

## Seção de Software

---

**Virgílio José Martins Ferreira Filho**  
Departamento de Engenharia Industrial  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro – RJ  
[virgilio@ufrj.br](mailto:virgilio@ufrj.br)

---

### O USO DE SOFTWARE DE MODELAGEM AIMMS NA SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO MATEMÁTICA

**Aníbal Alberto Vilcapoma Ignácio**  
Programa de Engenharia de Produção / COPPE  
Universidade Federal de Rio de Janeiro  
Rio de Janeiro – RJ  
[avilcap@pep.ufrj.br](mailto:avilcap@pep.ufrj.br)

**Virgílio José Martins Ferreira Filho \***  
Programa de Engenharia de Produção / COPPE e  
Departamento de Engenharia Industrial  
Universidade Federal de Rio de Janeiro  
Rio de Janeiro – RJ  
[virgilio@ufrj.br](mailto:virgilio@ufrj.br)

\* *Corresponding author*/autor para quem as correspondências devem ser encaminhadas

#### 1. Introdução

A pesquisa operacional, nos últimos tempos, tem desenvolvido uma grande variedade de modelos e algoritmos de otimização para a resolução de problemas em diversas áreas como: logística, telecomunicações, produção, finanças etc.. As ferramentas de apoio à decisão se fundamentam fortemente nas técnicas de otimização, entretanto, sua aplicação prática pode acarretar diversas dificuldades na fase de modelagem e no gerenciamento eficiente dos sistemas computacionais, o que muitas vezes inviabiliza o seu acesso por parte de diversas empresas e entidades governamentais. Segundo Geoffring (1987) e Matura *et al.* (2004), os modelos matemáticos de otimização têm duas formas gerais de uso. Na primeira, o modelo é usado ao nível estratégico, implicando em uma resolução do problema usada a intervalos de tempo consideravelmente longos, o que é intrínseco em uma decisão estratégica. Na segunda, o modelo atende a uma necessidade ao nível tático-operacional, os modelos de solução são

usados periodicamente e neste nível de decisão, nem sempre há necessidade de profissionais especializados em Pesquisa Operacional. O desenvolvimento das ferramentas, para cada um dos dois tipos, é obviamente distinto. Para atender a estas demandas, existem, atualmente, diversos sistemas de modelagem que facilitam o processo de implementação e solução dos modelos utilizando linguagem algébrica, os sistemas de Suporte Analítico à Decisão.

Normalmente, o sistema de suporte analítico à decisão é um sistema interativo que enfatiza técnicas computacionais avançadas que, muitas vezes são transparentes para os usuários, oferecendo amplas facilidades de análise dos problemas por sua visibilidade externa. As aplicações típicas desse tipo de ferramenta são:

- Representação de uma realidade complexa;
- Organização e uso de grande volume de dados multidimensionais, inter-relacionados, baseados em informações próprias do problema e de diversos parâmetros associados à solução do mesmo;
- Utilização de técnicas avançadas de ferramentas de otimização e de manipulações aritméticas para se encontrar uma solução;
- Através das técnicas analíticas ou dos experimentos computacionais pode-se avaliar as conseqüências de diversas soluções, sob diferentes cenários;
- Emprego de técnicas de visualização avançadas que fornecem uma percepção da solução e/ou da complexidade do problema;
- Possibilidade de mudanças se houver maior compreensão de uma realidade mutável ou percepções melhoradas.

Em geral é necessário muito conhecimento para se usar um sistema de apoio à decisão, dominando-se a linguagem de modelagem e os parâmetros que controlam diferentes aspectos do *solver*. As pessoas experientes em Pesquisa Operacional compreendem bem os modelos e conseguem com facilidade realizar modificações e detectar erros. No entanto, a delegação dessas tarefas para usuários finais é muito complicada, já que são tarefas que requerem grande especialização.

Os sistemas de suporte analítico à decisão mais conhecidos são AIMMS, AMPL, GAMS, LINGO, MPL, OPL Studio. O Quadro 1 a seguir apresenta os endereços de Internet onde podem ser obtidas informações mais detalhadas sobre cada um deles.

**Quadro 1** – Endereço dos *sites* mais conhecidos.

Nome	Empresa	Site
AIMMS	Paragon Decision Technology	<a href="http://www.aimms.com/">http://www.aimms.com/</a>
AMPL	Bell Laboratories	<a href="http://www.ampl.com/">http://www.ampl.com/</a>
GAMS	GAMS Development Corporation	<a href="http://www.gams.com/">http://www.gams.com/</a>
Xpress Mosel	Dash Optimization Ltd	<a href="http://dashoptimization.com">http://dashoptimization.com</a>
MPL	Maximal Software	<a href="http://www.maximal-usa.com/mpl/">http://www.maximal-usa.com/mpl/</a>
ILOG OPL Studio	ILOG	<a href="http://www.ilog.com/products/oplstudio/">http://www.ilog.com/products/oplstudio/</a>
Lindo	Lindo System	<a href="http://www.lindo.com/">http://www.lindo.com/</a>

O Quadro 2 apresenta uma comparação destes principais sistemas com respeito a suas funcionalidades, considerando tanto características da modelagem, quanto da interface com o usuário e outros sistemas.

**Quadro 2** – Comparação entre os principais sistemas.

	AIMMS	GAMS	OPL Studio	Xpress Mosel	AMPL	MPL	Lingo
<b>Facilidades da Linguagem</b>							
Programação Matemática	√	√	√	√	√	√	√
Controle de Fluxo	√	√	√	√	√		√
Procedimentos	√			√	√		
Gerência de Casos	√	√					
Calendário e Horizonte de Tempo	√						
<b>Interface de Desenvolvimento</b>							
Ferramentas de Diagnóstico ( <i>Cross Reference / Debugger / Profiler</i> )	√	√			√		
<b>Tipos de Solver</b>							
LP/MIP ( <i>Linear Programming / Mixed Integer Programming</i> )	√	√	√	√	√	√	√
QP ( <i>Quadratic Program</i> )	√	√		√	√	√	√
NLP ( <i>NonLinear Programming</i> )	√	√		√ (SLP)	√	√	√
MINLP ( <i>Mixed Integer Nonlinear Program</i> )	√	√		√ (SLP)	√		√
MCP ( <i>Mixed Complementarity Programming</i> )	√	√			√		
CP ( <i>Constraint Programming</i> )			√				
Global		√					√
<b>Características da Interface</b>							
Interface Gráfica – Usuário Final	√		√				
<b>Conectividade</b>							
ODBC/OLE DB	√		√	√	√	√	
XML	√					√	
Funções Externas	√	√			√		
<b>Desenvolvimento</b>							
Unix / Linux		√	√	√	√	√	√

Fonte: AIMMS

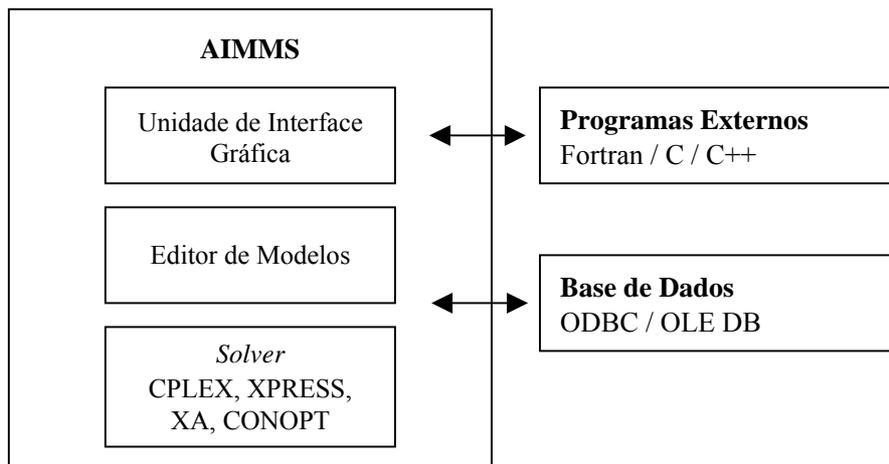
Neste artigo são apresentadas as características funcionais do AIMMS (*Advanced Integrated Multidimensional Modeling Software*). Segundo Bisschop (2002), o AIMMS é uma ferramenta que inclui as duas formas gerais de uso dos modelos matemáticos, já que permite construí-los ao nível estratégico e ao nível tático operacional, através de ferramentas de apoio à decisão iterativa para a solução dos problemas. É também apresentado um exemplo

que exibe uma importante característica iterativa do *Solver* e da linguagem AIMMS, que pode ser usada, por exemplo, para se explorar relaxações Lagrangeanas em problemas de forte complexidade computacional.

Uma versão de teste (*try version*), que pode ser usada durante 30 dias, pode ser obtida na página eletrônica do AIMMS. Esta versão suporta até cerca de 1200 variáveis.

## 2. AIMMS

O AIMMS – *Advanced Integrated Multidimensional Modeling Software* oferece um ambiente de desenvolvimento, no qual pessoas experientes em modelagem podem criar aplicações funcionais, prontas para serem usadas por pessoas mais leigas ou usuários finais. Como um ambiente de desenvolvimento de apoio analítico à decisão, o AIMMS possibilita uma combinação de características de métodos matemáticos e de ferramentas de desenvolvimento, tais como: o explorador de modelos num ambiente gráfico para se construir e manter aplicações de modelagem complexa; os métodos de solução exata; os procedimentos que permitem interagir com as linguagens C, C++, Fortran e ferramentas de interface com as bases de dados através de ODBC/OLE DB (ver na Figura 1).



**Figura 1** – Componentes do AIMMS.

O sistema AIMMS proporciona ambiente adequado para:

- Modelar problemas de otimização;
- Criar uma interface de usuário final ajustada às necessidades do usuário;
- Fornecer um ambiente de execução em tempo real para os que tomam as decisões.

O sistema AIMMS oferece uma notação de índice que possibilita capturar a complexidade de problemas reais. Além disso, permite expressar muitos cálculos complexos de uma maneira compacta, sem a preocupação com o gerenciamento da memória ou considerações de estocagem de dados. Uma das principais características do AIMMS é sua capacidade de

especificar e resolver modelos de otimização com restrições lineares e não lineares. Usando-se a mesma notação compacta disponível para declarações de procedimentos, as restrições simbólicas podem ser formuladas de modo simples e conciso. Com apenas uma simples declaração, um modelo de otimização pode ser transferido para o *solver* e resolvido por ferramentas, tais como CPLEX, XA, CONOPT e XPRESS.

## 2.1 Características de Linguagem

O AIMMS apresenta uma linguagem de programação que permite interagir com o usuário na modelagem e na construção de procedimentos que antecedem e/ou interagem na execução de um modelo matemático.

As características mais importantes dessa linguagem são:

- Conjunto de funções matemáticas, estatísticas e financeiras;
- Capacidade de expressar modelos baseados no tempo, através do uso de calendários e horizontes de tempo, inclusive suporte para intervalos de tempo dinâmico com agregação e desagregação automáticas;
- Capacidade de associar unidades de medida com identificadores de modelo, assegurando assim a consistência das unidades dentro das expressões;
- Suporte de modelagem para problemas mistos;
- Suporte de módulos, para facilitar o melhor desenvolvimento de um modelo AIMMS por múltiplos usuários. Os módulos são basicamente seções do modelo com um espaço de nome separado, prevenindo assim a ocorrência de colisão de nomes quando um ou mais módulos são importados em um modelo AIMMS;
- Os arquivos são salvos automaticamente a intervalos fixos de tempo, o que permite se retornar a versões anteriores com muita facilidade;
- Suporte de chamadas de funções e procedimentos externos que não estão implementados na linguagem AIMMS.

## 2.2 Interface do Usuário

O AIMMS dispõe de um ambiente de desenvolvimento integrado com todas as facilidades de um ambiente gráfico apresentado pelas ferramentas do Windows.

### **Construtor Integrado GUI (*Graphical User Interface*)**

O AIMMS oferece uma ferramenta integrada para a construção de uma interface gráfica personalizada (GUI) para o usuário, em uma aplicação de apoio à decisão. A tela final do usuário pode ser criada de modo simples, clicando-se e arrastando-se, tanto objetos gráficos comuns como também tabelas, gráficos e curvas, todos associados aos identificadores multidimensionais no modelo do usuário. Outros objetos incluem gráficos Gantt para se visualizar no tempo, o fluxo de atividades.

## Ferramentas GUI

### Ferramenta Administrativa de GUI:

A interface do usuário final é um aspecto importante na construção de diversos aplicativos. Esta ferramenta permite organizar todas as telas existentes em uma aplicação AIMMS, onde se pode ter aplicações em estrutura de árvore e o relacionamento das páginas definindo uma estrutura hierárquica, como páginas-pai, páginas-filho, próxima página ou página anterior. Esta ferramenta permite uma maneira fácil de navegar por todas as páginas de um determinado aplicativo como também adicionar outras, apagá-las ou modificá-las, respeitando-se uma política estrita de permissão.

### Administrador de Modelos de Página:

Os modelos (*template*) das páginas são usados para se especificar propriedades de objetos. Usando-se esta ferramenta se pode criar e administrar grande número de objetos (páginas) com características similares, como por exemplo esquema de cores, criação de áreas para títulos, ícones ou logotipos. Os *templates* podem ser organizados em estruturas hierárquicas, podendo ser administrados por esta ferramenta.

### Construtor de Menu:

O editor de menus é uma das ferramentas que permite se construir e se manter menus personalizados como menu de barras (*menubars*, *toolbars* e *popup*). Quaisquer *templates* ou *pages* podem ser embutidos em um menu personalizado com estruturas hierárquicas.

## 2.3 Manipulação dos Dados

As facilidades de gerenciamento de dados do AIMMS estão concentradas em conceitos de *Cases* e Conjuntos de Dados. Os *Cases* são um conjunto de dados associados com os identificadores dos modelos matemáticos e o Conjunto de Dados corresponde a um conjunto de informações relacionadas a uma categoria como demanda, suprimento, etc. Uma associação de Conjunto de Dados pode formar um *Case* ou cada Conjunto de Dados pode ser compartilhado por mais de um *Case*, permitindo análises com critérios condicionais (*if ... then*) que podem ser realizadas através de um grande número de *Cases* e serem executadas em *batch*.

O AIMMS fornece um conjunto de ferramentas de conexão para se incorporar dados no sistema de apoio à decisão, através do ODBC e do OLE DB. Procedimentos simplificados ajudam ao usuário relacionar as colunas da base de dados com os diversos identificadores multidimensionais do modelo. Uma vez estabelecido o relacionamento acima, as instruções de escrita e de leitura podem ser realizadas na base de dados.

## 2.4 Resolução de Modelos Matemáticos

Um modelo matemático consiste de um conjunto de variáveis a serem determinadas; uma coleção de restrições que devem ser satisfeitas; e uma função objetivo para ser otimizada. A meta desse modelo é encontrar uma solução, com o auxílio de um *solver*, tal que a função objetivo assuma um valor ótimo (mínimo ou máximo). Dependendo de diferentes tipos de

características das variáveis e restrições, um programa matemático em AIMMS pode ser classificado, de acordo com as seguintes condições:

- Se a função objetivo e todas as restrições contiverem somente expressões lineares (em termos de variáveis) e se todas as variáveis puderem assumir valores contínuos dentro de sua série, então o programa é *linear*.
- Se alguma variável em um programa linear puder assumir somente valores inteiros, então o programa é *inteiro linear misto*.
- Se o objetivo for uma função quadrática em termos de variáveis, enquanto as restrições forem lineares, então o programa é *quadrático*.
- Se o objetivo não for nem linear nem quadrático ou se algumas das restrições contiverem expressões não lineares, o programa é *não linear*.

O AIMMS chama automaticamente o *solver* apropriado para encontrar uma solução ótima.

### **Declaração e Atributos do MODELO MATEMÁTICO**

No AIMMS, um modelo matemático é definido dentro de um conjunto de atributos previamente definidos. O atributo DIRECTIONS permite escolher entre maximizar e minimizar a função objetivo. O atributo VARIABEL define o conjunto ou um subconjunto de variáveis que fazem parte do modelo a ser executado. Através do atributo OBJETIVE, é definida a variável como a função objetivo do Modelo. Seu valor pode ser uma referência para uma variável já definida ou para uma expressão em função das variáveis. Se não se especificar uma função objetivo, o modelo matemático é resolvido para se encontrar uma solução viável. O atributo DIRECTION, quando usado em conjunto com uma função objetivo, serve para indicar se o *solver* deve minimizar ou maximizar essa função. Com o atributo VARIABLES se pode especificar qual o conjunto de variáveis que deve ser incluído em um programa matemático. Este deve ser um conjunto pré-definido a partir de todas as variáveis do sistema (*AllVariables*) ou um subconjunto deste. O conjunto *AllVariables* é pré-definido pelo AIMMS e contém os nomes de todas as variáveis declaradas no modelo e seu conteúdo não pode ser modificado, na fase de compilação. Se o atributo VARIABLES for designado como um subconjunto do conjunto *AllVariables*, o AIMMS tratará todas as variáveis de fora do conjunto como se elas fossem parâmetros. O atributo NONVAR de uma variável em um programa matemático permite alterar o estado individual de uma variável, sem levar em consideração as classes de variáveis.

Existem alguns casos nos quais uma “variável” toma um valor fixo dentro de um modelo matemático. Nesse caso o AIMMS define uma restrição de igualdade associada ao valor fixo da variável. Tanto as variáveis quanto as restrições estão pré-definidas no conjunto *AllVariables* e *AllConstraints*, respectivamente. Pode-se acrescentar uma variável definida ao conjunto de variáveis e de restrições de um programa matemático, independentemente, nas seguintes condições:

- Se houver omissão de uma variável definida em um conjunto de variáveis de um dado modelo matemático, todas as ocorrências da variável serão fixadas a seu valor corrente.
- Se houver omissão de uma variável definida em um conjunto de restrições de um dado modelo matemático, a restrição definida não será gerada.

Com o atributo **CONSTRAINTS** pode-se especificar quais restrições fazem parte do programa matemático. Seus valores devem ser o conjunto *All-Constraints* (todas as restrições definidas no sistema) ou um subconjunto deste. Esse conjunto contém os nomes de todas as restrições declaradas e os nomes de todas as variáveis que tiverem um atributo de declaração. Seu conteúdo é calculado no tempo de compilação e não pode ser alterado no tempo de execução. A definição do conjunto de restrições segue as seguintes condições:

- Se for feita especificação do conjunto *All-Constraints*, o AIMMS gerará restrições individuais para todas as restrições e variáveis declaradas com uma definição.
- Se houver especificação de um subconjunto do *All-Constraints*, o AIMMS somente gerará restrições individuais para as restrições declaradas e para as variáveis definidas nesse subconjunto.
- Se o programa matemático tiver um objetivo que seja uma variável definida, sua definição será acrescentada ao conjunto de restrições geradas durante a geração.

#### Álgebra do Modelo:

Alterando-se o conteúdo dos conjuntos de identificadores que entraram nos atributos **VARIABLES** e **CONSTRAINTS** de um programa matemático, pode-se produzir uma forma simples de *álgebra do modelo*. Melhor dizendo, pode-se investigar os efeitos da adição ou remoção de restrições e/ou variáveis a partir de uma interface gráfica proporcionada pelo AIMMS.

Com o atributo **TYPE** de um modelo matemático pode se definir um tipo de solução adotada. Quando o tipo especificado não é compatível com o programa gerado, o AIMMS retornará uma mensagem de erro.

#### Tipos Disponíveis:

Os *solvers* disponíveis pelo AIMMS são descritos no Quadro 3. Quando o tipo **rmip** é especificado, todas as variáveis inteiras são tratadas como contínuas dentro de seus limites. O tipo **rmip** é uma versão global do atributo **RELAX** associado com variáveis individuais. Os tipos **ls** e **nls** só podem ser selecionados na ausência do atributo **OBJECTIVE**.

**Quadro 3** – Tipos de *solver* disponíveis no AIMMS.

<b>Lp</b>	Programa Linear
<b>Ls</b>	Sistema Linear
<b>Qp</b>	Programa Quadrático
<b>Nlp</b>	Programa Não Linear
<b>Nls</b>	Sistema Não Linear
<b>Mip</b>	Programa Inteiro Misto
<b>Rmip</b>	Programa Inteiro Misto Relaxado
<b>Minlp</b>	Programa Inteiro Misto Não Linear
<b>Ncp</b>	Programa de Complementaridade Misto

Pode-se usar o atributo `CONVENTION` para se especificar a convenção da unidade que se quer usar para a escala das variáveis e restrições de um programa matemático.

Com o atributo `VIOLATION PENALTY` se pode instruir o AIMMS a adicionar automaticamente termos artificiais às restrições para ajudar a resolver e/ou localizar inviabilidades em um programa.

O seguinte exemplo ilustra um modelo típico:

#### MATHEMATICAL PROGRAM

Name : TransportModel	Objective : TransportCost
Direction : Minimize	Constraints : AllConstraints
Variables : AllVariables	Type : lp

Neste exemplo define-se um modelo linear *TransportModel* que é construído a partir de todas as restrições e variáveis, definidas no sistema. A variável *TransportCost* define a função objetivo a ser minimizada.

O AIMMS é dotado de funções de acompanhamento que permitem avaliar algumas características do modelo, que podem ser utilizados para:

- Conseguir informações gerais sobre o programa matemático gerado;
- Obter informação sobre o processo de solução;
- Determinar quando e como ativar um procedimento *callback*.

As funções mais utilizadas no *Solver* são:

- Objective – favor da função objetivo corrente;
- LinearObjective – valor da solução objetivo relaxada corrente;
- ProgramStatus – estado corrente do modelo;
- SolverStatus – estado corrente do *solver*;
- Iterations Current – número de iterações;
- NumberOfConstraints – número de restrições individuais;
- NumberOfVariables – número de variáveis individuais;
- NumberOfNonzeros – números de não-zeros;
- NumberOfInfeasibilities – número final de soluções (*Solver*) das inviabilidades.

Após a execução dos *Solver*, o AIMMS fornece o controle do sistema para o usuário. Este processo é chamado de *Callback*. O AIMMS também proporciona um conjunto de funções para acompanhar este processo. Os exemplos mais comuns são:

- CallbackProcedure – nome do procedimento *Callback*;
- User CallbackIterations – retorno ao *callback* após certo número de iterações;
- CallbackStatusChange – nome do procedimento *callback* chamado depois de uma mudança de estado;
- CallbackNewIncumbent – nome do procedimento *callback* chamado para todo novo valor encarregado;
- CallbackReturnStatus – retorno ao estado do *callback SolverCalls*.

### 3. Exemplo de Aplicação

Com o objetivo de apresentar as características básicas do AIMMS, considera-se o problema de localização de concentradores apresentado por Ferreira Fo. & Galvão (1998). Dado um conjunto de localizações possíveis para os concentradores, determinar o número e o lugar onde os concentradores deverão ser utilizados e quais usuários deverão ser alocados a cada um deles sem violar a capacidade dos concentradores minimizando o custo total. A formulação é apresentada a seguir.

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m v_j y_j$$

Sujeito a

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (1)$$

$$\sum_{i=1}^n a_{ij} x_{ij} \leq b_j, \quad j = 1, 2, \dots, m \quad (2)$$

$$x_{ij} - y_j \leq 0, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (3)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (4)$$

O primeiro componente da função objetivo representa os custos de comunicação, o segundo componente captura custos fixos de instalação. A primeira restrição assegura que cada usuário deve ser conectado a, exatamente, um concentrador. A segunda restrição assegura que os usuários só podem ser alocados a concentradores com capacidades adequadas. A terceira restrição assegura que os nós sejam alocados somente a locais onde existam concentradores. E a última restrição expressa a natureza binária das variáveis de decisão

Este modelo matemático pode com muita facilidade ser modelado no AIMMS, através da ferramenta “**model explorer**”. Para se definir o modelo, primeiramente deve-se definir os índices (*set*) a serem usados, como por exemplo o conjunto de índices I e J que são chamados terminais e concentradores, respectivamente. Na declaração de índices também podem ser definidos outros índices como o *tableVersion* que identifica os conjunto de instâncias a serem consideradas. Uma vez definidos esses índices, pode-se definir, através do *Select Type of Identifier*, as variáveis, constantes, restrições em sua representação algébrica como mostrado na Figura 2.

A restrição 2 do modelo (Figura 3) é definida, usando-se sua representação algébrica para permitir a separação entre os dados e a construção do modelo.

Os parâmetros de entrada do modelo ( $c(i,j)$ ,  $a(i,j)$ ,  $v(j)$ ) podem estar associados a uma base de dados, para a qual devem ser definidos os apropriados ODBCs. Para esse exemplo considera-se uma base de dados no Microsoft Access que é mapeada para o parâmetro de custos como visto na Figura 4. O ODBC usado é definido como BD\_PLC e sua ligação com a base de dados é obtida pela associação de *Custo\_alocação\_R* com os parâmetros  $c(i,j)$ , através do mapeamento das colunas da tabela *Custo\_alocação*, com os índices  $i, j$  e o valor  $c$ .

Na definição do modelo matemático, considera-se uma variável como função objetivo, uma direção (maximizar ou minimizar) e um subconjunto de variáveis e restrições. No exemplo, foram consideradas todas as variáveis e todo o conjunto de restrições, sendo o tipo de método de solução definido como MIP (ver Figura 5).

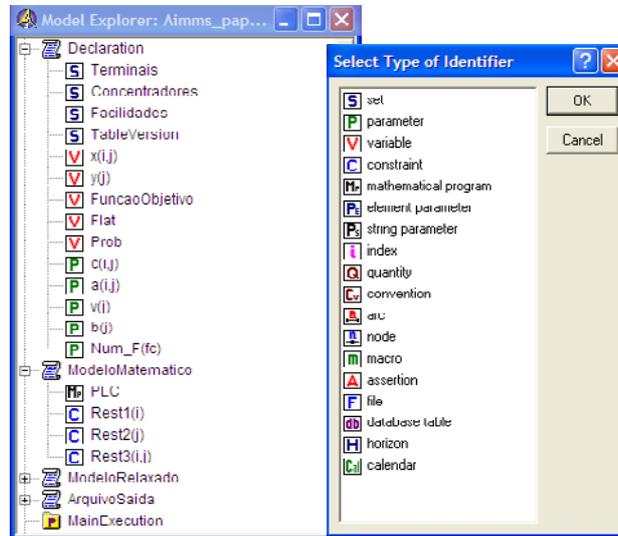


Figura 2 – Definição do modelo.

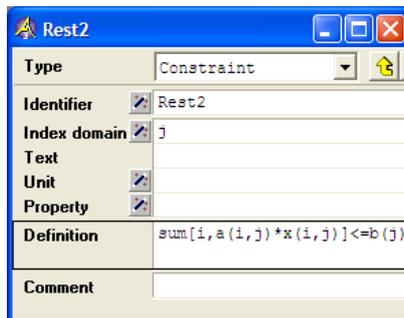


Figura 3 – Conjunto de restrições 2.

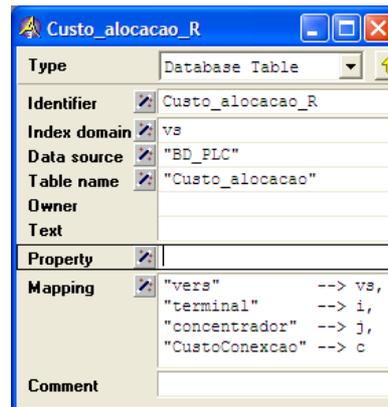


Figura 4 – Ligação com uma base de dados através do ODBC.

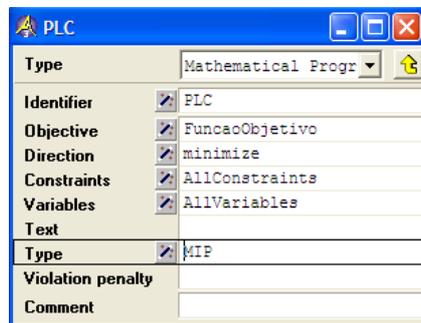


Figura 5 – Definição do modelo matemático.

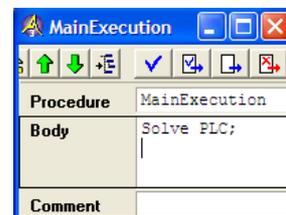


Figura 6 – Execução do modelo.

Para se resolver o modelo, deve-se definir a linha de comando Solve PLC, no procedimento “MainExecution”, como pode ser visto na Figura 6.

Em muitas ocasiões, existe o interesse em encontrar limites inferiores e superiores para um determinado problema, ou devido à complexidade do problema ou porque esses limites constituem passos intermediários na ferramenta de solução. O AIMMS fornece um conjunto de facilidades para se lidar com estes tipos de situação. O AIMMS, por incorporar uma linguagem de programação, permite se implementar heurísticas de solução, utilizando-se os métodos de solução exatos, fornecidos pelo *solver*. A seguir, é apresentada uma heurística de subgradiente para se encontrar limites superiores para um modelo de relaxação Lagrangeana para o problema apresentado.

Ferreira Fo. & Galvão (1998) sugerem que a relaxação da restrição (1) é a mais adequada para o problema. Considerando-se esta restrição, define-se uma nova função objetivo, como apresentado na Figura 7, onde  $\text{Alf}(i)$  são os multiplicadores de Lagrange.

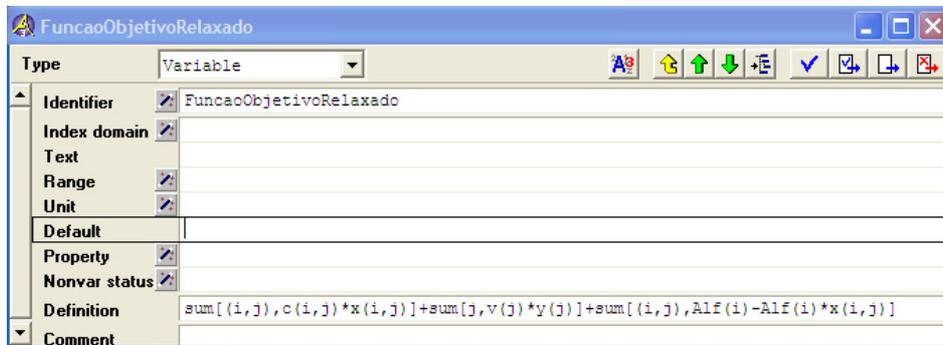


Figura 7 – Função objetivo para o problema Lagrangeano.

Para se estabelecer o espaço viável para o problema Lagrangeano, um subconjunto de Restrições e de variáveis é definido. Como foi relaxada a restrição 1, o conjunto de restrições válidas é composto pelas restrições 2 e 3 (ver Figura 8).

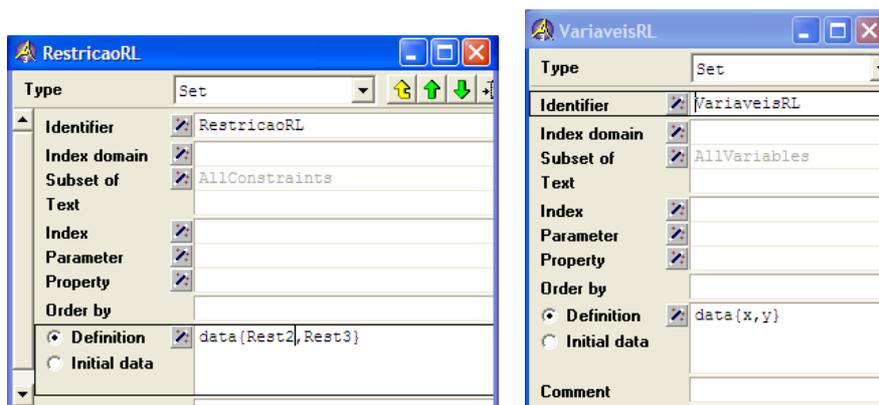


Figura 8 – Definição do subconjunto de restrições e de variáveis do problema relaxado.

Uma vez definidos a função objetivo, o conjunto de restrições e o conjunto de variáveis, pode ser definido com facilidade o modelo relaxado, como mostrado na Figura 9.

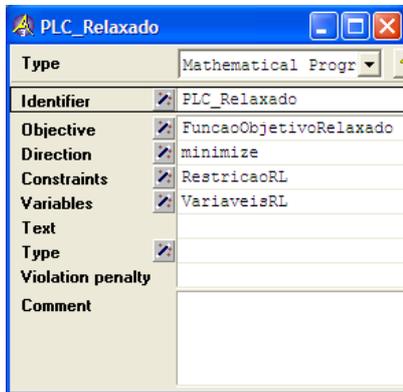


Figura 9 – Modelo de relaxação Lagrangeana.

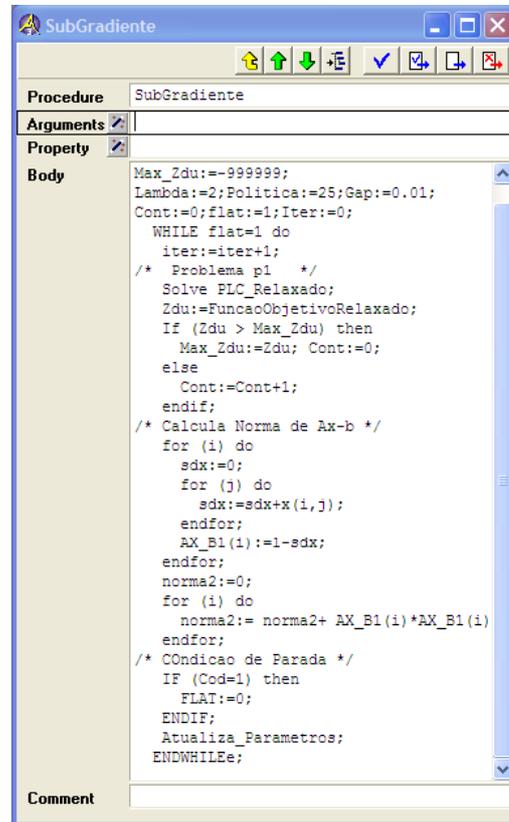


Figura 10 – Algoritmo subgradiente.

Com a definição deste modelo, pode ser construída facilmente, a heurística de subgradiente mostrada no procedimento da Figura 10. Os subproblemas resultantes da decomposição do problema Lagrangeano, podem ser resolvidos pelo *solver* do AIMMS, iterativamente. Esta grande facilidade permite se explorar diversas relaxações, considerando-se as diversas alternativas associadas a cada uma das restrições do modelo, sem a preocupação, no início, com a construção do método de solução associado a cada subproblema gerado. É importante ressaltar que métodos de solução desenvolvidos através da análise de cada subproblema podem ser muito mais eficientes, no que diz respeito a tempo computacional, do que o *solver* do AIMMS. Porém, na fase exploratória do modelo, a alternativa de solução do *solver* torna-se uma ferramenta muito importante.

O AIMMS por ser uma linguagem de programação permite a construção de qualquer procedimento ou função que se faça necessário para se desenvolver uma heurística de solução para problemas complexos. A facilidade que o AIMMS oferece no uso do *solver* em um modelo matemático, através de uma instrução dentro da linguagem, abre muitos horizontes em termos de desenvolvimento de novas heurísticas que também podem fazer uso de procedimentos já prontos em C, C++, e FORTRAN.

#### 4. Conclusões

Os sistemas de suporte analítico à decisão permitem melhorar a tomada de decisões, através do uso de modelos matemáticos que permitem o uso das ferramentas da Pesquisa Operacional. O AIMMS fornece ferramentas para se desenvolver e se construir sistemas de apoio à decisão, desde procedimentos que ajudam na modelagem, até aplicativos com interface amigável para usuários finais. No que diz respeito ao desenvolvimento do sistema, o AIMMS fornece recursos para a modelagem de problemas, para a manipulação de dados, para a criação de cenários e para a resolução de formas exatas de modelos matemáticos, tudo isso através de uma linguagem que utiliza como instrução o *Solver*, que permite também a construção de heurísticas de solução para problemas computacionalmente caros. No que diz respeito à criação de um sistema de apoio à decisão, o AIMMS fornece todo um conjunto de interfaces gráficas que possibilita, ao usuário final, uma melhor visualização do problema de forma abrangente.

#### Referências Bibliográficas

- (1) Bisschop, J. & Entriken, R. (2002). *AIMMS the Modeling System*. Paragon Decision Technology. <<http://www.aimms.com>>.
- (2) Ferreira Filho, V.J.M. & Galvão R.D. (1998). A tabu search Heuristic for the Concentrator Location Problem. *Location Science*, **6**, 211-231.
- (3) Geoffrion A. (1987). Introductions to Structured Modeling. *Management Science*, **33**(5), 547-588
- (4) Matura, A.; Ferrer, J.C. & Baraňao, F. (2004). Design and implementation-based decision support system generator. *European Journal of Operational Research*, **154**, 170-183.