99

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

MARIA DA PENHA BAIÃO SANTOS NEVES

Explorando Alternativas de Execução para Melhorar o Desempenho Econômico-Financeiro de Projetos Lineares de Construção Civil

Dissertação submetida ao
Programa de Pós-Graduação em Engenharia de Produção
como requisito para obtenção do
Título de Mestre em Engenharia de Produção

0.221.353-6
UFSC-BU

Florianópolis
1993

EXPLORANDO ALTERNATIVAS DE EXECUÇÃO PARA MELHORAR O DESEMPENHO ECONÔMICO-FINANCEIRO DE PROJETOS LINEARES DE CONSTRUÇÃO CIVIL

Maria da Penha Baião Santos Neves

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA À OBTENÇÃO DO TÍTULO DE **MESTRE EM ENGENHARIA**, ESPECIALIDADE ENGENHARIA DE PRODUÇÃO, E JULGADA ADEQUADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

OSMAR POSSAMAI, Dr.
Coordenador do Curso

BANCA EXAMINADORA:

PLINIO STANGE, Dr.
Presidente

EDGAR AUGUSTO LANZER, Ph.D.

LUIZ FERNANDO M. HEINECK, Ph.D.

LUIS OTÁVIO GUEDERT, M.Eng.

Aos meus pais
Lilian e Toninho **e**
Aos meus irmãos
Flávia e Tony

AGRADECIMENTOS

Ao ***Professor Plinio Stange***, pela orientação e incentivo durante a etapa final de desenvolvimento do trabalho.

Ao ***Professor Luiz Fernando Heineck***, pelas opiniões sinceras sobre o trabalho e pelo fornecimento de material bibliográfico.

Ao ***Professor Edgar Augusto Lanzer***, pelas sugestões apresentadas à formulação do modelo matemático que faz parte do trabalho.

A ***Luís Otávio Guedert***, por ter acreditado no trabalho, dando um apoio decisivo à conclusão desta pesquisa.

Aos ***funcionários e demais professores*** do Curso de Pós-Graduação em Engenharia de Produção, pelo apoio durante todo o mestrado.

À ***CAPES***, pelo apoio financeiro através da bolsa de estudo.

À ***Aldanei***, pela ajuda na organização da apresentação escrita e revisão das referências bibliográficas do trabalho.

À ***Rosany Martins***, pela revisão e depuração das principais rotinas do sistema computacional desenvolvido.

À ***família e aos amigos***, pelo carinho e incentivo sempre presentes.

SUMÁRIO

LISTA DE FIGURAS

CAPÍTULO 2

CAPÍTULO 4

CAPÍTULO 6

CAPÍTULO 7

LISTA DA TABELAS

CAPÍTULO 6

CAPÍTULO 7

LISTA DE QUADROS

ANEXO I

ANEXO II

RESUMO

Este trabalho apresenta um sistema computacional que foi desenvolvido com o objetivo de auxiliar na tarefa de planejamento e programação de projetos lineares de Construção Civil. Motivado pelas mudanças mercadológicas observadas neste setor industrial, o método proposto incorpora aspectos externos ao canteiro de obras à medida que procura compatibilizar os fluxos de recursos financeiros necessários e disponíveis do projeto. Para melhor compreender o método de programação desenvolvido, é apresentado e discutido o modelo matemático que lhe dá suporte. Por fim, o trabalho apresenta uma série de simulações realizadas com o sistema, procurando evidenciar, através dos resultados, a aderência do modelo à realidade da tarefa de programação de obras.

ABSTRACT

This dissertation introduces a computational system developed with the objective of aiding planning and programming of linear projects in building activities. Due to market changes that have been observed in this industrial field, external aspects to the site environment are incorporated into the proposed method, since the main objective is to assure the compatibility between necessary and available financial resources related to a specific project. For better understanding, the mathematical model that supports the method is presented and discussed. This work ends with several simulations that have been performed with the system proposed, trying to explicitate, through numeric results, the compatibility between the model proposed and the actual activity of programming building projects.

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo é apresentado o contexto no qual se insere a pesquisa realizada. A problemática do setor industrial Construção Civil é brevemente discutida, dando-se ênfase à falta de planejamento observada neste setor da economia. Feito isso, define-se o problema que este trabalho propõe-se a resolver, apresentando-se em seguida o objetivo, metodologia e justificativa desta pesquisa.

1.1- O Problema

" A alta incidência de fracassos de empresas que operam na construção é um fenômeno constatado não só no Brasil, como em outros países ocidentais. A razão deste fato não está ligada à incompetência técnica daqueles que operam na atividade, mas predominantemente na incapacidade de gestão" (Balarine,1990).

Essa assertiva de Balarine evidencia a deficiência do gerenciamento na Construção Civil. Também é dessa opinião Rocha Lima Junior (1987) quando diz que " muito do que se conhece das técnicas aplicadas pelas empresas está baseado numa visão distorcida do que é o processo do gerenciamento".

O processo de decidir na empresa de Construção Civil envolve dois grandes segmentos: o sistema de gestão e o sistema de produção.

O sistema de produção, que nos interessa mais de perto, está vinculado ao teor da decisão e ao resultado imediato que se tem quando ela é tomada. Cada ação de produção é resultante de um conjunto de decisões que permeiam os níveis hierárquicos da empresa. Desde a decisão de nível mais abrangente até aquelas vinculadas especificamente ao ato de produzir, ocorre na empresa um processo de manipulação de informações, absorção de diretrizes e tomadas de decisões.

O fluxo de decisões, de cima para baixo, exige instrumentação, através de informações e procedimentos e/ou técnicas para manipulação das informações.

Então, para produzir um plano ou programa de execução para qualquer projeto de construção, é necessário determinar, a priori, as informações e técnicas de programação mais compatíveis com o tipo de projeto e nível hierárquico de decisão.

Quanto mais alto o nível hierárquico de decisão, menos detalhado deve ser o programa de execução produzido e mais informações do ambiente externo à empresa devem ser consideradas no processo de tomada de decisão.

No plano financeiro, quando se fala em informações de origem externa refere-se à capacidade de pagamento do mercado, às regras de desembolso do Sistema Financeiro de Habitação e à estratégia de administração do capital de giro da empresa.

Traduzir isso tudo em um programa de execução para o projeto, significa procurar uma alternativa de programação que apresente uma melhor compatibilização entre os fluxos de recursos deste projeto.

Assim, define-se o problema com a seguinte pergunta: **como produzir um plano de execução tático para projetos lineares de Construção Civil, tentando incorporar no processo decisório aspectos externos ao canteiro de obras ?**

1.2- Objetivos do Trabalho

O objetivo principal desta pesquisa, respondendo ao problema definido no item anterior, é produzir uma ferramenta capaz de gerar um plano tático de execução para projetos lineares de Construção Civil, tentando minimizar a diferença entre a curva de agregação de recursos necessários à execução do projeto e a curva de agregação de recursos disponíveis.

Segundariamente, o trabalho pretende mostrar a validade e as limitações dos modelos formais da pesquisa operacional dentro do ambiente de planejamento e programação de projetos lineares de Construção Civil. Mais especificamente, defende-se a aplicabilidade dos modelos matemáticos para a resolução do problema de que trata esta pesquisa.

1.3- Justificativa e Motivação para o Tema

O trabalho justifica-se, a princípio, pela necessidade de melhorar a eficiência da Indústria da Construção no Brasil. O país apresenta um enorme déficit de moradia - em torno de 10 milhões

nos próximos dez anos (SECOVI apud Werna, 1989) - além de falta de hospitais, escolas e outras obras públicas. Também é grande a demanda por obras de infra-estrutura como: água, luz, saneamento e transporte.

Aliada à deficiência em qualidade e quantidade por obras civis, é um fato a escassez de recursos disponíveis para investimento nesse setor. A quantidade de dinheiro que o Governo tem investido em tais projetos, nos últimos vinte anos, tem sido drasticamente cortada devido à grande dívida externa do país (Roddick apud Formoso,1988).

Também a nível micro, a crise político-econômica evidenciou a ineficiência das empresas de construção do país. Segundo Cardoso (1993), as construtoras brasileiras precisam melhorar seu desempenho, sobretudo por uma questão de sobrevivência. Fazendo-se um breve histórico (Neves, Guedert,1993) , pode-se compreender melhor a realidade que se nos apresenta.

Durante muitas décadas, a produção industrial encontrava restrições apenas na sua própria capacidade produtiva, uma época de mercados carentes e bom poder aquisitivo. A ênfase da programação da produção concentrava-se na minimização dos tempos necessários à produção e por conseqüência na maximização da quantidade produzida. Daí surgiram as técnicas de programação PERT/CPM e as antigas filosofias de produção.

Com a nova realidade de aumento da oferta de bens e serviços e o conseqüente acirramento da competitividade, ganharam corpo as restrições de mercado. O objetivo das indústrias passou a ser o de produzir o que o mercado necessita, dentro de sua capacidade de pagamento e com a qualidade que o consumidor exige. A programação da produção submete-se às restrições externas. Surgem então as novas filosofias de fabricação.

Na construção habitacional brasileira, os efeitos dessa nova realidade ficaram encobertos pelo grande déficit habitacional e, principalmente, pela existência dos financiamentos que viabilizavam a atividade. Com a falência do Sistema Financeiro de Habitação (SFH) e a grande recessão da economia, as empresas de construção brasileiras foram colocadas bruscamente diante das restrições de mercado.

Então, as empresas do setor começaram a procurar estratégias alternativas, buscando, quase sempre, a viabilização dos empreendimentos através do autofinanciamento. Agora as empresas precisam identificar no seu mercado o que devem produzir, o preço e a forma que os clientes estão dispostos a pagar pelas unidades habitacionais.

Nesse contexto, a programação das obras ganha importância. Deixa de ser uma simples tarefa de engenharia e passa a ser uma atividade multifuncional, integrante do processo de concepção e viabilização dos empreendimentos. As principais variáveis de decisão da tarefa de programar passam a ser de ordem estratégica, condicionadas por fatores externos ao canteiro de obras. Faz-se necessário então o desenvolvimento de métodos de planejamento e programação de obras mais adequados a essa nova realidade.

1.4- A Metodologia Utilizada

Basicamente, o método de programação proposto neste trabalho é obtido a partir da construção de um modelo matemático. Este modelo é formulado através da incorporação das variáveis de programação da Técnica da Linha de Balanço e da curva de agregação de recursos necessários à execução do projeto. A curva assim obtida é então comparada com a curva de recursos disponíveis fornecida pelo usuário. Dessa forma, a diferença entre as curvas de agregação funciona como medida de qualidade para comparar planos de execução alternativos para o projeto em questão.

O modelo matemático resultante é um modelo de programação não linear que foi resolvido pelo algoritmo MINOS 5.0 contido no software GAMS - General Algebraic Modeling System - versão 2.25.

Para preparar a entrada de dados, necessários ao modelo, e transformar os valores ótimos resultantes em um plano de execução inteligível para o projeto, foi confeccionado um programa computacional em TURBO PASCAL 6.0.

Para a validação do método de programação proposto foram simulados noventa projetos fictícios. Tendo os dados de entrada sob controle, pôde-se verificar a aderência do método à realidade de programação e comparar os programas ***ótimos*** resultantes com programas de execução que programadores experimentados possivelmente obteriam para esses projetos.

1.5- Limitações do Trabalho

O método de programação aqui desenvolvido, como já mencionado anteriormente, limita-se a programar projetos lineares de Construção Civil tais como: estradas, redes de água e redes de esgoto ou projetos com características repetitivas tais como: conjuntos habitacionais de casas ou blocos de apartamentos e também edifícios altos, contendo muitos pavimentos iguais.

Operacionalmente, sua principal limitação encontra-se no nível de agregação das atividades da rede unitária do projeto. Conforme será apresentado e discutido no Capítulo 4, o número de atividades da rede determina o porte do modelo matemático embutido no método de programação desenvolvido e, em última instância, a convergência do modelo. À medida que o tempo de convergência do modelo matemático pode inviabilizar computacionalmente o método de programação proposto, a preocupação em limitar o número de atividades da rede unitária mostra-se relevante.

Deve ser lembrado, no entanto, que o método de programação propõe-se a gerar um plano tático para o projeto, justificando portanto um maior nível de agregação para as atividades da rede unitária. Considerando-se também a rápida evolução da informática, pode-se prever que a limitação operacional do método proposto tende a desaparecer em pouco tempo.

1.6 - A Estrutura do Trabalho

O trabalho está dividido em duas partes principais. A primeira parte compreende os Capítulos 1 a 3, onde é apresentada a base teórica sobre o assunto em questão. A segunda parte do trabalho, os Capítulos 4 a 8, descreve o desenvolvimento do método de programação de projetos lineares de Construção Civil proposto nesta pesquisa.

No Capítulo 2 o assunto é o Gerenciamento e a Construção Civil. Nele apresentam-se as principais técnicas e ferramentas de planejamento e programação de projetos, dando-se ênfase à adequação ou não destas técnicas ao processo produtivo da construção. Encerrando o capítulo, é apresentada de forma detalhada a Técnica da Linha de Balanço - a ferramenta de programação que serve de base para construção do método proposto.

O Capítulo 3 apresenta e discute os principais modelos de otimização produzidos dentro do ambiente de planejamento e programação de projetos. Ainda, classifica os modelos segundo o aspecto - econômico, financeiro ou operacional - que cada um procura otimizar.

O Capítulo 4 trata exclusivamente do modelo matemático incorporado ao método proposto. Apresenta inicialmente os dados de entrada e as variáveis de decisão do modelo, com suas respectivas unidades de medida. Em seguida, apresenta o modelo propriamente dito - a função objetivo e as restrições. Terminando o capítulo, é realizada uma análise crítica do modelo de otimização desenvolvido.

No Capítulo 5 é apresentado e analisado o algoritmo desenvolvido para produzir uma solução inicial para o modelo matemático incorporado ao método de programação.

O Capítulo 6 apresenta o sistema computacional com seus principais módulos - Módulo de Suporte, Módulo de Otimização e Módulo de Planejamento. Inclui também o Manual de Utilização do Sistema, apresentado através da simulação de um projeto-exemplo.

O Capítulo 7 trata da validação do sistema como ferramenta de programação de projetos lineares. São discutidos os critérios utilizados para se realizar as simulações com o sistema e, em seguida, são apresentados e analisados todos os resultados dos projetos simulados.

Finalmente, no Capítulo 8 é apresentada a conclusão do trabalho, fazendo-se uma análise geral do método de programação desenvolvido. Em função de sua utilidade prática e das limitações operacionais observadas, são propostos vários temas para futuras pesquisas nesta área.

CAPÍTULO 2

O GERENCIAMENTO NA CONSTRUÇÃO CIVIL

Neste capítulo o assunto é a Indústria da Construção Civil. São apontadas as características que diferenciam este setor dos demais setores industriais, apresentando a estrutura organizacional de suas unidades produtivas e as principais mudanças observadas neste contexto em época de crise. Também, são apresentadas e discutidas as principais técnicas e ferramentas de planejamento e programação de projetos. Por fim, discute-se o conceito da Linha de Balanço e sua melhor adequação ao processo produtivo da construção.

2.1- Introdução

A Construção Civil, em função de sua estrutura, atua como freio ou acelerador do sistema econômico-social no qual se insere. Apesar desta constatação, a maioria dos países dispensa limitada atenção a esse importante setor da Economia. As relações com outros setores não são analisadas profundamente, impedindo o estabelecimento de metas precisas a serem alcançadas (Mascaró,1981).

Pode-se dizer ainda que, a Construção é o único ramo da produção urbana claramente diferenciado do resto da produção industrial. Tem características e variáveis difíceis de serem individualizadas e avaliadas corretamente. Fato decorrente de sua enorme dispersão geográfica e do longo ciclo de produção.

Para Mascaró (1981), a estrutura do setor faz com que seu estudo seja complexo, tanto pelos tipos de problemas que estabelece como pela indiscutível necessidade de teorias que lhe sejam próprias. Teorias que permitam encarar com eficácia sua análise e planificação.

A nível micro, observa-se que a empresa construtora normalmente possui estrutura bastante complexa, em consequência de seu porte, diversificação de atividades, territórios e formas contratuais. O planejamento e a estruturação dessas organizações envolvem análises sobre objetivos a médio e longo prazos, aspectos econômico-financeiros e tendências macro-econômicas (Balarine,1990).

Hoje, a realidade que se apresenta aos atores envolvidos nessa atividade industrial é bastante difícil em todo o mundo: o mercado contraiu-se, as margens de lucro diminuíram, faltam recursos e a concorrência aumentou (Cardoso, 1993). E mais, embalados pelo ***boom*** da variável qualidade, os consumidores de unidades habitacionais tornaram-se mais exigentes.

No Brasil, as mudanças, guardadas as diferenças culturais e estruturais da economia, foram sentidas com igual intensidade. Sob o ponto de vista social, observa-se por exemplo uma crise de competência operária. Este fato constitui um aspecto do desenvolvimento recente da atividade de construção que, num contexto de crise do setor, ganhou grande relevância (Farah, 1993).

A segunda contradição presente, que a crise evidencia, diz respeito à introdução de inovações no processo produtivo, sejam com novos materiais e componentes, sejam novos equipamentos e sistemas construtivos (Farah, 1993).

No quadro econômico-mercadológico podem ser identificadas mudanças na estratégia de gestão do processo construtivo. Algumas empresas procuram superar as contradições apontadas, orientando-se para o incremento da produtividade, a redução dos custos e a garantia da qualidade. A ênfase nesse caso, recai sobre o projeto, sobre o planejamento da execução e sobretudo sobre a articulação entre as diversas etapas do empreendimento.

Na área de Gerenciamento observa-se que a evolução deu-se de forma dispersa e as técnicas evoluíram em razão do processo experimento-erro, tanto nas empresas como na área acadêmica.

Para Rocha Lima Junior (1987), a falta de disciplina é a principal responsável pela evolução desnorteada, não havendo a acumulação de conceitos que possa mostrar a solidificação das técnicas ou caracterizar patamares no processo evolutivo.

Na área de Programação, seja operacional ou econômico-financeira, os trabalhos acadêmicos, na sua maioria, são montados numa base de dados difícil de ser manuseada na prática ou até imprópria de ser operada (Rocha Lima Junior,1987).

Todo esse quadro acaba por exigir, entre outras coisas, o desenvolvimento de ferramentas mais adequadas de planejamento, programação e controle da atividade construtiva. Ratificando esta colocação Balarine (1990) nos diz que "a autêntica constelação de variáveis contidas no negócio produzem número tão grande de inter-relações, que ultrapassam a capacidade empírica de administrar " .

2.2- Técnicas e Ferramentas de Planejamento e Programação de Obras

Existem, basicamente, três tipos de ferramentas de decisão para assistir o planejamento e programação de projetos construtivos (Warszawski, 1985):

a) os ***modelos analógicos*** tais como: gráficos, quadros, diagramas, redes e linhas de fluxo. Estes modelos representam, de uma maneira gráfica, os principais atributos do projeto e seus efeitos na programação e no orçamento;

b) os ***modelos matemáticos*** são aqueles que, através da manipulação de relações entre atributos do projeto, indicam soluções de alocação ótimas em termos de custo ou lucro. Tais modelos podem ser usados para locação, licitação, estimativas, análise de investimentos e outros tipos de decisões gerenciais;

c) o terceiro grupo de ferramentas de decisão é conhecido como ***sistemas especialistas***. Eles podem manipular dados normativos como preços, produtividade, dimensão e capacidade de equipamentos. Usam algoritmos matemáticos para solução e também empregam regras de decisão não estruturadas, baseadas em opiniões de especialistas.

Neste item serão apresentados e discutidos os modelos analógicos mais conhecidos tais como: Gráfico de Gantt, Curvas de Agregação de Recursos e Redes PERT/CPM. Eles são utilizados de forma direta - para planejar e programar projetos construtivos - ou indireta - como suporte teórico para os modelos matemáticos e sistemas especialistas.

O Gráfico de Gantt ou Diagrama de Barras continua a ser a ferramenta de programação e controle mais usada na Construção Civil. Foi desenvolvida por Henry L. Gantt no início do século para representar o programa de utilização das máquinas de uma fábrica (Pilcher, 1976).

Na programação de projetos construtivos, o Diagrama de Barras representa graficamente as datas de início, término e durações das atividades por meio de uma barra ou linha horizontal. Eles apresentam algumas vantagens sobre outras ferramentas de programação: sua forma gráfica simples resulta na compreensão relativamente fácil e requerem menos revisão e atualização por trabalharem com informações mais agregadas.

Essa última característica é especialmente útil nos estágios iniciais de um projeto de engenharia, quando ocorrem frequentes modificações e revisões. Mas o uso dos diagramas tem uma série de limitações gerais. Por exemplo, fica difícil representar as interconexões e restrições lógicas entre as atividades do projeto. Isso impossibilita visualizar os efeitos que

eventuais mudanças em uma atividade podem provocar em outras ou no programa do projeto. No entanto, um entendimento de suas limitações é importante para uma utilização apropriada e eficaz.

As Curvas de Agregação de Recursos, Curvas de Progresso ou Curvas-S, graficamente, plotam medidas de progresso no eixo vertical contra o tempo no eixo horizontal. A agregação de recursos, ou simplesmente progresso, pode ser medida em termos de dinheiro desembolsado, horas de mão de obra ou qualquer medida que faça sentido (Heineck,1989).

As medidas da Curva de Agregação podem ser expressas em unidades reais ou em percentual da quantidade total estimada. O valor, por sua vez, pode representar o custo dos fatores de produção ou o preço que por eles está sendo cobrado.

A nível macro-econômico, as Curvas de Agregação podem ser usadas no equacionamento de programas de construção, pelas autoridades governamentais ou pelos órgãos de classe. A nível micro-econômico são úteis tanto no canteiro de obras quanto na compreensão de todas as fases de um empreendimento - projeto, contratação, execução e comercialização (Heineck,1989).

Nota-se que as Curvas de Agregação e os Diagramas de Barras têm algo em comum - ambos são bastante úteis nos estágios iniciais de planejamento e programação, em que informações de caráter global são necessárias.

Essas ferramentas também são úteis como meio de comunicação. No entanto, para efeito de controle físico e financeiro do projeto, deixam muito a desejar. Primeiramente por não conterem informações de custo para intervalos de tempo menores e, principalmente, por não terem meios de obter e representar o progresso físico em termos de progresso financeiro.

As Redes de Precedência como o Método do Caminho Crítico CPM (1957) e PERT (1958) adicionam outra dimensão ao planejamento, programação e controle. As atividades são representadas graficamente em uma rede ou grafo, indicando as dependências entre elas.

As dependências na rede e as durações estimadas das atividades são usadas para calcular o tempo de conclusão requerido para o projeto, o caminho de atividades críticas e as folgas no programa.

Segundo Battersby (1971), a principal contribuição das Redes de Precedência é a possibilidade de separar a concepção lógica da obra da atribuição de durações das atividades.

No entanto, os métodos de análise de redes, originalmente desenvolvidos como uma ferramenta administrativa para a coordenação e controle de projetos complexos, são incapazes de fornecer um programa de construção prático para o gerenciamento da obra.

A aplicação prática das redes na Construção encontra uma série de dificuldades citadas por Heineck (1984): incapacidade de definir com propriedade o que são atividades numa rede; dependência de critérios subjetivos para a determinação da duração das atividades; inexistência de modelos para predizerem a produtividade nas diversas situações de obra; impraticabilidade no estabelecimento de curvas de tempo x custo reais, colocando por terra um dos aspectos mais potentes da técnica, entre outros.

Os resultados insatisfatórios, obtidos na tentativa de se utilizar as Redes de Precedência na Construção, denunciam a incompatibilidade dessa técnica com a essência do processo construtivo.

No caso de projetos lineares a incompatilidade da técnica fica mais evidente, pois neste tipo de projeto o número de atividades é desproporcional à complexidade do projeto e as atividades não têm claras dependências que definam o progresso requerido para o projeto como um todo (Carr,Meyer,1974).

Essas considerações levam-nos a concluir que na Construção as decisões cruciais de planejamento não são as durações precisas de atividades específicas, suas folgas, recursos alocados a tarefas individuais ou a probabilidade de uma tarefa terminar como planejado. O mais importante é determinar a sequência do objeto a ser construído em locações de trabalho mais adequadas; a sequência em que todas as equipes de trabalho passarão por essas locações; o dimensionamento adequado do binômio tamanho da equipe x volume de trabalho e, finalmente, a precisão na mensuração do tempo total de construção (Birrel,1980).

2.3- A Técnica da Linha de Balanço

Neste item é apresentada a origem e os princípios dessa ferramenta de planejamento e programação. Comenta-se sobre a utilização inicial da técnica e sua posterior aplicação no planejamento de projetos de Construção. São discutidas também as vantagens e limitações de seu uso para o planejamento e programação do processo construtivo.

2.3.1- O Conceito da Linha de Balanço

A Técnica da Linha de Balanço consiste de uma família de ferramentas gráficas e/ou analíticas de planejamento e programação de projetos (Figura 2.1), incluindo segundo Lutz, Hijazi (1993), a ***line of balance scheduling*** (LOB) de Khisty (1970); o ***velocity diagrams*** de Roech (1972); o ***Vertical Production Method*** (VPM) de O'Brien (1975); o ***Linear Scheduling Method*** (LSM) de Johnston (1981); o ***Time Space Scheduling Method*** (TSSM) de Stradal, Cacha (1982) e o ***Repetitive Project Model*** (RPM) de Reda (1990).

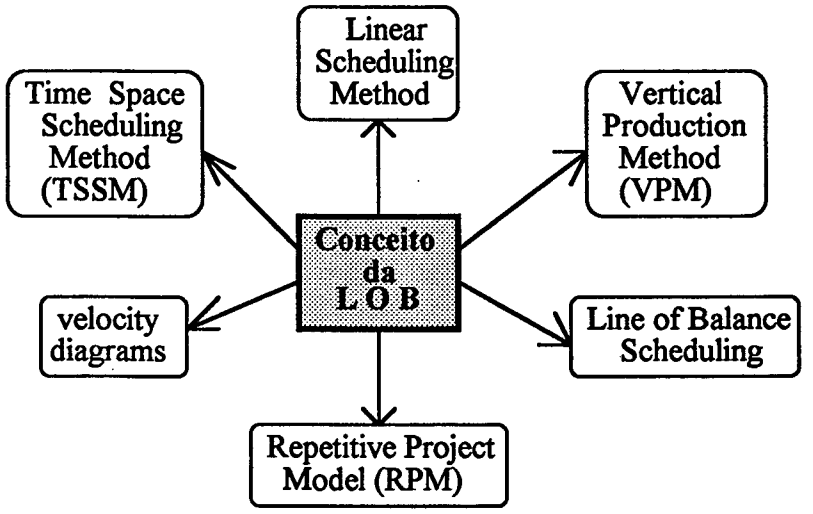


Fig.2.1- Técnicas de Planejamento baseadas no Conceito da Linha de Balanço
* Adaptado de Lutz, Hijazi (1993)

A técnica teve origem na Goodyear Company nos anos 40. Foi desenvolvida pelo U.S.Navy durante a 2ª Guerra Mundial (NAVMAT apud Turban,1962) e aprovada em sua forma presente em 1962 (NAVEXOS apud O'Brien,1962).

A aplicação inicial da técnica foi na indústria manufatureira para controle da produção, com o objetivo de encontrar uma razão de produção na linha de fluxo de produtos acabados. Sua utilização foi expandida para Construção Civil pela identificação das características de projetos repetitivos com as linhas de produção de uma fábrica.

Por causa da imensa popularidade das técnicas de programação baseadas em redes - CPM e PERT principalmente - a Linha de Balanço nunca foi totalmente desenvolvida e implementada na Indústria de Construção Americana. No entanto, este método tem sido bastante utilizado pelos empreiteiros europeus (Lutz, Hijazi, 1993).

A técnica consiste basicamente de:

1) um ***programa de execução*** para a conclusão de uma unidade repetitiva, mostrando as durações e as datas de início e término das atividades. Este programa pode ser produzido por qualquer técnica de planejamento e programação conhecida (Figura 2.2);

2) um ***gráfico***, mostrando o progresso cumulativo de cada atividade da rede e o ritmo de construção, ou seja, o ritmo de conclusão das unidades repetitivas (Figura 2.3).

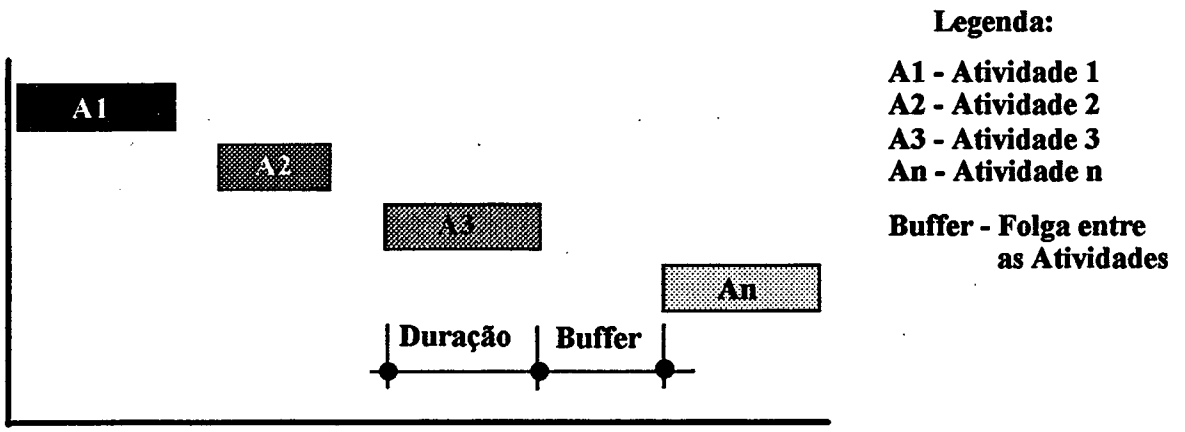


Fig. 2.2 - A Rede Unitária

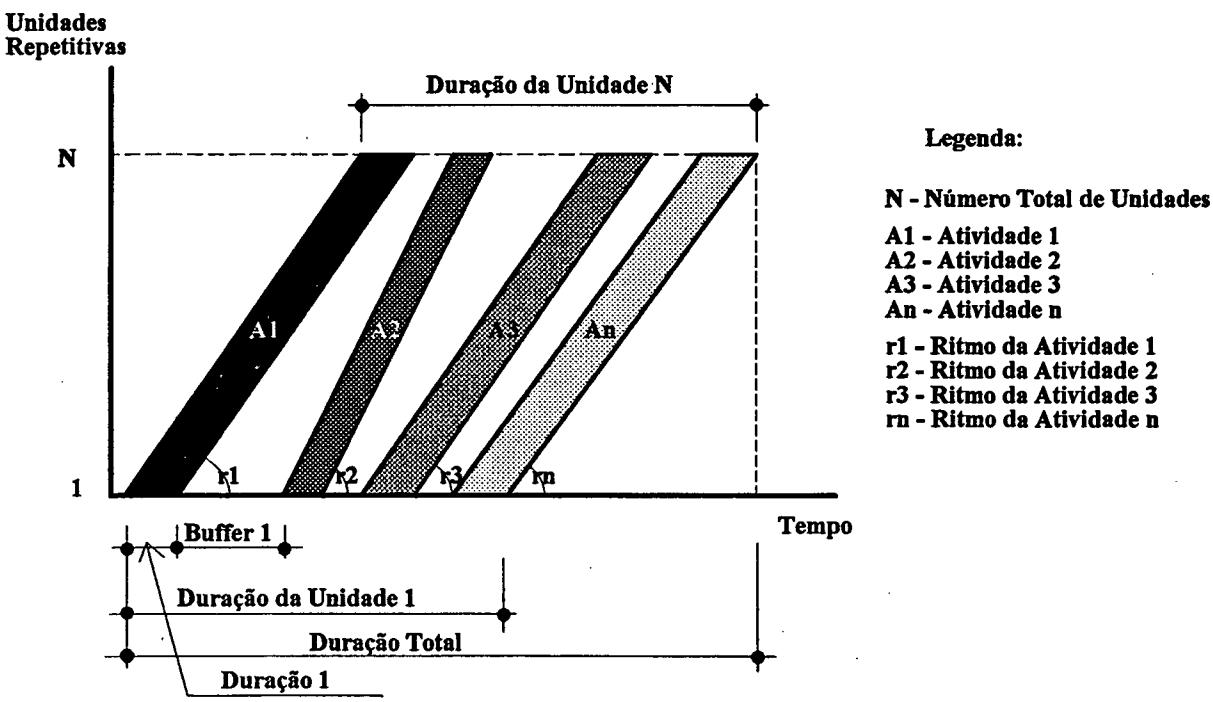


Fig. 2.3 - O Gráfico da Linha de Balanço

O método consiste em determinar os recursos necessários para cada atividade, de modo que as atividades seguintes não sofram interferências e que uma razão de construção das unidades seja mantida. Essas características facilitam o controle do processo e tiram proveito da repetitividade do trabalho (Maziero,1990).

O ritmo ou razão de construção é obtido em função do número de unidades construtivas, da duração total do projeto e da duração de cada unidade (Figura 2.4).

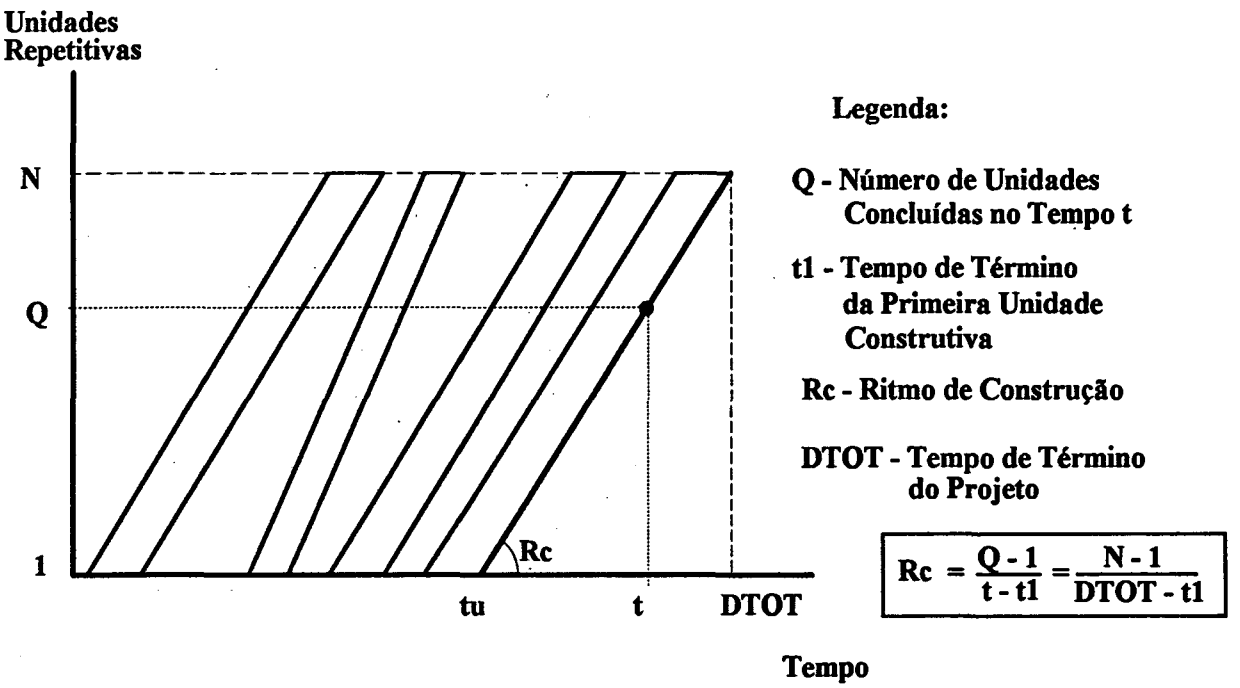


Fig.2.4 - Ritmo de Construção

Existem dois métodos de programação pela Linha de Balanço: a Programação Paralela e a Programação por Recursos. O que os diferencia é o prazo de conclusão da obra e a determinação do ritmo de construção.

O principio da Programação Paralela é a determinação de um ritmo único para todas as atividades que coincide com o ritmo de construção do projeto (Figura 2.5).

Na Programação por Recursos, as atividades seguem seus ritmos naturais, obtidos em função da duração de cada atividade e o ritmo de conclusão das unidades é determinado pelo ritmo da última atividade (Figura 2.6).

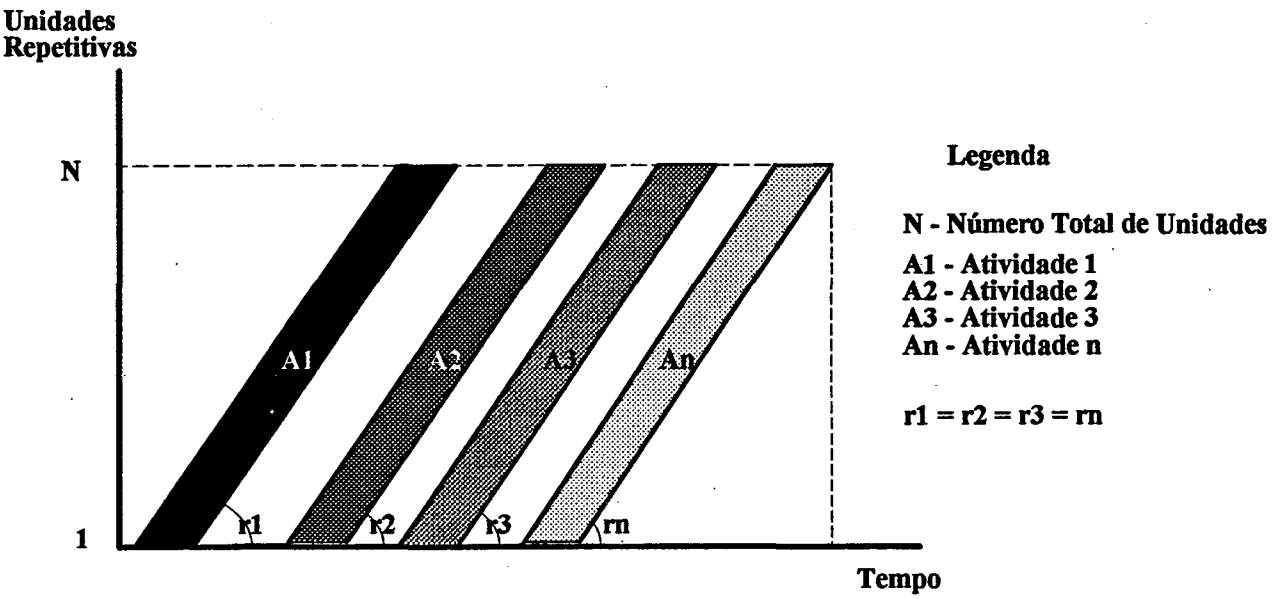


Fig. 2.5 - Programação Paralela

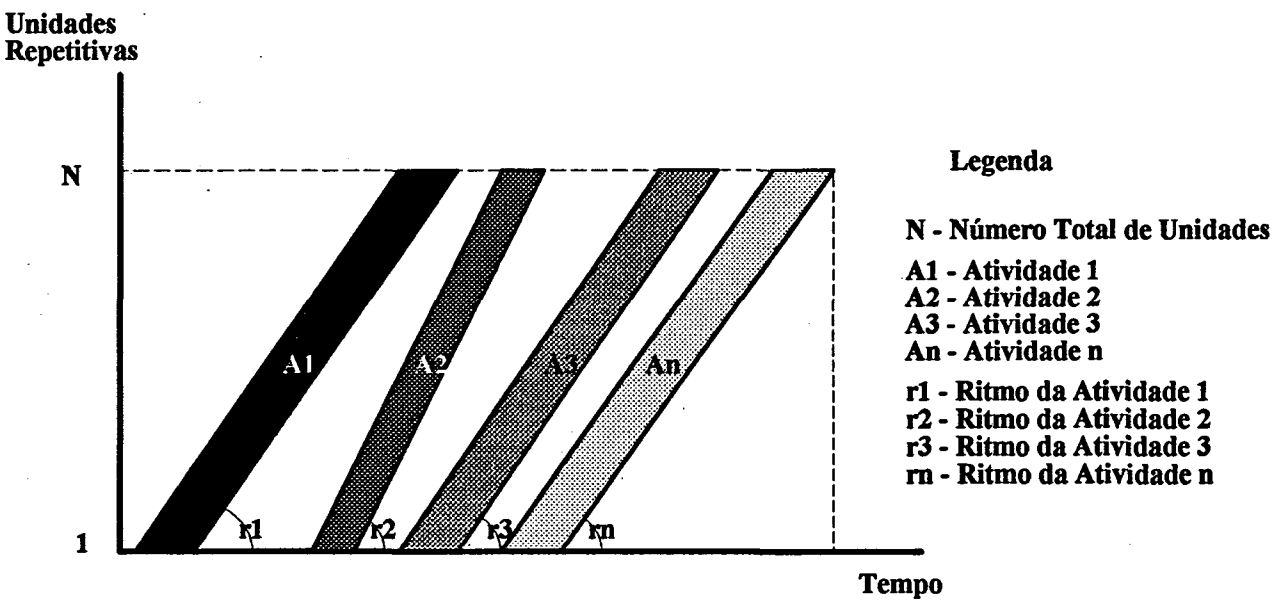


Fig. 2.6 - Programação por Recursos

É fácil verificar que a Programação Paralela possui maior simplicidade de utilização e de entendimento, facilitando o processamento de informações de controle. Ela permite um tempo total de execução menor, porém com acréscimo de custos em relação à Programação por Recursos. Isso acontece porque o ritmo de construção é diferente dos ritmos naturais das atividades, provocando o aparecimento do fenômeno denominado ***ociosidade forçada*** (Scomazzon, 1985).

2.3.2- Análise da Linha de Balanço como Ferramenta de Programação

Apresentados os conceitos básicos da Linha de Balanço, discute-se a seguir suas vantagens e desvantagens, enfatizando sua melhor aderência ao processo produtivo da Construção.

A Técnica da Linha de Balanço é uma ferramenta de programação aplicável a empreendimentos onde possam ser identificados elementos repetitivos (Scomazzon, 1985). Busca o aumento da eficiência através da exploração, dentre outros, dos seguintes fatores:

1) efeito aprendizagem;
2) efeito continuidade;
3) redução da rotatividade e ociosidade;
4) racionalização do uso de equipamentos.

O efeito aprendizagem consiste no aumento de produtividade que a repetição, o treinamento, e a experiência na execução de uma determinada tarefa proporcionam. Está intimamente relacionado à continuidade na execução da tarefa.

A obra programada com a Linha de Balanço caracteriza-se pela redução do número de operários por atividade. Cada equipe desloca-se de um elemento repetitivo para outro para a execução de sua tarefa em um ritmo pré-determinado. Assim, aumenta-se o tempo de permanência de cada operário na obra e o número de repetições que este executará. Além disso, o menor número de profissionais por atividade implica em menores investimentos em equipamentos.

Em contrapartida, a técnica impõe implicitamente a rigidez da idéia taylorista, aumentando a necessidade de gerência em função do grande número de equipes no canteiro. Também induz a um controle de qualidade por produto, dificultando a homogeneização da produção e qualidade das unidades construtivas.

No entanto, apesar das desvantagens acima mencionadas, a Técnica da Linha de Balanço permite programar, com maior fidelidade, o processo produtivo da Construção Civil, principalmente em se tratando de projetos com características repetitivas.

Primeiramente, por ser um modelo analógico, é evidente a utilidade da Linha de Balanço como meio de comunicação. Representa muitas informações necessárias de forma simples, clara e rápida e com melhor qualidade.

Além disso, permite diminuir o capital imobilizado ao longo da obra e também a necessidade de equipamentos e estoques, explorando o sequenciamento das atividades e o efeito aprendizagem. Gerencialmente, possibilita o controle da obra em pacotes de trabalho - conceito preconizado por Birrel (1980) - , motivando a produtividade através do ritmo de construção.

Pode-se concluir, portanto, que a Técnica da Linha de Balanço permite, dentro do processo de tomada de decisão da tarefa de programação de projetos, a incorporação de aspectos econômicos, financeiros e gerenciais, com um nível de agregação das informações compatível com o nível hierárquico de decisão. À medida que proporciona a representação desses fatores nos programas de execução que produz, a Linha de Balanço constitui-se numa poderosa ferramenta de planejamento e programação de obras.

CAPÍTULO 3

MODELOS DE OTIMIZAÇÃO PARA O PLANEJAMENTO E PROGRAMAÇÃO DE OBRAS

Neste capítulo são apresentados e discutidos os principais modelos de otimização produzidos para o ambiente de planejamento e programação de projetos. Classificam-se os modelos segundo o aspecto - econômico, financeiro ou operacional - que cada um procura otimizar, procurando enfatizar a fidelidade ou não dos modelos na representação do processo produtivo da Construção. Defende-se também a utilização, com critérios, dos métodos científicos para auxiliar na tarefa de programação de projetos.

3.1- A Tarefa de Programação de Obras

O desenvolvimento de procedimentos formais de tomada de decisão na administração da construção tem despertado o interesse de pesquisadores e práticos, mesmo antes desde campo de conhecido tornar-se distinto e reconhecido como uma disciplina de estudo acadêmico (Warszawski, 1985).

A primeira produção de ferramentas de decisão foram modelos analógicos simples - gráficos, quadros, diagramas - que através da representação clara de características relevantes de um projeto, ajudavam o decisor em suas tarefas de planejamento e controle.

Versões mais sofisticadas dessas ferramentas - redes de precedência, linhas de fluxo, etc. - permitiram a manipulação de atributos-chaves do sistema, revelando sua influência na solução do problema.

Um passo adicional foi a introdução de modelos matemáticos que, através da representação rigorosa das principais relações entre os parâmetros do sistema e a aplicação de técnicas matemáticas sofisticadas, poder-se-ia esperar a produção de uma solução ótima para o problema sob consideração.

Obviamente, tais modelos poderiam ser efetivamente usados somente quando as relações pertinentes ao sistema em consideração pudessem ser quantificadas e todos os dados, necessários para esta operação, estivessem disponíveis.

Acontece que nem sempre isso é possível, principalmente quando se trata de problemas de decisão não estruturados, comuns ao nível de decisão estratégico. Então a atenção das ciências da administração voltou-se para ferramentas menos formalizadas.

Essas ferramentas, conhecidas como Sistemas Especialistas, poderiam refletir modelos de decisão humana de uma forma melhor e também utilizar de maneira mais ordenada e consistente todos os dados disponíveis - quantitativos e não quantitativos - relevantes para a solução do problema.

Existem inúmeros exemplos da utilização e desenvolvimento de procedimentos formais de tomada de decisão na administração da atividade de construção, desde modelos analógicos até ferramentas mais sofisticadas como os sistemas especialistas. No entanto, serão discutidos apenas os modelos matemáticos de otimização por apresentarem afinidade com o tema desta pesquisa.

3.2- Modelos de Otimização para Programação de Projetos Lineares

Os modelos matemáticos mais comuns desenvolvidos para a Indústria da Construção Civil são: modelos de programação, modelos de locação, modelos de licitação, modelos de substituição de equipamentos, modelos de seleção de investimentos e modelos de simulacão de sistemas, entre outros (Warszawski, 1985).

Os modelos de otimização para planejamento e programação de projetos consistem, basicamente, de restrições para as datas das atividades - geradas pela técnica de planejamento utilizada, restrições de recursos físicos e financeiros, restrições de prazos e uma função objetivo para comparar os planos alternativos.

A função objetivo, que representa a medida de qualidade do plano ou programa produzido, normalmente é representada por:

a) minimização do ***custo*** direto ou custo total;
b) minimização do ***tempo*** total de execução do projeto;
c) minimização do custo de ***aluguel*** de equipamentos;

d) maximização do valor presente ou *taxa interna de retorno* do projeto;

Observa-se, no entanto, que os modelos de otimização são baseados, na sua maioria, em técnicas de programação não condizentes com o processo produtivo da Construção. Além disso, eles enfatizem a otimização de aspectos econômicos ou operacionais, calcados em informações pouco confiáveis ou obtidas teoricamente sem comprovação na prática da atividade.

Dentre os modelos de otimização pesquisados destacam-se os modelos de Perera (1983), Claure (1986) e Reda (1990). Esses modelos utilizam a Linha de Balanço - ou variações dela - como técnica de planejamento e programação, possibilitando a aplicação em projetos que apresentem características repetitivas. Discute-se cada um deles a seguir.

Perera (1983), desenvolveu um metódo de alocação de recursos baseado na quantidade de recursos-hora empregados. O método usa um modelo de programação linear, construído para determinar o ritmo máximo de construção e os requerimentos de recursos para todas atividades, considerando as limitações de recursos e o balanceamento de equipes. Também analisa se é mais econômico utilizar recursos adicionais ou horas-extras.

Conceitualmente é um bom modelo. Utiliza a Linha de Balanço como pano de fundo para programar as atividades, porém não supera as deficiências apresentadas por outros modelos, ou seja, confia na invariabilidade dos dados de produtividade e duração das atividades.

No modelo de Reda (1990), as variáveis de programação da Linha de Balanço aparecem mais explicitamente. Suas restrições garantem um ritmo de produção constante para cada equipe em cada atividade e também a continuidade do trabalho de uma unidade para outra.

O modelo permite ainda a incorporação na programação de ***buffers*** horizontais - entre atividades adjacentes na mesma unidade repetitiva - e ***buffers*** verticais - entre atividades adjacentes para um dado período de tempo. No entanto, apesar da qualidade da formulação, este modelo apresenta um erro crucial: incorpora as curvas teóricas de tempo x custo, na qual se baseia para otimizar o custo direto do projeto.

Claure (1986), estrutura seu modelo sobre o Método Espaço x Tempo que, conceitualmente, é a própria Linha de Balanço. As variáveis de decisão, no entanto, representam o programa de execução de maneira mais detalhada , permitindo maior liberdade na programação das atividades de uma unidade repetitiva para outra. Porém, mais uma vez, o

modelo *peca* pela incorporação, no processo de decisão, de dados e procedimentos com validade apenas no ambiente acadêmico.

O modelo de otimização proposto neste trabalho pretende preencher a lacuna deixada pelos modelos anteriores, assistindo o programador nos estágios preliminares de planejamento do projeto. À medida que é estruturado de forma mais agregada, o modelo diminui a necessidade de se utilizar dados e informações de caráter operacional, não existentes ou de origem e confiabilidade muito duvidosas no ambiente da Construção Civil.

Uma outra qualidade do modelo é a incorporação de aspectos externos ao canteiro de obras, representados matematicamente pela tentativa de compatibilizar as curvas de agregação de recursos necessários e disponíveis, ou melhor, de minimizar os desvios entre as curvas de agregação.

3.3- A Utilidade Prática dos Modelos de Otimização

Existem controvérsias sobre a utilidade prática dos modelos formais de pesquisa operacional na resolução de problemas reais.

Dreyfus apud Fuks (1981), aceita a visão analítica para problemas que podem ser objetivamente reconhecidos e definidos - problemas tecnológicos com restrições físicas claras e função objetivo bem especificada.

No domínio da intuição ele coloca os problemas não estruturados, como expansão e diversificação de projetos e estruturas organizacionais, onde existe enorme gama de elementos subjetivos e não quantificáveis.

A característica comum, no entanto, dos problemas de planejamento e programação de obras é que seu processo de solução pode ser formalizado. Isso significa que, dadas as informações necessárias e critérios eficientes, um procedimento pode ser desenvolvido para conduzir a uma solução ótima ou, no mínimo, satisfatória.

Recorrer aos sistemas especialistas para tentar fugir das limitações apresentadas pelos modelos matemáticos pode, às vezes, induzir ao erro. Esses sistemas confiam a qualidade da resposta à base de dados fornecida pelos *experts* e também ao processo de manipulação desses dados e informações. Acontece que na Construção Civil, incorporar procedimentos é quase sempre incorporar os erros cometidos na prática.

Portanto, a questão não é a validade ou não dos métodos científicos para resolução de problemas. A questão é a adequação entre a ferramenta utilizada e o tipo de problema que se quer resolver e, ainda, a fidelidade dos modelos desenvolvidos em relação à realidade que eles pretendem representar.

CAPÍTULO 4

O MODELO PROPOSTO

Este capítulo trata exclusivamente da apresentação e discussão do modelo de otimização que dá suporte ao método de programação proposto por este trabalho de pesquisa.

4.1- Introdução

Na Construção Civil é indispensável que os custos de construção, a cada período, sejam compatíveis com recursos financeiros disponíveis para a execução do empreendimento. Num contexto de crise do setor, a programação de obras ganha importância, deixando de ser uma simples tarefa de engenharia para ser uma atividade de ordem estratégica.

O modelo de otimização desenvolvido neste trabalho tem o propósito de auxiliar o decisor na definição da melhor estratégia de execução para o projeto. O modelo incorpora condicionantes financeiras como uma forma de comparar planos alternativos. Operacionalmente, isso significa diminuir a diferença entre as curvas de agregação de recursos necessários e disponíveis, minimizando assim o montante de capital de giro necessário ao desenvolvimento da atividade produtiva da empresa.

4.2- O Modelo Matemático

Conforme mencionado anteriormente, o modelo foi construído para programar projetos com características repetitivas. Utiliza-se do conceito da Linha de Balanço e das Curvas de Agregação de Recursos para gerar um programa de execução ***ótimo*** para o projeto.

O programa ótimo é obtido da seguinte forma:

A partir do programa de execução da obra, obtém-se o custo de cada atividade em cada período, gerando a curva de agregação de recursos necessários. A curva de recursos disponíveis é conhecida a priori, em função da estratégia de administração financeira da empresa e também das condicionantes externas à mesma.

Então, os programas alternativos são comparados através de uma medida de qualidade que determinará o programa de execução ***ótimo*** para o projeto. A medida de qualidade, ou função

objetivo, nada mais é do que o somatório das diferenças entre os valores das duas curvas de agregação de recursos em cada período de tempo.

4.2.1- As Variáveis de Decisão e Dados de Entrada do Modelo

Serão apresentadas a seguir todas as ***variáveis de decisão do modelo*** formulado. Essas variáveis representam, em sua maioria, os parâmetros que definem um programa de execução para o projeto. A representação gráfica (Figura 4.7 e Figura 4.8) facilitará a compreensão do modelo.

1- **DE(i)**: Data do Evento(i). Representa a data de término da atividade((i-1)',i) na primeira unidade repetitiva(em dias úteis);
2- **DE(i')**: Data do Evento(i'). Representa a data de início da atividade(i',i+1) na primeira unidade repetitiva (em dias úteis);
3- **DT(i',i+1)**: Data de Término da atividade(i',i+1) na última unidade repetitiva (em dias úteis);
4- **DU**: Duração de cada Unidade repetitiva (em dias úteis);
5- **DTOT**: Duração TOTal do projeto (em dias úteis);
6- **RITMO**: Representa o ritmo em que cada atividade é executada. Neste modelo, todas as atividades têm o mesmo ritmo que coincide com o ritmo de conclusão das unidades repetitivas (em dias/unidade repetitiva);
7- **Y(i',i+1,t)**: Representa o custo percentual (acumulado) da atividade(i',i+1) no período (t);
8- **δ(i',i+1,t,d)**: São variáveis binárias que indicam, para cada período de construção (t-1) → (t), se a atividade(i',i+1) ainda não começou, se está sendo executada ou se já terminou;
9- **SQRDIF**: É o somatório das diferenças percentuais (elevadas ao quadrado) entre as curvas de agregação de recursos necessários e disponíveis.

A diferenciação entre os índices **i** e **i'** torna-se necessária pela possibilidade de haver folga no programa unitário entre a data de término de uma atividade e a data de início da atividade sucedente. Então, o evento que representa a data de término da atividade(i',i+1) - DE(i+1) - não necessariamente será igual ao evento que representa a data de início da atividade((i+1)',i+2) - DE((i+1)'). Assim, o índice **i'** irá variar de **1'** a **(n° de ativ.)'** e o índice **i**, de **2** a **(n° de ativ.+1)**.

Os *dados de entrada do modelo* são:

1- **N**: Número total de unidades repetitivas;

2-**NEV**: Número de EVentos(i) da rede unitária. Como a rede é linear, o número de eventos é igual ao número de atividades mais um;

3- **DI**: Data de Início da obra (em dias úteis);

4- **RITMOmáx**: É o limite máximo permitido para o ritmo de construção do projeto. O usuário fornece o valor limite em unidades/dia. Este valor posteriormente é invertido para linearizar algumas restrições do modelo;

5- **dATIV(i',i+1)**: Duração da ATIVidade(i',i+1) para uma unidade repetitiva (em dias úteis);

6- **BUFFERp(i)**: É o vetor que contém as folgas mínimas exigidas entre duas atividades consecutivas, numa mesma unidade repetitiva (em dias);

7- **Ct(i',i+1)**: Custo total da atividade(i',i+1). O custo de cada atividade é representado em percentuais do custo total do projeto;

8- **cronograma**: Número de parcelas da curva de agregação de recursos disponíveis;

9- **DATACRON(t)**: Representa o vetor de períodos da curva de agregação de recursos disponíveis (em dias úteis);

10- **PARCELA(t)**: Representa o vetor de valores percentuais (não acumulados) correspondentes a DATACRON(t).

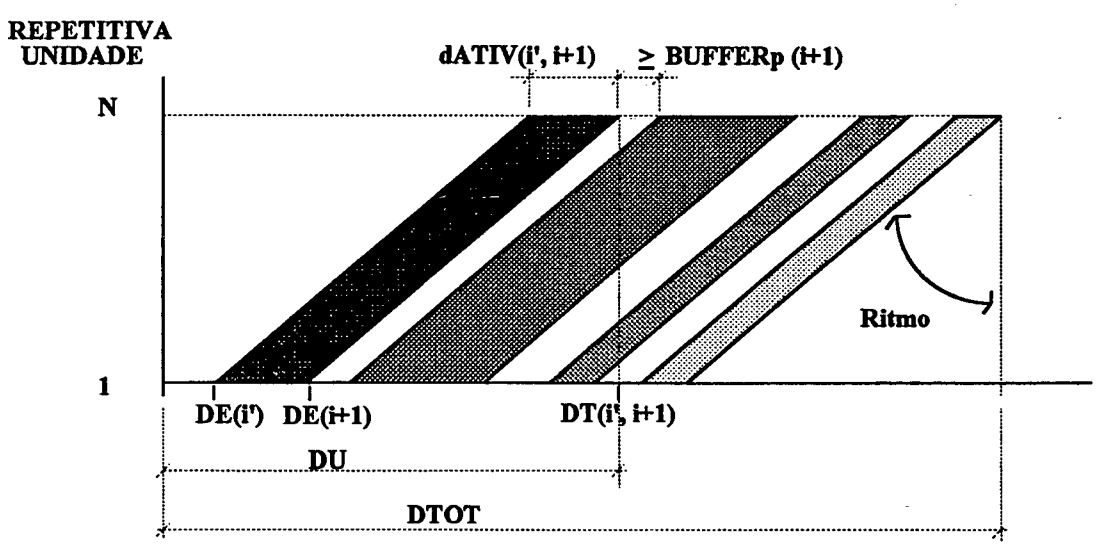


Fig. 4.1- Variáveis de Programação da Linha de Balanço

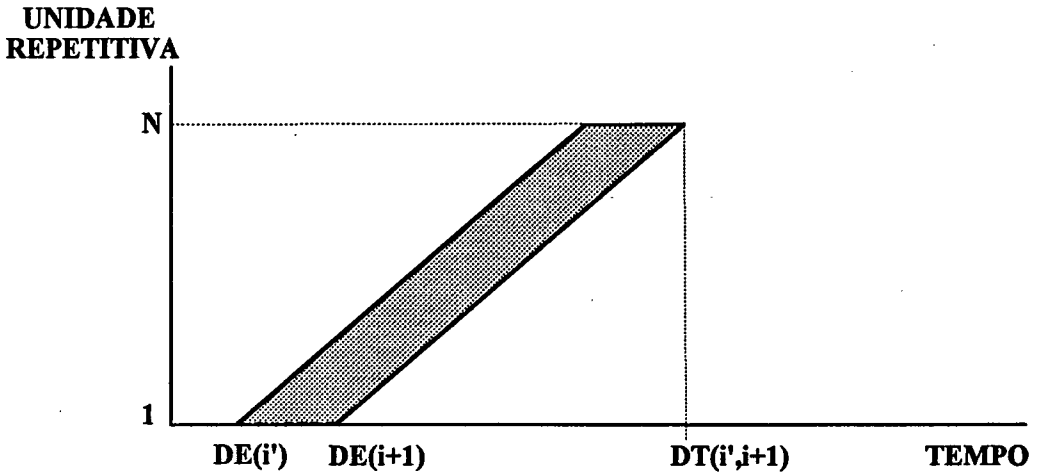


Fig. 4.2(a) - Linha de Balanço

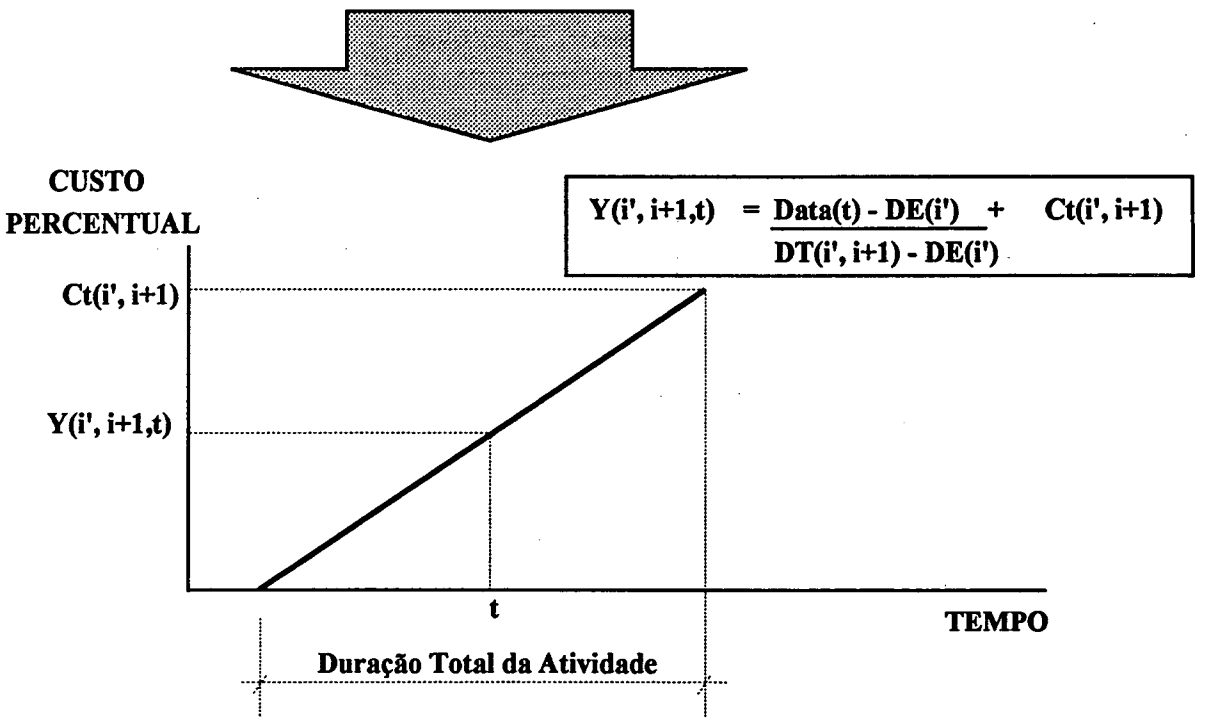


Fig. 4.2(b) - Curva Tempo x Custo Acumulado da Atividade

4.2.2- As Restrições do Modelo e a Função Objetivo

Existem quatro grandes grupos de restrições a saber:

1°) restrições que definem as relações entre as variáveis de programação da Linha de Balanço;

2°) restrições que constroem a curva de agregação de recursos, vinculada às variáveis de programação;

3°) restrições que são geradas por artifícios matemáticos. Têm origem na necessidade de se vincular o valor de certa variável de decisão à ocorrência de determinada situação do problema;

4°) restrições que definem limites inferiores e/ou superiores para as variáveis de decisão.

O primeiro grupo de restrições pode ser facilmente visualizado na Figura 4.7. As inequações do quarto grupo de restrições têm origem nas exigências do usuário ou nas relações entre as variáveis de programação. As restrições do segundo e terceiro grupos modelam a curva de agregação de recursos necessários, sendo explicadas a seguir.

Na Figura 4.8 foi definida a curva de custo de cada atividade, ou seja, Y(i',i+1,t). A equação da reta é contínua, mas a curva de custo sofre descontinuidade na data de início - DE(i') - e na data de término da atividade - DT(i',i+1). Então, podem ocorrer três situações durante o processo de busca do algoritmo de resolução do modelo:

1ª) DATACRON(t) ≤ DE(i'). Nesse caso, a curva de custo da atividade deve ser ajustada para Y(i',i+1,t) = 0;

2ª) DE(i') ≤ DATACRON(t) ≤ DT(i',i+1). Nesse caso, a atividade está em curso, não sendo necessário fazer ajustes;

3ª) DATACRON(t) ≥ DT(i',i+1). A partir desta data é necessário ajustar o valor acumulado da curva para Y(i',i+1,t) = Ct(i',i+1).

Assim, para se obter corretamente o valor do custo acumulado de cada atividade em cada período, foram criadas três variáveis binárias e a curva de custo da atividade foi ajustada para:

$$Y(i',i+1,t) = 0*\delta(i',i+1,t,1) + \delta(i',i+1,t,2)*Ct(i',i+1)*\frac{(DATACRON(t) - DE(i'))}{(DT(i',i+1) - DE(i'))} + \\ + \delta(i',i+1,t,3)*Ct(i',i+1), \forall i,t$$

Acontece que, conforme será constatado posteriormente, o modelo matemático obtido constitui um problema de programação não linear mista , não existindo, até o presente momento, nenhum algoritmo genérico que resolva satisfatoriamente problemas dessa natureza.

Então, para ***obrigar*** os valores de δ a convergirem para zero ou um, a função objetivo precisa ser penalizada em $(1 - \delta(i',i+1,t,d))*\delta(i',i+1,t,d)$, sendo introduzidas as restrições limites $\delta(i',i+1,t,d) \geq 0$, $\delta(i',i+1,t,d) \leq 1$.

Mas essas restrições ainda não são suficientes para construir corretamentes as curvas de custo das atividades. Faz-se necessário selecionar os valores de δ correspondentes a cada uma das três situações mencionadas acima.

Para a primeira situação, quando $\delta(i',i+1,t,1) = 1$, $\delta(i',i+1,t,2) = 0$ e $\delta(i',i+1,t,3) = 0$, as restrições ativas serão:
$$\begin{cases} (DATACRON(t) - DE(i'))*\delta(i',i+1,t,3) \geq 0 \text{ e} \\ (DATACRON(t) - DE(i'))*\delta(i',i+1,t,2) \geq 0. \end{cases}$$

Para a segunda situação, quando $\delta(i',i+1,t,1) = 0$, $\delta(i',i+1,t,2) = 1$ e $\delta(i',i+1,t,3) = 0$, as restrições ativas serão:
$$\begin{cases} (DATACRON(t) - DE(i'))*\delta(i',i+1,t,1) \leq 0 \text{ e} \\ (DT(i',i+1) - DATACRON(t))*\delta(i',i+1,t,3) \leq 0. \end{cases}$$

Para a terceira situação, quando $\delta(i',i+1,t,1) = 0$, $\delta(i',i+1,t,2) = 0$ e $\delta(i',i+1,t,3) = 1$, as restrições ativas serão:
$$\begin{cases} (DT(i',i+1) - DATACRON(t))*\delta(i',i+1,t,1) \geq 0 \text{ e} \\ (DT(i',i+1) - DATACRON(t))*\delta(i',i+1,t,2) \geq 0. \end{cases}$$

E finalmente, é necessário garantir que só uma das variáveis binárias esteja ativa, ou seja:

$$\sum_{d=1}^{3} \delta(i',i+1,t,d) = 1, \forall i,t$$

Na função objetivo, o artifício matemático utilizado foi elevar ao quadrado as diferenças entre os valores das curvas de agregação, para eliminar a descontinuidade que o uso do módulo provocaria e também aumentar a convergência do modelo. No item 4.2.3, a seguir, será apresentada a formulação matemática propriamente dita.

4.2.3- A Formulação

$$\text{Mín SQRDIF} = \sum_{t=1}^{\text{cronog.}} \left\{ \left[\sum_{i=1}^{(NEV-1)} [Y(i',i+1,t) - Y(i',i+1,t-1)] - PARCELA(t) \right]^2 + \right.$$

$$\left. + \sum_{i=1}^{(NEV-1)} \left[\sum_{d=1}^{3} [(1-\delta(i',i+1,t,d)) * \delta(i',i+1,t,d)] \right] \right\} \qquad (4.2.1)$$

Sujeita a

$$DE(1') \geq DI \qquad (4.2.2)$$

$$DE(i+1) = DE(i') + dATIV(i',i+1), \forall i \qquad (4.2.3)$$

$$DE((i+1)') \geq DE(i+1) + BUFFERp(i+1), \forall i \qquad (4.2.4)$$

$$DU = DE(NEV) - DE(1') \qquad (4.2.5)$$

$$DTOT = DT((NEV-1)', NEV) \qquad (4.2.6)$$

$$RITMO * (N-1) = DT(i',i+1) - DE(i') - dATIV(i',i+1), \forall i \qquad (4.2.7)$$

$$DTOT \leq DATACRON(\text{cronograma}) \qquad (4.2.8)$$

$$Y(i',i+1,t) = 0 * \delta(i',i+1,t,1) + \delta(i',i+1,t,2) * * Ct(i',i+1) * \frac{[DATACRON(t) - DE(i')]}{[DT(i',i+1) - DE(i')]} + + \delta(i',i+1,t,3) * Ct(i',i+1), \forall i,t \qquad (4.2.9)$$

$$\delta(i',i+1,t,1) + \delta(i',i+1,t,2) + \delta(i',i+1,t,3) = 1, \forall i,t \qquad (4.2.10)$$

$$[DATACRON(t) - DE(i')] * \delta(i',i+1,t,3) \geq 0, \forall i,t \qquad (4.2.11)$$

$$[DATACRON(t) - DE(i')] * \delta(i',i+1,t,2) \geq 0, \forall i,t \qquad (4.2.12)$$

$$[DT(i',i+1) - DATACRON(t)] * \delta(i',i+1,t,1) \geq 0, \forall i,t \qquad (4.2.13)$$

$$[DT(i',i+1) - DATACRON(t)] * \delta(i',i+1,t,2) \geq 0, \forall i,t \qquad (4.2.14)$$

$$[DATACRON(t) - DE(i')] * \delta(i',i+1,t,1) \leq 0, \forall i,t \qquad (4.2.15)$$

$$[DT(i',i+1) - DATACRON(t)] * \delta(i',i+1,t,3) \leq 0, \forall i,t \qquad (4.2.16)$$

$$RITMO \leq \frac{DATACRON(\text{cronograma}) - DI - \sum_{i=1}^{(NEV-1)} dATIV(i',i+1) - \sum_{i=1}^{(NEV-2)} BUFFERp(i+1)}{(N-1)} \qquad (4.2.17)$$

$$RITMO \geq 1/RITMOmáx \qquad (4.2.18)$$

$$\delta(i',i+1,t,d) \geq 0, \forall i,t,d \quad (4.2.19)$$

$$\delta(i',i+1,t,d) \leq 1, \forall i,t,d \quad (4.2.20)$$

$$\{ DE, DT, DU, DTOT, Y \} \geq 0, \forall i,t \quad (4.2.21)$$

4.3- As Limitações do Modelo

As principais limitações do modelo são sua complexidade e seu porte. Complexidade, porque se trata de um problema de programação não linear não convexo. Porte porque esse parâmetro cresce exponencialmente com o número de atividades da rede unitária.

A complexidade do modelo deve-se aos incontáveis máximos e mínimos locais que ele apresenta, tornando a qualidade da resposta altamente dependente da solução inicial - o ponto de partida do algoritmo de busca .

O porte do modelo tem origem na quantidade de variáveis e restrições que são criadas em função do número de atividades do programa unitário. Conforme foi apresentado na formulação matemática, cada atividade gera diretamente três(3) variáveis - DE(i'), DE(i), DT(i',i+1) - e [3*(n° ativ.) - 1] restrições. As restrições são provenientes das datas dos eventos de início e término das atividades e das folgas entre as atividades - inequações (4.2.3), (4.2.7) e (4.2.4) respectivamente.

Indiretamente, o número de atividades, juntamente com o número de parcelas da curva de agregação, é responsável por [4*(n° ativ.)*(n° parc.)] variáveis - Y e as três(3) variáveis δ - e [8*(n° ativ.)*(n° parc.)] restrições - inequações (4.2.9) a (4.2.16).

Para resolver o problema da escolha de um ponto inicial, foi desenvolvida uma heurística, obtendo-se resultados bastante satisfatórios, tanto em termos de tempo quanto de qualidade da solução inicial obtida. A apresentação e análise dessa heurística será assunto do próximo capítulo.

O porte do modelo não chega a ser um fator limitante, visto que a intenção do sistema é produzir um plano tático para o projeto, justificando um maior nível de agregação para o programa de execução de cada unidade repetitiva.

Existem ainda outras limitações de caráter operacional como por exemplo: ritmos iguais para todas as atividades; rede unitária linear; programação apenas de projetos de unidades

unifamiliares (casas); não inclusão de restrições de datas intermediárias e outras restrições provenientes de limitações físico-financeiras das empresas de Construção.

No entanto, essas limitações podem ser facilmente solucionadas com a ampliação e ajustes feitos no modelo proposto. Não se pode esquecer que a intenção deste trabalho é desenvolver um modelo que forneça um esqueleto, uma estrutura básica que consiga representar a realidade da tarefa de programação de projetos lineares. As particulares de cada caso podem ser incorporadas através de pequenas alterações feitas no modelo básico.

Observa-se também que o dimensionamento e quantidades de equipes de trabalho não foram incorporados à estrutura do modelo básico por se tratar de uma decisão de nível operacional, estando assim fora do propósito deste trabalho.

CAPÍTULO 5

A SOLUÇÃO INICIAL

Neste capítulo, a heurística desenvolvida para produzir a solução inicial para o modelo é apresentada e analisada. Discute-se o processo de desenvolvimento de seu algoritmo e de outras tentativas anteriores, apresentando-se também as principais qualidades e limitações do algoritmo produzido.

5.1- O Processo de Desenvolvimento da Heurística

Conforme exposto no item anterior, o modelo de otimização proposto neste trabalho apresenta muitos máximos e mínimos locais distintos. Então ele converge para a solução mais próxima do ponto onde se inicia a busca, ficando preso no *vale* que contém a solução inicial. Essa dificuldade obriga o modelador a fazer uma escolha criteriosa do ponto inicial.

Inicialmente, tentou-se uma exploração combinatória, através de um delineamento fatorial no espaço de soluções viáveis. Explicando melhor, foram definidos cinco níveis de grandeza - pequeno, pequeno-médio, médio, médio-grande, grande - para cada variável de decisão e, variando esses valores, era obtida a melhor combinação, ou seja, aquela que determinava o menor valor para a função objetivo.

No entanto, o tempo de execução do algoritmo era razoavelmente grande - em torno de cinco minutos para rodar projetos de 36 meses. Essa, porém, não foi sua maior deficiência. A qualidade dos resultados era, um tanto quanto instável, oscilando aleatoriamente entre muito boa e muito ruim.

Optou-se, então, por tentar incorporar o procedimento de um bom programador que se utiliza da Linha de Balanço como técnica de programação para projetos lineares.

0.221.353-6

5.2- Descrição do Algoritmo Proposto

Um bom programador de projetos lineares procede assim:

1°) começa escolhendo um ritmo alto para a primeira atividade, obtendo desta forma maior liberdade para programar as demais atividades;

2°) procura o posicionamento da segunda atividade para tentar ***fechar*** o custo total dos períodos adjacentes ao posicionamento escolhido. Fechar os custos significa zerar a diferença entre os valores das curvas de agregação de recursos necessários e disponíveis nos referidos períodos de tempo;

3°) posiciona as demais atividades, utilizando o mesmo critério e tentando manter o mesmo ritmo. A última atividade tem maior liberdade de posicionamento, dependendo do espaço de tempo entre a posição da penúltima atividade e o prazo limite do projeto;

4°) calcula o somatório das diferenças entre as curvas. Se não considerar o valor satisfatório, escolhe outro ritmo menor que o anterior e reinicia o processo.

A idéia de ***procurar o posicionamento de cada atividade*** inspirou o desenvolvimento da heurística descrita a seguir.

Primeiramente, são calculados quarenta ritmos (r=1,...,40), distribuídos linearmente no intervalo entre o ritmo mínimo possível e o ritmo máximo permitido pelo usuário. Em seguida, calcula-se a tolerância média para o projeto que está sendo simulado, através de um ajuste feito na tolerância padrão do algoritmo.

Para se chegar ao número de quarenta (40) ritmos, foi feita uma experimentação - depois do algoritmo pronto - com o objetivo de observar a melhoria gradativa do resultado final, em função do aumento do número de ritmos testados. Pôde-se perceber que essa melhoria era insignificante a partir de quarenta ritmos, constituindo-se esse número no limite do algoritmo proposto.

O valor da tolerância padrão foi fornecido tendo-se em mente o seguinte princípio: qual o valor máximo de descompatibilização entre as curvas de agregação de recursos é considerado aceitável para um projeto de 12 meses ? Concluiu-se que 12%, ou 1% por período, seria um valor razoável, sendo este portanto o valor da tolerância padrão do algoritmo.

A partir do valor da tolerância padrão, obtém-se a tolerância média para o projeto. Esse é um ponto que deve ser melhor explicado por ser fundamental à qualidade do resultado obtido pelo algoritmo. A tolerância média é o parâmetro que define o posicionamento de cada atividade e, conforme dito anteriormente, é obtida através de um ajuste feito no valor da tolerância padrão do algoritmo.

Esse ajuste é função do número de parcelas e da forma da curva de agregação de recursos disponíveis do referido projeto, sendo inversamente proporcional ao primeiro e diretamente proporcional à irregularidade da curva.

Explicações dadas, apresenta-se o algoritmo propriamente dito:

PASSO 1: Inicializar o índice r no valor 1;

PASSO 2: Posicionar a primeira atividade (i = 1) em DE(i') = 0 para RITMO = RITMO(r). Neste algoritmo, a atividade será apresentada como um vetor unidimensional;

PASSO 3: A partir do posicionamento da atividade i, calculam-se as datas relativas de início mais cedo e mais tarde para as demais atividades;

PASSO 4: Calcular também uma curva de desvios D(i,j), $\forall j$, proveniente da diferença entre os recursos necessários - decorrentes do posicionamento da atividade - e os recursos disponíveis - fornecidos pelo usuário. Na curva cumulativa D, o índice i representa a atividade que está sendo posicionada e j, os períodos de tempo;

PASSO 5: Comparar cada valor da curva D com a tolerância média até encontrar um desvio negativo que seja superior, em módulo, ao valor da tolerância. Determinado o período j, o provável posicionamento da atividade (i+1) será no intervalo (j-1) a j;

PASSO 6: Determinar, dentro desse intervalo, o valor de DE((i+1)') que zera o desvio D(i+1,j). Se |D(i+1,j)| não puder ser zerado, a atividade (i+1) será posicionada em (j-1), por proporcionar a maior contribução possível dessa atividade no período j.

Comparar DE com DE+cedo:

- Se DE < DE+cedo, fazer DE = DE+cedo;

PASSO 7: Calcular os valores provisórios da curva D(i+1,j);

PASSO 8: Para determinar o posicionamento definitivo da atividade (i+1) comparar os valores de $\sum_{j=DE(i+1)'}^{DT(i+1)'} |D(i,j)|$ e $\sum_{j=DE(i+1)'}^{DT(i+1)'} |D(i+1,j)|$ para o intervalo entre a data de início da atividade na primeira unidade - DE - e a data de término da última unidade construtiva - DT:

- Se $\sum_{j=DE(i+1)'}^{DT(i+1)'} |D(i+1,j)| < \sum_{j=DE(i+1)'}^{DT(i+1)'} |D(i,j)|$, então está determinado o posicionamento definitivo da atividade (i+1);
- Senão, incrementar a data de início da atividade (i+1) em 1 período e testar:
  - Se DE((i+1)') > DE+tarde((i+1)'), conclui-se que não há posição vantajosa para a atividade (i+1), programando-a na posição menos ruim, ou seja, naquela em que $\sum_{j=DE(i+1)'}^{DT(i+1)'} |D(i+1,j)|$ é o menor para todas as posições testadas;
  - Senão, voltar ao PASSO 7;

PASSO 9: Incrementar i e testar:

- Se $i \leq$ (n° ativ. - 1), voltar ao PASSO 3;
- Senão, vai para o PASSO 10;

PASSO 10: Posicionar a última atividade da rede unitária. Ao contrário das demais atividades, esta atividade tem maior liberdade para ser posicionada porque, neste algoritmo, a primeira atividade é fixada em DE(1') = 0 e as outras são programadas tão logo se consiga uma posição vantajosa para elas. A última percorrerá todo o intervalo entre DE+cedo(NEV-1) e DE+tarde(NEV-1) procurando a posição mais vantajosa;

PASSO 11: Incrementar r e testar:

- Se $r \leq 40$, voltar ao PASSO 2;

PASSO 12: Comparar os programas obtidos para cada ritmo, escolhendo o melhor - aquele que minimizar a função objetivo, representada neste algoritmo por $\sum_{j=1}^{cronog} |D(NEV-1,j)|$.

5.3- Análise da Heurística

Ótimos resultados foram obtidos em um tempo de execução muito pequeno - em torno de um (1) segundo de processamento com um PC-AT386. Obtendo-se programas de execução para 40 ritmos diferentes, garante-se uma exploração satisfatória do espaço de soluções viáveis.

Entretanto, algumas limitações podem ser constatadas na concepção do algoritmo. A primeira, e mais importante, é a grande dependência que a qualidade da solução inicial tem em relação ao valor da tolerância média do projeto, cogitando-se a necessidade de maior experimentação para definir os critérios utilizados na obtenção desse valor.

Outra limitação do algoritmo é na verdade uma simplificação operacional. Como ficaria difícil para o algoritmo reconhecer a forma da curva de recursos disponíveis apenas pelos valores de suas parcelas, o fator de ajuste da tolerância padrão (referente a este parâmetro) é obtido em função da diferença percentual entre o desvio médio da curva do projeto e o desvio médio de uma curva padrão do tipo trapezoidal clássica.

No entanto, em face dos resultados apresentados, conclui-se não ser vantajoso investir na melhoria do algoritmo, principalmente pelo nível de desenvolvimento que este trabalho propõe-se a alcançar.

CAPÍTULO 6

O SISTEMA COMPUTACIONAL

Este capítulo descreve o sistema computacional que foi produzido para incorporar o método de programação desenvolvido por este trabalho. E ainda, faz a apresentação do sistema computacional em forma de um manual de utilização, mostrando todos os caminhos que se pode percorrer dentro do sistema.

6.1- Introdução

Foi necessário desenvolver um sistema computacional, em linguagem TURBO PASCAL 6.0, para integrar as seguintes funções:

1°) permitir a entrada de dados pelo usuário; transformar os dados para utilizá-los no modelo matemático; obter a solução inicial e preparar o modelo para ser resolvido pelo GAMS 2.25 - ***Módulo de Suporte***;

2°) promover a interface com o software GAMS 2.25 - ***Módulo de Otimização***;

3°) recuperar os valores ótimos obtidos para as variáveis de decisão do modelo e transformá-los em um programa de execução para o projeto - ***Módulo de Planejamento***.

Os dados que o usuário deve fornecer são, basicamente, os mesmos apresentados como dados de entrada do modelo matemático. Exceto pelo valor do número de eventos da rede unitária (NEV) que é fixado pelo sistema em oito(8) para limitar o porte do modelo.

A saída do sistema, conforme descrito na terceira função acima é o programa de execução *ótimo* para o projeto, ou melhor, as datas de início de todas atividades da rede unitária - DE(i') - e do ritmo de construção - RITMO - que minimizam o valor da função objetivo.

6.2- A Arquitetura do Sistema

Será mostrada, através de um fluxograma (Figura 6.1), a estrutura do sistema computacional desenvolvido, apresentando seus principais módulos e respectivas subrotinas, assim como, o procedimento utilizado para administrar o acesso aos dados e resultados de projetos já simulados.

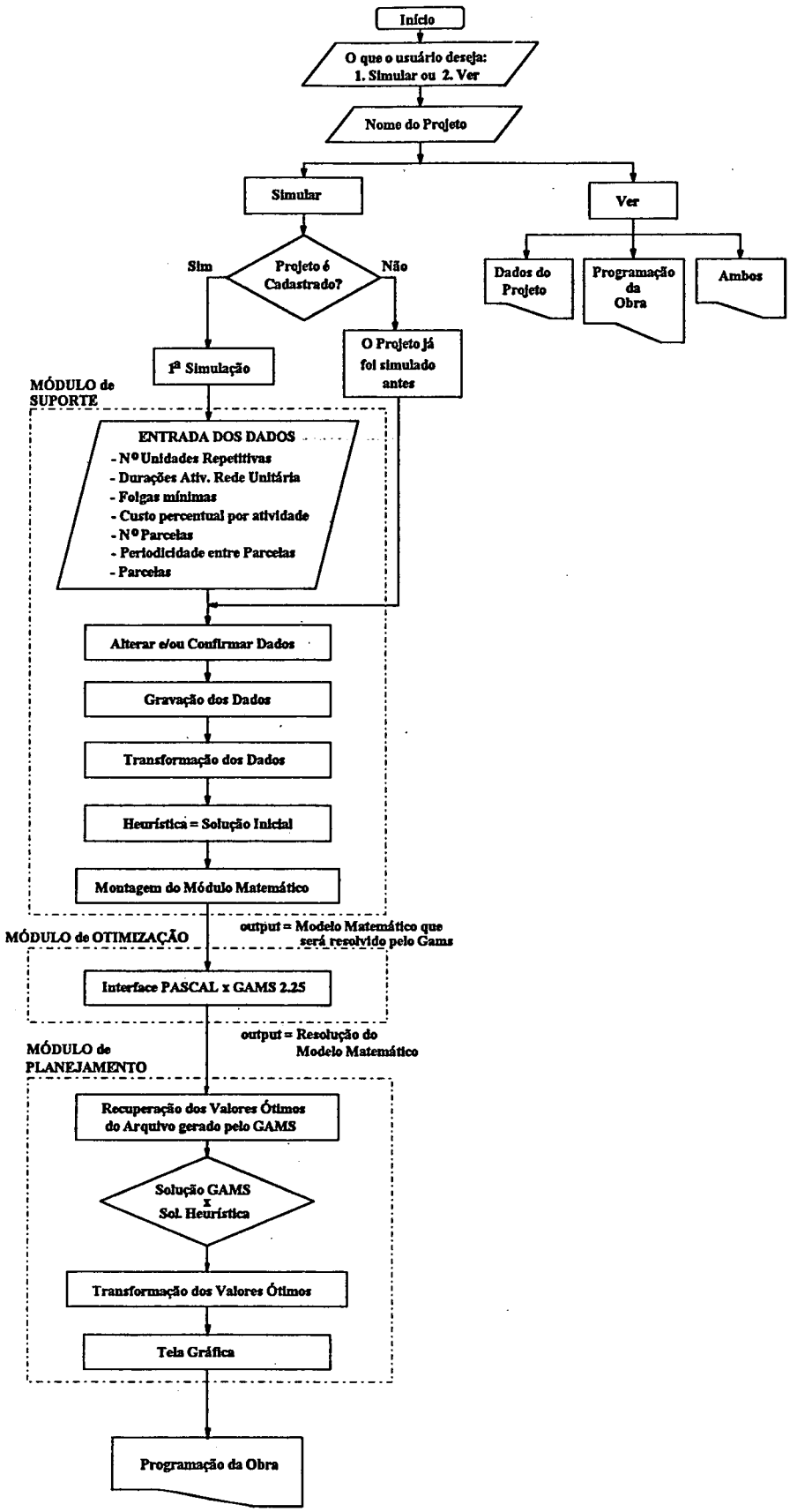


Fig.6.1 - A Arquitetura do Sistema

6.2.1- Módulo de Suporte

Operacionalmente falando, o produto final deste módulo é um arquivo contendo o modelo matemático que será executado pelo software GAMS 2.25.

O ***Módulo de Suporte*** compreende as seguintes subrotinas:

- subrotina de ***Entrada de Dados***. Como o próprio nome indica, esta subrotina permite a entrada de todos os dados necessários ao sistema, através de telas auto-explicativas;

- subrotina de ***Alteração e/ou Confirmação dos Dados***. Esta subrotina mostra ao usuário, em uma tela resumo, todos os dados do projeto para que ele possa alterar, se necessário;

- subrotina de ***Gravação dos Dados***. Grava os dados, já confirmados, em um arquivo com o nome do projeto;

- subrotina de ***Transformação de Dados***. Esta subrotina é necessária para transformar alguns valores em unidades de medida mais adequadas e também para obter dados indiretos - como por exemplo os quarenta valores de ritmo que precisam ser interpolados entre o mínimo e o máximo;

- a ***Heurística***. Esta subrotina contém o algoritmo descrito no capítulo anterior. Oseu objetivo, conforme já exposto, é obter a solução inicial para o modelo matemático;

- subrotina de ***Montagem do Modelo***. Esta subrotina monta um arquivo - com extensão GMS - contendo os dados e o modelo matemático, dentro dos moldes exigidos pelo GAMS para que o mesmo possa ser compilado e executado.

6.2.2- Módulo de Otimização

Este módulo constrói a interface entre a linguagem TURBO PASCAL 6.0 e o software GAMS 2.25, promovendo a execução automática do arquivo preparado no módulo anterior.

Abre-se um espaço neste parágrafo para falar um pouco sobre o software GAMS 2.25. O General Algebraic Modeling System, ou simplesmente GAMS, é uma ferramenta utilizada para resolver modelos matemáticos de Programação Linear, Inteira e Não Linear.

A principal vantagem deste software, quando comparado com outros similares, é sua maior habilidade e flexibilidade para montar o modelo matemático. Apesar de não conter um ambiente de edição, ele tem uma linguagem própria que possibilita a entrada literal das inequações do modelo. Esta característica é altamente desejada para modelos com grande indexação de dados e variáveis de decisão, facilitando também o desenvolvimento de interfaces com outros softwares.

Para utilizar o GAMS, basta criar um arquivo em qualquer editor - TURBO PASCAL, EDIT do software WINDOWS e similares - e depois executá-lo com uma linha de comando. O importante, no entanto, é respeitar as regras que a linguagem GAMS estabelece para definição dos dados - conjuntos, parâmetros e tabelas - , variáveis e inequações do modelo.

O GAMS tem ainda uma outra característica bastante desejável: permite, através de um arquivo de opções - o MINOS5.OPT para o caso de problemas de programação não linear - modificar os valores dos principais parâmetros utilizados pelos algoritmos de resolução dos modelos ou algoritmos de busca. Esta característica também foi bastante explorada no desenvolvimento do sistema, à medida que, através de experimentação, foram alterados os valores de alguns parâmetros para melhorar a convergência global do modelo matemático proposto.

6.2.3- Módulo de Planejamento

O ***Módulo de Planejamento*** tem a finalidade de transformar os valores ótimos do modelo matemático em um programa de execução para o projeto simulado.

As principais subrotinas deste módulo são:

- subrotina de ***Recuperação da Solução Ótima***. A função desta subrotina é recuperar os valores ótimos de um arquivo - extensão LST - gerado pela execução do GAMS;

- subrotina de ***Transformação dos Valores Ótimos***. Prepara os valores ótimos para serem apresentados ao usuário de forma mais compreensível;

- subrotina da ***Tela Gráfica***. Esta subrotina gera, a partir dos valores ótimos já transformados, uma tela que constrói o gráfico da Linha de Balanço, as curvas de agregação de recursos e da diferença, mostrando também um conjunto de informações necessárias à implementação do programa de execução do projeto.

Para administrar o sistema como um todo foram acrescentadas algumas subrotinas e procedimentos necessários à manutenção dos arquivos de cadastro, de dados e de programas de execução de projetos já simulados, permitindo o acesso do usuário a essas informações a qualquer tempo.

Existe ainda um procedimento executado pelo sistema para comparar os resultados obtidos com a execução do GAMS e a Solução Inicial obtida pela heurística.

A princípio, era de se esperar que o resultado do GAMS melhorasse ou, na pior das hipóteses, mantivesse o valor ótimo obtido inicialmente pela heurística. Isso realmente acontece, porém, o artifício matemático incluído na formulação do modelo para eliminar a descontinuidade da função objetivo - os módulos das diferenças entre os valores das curvas de agregação foram transformados em diferenças quadradas - pode provocar a ocorrência de situações indesejáveis.

O exemplo a seguir mostra um desses casos:

Tabela 6.1 - Tabela Ilustrativa

Nº	PARCELA	(1) HEURÍSTICA			(2) GAMS 2.25		
		SOMA	$\|\Delta\|$	Δ^2	SOMA	$\|\Delta\|$	Δ^2
1	0.22	0.22	0.00	0.000	0.20	0.02	0.004
2	0.18	0.18	0.00	0.000	0.20	0.02	0.004
3	0.40	0.45	0.05	0.025	0.42	0.02	0.004
4	0.10	0.10	0.00	0.000	0.12	0.02	0.004
5	0.10	0.05	0.05	0.025	0.06	0.04	0.016
TOTAL	**1.00**	**1.00**	**0.10**	**0.050**	**1.00**	**0.12**	**0.032**

Onde: $\Sigma \Delta^{2(2)}$ é melhor que $\Sigma \Delta^{2(1)}$, mas
$\Sigma |\Delta|^{(2)}$ é pior que $\Sigma |\Delta|^{(1)}$.

Feito todos os esclarecimentos necessários à compreensão do método de programação desenvolvido, será apresentado, em forma de exemplo, o manual de utilização do sistema.

6.3- O Manual de Utilização

Primeiramente, será apresentado o exemplo que será simulado, com seus respectivos dados, para que se possa compreender melhor o manual.

O projeto a ser simulado é um conjunto habitacional de duzentas (200) casas unifamiliares, distribuídas em dezessete quadras (conforme a planta de locação localizada no ANEXO I) situado no município de Porto União.

Os dados de entrada do sistema são:

a) o número de unidades repetitivas igual a duzentas (200) casas;

b) a curva de agregação de recursos disponíveis, compreendendo as datas e respectivos valores (Tabela 6.2);

c) a rede ou programa unitário, compreendendo a definição das atividades e respectivas durações e folgas para execução de uma unidade repetitiva (Tabela 6.3);

d) custos percentuais de cada atividade (Tabela 6.4);

e) ritmo de construção máximo: 6 casas/dia útil.

Observa-se que o número de unidades é definido pelas características do projeto e a curva de agregação, pela disponibilidade financeira de recursos na empresa.

Na verdade, a determinação do número de unidades torna-se mais flexível, à medida que as unidades repetitivas podem ser mais ou menos agregadas, de acordo com o interesse do programador. Por exemplo, neste projeto poder-se-ia agregar as casas em quadras, mas, devido à irregularidade na distribuição das casas, esta possibilidade foi totalmente descartada.

A curva de recursos, por sua vez, é definida em função da capacidade de pagamento do mercado, das regras de desembolso do Sistema Financeiro de Habitação e da estratégia de administração de capital de giro da empresa.

Tabela 6.2- A Curva de Agregação de Recursos Disponíveis do Projeto-Exemplo

DATA	VALOR (%)	DATA	VALOR (%)
1° Mês	0.00	7° Mês	13.00
2° Mês	2.00	8° Mês	14.00
3° Mês	5.00	9° Mês	10.00
4° Mês	11.00	10° Mês	8.00
5° Mês	13.00	11° Mês	7.00
6° Mês	14.00	12° Mês	3.00

Tabela 6.3 - O Programa Unitário do Projeto-Exemplo

DESCRIÇÃO DA ATIVIDADE	DURAÇÃO (Dias Úteis)	FOLGA (Dias Úteis)
Fundações	3.00	1.00
Estrut. / Alvenaria / Cobertura	8.50	-
Tubulações	2.00	-
Batentes e Contram. / Revest. / Pisos	4.00	-
Forro / Esquadrias / Ferrag. / Vidros	2.33	-
Pintura	2.00	-
Fiação e Aparelhos / Compl. / Limpeza	0.83	

Tabela 6.4- Os Custos Percentuais do Projeto-Exemplo

DESCRIÇÃO DA ATIVIDADE	CUSTO (%)
Fundações	3.12
Estrut. / Alvenaria / Cobertura	29.19
Tubulações	7.55
Batentes e Contram. / Revest. / Pisos	27.88
Forro / Esquadrias / Ferrag. / Vidros	6.76
Pintura	7.13
Fiação e Aparelhos / Compl. / Limpeza	18.37
TOTAL	100.00

Para que o usuário saiba que decisões deve tomar, percorrendo os caminhos por ele desejados, é apresentada a seguir a árvore de decisões do sistema (Figura 6.2).

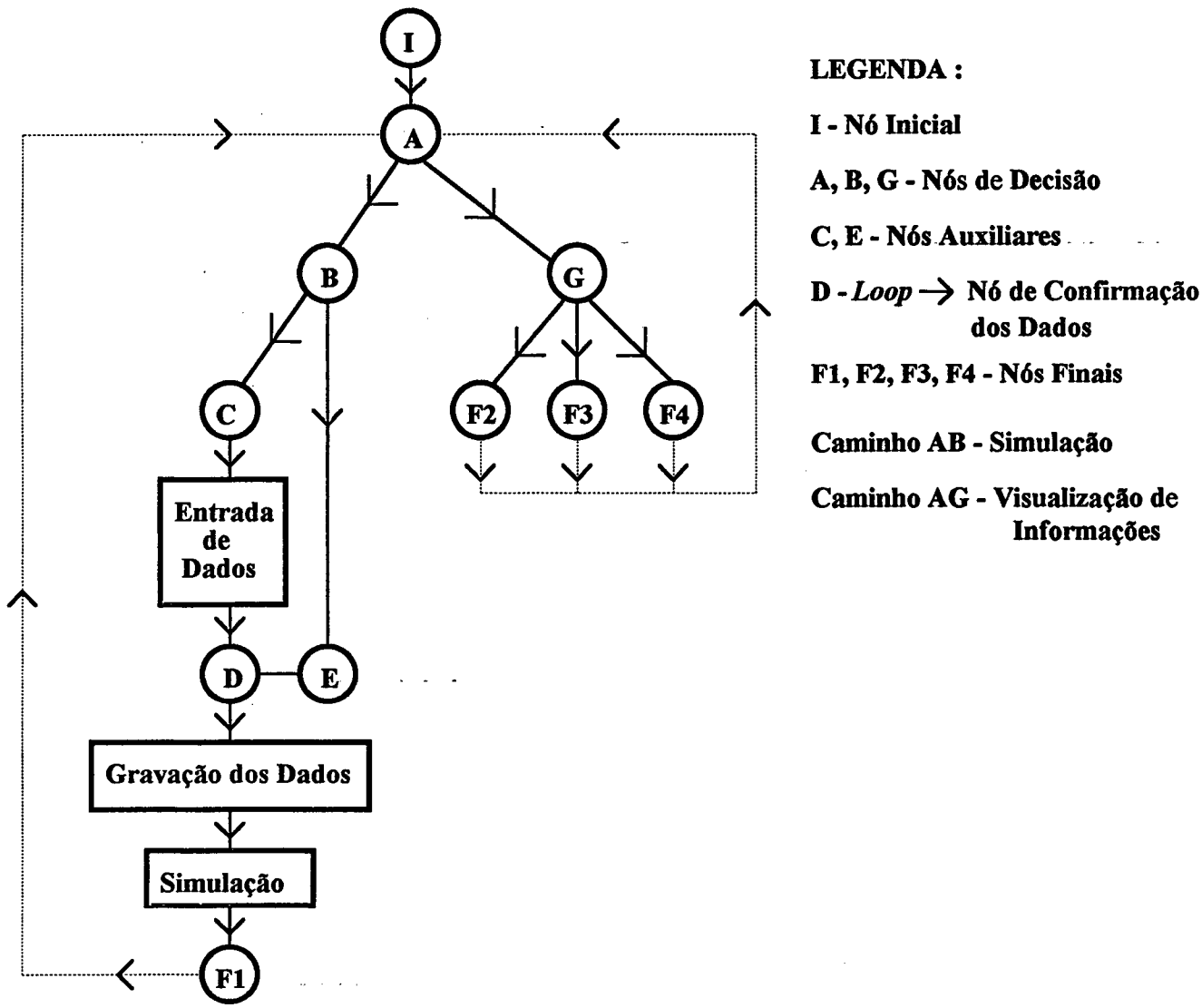


Fig.6.2 - A Árvore de Decisão do Sistema

Inicia-se, então, o manual de utilização propriamente dito:

De posse do arquivo executável do sistema e dentro do ambiente DOS do microcomputador, o usuário deve digitar SISTEMA e apertar a tecla ENTER.

Nesse momento, aparecerá a seguinte tela:

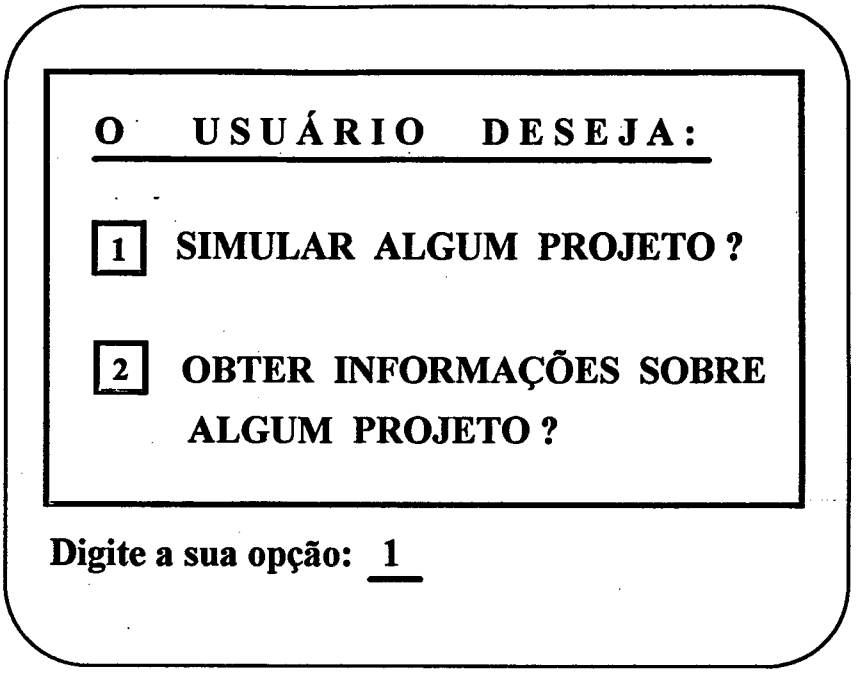


Fig.6.3- Primeira Tela do Sistema

A tela acima representa o nó A da árvore de decisões. Explorando inicialmente o caminho ABCDF1 , o usuário deve entrar com a opção[1].

Neste momento, aparecerá a segunda tela do sistema pedindo ao usuário que entre com o nome do projeto que deseja simular.

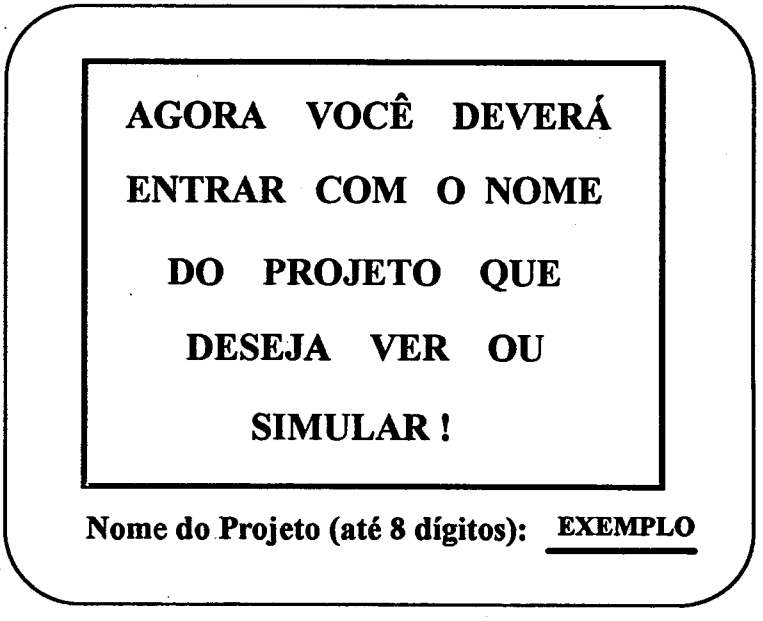


Fig.6.4 - Segunda Tela Sistema

Feito isso, é acionada uma subrotina do sistema para procurar pelo projeto EXEMPLO no arquivo de projetos cadastrados - nó B. Considerando que se trata da primeira simulação do projeto, o sistema percorrerá o caminho BC. Dá-se, então o processo de entrada de dados (Figura 6.5).

ATENÇÃO USUÁRIO! AGORA VOCÊ DEVERÁ ENTRAR COM OS SEGUINTES DADOS!!!

OBS: A cada entrada de dados, tecle ENTER

[A] O NÚMERO TOTAL DE UNIDADES A SEREM CONSTRUÍDAS: 200

[B] O RITMO DE CONSTRUÇÃO MÁXIMO PERMITIDO (Em casas/dia): 6

[C] ENTRE COM A DURAÇÃO DE CADA ATIVIDADE (Em dias):

Fundação : 3
Est/Alv/Cob : 8.5
Tubulações : 2
Revest/Pisos : 4
Esq/Vidros : 2.33
Pintura : 2
Serv.Compl. : 0.83

[D] AS PRECEDÊNCIAS TÉCNICAS MÍNIMAS (EMdias) :

Entre Fundação e Est/Alv/Cob : 1
Entre Est/Alv/Cob e Tubulações : 0
Entre Tubulações e Revest/Pisos : 0
Entre Revest/Pisos e Esq/Vidros : 0
Entre Esq/Vidros e Pintura : 0
Entre Pintura e Serv.Compl. : 0

[E] O CUSTO DE CADA ATIVIDADE (Em %) :

Fundação : 3.12
Est/Alv/Cob : 29.19
Tubulações : 7.55
Revest/Pisos : 27.88
Esq/Vidros : 6.76
Pintura : 7.13
Serv.Compl. : 18.37

Fig.6.5(a) - Tela de Entrada de Dados

[F] AGORA ENTRE COM OS DADOS DA CURVA DE RECURSOS DISPONÍVEIS :

[F.1] O nº de parcelas da curva de recursos disponíveis :

[F.2] A periodicidade das parcelas (EM MESES) : 1

[F.3] Entre com as parcelas (Em %) :

1ª Parcela : 0
2ª Parcela : 2
3ª Parcela : 5
4ª Parcela : 11
5ª Parcela : 13
6ª Parcela : 14
7ª Parcela : 13
8ª Parcela : 14
9ª Parcela : 10
10ª Parcela : 8
11ª Parcela : 7
12ª Parcela : 3

Fig.6.5(b) - Tela de Entrada de Dados

Após a entrada de todos os dados, o sistema pedirá a confirmação de cada valor através de duas telas-resumos (Figura 6.6 e Figura 6.7).

PROJETO : EXEMPLO

[1] No. DE UNIDADES : 200 casa (s)

[2] RITMO MÁXIMO : 6 casa (s) / dia

ATIVIDADE	[3] DURAÇÃO	[4] PRECEDÊNCIA	[5] CUSTO (%)
Fundação	3.00	-	3.12
Est / Alv / Cob	8.50	1.00	29.19
Tubulações	2.00	0.00	7.55
Revest / Pisos	4.00	0.00	27.88
Esq / Vidros	2.33	0.00	6.76
Pintura	2.00	0.00	7.13
Serv. Compl.	0.83	0.00	18.37

Confirma (S/N) ? S

Fig.6.6- Tela de Confirmação de Dados da Rede Unitária

CURVA DE RECURSOS DISPONÍVEIS :

No. DE PARCELAS : 12 parcela (s)
PERIODICIDADE : 1.0 mês (es)

PARCELA	%
1.0 MÊS	0.00
2.0 MÊS	2.00
3.0 MÊS	5.00
4.0 MÊS	11.00
5.0 MÊS	13.00
6.0 MÊS	14.00
7.0 MÊS	13.00
8.0 MÊS	14.00
9.0 MÊS	10.00
10.0 MÊS	8.00
11.0 MÊS	7.00
12.0 MÊS	3.00

Confirma (S/N) ? S

Fig.6.7 - Tela de Confirmação de Dados da Curva de Agregação de Recursos Disponíveis

Neste ponto - nó D - o usuário tem possibilidade de alterar quaisquer dados que desejar. Para fazê-lo, deverá digitar N, não confirmando os valores apresentados em cada tela-resumo. A isso, seguirá a pergunta: **Digite o n° do dado que deseja alterar :** . Esse processo repetir-se-á até que o usuário confirme todos os dados - opção S.

Depois de confirmados, todos os dados serão gravados em um arquivo com o nome do projeto e extensão DAT, sendo gravado também o nome do projeto no arquivo NOMES.DAT. Dar-se-á, então, a simulação propriamente dita. O resultado da simulação será uma tela que mostrará o programa de execução do projeto (Figura 6.8).

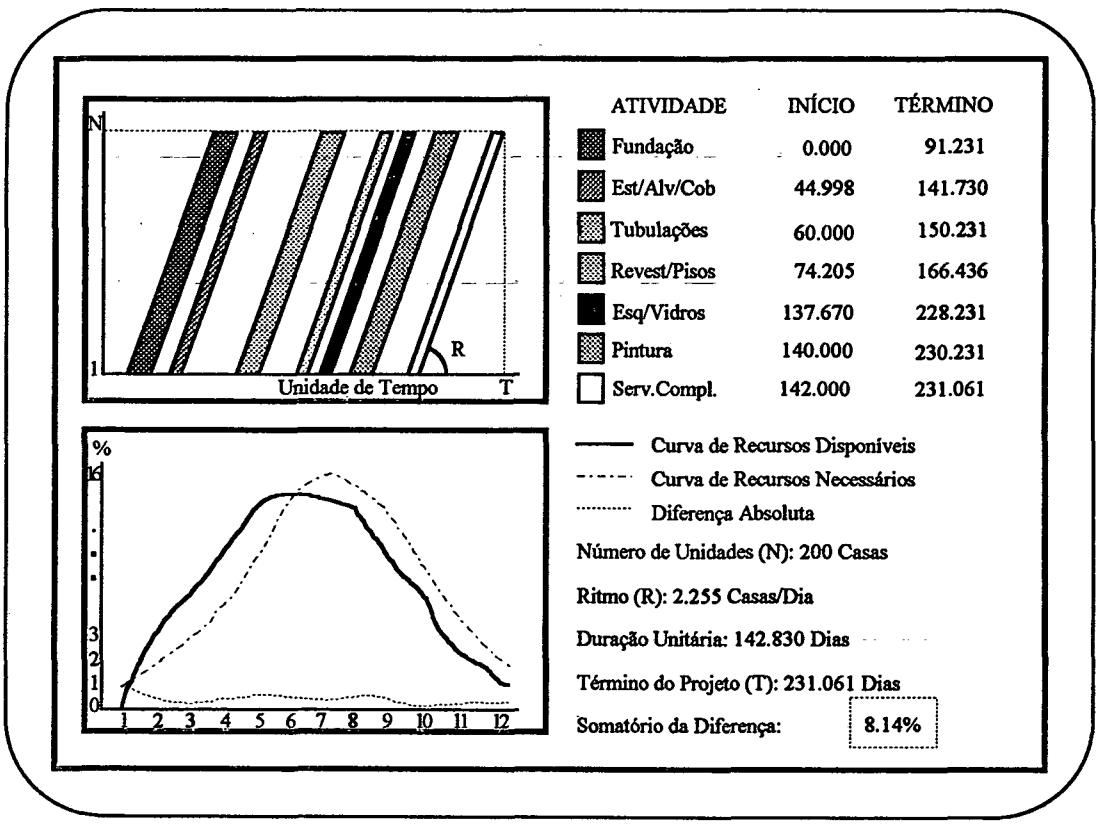


Fig.6.8 - Tela com o Programa de Execução do Projeto

Essa tela explica-se por si só. Os principais parâmetros - físicos e financeiros - da programação do projeto podem ser vistos gráfica e/ou textualmente no monitor do PC.

Depois de analisado o programa de execução sugerido, o usuário deverá apertar uma tecla qualquer para continuar. A última tela do sistema perguntará: **O USUÁRIO DESEJA TERMINAR (S/N) ?** , possibilitando ao usuário percorrer outros caminhos sem sair do sistema.

Para apresentar todo o sistema ao leitor, explorando cada um dos demais caminhos, neste ponto - nó F1- será escolhida a opção N.

Voltando ao nó A, suponhamos que o usuário deseje simular novamente o mesmo projeto, modificando porém alguns valores da curva de recursos disponíveis.

Respondendo opção [1] à primeira tela (Figura 6.3) e digitando o nome do projeto em seguida (Figura 6.4), o usuário percorrerá o caminho ABED, indo diretamente às telas de confirmação de dados (Figura 6.6 e Figura 6.7). Uma pesquisa ao arquivo NOMES.DAT provocará este desvio, evitando que o usuário tenha que entrar com todos os dados novamente.

Estando no nó D, o usuário poderá alterar a curva de agregação de recursos conforme desejar. Saindo do ***loop***, ou seja, confirmados todos os dados, o sistema percorrerá igualmente o caminho DF1, obtendo o novo programa de execução do projeto EXEMPLO.

E finalmente, para que o usuário possa obter, a qualquer tempo, informações sobre este ou outro projeto simulado anteriormente, basta que ele escolha a opção[2] na primeira tela do sistema (Figura 6.3) e, chegando ao nó de decisão G, digite a opção desejada (Figura 6.9).

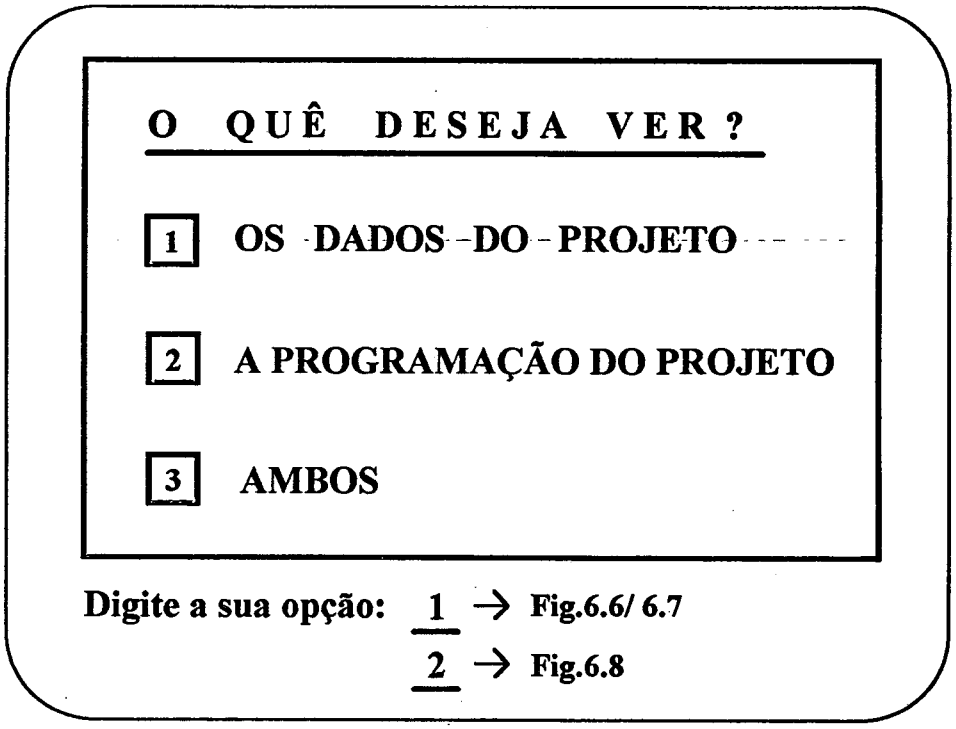


Fig. 6.9- Tela de Opções

Ao fazer sua opção, o usuário poderá ver, respectivamente:

- na ***opção[1]***, as telas-resumos com todos os dados do projeto. São as telas das Figura 6.6 e Figura 6.7. A diferença é que neste caminho - AG - não se pode alterar os dados do projeto;

- na ***opção[2]***, o programa de execução do projeto - tela da Figura 6.8;

- na ***opção[3]***, ambas informações - os dados e o programa de execução.

Percorrendo todos os caminhos possíveis, termina aqui o manual de utilização do sistema computacional desenvolvido neste trabalho.

6.4- Análise Operacional do Sistema

As limitações operacionais do sistema são, em ordem de importância, basicamente:

a) só programa as atividades repetitivas, ou seja , aquelas que fazem parte da Linha de Balanço, sendo necessário integrar o programa de execução obtido a outras atividades como serviços preliminares e serviços de infra-estrutura do conjunto habitacional. Considerando que o maior potencial de programação de projetos lineares está, justamente, na exploração das atividades repetitivas, promover essa integração a posteriori não chega a ser um fator muito limitante;

b) o sistema não possui um calendário embutido, sendo necessário transformar as datas, fornecidas em dias úteis, para valores reais. A idéia do sistema, no entanto, é explorar estratégias de construção para o projeto, obtendo, não um programa de execução ***fechado***, e sim parâmetros mais agregados que possam medir a qualidade da alternativa

de programação sugerida. Dessa forma, não se justifica incorporar um calendário ao sistema;

c) com o mesmo nível de importância da limitação anterior, observa-se que o sistema computacional produz um programa de execução, cujas variáveis de decisão são contínuas, ou seja, não necessariamente apresentam valores ***ótimos*** inteiros. Novamente, não se justifica investir na definição de critérios de arredondamento para se obter, em última análise, uma sugestão para a estratégia de execução do projeto simulado.

CAPÍTULO 7

A VALIDAÇÃO DO SISTEMA

Este capítulo dedica-se a apresentar e analisar os vários programas de execução dos projetos simulados com o sistema aqui desenvolvido. A análise do sistema, conforme mencionado anteriormente, fundamenta-se na comparação de seus resultados com a lógica de programação e a prática corrente. Serão descritos a seguir os critérios utilizados para a definição dos experimentos.

7.1- Descrição dos Experimentos

Noventa (90) projetos fictícios foram simulados com o intuito de verificar o comportamento do sistema às mais diversas situações reais. O critério utilizado para gerar esses projetos foi, basicamente, descobrir quais os dados de entrada do sistema que são mais determinantes na definição do programa de execução. Posteriormente, variar isoladamente cada um desses dados, observando sua influência no programa obtido.

Sendo assim, os dados de entrada foram agrupados da seguinte forma:

- projetos de 12, 18, 24, 30 ou 36 parcelas com, respectivamente, 200, 300, 400, 500 ou 600 unidades construtivas;

- curvas de agregação de recursos retangulares (parcelas iguais), trapezoidais (curva clássica), triangulares e irregulares. Foram acrescentadas também, uma curva com dados aleatórios e uma curva denominada ***ideal*** - a primeira para observar o comportamento do sistema em uma situação totalmente adversa e a segunda para verificar se o sistema converge para um programa de execução em que a compatibilização entre as curvas é total, ou seja, o somatório das diferenças entre as curvas de agregação é zero. Os valores percentuais de todas as curvas estão no ANEXO II;

- projetos com custos iguais para todas as atividades ou com um custo muito superior aos demais - para a primeira atividade, para uma atividade central ou para a última atividade da rede unitária do projeto;

- complementando os dados, durações unitárias e folgas mínimas iguais a zero, compondo o programa unitário do projeto.

As combinações desses grupos de dados geraram então os noventa projetos fictícios, que foram posteriormente simulados pelo sistema.

Verificando o ANEXO II, pode-se questionar a inclusão de projetos proporcionalmente semelhantes. Todos os projetos de 18, 24, 30 e 36 parcelas, exceto para aqueles com curvas de agregação irregulares, foram gerados proporcionalmente aos de 12 parcelas. Como o objetivo desse experimento é justamente observar o comportamento do sistema através da análise dos programas de execução que ele sugere, a inclusão de projetos semelhantes poderia medir a variabilidade de suas respostas.

7.2- Apresentação dos Resultados

Os resultados de todos os projetos, ou seja, os programas de execução, podem ser vistos nas Tabelas 7.1, 7.2, 7.3, 7.4 e 7.5 a seguir.

Antes porém da apresentação dos resultados, são necessários três esclarecimentos: o primeiro diz respeito aos códigos que aparecem na primeira coluna da tabela, o segundo, aos valores da penúltima coluna ($\Sigma|\Delta|$) e o terceiro, aos valores de Ritmo.

Os dígitos do *código* dizem respeito, nesta ordem, ao número de parcelas da curva, ao tipo de curva (A = trapezoidal, B = triangular, C = retangular e D = irregular) e aos custos das atividades (1 = custos iguais, 21 = o custo da primeira atividade maior que os demais, 22 = custo maior para uma atividade central, 23 = o custo da última atividade é maior). Os dois últimos códigos referem-se, respectivamente, à curva aleatória (RAM) e ideal .

Quanto ao segundo esclarecimento, deve-se salientar que a variável ***SQRDIF*** tem utilidade apenas para a otimização do modelo matemático, sendo desprovida de significado real. Os

valores apresentados na tabela referem-se à soma real das diferenças entre as curvas de agregação.

Finalmente, observa-se que os valores de ***Ritmo*** são apresentados nas tabelas de forma invertida em relação à unidade de medida utilizada na formulação do modelo matemático. O valor de ***Ritmo*** em unidades repetitivas/dia facilita a interpretação do programa de execução sugerido pelo sistema. No modelo, sua utilização em dias/unidade repetitiva teve por objetivo linearizar muitas das restrições contidas na formulação matemática.

Seguem então as tabelas com os resultados de todos os projetos simulados.

Tabela 7.1 - Programas de Execução para os Projetos de Doze Parcelas

Código	Ritmo	DU	DE(1')	DE(2')	DE(3')	DE(4')	DE(5')	DE(6')	DE(7')	Σ|Δ|%	Tempo*
12A1	2.657	156.9	4.1	45.1	72.0	80.2	122.6	144.1	160.0	3.424	34"
12A21	1.202	64.3	10.2	40.0	53.5	56.8	71.5	72.5	73.5	10.169	35"
12A22	1.095	48.9	9.4	10.4	17.3	35.4	55.3	56.3	57.3	7.309	22"
12A23	1.118	59.8	0.0	1.0	20.0	21.0	40.0	41.0	58.8	6.114	29"
12B1	1.767	112.2	6.4	26.4	47.8	68.9	90.0	116.6	117.6	2.436	33"
12B21	1.252	67.3	11.1	40.0	60.0	61.0	75.4	76.4	77.4	8.840	28"
12B22	1.420	95.0	0.0	18.8	20.9	51.4	92.0	93.0	94.0	6.384	26"
12B23	1.314	77.9	0.0	7.5	24.7	27.5	47.5	48.9	76.9	5.602	35"
12C1	0.854	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	4.260	19"
12C21	1.672	121.0	0.0	1.0	116.0	117.0	118.0	119.0	120.0	1.847	22"
12C22	0.854	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	4.260	19"
12C23	1.672	121.0	0.0	1.0	2.0	3.0	4.0	119.0	120.0	1.847	23"
12D1	5.773	200.0	1.0	42.9	78.1	109.0	132.2	164.9	200.0	4.218	40"
12D21	1.655	119.7	0.0	63.0	100.0	115.7	116.7	117.7	118.7	6.293	34"
12D22	2.519	161.0	0.0	1.0	3.0	83.0	138.0	159.0	160.0	9.250	27"
12D23	1.716	121.0	0.0	1.0	2.0	3.0	63.0	95.6	120.0	6.376	25"
RAM12	2.740	144.0	0.0	9.0	44.1	109.0	120.0	131.0	138.0	34.905	26"
IDEAL12	0.952	25.7	0.0	1.0	10.1	21.7	22.7	23.7	24.7	0.729	28"

* A última coluna refere-se aos tempos totais de cada simulação. Esses valores foram obtidos com um PC-AT 386 de 40MHZ e coprocessador numérico.

Tabela 7.2 - Programas de Execução para os Projetos de Dezoito Parcelas

Código	Ritmo	DU	DE(1')	DE(2')	DE(3')	DE(4')	DE(5')	DE(6')	DE(7')	Σ|Δ|%	Tempo*
18A1	2.710	232.3	8.7	66.6	108.4	124.4	184.4	220.0	240.0	5.289	38"
18A21	1.143	83.8	14.5	60.0	77.3	85.5	92.7	93.7	97.3	9.476	35"
18A22	1.105	89.5	0.0	20.0	28.5	56.0	86.5	87.5	88.5	6.780	35"
18A23	1.119	90.3	0.0	20.0	32.3	50.8	51.8	71.8	89.3	6.085	41"
18B1	1.767	176.1	4.9	41.7	71.5	102.2	132.0	158.8	180.0	3.515	44"
18B21	1.365	122.9	14.6	60.0	80.0	100.0	120.0	135.5	136.5	8.644	45"
18B22	1.365	139.6	1.4	20.0	40.0	71.2	120.0	132.0	140.0	7.527	45"
18B23	1.503	133.5	0.0	20.0	40.0	58.0	75.9	94.4	132.5	6.040	43"
18C1	0.847	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.991	25"
18C21	0.847	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.991	24"
18C22	0.847	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.991	24"
18C23	0.847	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.991	25"
18D1	4.503	261.0	0.0	1.0	67.1	140.0	192.6	212.6	260.0	11.848	37"
18D21	3.371	261.0	0.0	100.0	104.1	180.0	189.7	200.0	260.0	16.593	38"
18D22	1.056	21.0	0.0	1.0	2.0	3.0	4.0	5.0	20.0	32.743	34"
18D23	3.256	195.4	0.0	1.0	2.0	30.4	101.7	140.0	194.4	24.640	37"
RAM18	3.352	257.6	0.0	5.0	40.0	111.5	180.0	213.4	251.6	40.766	41"
IDEAL18	1.000	61.0	0.0	9.8	20.0	30.0	40.0	50.2	60.0	0.073	39"

* Idem Tabela 7.1.

Tabela 7.3 - Programas de Execução para os Projetos de Vinte e Quatro Parcelas

Código	Ritmo	DU	DE(1')	DE(2')	DE(3')	DE(4')	DE(5')	DE(6')	DE(7')	Σ|Δ|%	Tempo*
24A1	1.113	121.7	0.0	40.0	60.0	82.0	106.4	119.7	120.7	6.090	40"
24A21	1.160	122.2	13.8	80.0	100.0	115.0	119.0	120.0	135.0	12.182	52"
24A22	1.111	121.0	0.0	20.0	40.0	73.8	115.7	116.7	120.0	6.802	42"
24A23	1.111	114.3	0.0	20.0	40.0	60.0	75.3	84.2	113.3	5.696	47"
24B1	1.625	210.6	10.4	48.3	86.2	124.2	162.1	200.0	220.0	4.260	57"
24B21	1.430	176.9	16.2	100.0	120.0	140.0	161.1	191.1	192.1	9.874	50"
24B22	1.677	240.1	0.9	32.9	60.7	113.5	200.0	220.0	240.0	9.170	52"
24B23	1.702	212.3	0.0	34.2	62.1	88.4	120.0	144.6	211.3	7.694	58"
24C1	0.844	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.294	29"
24C21	0.844	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.294	29"
24C22	0.844	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.294	28"
24C23	0.844	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	2.294	29"
24D1	2.404	314.0	0.0	1.0	2.0	120.0	280.0	312.0	313.0	29.617	43"
24D21	2.728	333.7	0.0	140.0	192.7	280.0	330.7	331.7	332.7	25.666	43"
24D22	0.844	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	41.705	33"
24D23	3.353	358.0	0.0	1.0	2.0	72.0	125.7	240.0	357.0	25.833	47"
RAM24	4.234	286.0	0.0	5.0	80.0	120.9	140.0	200.5	280.0	35.731	41"
IDEAL24	1.113	121.0	0.0	20.0	40.0	60.0	80.3	100.3	120.0	0.325	32"

* Idem Tabela 7.1.

Tabela 7.4 - Programas de Execução para os Projetos de Trinta Parcelas

Código	Ritmo	DU	DE(1')	DE(2')	DE(3')	DE(4')	DE(5')	DE(6')	DE(7')	Σ|Δ|%	Tempo*
30A1	1.118	144.9	9.2	53.1	77.1	103.1	133.1	151.8	153.1	6.200	1'08"
30A21	1.137	143.9	17.1	100.0	120.0	140.0	151.8	152.8	160.0	10.835	1'06"
30A22	1.105	143.7	4.7	40.0	59.6	94.7	145.4	146.4	147.4	6.944	59"
30A23	1.095	132.5	3.1	40.0	57.7	71.8	85.6	103.1	134.6	6.371	1'03"
30B1	1.533	247.5	13.5	62.4	103.3	144.3	185.3	226.2	260.0	3.832	1'05"
30B21	1.390	215.5	16.3	120.0	140.0	180.0	200.0	229.8	230.8	11.342	1'07"
30B22	1.404	237.4	3.6	32.2	63.6	114.5	200.0	226.1	240.0	8.077	1'14"
30B23	1.472	212.7	0.0	33.8	65.7	91.4	123.2	148.9	211.7	6.632	1'14"
30C1	0.841	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.872	36"
30C21	0.841	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.872	35"
30C22	0.841	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.872	36"
30C23	0.841	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.872	36"
30D1	1.675	298.6	0.2	1.2	60.0	120.0	169.7	228.8	297.8	7.928	1'13"
30D21	1.228	192.1	0.0	59.0	60.0	119.0	120.0	180.0	191.1	10.638	59"
30D22	1.394	241.0	0.0	1.0	2.0	70.5	167.2	180.0	240.0	10.769	1'05"
30D23	2.038	241.0	0.0	1.0	54.1	61.8	120.0	174.7	240.0	13.122	1'04"
RAM30	1.908	338.4	0.0	5.0	120.0	161.6	314.4	325.4	332.4	57.465	1'45"
IDEAL30	1.042	121.0	0.0	20.0	40.0	60.0	80.0	100.0	120.0	0.074	1'18"

* Idem Tabela 7.1.

Tabela 7.5 - Programas de Execução para os Projetos de Trinta e Seis Parcelas

Código	Ritmo	DU	DE(1')	DE(2')	DE(3')	DE(4')	DE(5')	DE(6')	DE(7')	Σ|Δ|%	Tempo*
36A1	1.111	161.4	14.9	60.0	92.9	122.3	161.4	174.3	175.3	6.013	1'15"
36A21	1.111	159.0	18.0	120.0	140.0	160.0	174.0	175.0	176.0	10.124	1'15"
36A22	1.109	179.9	0.0	40.0	60.0	111.7	176.9	177.9	178.9	6.696	1'19"
36A23	1.081	151.3	4.8	40.0	60.0	84.8	104.8	124.8	155.1	7.166	1'43"
36B1	1.649	325.3	15.7	83.2	128.4	186.1	243.7	301.3	340.0	5.105	1'27"
36B21	1.364	264.6	16.4	140.0	180.0	200.2	240.0	260.0	280.0	12.618	1'25"
36B22	1.383	279.3	5.5	45.9	68.7	135.1	240.0	265.9	283.8	9.292	1'45"
36B23	1.789	314.1	0.0	60.0	100.0	140.0	184.8	225.1	313.1	9.023	1'47"
36C1	0.840	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.556	42"
36C21	0.840	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.556	41"
36C22	0.840	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.556	41"
36C23	0.840	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	1.556	40"
36D1	2.580	487.9	0.0	1.0	2.0	89.0	245.3	480.0	486.9	28.587	1'20"
36D21	2.483	478.7	0.0	1.0	120.0	240.0	475.7	476.7	477.7	27.331	1'06"
36D22	0.840	7.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	41.636	42"
36D23	3.056	517.4	0.0	1.0	2.0	3.0	120.0	200.0	516.4	29.614	1'11"
RAM36	1.465	306.0	0.0	5.0	50.0	60.0	120.0	180.0	300.0	57.366	1'25"
IDEAL36	0.999	120.2	0.0	19.6	40.0	60.0	79.2	100.0	119.2	0.279	1'22"

* Idem Tabela 7.1.

7.3- Análise dos Resultados

A base da análise é o cruzamento entre os grupos de dados e os programas de execução sugeridos pelo sistema. Procura-se observar tendências nos programas de execução em função dos dados de entrada e verificar se eles refletem a lógica de programação da Técnica da Linha de Balanço, observando-se também os valores correspondentes da função objetivo.

A análise dos programas de execução é feita a partir de variáveis de programação mais agregadas. Explicando melhor, são observadas as variáveis de programação que realmente definem um programa de execução a nível tático, ou seja, o ritmo de construção, a duração unitária e os ***buffers*** horizontais da rede unitária.

Sabe-se que a forma da curva de agregação de recursos necessários, gerada pela programação com a Linha de Balanço, aproxima-se de um trapézio, com o tamanho da base menor proporcional ao ritmo de construção e à duração unitária (Figura 7.1).

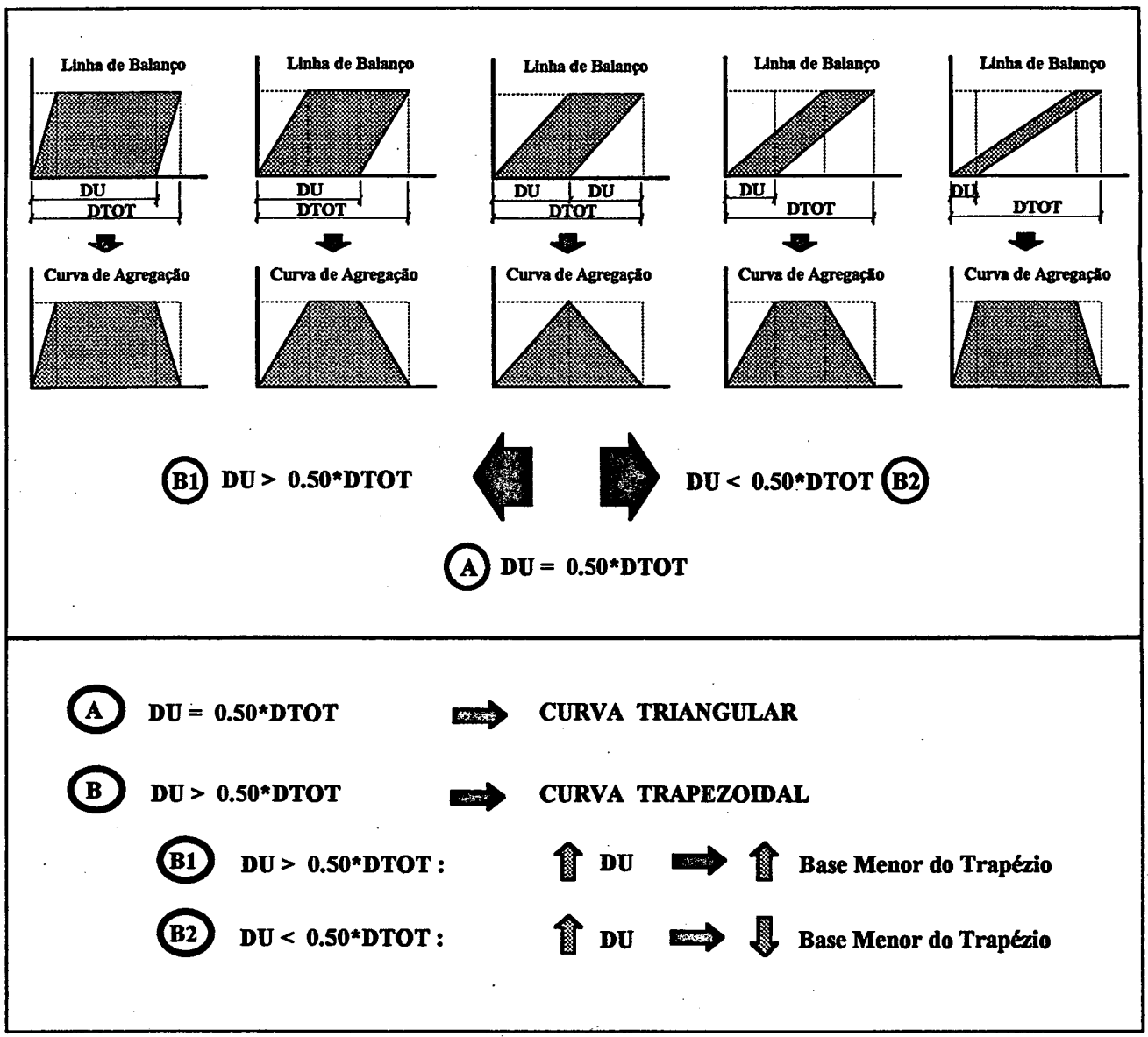


Fig. 7.1 - Linha de Balanço x Curva de Agregação

Partindo-se do princípio que a concepção do sistema reproduz esta assertiva, procede-se a análise das diferentes combinações de grupos de dados de entrada. Antes porém, serão apresentadas tabelas complementares (Tabela 7.6 a 7.10) com os valores dos ***buffers horizontais*** observados em cada programa de execução e, também, uma tabela-resumo com a compilação dos resultados obtidos na experimentação.

Tabela 7.6 - *Buffers* Horizontais / Projetos de Doze Parcelas

Código	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)	BUFFER(6)	BUFFER(7)
12A1	4.1	40.0	25.9	7.2	41.4	20.5	14.9
12A21	10.2	28.8	12.5	2.3	13.7	0.0	0.0
12A22	9.4	0.0	5.9	17.1	18.9	0.0	0.0
12A23	0.0	0.0	18.0	0.0	18.0	0.0	16.8
12B1	6.4	19.0	20.4	20.1	20.1	25.6	0.0
12B21	11.1	27.9	19.0	0.0	13.4	0.0	0.0
12B22	0.0	17.8	1.1	29.5	39.6	0.0	0.0
12B23	0.0	6.5	16.2	1.8	19.0	0.4	27.0
12C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12C21	0.0	0.0	114.0	0.0	0.0	0.0	0.0
12C22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12C23	0.0	0.0	0.0	0.0	0.0	114.0	0.0
12D1	1.0	40.9	34.2	29.9	22.2	31.7	34.1
12D21	0.0	62.0	36.0	14.7	0.0	0.0	0.0
12D22	0.0	0.0	1.0	79.0	54.0	20.0	0.0
12D23	0.0	0.0	0.0	0.0	59.0	31.6	23.4
RAM12	0.0	6.0	29.1	56.9	7.0	2.0	2.0
IDEAL12	0.0	0.0	8.1	10.6	0.0	0.0	0.0

Tabela 7.7 - *Buffers* Horizontais / Projetos de Dezoito Parcelas

Código	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)	BUFFER(6)	BUFFER(7)
18A1	8.7	56.9	40.8	15.0	59.0	34.6	19.0
18A21	14.5	44.5	16.3	7.2	6.2	0.0	2.6
18A22	0.0	19.0	7.5	26.5	29.5	0.0	0.0
18A23	0.0	19.0	11.3	17.5	0.0	19.0	16.5
18B1	4.9	35.8	28.8	29.7	28.8	25.8	20.2
18B21	14.6	44.4	19.0	19.0	19.0	14.5	0.0
18B22	1.4	17.6	19.0	30.2	47.8	11.0	7.0
18B23	0.0	19.0	19.0	17.0	16.9	17.5	37.1
18C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18C21	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18C22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18C23	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18D1	0.0	0.0	65.1	71.9	51.6	19.0	46.4
18D21	0.0	99.0	3.1	74.9	8.7	9.3	59.0
18D22	0.0	0.0	0.0	0.0	0.0	0.0	14.0
18D23	0.0	0.0	0.0	27.4	70.3	37.3	53.4
RAM18	0.0	2.0	29.0	63.5	64.5	24.4	33.2
IDEAL18	0.0	8.8	9.2	9.0	9.0	9.2	8.8

Tabela 7.8 - *Buffers* Horizontais / Projetos de Vinte e Quatro Parcelas

Código	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)	BUFFER(6)	BUFFER(7)
24A1	0.0	39.0	19.0	21.0	23.4	12.3	0.0
24A21	13.8	65.2	19.0	14.0	3.0	0.0	14.0
24A22	0.0	19.0	19.0	32.8	40.9	0.0	2.3
24A23	0.0	19.0	19.0	19.0	14.3	7.9	28.1
24B1	10.4	36.9	36.9	37.0	36.9	36.9	19.0
24B21	16.2	82.8	19.0	19.0	20.1	29.0	0.0
24B22	0.9	31.0	26.8	51.8	85.5	19.0	19.0
24B23	0.0	33.2	26.9	25.3	30.6	23.6	65.7
24C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24C21	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24C22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24C23	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24D1	0.0	0.0	0.0	117.0	159.0	31.0	0.0
24D21	0.0	139.0	51.7	86.3	49.7	0.0	0.0
24D22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24D23	0.0	0.0	0.0	69.0	52.7	113.3	116.0
RAM24	0.0	2.0	69.0	32.9	15.1	51.5	74.5
IDEAL24	0.0	19.0	19.0	19.0	19.3	19.0	18.7

Tabela 7.9 - *Buffers* Horizontais / Projetos de Trinta Parcelas

Código	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)	BUFFER(6)	BUFFER(7)
30A1	9.2	42.9	23.0	25.0	29.0	17.7	0.3
30A21	17.1	81.9	19.0	19.0	10.8	0.0	6.2
30A22	4.7	34.3	18.6	34.1	49.7	0.0	0.0
30A23	3.1	35.9	16.7	13.1	12.8	16.5	30.5
30B1	13.5	47.9	39.9	40.0	40.0	39.9	32.8
30B21	16.3	102.7	19.0	39.0	19.0	28.8	0.0
30B22	3.6	27.6	30.4	49.9	84.5	25.1	12.9
30B23	0.0	32.8	30.9	24.7	30.8	24.7	61.8
30C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30C21	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30C22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30C23	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30D1	0.2	0.0	57.8	59.0	48.7	58.1	68.0
30D21	0.0	58.0	0.0	58.0	0.0	59.0	10.1
30D22	0.0	0.0	0.0	67.5	95.7	11.8	59.0
30D23	0.0	0.0	52.1	6.7	57.2	53.7	64.3
RAM30	0.0	2.0	109.0	33.6	148.8	2.0	2.0
IDEAL30	0.0	19.0	19.0	19.0	19.0	19.0	19.0

Tabela 7.10 - *Buffers* Horizontais / Projetos de Trinta e Seis Parcelas

Código	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)	BUFFER(6)	BUFFER(7)
36A1	14.9	44.1	31.9	28.4	38.1	11.9	0.0
36A21	18.0	101.0	19.0	19.0	13.0	0.0	0.0
36A22	0.0	39.0	19.0	50.7	64.2	0.0	0.0
36A23	4.8	34.2	19.0	23.8	19.0	19.0	29.3
36B1	15.7	66.5	44.2	56.7	56.6	56.6	37.7
36B21	16.4	122.6	39.0	19.2	38.8	19.0	19.0
36B22	5.5	39.4	21.8	65.4	103.9	24.9	16.9
36B23	0.0	59.0	39.0	39.0	43.8	39.3	87.0
36C1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36C21	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36C22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36C23	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36D1	0.0	0.0	0.0	86.0	155.3	233.7	5.9
36D21	0.0	0.0	118.0	119.0	234.7	0.0	0.0
36D22	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36D23	0.0	0.0	0.0	0.0	116.0	79.0	315.4
RAM36	0.0	2.0	39.0	2.0	56.0	51.0	115.0
IDEAL36	0.0	18.6	19.4	19.0	18.2	19.8	18.2

Tabela 7.11 - Resumo dos Resultados

PROJETO	RITMO (unidades/dia)	DURAÇÃO UNITÁRIA (em %Duração Total)	Σ|Δ|%
A1	Entre 1.11 e 2.71	Entre 23 e 68%	Entre 3.42 e 6.20
B1	Entre 1.53 e 1.76	Entre 43 e 51%	Entre 2.43 e 5.10
C1	Em torno de 0.85	Em torno de 2%	Entre 1.87 e 4.26
D1	Entre 1.67 e 5.77	Entre 50 e 85%	Entre 4.22 e 29.62
A21	Entre 1.11 e 1.20	Entre 22 e 27%	Entre 9.47 e 12.18
B21	Entre 1.25 e 1.43	Entre 30 e 38%	Entre 8.64 e12.62
C21	Entre 0.84 e 1.67	Entre 1 e 50%	Entre 1.55 e 2.99
D21	Entre 1.22 e 3.37	Entre 32 e 74%	Entre 6.29 e 27.33
A22	Em torno de 1.11	Em torno de 25%	Entre 6.69 e 7.31
B22	Entre 1.36 e 1.67	Entre 39 e 50%	Entre 6.38 e 9.29
C22	Em torno de 0.85	Em torno de 2%	Entre 1.55 e 4.26
D22	Entre 0.84 e 2.52	Entre 1 e 67%	Entre 9.25 e 41.70
A23	Em torno de 1.11	Em torno de 25%	Entre 5.69 e 7.26
B23	Entre 1.31 e 1.79	Entre 34 e 48%	Entre 5.60 e 9.02
C23	Entre 0.84 e 1.67	Entre 1 e 50%	Entre 1.55 e 2.99
D23	Entre 1.72 e 3.35	Entre 50 e 75%	Entre 6.37 e 30.25
RAM	Entre 1.46 e 4.23	Entre 43 e 76%	Entre 34.91 e 57.46
IDEAL	Entre 0.95 e 1.11	Entre 11 e 25%	Entre 0.07 e 0.73

Aos projetos que têm ***custos iguais*** para todas as atividades e curva de agregação de recursos disponíveis na forma ***trapezoidal (A1)***, resta somente explorar o binômio ritmo x ***buffers*** para tentar se adequar à curva de agregação de recursos imposta.

A simulação desses projetos gerou programas com ritmos entre 1.11 e 2.71 unidades construtivas por dia e durações unitárias entre 23 e 68% das respectivas durações totais. Apesar da relativa variabilidade nos programas obtidos, os baixos valores da função objetivo - entre 3.42 e 6.20% - demonstram que o posicionamento das atividades proporcionou a compatibilização desejada.

A variabilidade mencionada acima deve-se à grande quantidade de alternativas de programação, possíveis e satisfatórias, para cada projeto simulado. Isso explica porque projetos proporcionalmente semelhantes - de 12, 18, 24, 30 e 36 parcelas - obtiveram programas de execução proporcionalmente diferentes. Já a variabilidade na qualidade dos programas, representada pelos valores da função objetivo, deve-se aos inúmeros máximos e mínimos locais que o modelo matemático, que dá suporte ao sistema, contém.

Os projetos com ***custos iguais*** e curva de agregação ***triangular (B1)*** apresentaram programas com ritmos entre 1.53 e 1.76 unidades por dia e durações unitárias entre 43 e 51% da durações totais.

Se o sistema permitisse ritmos diferenciados, uma alternativa de programação para esses projetos seria posicionar atividades com ritmos mais altos, concentradas na região do pico da curva de recursos disponíveis. Como isso não é possível, a solução sugerida pelo sistema foi um programa de execução com ritmos e ***buffers*** relativamente baixos para que, na região do pico, contribuíssem paralelamente atividades em fase de término e na fase inicial.

Esperava-se ritmos maiores que os da curva trapezoidal, mas em alguns projetos, entre as inúmeras alternativas factíveis, o sistema convergiu para soluções onde o ritmo é pequeno e os ***buffers*** quase constantes.

Não se pode culpar o sistema por essa aparente incoerência. Seu modelo matemático é geral e, portanto, pouco restritivo. Sendo assim, tem maior liberdade para produzir programas de execução bem diversificados, guiado apenas pela medida de qualidade da solução alternativa - a função objetivo. Neste aspecto, pode-se dizer que as soluções sugeridas são de ótima qualidade - valores entre 2.43 e 5.10%.

Já os programas de execução de projetos com curvas de agregação ***retangulares*** tenderam, de modo geral, a ritmos e durações unitárias mínimas, tanto para projetos com ***custos iguais (C1)*** quanto para projetos com ***custos diferentes (C21, C22, C23)***.

Os valores dos ritmos giraram em torno de 0.85 unidades repetitivas por dia e as durações unitárias em 1% das respectivas durações totais, resultando em ótimos programas de execução - funções objetivos entre 1.55 e 4.26%.

A análise dos projetos com curvas de agregação ***irregulares (D)*** e ***ramdômicas (RAM)*** é feita de forma globalizada. Nestes projetos, o sistema, na maioria das vezes, não conseguiu obter uma boa compatibilização entre as curvas de agregação de recursos. Os valores foram

muito altos - entre 4.22 e 57.46% para as curvas D e entre 34.91 e 57.36% para as curvas RAM - porque as melhores alternativas de execução estavam fora do espaço de soluções viáveis, ou seja, ritmos diferenciados para permitir maior adequação aos picos e vales da curva de agregação de recursos disponíveis.

Analisando agora os projetos com ***custos diferentes*** e curvas de agregação ***trapezoidais*** e ***triangulares***, verifica-se que houve bastante regularidade nas soluções sugeridas pelo sistema.

Os projetos ***A21*** apresentaram os seguintes valores: ritmos entre 1.11 e 1.20 unidades construtivas por dia, durações unitárias entre 22 e 27% das durações totais e funções objetivos entre 9.47 e 12.18%. Como era de se esperar, em função do maior custo da primeira atividade, foram obtidos programas com ***buffers*** grandes no início e pequenos entre as demais atividades da rede unitária.

Os projetos ***A22*** apresentaram os seguintes valores: ritmos em torno de 1.11 unidades construtivas por dia, durações unitárias em torno de 25% das durações totais e funções objetivos entre 6.69 e 7.31%. Nestes projetos a atividade mais cara está posicionada na região central da rede unitária e, também aqui, o sistema conseguiu refletir a lógica de programação, produzindo programas com ***buffers*** pequenos no início e fim da rede e grandes entre as atividades centrais.

Os projetos ***A23*** obtiveram resultados bem semelhantes aos anteriores: ritmos em torno de 1.11 unidades construtivas por dia, durações unitárias em torno de 25% das durações totais e funções objetivos entre 5.69 e 7.26%. Os ***buffers*** desta vez tendem a ser maiores entre as atividades finais da rede unitária, devido ao maior custo da última atividade. Deve ser colocado que nem sempre isso ocorreu, devido principalmente a limitações da heurística desenvolvida neste trabalho e à generalidade do modelo matemático apresentado.

Os projetos ***B21*** apresentaram os seguintes resultados: ritmos entre 1.25 e 1.43 unidades construtivas por dia, durações unitárias entre 30 e 38% das durações totais e funções objetivos entre 8.64 e 12.62%.

Nesse tipo de projeto, a tendência é produzir programas com grandes ***buffers*** entre as primeiras atividades e ***buffers*** menores entre as demais, tanto pelo maior custo da primeira atividade quanto pela forma triangular da curva de agregação de recursos disponíveis. Como pode ser verificado nos valores apresentados nas Tabelas 7.1 a 7.5, os programas de execução sugeridos pelo sistema refletem essa lógica de programação.

Os projetos ***B22*** apresentaram os seguintes resultados: ritmos entre 1.36 e 1.67 unidades construtivas por dia, durações unitárias entre 39 e 50% das durações totais e funções objetivos entre 6.38 e 9.29%. Aqui, a combinação dos dados - curva de agregação triangular e atividade central com custo maior - foi favorável, forçando a produção de programas com ritmos um pouco maiores que os anteriores e ***buffers*** grandes em torno da atividade central, posicionada próximo ao pico da curva.

Os projetos ***B23*** apresentaram os seguintes resultados: ritmos entre 1.31 e 1.79 unidades construtivas por dia, durações unitárias entre 34 e 48% das durações totais e funções objetivos entre 5.60 e 9.02%. Forçado a grandes ***buffers*** no início da rede unitária, por conta da forma da curva de agregação e no final da rede, tanto pela forma da curva como pelo fato da última atividade apresentar custo maior, o sistema produziu, como era de se esperar, programas de execução com ritmos mais elevados e mais atividades concentradas na região do pico da curva.

Fazendo-se uma análise conclusiva sobre os projetos A2 e B2, constata-se que o sistema apresenta maior dificuldade para programar projetos que possuam as primeiras atividades mais caras que as demais (A21, B21) e, ao contrário, maior facilidade para programar projetos que possuam as últimas atividades mais caras que as demais (A23, B23).

Pode-se dizer, sem risco de cometer um engano, que a heurística desenvolvida para produzir uma solução inicial para o modelo matemático é a responsável pelas afirmações feitas acima.

No primeiro caso, os piores resultados apresentados pelos projetos A21 e B21 têm origem no passo 2 do algoritmo, que fixa a data de início da primeira atividade em zero. Como se sabe, devido aos muitos máximos e mínimos locais que o modelo matemático apresenta, sua solução final pouco difere da solução inicial produzida pela heurística. Isso impede que a primeira atividade - a mais cara - seja programada muito depois da data zero.

Em contrapartida, o passo 10 do algoritmo proporciona maior liberdade para se programar a última atividade, justificando os melhores resultados apresentados pelos projetos A23 e B23.

Por fim, apresentam-se os resultados dos projetos que têm as curvas de agregação denominadas ***ideais***. Na verdade, não existe uma curva teoricamente ideal. Cada uma dessas curvas foi obtida a partir da curva de agregação de recursos necessários, gerada por um programa de execução qualquer para cada projeto.

O objetivo da inclusão desse experimento parece transparente. Pretende-se observar se o sistema converge para programas de execução que apresentam compatibilização máxima entre

as curvas de agregação de recursos, representada matematicamente por valores próximos de zero para a função objetivo do modelo.

Como pode ser verificado pelo resultado da simulação, o sistema confirmou as expectativas, apresentando valores bem baixos para a função objetivo - entre 0.07 e 0.73% de descolamento entre as curvas de agregação de recursos.

7.4- A Utilização do Sistema como Ferramenta de Programação de Obras

Ao longo de toda a análise, foram constatadas virtudes e limitações operacionais que o sistema apresenta ao tentar representar a realidade da tarefa de planejamento e programação de projetos lineares. A maioria das limitações do sistema decorre das características do modelo matemático que lhe dá suporte. As outras limitações, no entanto, são provocadas pela generalidade do modelo, ou melhor, pela incorporação apenas de restrições consideradas básicas para a representação da realidade de programação desses tipos de projetos.

Para atenuar as limitações decorrentes da não convexidade do modelo matemático, foi desenvolvida uma heurística que produz os valores iniciais das variáveis de decisão do modelo. Com esta heurística foram obtidas soluções iniciais de ótima qualidade, conseguindo-se, na maioria das vezes, eliminar a dificuldade imposta pelas características inerentes ao modelo matemático incorporado ao método de programação.

As outras limitações decorrem de um espaço de soluções viáveis pouco restritivo, combinado com os incontáveis máximos e mínimos locais que o modelo apresenta. Isso pode levar o sistema a sugerir soluções, muitas vezes, incoerentes com a prática salutar. Um exemplo claro disso são os grandes ***buffers*** horizontais observados em alguns programas de execução apresentados nas Tabelas 7.1 a 7.5.

Para limitar as alternativas viáveis e forçar o sistema a convergir para uma estratégia de execução desejada, é necessário acrescentar portanto restrições de tempo e/ou custo que possam refletir as preferências do programador de projetos - o usuário potencial do sistema.

Terminando a análise operacional do sistema pode-se dizer que, de um modo geral, ele comporta-se como esperado, ou seja, reflete o bom senso de um programador experiente nos programas de execução que sugere. O tempo total de simulação de cada projeto também é muito favorável, podendo-se concluir que o sistema é viável.

Deve ser ressaltado que a experimentação aqui realizada não teve qualquer preocupação de ordem estatística e que, portanto, todas as conclusões provenientes dessa análise não podem ser encaradas como definitivas. Servem apenas como indicativo do comportamento do sistema às diversas situações testadas.

E ainda, pelas inúmeras alternativas de programação consideradas satisfatórias que cada projeto apresenta, podendo-se, pelas características do modelo matemático, convergir para qualquer uma delas, a realização de uma análise de sensibilidade a esse nível de desenvolvimento do sistema mostra-se de pouca utilidade prática.

Faz-se necessário, neste momento, empreender uma análise do potencial de utilização do sistema em situações reais de programação. Só para relembrar, o sistema foi desenvolvido com a proposta de tornar-se uma ferramenta de auxílio à programação de projetos lineares, a um nível de decisão tático.

A esse nível de decisão, o objetivo principal da programação de projetos é encontrar a melhor estratégia de construção, definindo os grandes marcos da obra e o volume de recursos físicos e financeiros necessários à execução do programa de execução obtido. É preciso lembrar também que o tipo e nível de agregação das informações de entrada e saída desse processo decisório, têm que ser compatíveis com o nível hierárquico de decisão.

A utilidade prática do sistema dá-se à medida que ele produz um programa de execução agregado - em que define as dadas de início e término dos principais serviços de obra e o ritmo de construção do projeto -, tentando compatibilizar os fluxos de recursos financeiros.

Através da incorporação de condicionantes financeiras - representadas pela curva de agregação de recursos disponíveis - e limitações de ordem técnica e prática, o sistema gera uma alternativa de construção bastante favorável do ponto de vista financeiro e, ao mesmo tempo, dentro das exigências técnicas que o processo construtivo impõe à execução do projeto.

Analisando-se uma das principais limitações do sistema - um número máximo de sete atividades na rede unitária do projeto - poder-se-ia dizer que isso não chega a ser um fator realmente limitante, considerando-se que não existem mais que sete grandes marcos em um projeto de construção.

A este ponto, já promovida a qualidade da resposta de sistema, poder-se-ia questionar o parâmetro tempo total de simulação de cada projeto, ou melhor, tempo de resposta do sistema.

Comparando-se grosseiramente o tempo necessário para um programador experiente obter o programa de execução de um projeto - em torno de duas horas - e o tempo de resposta do sistema - em média 40 segundos - também sob este aspecto esta ferramenta apresenta-se altamente atraente.

Estimulado pelo tempo de resposta do sistema, muito favorável, o usuário pode simular várias alternativas de programação, variando os dados de entrada do sistema. Simulando o projeto para diversas possibilidades de fluxo de recursos, a empresa de Construção pode, dentro de certos limites, programar-se financeiramente da forma que mais lhe convier.

CAPÍTULO 8

CONCLUSÕES

Neste capítulo conclusivo são compiladas as principais lições que o desenvolvimento do trabalho proporcionou. São apresentadas conclusões gerais, relativas à Construção Civil e ao processo de planejamento de sua atividade produtiva e, também, conclusões específicas sobre o método de programação de projetos lineares proposto.

8.1- Conclusões Gerais

O trabalho inicialmente contextualiza a Indústria da Construção Civil, apresentando as principais características que tornam este setor diferente dos outros setores industriais.

Em seguida é feita uma breve exposição das mudanças ocorridas, nos últimos anos, nos quadros social, tecnológico e econômico-mercadológico dessa atividade produtiva, apresentando-se como a Construção Civil Brasileira e as empresas do setor comportaram-se diante da mudança de contexto.

Observa-se que no quadro econômico-mercadológico a mudança de contexto promove a programação da produção a uma atividade de importância estratégica para a empresa, demonstrando a necessidade de se desenvolver ferramentas de planejamento, programação e controle mais adequadas ao processo construtivo e à nova realidade que se apresenta.

Em seguida, para preparar terreno para a exposição do método de programação desenvolvido, o trabalho apresenta e discute as principais técnicas de planejamento e programação de obras, procurando enfatizar a adequação ou não de cada técnica ao processo construtivo e, mais especificamente, à execução de projetos lineares.

A utilidade dos modelos formais de pesquisa operacional no auxílio à tomada de decisões de planejamento e programação de projetos é também discutida. Observa-se que a característica comum desse tipo de problema é que seu processo de solução pode ser formalizado. Isso significa que, dadas as informações necessárias e critérios eficientes, um procedimento pode ser desenvolvido para conduzir a uma solução ótima ou, no mínimo, satisfatória.

A apresentação do método de programação desenvolvido inicia-se -se com a exposição do modelo matemático que lhe dá suporte. O modelo matemático, cerne do método proposto, é apresentado e discutido, tentando-se demonstrar como sua formulação incorpora as variáveis de programação da Linha de Balanço e as restrições financeiras do mercado.

A partir de um programa de execução genérico, o modelo de otimização obtém uma alternativa de programação que se apresenta mais promissora na tentativa de compatibilizar os fluxos de recursos financeiros disponíveis e necessários à execução do projeto.

O fluxo de recursos disponíveis é determinado pela capacidade de pagamento do mercado, pelas regras de desembolso do Sistema Financeiro de Habitação e/ou pela estratégia de administração do capital de giro da empresa e o fluxo de recursos necessários é conseqüência da estratégia de execução do projeto.

A interface do método de programação proposto, representada pelo sistema computacional desenvolvido, é didaticamente apresentada através da simulação um projeto-exemplo. Sob a forma de um manual de utilização, navega-se por todos os caminhos possíveis dentro do sistema, mostrando todas as funções que ele oferece.

Apesar de algumas limitações do modelo matemático, a validação do sistema no Capítulo 7 demonstra não apenas a sua qualidade operacional, mas também todo o potencial de utilização prática que ele proporciona.

Também, em termos de tempo de resposta, o sistema é muito promissor, economizando tempo e possibilitando, através de várias simulações, encontrar a estratégia de execução mais adequada para cada projeto, em função das limitações de ordem física e financeira que se apresentarem ao programador.

8.2- Conclusões Específicas

Neste item são compiladas as principais características e limitações do modelo matemático sugerido, da heurística e do sistema computacional desenvolvido, nesta ordem.

Sobre o ***modelo matemático*** poder-se-ia dizer que:

- é um modelo básico, isto é, suas restrições são apenas suficientes para incorporar o conceito de programação da Linha de Balanço e construir a curva de recursos necessários;

- constitui-se num problema de programação não linear não convexo, com muitos máximos e mínimos locais. Esta característica obriga a se fazer uma exploração inicial no espaço de soluções viáveis, para se obter um ponto de partida que conduza a uma solução final satisfatória;

- o programa de execução de cada unidade repetitiva é definido a partir de uma rede unitária linear, ou seja, parte-se do princípio que as unidades são construídas com a execução em série de todas as atividades constantes do projeto;

- para limitar o porte do modelo e impedir que o tempo de resposta do sistema cresça muito, o número máximo de atividades da rede unitária é fixado em oito (7). Esse não é exatamente o limite viável do número de atividades na rede, mas considera-se que sete grandes marcos intermediários são mais que suficientes para se obter um plano tático para projetos lineares;

- considera que os ritmos de execução de cada atividade da rede unitária são iguais ao ritmo de construção do projeto, ou seja, incorpora o conceito da Linha de Balanço apenas na sua forma paralela - Programação Paralela;

- e finalmente, o modelo básico foi construído para programar somente projetos de unidades unifamiliares - casas.

Cabe salientar que a programação de outros tipos de projetos lineares - estradas, redes de água e drenagem, edifícios altos e conjunto de blocos de apartamentos - pode ser efetuada a partir da adaptação do modelo matemático básico às particularidades de cada um desses projetos.

Sobre a ***heurística*** desenvolvida observa-se que:

- a qualidade da resposta é altamente dependente do parâmetro tolerância. É importante uma escolha criteriosa do valor da tolerância, pois este parâmetro praticamente define o

posicionamento de cada atividade da rede unitária, ou melhor, os valores iniciais para as datas de início de cada atividade;

- o algoritmo fixa a data de início da primeira atividade em zero, impedindo que esta atividade procure um melhor posicionamento como as demais atividades da rede unitária. Essa deficiência fica evidente quando o custo da primeira atividade é muito superior aos demais e as primeiras parcelas da curva de recursos disponíveis são muito baixas. A introdução de uma primeira atividade fictícia certamente resolveria este problema.

Pode-se dizer também que o algoritmo obtém soluções apenas satisfatórias para cada ritmo testado, à medida que determina o posicionamento da atividade assim que encontra uma posição que satisfaça seu critério de julgamento, ou seja, melhorar o somatório acumulado do vetor D de diferenças.

Sobre o ***sistema computacional*** construído a partir do modelo matemático pode-se dizer que:

- produz soluções finais que apresentam muita variabilidade em função dos dados de entrada. Pode-se dizer que o modelo matemático que lhe dá suporte é muito sensível a pequenas variações nos dados de entrada. O motivo, como já foi exposto antes, é um espaço viável pouco restritivo e os incontáveis máximos e mínimos locais que o modelo matemático apresenta;

- os resultados apresentados na experimentação demonstram que o sistema, apesar da variabilidade, produz soluções bastante satisfatórias, conseguindo, quando possível, ajustar a curva de recursos necessários ao fluxo de recursos disponíveis;

- a utilidade prática do sistema é representada pela produção de um programa de execução agregado, onde ele define os valores das variáveis de programação que conduzem à

melhor compatibilização entre os fluxos de recursos financeiros;

- tanto sua qualidade como seu tempo de resposta são altamente atraentes, podendo-se concluir que o sistema é viável;

- o tempo de resposta do sistema estimula o usuário a simular o mesmo projeto para quaisquer alternativas de fluxo de recursos disponíveis que possam se apresentar, permitindo à empresa uma programação financeira muito mais criteriosa.

8.3- Sugestões para Futuros Trabalhos

Antes de se sugerir outras temas de pesquisa serão expostas mais uma vez as principais limitações deste trabalho:

PRIMEIRA - o sistema desenvolvido limita-se a programar, somente, projetos de unidades unifamiliares - casas;

SEGUNDA - todas as atividades da rede unitária têm que ser programadas com o mesmo ritmo de execução;

TERCEIRA - a rede unitária é linear e com, no máximo, oito (7) atividades.

Explorando as limitações que este trabalho apresenta e os temas afins que não puderam ser cobertos por ele, sugere-se:

- realizar maior experimentação do modelo matemático para descobrir o limite viável para o número de atividades da rede unitária. Entende-se por limite viável, o número máximo de atividades que conduz a um tempo de resposta do sistema comparavelmente menor que o processo manual, ou seja, a execução manual da tarefa de planejamento e programação do projetos;

- para um maior número de atividades, obtido na experimentação acima, estruturar o modelo matemático para representar uma rede unitária contendo atividades paralelas;

- alterar o modelo matemático para permitir que cada atividade possa ser programada com seu próprio ritmo. Considerando-se que o modelo permanecerá com muitos máximos e mínimos locais e que a heurística continuará a fornecer um ritmo inicial igual para todas as atividades, o programa de execução obtido não será muito diferente do anterior, mas provavelmente irá melhorar o valor da função objetivo;

- sobre a heurística, também não se esgotou o limite viável do número de ritmos testados pelo algoritmo. Também é possível obter-se melhores resultados, pesquisando um valor mais adequado para tolerância. Depois de explorado todo o potencial que essa heurística oferece, poder-se-á então pensar no desenvolvimento de outra heurística que conduza o sistema a resultados melhores;

- adaptar o modelo básico para permitir a programação de outros tipos de projetos lineares;

- automatizar a integração do programa de execução das atividades repetitivas - Linha de Balanço - à programação das atividades não repetitivas do projeto;

- desenvolver, para o método de programação proposto, a função ***controle***, tornando esta ferramenta de auxílio, uma ferramenta dinâmica, capaz de adaptar-se às mudanças - físicas e financeiras - ocorridas durante o processo construtivo;

- para tornar a função objetivo do modelo matemático mais realista, definir um critério para dar pesos diferentes aos desvios positivos e negativos, verificados entre os valores das curvas de agregação de recursos do projeto;

- crescendo em profundidade, pesquisar outras funções objetivo que possam traduzir, a nível tático, as estratégias de ordem tecnológica, econômica e financeira da empresa, incorporando também aspectos de qualidade.

Para concluir este trabalho, demonstrando a necessidade de mais pesquisas na Construção Civil, transcreve-se uma frase de Rocha Lima Junior (1987), " somente a união e o confronto das especulações de caráter acadêmico com a problemática específica encontrada pelas empresas na aplicação dos sistemas, no gerenciamento de empreendimentos, pode superar tais deficiências e levar o processo avante com a qualidade que se exige ".

REFERÊNCIAS BIBLIOGRÁFICAS

BALARINE, Oscar F. O. **Administração e finanças para construtores e incorporadores**. Porto Alegre: EDIPUCRS, 1990. p.115

_____. _____. p.193

BATTERSBY, Albert. **Network analysis for planning and scheduling**. 3.ed. London: Macmillar, 1971. 332p.

CARDOSO, Francisco Ferreira. Novos enfoques sobre a gestão de produção. Como melhorar o desempenho das empresas de construção civil. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 1993, São Paulo. **Anais...** São Paulo, 1993. v.2, p.557-569

CARR, Robert I, Meyer W. L. Planning construction of repetitive building units. **Journal of the Construction Division**, v.100, n.CO3, p.403-412, Sept.1974

CLAURE, Jorge Eduardo Z. **Otimização de projetos lineares em construção civil através do método espaço-tempo**. Florianópolis: UFSC, 1986, 73p. Dissertação - (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, 1986

FARAH, Marta Ferreira. Estratégias empresariais e mudanças no processo de trabalho na construção habitacional no Brasil. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 1993, São Paulo. **Anais...** São Paulo, 1993. v.2, p.581-590

FORMOSO, C. T. **A knowledge based framewok for planning house building projects**. London: University of Salford, 1991. 327p. Thesis

FUKS, Saul. Limitações dos modelos formais da pesquisa operacional: uma visão epistemológica. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 1985, São José dos Campos. **Anais...** São José dos Campos, 1985. p.34-45

HEINECK, L. F. Modelos para o planejamento de obras. In: II ENCONTRO DE PESQUISA OPERACIONAL DO RIO GRANDE DO SUL, 1984, Santa Maria

_____. **Curvas de agregação de recursos no planejamento e controle da edificação - aplicações a obras e a programas de construção**, Porto Alegre, UFRGS, 1989. 49p.

LUTZ, James, HIJAZI, Adib. Planning repetitive construction: current practice. **Construction Management and Economics**, n.11, p.99-110, 1993

MASCARÓ, Lúcia, MASCARÓ, Juan Luis. **A construção na economia nacional**. 2.ed. São Paulo: Editora Pini Ltda, 1981. 112p.

MAZIERO, Lucia Teresinha Peixe. **Aplicação do conceito do método da linha de balanço no planejamento de obras repetitivas. Um levantamento das decisões fundamentais para sua aplicação**. Florianópolis: UFSC, 1990, 147p. Dissertação - (Mestrado em Engenharia) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 1990

NEVES, Maria B. S., GUEDERT, L.O. Programação sujeita às restrições externas. In: CONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 1993, São Paulo. **Anais...** São Paulo, 1993. v.2, p.547-556

O'BRIEN, James. Line of Balance. In: _____. **Scheduling Handbook**. New York: McGraw-Hill, 1969. 605p. Cap.10, p.246-255

PERERA, Srilal. Resource sharing in linear construction. **Journal of Construction Engineering and Management**, v.109, n.1, p.102-111, Mar.1983

PILCHER, Roy. **Principles of construction management**. 2.ed. England: McGraw-Hill, 1976. 370p.

REDA, Rehab M. RPM: Repetitive project modeling. **Journal of Construction Engineering and Management**, v.116, n.2, p.316-330, Jun.1990

ROCHA LIMA JUNIOR, João da. O gerenciamento na construção civil - uma visão sistêmica. **Revista Politécnica**, São Paulo, n.195, p.10, Mar.1987

SCOMAZZON, Beatriz et al. **Programação e controle de obras repetitivas - técnica da linha de balanço**. Porto Alegre: UFRGS, 1985

WARSZAWSKI, A. Decision models and expert systems in construction management. **Building and Environment**, Technion, v.20, n.4, p.201-210, 1985

WERNA, Edmundo. The concomitant evolution and stagnation of Brazilian building industry. **Construction Management and Economics**, n.11, p.194-202, 1993

BIBLIOGRAFIA

1- ANTILL, James M., WOODHEAD, Ronald W. **CPM aplicado às construções**. Rio de Janeiro: Livros Técnicos e Científicos, 1971. 301p.

2- BARRIE, Donald S.,PAULSON, Boyd C. **Professional construction management**. 2.ed. New York: McGraw-Hill, 1984. 453p.

3- BIRREL, George S. Construction planning - beyond the critical path. **Journal of the Construction Division**, v.106, n.CO3, p.389-407, Sept.1980

4- BOBROFF, Jacotte. The project management: a new profile for the actors in the building industry. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 1993, São Paulo. **Anais...** São Paulo, 1993. v.1, p.41-51

5- BROOKE, A. et al. **GAMS. A user's guide**. The Scientific Press, 1988. 289p.

6- CAMPAGNAC, Elisabeth. Mutations des marchés et évolutions des systèmes de production et de travail le Bâtiment en France et en Europe. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 1993, São Paulo. **Anais...** São Paulo, 1993. v.1, p.27-40

7- DIGMAN, L.A. PERT/LOB: life-cycle technique. **The Journal of Industrial Engineering**, v.18, n.2, p.154-158, Feb.1967

8- FISCHER, Sérgio. **A indústria da construção: uma análise econômico-financeira**. Porto Alegre: Fundação de Economia e Estatística, 1984. 175p.

9- GATES, Marvin. Learning and experience curves. **Journal of the Construction Division**, v.98, n.CO1, p.79-101, Mar.1972

10- GOLDMAN, Pedrinho. **Introdução ao planejamento e controle de custos na construção civil**. 2.ed. São Paulo: Editora Pini Ltda, 1986. 125p.

11- HEINECK, L. F., RAWCLIFFE, J. A graphical picture of what really happens on site. In: SYMPOSIUM ON ORGANIZATION AND MANAGEMENT OF CONSTRUCTION, 1984, Ontario

12- JAAFARI, Ali. Criticism of CPM for project planning analysis. **Journal of Construction Engineering and Management**, v.110, n.2, p.222-233, June 1984

13- LEVITT, Harry P. Computed line of balance technique. **The Journal of Industrial Engineering**, v.19, n.2, p.61-66, Feb.1968

14- MADERS, Berenice. **Técnica de programação e controle da construção repetitiva - linha de balanço: estudo de caso de um conjunto habitacional**. Porto Alegre: UFRGS, 1987. 175p. Dissertação - (Mestrado em Engenharia Civil) - Universidade Federal do Rio Grande do Sul, 1987

15- MURGEL, Sérgio R. **Planejamento e gerência de empreendimentos (conceitos e instrumentos) programação e controle de obras**. São Paulo: USP, 1981. 121p. Dissertação - (Mestrado em Engenharia de Produção) - Universidade de São Paulo, 1981

16- THE NATIONAL BUILDING AGENCY. **Industrialised two-storey housing - A productivity study**. London, 1970. 35p.

17- O'BRIEN, James. VPM scheduling for high-rise buildings. **Journal of the Construction Division**, v.101, n.CO4, p.895-905, Dec.1975

18- PEER, Shlomo. Network analysis and construction planning. **Journal of the Construction Division**, v.100, n.CO3, p.203-211, Sept.1974

19- _____. Application of cost-flow forecasting models. **Journal of the Construction Division**, v.108, n.CO2, p.226-232, Jun.1982

20- PEER, Shlomo, SELINGER, Shlomo. CPT - New Approach to Construction Planning. In: SYPOSIUM ON ORGANIZATION AND MANAGEMENT OF CONSTRUCTION, 1976, Washington DC. **Proceeding...**Washington: CIB, 1976. p156-163

21- PETERMAN, G. G. A way to forecast cash flow. **World Construction**, p.17-22, Oct.1973

22- PULTAR, M. Progress-based construction scheduling. **Journal of Construction Engineering and Management**, v.116, n.4, p.670-688, Dec.1990

23- REINSCHMIDT, K., FRANK, W. Construction cash flow management system. **Journal of the Costruction Division**, v.102, n.CO4, p.615-627, Dec.1976

24- ROCHA LIMA JUNIOR, João da. Sistemas de informação para o planejamento na construção civil - gênese e informatização. **Boletim Técnico da EPUSP**, São Paulo, n.26/90, p. 1-69, 1990

25- RUSSEL, A. H. Cash flows in networks. **Management Science**, v.16, n.5, p.357-373, Jan.1970

26- SARRAJ, Zohair. Formal development of line-of-balance technique. **Journal of Construction Engineering and Management**, v.116, n.4, p.689-704, Dec.1990

27- SILVA, M. Angélica Covelo. **Identificação e análise dos fatores que afetam a produtividade sob a ótica dos custos de produção de empresas de edificação**. Porto Alegre: UFRGS, 1986, 295p. Dissertação - (Mestrado em Engenharia Civil) - Universidade Federal do Rio Grande do Sul, 1986

28- STEVENS, James D. Modified CPM - a scheduler's best friend. **Cost Engineering**, v.30, n.10, p.9-11, Oct. 1988

29- TOKUNAGA, N. et al. Construction planning and scheduling for civil engineering works. **Mitsubishi Heavy Industries Ltd**, v.21, n.2, p.107-112. Jun.1984

30- TURBAN, Efraim. The line of balance - a management by exception tool. **The Journal of Industrial Engineering**, v.19, n.9, p.440-448, Sept.1968

ANEXO I

Quadro I.1(a) - Cronograma Financeiro do Projeto-Exemplo

DISCRIMINAÇÃO DOS SERVIÇOS	1° MÊS		2° MÊS		3° MÊS	
	% Item	% Total	% Item	% Total	% Item	% Total
Casas de 30m²	0.00		2.00	1.25	5.00	3.13
Abastec.to de Água	0.00		0.00		0.00	
Esgoto Sanitário	0.00		0.00		0.00	
Energia Elétrica	0.00		0.00		0.00	
Drenagem Pluvial	65.00	2.41	35.00	1.31	0.00	
Centro Comunitário	0.00		15.00	0.32	28.00	0.60
TOTAL PREVISTO		2.41		2.88		3.73
TOTAL ACUM.		2.41		5.29		9.02

Quadro I.1(b) - Cronograma Financeiro do Projeto-Exemplo

DISCRIMINAÇÃO DOS SERVIÇOS	4° MÊS		5° MÊS		6° MÊS	
	% Item	% Total	% Item	% Total	% Item	% Total
Casas de 30m²	11.00	6.89	13.00	8.15	14.00	8.77
Abastec.to de Água	0.00		0.00		0.00	
Esgoto Sanitário	0.00		0.00		0.00	
Energia Elétrica	0.00		0.00		0.00	
Drenagem Pluvial	0.00		0.00		0.00	
Centro Comunitário	45.00	0.96	0.00		0.00	
TOTAL PREVISTO		7.85		8.15		8.77
TOTAL ACUM.		16.87		25.02		33.79

Quadro I.1(c) - Cronograma Financeiro do Projeto-Exemplo

DISCRIMINAÇÃO DOS SERVIÇOS	7° MÊS		8° MÊS		9° MÊS	
	% Item	% Total	% Item	% Total	% Item	% Total
Casas de 30m²	**13.00**	**8.15**	**14.00**	**8.77**	**10.00**	**6.27**
Abastec.to de Água	0.00		0.00		33.00	1.15
Esgoto Sanitário	0.00		0.00		33.00	6.23
Energia Elétrica	0.00		0.00		0.00	
Drenagem Pluvial	0.00		0.00		0.00	
Centro Comunitário	0.00		0.00		0.00	
TOTAL PREVISTO		8.15		8.77		13.65
TOTAL ACUM.		41.94		50.71		64.36

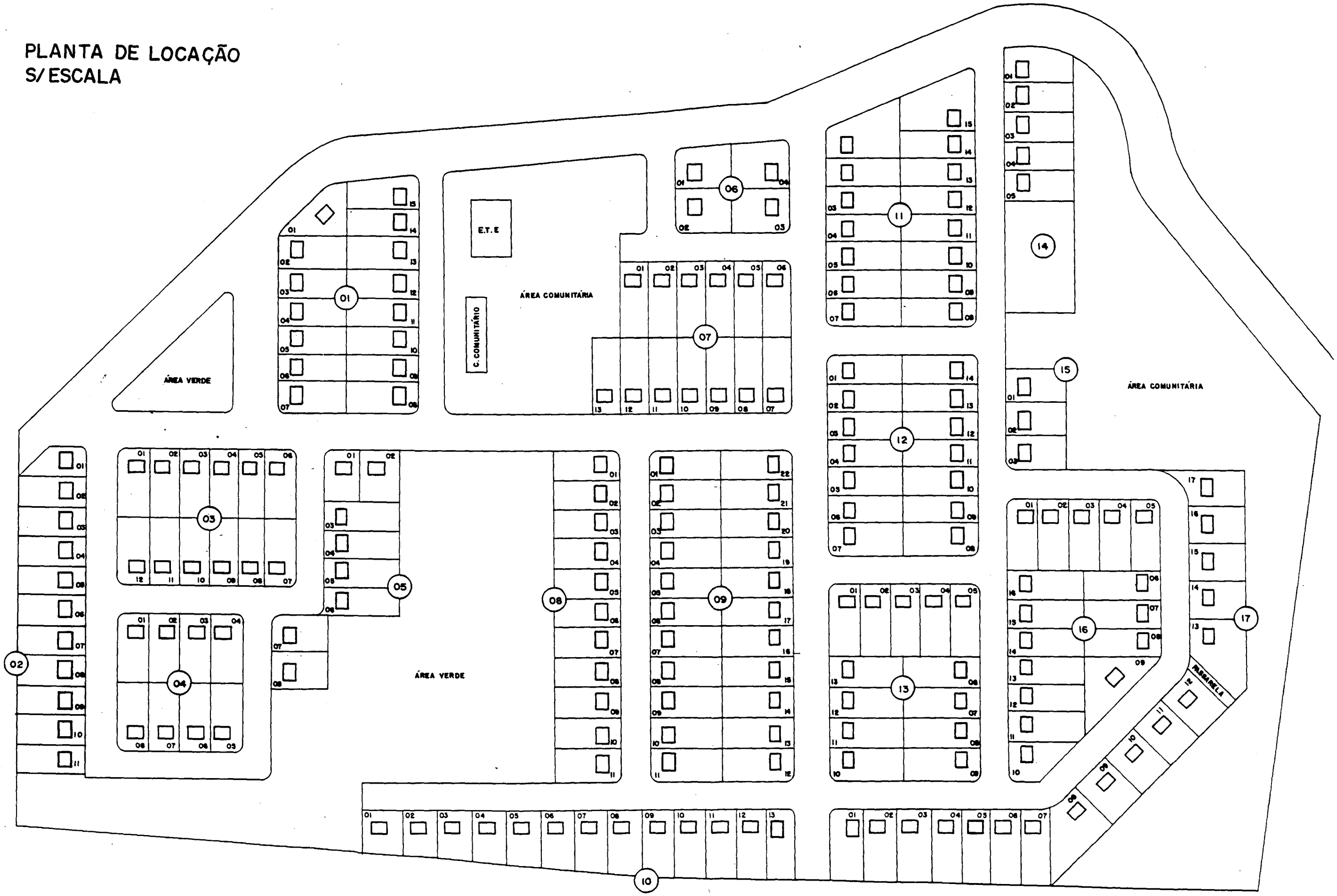
Quadro I.1(d) - Cronograma Financeiro do Projeto-Exemplo

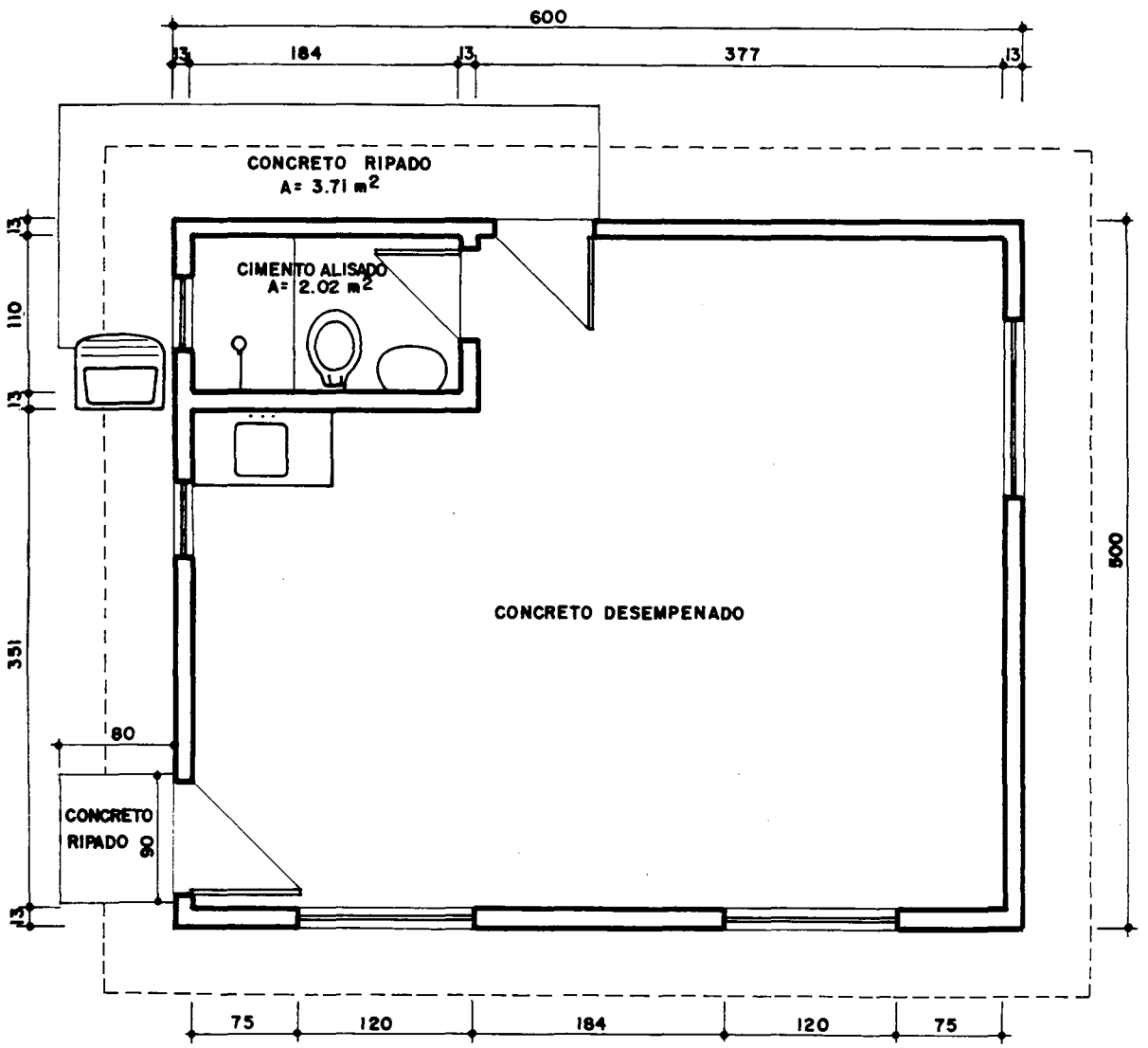
DISCRIMINAÇÃO DOS SERVIÇOS	10° MÊS		11MÊS		12MÊS		TOTAL
	% Item	% Total	% Item	% Total	% Total	% Item	
Casas de 30m²	**8.00**	**5.01**	**7.00**	**4.39**	**3.00**	**1.88**	**62.66**
Abastec.to de Água	33.00	1.16	33.00	1.16	0.00		3.47
Esgoto Sanitário	34.00	6.43	33.00	6.23	0.00		18.89
Energia Elétrica	25.00	2.28	50.00	4.57	25.00	2.28	9.13
Drenagem Pluvial	0.00		0.00		0.00		3.72
Centro Comunitário	0.00		0.00		12.00	0.25	2.13
TOTAL PREVISTO		14.88		16.35		4.41	
TOTAL ACUM.		79.24		95.59		100.00	100.00

Quadro I.2 - Os Custos de cada Atividade: COHAB/SISTEMA

C O H A B		S I S T E M A	
SERVIÇO	% ITEM	SERVIÇO	% ITEM
1- Serviços Iniciais	2.26	1- Fundações	3.12
2- Fundações	0.86	2- Estrut ./ Alvenaria / Cobertura	29.19
3- Alvenaria e Estrutura	19.13	3- Tubulações	7.55
4- Cobertura	10.06	4- Batentes e Contram. / Revest. / Pisos	27.88
5- Tubulações	7.55	5- Forro / Esquadrias / Ferrag. / Vidros	6.76
6- Batentes e Janelas	8.19	6- Pintura	7.13
7- Chapisco	1.93	7- Fiação e Aparelhos / Compl. / Limpeza	18.37
8- Reboco	11.93		
9- Pisos e Pavimentos	5.83		
10- Vidros	1.53		
11- Forro e Portas	5.23		
12- Instalações Elétricas	7.75		
13- Pintura	7.13		
14- Aparelhos	7.92		
15- Complementos e Limpeza	1.62		
16- Habite-se	1.08		

PLANTA DE LOCAÇÃO
S/ESCALA





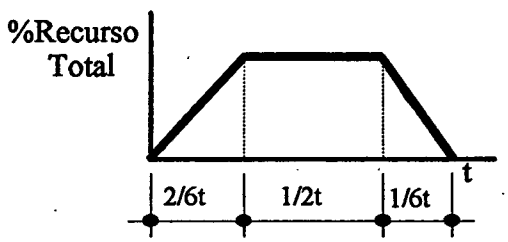
PLANTA BAIXA

ESCALA: 1/50

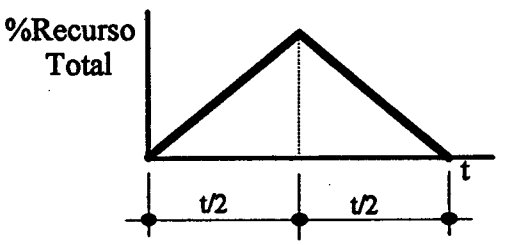
ANEXO II

1.1- Curvas de Agregação de Recursos Disponíveis

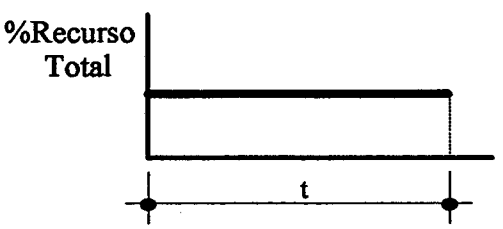
A = Curvas Trapezoidais



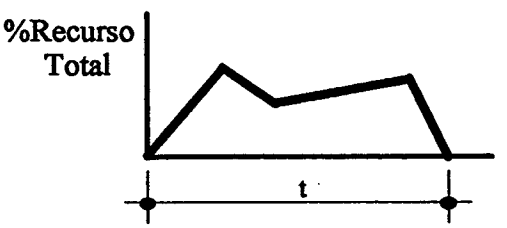
B = Curvas Triangulares



C = Curvas Retangulares



D = Curvas Irregulares



RAM = Curvas Ramdômicas

IDEAL = Curvas *Ideais*

REAL = Curva Real

Quadro II.1 - Projetos com Doze Parcelas

Nº DA PARCELA	VALORES (%)						
	A	B	C	D	RAM	IDEAL	REAL
1	2.22	2.19	8.33	7.00	5.00	3.42	0.00
2	4.44	4.39	8.33	6.00	6.00	8.50	2.00
3	6.67	6.59	8.33	7.00	10.00	9.52	5.00
4	8.89	8.79	8.33	8.00	6.00	9.52	11.00
5	11.11	10.99	8.34	9.00	3.00	9.52	13.00
6	11.11	13.19	8.34	10.00	6.00	9.52	14.00
7	11.11	15.38	8.34	11.00	11.00	9.52	13.00
8	11.11	12.81	8.34	10.00	15.00	9.52	14.00
9	11.11	10.26	8.33	9.00	9.00	9.52	10.00
10	11.11	7.70	8.33	8.00	10.00	9.52	8.00
11	7.41	5.14	8.33	8.00	2.00	8.50	7.00
12	3.71	2.57	8.33	7.00	17.00	3.42	3.00

Quadro II.2 - Projetos com Dezoito Parcelas

Nº DA PARCELA	VALORES (%)					
	A	B	C	D	RAM	IDEAL
1	1.06	1.06	5.55	9.00	8.00	1.44
2	2.13	2.12	5.55	8.00	5.00	3.33
3	3.19	3.18	5.55	7.00	2.00	5.25
4	4.25	4.24	5.55	6.00	7.00	6.66
5	5.31	5.25	5.56	5.00	9.00	6.66
6	6.36	6.30	5.56	4.00	3.00	6.66
7	7.40	7.35	5.56	3.00	9.00	6.66
8	7.40	8.40	5.56	4.00	10.00	6.66
9	7.40	9.45	5.56	5.00	4.00	6.66
10	7.40	10.53	5.56	6.00	3.00	6.66
11	7.40	9.36	5.56	7.00	2.00	6.66
12	7.40	8.19	5.56	8.00	3.00	6.66
13	7.40	7.02	5.56	9.00	5.00	6.66
14	7.40	5.85	5.56	10.00	10.00	6.66
15	7.40	4.68	5.55	3.00	11.00	6.66
16	5.55	3.51	5.55	3.00	3.00	5.25
17	3.70	2.34	5.55	2.00	4.00	3.33
18	1.85	1.17	5.55	1.00	2.00	1.44

Quadro II.3 - Projetos com Vinte e Quatro Parcelas

Nº DA PARCELA	VALORES (%)					
	A	B	C	D	RAM	IDEAL
1	0.62	0.62	4.16	6.00	6.00	0.80
2	1.24	1.23	4.16	6.00	7.00	1.60
3	1.86	1.85	4.16	5.00	5.00	2.39
4	2.48	2.46	4.16	5.00	4.00	3.18
5	3.10	3.08	4.17	4.00	3.00	3.97
6	3.72	3.69	4.17	4.00	2.00	4.76
7	4.33	4.31	4.17	3.00	7.00	5.55
8	4.95	4.92	4.17	3.00	9.00	5.55
9	5.55	5.54	4.17	2.00	3.00	5.55
10	5.55	6.15	4.17	2.00	6.00	5.55
11	5.55	6.77	4.17	1.00	8.00	5.55
12	5.55	7.38	4.17	1.00	3.00	5.55
13	5.55	8.00	4.17	2.00	3.00	5.55
14	5.55	7.33	4.17	2.00	3.00	5.55
15	5.55	6.67	4.17	3.00	5.00	5.55
16	5.55	6.00	4.17	3.00	4.00	5.55
17	5.55	5.33	4.17	4.00	7.00	5.55
18	5.55	4.67	4.17	5.00	4.00	5.55
19	5.55	4.00	4.17	5.00	3.00	4.76
20	5.55	3.33	4.17	6.00	2.00	3.97
21	4.44	2.67	4.16	6.00	1.00	3.18
22	3.33	2.00	4.16	7.00	1.00	2.39
23	2.22	1.33	4.16	7.00	2.00	1.60
24	1.11	0.67	4.16	8.00	2.00	0.80

Quadro II.4 - Projetos com Trinta Parcelas

Nº DA PARCELA	VALORES (%)					
	A	B	C	D	RAM	IDEAL
1	0.41	0.41	3.33	2.00	4.00	0.60
2	0.82	0.82	3.33	2.00	5.00	1.20
3	1.22	1.21	3.33	2.00	1.00	1.78
4	1.62	1.61	3.33	3.00	1.00	2.39
5	2.03	2.02	3.33	3.00	6.00	2.97
6	2.43	2.42	3.33	3.00	2.00	3.57
7	2.84	2.82	3.33	4.00	1.00	4.17
8	3.24	3.23	3.33	4.00	1.00	4.17
9	3.64	3.63	3.33	4.00	5.00	4.17
10	4.05	4.03	3.33	5.00	2.00	4.17
11	4.44	4.43	3.34	5.00	3.00	4.17
12	4.44	4.84	3.34	5.00	2.00	4.17
13	4.44	5.24	3.34	6.00	1.00	4.17
14	4.44	5.64	3.34	6.00	1.00	4.17
15	4.44	6.05	3.34	6.00	1.00	4.17
16	4.44	6.45	3.34	5.00	2.00	4.17
17	4.44	6.02	3.34	5.00	4.00	4.17
18	4.44	5.59	3.34	5.00	6.00	4.17
19	4.44	5.16	3.34	4.00	8.00	4.17
20	4.44	4.73	3.34	4.00	10.00	4.17
21	4.44	4.30	3.33	4.00	3.00	4.17
22	4.44	3.87	3.33	3.00	3.00	4.17
23	4.44	3.44	3.33	2.00	8.00	4.17
24	4.44	3.01	3.33	2.00	5.00	4.17
25	4.44	2.58	3.33	1.00	5.00	3.57
26	3.70	2.15	3.33	1.00	2.00	2.97
27	2.96	1.72	3.33	1.00	2.00	2.39
28	2.22	1.29	3.33	1.00	2.00	1.78
29	1.48	0.86	3.33	1.00	2.00	1.20
30	0.74	0.43	3.33	1.00	2.00	0.60

Quadro II.5 - Projetos com Trinta e Seis Parcelas

Nº DA PARCELA	VALORES (%)					
	A	B	C	D	RAM	IDEAL
1	0.29	0.29	2.77	5.00	1.00	0.48
2	0.58	0.58	2.77	5.00	1.00	0.95
3	0.86	0.86	2.77	5.00	1.00	1.36
4	1.15	1.15	2.77	5.00	5.00	1.90
5	1.43	1.43	2.78	4.00	7.00	2.38
6	1.72	1.71	2.78	4.00	2.00	2.86
7	2.00	1.99	2.78	4.00	2.00	3.33
8	2.29	2.28	2.78	4.00	3.00	3.33
9	2.57	2.56	2.78	3.00	1.00	3.33
10	2.86	2.84	2.78	3.00	9.00	3.33
11	3.14	3.13	2.78	3.00	4.00	3.33
12	3.42	3.41	2.78	3.00	3.00	3.33
13	3.70	3.70	2.78	2.00	6.00	3.33
14	3.70	3.98	2.78	2.00	1.00	3.33
15	3.70	4.27	2.78	2.00	1.00	3.33
16	3.70	4.55	2.78	2.00	1.00	3.33
17	3.70	4.84	2.78	1.00	1.00	3.33
18	3.70	5.12	2.78	1.00	6.00	3.33
19	3.70	5.41	2.78	1.00	2.00	3.33
20	3.70	5.10	2.78	1.00	8.00	3.33
21	3.70	4.80	2.78	1.00	8.00	3.33
22	3.70	4.50	2.78	1.00	2.00	3.33
23	3.70	4.20	2.78	1.00	2.00	3.33
24	3.70	3.90	2.78	1.00	2.00	3.33
25	3.70	3.60	2.78	2.00	1.00	3.33
26	3.70	3.30	2.78	2.00	2.00	3.33
27	3.70	3.00	2.78	2.00	7.00	3.33
28	3.70	2.70	2.78	2.00	1.00	3.33
29	3.70	2.40	2.78	3.00	1.00	3.33
30	3.70	2.10	2.78	3.00	1.00	3.33
31	3.17	1.80	2.78	3.00	1.00	2.86
32	2.64	1.50	2.78	3.00	1.00	2.38
33	2.11	1.20	2.77	4.00	1.00	1.90
34	1.59	0.90	2.77	4.00	1.00	1.43
35	1.06	0.60	2.77	4.00	2.00	0.95
36	0.52	0.30	2.77	4.00	2.00	0.48

1.2- Custo de cada Atividade

Quadro II.6(a) - Custos Percentuais das Atividades da Rede Unitária

ATIVIDADES	CUSTOS IGUAIS	CUSTOS DIFERENTES		
		1ª Posição	4ª Posição	7ª Posição
1-Fundações	14.29	40.00	10.00	10.00
2-Estrut./Alven./Cobertura	14.29	10.00	10.00	10.00
3-Tubulações	14.29	10.00	10.00	10.00
4-Batentes e Contram./Revest./Pisos	14.29	10.00	40.00	10.00
5-Forro/Esquadrias/Ferrag./Vidros	14.28	10.00	10.00	10.00
6-Pintura	14.28	10.00	10.00	10.00
7-Fiação e Aparelhos/Compl./Limpeza	14.28	10.00	10.00	40.00

Quadro II.6(b) - Custos Percentuais das Atividades da Rede Unitária

ATIVIDADES	RAM	IDEAL	REAL
1-Fundações	3.00	14.28	3.12
2-Estrut./Alven./Cobertura	30.00	14.28	29.19
3-Tubulações	8.00	14.28	7.55
4-Batentes e Contram./Revest./Pisos	28.00	14.29	27.88
5-Forro/Esquadrias/Ferrag./Vidros	7.00	14.29	6.76
6-Pintura	6.00	14.29	7.13
7-Fiação e Aparelhos/Compl./Limpeza	18.00	14.29	18.37

1.3- Durações e Precedências Mínimas entre as Atividades

Quadro II.7(a) - Durações e Precedências Mínimas da Rede Unitária

ATIVIDADES	RAM		REAL	
	DURAÇÃO (Dia Útil)	FOLGA MÍNIMA	DURAÇÃO (Dia Útil)	FOLGA MÍNIMA
1-Fundações	3.00	—	3.00	—
2-Estrut./Alven./Cobertura	6.00	2.00	8.50	1.00
3-Tubulações	8.00	2.00	2.00	0.00
4-Batentes e Contram./Revest./Pisos	4.00	2.00	4.00	0.00
5-Forro/Esquadrias/Ferrag./Vidros	9.00	2.00	2.33	0.00
6-Pintura	5.00	2.00	2.00	0.00
7-Fiação e Aparelhos/Compl./Limpeza	6.00	2.00	0.83	0.00

Quadro II.7(b) - Durações e Precedências Mínimas da Rede Unitária

ATIVIDADES	DEMAIS PROJETOS	
	DURAÇÃO (Dia Útil)	FOLGA MÍNIMA
1-Fundações	1.00	—
2-Estrut./Alven./Cobertura	1.00	0.00
3-Tubulações	1.00	0.00
4-Batentes e Contram./Revest./Pisos	1.00	0.00
5-Forro/Esquadrias/Ferrag./Vidros	1.00	0.00
6-Pintura	1.00	0.00
7-Fiação e Aparelhos/Compl./Limpeza	1.00	0.00

1.4- Outros Dados

Quadro II.8 - Número de Unidades Construtivas e Ritmo Máximo Permitido

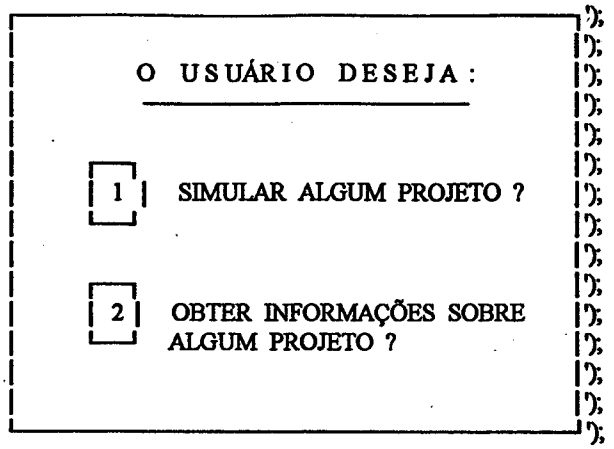
PROJETO	Nº UNIDADES	RITMOmáx (Un./Dia)
12 Parcelas	200	6
18 Parcelas	300	6
24 Parcelas	400	6
30 Parcelas	500	6
36 Parcelas	600	6

APÊNDICE

```
PROGRAM SISTEMA;

{$M 8192,0,60000}

uses dos,crt,printer,graph;

const
 NEV = 8;
 dd = 3;
 DI = 0;
 NomeAtiv : array[1..(NEV-1)] of string = ('Fundação','Estr/Alv/Cob','Tubulações','Revest/Pisos',
                                           'Esq/Vidros','Pintura','Serv.Compl.');

Type
 ptela  = ^tela;
 tela   = array[1..3*16384] of byte;

var

{ Strings e afins }

 tela1                  : ptela;
 arq4                   : file;
 arq0,arq6              : text;
 ch,resposta,aux,sai    : string;
 projeto,projeto_aux    : string;
 str8                   : string[8];

{ Variáveis Inteiras }

 driverVar, modeVar     : integer;  { Variáveis que inicializam o módulo gráfico - INITGRAPH }
 code                   : integer;  { Variável auxiliar p/ conversão de inteiros }
 N,N_aux                : word;  { N° de unidades construtivas }
 cronograma             : byte;  { N° de parcelas da curva de recursos }
 cronograma_aux         : byte;  { N° de parcelas provisório }
 opcao0,opcao,opcao1    : byte;  { Escolher o n° do dado a ser alterado }
 A0,A1,B1               : byte;  { Utilizadas na subrotina que busca a solução inicial }
 j,k,a,x                : byte;  { Utilizadas para iterações }
 i                      : longint;

{ Variáveis Reais }

 RITMOMAX_aux           : real;  { Ritmo máximo permitido pelo usuário - provisório }
 DIFRITMO               : real;  { Intervalo entre cada nível de ritmo - e a razão da P.A. }
 DTOT,DU,RITMO          : real;  { Duração total, duração unitária e ritmo obtidos na execução de modelo.GMS }
 RITMOotm               : real;  { Valor inicial do ritmo obtido na busca exploratória de SOL_INICIAL }
 BUFFERp_SOMA           : real;  { Somatorio dos buffers mínimos }
 SQRDIF                 : real;  { Valor mínimo da f. objetivo para o modelo.GMS }
 SQRDIFmin              : real;  { Valor mínimo da f. objetivo de cada busca da heurística }
 SOMABUFFER             : real;  { Soma dos buffer durante uma busca }
 SOMATIV                : real;  { Somatório das durações de todas as atividades }
 DIFSOMA                : real;  { Somatório definitivo das diferenças entre as curvas de agregação não acumuladas }
 periodo,periodo_aux    : real;  { Periodicidade, confirmada e provisória }
 pc                     : real;  { Calcula cada período, a partir da periocicidade, para construir as telas de dados }
 status                 : real;  { E o status resultante da execução do modelo.GMS }
```

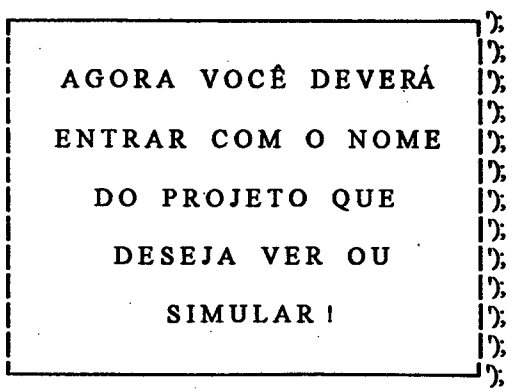
```
PARCELA_MEDIA        : real;  { E o valor médio da curva de recursos disponíveis }
SDesvio              : real;  { Somatório de desvios entre cada PARCELA e a PARCELA_MEDIA }
Desvio_Medio         : real;  { Desvio médio encontrado entre as parcelas da curva de recursos disponíveis }
f_Desvio             : real;  { Define o índice que será multiplicado pela Tol_Padrao em função do Desvio_Medio }
f_PARCELA            : real;  { Define o índice que será multiplicado pela Tol_Padrao em função do n° parcelas }
Tol_Padrao,Tol_Media : real;  { Tolerâncias padrão e média permitidas para os valores negativos de D }
VETORmin             : real;  { E a variável que guarda a melhor solução durante o processo de busca }
SOMADa               : real;  { E a soma de D acumulada ate o posicionamento da atividade "a" }
SOMADa_1             : real;  { E a soma de D acumulada ate o posicionamento da atividade "a-1" }
RITMOi               : array[1..40] of real; { Níveis de ritmos pesquisados no algoritmo que busca a sol. inicial }
DATACRON             : array[1..50] of real; { Vetor de datas (em dias) da curva de recursos disponíveis }
PARCELA              : array[1..50] of real; { Vetor de parcelas não acumuladas confirmadas }
PARCELA_aux          : array[1..50] of real; { Vetor de parcelas não acumuladas provisórias }
Desvio               : array[1..50] of real; { Vetor de desvios entre cada PARCELA e a PARCELA_MEDIA }
SOMA                 : array[1..50] of real; { Vetor de parcelas nao acumuladas da curva da obra }
SOMAY                : array[1..50] of real; { Vetor de parcelas acumuladas da curva da obra }
DE                   : array[1..(NEV-1)] of real; { Datas de início de cada atividade na 1a. unidade construtiva }
DEotm                : array[1..(NEV-1)] of real; { As datas de início geradas na busca de valores iniciais }
DEcedo               : array[1..(NEV-1)] of real; { Datas mais cedo geradas durante a busca de valores iniciais }
DEtarde              : array[1..(NEV-1)] of real; { Datas mais tarde geradas durante a busca de valores iniciais }
DE_aux               : array[1..(NEV-1)] of real; { Datas auxiliares geradas durante a busca de valores iniciais }
DE_bom               : array[1..(NEV-1)] of real; { As datas ótimas para cada RITMOi }
BUFFER               : array[2..(NEV-1)] of real; { Os buffers que produzem o valor mínimo de DIFPERC }
BUFFERp              : array[2..(NEV-1)] of real; { Precedências mínimas exigidas pelo usuário }
BUFFERp_aux          : array[2..(NEV-1)] of real; { Precedências mínimas - valores ainda não confirmados }
dATIV,dATIVaux       : array[1..(NEV-1),2..NEV] of real; { Vetor de duração das atividades em cada unidade }
DT,Ct,Ct_aux         : array[1..(NEV-1),2..NEV] of real; { Datas de término e custos % das atividades }
DURACAO              : array[1..(NEV-1),2..NEV] of real; { Duração total de cada atividade para as N unidades }
D                    : array[1..(NEV-1),1..50] of real;  { As diferenças entre as curvas, calculadas à medida que
                                                           cada atividade é posicionada pela Heurística  }
Y                    : array[1..(NEV-1),2..NEV,1..50] of real; { Custo % acum. de cada ativ. em cada período }


PROCEDURE DESLIGA_CURSOR;  { Evita o aparecimento do cursor. Procedimento utilizado para melhor visualização
                             da tela  gráfica }
begin
  port[$03D4]:=10;
  port[$03D5]:=40;
end;


PROCEDURE LIGA_CURSOR;  { Torna o cursor novamente visível }

begin
  port[$03D4]:=10;
  port[$03D5]:=13;
end;


PROCEDURE USUARIO; { Esta subrotina e a primeira tela a aparecer. Nela o usuário escolhe o que deseja ao rodar
                     o sistema }
begin
  clrscr;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
```

```
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln;
 repeat
  write('          Digite a sua opção : ');
  readln(ch);
  if (ch = '1') or (ch = '2') then
  begin
   val(ch,opcao0,code);
  end
  else
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
  end;
 until (ch = '1') or (ch = '2');
end;
```



PROCEDURE NOME_PROJETO; { Esta subrotina e responsável pela entrada do nome do projeto em questão }

```
begin
 clrscr;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln('
 writeln;
 write('          Nome do Projeto (até 8 dígitos) : ');
 readln(projeto_aux);
 repeat
  if (length(projeto_aux)>8) then
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
```



```
    write('              Digite novamente o Nome do Projeto : ');
    readln(projeto_aux);
   end;
  until (length(projeto_aux)≤8);
  for i:= 1 to length(projeto_aux) do
  begin
   projeto_aux[i]:= UpCase(projeto_aux[i]);
  end;
end;


PROCEDURE N_UNIDADES;  { Esta subrotina faz parte da subrotina ENTRADA_DADOS, sendo responsável
                                          pela entrada do número total de unidades construtivas }
begin
 writeln;
 repeat
  write('[A]  O NÚMERO TOTAL DE UNIDADES A SEREM CONSTRUÍDAS : ');
  readln(aux);
  val(aux,N_aux,code);
  if (code< >0) or (N_aux=1) then
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
   case N_aux of
   1:
    begin
     writeln('ATENÇÃO! A LINHA DE BALANÇO SÓ PROGRAMA PROJETOS');
     write('              COM MAIS DE 1(UMA) UNIDADE CONSTRUTIVA!');
     delay(2500);
     gotoxy(whereX,whereY);
     DelLine;
     gotoxy(whereX,whereY-1);
     DelLine;
     gotoxy(1,whereY);
    end;
   end;
  end;
 until (code=0) and (N_aux< >1);
end;


PROCEDURE RITMO_MAXIMO;  { Esta subrotina também faz parte da subrotina ENTRADA_DADOS, sendo
                                          responsável pela entrada do ritmo máximo de construção permitido }

begin
 writeln;
 repeat
  write('[B]  O RITMO DE CONSTRUÍDO MÁXIMO PERMITIDO (Em casas/dia) : ');
  readln(aux);
  val(aux,RITMOMAX_aux,code);
  if code< >0 then
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
  end;
 until code=0;
end;


PROCEDURE DURACAO_ATIV;  { Esta subrotina também faz parte da subrotina ENTRADA_DADOS, sendo
                                          responsável pela entrada das durações de cada atividade }
```

```
begin
 writeln;
 writeln('[C]  ENTRE COM A DURAÇÃO DE CADA ATIVIDADE (Em dias) :');
 writeln;
 for i:=1 to (NEV-1) do
 begin
  repeat
   write('          ',NomeAtiv[i],' : ');
   readln(aux);
   val(aux,dATIVaux[i,i+1],code);
   if code< >0 then
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until code=0;
 end;
end;


PROCEDURE PRECEDENCIA;  { Esta subrotina também faz parte da subrotina ENTRADA_DADOS, sendo
                                                responsável pela entrada das precedências mínimas entre cada atividade }
begin
 writeln;
 writeln('[D]  AS PRECEDÊNCIAS TÉCNICAS MÍNIMAS (Em dias) :');
 writeln;
 for i:=2 to (NEV-1) do
 begin
  repeat
   write('          Entre  ',NomeAtiv[i-1],' e ',NomeAtiv[i],' : ');
   readln(aux);
   val(aux,BUFFERp_aux[i],code);
   if code< >0 then
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until code=0;
 end;
end;


PROCEDURE CUSTO_ATIV;  { Esta subrotina também faz parte da subrotina ENTRADA_DADOS, sendo
                                        responsável pela entrada dos custos percentuais de cada atividade }
begin
 writeln;
 writeln('[E]  O CUSTO DE CADA ATIVIDADE (Em %) :');
 writeln;
 for i:=1 to (NEV-1) do
 begin
  repeat
   write('          ',NomeAtiv[i],' : ');
   readln(aux);
   val(aux,Ct_aux[i,i+1],code);
   if code< >0 then
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until code=0;
 end;
end;
```

```
PROCEDURE CURVA_AGREG; { Esta subrotina também faz parte da subrotina ENTRADA_DADOS, sendo
                         responsável pela entrada de todos os dados da curva de agregação de recursos
                         disponíveis, ou seja: o n° de parcelas, a periodicidade entre elas e seus respectivos
                         valores percentuais }
begin
 writeln;
 writeln('[F] AGORA ENTRE COM OS DADOS DA CURVA DE RECURSOS DISPONÍVEIS :');
 writeln;
 repeat
  write('   [F.1] O n° de parcelas da curva de recursos disponíveis : ');
  readln(aux);
  val(aux,cronograma_aux,code);
  if code< >0 then
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
  end;
 until code=0;
 writeln;
 repeat
  write('   [F.2] A periodicidade das parcelas (EM MESES) : ');
  readln(aux);
  val(aux,periodo_aux,code);
  if code< >0 then
  begin
   gotoxy(whereX,whereY-1);
   DelLine;
  end;
 until code=0;
 writeln;
 writeln('   [F.3] Entre com as parcelas (Em %) : ');
 writeln;
 for i:=1 to (cronograma_aux) do
 begin
  repeat
   write('         ',i,'a. Parcela : ');
   readln(aux);
   val(aux,PARCELA_aux[i],code);
   if code< >0 then
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until code=0;
 end;
end;


PROCEDURE CONFIRMA_DADOS1; { Esta subrotina constrói uma tela, resumindo os dados de entrada do usuário
                             - exceto a curva de agregação. Será utilizada nas subrotinas :
                             (a) CONFIRMA_ALTERA - para que possam ser confirmados ou
                             modificados antes de uma simulação - e em (b) OPCAO_VER, para que
                             possam ser vistos a qualquer tempo }

begin
 clrscr;
 write('┌──────────────────────────────────────────────────────────────────────────┐');
 write('│                                                                          │');
 write('│      PROJETO : ',projeto_aux:8,'                                         │');
 write('│      ─────────                                                           │');
 write('│                                                                          │');
```

```
 write('|    [1] N° DE UNIDADES : ',N_aux:4,' casa(s)                                    |');
 write('|                                                                                  |');
 write('|    [2] RITMO MÁXIMO : ',RITMOMAX_aux:4:1,' casa(s)/dia                           |');
 write('|                                                                                  |');
 write('|      ──────────┬──────────┬──────────┬──────────                                |');
 write('|      ATIVIDADE  | [3] DURAÇÃO  | [4] PRECEDÊNCIA | [5] CUSTO(%)    |');
 write('|      ──────────┼──────────┼──────────┼──────────                                |');
 for i:=1 to (NEV-2) do
 begin
  if (i = 1) then
  begin
   write('|       ',NomeAtiv[i]:12,'  |  ',dATIVaux[i,i+1]:6:2,'  |      —      |  ');
   write(Ct_aux[i,i+1]:6:2,'          |');
  end;
  write('|       ',NomeAtiv[i+1]:12,'  |  ',dATIVaux[i+1,i+2]:6.2,'  |     ',BUFFERp_aux[i+1]:6:2);
  write('    |  ',Ct_aux[i+1,i+2]:6:2,'          |');
 end;
 write('|      ──────────┴──────────┴──────────┴──────────                                |');
 write('|                                                                                  |');
 write('└──────────────────────────────────────────────────────────────────────────────────┘');
end;
```

PROCEDURE CONFIRMA_DADOS2; { Idem CONFIRMA_DADOS1 para os dados da curva de agregação }

```
begin
 clrscr;
 write('┌──────────────────────────────────────────────────────────────────────────────────┐');
 write('|                    CURVA DE RECURSOS DISPONÍVEIS :                               |');
 write('|                    ──────────────────────────────                                |');
 write('|                                                                                  |');
 write('|      N° DE PARCELAS : ',cronograma_aux:2,' parcela(s)                            |');
 write('|      PERIODICIDADE : ',periodo_aux:3:1,' mês(es)                                 |');
 if (cronograma_aux ≤ 12) then
 begin
  write('|                              ──────────┬──────                                 |');
  write('|                              PARCELA  |  %                                     |');
  write('|                              ──────────┼──────                                 |');
  pc := periodo_aux;
  for i:=1 to (cronograma_aux) do
  begin
   write('|                                ',pc:4:1,'° MÊS | ',parcela_aux[i]:5:2);
   write('                    |');
   pc := pc + periodo_aux;
  end;
  write('|                              ──────────┴──────                                 |');
 end;
 if (cronograma_aux > 12) and (cronograma_aux ≤ 24) then
 begin
  write('|                ──────────┬──────          ──────────┬──────                   |');
  write('|                PARCELA  |  %           PARCELA  |  %                         |');
  write('|                ──────────┼──────          ──────────┼──────                   |');
  pc := periodo_aux;
  for i:=1 to (cronograma_aux-12) do
  begin
   write('|                  ',pc:4:1,'° MÊS | ',parcela_aux[i]:5:2,'          ',(pc+12*periodo_aux):4:1);
   write('° MÊS | ',parcela_aux[i+12]:5:2,'          |');
   pc := pc + periodo_aux;
  end;
  if (cronograma_aux < > 24) then
  begin
```

```
    write('|                ',pc:4:1,'º MÊS | ',parcela_aux[i+1]:5:2,'                              |');
    pc := pc + periodo_aux;
    for i:=(cronograma_aux-12+2) to 12 do
    begin
      write('|                ',pc:4:1,'º MÊS | ',parcela_aux[i]:5:2);
      write('                                                |');
      pc := pc + periodo_aux;
    end;
    write('|                                                                    |');
  end
  else
  begin
    write('|                                                                    |');
  end;
 end;
 if (cronograma_aux >24) then
 begin
  write('|                                                                    |');
  write('|    PARCELA | %      PARCELA | %      PARCELA | %                    |');
  write('|                                                                    |');
  pc := periodo_aux;
  for i:=1 to (cronograma_aux-24) do
  begin
    write('|      ',pc:4:1,'º MÊS | ',parcela_aux[i]:5:2,'      ',(pc+12*periodo_aux):4:1,'º MÊS | ');
    write(parcela_aux[i+12]:5:2,'      ',(pc+24*periodo_aux):4:1,'º MÊS | ',parcela_aux[i+24]:5:2,'      |');
    pc := pc + periodo_aux;
  end;
  if (cronograma_aux < > 36) then
  begin
    write('|      ',pc:4:1,'º MÊS | ',parcela_aux[i+1]:5:2,'      ',(pc+12*periodo_aux):4:1,'º MÊS | ');
    write(parcela_aux[i+13]:5:2,'                                  |');
    pc := pc + periodo_aux;
    for i:=(cronograma_aux-24+2) to 12 do
    begin
      write('|      ',pc:4:1,'º MÊS | ',parcela_aux[i]:5:2,'      ',(pc+12*periodo_aux):4:1,'º MÊS | ');
      write(parcela_aux[i+12]:5:2,'                                |');
      pc := pc + periodo_aux;
    end;
    write('|                                                                    |');
  end
  else
  begin
    write('|                                                                    |');
  end;
 end;
 write('└────────────────────────────────────────────────────────────────────┘');
end;
```

PROCEDURE GRAVA_DADOS0; { Esta subrotina acrescenta ao arquivo NOMES.DAT um novo projeto - depois de confirmado em CONFIRMA_DADOS1 }

```
begin
 append(arq0);
 writeln(arq0,projeto_aux);
 close(arq0);
end;
```

PROCEDURE GRAVA_DADOS1; { Esta subrotina grava em um arquivo 'projeto'.DAT os dados confirmados em CONFIRMA_DADOS1 }

```
begin
 assign(arq6,'C:\PENHA\'+projeto_aux+'.DAT');
 rewrite(arq6);
 writeln(arq6,N_aux,RITMOMAX_aux);
 for i:=1 to (NEV-1) do
 begin
  case i of
  1: writeln(arq6,dATIVaux[i,i+1],Ct_aux[i,i+1]);
  2..(NEV-1): writeln(arq6,dATIVaux[i,i+1],Ct_aux[i,i+1],BUFFERp_aux[i]);
  end;
 end;
 close(arq6);
end;


PROCEDURE GRAVA_DADOS2; { Esta subrotina acrescenta no arquivo 'projeto'.DAT os dados confirmados em
                                        CONFIRMA_DADOS2 }
begin
 append(arq6);
 writeln(arq6,cronograma_aux,periodo_aux);
 for i:=1 to (cronograma_aux) do
 begin
  writeln(arq6,PARCELA_aux[i]);
 end;
 close(arq6);
end;

PROCEDURE CONFIRMA_ALTERA; { Esta subrotina e utilizada toda fez que o usuário desejar simular um projeto.
                                        Serve para confirmar ou alterar os dados necessários a simulação }
begin
 repeat
  Confirma_Dados1;
  writeln;
  write(' Confirma(S/N) ? ');
  readln(resposta);
  if (resposta = 'N') or (resposta = 'n') then
  begin
   writeln;
   write(' Digite o n° do dado que deseja alterar : ');
   readln(opcao);
   clrscr;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   case (opcao) of
   1: N_Unidades;
   2: Ritmo_Maximo;
   3: Duracao_Ativ;
   4: Precedencia;
   5: Custo_Ativ;
   end;
  end;
 until (resposta = 'S') or (resposta = 's');
 Grava_Dados1;
 repeat
  Confirma_Dados2;
  writeln;
  write(' Confirma(S/N) ? ');
  readln(resposta);
```

```
  if (resposta = 'N') or (resposta = 'n') then
  begin
    clrscr;
    writeln;
    writeln;
    Curva_Agreg;
  end;
 until (resposta = 'S') or (resposta = 's');
 Grava_Dados2;
end;


PROCEDURE ENTRADA_DADOS;  { Esta subrotina e responsável pela entrada de todos os dados necessários,
                            alteração e confirmação dos mesmos }
begin
 clrscr;
 writeln('-------------------------------------------------------------------------------');
 writeln('ATENÇÃO USUÁRIO!!!  AGORA VOCÊ DEVERÁ ENTRAR COM OS SEGUINTES DADOS!!!');
 writeln('-------------------------------------------------------------------------------');
 writeln('OBS: A cada entrada de dados, tecle ENTER');
 writeln;
 writeln;
 N_Unidades;
 Ritmo_Maximo;
 Duracao_Ativ;
 Precedencia;
 Custo_Ativ;
 Curva_Agreg;
 Confirma_Altera;
end;
PROCEDURE AGUARDE; { Esta subrotina faz uma tela com os dizeres: AGUARDE CALCULANDO... }


begin
 clrscr;
 desliga_cursor;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln;
 writeln('                ┌──────────────────────────────────────┐');
 writeln('                │                                      │');
 writeln('                │                                      │');
 writeln('                │     AGUARDE  CALCULANDO...           │');
 writeln('                │                                      │');
 writeln('                │                                      │');
 writeln('                └──────────────────────────────────────┘');
end;


PROCEDURE SUPORTE; { Esta subrotina 'carrega' os valores_aux dos dados depois que estes são confirmados e calcula,
                     através desses valores, os outros dados necessários a execução do modelo.GMS.
                     Ex: DATACRON, PARCELA/100 e RITMOi }

begin
 projeto := projeto_aux;
 N := N_aux;
 RITMOi[1] := 1/RITMOMAX_aux;
 SOMATIV := 0;
```

```
 for i:=1 to (NEV-1) do
 begin
  dATIV[i,i+1] := dATIVaux[i,i+1];
  SOMATIV := SOMATIV + dATIV[i,i+1];
  Ct[i,i+1] := Ct_aux[i,i+1]/100;
 end;
 BUFFERp_SOMA := 0;
 for i:= 2 to (NEV-1) do
 begin
  BUFFERp[i] := BUFFERp_aux[i];
  BUFFERp_SOMA := BUFFERp_SOMA + BUFFERp[i];
 end;
 cronograma := cronograma_aux;
 periodo := periodo_aux;
 PARCELA[1] := PARCELA_aux[1]/100;
 DATACRON[1] := periodo*20; { Transforma data em 'meses' para 'dias úteis' }
 for i:=2 to (cronograma) do
 begin
  PARCELA[i] := PARCELA_aux[i]/100;
  DATACRON[i] := i*DATACRON[1];
 end;
 RITMOi[40] := (DATACRON[cronograma] - (DI + SOMATIV + BUFFERp_SOMA))/(N-1);
 DIFRITMO := (RITMOi[40] - RITMOi[1])/39{19};
 for i:=2 to 39{19} do
 begin
  RITMOi[i] := RITMOi[1] + DIFRITMO*(i-1);
 end;
end;


PROCEDURE SOL_INICIAL;  { Esta subrotina obtém os valores iniciais para as variáveis do modelo.GMS }

begin
 SQRDIFmin := 10000;
 PARCELA_MEDIA := 1/cronograma;
 SDesvio := 0;
 Tol_Padrao := 0.01;
 for x:=1 to (cronograma) do
 begin
  Desvio[x] := ABS(PARCELA[x] - PARCELA_MEDIA);
  SDesvio := SDesvio + Desvio[x];
 end;
 Desvio_Medio := SDesvio/cronograma;
 for i:=1 to 40 do
 begin
  case (cronograma) of
  12:begin
      f_PARCELA := 1;
      if (Desvio_Medio ≤ 0.0287) then
      begin
       f_Desvio := 1;
      end
      else
      begin
       f_Desvio := Desvio_Medio/0.0287;
      end;
     end;
  18:begin
      f_PARCELA := 0.66666667;
      if (Desvio_Medio ≤ 0.0193) then
      begin
       f_Desvio := 1;
```

```
    end
    else
    begin
>     f_Desvio := Desvio_Medio/0.0193;
    end;
  end;
24:begin
    f_PARCELA := 0.5;
    if (Desvio_Medio ≤ 0.0148) then
    begin
      f_Desvio := 1;
    end
    else
    begin
      f_Desvio := Desvio_Medio/0.0148;
    end;
  end;
30:begin
    f_PARCELA := 0.4;
    if (Desvio_Medio ≤ 0.0120) then
    begin
      f_Desvio := 1;
    end
    else
    begin
      f_Desvio := Desvio_Medio/0.0120;
    end;
  end;
36:begin
    f_PARCELA := 0.33333333;
    if (Desvio_Medio ≤ 0.0101) then
    begin
      f_Desvio := 1;
    end
    else
    begin
      f_Desvio := Desvio_Medio/0.0101;
    end;
  end;
end;
RITMO := RITMOi[i];
Tol_Media := f_Desvio*f_PARCELA*Tol_Padrao;
DURACAO[1,2] := dATIV[1,2] + RITMO*(N-1);
DE[1] := DI;
DEcedo[1] := DE[1];
DT[1,2] := DE[1] + DURACAO[1,2];
DEtarde[NEV-1] := DATACRON[cronograma] - dATIV[NEV-1,NEV] - RITMO*(N-1);
for k:=2 to (NEV-1) do
begin
  DURACAO[k,k+1] := dATIV[k,k+1] + RITMO*(N-1);
  DEtarde[NEV-k] := DEtarde[NEV-k+1] - dATIV[NEV-k,NEV-k+1] -
            BUFFERp[NEV-k+1];
end;
A0 := Trunc(DT[1,2]/(20*periodo)) + 1;
for j:=1 to A0 do
begin
  if (DATACRON[j] > DT[1,2]) then
  begin
    if (j=1) then
    begin
      D[1,j] := Ct[1,2] - PARCELA[j];
    end
    else
```

```
  begin
   D[1,j] := ((DT[1,2]-DATACRON[j-1])/DURACAO[1,2])*Ct[1,2] -
         PARCELA[j];
  end;
 end
 else
 begin
   D[1,j] := 20*periodo*Ct[1,2]/DURACAO[1,2] - PARCELA[j];
 end;
end;
for j:=A0+1 to (cronograma) do
begin
 D[1,j] := (-1)*PARCELA[j];
end;
DEcedo[2] := DEcedo[1] + dATIV[1,2] + BUFFERp[2];
j := 0;
for a:=2 to (NEV-2) do
begin
 repeat
  inc(j);
 until (D[a-1,j]<0) and (ABS(D[a-1,j])>Tol_Media) or (j>cronograma);
 if (j>cronograma) then
 begin
  k := 1;
  VETORmin := 1000;
  DE[a] := DEcedo[a];
  DE_aux[a] := DE[a];
  DT[a,a+1] := DE[a] + dATIV[a,a+1] + RITMO*(N-1);
  repeat
   SOMADa := 0;
   A1 := Trunc(DE[a]/(20*periodo)) + 1;
   B1 := Trunc(DT[a,a+1]/(20*periodo)) + 1;
   if (A1>cronograma) or (B1>cronograma) then
   begin
    if (A1>cronograma) then
    begin
     A1 := cronograma;
     B1 := cronograma;
    end
    else
    begin
     B1 := cronograma;
    end;
   end;
   for j:= A1 to B1 do
   begin
    if j=A1 then
    begin
     D[a,j] := ((DATACRON[j]-DE[a])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
    end
    else
    begin
     if j=B1 then
     begin
      D[a,j] := ((DT[a,a+1]-DATACRON[j-1])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
     end
     else
     begin
      D[a,j] := ((20*periodo)/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
     end;
    end;
    SOMADa := SOMADa + ABS(D[a,j]);
   end;
```

```
  if (VETORmin > SOMADa) then
  begin
    VETORmin := SOMADa;
    DE_bom[a] := DE[a];
  end;
  DE[a] := (Trunc(DE_aux[a]/(20*periodo)+k))*(20*periodo);
  DT[a,a+1] := DE[a] + dATIV[a,a+1] + RITMO*(N-1);
  if (DE[a] ≤ DEtarde[a]) then
  begin
    inc(k);
  end;
 until (DE[a] > DEtarde[a]);
 DE[a] := DE_bom[a];
end
else
begin
 if (ABS(D[a-1,j])≥(20*periodo*Ct[a,a+1]/DURACAO[a,a+1])) then
 begin
   if (j=1) then
   begin
     DE[a] := DEcedo[a];
   end
   else
   begin
     DE[a] := DATACRON[j-1];
   end;
 end
 else
 begin
   DE[a] := DATACRON[j] - ABS(D[a-1,j])*DURACAO[a,a+1]/Ct[a,a+1];
 end;
 if (DE[a]<DEcedo[a]) or (DE[a]>DEtarde[a]) then
 begin
   if (DE[a]≤DEcedo[a]) then
   begin
     DE[a] := DEcedo[a];
   end
   else
   begin
     DE[a] := DEtarde[a];
   end;
 end;
 DE_aux[a] := DE[a];
 DT[a,a+1] := DE[a] + dATIV[a,a+1] + RITMO*(N-1);
 VETORmin := 1000;
 k := 1;
 repeat
   SOMADa := 0;
   SOMADa_1 := 0;
   A1 := Trunc(DE[a]/(20*periodo)) + 1;
   B1 := Trunc(DT[a,a+1]/(20*periodo)) + 1;
   if (A1>cronograma) or (B1>cronograma) then
   begin
     if (A1>cronograma) then
     begin
       A1 := cronograma;
       B1 := cronograma;
     end
     else
     begin
       B1 := cronograma;
     end;
   end;
```

```
  for j:=A1 to B1 do
  begin
   if j=A1 then
   begin
    D[a,j] := ((DATACRON[j]-DE[a])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
   end
   else
   begin
    if j=B1 then
    begin
     D[a,j] := ((DT[a,a+1]-DATACRON[j-1])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
    end
    else
    begin
     D[a,j] := ((20*periodo)/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
    end;
   end;
   SOMADa := SOMADa + ABS(D[a,j]);
   SOMADa_1 := SOMADa_1 + ABS(D[a-1,j]);
  end;
  if (VETORmin > SOMADa) then
  begin
   VETORmin := SOMADa;
   DE_bom[a] := DE[a];
  end;
  if (SOMADa ≥ SOMADa_1) then
  begin
   DE[a] := (Trunc(DE_aux[a]/(20*periodo)+k))*(20*periodo);
   if (DE[a] ≤ DEtarde[a]) then
   begin
    inc(k);
    DT[a,a+1] := DE[a] + dATIV[a,a+1] + RITMO*(N-1);
   end;
  end;
 until (SOMADa < SOMADa_1) or (DE[a] > DEtarde[a]);
 if (DE[a] > DEtarde[a]) then
 begin
  DE[a] := DE_bom[a];
 end;
end;
DEcedo[a+1] := DE[a] + dATIV[a,a+1] + BUFFERp[a+1];
DT[a,a+1] := DE[a] + dATIV[a,a+1] + RITMO*(N-1);
if (SOMADa ≥ SOMADa_1) then
begin
 A1 := Trunc(DE[a]/(20*periodo)) + 1;
 B1 := Trunc(DT[a,a+1]/(20*periodo)) + 1;
 if (A1>cronograma) or (B1>cronograma) then
 begin
  if (A1>cronograma) then
  begin
   A1 := cronograma;
   B1 := cronograma;
  end
  else
  begin
   B1 := cronograma;
  end;
 end;
 for j:=A1 to B1 do
 begin
  if j=A1 then
  begin
   D[a,j] := ((DATACRON[j]-DE[a])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
```

```
    end
    else
    begin
     if j=B1 then
     begin
      D[a,j] := ((DT[a,a+1]-DATACRON[j-1])/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
     end
     else
     begin
      D[a,j] := ((20*periodo)/DURACAO[a,a+1])*Ct[a,a+1] + D[a-1,j];
     end;
    end;
   end;
  end;
  if (A1<>1) then
  begin
   for j:=1 to (A1-1) do
   begin
    D[a,j] := D[a-1,j];
   end;
  end;
  if (B1<> cronograma) then
  begin
   for j:=B1+1 to (cronograma) do
   begin
    D[a,j] := D[a-1,j];
   end;
  end;
  j := A1 - 1;
 end;
 k := 1;
 VETORmin := 1000;
 DE[a+1] := DEcedo[a+1];
 DE_aux[a+1] := DE[a+1];
 DT[a+1,a+2] := DE[a+1] + dATIV[a+1,a+2] + RITMO*(N-1);
 repeat
  SOMADa := 0;
  A1 := Trunc(DE[a+1]/(20*periodo)) + 1;
  B1 := Trunc(DT[a+1,a+2]/(20*periodo)) + 1;
  if (A1>cronograma) or (B1>cronograma) then
  begin
   if (A1>cronograma) then
   begin
    A1 := cronograma;
    B1 := cronograma;
   end
   else
   begin
    B1 := cronograma;
   end;
  end;
  for j:= A1 to B1 do
  begin
   if j=A1 then
   begin
    D[a+1,j] := ((DATACRON[j]-DE[a+1])/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
   end
   else
   begin
    if j=B1 then
    begin
     D[a+1,j] := ((DT[a+1,a+2]-DATACRON[j-1])/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
    end
```

```
    else
    begin
     D[a+1,j] := ((20*periodo)/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
    end;
   end;
   SOMADa := SOMADa + ABS(D[a+1,j]);
  end;
  if (VETORmin > SOMADa) then
  begin
   VETORmin := SOMADa;
   DE_bom[a+1] := DE[a+1];
  end;
  DE[a+1] := (Trunc(DE_aux[a+1]/(20*periodo)+k))*(20*periodo);
  DT[a+1,a+2] := DE[a+1] + dATIV[a+1,a+2] + RITMO*(N-1);
  if (DE[a+1] ≤ DEtarde[a+1]) then
  begin
   inc(k);
  end;
until (DE[a+1] > DEtarde[a+1]);
DE[a+1] := DE_bom[a+1];
DT[a+1,a+2] := DE[a+1] + dATIV[a+1,a+2] + RITMO*(N-1);
A1 := Trunc(DE[a+1]/(20*periodo)) + 1;
B1 := Trunc(DT[a+1,a+2]/(20*periodo)) + 1;
if (A1>cronograma) or (B1>cronograma) then
begin
 if (A1>cronograma) then
 begin
  A1 := cronograma;
  B1 := cronograma;
 end
 else
 begin
  B1 := cronograma;
 end;
end;
for j:= A1 to B1 do
begin
 if j=A1 then
 begin
  D[a+1,j] := ((DATACRON[j]-DE[a+1])/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
 end
 else
 begin
  if j=B1 then
  begin
   D[a+1,j] := ((DT[a+1,a+2]-DATACRON[j-1])/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
  end
  else
  begin
   D[a+1,j] := ((20*periodo)/DURACAO[a+1,a+2])*Ct[a+1,a+2] + D[a,j];
  end;
 end;
end;
if (A1< >1) then
begin
 for j:=1 to (A1-1) do
 begin
  D[a+1,j] := D[a,j];
 end;
end;
if (B1< > cronograma) then
begin
 for j:=B1+1 to (cronograma) do
```

```
    begin
      D[a+1,j] := D[a,j];
    end;
  end;
  SQRDIF := 0;
  for j:=1 to (cronograma) do
  begin
    SQRDIF := SQRDIF + ABS(D[NEV-1,j]);
  end;
  if (SQRDIF < SQRDIFmin) then
  begin
    SQRDIFmin := SQRDIF;
    RITMOotm := RITMO;
    for j:=1 to (NEV-1) do
    begin
      DEotm[j] := DE[j];
    end;
  end;
 end;
end;


PROCEDURE MONTAGEM1; { Abre o arquivo SIS.GMS, montando o modelo desejado }

var arq1 : text;

begin
 assign(arq1,'C:\PENHA\SIS.GMS');
 rewrite(arq1);
 writeln(arq1,'$OFFSYMXREF OFFSYMLIST');
 writeln(arq1,'SCALAR');
 writeln(arq1,' N    /',N,'/');
 writeln(arq1,' NEV  /',NEV,'/');
 writeln(arq1,' DI   /',DI,'/;');
 writeln(arq1,'SETS');
 writeln(arq1,' EV        /1*',NEV,',1A*',(NEV-1),'A/');
 writeln(arq1,' cronograma /1*',cronograma,'/');
 writeln(arq1,' dd        /1*',dd,'/;');
 writeln(arq1,'PARAMETER dATIV(EV,EV)');
 for i:=1 to (NEV-1) do
 begin
  case i of
  1: writeln(arq1,' / ',i,'A.',(i+1),' ',dATIV[i,(i+1)]);
  2..(NEV-2): writeln(arq1,'   ',i,'A.',(i+1),' ',dATIV[i,(i+1)]);
  (NEV-1): writeln(arq1,'   ',i,'A.',(i+1),' ',dATIV[i,(i+1)],' /;');
  end;
 end;
 writeln(arq1,'PARAMETER FOLGA(EV)');
 for i:=2 to (NEV-1) do
 begin
  case i of
  2: writeln(arq1,' / ',i,'  ',BUFFERp[i]);
  3..(NEV-2): writeln(arq1,'   ',i,'  ',BUFFERp[i]);
  (NEV-1): writeln(arq1,'   ',i,'  ',BUFFERp[i],' /;');
  end;
 end;
 writeln(arq1,'PARAMETER DATACRON(cronograma)');
 for i:=1 to (cronograma) do
 begin
  case i of
  1: writeln(arq1,' / ',i,'  ',DATACRON[i]);
  end;
  if (i ≥2) and (i ≤ cronograma-1) then
```

```
  begin
   writeln(arq1,'    ',i,'  ',DATACRON[i]);
  end;
  if (i=cronograma) then
  begin
   writeln(arq1,'    ',i,'  ',DATACRON[i],' /;');
  end;
end;
writeln(arq1,'PARAMETER  PARCELA(cronograma)');
for i:=1 to (cronograma) do
begin
  case i of
  1: writeln(arq1,' / ',i,'  ',PARCELA[i]);
  end;
  if (i ≥ 2) and (i ≤ cronograma-1) then
  begin
   writeln(arq1,'    ',i,'  ',PARCELA[i]);
  end;
  if (i=cronograma) then
  begin
   writeln(arq1,'    ',i,'  ',PARCELA[i],' /;');
  end;
end;
writeln(arq1,'PARAMETER  Ct(EV,EV)');
for i:=1 to (NEV-1) do
begin
  case i of
  1: writeln(arq1,' / ',i,'A.',(i+1),'  ',Ct[i,(i+1)]);
  2..(NEV-2): writeln(arq1,'    ',i,'A.',(i+1),'  ',Ct[i,(i+1)]);
  (NEV-1): writeln(arq1,'    ',i,'A.',(i+1),'  ',Ct[i,(i+1)],' /;');
  end;
end;
writeln(arq1,'VARIABLES');
writeln(arq1,' DE(EV)');
writeln(arq1,' DU');
writeln(arq1,' DTOT');
writeln(arq1,' DT(EV,EV)');
writeln(arq1,' RITMO');
writeln(arq1,' Y(EV,EV,cronograma)');
writeln(arq1,' delta(EV,EV,cronograma,dd)');
writeln(arq1,' DURACAO(EV,EV)');
writeln(arq1,' SOMAY(cronograma)');
writeln(arq1,' SOMAPERC(cronograma)');
writeln(arq1,' SQRDIF;');
writeln(arq1,'POSITIVE  VARIABLES  DE,DU,DTOT,DT,RITMO,Y,delta,DURACAO');
writeln(arq1,'                SOMAY,SOMAPERC;');
writeln(arq1,' RITMO.L = ',RITMOotm,';');
writeln(arq1,' RITMO.LO = ',RITMOi[1],';');
writeln(arq1,' RITMO.UP = ',RITMOi[40],';');
writeln(arq1,' DE.L("1A") = ',DI,';');
for i:=2 to (NEV-1) do
begin
  writeln(arq1,' DE.L(",i,") = DE.L(",(i-1),'A") + dATIV(",(i-1),'A",",i,");');
  writeln(arq1,' DE.L(",i,'A") = ',DEotm[i],';');
end;
writeln(arq1,' DE.L(",NEV,") = DE.L(",(NEV-1),'A") + dATIV(",(NEV-1),'A",",NEV,");');
writeln(arq1,' DE.LO("1A") = ',DI,';');
for i:=2 to (NEV-1) do
begin
  writeln(arq1,' DE.LO(",i,") = DE.LO(",(i-1),'A") + dATIV(",(i-1),'A",",i,");');
  writeln(arq1,' DE.LO(",i,'A") = DE.LO(",i,") + FOLGA(",i,");');
end;
writeln(arq1,' DE.LO(",NEV,") = DE.LO(",(NEV-1),'A") + dATIV(",(NEV-1),'A",",NEV,");');
```

```
  writeln(arq1,' DE.UP(",NEV,") = DATACRON(",cronograma,") - RITMO.LO*(',N-1,');');
  for i:=(NEV-1) downto 2 do
  begin
    writeln(arq1,' DE.UP(",i,'A") = DE.UP(",(i+1),") - dATIV(",i,'A",",(i+1),");');
    writeln(arq1,' DE.UP(",i,") = DE.UP(",i,'A") - FOLGA(",i,");');
  end;
  writeln(arq1,' DE.UP("1A") = DE.UP("2") - dATIV("1A","2");');
  writeln(arq1,' DURACAO.LO(EV+NEV,EV+1) = (dATIV(EV+NEV,EV+1)+RITMO.LO*(',N-1,'))$(ORD(EV) lt
',NEV,');');
  writeln(arq1,' DURACAO.UP(EV+NEV,EV+1) = (dATIV(EV+NEV,EV+1)+RITMO.UP*(',N-1,'))$(ORD(EV) lt
',NEV,');');
  writeln(arq1,' DT.L(EV+NEV,EV+1) = (RITMO.L*(',N-1,') + DE.L(EV+NEV) +');
  writeln(arq1,'                dATIV(EV+NEV,EV+1))$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' DURACAO.L(EV+NEV,EV+1) = (DT.L(EV+NEV,EV+1) - DE.L(EV+NEV))');
  writeln(arq1,'                   $(ORD(EV) lt ',NEV,');');
  writeln(arq1,' DT.LO(EV+NEV,EV+1) = (DE.LO(EV+NEV)+DURACAO.LO(EV+NEV,EV+1))$(ORD(EV) lt
',NEV,');');
  writeln(arq1,' DT.UP(EV+NEV,EV+1) = (DE.LO(EV+NEV)+DURACAO.UP(EV+NEV,EV+1))$(ORD(EV) lt
',NEV,');');
  writeln(arq1,' DU.LO = DE.LO(",NEV,") - DE.LO("1A");');
  writeln(arq1,' DU.UP = DE.UP(",NEV,") - DE.LO("1A");');
  writeln(arq1,' DTOT.LO = DT.LO(",NEV-1,'A",",NEV,");');
  writeln(arq1,' DTOT.UP = DT.UP(",NEV-1,'A",",NEV,");');
  writeln(arq1,' delta.L(EV+NEV,EV+1,cronograma,"1") = (1$((DATACRON(cronograma)-');
  writeln(arq1,'      DE.L(EV+NEV)) lt 0) + 0$((DATACRON(cronograma)-DE.L(EV+NEV)) ge 0))$');
  writeln(arq1,'      (ORD(EV) lt ',NEV,');');
  writeln(arq1,' delta.L(EV+NEV,EV+1,cronograma,"2") = (1$((DATACRON(cronograma)-');
  writeln(arq1,'      DE.L(EV+NEV)) ge 0 AND (DT.L(EV+NEV,EV+1)-DATACRON(cronograma)) ge 0)');
  writeln(arq1,'      + 0$((DATACRON(cronograma) - DE.L(EV+NEV)) lt 0)');
  writeln(arq1,'      + 0$((DT.L(EV+NEV,EV+1) - DATACRON(cronograma)) lt 0))');
  writeln(arq1,'      $(ORD(EV) lt ',NEV,');');
  writeln(arq1,' delta.L(EV+NEV,EV+1,cronograma,"3") = (1 - delta.l(EV+NEV,EV+1,cronograma,"2")');
  writeln(arq1,'      - delta.L(EV+NEV,EV+1,cronograma,"1"))$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' delta.LO(EV+NEV,EV+1,cronograma,dd) = 0$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' delta.UP(EV+NEV,EV+1,cronograma,dd) = 1$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' Y.L(EV+NEV,EV+1,cronograma) = (0*delta.L(EV+NEV,EV+1,cronograma,"1") +');
  writeln(arq1,'      delta.L(EV+NEV,EV+1,cronograma,"2")*Ct(EV+NEV,EV+1)');
  writeln(arq1,'      *((DATACRON(cronograma)-DE.L(EV+NEV))/DURACAO.L(EV+NEV,EV+1)) +');
  writeln(arq1,'      delta.L(EV+NEV,EV+1,cronograma,"3")*Ct(EV+NEV,EV+1))$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' Y.LO(EV+NEV,EV+1,cronograma) = 0$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' Y.UP(EV+NEV,EV+1,cronograma) = Ct(EV+NEV,EV+1)$(ORD(EV) lt ',NEV,');');
  writeln(arq1,' SOMAY.L(cronograma) = SUM(EV$(ORD(EV) lt ',NEV,'),Y.L(EV+NEV,EV+1,cronograma));');
  writeln(arq1,' SOMAY.LO(cronograma) = 0;');
  writeln(arq1,' SOMAY.UP(cronograma) = 1;');
  writeln(arq1,' SOMAPERC.LO(cronograma) = 0;');
  writeln(arq1,' SOMAPERC.UP(cronograma) = 1;');
  writeln(arq1,'EQUATIONS');
  writeln(arq1,' PC');
  writeln(arq1,' PC1(EV)');
  writeln(arq1,' PC2(EV)');
  writeln(arq1,' R1(EV)');
  writeln(arq1,' R2(EV)');
  writeln(arq1,' R1(EV)');
  writeln(arq1,' R3(EV)');
  writeln(arq1,' R4(EV)');
  writeln(arq1,' PZ(cronograma)');
  writeln(arq1,' S(cronograma)');
  writeln(arq1,' SS0');
  writeln(arq1,' SS(cronograma)');
  writeln(arq1,' YT(EV,cronograma)');
  writeln(arq1,' SD(EV,cronograma)');
  writeln(arq1,' CD1(EV,cronograma)');
  writeln(arq1,' CD2(EV,cronograma)');
```

```
writeln(arq1,' CD3(EV,cronograma)');
writeln(arq1,' CD4(EV,cronograma)');
writeln(arq1,' CD5(EV,cronograma)');
writeln(arq1,' CD6(EV,cronograma)');
writeln(arq1,' OBJ;');
writeln(arq1,'PC.. DE("1A") =G= ',DI,';');
writeln(arq1,'PC1(EV)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' DE(EV+1) =E= DE(EV+NEV) + dATIV(EV+NEV,EV+1);');
writeln(arq1,'PC2(EV)$(ORD(EV) lt ',NEV-1,')..');
writeln(arq1,' DE(EV+',NEV+1,') =G= DE(EV+1) + FOLGA(EV+1);');
writeln(arq1,'R1(EV)$(ORD(EV) eq NEV).. DU =E= DE(EV) - DE("1A");');
writeln(arq1,'R2(EV)$(ORD(EV) eq NEV).. DTOT =E= DT(EV+',NEV-1,',EV);');
writeln(arq1,'R3(EV)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' RITMO*(N-1) =E= DT(EV+NEV,EV+1) - DE(EV+NEV) -dATIV(EV+NEV,EV+1);');
writeln(arq1,'R4(EV)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' DURACAO(EV+NEV,EV+1) =E= DT(EV+NEV,EV+1) - DE(EV+NEV);');
writeln(arq1,'PZ(cronograma)$(ORD(cronograma) eq ',cronograma,')..');
writeln(arq1,' DTOT =L= DATACRON(cronograma);');
writeln(arq1,'S(cronograma).. SOMAY(cronograma) =E= SUM(EV$(ORD(EV) lt ',NEV,'),');
writeln(arq1,' Y(EV+NEV,EV+1,cronograma));');
writeln(arq1,'SS0.. SOMAPERC("1") =E= SQR(SOMAY("1") - PARCELA("1"));');
writeln(arq1,'SS(cronograma)$(ORD(cronograma) ne 1)..');
writeln(arq1,' SOMAPERC(cronograma) =E= SQR(SOMAY(cronograma) -');
writeln(arq1,' SOMAY(cronograma-1) - PARCELA(cronograma));');
writeln(arq1,'YT(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' Y(EV+NEV,EV+1,cronograma) =E= 0*delta(EV+NEV,EV+1,cronograma,"1") +');
writeln(arq1,' delta(EV+NEV,EV+1,cronograma,"2")*Ct(EV+NEV,EV+1)*((DATACRON');
writeln(arq1,' (cronograma)-DE(EV+NEV))/DURACAO(EV+NEV,EV+1)) + delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"3")*Ct(EV+NEV,EV+1);');
writeln(arq1,'SD(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' SUM(dd,delta(EV+NEV,EV+1,cronograma,dd)) =E= 1;');
writeln(arq1,'CD1(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DATACRON(cronograma) - DE(EV+NEV))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"3") =G= 0;');
writeln(arq1,'CD2(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DATACRON(cronograma) - DE(EV+NEV))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"3") =G= 0;');
writeln(arq1,'CD3(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DT(EV+NEV,EV+1) - DATACRON(cronograma))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"1") =G= 0;');
writeln(arq1,'CD4(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DT(EV+NEV,EV+1) - DATACRON(cronograma))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"2") =G= 0;');
writeln(arq1,'CD5(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DATACRON(cronograma) - DE(EV+NEV))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"1") =L= 0;');
writeln(arq1,'CD6(EV,cronograma)$(ORD(EV) lt ',NEV,')..');
writeln(arq1,' (DT(EV+NEV,EV+1) - DATACRON(cronograma))*delta(EV+NEV,EV+1,');
writeln(arq1,' cronograma,"3") =L= 0;');
writeln(arq1,'OBJ.. SQRDIF =E= SUM(cronograma,SOMAPERC(cronograma)) +');
writeln(arq1,'     SUM(cronograma,(SUM(EV$(ORD(EV) lt ',NEV,'),(1-delta');
writeln(arq1,'     (EV+NEV,EV+1,cronograma,"1"))*delta(EV+NEV,EV+1,');
writeln(arq1,'     cronograma,"1") + (1-delta(EV+NEV,EV+1,cronograma,"2"))*');
writeln(arq1,'     delta(EV+NEV,EV+1,cronograma,"2") + (1-delta(EV+NEV,');
writeln(arq1,'     EV+1,cronograma,"3"))*delta(EV+NEV,EV+1,cronograma,"3"))));');
writeln(arq1,'MODEL LBALANCO /ALL/;');
writeln(arq1,'OPTION LIMROW = 0;');
writeln(arq1,'OPTION LIMCOL = 0;');
writeln(arq1,'OPTION SOLPRINT = OFF;');
writeln(arq1,'SOLVE LBALANCO USING DNLP MINIMIZING SQRDIF;');
writeln(arq1,'FILE RES /STATUS.DAT/;');
writeln(arq1,'PUT RES;');
writeln(arq1,'PUT LBALANCO.MODELSTAT;');
```

```
 close(arq1);
end;


PROCEDURE MONTAGEM2;  { Abre o arquivo SIS_OTM.GMS para separar no arquivo de impressão.LST
                                    o modelo em si e os valores ótimos, facilitando a recuperação dos mesmos }
var  arq2 : text;

begin
 assign(arq2,'C:\PENHA\SIS_OTM.GMS');
 rewrite(arq2);
 writeln(arq2,'OPTION DE:6:0:1;DISPLAY  DE.L;');
 writeln(arq2,'OPTION DT:6:0:1;DISPLAY  DT.L;');
 writeln(arq2,'OPTION SOMAY:8:0:1;DISPLAY  SOMAY.L;');
 writeln(arq2,'OPTION DECIMALS=6;');
 writeln(arq2,'DISPLAY DTOT.L;');
 writeln(arq2,'DISPLAY DU.L;');
 writeln(arq2,'DISPLAY RITMO.L;');
 close(arq2);
end;


PROCEDURE STATUS_SOLVER;  { Recupera do arquivo STATUS.DAT o n° correspondente ao status do solver do
                                    modelo depois de executado }
var
 arq5     : text;
 string7  : string[7];

begin
 assign(arq5,'C:\STATUS.DAT');
 reset(arq5);
 readln(arq5,string7,status);
 close(arq5);
end;

PROCEDURE VALORES_OTIMOS; { Recupera do arquivo de 'displays'.LST os valores ótimos das variáveis de decisão }

var
 arq        : text;
 string4    : string[4];
 string3    : string[3];
 string5    : string[5];
 string44   : string[44];

begin
 assign(arq,'C:\PENHA\SIS_OTM.LST');
 reset(arq);
 repeat
  readln(arq,string4);
 until (string4 = '----');
 begin
  readln(arq,string44);
  for i:=1 to (NEV-1) do
  begin
   readln(arq,string44);
  end;
  read(arq,string3);
  if (string3 = '1A ') then
  begin
   j := 2;
   readln(arq,DE[1]);
  end
  else
```

```
  begin
   j := 3;
   DE[1] := 0;
   readln(arq,DE[2]);
  end;
  for i:=j  to (NEV-1) do
  begin
   readln(arq,string3,DE[i]);
  end;
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44);
  for i:=1 to (NEV-1) do
  begin
   readln(arq,string5,DT[i,(i+1)]);
  end;
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44);
  read(arq,string3);
  if (string3 = '1  ') then
  begin
   j := 2;
   readln(arq,SOMAY[1]);
  end
  else
  begin
   j := 3;
   SOMAY[1] := 0;
   readln(arq,SOMAY[2]);
  end;
  for i:=j to (cronograma) do
  begin
   readln(arq,string3,SOMAY[i]);
  end;
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44,DTOT);
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44,DU);
  readln(arq,string44);
  readln(arq,string44);
  readln(arq,string44,RITMO);
  close(arq);
 end;
end;


PROCEDURE PROGRAMA_INICIAL; { Esta subrotina recupera os valores iniciais obtidos em SOL_INICIAL e
                              transfere para a TELA_GRAFICA. Isto acontece quando o
                              modelo.GMS não converge }
begin
 RITMO := RITMOotm;
 for i:=1 to (NEV-1) do
 begin
  DE[i] := DEotm[i];
  DT[i,i+1] := DE[i] + dATIV[i,i+1] + RITMO*(N-1);
  DURACAO[i,i+1] := DT[i,i+1] - DE[i];
 end;
 DTOT := DT[NEV-1,NEV];
```

```
 DU := (DE[NEV-1]+dATIV[NEV-1,NEV]) - DE[1];
 for j:=1 to (cronograma) do
 begin
  SOMAY[j] := 0;
  for i:=1 to (NEV-1) do
  begin
   if ((DATACRON[j] - DE[i]) > 0) and ((DT[i,i+1] - DATACRON[j]) > 0) then
   begin
    Y[i,i+1,j] := ((DATACRON[j] - DE[i])/DURACAO[i,i+1])*Ct[i,i+1];
   end
   else
   begin
    if ((DATACRON[j] - DE[i] ≤ 0)) then
    begin
     Y[i,i+1,j] := 0;
    end
    else
    begin
     Y[i,i+1,j] := Ct[i,i+1];
    end;
   end;
   SOMAY[j] := SOMAY[j] + Y[i,(i+1),j];
  end;
 end;
end;


PROCEDURE SOMA_DELTAS; { Calcula a curva - não acumulada - de agregação de recursos da obra e o somatório das
                         diferenças entre as curvas }
begin
 SOMA[1] := SOMAY[1];
 for i:=2 to (cronograma) do
 begin
  SOMA[i] := SOMAY[i] - SOMAY[i-1];
 end;
 DIFSOMA := 0;
 for i:=1 to (cronograma) do
 begin
  DIFSOMA := DIFSOMA + ABS(PARCELA[i] - SOMA[i]);
 end;
end;


PROCEDURE SALVA_TELA(var buf : ptela);  { Salva a tela corrente para que possa ser vista quando se queira, sem a
                                          necessidade de rodar o programa novamente }
begin
 move(mem[$a000:0],mem[seg(buf^):ofs(buf^)],3*16384);
end;



PROCEDURE RESTAURA_TELA(buf : ptela);   { Restaura a tela salva na subrotina SALVA_TELA }

begin
 move(mem[seg(buf^):ofs(buf^)],mem[$a000:0],3*16384);
end;


PROCEDURE  TELA_GRAFICA;  { Constrói, a partir dos valores ótimos das variáveis de decisão -
                            capturados pela subrotina VALORES_OTIMOS, o gráfico da L.O.B.
                            e as curvas de agregação positivas e negativas do projeto - geradas pelo
                            'programa ótimo' obtido }
var
```

```
  EscalaX,EscalaY                          : word;
  dRITMO,AngMax                            : word;
  EscX,EscY                                : real;
  XDE,XDT                                  : array[1..(NEV-1),2..NEV,1..2] of integer;
  St                                       : string;
  Poly                                     : array[1..5] of pointtype;
  cor,cor1                    : byte;
  XDATA,YPARCELA,YSOMA,dPARCELA            : array[1..50] of integer;
  RMEDIO                                   : real; {Recurso percentual médio do projeto.}
  pt                                       : byte;

begin

  { Construção da Linha de Balanço }

  Desliga_Cursor;
  clrscr;
  driverVar := 9;
  modeVar := 2;
  INITGRAPH(driverVar,modeVar,'C:\TP\bgi');
  RECTANGLE(0,0,GetMaxX,GetMaxY);
  RECTANGLE(10,10,Trunc(GetMaxX/2),Trunc((GetMaxY-30)/2+10));
  EscalaX := (Trunc(GetMaxX/2)-30) - 30;
  EscX := EscalaX/DATACRON[cronograma];
  dRITMO := Trunc(RITMO*(N-1)*EscX);
  for i:=1 to (NEV-1) do
  begin
    XDE[i,i+1,1] := Trunc(DE[i]*EscX) + 30;
    XDT[i,i+1,1] := XDE[i,i+1,1] + dRITMO;
    XDE[i,i+1,2] := XDE[i,i+1,1] + TRUNC(dATIV[i,i+1]*EscX);
    XDT[i,i+1,2] := XDE[i,i+1,2] + dRITMO;
  end;
  AngMax := TRUNC(((ArcTan(Trunc((GetMaxY-30)/2-50)/dRITMO))*180)/PI);
  SETFILLSTYLE(0,0);
  PIESLICE(XDE[NEV-1,NEV,2],Trunc((GetMaxY-30)/2-10),0,AngMax,20);
  cor1 := GETCOLOR;
  cor := 9;
  for i:=1 to (NEV-1) do
  begin
    SETFILLSTYLE(1,cor);
    SETCOLOR(cor);
    Poly[1].X := XDE[i,i+1,1]; Poly[1].Y := Trunc((GetMaxY-30)/2+10)-20;
    Poly[2].X := XDT[i,i+1,1]; Poly[2].Y := 40;
    Poly[3].X := XDT[i,i+1,2]; Poly[3].Y := 40;
    Poly[4].X := XDE[i,i+1,2]; Poly[4].Y := Trunc((GetMaxY-30)/2+10)-20;
    Poly[5].X := XDE[i,i+1,1]; Poly[5].Y := Trunc((GetMaxY-30)/2+10)-20;
    FILLPOLY(5,Poly);
    cor := cor + 1;
  end;
  LINE(30,30,30,Trunc((GetMaxY-30)/2-10));
  LINE(30,Trunc((GetMaxY-30)/2-10),Trunc(GetMaxX/2)-20,Trunc((GetMaxY-30)/2-10));
  SETCOLOR(cor1);
  SETTEXTJUSTIFY(LeftText,BottomText);
  SETTEXTSTYLE(DefaultFont,HorizDir,1);
  OUTTEXTXY(Trunc(GetMaxX/2)+15,25,'ATIVIDADE        INÍCIO   TÉRMINO');
  cor := 9;
  for i:=0 to (NEV-2) do
  begin
```

```
  SETFILLSTYLE(1,cor);
  BAR(Trunc(GetMaxX/2)+15,40+25*i,Trunc(GetMaxX/2)+30,55+25*i);
  SETTEXTJUSTIFY(LeftText,BottomText);
  SETTEXTSTYLE(DefaultFont,HorizDir,1);
  SETCOLOR(cor1);
  OUTTEXTXY(Trunc(GetMaxX/2)+45,55+25*i,NomeAtiv[i+1]);
  STR(DE[i+1]:4:3,St);
  SETTEXTJUSTIFY(LeftText,BottomText);
  SETTEXTSTYLE(DefaultFont,HorizDir,1);
  OUTTEXTXY(Trunc(GetMaxX/2)+165,55+25*i,St);
  STR(DT[i+1,i+2]:4:3,St);
  SETTEXTJUSTIFY(LeftText,BottomText);
  SETTEXTSTYLE(DefaultFont,HorizDir,1);
  OUTTEXTXY(Trunc(GetMaxX/2)+245,55+25*i,St);
  inc(cor);
end;
SETLINESTYLE(CenterLn,0,NormWidth);
LINE(30,40,XDT[NEV-1,NEV,2],40);
SETLINESTYLE(CenterLn,0,NormWidth);
LINE(XDT[NEV-1,NEV,2],40,XDT[NEV-1,NEV,2],Trunc((GetMaxY-30)/2-10));
SETLINESTYLE(0,0,1);
STR(N:4,St);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+15,25*(NEV-2)+205,'Número de Unidades(N):');
OUTTEXTXY(Trunc(GetMaxX/2)+200,25*(NEV-2)+205,St);
OUTTEXTXY(Trunc(GetMaxX/2)+245,25*(NEV-2)+205,'Casas');
STR(1/RITMO:4:3,St);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+15,25*(NEV-2)+225,'Ritmo(R) :');
OUTTEXTXY(Trunc(GetMaxX/2)+105,25*(NEV-2)+225,St);
OUTTEXTXY(Trunc(GetMaxX/2)+160,25*(NEV-2)+225,'Casas/Dia');
STR(DU:4:3,St);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+15,25*(NEV-2)+245,'Duração Unitária:');
OUTTEXTXY(Trunc(GetMaxX/2)+160,25*(NEV-2)+245,St);
OUTTEXTXY(Trunc(GetMaxX/2)+230,25*(NEV-2)+245,'Dias');
STR(DTOT:4:3,St);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+15,25*(NEV-2)+265,'Término do Projeto(T):');
OUTTEXTXY(Trunc(GetMaxX/2)+200,25*(NEV-2)+265,St);
OUTTEXTXY(Trunc(GetMaxX/2)+270,25*(NEV-2)+265,'Dias');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(20,40,'N');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(XDT[NEV-1,NEV,2]-2,Trunc((GetMaxY-30)/2+5),'T');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(XDE[NEV-1,NEV,2]+25,Trunc((GetMaxY-30)/2-20),'R');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,VertDir,1);
OUTTEXTXY(25,Trunc((GetMaxY-30)/2-20),'Unidades Repetitivas');
```

```
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(90,Trunc((GetMaxY-30)/2+10)-5,'Unidade de Tempo');

{ Construção das Curvas de Agregação }

RECTANGLE(10,Trunc((GetMaxY-30)/2+20),Trunc(GetMaxX/2),GetMaxY-10);
LINE(30,Trunc((GetMaxY-30)/2+40),30,GetMaxY-30);
LINE(30,GetMaxY-30,Trunc(GetMaxX/2)-20,GetMaxY-30);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(20,Trunc((GetMaxY-30)/2+45),'%');
RMEDIO := 1/cronograma;
EscalaY := (GetMaxY-30) - (Trunc((GetMaxY-30)/2+10)+30) - 10;
EscY := EscalaY/(2*RMEDIO);
pt := Trunc(200*RMEDIO);
for i:=0 to (pt) do
begin
 STR(i:2,St);
 SETTEXTJUSTIFY(LeftText,CenterText);
 SETTEXTSTYLE(DefaultFont,HorizDir,1);
 OUTTEXTXY(13,(GetMaxY-30)-TRUNC(i*EscY/100),St);
end;
for i:=1 to (cronograma) do
begin
 XDATA[i] := TRUNC(DATACRON[i]*EscX) + 30;
 YPARCELA[i] := (GetMaxY-30) - TRUNC(PARCELA[i]*EscY);
 YSOMA[i] := (GetMaxY-30) - TRUNC(SOMA[i]*EscY);
 dPARCELA[i] := (GetMaxY-30) - ABS(YPARCELA[i] - YSOMA[i]);
end;
SETTEXTJUSTIFY(CenterText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(TRUNC(DATACRON[1]*EscX)+30+1,GetMaxY-18,'1');
for i:=1 to (cronograma) do
begin
 case (cronograma) of
 12:begin
    STR(i:2,St);
    SETTEXTJUSTIFY(CenterText,BottomText);
    SETTEXTSTYLE(DefaultFont,HorizDir,1);
    OUTTEXTXY(TRUNC(DATACRON[i]*EscX)+30-2,GetMaxY-18,St);
   end;
 18:begin
    if i≤9 then
    begin
     STR(i*2:2,St);
     SETTEXTJUSTIFY(CenterText,BottomText);
     SETTEXTSTYLE(DefaultFont,HorizDir,1);
     OUTTEXTXY(TRUNC(DATACRON[i*2]*EscX)+30-2,GetMaxY-18,St);
    end;
   end;
 24:begin
    if i≤12 then
    begin
     STR(i*2:2,St);
     SETTEXTJUSTIFY(CenterText,BottomText);
     SETTEXTSTYLE(DefaultFont,HorizDir,1);
```

```
      OUTTEXTXY(TRUNC(DATACRON[i*2]*EscX)+30-2,GetMaxY-18,St);
     end;
    end;
 30:begin
     if i≤10 then
     begin
      STR(i*3:2,St);
      SETTEXTJUSTIFY(CenterText,BottomText);
      SETTEXTSTYLE(DefaultFont,HorizDir,1);
      OUTTEXTXY(TRUNC(DATACRON[i*3]*EscX)+30-2,GetMaxY-18,St);
     end;
    end;
 36:begin
     if i≤12 then
     begin
      STR(i*3:2,St);
      SETTEXTJUSTIFY(CenterText,BottomText);
      SETTEXTSTYLE(DefaultFont,HorizDir,1);
      OUTTEXTXY(TRUNC(DATACRON[i*3]*EscX)+30-2,GetMaxY-18,St);
     end;
    end;
 end;
end;
for i:=1 to (cronograma-1) do
begin
 SETLINESTYLE(SolidLn,0,NormWidth);
 LINE(XDATA[i],YPARCELA[i],XDATA[i+1],YPARCELA[i+1]);
end;
SETLINESTYLE(SolidLn,0,NormWidth);
LINE(Trunc(GetMaxX/2)+15,25*(NEV-2)+100,Trunc(GetMaxX/2)+35,25*(NEV-2)+100);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+60,25*(NEV-2)+103,'Curva de Recursos Disponíveis');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+60,25*(NEV-2)+133,'Curva de Recursos Necessários');
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+60,25*(NEV-2)+168,'Diferença Absoluta');
STR(100*DIFSOMA:4:2,St);
SETTEXTJUSTIFY(LeftText,BottomText);
SETTEXTSTYLE(DefaultFont,HorizDir,1);
OUTTEXTXY(Trunc(GetMaxX/2)+15,25*(NEV-2)+305,'Somatório da Diferença:');
OUTTEXTXY(Trunc(GetMaxX/2)+215,25*(NEV-2)+305,St);
OUTTEXTXY(Trunc(GetMaxX/2)+265,25*(NEV-2)+305,'%');
SETCOLOR(10);
SETLINESTYLE(CenterLn,0,3);
LINE(Trunc(GetMaxX/2)+15,25*(NEV-2)+130,Trunc(GetMaxX/2)+35,25*(NEV-2)+130);
SETCOLOR(12);
SETLINESTYLE(DottedLn,0,3);
LINE(Trunc(GetMaxX/2)+15,25*(NEV-2)+165,Trunc(GetMaxX/2)+35,25*(NEV-2)+165);
for i:=1 to (cronograma-1) do
begin
 SETCOLOR(10);
 SETLINESTYLE(CenterLn,0,3);
 LINE(XDATA[i],YSOMA[i],XDATA[i+1],YSOMA[i+1]);
 SETCOLOR(12);
```

```
  SETLINESTYLE(DottedLn,0,3);
  LINE(XDATA[i],dPARCELA[i],XDATA[i+1],dPARCELA[i+1]);
  RECTANGLE(Trunc(GetMaxX/2)+205,25*(NEV-2)+285,Trunc(GetMaxX/2)+285,25*(NEV-2)+315);
 end;
 repeat until keypressed;
 Liga_Cursor;
 salva_tela(tela1);
 assign(arq4,'C:\PENHA\'+projeto+'.TEL');
 rewrite(arq4,16384);
 for i:= 0 to 2 do
  blockwrite(arq4,mem[seg(tela1^):ofs(tela1^)+i*16384],1);
 close(arq4);
 CLOSEGRAPH;
 Clrscr;
end;


PROCEDURE SIMULACAO; { Esta subrotina compreende a simulação propriamente dita, contendo as subrotinas de
                       SUPORTE, SOL_INICIAL, MONTAGEM1 e 2,execução dos modelos.GMS,
                       VALORES_OTIMOS e TELA_GRAFICA }
begin
 Aguarde;
 Suporte;
 Sol_Inicial;
 Montagem1;
 Montagem2;
 SwapVectors;
 EXEC(getenv('comspec'),'/C GAMSW C:\PENHA\SIS S=AUXILIAR OPTFILE=1 O=C:\PENHA\SIS.LST');
 EXEC(getenv('comspec'),'/C GAMSW C:\PENHA\SIS_OTM R=AUXILIAR PS=9999 O=C:\PENHA\SIS_OTM.LST');
 SwapVectors;
 Status_Solver;
 if (status = 1) or (status = 2) then
 begin
  Valores_Otimos;
  Soma_Deltas;
  if (DIFSOMA ≤ SQRDIFmin) then
  begin
   Tela_Grafica;
  end
  else
  begin
   Programa_Inicial;
   Soma_Deltas;
   Tela_Grafica;
  end;
 end
 else
 begin
  Programa_Inicial;
  Soma_Deltas;
  Tela_Grafica;
 end;
end;


PROCEDURE CARREGA_VARIAVEIS; { Esta subrotina lê os valores dos dados no arquivo 'projeto'.DAT para
                               serem vistos ou alterados para nova simulação do mesmo projeto }
begin
 assign(arq6,'C:\PENHA\'+projeto_aux+'.DAT');
 reset(arq6);
 readln(arq6,N_aux,RITMOMAX_aux);
```

```
 for i:=1 to (NEV-1) do
 begin
  case i of
  1: readln(arq6,dATIVaux[i,i+1],Ct_aux[i,i+1]);
  2..(NEV-1): readln(arq6,dATIVaux[i,i+1],Ct_aux[i,i+1],BUFFERp_aux[i]);
  end;
 end;
 readln(arq6,cronograma_aux,periodo_aux);
 for i:=1 to (cronograma_aux) do
 begin
  readln(arq6,PARCELA_aux[i]);
 end;
 close(arq6);
end;


PROCEDURE PROCURA_PROJETO; { Esta subrotina procura o nome do projeto, dado pelo usuário, no arquivo
                             NOMES.DAT }

begin
 assign(arq0,'C:\PENHA\NOMES.DAT');
 reset(arq0);
 str8:= '';
 while (not EoF(arq0)) and (str8 < > projeto_aux) do
  readln(arq0,str8);
 close(arq0);
end;


PROCEDURE OPCAO_SIMULAR; { Esta subrotina compreende todas as subrotinas necessárias a simulação }


begin
 Procura_Projeto;
 if (str8=projeto_aux) then
 begin
  Carrega_Variaveis;
  Confirma_Altera;
  Simulacao;
 end
 else
 begin
  Entrada_Dados;
  Grava_Dados0;
  Simulacao;
 end;
end;


PROCEDURE VER_DADOS; { Esta subrotina usa CARREGA_DADOS para ler o arquivo de dados 'projeto'.DAT e
                       depois dispõe estes dados na tela }

begin
 Carrega_Variaveis;
 Confirma_Dados1;
 writeln;
 write('Pressione qualquer tecla para continuar ! ');
 ch:= readkey;
 Confirma_Dados2;
 writeln;
 write('Pressione qualquer tecla para continuar ! ');
 ch:= readkey;
end;
```

```
PROCEDURE VER_PROGRAMA; { Esta subrotina recupera a programação do projeto - a tela gráfica gravada em
                                        'projeto'.TEL }
begin
 clrscr;
 INITGRAPH(driverVar,modeVar,'C:\TP\bgi');
 assign(arq4,'C:\PENHA\'+projeto+'.TEL');
 reset(arq4,16384);
 for i:= 0 to 2 do
   blockread(arq4,mem[seg(tela1^):ofs(tela1^)+i*16384],1);
 close(arq4);
 Restaura_Tela(tela1);
 repeat until keypressed;
 CLOSEGRAPH;
end;


PROCEDURE OPCAO_VER; { Esta subrotina compreende todas as outras necessárias a que o usuário possa ver os dados
                                    e/ou a programação de determinado projeto }
begin
 clrscr;
 Procura_Projeto;
 if (str8=projeto_aux) then
 begin
  writeln;
  writeln;
  writeln;
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln('                O QUE DESEJA VER ?                                  ');
  writeln('                ------------------                                  ');
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln('                  1   OS DADOS DO PROJETO                           ');
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln('                  2   A PROGRAMAÇÃO DO PROJETO                      ');
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln('                  3   AMBOS                                         ');
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln('                                                                    ');
  writeln;
  repeat
   write('          Digite a sua opção : ');
   readln(ch);
   if (ch = '1') or (ch = '2') or (ch = '3') then
   begin
    val(ch,opcao1,code);
   end
   else
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until (ch = '1') or (ch = '2') or (ch = '3');
  case (opcao1) of
  1: Ver_Dados;
  2: Ver_Programa;
  3:
```

```
    begin
      Ver_Dados;
      Ver_Programa;
    end;
  end;
 end
 else
 begin
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln('                                    ATENÇÃO!');
  writeln('                                    ————————');
  writeln;
  writeln('          ESTE PROJETO AINDA NÃO FOI CADASTRADO.');
  writeln('          VERIFIQUE SEU NOME E DIGITE CORRETAMENTE!');
  delay(2700);
 end;
end;


(*********************PROGRAMA PRINCIPAL************************)

BEGIN
 New(tela1);
 repeat
  Usuario;
  Nome_Projeto;
  case (opcao0) of
  1: Opcao_Simular;
  2: Opcao_Ver;
  end;
  clrscr;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  liga_cursor;
  repeat
   write('          O USUÁRIO DESEJA TERMINAR (S/N) ? ');
   readln(sai);
   if (sai<>'S') or (sai<>'s') or (sai<>'N') or (sai<>'n') then
   begin
    gotoxy(whereX,whereY-1);
    DelLine;
   end;
  until (sai='S') or (sai='s') or (sai='N') or (sai='n');
```

```
  until (sai='S') or (sai='s');
  Dispose(Tela1);
  clrscr;
END.
```