24, p. 197-210, 2004.

KOROISHI, G. O.. **Otimização de medidas de gerenciamento de fluxo de tráfego aéreo para múltiplos elementos regulados**. Dissertação (Mestrado). Universidade de São Paulo, São Paulo, 2019.

LACHTERMACHER, G.. **Pesquisa Operacional na Tomada de Decisões: modelagem em EXCEL**. 5ª edição. Rio de Janeiro: LTC, 2016.

LAKATOS, E. M; MARCONI, M. A. **Fundamentos de Metodologia Científica**. 5ª ed.. São Paulo: Atlas, 2003.

LINDO, Lindo Systems Inc. **LINGO: Optimization Modeling Software for Linear, Nonlinear, and Integer Programming**. Disponível em: <https://www.lindo.com/>. Acesso em: 16 set. 2020.

LONGO, H. J. **Técnicas para Programação Inteira e Aplicações em Problemas de Roteamento de Veículos**. Tese (Doutorado). Pontifícia Universidade Católica do Rio de Janeiro, 2004.

LOUREIRO, S. A. et. al. O uso do método de revisão sistemática da literatura na pesquisa em logística, transportes e cadeia de suprimentos. **Revista Transportes**. v. 24. n. 1. p. 95-10, 2016.

PIERROTTI, F. P. R.; SILVA JÚNIOR, O. S.. Rebalanceamento do Sistema de Bicicletas Compartilhadas: Experimento Realizado na Cidade do Rio de Janeiro. **Simpósio de Pesquisa Operacional e Logística da Marinha**, Rio de Janeiro, 2019.

REFELDT, M. J. H. **A aplicação de modelos matemáticos em situações-problemas empresariais, com uso do software LINDO**. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

SILVA JÚNIOR, A. L. D. *et al.* Planejamento de Lavra: um estudo de caso. **14º Simpósio Brasileiro de Automação Inteligente**. Ouro Preto, Minas Gerais, out./2019.

SILVA, E. L.; MENEZES, E. M.. **Metodologia da pesquisa e elaboração de dissertação**. 4 ed. Florianópolis: UFSC, 2005.

SOBRAPO. **Sociedade Brasileira de Pesquisa Operacional**. Disponível em: <https://www.sobrapo.org.br/>. Acesso em: 08 jun. 2020.

TAHA, H. A.; MARQUES, A. S.; SCARPEL, R. A.. **Pesquisa Operacional**. Pearson Education do Brasil, 2008.

WINSTON, W. L.; GOLDBERG, J. B. **Operations research: applications and algorithms**. 4ed. Thomson Brooks/Cole, 2004.

## APÊNDICES

### Apêndice A – Modelo de Designação AIMMS

```

Set NOS {Index: I, J;}

Set ORIGEM {SubsetOf: NOS; Index: O;}

Set DESTINO { SubsetOf: NOS; Index: D; }

Set NosAtivos { SubsetOf: NOS; Index: n, n1; }

Set OrigemAtiva { SubsetOf: NosAtivos; Index: oa; }

Set DestinoAtivo { SubsetOf: NosAtivos; Index: da; }

Set PESSOAS { Index: P; }

Set GRUPOS { Index: GR; }

Set MISSAO { Index: K; }

Set PERIODO { SubsetOf: Integers; Index: T; Definition: DATA{1}; }

Parameter ARCO { IndexDomain: (I,J); }

Parameter DISPONIBILIDADE { IndexDomain: (T,P); }

Parameter EQUIPE { IndexDomain: (GR); }

Parameter DURACAO { IndexDomain: (GR); }

Parameter DEMANDA { IndexDomain: (T,D,GR,K); }

Parameter PessoaGrupo { IndexDomain: (P,GR); }

Parameter PessGRupoK { IndexDomain: (P,GR,K); Definition:
sum((t,d),DEMANDA(T, D, GR, K))$PessoaGrupo(P, GR); }

Parameter OFERTA {IndexDomain: (T,O,P,GR); }

Parameter CUSTO {IndexDomain: (I,J); }

Parameter TEMPO {IndexDomain: (I,J); }

Variable ALOCADOGERAL
{IndexDomain:T,n,da,P}$ARCO(n,da)$sum((gr,k),PessGRupoK(P, GR, K));
Range: binary;}

Variable ALOCADO {IndexDomain: (T,n,da,P,GR,K)$ ARCO(n, da)$ PessGRupoK(P,
GR, K);
Range: nonnegative;}

Variable TEMPOUTILIZADO {IndexDomain: (T,P);
Range: nonnegative;}

```



```

Variable ATENDIDA {IndexDomain: (T,P,da,GR,K)$DEMANDA(T, da, GR,
K)$PessoaGrupo(P, GR);
Range: binary;}

Variable FUNCAOOBJETIVO {Range: free;
Definition: SUM((T,n,da,P),ALOCADOGERAL(T, n, da, P)*2*CUSTO(n, da)); }

MathematicalProgram FO {Objective: FUNCAOOBJETIVO;Direction:
minimize;Constraints: AllConstraints;Variables:
AllVariables;Type:Automatic; }

Constraint R1 {IndexDomain: (T,OA,DA,P,GR)$sum((k),PessGRupoK(P, GR,K));
Definition: SUM((K),ALOCADO(T, OA, DA, P, GR, K))<=OFERTA(T, OA, P, GR); }

Constraint R2 {IndexDomain: (T,P,Da,GR,K)$PessGRupoK(P, GR, K);Definition:
SUM((n),ALOCADO(T, n, Da, P, GR, K))=(ATENDIDA(T, P, Da, GR,
K)+SUM((n1),ALOCADO(T, Da, n1, P, GR, K))); }

Constraint R3 {IndexDomain: (T,Da,P,GR,K)$PessGRupoK(P, GR, K);Definition:
SUM((n1),ALOCADO(T, Da, n1, P, GR, K))<=SUM((n),ALOCADO(T, n, Da, P, GR,
K)); }

Constraint R4 {IndexDomain: (T,Da,GR,K)$DEMANDA(T, Da, GR, K);Definition:
SUM((P),ATENDIDA(T, P, Da, GR, K))=(EQUIPE(GR)*DEMANDA(T, Da, GR, K)); }

Constraint R5 {IndexDomain: (T,P)$sum((gr,k),PessGRupoK(P, GR,
K));Definition: SUM((Da,GR,K),ATENDIDA(T, P, Da, GR, K)*DURACAO(GR))+
SUM((N,DA), (ALOCADOGERAL(T,N,DA,P)*2*TEMPO(N,DA)))=TEMPOUTILIZADO(T, P); }

Constraint R6 {IndexDomain: (T,P)$sum((gr,k),PessGRupoK(P, GR,
K));Definition: TEMPOUTILIZADO(T, P)<=DISPONIBILIDADE(T, P); }

Constraint R7 { IndexDomain: (T,n,Da,P,GR,K)$ ARCO(n, Da)$PessGRupoK(P, GR,
K); Definition: ALOCADOGERAL(T, n, Da, P)>=ALOCADO(T, n, Da, P, GR, K); }

```

## Apêndice B – Modelo de Designação LINGO

!Modelo de Designação LINGO;

Sets:

nos/@ole()//;

origem(nos)/@ole()/:OrigemAtiva;

destino(nos)/@ole()/:DestinoAtivo;

Arco(nos,nos)/@ole()/:Custo,tempo;

Periodo/1/;

missao/@ole()//;

Pessoas/@ole()//;

grupo/@ole()/:duracao,equipe,GrupoAtivo;

PessoaAtiva(Pessoas);

DATIVO(destino)|DestinoAtivo(&1)#GE#1;

OATIVA(origem)|OrigemAtiva(&1)#GE#1;

NOSATIVOS(nos)| @in(OATIVA, nos(&1)) #OR# @in(DATIVO, nos(&1))::

ARCOA(nosativos,nosativos);

ARCOATIVO(nosativos,nosativos);

TPe(pessoas,periodo):tempoutilizado,disponibilidade;

```

pessoaquebrada (pessoas, grupo) /@ole() /;
PO (pessoas, origem) /@ole() /;
NEPO (pessoas, origem) | #not# @in (PO, &1, &2);
PGM (pessoas, grupo, missao);

```

```

ExisteMissao (missao, grupo) /@ole() /;
GRATIVO (grupo) | GrupoAtivo (&1) #GE#1;
PGATIVO (pessoas, grupo);

```

```

moradia (pgativo, oativa) : oferta;
dem (periodo, missao, grupo, destino) /@ole() / : demanda;

```

```

ArcoMi (periodo, arcoativo, PGM) : Alocado;
NoPeGr (periodo, missao, grupo, destino, pessoas) : Atendida;
VarInt (periodo, arcoativo, pessoas) : AlocadoGeral;

```

Endsets

Data:

```

disponibilidade, duracao, equipe, custo, tempo, grupoativo, destinoativo, origemativa = @ole();

```

Enddata

```

Submodel Alocacao:
[FO]

```

```

MIN = @SUM (VarInt (t, i, j, p) : AlocadoGeral (t, i, j, p) * 2 * Custo (i, j));

```

```

@for (dem (t, k, gr, d) : [demmissao] demanda (t, k, gr, d) = 1);

```

```

@for (PO (p, o) : @for (moradia (p, gr, o) | p #LE# @index (pessoas, Dummy3) :
[ofertaexist] oferta (p, gr, o) = 100));

```

```

@for (PO (p, o) : @for (moradia (p, gr, o) | p #GT# @index (pessoas, Dummy3) :
[ofertaexist1] oferta (p, gr, o) = 10));

```

```

@for (NEPO (p, o) : @for (moradia (p, gr, o) : [ofertanaoexist] oferta (p, gr, o) = 0));

```

```

@for (PGATIVO (p, gr) : @for (oativa (oa) : @for (dativo (e) :
@for (periodo (t) : [oft] @sum (missao (k) | @in (ArcoMi, t, oa, e, p, gr, k) : alocado (t, oa,
e, p, gr, k)) <= oferta (p, gr, oa))));

```

```

@for (nopegr (t, k, gr, d, p) : [balanco]
@sum (nosativos (h) | @in (ArcoMi, t, h, d, p, gr, k) : alocado (t, h, d, p, gr, k)) =
atendida (t, k, gr, d, p) +
@sum (nosativos (h) | @in (ArcoMi, t, d, h, p, gr, k) : Alocado (t, d, h, p, gr, k)));

```

```

@for (PGM (p, gr, k) : @for (periodo (t) : @for (dativo (e) : [fixa] @sum (nosativos (h) | @in
(ArcoMi, t, h, e, p, gr, k) : alocado (t, h, e, p, gr, k)) >=
@sum (nosativos (h) | @in (ArcoMi, t, e, h, p, gr, k) : Alocado (t, e, h, p, gr, k))));

```

```

@for (dem (t, k, gr, d) : [equi] @sum (nopegr (t, k, gr, d, p) : atendida (t, k, gr, d, p)) =
Equipe (gr) * demanda (t, k, gr, d));

```

```

@for (periodo (t) : @for (pessoas (p) : [temp] @sum (nopegr (t, k, gr, d, p) : atendida (t, k,
gr, d, p) * duracao (gr)) + @sum (VarInt (t, i, j, p) :
Alocadogeral (t, i, j, p) * 2 * tempo (i, j)) = TempoUtilizado (p, t));

```

```

@for (TPE (p, t) : [dispontempo] TempoUtilizado (p, t) <= disponibilidade (p, t));

```

```

@for(ArcoMi(t,i,j,p,gr,k):[aloc] alocadogeral(t,i,j,p) >=
alocado(t,i,j,p,gr,k));

@for(NoPeGr(t,k,gr,d,p):@bin(atendida(t,k,gr,d,p)));!Restrição de binario;

@for(VarInt(t,i,j,p):@bin(alocadogeral(t,i,j,p)));!Restrição de binario;

Endsubmodel

calc:

@set('terseo', 1); !Reduce output level;
@set('loopop', 1); !Loop optimization;

@for(pessoaquebrada(p,gr): @for(existemissao(k,gr): @insert(PGM, p, gr,
k)));

@for(pessoaquebrada(p,gr): @for(dem(t,k,gr,d):@insert(nopegr,t,k,gr,d,p)));

@for(pessoaquebrada(p,gr): @for(grativo(gr): @insert(pgativo,p,gr)));

@for(arco(i,j): @for(arcoa(i,j): @insert(arcoativo,i,j)));

@DIVERT('SPO_Solucao.txt');
@solve(Alocacao);
@write('Periodo origem destino pessoa alocadogeral',@NEWLINE(1));
@writefor(VarInt(t,i,j,p) | alocadogeral(t,i,j,p) #GT# 0: Varint(t,i,j,p),
' ', Alocadogeral(t,i,j,p), @NEWLINE(1)) ;
@write('Periodo missao atividade destino pessoas Atendida',@NEWLINE(1));
@writefor(NoPeGr(t,kat,gr,j,p) | atendida(t,k,gr,j,p) #GT# 0:
NoPeGr(t,k,gr,j,p), ' ', endida(t,k,gr,j,p), @NEWLINE(1)) ;
@write('Pessoa Periodo Tempo Utilizado',@NEWLINE(1));
@writefor(TPe(p,t) | TempoUtilizado(p,t) #GT# 0: TPe(p,t), ' ',
TempoUtilizado(p,t), @NEWLINE(1)) ;
@DIVERT();

endcalc

```

## Apêndice C – Modelagem Completa AIMMS

The image displays two screenshots of the AIMMS IDE, showing the Model Explorer and the Procedure Editor for two different procedures.

**Top Screenshot: LEITURA\_ENTRADAS**

**Model Explorer:** Shows the project structure for ANACREDUZZ2. The 'LERENTRADAS' folder is expanded, showing sub-folders 'LEITURA\_ENTRADAS' and 'RADOMIZE', and a file 'ARQUIVOENTRADA'. Under 'Declaration', there are several procedure icons: 'MainInitialization', 'PostMainInitialization', 'MainExecution', 'CheckExcelError', 'PreMainTermination', and 'MainTermination'. A 'Predeclared Identifiers [read-only]' folder is also visible.

**Procedure Editor:** The 'Procedure' is 'LEITURA\_ENTRADAS'. The 'Arguments' and 'Property' sections are empty. The 'Body' contains the following code:

```

EXCELSETACTIVESHEET(SPO,"ARCOS");
CheckExcelError;
EXCELRETRIEVESET(SPO,NOS,"NOS");
CheckExcelError;
EXCELRETRIEVESET(SPO,ORIGEM,"ORIGEM");
CheckExcelError;
EXCELRETRIEVESET(SPO,DESTINO,"DESTINO");
CheckExcelError;
EXCELRETRIEVESET(SPO,NosAtivos,"nosativos");
CheckExcelError;
EXCELRETRIEVESET(SPO,OrigemAtiva,"origemativa");
CheckExcelError;
EXCELRETRIEVESET(SPO,DestinoAtivo,"destinoativo");
CheckExcelError;
EXCELRETRIEVETABLE(SPO,ARCO,"ARCO","arc_ARCO","");
CheckExcelError;
EXCELRETRIEVETABLE(SPO,CUSTO,"CUSTO","arc_Arco","");
CheckExcelError;
EXCELRETRIEVETABLE(SPO,TEMPO,"TEMPO","arc_Arco","");
CheckExcelError;
EXCELSETACTIVESHEET(SPO,"DISPONIBILIDADE");
CheckExcelError;
EXCELRETRIEVESET(SPO,PESSOAS,"PESSOAS");
CheckExcelError;

```

**Bottom Screenshot: RADOMIZE**

**Model Explorer:** The same project structure is shown, but the 'RADOMIZE' folder is now selected in the Model Explorer.

**Procedure Editor:** The 'Procedure' is 'RADOMIZE'. The 'Arguments' and 'Property' sections are empty. The 'Body' contains the following code:

```

ExcelSetActiveSheet(SPO,"PLAN1");
Leitura_Entradas;

```

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - MainInitialization
  - PostMainInitialization
  - MainExecution
  - CheckExcelError
  - PreMainTermination
  - MainTermination
- Predeclared Identifiers [read-only]

Type: String Parameter

Identifier: SPO

Index domain: [edit]

Text: [edit]

Default: [edit]

Property: [edit]

Definition: "SPO.XLSX"

Initial data

---

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - NOS
  - ORIGEM
  - DESTINO
  - NosAtivos
  - OrigemAtiva
  - DestinoAtivo
  - RedeAtivaTirar
  - PESSOAS

Type: Set

Identifier: NOS

Index domain: [edit]

Subset of: [edit]

Text: [edit]

Index: I, J

Parameter: [edit]

Property: [edit]

Order by: [edit]

Definition: [edit]

Initial data

---

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - NOS
  - ORIGEM
  - DESTINO
  - NosAtivos
  - OrigemAtiva
  - DestinoAtivo
  - RedeAtivaTirar

Type: Set

Identifier: ORIGEM

Index domain: [edit]

Subset of: NOS

Text: [edit]

Index: O

Parameter: [edit]

Property: [edit]

Order by: [edit]

Definition: [edit]

Initial data

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - NOS
  - ORIGEM
  - DESTINO
  - NosAtivos
  - OrigemAtiva
  - DestinoAtivo
  - RedeAtivaTirar

Type: Set

Identifier: DESTINO

Index domain: [edit]

Subset of: NOS

Text: [edit]

Index: D

Parameter: [edit]

Property: [edit]

Order by: [edit]

Definition [edit]

Initial data

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - NOS
  - ORIGEM
  - DESTINO
  - NosAtivos
  - OrigemAtiva
  - DestinoAtivo
  - RedeAtivaTirar

Type: Set

Identifier: NosAtivos

Index domain: [edit]

Subset of: NOS

Text: [edit]

Index: n, n1

Parameter: [edit]

Property: [edit]

Order by: [edit]

Definition [edit]

Initial data

Model Explorer: ANACREDUZ2.ams

LEITURA\_ENTRADAS RADOMIZE ARQUIVOENTRADA

\* Main ANACREDUZ2

- LERENTRADAS
  - LEITURA\_ENTRADAS
  - RADOMIZE
  - ARQUIVOENTRADA
    - SPO
- Declaration
  - NOS
  - ORIGEM
  - DESTINO
  - NosAtivos
  - OrigemAtiva
  - DestinoAtivo
  - RedeAtivaTirar

Type: Set

Identifier: OrigemAtiva

Index domain: [edit]

Subset of: NosAtivos

Text: [edit]

Index: oa

Parameter: [edit]

Property: [edit]

Order by: [edit]

Definition [edit]

Initial data

The image displays three sequential screenshots of the Model Explorer interface for a project named ANACREDUZ2.ams. Each screenshot shows a tree view on the left and a configuration panel on the right.

**Top Screenshot:** The tree view shows the hierarchy: Main ANACREDUZ2 > LERENTRADAS > LEITURA\_ENTRADAS. The configuration panel for LEITURA\_ENTRADAS shows:
 

- Type: Set
- Identifier: DestinoAtivo
- Index domain: [edit]
- Subset of: NosAtivos
- Text: [edit]
- Index: da
- Parameter: [edit]
- Property: [edit]
- Order by: [edit]
- Definition:  (selected)
- Initial data:

**Middle Screenshot:** The tree view shows: Main ANACREDUZ2 > Declaration > PESSOAS. The configuration panel for PESSOAS shows:
 

- Type: Set
- Identifier: PESSOAS
- Index domain: [edit]
- Subset of: [edit]
- Text: [edit]
- Index: P
- Parameter: [edit]
- Property: [edit]
- Order by: [edit]
- Definition:  (selected)
- Initial data:

**Bottom Screenshot:** The tree view shows: Main ANACREDUZ2 > Declaration > GRUPOS. The configuration panel for GRUPOS shows:
 

- Type: Set
- Identifier: GRUPOS
- Index domain: [edit]
- Subset of: [edit]
- Text: [edit]
- Index: GR
- Parameter: [edit]
- Property: [edit]
- Order by: [edit]
- Definition:  (selected)
- Initial data:

The image displays three sequential screenshots of the Model Explorer interface for the project ANACREDUZ2.ams, showing the configuration of different entities.

**Top Screenshot: Entity MISSAO**

- Type:** Set
- Identifier:** MISSAO
- Index domain:** (empty)
- Subset of:** (empty)
- Text:** (empty)
- Index:** K
- Parameter:** (empty)
- Property:** (empty)
- Order by:** (empty)
- Definition:**  (Selected)
- Initial data:**  (Not selected)

**Middle Screenshot: Entity PERIODO**

- Type:** Set
- Identifier:** PERIODO
- Index domain:** (empty)
- Subset of:** Integers
- Text:** (empty)
- Index:** T
- Parameter:** (empty)
- Property:** (empty)
- Order by:** (empty)
- Definition:**  DATA{1}
- Initial data:**  (Not selected)

**Bottom Screenshot: Entity ARCO(I,J)**

- Type:** Parameter
- Identifier:** ARCO
- Index domain:** (I,J)
- Text:** (empty)
- Range:** (empty)
- Unit:** (empty)
- Default:** (empty)
- Property:** (empty)
- Definition:**  (Selected)
- Initial data:**  (Not selected)



File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

PO NOS ORIGEM DESTINO NosAtivos

DestinoAtivo  
 PESSOAS  
 GRUPOS  
 MISSAO  
 PERIODO  
 ARCO(I,J)  
 DISPONIBILIDADE(T,P)  
 EQUIPE(GR)  
 DURACAO(GR)  
 DEMANDA(T,D,GR,K)  
 PessoaGrupo(P,GR)  
 PessGRupoK(P,GR,K)  
 OFERTA(T,O,P,GR)

Type: Parameter

Identifier: DISPONIBILIDADE

Index domain: (T,P)

Text:

Range:

Unit:

Default:

Property:

Definition

Initial data

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

ORIGEM DESTINO NosAtivos OrigemAtiva

DestinoAtivo  
 PESSOAS  
 GRUPOS  
 MISSAO  
 PERIODO  
 ARCO(I,J)  
 DISPONIBILIDADE(T,P)  
 EQUIPE(GR)  
 DURACAO(GR)  
 DEMANDA(T,D,GR,K)  
 PessoaGrupo(P,GR)  
 PessGRupoK(P,GR,K)

Type: Parameter

Identifier: EQUIPE

Index domain: (GR)

Text:

Range:

Unit:

Default:

Property:

Definition

Initial data

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

DESTINO NosAtivos OrigemAtiva R5 R6

DestinoAtivo  
 PESSOAS  
 GRUPOS  
 MISSAO  
 PERIODO  
 ARCO(I,J)  
 DISPONIBILIDADE(T,P)  
 EQUIPE(GR)  
 DURACAO(GR)  
 DEMANDA(T,D,GR,K)  
 PessoaGrupo(P,GR)  
 PessGRupoK(P,GR,K)  
 OFERTA(T,O,P,GR)

Type: Parameter

Identifier: DURACAO

Index domain: (GR)

Text:

Range:

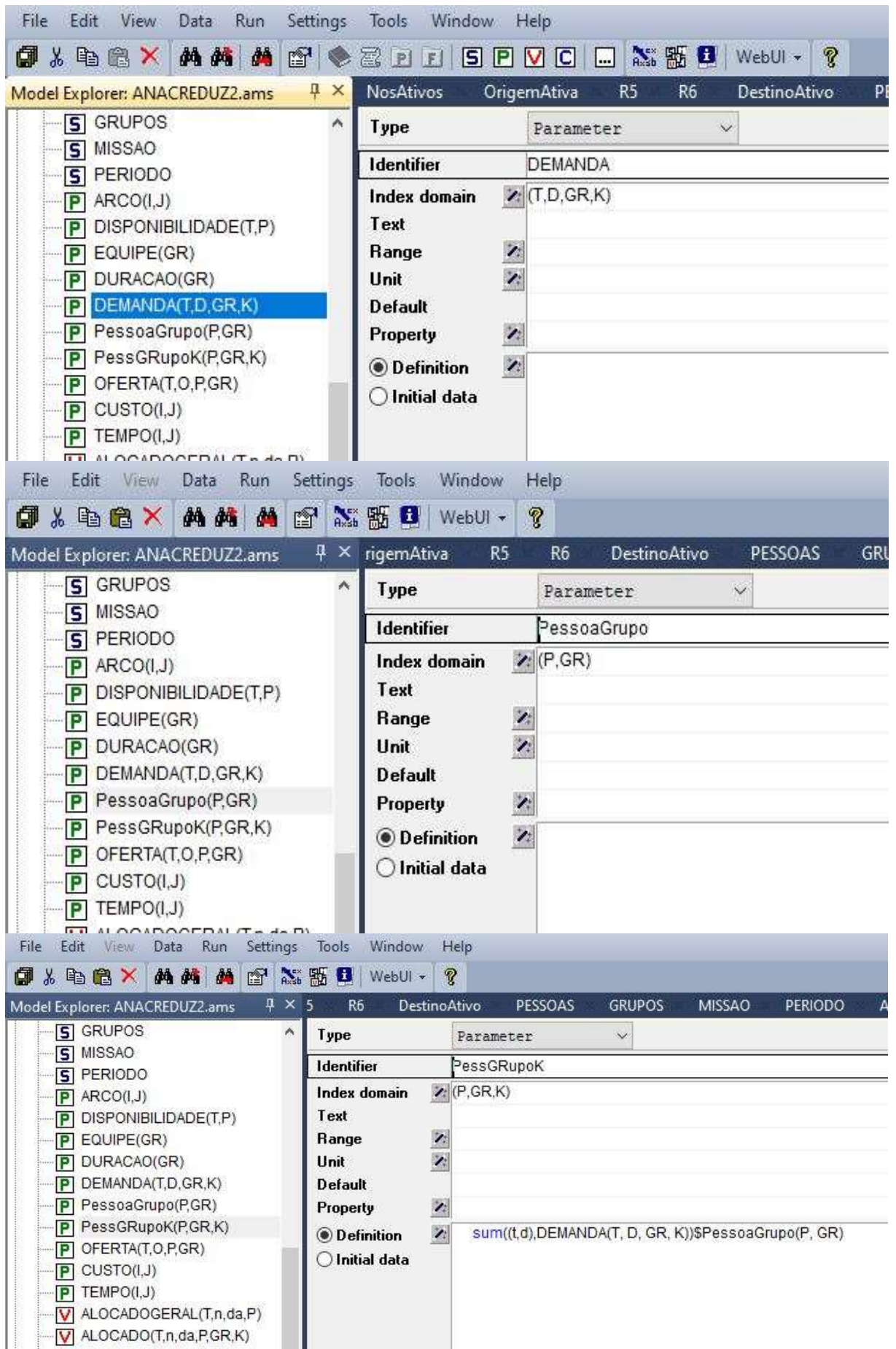
Unit:

Default:

Property:

Definition

Initial data



The image displays three sequential screenshots of the ANACREDUZ2.ams Model Explorer interface, showing the configuration of different parameters. Each screenshot includes a menu bar (File, Edit, View, Data, Run, Settings, Tools, Window, Help), a toolbar, and a list of model elements on the left. The right pane shows the detailed properties for the selected parameter.

**Screenshot 1: OFERTA**

- Type:** Parameter
- Identifier:** OFERTA
- Index domain:** (T,O,P,GR)
- Text:**
- Range:**
- Unit:**
- Default:**
- Property:**
- Definition:**  (Selected)
- Initial data:**

**Screenshot 2: CUSTO**

- Type:** Parameter
- Identifier:** CUSTO
- Index domain:** (I,J)
- Text:**
- Range:**
- Unit:**
- Default:**
- Property:**
- Definition:**  (Selected)
- Initial data:**

**Screenshot 3: TEMPO**

- Type:** Parameter
- Identifier:** TEMPO
- Index domain:** (I,J)
- Text:**
- Range:**
- Unit:**
- Default:**
- Property:**
- Definition:**  (Selected)
- Initial data:**

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

ARCO DISPONIBILIDADE EQUIPE DURACAO DEMANDA PessoaGrupo

Type	Variable
Identifier	ALOCADOGERAL
Index domain	$(T,n,da,P)\$ARCO(n, da)\$sum((gr,k),PessGRupoK(P, GR, K))$
Text	
Range	binary
Unit	
Default	
Property	
Priority	
Nonvar status	
Relax status	
Definition	

Model Explorer: ANACREDUZ2.ams

- ARCO(I,J)
- DISPONIBILIDADE(T,P)
- EQUIPE(GR)
- DURACAO(GR)
- DEMANDA(T,D,GR,K)
- PessoaGrupo(P,GR)
- PessGRupoK(P,GR,K)
- OFERTA(T,O,P,GR)
- CUSTO(I,J)
- TEMPO(I,J)
- ALOCADOGERAL(T,n,da,P)
- ALOCADO(T,n,da,P,GR,K)
- TEMPOUTILIZADO(T,P)
- ATENDIDA(T,P,da,GR,K)
- FUNCAOOBJETIVO

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

DISPONIBILIDADE EQUIPE DURACAO DEMANDA PessoaGrupo

Type	Variable
Identifier	ALOCADO
Index domain	$(T,n,da,P,GR,K)\$ARCO(n, da)\$PessGRupoK(P, GR, K)$
Text	
Range	nonnegative
Unit	
Default	
Property	
Priority	
Nonvar status	
Definition	

Model Explorer: ANACREDUZ2.ams

- ARCO(I,J)
- DISPONIBILIDADE(T,P)
- EQUIPE(GR)
- DURACAO(GR)
- DEMANDA(T,D,GR,K)
- PessoaGrupo(P,GR)
- PessGRupoK(P,GR,K)
- OFERTA(T,O,P,GR)
- CUSTO(I,J)
- TEMPO(I,J)
- ALOCADOGERAL(T,n,da,P)
- ALOCADO(T,n,da,P,GR,K)
- TEMPOUTILIZADO(T,P)
- ATENDIDA(T,P,da,GR,K)
- FUNCAOOBJETIVO

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

EQUIPE DURACAO DEMANDA PessoaGrupo PessGRupoK

Type	Variable
Identifier	TEMPOUTILIZADO
Index domain	(T,P)
Text	
Range	nonnegative
Unit	
Default	
Property	
Priority	
Nonvar status	
Definition	

Model Explorer: ANACREDUZ2.ams

- ARCO(I,J)
- DISPONIBILIDADE(T,P)
- EQUIPE(GR)
- DURACAO(GR)
- DEMANDA(T,D,GR,K)
- PessoaGrupo(P,GR)
- PessGRupoK(P,GR,K)
- OFERTA(T,O,P,GR)
- CUSTO(I,J)
- TEMPO(I,J)
- ALOCADOGERAL(T,n,da,P)
- ALOCADO(T,n,da,P,GR,K)
- TEMPOUTILIZADO(T,P)
- ATENDIDA(T,P,da,GR,K)
- FUNCAOOBJETIVO

The image displays three sequential screenshots of the ANACREDUZ2.ams model explorer interface, showing the configuration of different model components.

**Top Screenshot: Variable ATENDIDA**

- Type:** Variable
- Identifier:** ATENDIDA
- Index domain:** (T,P,da,GR,K)\$DEMANDA(T, da, GR, K)\$PessoaGrupo(P, GR)
- Range:** binary

**Middle Screenshot: Objective Function FUNCAOOBJETIVO**

- Type:** Variable
- Identifier:** FUNCAOOBJETIVO
- Range:** free
- Definition:**  $SUM((T,n,da,P),ALOCADOGERAL(T, n, da, P)*2*CUSTO(n, da))$

**Bottom Screenshot: Mathematical Program FO**

- Type:** Mathematical Progr
- Identifier:** FO
- Objective:** FUNCAOOBJETIVO
- Direction:** minimize
- Constraints:** AllConstraints
- Variables:** AllVariables
- Type:** Automatic

The image displays four sequential screenshots of the ANACREDUZ2.ams model editor, each showing a different constraint definition (R1, R2, R3, and R4) selected in the Model Explorer and its details in the Properties window.

**Constraint R1:**

- Type: Constraint
- Identifier: R1
- Index domain:  $(T, OA, DA, P, GR) \$sum((k), PessGRupoK(P, GR, K))$
- Text: (empty)
- Unit: (empty)
- Property: (empty)
- Definition:  $SUM((K), ALOCADO(T, OA, DA, P, GR, K)) \leq OFERTA(T, OA, P, GR)$

**Constraint R2:**

- Type: Constraint
- Identifier: R2
- Index domain:  $(T, P, Da, GR, K) \$PessGRupoK(P, GR, K)$
- Text: (empty)
- Unit: (empty)
- Property: (empty)
- Definition:  $SUM((n), ALOCADO(T, n, Da, P, GR, K)) = ATENDIDA(T, P, Da, GR, K) + SUM((n1), ALOCADO(T, Da, n1, P, GR, K))$

**Constraint R3:**

- Type: Constraint
- Identifier: R3
- Index domain:  $(T, Da, P, GR, K) \$PessGRupoK(P, GR, K)$
- Text: (empty)
- Unit: (empty)
- Property: (empty)
- Definition:  $SUM((n1), ALOCADO(T, Da, n1, P, GR, K)) \leq SUM((n), ALOCADO(T, n, Da, P, GR, K))$

**Constraint R4:**

- Type: Constraint
- Identifier: R4
- Index domain:  $(T, Da, GR, K) \$DEMANDA(T, Da, GR, K)$
- Text: (empty)
- Unit: (empty)
- Property: (empty)
- Definition:  $SUM((P), ATENDIDA(T, P, Da, GR, K)) = EQUIPE(GR) * DEMANDA(T, Da, GR, K)$

The image displays three sequential screenshots of the ANACREDUZ2.ams software interface, showing the configuration of constraints and the execution of a procedure.

**Top Screenshot: Constraint R5**

- Type:** Constraint
- Identifier:** R5
- Index domain:** (T,P)\$sum((gr,k),PessGRupoK(P, GR, K))
- Text:**
- Unit:**
- Property:**
- Definition:**  $SUM(Da,GR,K)ATENDIDA(T,P,Da,GR,K)DURACAO(GR) - SUM(N,DA)(ALOCADOGERAL(T,N,DA,P)^2TEMPOIN,DA) - TEMPOUTILIZADO(T,P)$

**Middle Screenshot: Constraint R6**

- Type:** Constraint
- Identifier:** R6
- Index domain:** (T,P)\$sum((gr,k),PessGRupoK(P, GR, K))
- Text:**
- Unit:**
- Property:**
- Definition:**  $TEMPOUTILIZADO(T,P) \leq DISPONIBILIDADE(T,P)$

**Bottom Screenshot: Constraint R7**

- Type:** Constraint
- Identifier:** R7
- Index domain:** (T,n,Da,P,GR,K)\$ ARCO(n, Da)\$PessGRupoK(P, GR, K)
- Text:**
- Unit:**
- Property:**
- Definition:**  $ALOCADOGERAL(T, n, Da, P) \geq ALOCADO(T, n, Da, P, GR, K)$

**Bottom Screenshot: Procedure Execution**

- Procedure:** MainExecution
- Body:**

```
SHOWPROGRESSWINDOW(1);
LEITURA_ENTRADAS;
SOLVE FO;
|
```

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

- R2(T,P,da,GR,K)
- R3(T,da,P,GR,K)
- R4(T,da,GR,K)
- R5(T,P)
- R6(T,P)
- R7(T,n,da,P,GR,K)
- MainInitialization
- PostMainInitialization
- MainExecution
- CheckExcelError
- PreMainTermination
- MainTermination

Procedure: CheckExcelError

Arguments:

Property:

```

Body
if CurrentErrorMessage then
  DialogMessage(CurrentErrorMessage);
  empty CurrentErrorMessage;
endif;

```

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

- R2(T,P,da,GR,K)
- R3(T,da,P,GR,K)
- R4(T,da,GR,K)
- R5(T,P)
- R6(T,P)
- R7(T,n,da,P,GR,K)
- MainInitialization
- PostMainInitialization
- MainExecution
- CheckExcelError
- PreMainTermination
- MainTermination

Procedure: PreMainTermination

```

Body
return DataManagementExit();

```

File Edit View Data Run Settings Tools Window Help

Model Explorer: ANACREDUZ2.ams

- R2(T,P,da,GR,K)
- R3(T,da,P,GR,K)
- R4(T,da,GR,K)
- R5(T,P)
- R6(T,P)
- R7(T,n,da,P,GR,K)
- MainInitialization
- PostMainInitialization
- MainExecution
- CheckExcelError
- PreMainTermination
- MainTermination

Procedure: MainTermination

```

Body
return 1;

```



## Apêndice D – Modelagem Completa LINGO



```

Sets:
    nos/@ole()/;
    origem(nos)/@ole()/:OrigemAtiva;
    destino(nos)/@ole()/:DestinoAtivo;
    Arco(nos,nos)/@ole()/:Custo,tempo;

    Período/1/;
    missao/@ole()/;
    Pessoas/@ole()/;
    grupo/@ole()/:duracao,equipe,GrupoAtivo;
    PessoaAtiva(Pessoas);

    DATIVO(destino)|DestinoAtivo(&1)#GE#1;
    OATIVA(origem)|OrigemAtiva(&1)#GE#1;
    NOSATIVOS(nos)|@in(OATIVA,nos(&1))#OR#@in(DATIVO,nos(&1));;
    ARCOA(nosativos,nosativos);
    ARCOATIVO(nosativos,nosativos);

    TPe(pessoas,período):tempoutilizado,disponibilidade;
    pessoaquebrada(pessoas,grupo)/@ole()/;
    PO(pessoas,origem)/@ole()/;
    NEPO(pessoas,origem)|#not#@in(PO,&1,&2);
    PGM(pessoas,grupo,missao);

    ExisteMissao(missao,grupo)/@ole()/;
    GRATIVO(grupo)|GrupoAtivo(&1)#GE#1;
    PGATIVO(pessoas,grupo);

    moradia(pgativo,oativa):oferta;
    dem(período,missao,grupo,destino)/@ole()/:demanda;

    ArcoMi(período,arcoativo,PGM):Alocado;
    NoPeGr(período,missao,grupo,destino,pessoas):Atendida;
    VarInt(período,arcoativo,pessoas):AlocadoGeral;

Endsets
  
```

```

File Edit Solver Window Help
[Icons]

Endsets
Data:
    disponibilidade,duracao,equipe,custo,tempo,grupoativo,destinoativo,origemativa = @ole();
Enddata
Submodel Alocacao:
[FO]
    MIN = @SUM(VarInt(t,i,j,p): AlocadoGeral(t,i,j,p)*2*Custo(i,j));
    @for(dem(t,k,gr,d): [demissao] demanda(t,k,gr,d) = 1);
    @for(PO(p,o):@for(moradia(p,gr,o)|p #LE# @index(pessoas, Dummy3): [ofertaexist] oferta(p,gr,o) = 100));
    @for(PO(p,o):@for(moradia(p,gr,o)|p #GT# @index(pessoas, Dummy3): [ofertaexist1] oferta(p,gr,o) = 10));
    @for(NEPO(p,o):@for(moradia(p,gr,o): [ofertanaoexist] oferta(p,gr,o) = 0));
    @for(PGATIVO(p,gr): @for(oativa(oa): @for(dativo(e): @for(periodo(t): [oft]@sum(missao(k)|@in(ArcoMi,t,oa,
    @for(nopegr(t,k,gr,d,p):[balanco] @sum(nosativos(h)|@in(ArcoMi,t,h,d,p,gr,k): alocado(t,h,d,p,gr,k)) = at
    @for(PGM(p,gr,k):@for(periodo(t):@for(dativo(e):[fixa]@sum(nosativos(h)|@in(ArcoMi,t,h,e,p,gr,k): alocad
    @for(dem(t,k,gr,d):[equi]@sum(nopegr(t,k,gr,d,p): atendida(t,k,gr,d,p)) = Equipe(gr)*demanda(t,k,gr,d));
    @for(periodo(t):@for(pessoas(p):[temp] @sum(nopegr(t,k,gr,d,p):atendida(t,k,gr,d,p)*duracao(gr)) + @sum(
    @for(IPE(p,t): [dispontempo] TempoUtilizado(p,t) <= disponibilidade(p,t));
    @for(ArcoMi(t,i,j,p,gr,k):[aloc] alocadogeral(t,i,j,p) >= alocado(t,i,j,p,gr,k));
    @for(NopeGr(t,k,gr,d,p):@bin(atendida(t,k,gr,d,p));!Restrição de binario;
    @for(VarInt(t,i,j,p):@bin(alocadogeral(t,i,j,p));!Restrição de binario;
Endsubmodel

,o) = 100));
r,o) = 10));

n(ArcoMi,t,oa,e,p,gr,k):alocado(t,oa,e,p,gr,k) <= oferta(p,gr,oa))));
,p,gr,k)) = atendida(t,k,gr,d,p) + @sum(nosativos(h)|@in(ArcoMi,t,d,h,p,gr,k):Alocado(t,d,h,p,gr,k));
gr,k): alocado(t,h,e,p,gr,k))>= @sum(nosativos(h)|@in(ArcoMi,t,e,h,p,gr,k):Alocado(t,e,h,p,gr,k))));
(t,k,gr,d));
(gr)) + @sum(VarInt(t,i,j,p): Alocadogeral(t,i,j,p)*2*tempo(i,j)) = TempoUtilizado(p,t));

```

```

File Edit Solver Window Help
[Icons]

%for(TPe(p,t): [dispon tempo] TempoUtilizado(p,t) <= disponibilidade(p,t));
%for(ArcosHi(t,i,j,p,gr,k):[aloc] alocadogeral(t,i,j,p) >= alocado(t,i,j,p,gr,k));

%for(NoPeGr(t,k,gr,d,p):%bin(atendida(t,k,gr,d,p)):'Restrição de binario';
%for(VarInt(t,i,j,p):%bin(alocadogeral(t,i,j,p)):'Restrição de binario');

Endsubmodel

calcl:

%set('terseo', 1): 'Reduce output level';
%set('looppp', 1): 'Loop optimization';

%for(pessoaquebrada(p,gr): %for(existenissao(k,gr): %insert(PGM, p, gr, k));

%for(pessoaquebrada(p,gr): %for(dem(t,k,gr,d): %insert(nopegr,t,k,gr,d,p));

%for(pessoaquebrada(p,gr): %for(gravativo(gr): %insert(pgravativo,p,gr));

%for(arco(i,j): %for(arcoa(i,j): %insert(arcoativo,i,j));

@DIVERT('SPO_Solucao.txt');
@solve(Alocacao);
@write('Período origem destino pessoa alocadogeral',%NEWLINE(1));
@writefor(VarInt(t,i,j,p) | alocadogeral(t,i,j,p) #GT# 0: VarInt(t,i,j,p), ' ', Alocadogeral(t,i,j,p), %NEWLINE(1));
@write('Período missao atividade destino pessoas Atendida',%NEWLINE(1));
@writefor(NoPeGr(t,k,gr,j,p) | atendida(t,k,gr,j,p) #GT# 0: NoPeGr(t,k,gr,j,p), ' ', atendida(t,k,gr,j,p), %NEWLINE(1));
@write('Pessoa Período Tempo Utilizado',%NEWLINE(1));
@writefor(TPe(p,t) | TempoUtilizado(p,t) #GT# 0: TPe(p,t), ' ', TempoUtilizado(p,t), %NEWLINE(1));
@DIVERT();

endcalcl

```