Universidade Federal da Paraíba

Pró - Reitoria para Assuntos do Interior

Centro de Ciências e Tecnologia

Departamento de Engenharia Civil

Relatório final do estágio supervisionado

Sistema de apoio à decisão para operação de reservatórios

Bolsa de Iniciação Cientifica Institucional

Orientador: Carlos de Oliveira Galvão

Aluna: Emilia Rahnemay Rabbani matrícula: 9011234-1

Agosto de 1994

Universidade Federal da Paraíba

Pró - Reitoria para Assuntos do Interior

Centro de Ciências e Tecnologia

Departamento de Engenharia Civil

Relatório final do estágio supervisionado

Sistema de apoio à decisão

para operação de reservatórios

Carlos Galvão

Carlos de Oliveira Galvão
(orientador)

Emília Rahnemay Rabbani

Emilia Rahnemay Rabbani
(estagiária)

Agosto de 1994

Índice

1 - Apresentação

Este trabalho teve inicio em setembro de 1993, sob orientação do professor Carlos O. Galvão da Área de Recursos Hídricos do Departamento de Engenharia Civil do Centro de Ciências e Tecnologia da UFPB - Campus II.

O trabalho faz parte de um projeto de pesquisa que tem como objetivo desenvolver um sistema para auxílio à decisão (SAD) sobre o controle de operação de reservatórios de acumulação de água. A partir das informações de previsão meteorológica de curto, médio e longo prazos, o SAD estimará a vazão afluente aos reservatórios. Estas estimativas, ao lado de dados monitorados sobre os reservatórios, comporão o cenário para a decisão sobre a estratégia de operação.

De acordo com o plano de trabalho proposto para o estágio supervisionado o trabalho se constituiria em uma revisão bibliográfica sobre operação de reservatórios, sistemas de apoio à decisão e métodos de programação matemática; de um treinamento em ferramentas computacionais de processamento de texto, banco de dados e processamento gráfico e da preparação, implantação em computador e testes de algoritmos baseados em programação dinâmica para operação de reservatórios.

Este relatório refere-se ao período completo do estágio supervisionado. Todas as etapas planejadas foram cumpridas.

2 - Introdução

O aumento das demandas de água nos últimos anos, seja para uso doméstico das populações urbanas, usos industriais, de irrigação, ou diluição de poluentes, não tem sido acompanhado do aumento da oferta de água pelo aproveitamento de novos mananciais ou ampliação dos já existentes. Este fato, associado muitas vezes a problemas de natureza climática, como a irregularidade de chuvas, tem gerado escassez e provocado medidas emergenciais de redução na distribuição aos consumidores. Este desequilíbrio entre oferta e demanda na área dos recursos hídricos, tem imposto a necessidade de soluções cada vez mais elaboradas (Braga, 1987).

No caso do semi-árido brasileiro, a má distribuição anual das chuvas, aliada a temperatura e evaporação elevadas, ocasiona diversos problemas de ordem sócio-econômica. Nestas circunstâncias, de recursos hídricos escassos, o manejo dos mananciais precisa receber atenção especial.

O planejamento dos recursos hídricos deve levar em consideração múltiplos usuários, múltiplos propósitos e múltiplos objetivos. Pessoas diferentes têm objetivos, perspectivas e valores diferentes. Planejar para um máximo de benefícios econômicos líquidos não é suficiente. A distribuição igual dos riscos, redistribuição das riquezas naturais, qualidade do meio ambiente e o bem estar social são tão importantes como os benefícios econômicos. Se torna claramente impossível o desenvovimento de um único objetivo que satisfaça todos os interesses, todas as adversidades e todos pontos de vistas sociais e políticos (Loucks et al., 1981).

A tarefa de um gerente de sistema de reservatório de água pode ser descrita de modo grosseiro como a identificação e o desenvolvimento de um possível modelo de sistema de reservatório de água ou plano de gerenciamento e avaliar os seus impactos: econômico, ecológico, ambiental e social. O desenvolvimento de um modelo para qualquer problema específico de recursos hídricos depende de: (1) objetivos da análise; (2) a data que se deseja avaliar o projeto; (3) o tempo, dinheiro e as facilidades computacionais disponíveis para a análise; (4) o conhecimento e habilidade do modelador. O desenvolvimento de um modelo é uma arte, requer julgamento abstrato dos componentes do mundo real que são importantes para a decisão a ser feita e que pode ser iluminada pelos metodos quantitativos, e requer julgamento em expressar aqueles componentes e suas interrelações matemáticas na forma de um modelo (Loucks et al., 1981).

3 - Revisão de literatura

Procuramos nos inteirar dos trabalhos mais recentes feitos no Brasil sobre a utilização de programação dinâmica na operação de reservatórios. Foi feita uma revisão de literatura consultando todos os Anais dos Simpósios Brasileiros de Recursos Hídricos entre 1981 e 1993, sobre o assunto. Foram encontrados 9 trabalhos, sendo que a maioria trata da programação dinâmica (seja ela simples, estocástica ou estocástica dual) de operação de sistemas hidrelétricos. Eles são os de: Cunha (1985), Costa et al. (1989), Silveira et al. (1991), Reis e Chaudhry (1991a), Reis e Chaudhry (1991b) e Soares et al. (1993). Esses trabalhos utilizam dados de período imediatamente anterior (modelos regressivos, autoregressivos, etc). Percebe-se ainda que nesses trabalhos há a possibilidade de utilização de intervalos de tempos diferentes, podendo ser mensal, anual, sanzonal ou semanal, e da divisão do horizonte de planejamento em intervalos de curto, médio e longo prazos.

Cunha (1985) faz uma análise conjunta de dois reservatórios equivalentes operando-os de modo a minimizar a soma do valor esperado dos vertimentos. O trabalho mostra a utilização de programação dinâmica estocástica simplificada, baseado no modelo desenvolvido no Centro de Hidráulica e Hidrologia Prof. Parigot de Souza - CEHPAR (1982), para a verificação do desempenho de sistemas interligados.

Costa et al. (1989) utilizam a programação dinâmica estocástica na operação de sistema hidrotérmicos a fim de minimizar o valor do cuto de operação ao longo do tempo. Apresenta portanto uma função objetivo baseada em custos, levando em conta os custos dos combustíveis nas unidades térmicas e eventuais penalidades pelo não atendimento a demandas. O modelo foi aplicado no sistema Sul/Sudeste composto de 39 usinas hidrelétricas sendo 22 com reservatórios e 17 a fio d'água.

Silveira et al. (1991) mostra a operação de um sistema hidrelétrico com embasamento em reservatórios equivalentes. Duas metodologias são empregadas: a simulação e a otimização, com uso de programação dinâmica. A otimização garante a operação mais econômica enquanto que a simulação garante o atendimento da demanda.

No trabalho de Reis e Chaudhry (1991a) a operação do sistema hidrelétrico do Rio Tietê é enfocada com vistas a sua caracterização estocástica, utilizando-se programação dinâmica estocástica dual (PDED). O problema de otimização foi expresso como o de minimização de custo esperado do importe de energia no horizonte de planejamento.

Reis e Chaudhry (1991b) abordam a questão da variabilidade das respostas obtidas para operação de reservatórios pela PDED proposta por Pereira e Pinto (1988). O trabalho tem como objetivo analisar as respostas ótimas obtidas pela programação dinâmica estocástica em termos de decisões de liberações na operação do sistema hidrelétrico do rio Tietê.

Soares et al. (1993) utilizam o modelo de reservatório equivalente como uma manipulação que permite o uso de programação dinâmica em planejamento de longo prazo da operação hidrotérmica, contornando assim a conhecida "maldição da dimensionalidade" da programação dinâmica que reduz a aplicação prática desta técnica a sistemas com um reduzido número de reservatórios.

O trabalho de Molle et al. (1985) apesar de se basear na simulação, fala sobre o funcionamento de pequenos reservatórios com múltiplos usos explorando a potencialidade hidroagrícola.

Os trabalhos de Zahed (1987) e Born e Stedinger (1989) fazem uma revisão bibliográfica dando definições básicas dos diferentes tipos de modelos existentes para operação de um sistema de reservatórios, inclusive programação dinâmica.

As funções objetivo apresentadas eram bastante específicas e relacionadas a sistemas hidrelétricos.

Além destes Anais foram consulatados as teses de mestrado de Zahed (1984) e Ribeiro (1990), os artigos de Harboe (1983) e Braga (1987) e livro de Loucks et al. (1981).

O trabalho de Zahed descreve e classifica diversas metodologias (otimização e simulação) utilizadas para o dimensionamento e a operação de reservatórios simples ou de sistemas com diversos reservatórios. Ele conclui que a extensa e variada bibliografia sobre o assunto revela a inexistência de um algoritmo geral para a solução do problema. A escolha do método depende das características do sistema, da disponibilidade de dados, dos objetivos e das restrições exitentes no sistema.

Ribeiro (1990) apresenta um modelo genérico de simulação da operação de reservatórios aplicado a um sistema composto de dois reservatórios em série, Engenheiro Ávidos e São Gonçalo, localizado na região semi-árida da Paraíba, com a finalidade de avaliar a disponibilidade dos recursos hídricos superficiais e indicar critérios de operação que ajudassem a diminuir a escassez hídrica. O programa utilizado foi o HEC-3.

Loucks et al. (1981) fazem uma revisão dos métodos comumente usados pelos planejadores dos sistemas de recursos hídricos. Estes métodos incluem otimização matemática e técnicas de simulação. A programação dinâmica foi aplicada para três problemas de planejamento de recursos hídricos: alocação de água, operação de reservatórios e expansão da capacidade.

Braga (1987) apresenta técnicas de otimização e simulação aplicadas em sistemas de recursos hídricos. No trabalho foram introduzidos noções básicas da programação matemática aplicada ao planejamento e gestão de sistemas de recursos hídricos, foram enfatizados aspectos básicos da programação linear, programação dinâmica e simulação no sentido de possibilitar análises detalhadas em termos probabilísticos e foram ainda tratados os problemas de recursos hídricos através da consideração de múltiplos objetivos.

Harboe (1983) analisa a operação ótima de reservatórios utilizando programação dinâmica. Mostra dois tipos de problemas que nos dão diferentes funções objetivos. A função objetivo max-min, cujo desenvolvimento matemático do modelo é apresentado, é aplicada a um problema de operação ótima em reservatórios com hidroelétricas para múltiplos propósitos.

4 - Materiais e Métodos

4.1 - Informações hidrometeorológicas da Paraíba

Com o objetivo de facilitar a utilização do modelo proposto para o SAD formou-se um banco de dados composto pelas informações da Paraíba que serão utilizados no modelo. As informações previstas para armazenamento no banco de dados foram os dados sobre pluviometria, evaporação e vazão.

Pluviometria

Os dados de pluviometria foram retirados do diagnóstico recente realizado na rede pluviométrica da Paraíba (Silva e Galvão, 1993), no qual as séries históricas da maioria dos postos do estado foram fornecidos pelo Departamento Nacional de Águas e Energia Elétrica (DNAEE). Temos a seguir um exemplo do formato dos dados (Figura 1).

Posto Ano Mês	dia1	02	03 ...						
006350271972 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
006350271972 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	[illegible]
006350271972 3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
006350271972 4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0	0.
006350271972 5	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	0.[illegible]
006350271972 6	0.0	6.0	12.0	0.0	0.0	0.0	0.0	0.0	32.[illegible]
006350271972 7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22.[illegible]
006350271972 8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
006350271972 9	0.0	0.0	0.0	0.0	8.0	0.0	0.0	3.0	0
0063502719721 0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	[illegible]
0063502719721 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	[illegible]
0063502719721 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	[illegible]
006350271973 1	0.0	0.0	0.0	0.0	66.6	10.4	0.0	0.0	0
006350271973 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	0.
006350271973 3	0.0	0.0	0.0	0.0	39.6	0.0	0.0	0.0	22.8
006350271973 4	0.0	0.0	0.0	0.0	9.2	0.0	0.0	0.0	35.6
006350271973 5	0.0	0.0	0.0	0.0	25.2	0.0	0.0	18.4	13.2
006350271973 6	0.0	0.0	23.6	0.0	0.0	9.2	10.8	0.0	0.
006350271973 7	0.0	8.2	0.0	52.2	0.0	0.0	0.0	3.0	0
006350271973 8	0.0	0.0	0.0	0.0	0.0	8.4	0.0	0.0	[illegible]
006350271973 9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	[illegible]
0063502719731 0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	[illegible]
0063502719731 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
[illegible]	[illegible]	0.0	0.0	[illegible]	[illegible]	0.0	[illegible]	[illegible]	[illegible]

Figura 1. Formato dos dados de pluviometria.

Evaporação

O Núcleo de Meteorologia aplicada do Departamento de Ciências Atmosféricas da UFPB (NMA/DCA) forneceu os dados de 4 estações evaporimétricas. A figura 2 apresenta um exemplo de formato dos dados de evaporação.

```
SAO GONCALO
CODIGO-82689  ANO-  61
 1   7.0  3.0  1.0  1.5  1.1  5.7  4.5  7.1  8.3  8.3  8.1  8.4
 2   6.0  2.1  3.0  3.2  0.6  5.9  5.6  8.1  6.6  7.9 10.3  8.4
 3   6.5  2.9  5.0  2.9  0.9  5.6  6.4  8.2  8.5  8.7  8.4  6.2
 4   6.0  3.1  5.6  2.9  3.0  3.8  7.1  7.8  7.7  7.7  7.5  4.8
 5   7.0  2.8  2.4  2.8  3.1  4.2  6.7  6.8  6.1  6.9  5.1  5.7
 6   6.2  2.7  5.0  2.2  3.3  5.1  7.2  7.4  7.2  7.8  2.4  6.9
 7   7.0  2.8  5.1  2.1  3.7  6.6  8.2  7.5  7.4  9.1  5.2  7.4
 8   5.5  3.7  3.9  1.9  3.9  4.0  8.4  7.8  6.8  8.8  8.6  6.6
 9   4.7  1.1  3.0  2.0  2.1  4.9  8.7  7.6  6.5  7.6  9.3  5.8
10   5.5  2.9  5.0  4.0  2.9  6.0  8.8  7.9  6.7  8.2  9.8  6.6
[illegible]  [illegible]  2.0  [illegible]  [illegible]  [illegible]  [illegible]  8.5  [illegible]  [illegible]  [illegible]  [illegible]  4.8

30   5.0 99.9  1.0  2.0  [illegible]  [illegible]  [illegible]  8.8  [illegible]  [illegible]  8.1  7.1
31   2.0 99.9  2.5 99.9  5.1 99.9  8.3  7.9 99.9  8.0 99.9  7.5
CODIGO-82689  ANO-  62
 1   7.9  1.1  0.5  1.5  3.2  3.3  3.0  8.0  8.8  9.2  8.0  6.4
 2   7.7  1.5  0.9  1.9  3.6  2.8  3.8  7.3  8.6  8.7  6.0  6.6
 3   7.8  2.9  1.2  2.4  2.6  2.4  4.3  8.4  9.0  9.2  6.6  7.9
 4   6.8  4.0  1.5  1.5  3.6  3.1  5.0  8.8  8.2  8.8  6.4  7.2
 5   7.0  5.0  1.6  1.4  3.5  3.6  6.1 10.0  8.7  8.0  6.0  6.8
 6   7.6  4.0  1.3  1.1  2.7  3.8  6.4  9.1  8.1  7.7  5.6  6.9
[illegible]  [illegible]  [illegible]  [illegible]  [illegible]  [illegible]  [illegible]  5.0  [illegible]  [illegible]  [illegible]  4.5  5.8
 ↑     ↑    ↑    ↑
dia  jan. fev. mar. ...
```

Figura 2. Formato dos dados de evapotranspiração

Vazão

Os dados de vazão foram retiradas do levantamento recente das estações fluviométricas (Ribeiro, 1993) no qual DNAEE e SUDENE forneceu os dados das estações evaporimétricas. Um exemplo do formato dos dados está mostrado na figura 3.

```
Posto   Ano Mês dia01  02   03 ...
372600001969 1  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
372600001969 2  0.000 0.000 0.000 0.000 0.000 0.006 0.000 0.000 0.00
372600001969 3  0.000 0.000 0.979  5.78  1.38 0.119 0.000 0.000 0.00
372600001969 4  0.940  3.37  4.21  31.5  49.1  48.7  51.5  69.9  52.6
372600001969 5  0.160 0.221 0.240 0.140 0.100 0.100 0.100 0.260 0.100
372600001969 6   1.12  2.30  2.21  1.20 0.279 0.279 0.619 0.498 0.402
372600001969 7  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001969 8  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001969 9  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001969 10 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001969 11 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
372600001969 12 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.0
372600001970 1  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.0
372600001970 2  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
372600001970 3  0.000 0.000  1.10  6.69  4.68  10.9  37.2  33.9  16.5
372600001970 4  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001970 5  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001970 6  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001970 7  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
372600001970 8  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
372600001970 9  0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.00
372600001970 10 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.0
372600001970 11 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.0
```

Figura 3. Formato dos dados de vazão.

4.2 - Informações sobre reservatórios da Paraíba

O sistema para apoio a decisão (SAD), a fim de facilitar sua utilização, precisa de várias informações sobre os reservatórios da Paraíba como: capacidade, volume morto, curva cota-área-volume e séries históricas de vazões afluentes ao reservatório.

Estudo realizado pela Área de Recurso Hídricos da UFPB (ATECEL, 1993) sistematizou dados de 24 açudes do Estado. Estes dados foram incorporados ao banco de dados para o SAD. A tabela 1 os dado dos açudes e a Figura 4 mostra um exemplo do formato dos dados. No anexo 1 há arquivos de dados do reservatório Engenheiro Arcoverde.

Tabela 1. Açudes com dados para o SAD.

AÇUDE	VOLUME MÁXIMO (m^3)
Est. Marinho	720.000.000
Mãe d'agua	638.700.000
Eng. Avidos	255.000.000
Saco	98.000.000
Lagoa do Arroz	80.220.750
Capoeira	53.450.000
São Gonçalo	44.600.000
Eng. Arcoverde	36.834.000
Carneiro	31.285.875
Tapera	26.418.660
Riacho dos Cavalos	17.699.000
Bartolomeu I	17.570.556
Jatoba I	17.516.000
Escondido I	16.579.000
São Mamede	15.791.280
Queimadas	15.625.338
Timbauba	15.438.573
Santa Luzia	11.960.250
Vazante	9.091.200
Video	6.040.263
São Francisco	4.920.720
Martelo	4.567.800
Frutuoso II	3.517.220
Caraibeira	2.709.260

```
T1                      Plano Est. Rec. Hidricos
T2                            ENG. AVIDOS
T3                           setembro 1993
RS  4989   12259   13275   39232   54483   71410  106989  137896  175830  255000 ┐→ curva volume (mil m³)/
RA  1734    2919    4440    6413    8060   10045   13814   16978   20772   28449 │  área (mil m²)/cota (m)
RE   294     297     300     303     305     307     310     312     314   317.2 ┘
IN   963  1.86 16.55 28.01  4.95  1.16   .39   .00   .00   .02   .17   .37  1.10 → vazão mensal afuente
YE    63   201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5   ao açude (IN)
IN   964  2.48  9.40 12.99 17.06  4.75   .48   .05   .29   .49   .11   .17   .06
YE    64   201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5 → evaporação mensal-YE
IN   965  1.13   .32  6.64 22.05  1.80  2.59   .20   .26   .07   .34   .06   .01
YE    65   201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5
IN   966  1.22  1.59   .82  2.16   .97   .73   .21   .09   .15   .14   .33   .46
YE    66   201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5
IN   967  1.15  9.75 18.69 34.25 22.31   .13   .07   .03   .01   .00   .01   .31
YE    67   201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5
IN   968  3.29   .45 26.88  3.58  2.25   .26   .09   .01   .00   .01   .37   .28
YE    68   201 152.3 145.5 129.8 133.5 138.8
           ↑     ↑     ↑     ↑
          ano   jan.  fev.  mar. ...
```

Figura 4. Formato dos dados sobre reservatórios.

4.2.1 - Reservatório Engenheiro Arcoverde

Testes de validação do modelo serão feitos num caso real, para tanto foi escolhido o reservatório Engenheiro Arcoverde, situado no semi-árido da Paraíba, no município de Condado. O reservatório tem 36 milhões de m³ de capacidade (dos quais 32 milhões de m³ úteis) abastece demandas de abstecimento urbano e irrigação. O açude é administrado pelo DNOCS.

A região apresenta clima semi-árido quente mediano, com 7 a 8 meses secos (Nimar, 1989 citado por Ribeiro, 1990), a bacia é caracterizada por uma má distribuição anual das chuvas que aliada a temperatura e evaporação elevadas ocasiona os problemas de ordem sócio econômicos peculiares ao sem-árido nordestino.

O período chuvoso tem início em janeiro estendendo-se até maio, sendo o trimestre mais chuvoso fevereiro até abril. O restante do ano a região fica sobre estiagem prolongada, agosto até outubro, representam os meses mais secos (Ribeiro, 1990).

4.3 - Banco de dados

Os formatos de dados deverão ser mantidos como entrada para o SAD, de modo que não seja necessário a conversão de formatos dos dados originais. Assim, os dados provenientes das instituições fornecedoras (DNAEE, DMA/DCA) podem ser imediatamente incorpordos ao banco de dados.

O programa de leitura de dados do SAD, em desenvolvimento por outros membros da equipe do projeto, deverá aceitar esses diferentes formatos.

4.4 - Definição de regras para operação de reservatórios

No processo de escolha da alternativa ótima (ou conjunto ótimo de regras), ao longo dos anos, várias técnicas foram desenvolvidas, sendo as mais conhecidas: a programação linear, a programação dinâmica e a simulação. Essas técnicas podem ser utilizadas na operação de um reservatório,.

A simulação é a técnica mais utilizada na prática e está baseada na tentativa e erro para identificar as soluções aproximadamente ótimas. A dificuldade com a técnica de simulação é que geralmente há um número frustrante de soluções ou planos praticáveis. Ainda quando combinada com técnicas eficientes para a escolha do valor de cada variável de decisão, um enorme esforço computacional provavelmente nos levará a uma solução que ainda estará longe do melhor possível.

A programação linear e a programação dinâmica podem ser classificados como técnicas de otimização. A solução desses tipos de procedimento depende em alto grau de uma estruturação matemática do modelo de gerenciamento e requerem um algoritmo para encontrar valores ótimos das variáveis de decisão (Harboe, 1987).

Para se definir entre um modelo otimizante ou um modelo de simulação deve-se considerar a complexidade do sistema estudado. Casos simples onde o conhecimento dos fenômenos a serem tratados é suficiente para escrever equações que descrevem e governam o sistema, podem ser muito bem tratados por um modelo otimizante. Para sistemas complexos que não podem ser analisados diretamente através das metodologias analíticas formais, usa-se a simulação. No caso de um sistem simples, onde se deseja um maior conhecimento do comportamento do sistema, parte-se para a simulação. Uma concepção geral, que vem sendo formada, é que os modelos de simulação e otimização devem ser entendidos como técnicas complementares para auxilio no processo de análise de sistemas de recursos hídricos (Labadie, 1978; Zahed, 1984; Braga, 1987 citado por Ribeiro, 1990).

Neste trabalho utilizaremos a programação dinâmica que é um dos métodos mais usados no planejamento de sistemas de reservatórios de água.

Os típicos modelos de planejamento geralmente incluem pelo menos uma função objetivo que deve ser maximizada ou minimizada que serve para classificar as soluções ou planos alternativos. Em quase todos os casos a função objetivo é uma função escalar, onde a dimensão de cada termo deve ser homogêneo, não importando quantos termos estão incluidos.

Além do objetivo os problemas de planejamento incorporam um número de requesitos que são formulados como restrições.

A solução ótima de um problema de planejamento é um plano que atinja os maiores (ou menores) valores do objetivo ao mesmo tempo que satisfaça as restrições.

As restrições podem ser de dois tipos. Um tipo de restrição expressa a limitação física que não pode ser violada a nenhum custo. O segundo tipo de restrição é em certos casos um objetivo implícito ou meta que de fato pode ser violada, mesmo que o custo de tal violação possa ser alto.

4.5 - Programação dinâmica

Em muitas situações, o problema de otimização em recursos hídricos é dado por uma seqüência de decisões que evoluem no tempo e no espaço (Braga, 1987).

Entende-se por sistema dinâmico qualquer sistema que evolua portanto tanto no tempo como no espaço.

A idéia básica da programação dinâmica é subdividir o problema inicial em um conjunto de problemas mais simples ao invés de tentar resolver o problema complexo de uma vez. Cada problema simples é membro de uma classe de problemas similares e não um problema isolado para o estágio inicial e estado inicial. Portanto, o enfoque é encaixar o problema inicial em uma família de problemas similares. A utilidade desse enfoque dependerá da viabilidade de se encontrar um encaixe tal que: 1) pelo menos um membro da família tenha uma solução simples; e 2) seja possível estabelecer relações funcionais entre os vários membros da família (Braga, 1987).

4.5.1 - Alocação de água

O primeiro passo no processo de programação dinâmica é estruturar o problema de alocação sequencial ou um processo de tomada de decisão com multiestágios. A alocação para cada usuário é considerada a decisão de estágio numa sequência de decisões. Quando uma porção x_j de um total de água Q é alocado num estágio j, isto resultará num benefício líquido $R_j(x_j)$ (chamada de função de retorno). O estado variável s_j é definido como a quantidade de água disponível para os usuários ou estágios remanescentes. Finalmente, a função de transformação de estado $s_{j+1}=s_j-x_j$ define o estado no próximo estágio como função do estágio corrente, da alocação e da decisão corrente (Loucks et al., 1981).

A função f(Q) é o máximo de benefícios líquidos que podem ser obtidos pela alocação de uma quantidade de água Q aos usuários j.

Tendo em mãos esses dados foi feito inicialmente um algoritmo com 3 variáveis de decisão (3 usuários) onde cada alocação x não podia ser negativa e sua soma não podia exceder Q. O problema de alocação foi definido pela equação:

$$f(Q) = \max[R_1(x_1) + R_2(x_2) + R_3(x_3)], \qquad (1)$$

$$x_1+x_2+x_2 \leq q \ ; \ x(1), x(2), x(3) \geq 0 \qquad (2)$$

Utilizamos relações de recorrência, ou seja, equações que devem ser resolvidas em seqüência. As relações recursivas envolvem sucessivas soluções para uma série de equações, onde cada uma depende dos valores obtidos nas equações anteriores.

Uma série de equações recursivas são fundamentais para a programação dinâmica. É muitas vezes mais fácil e rápido de resolver numerosos problemas com uma única variável do que um único problema com muitas variáveis. Cada relação recursiva representa um estágio no qual a decisão é requerida.

Quando as variáveis de estado ou a quantidade de água disponível s_j num estágio j e as variáveis de decisão ou alocação x_j do usuário j estão dispostos em uma sequencia finita de valores discretos, o problema é chamado de problema discreto de programação dinâmica. A solução será sempre o máximo (ou o mínimo) global que independe da concavidade, complexidade ou mesmo continuidade das funções.

A resolução de problemas discretos de programação dinâmica, a fim de encontrar o valor da função objetivo e também os valores das variáveis de decisão que maximize ou minimize a função objetivo, pode ser melhor entendido com o uso de tabelas, uma para cada estágio do processo de tomada de decisão.

O método de solução de programação dinâmica em que o último estágio é resolvido e as relações funcionais (ou de recorrência) são utilizadas para determinação da solução do fim para o início do processo é chamado de programação dinâmica regressiva. Em muitas situações, a determinação da política ótima é obtida progressivamente ao invés de regressivamente. Um exemplo do uso da programação dinâmica progressiva é a operação de reservatórios em tempo real, onde o estado do sistema é conhecido no tempo atual.

Existem várias pesquisas sobre a aplicação de programação dinâmica em operações de reservatórios, por exemplo veja Harboe (1983). No nosso trabalho usamos inicialmente o método de Loucks, et al. (1981), baseado no qual foi feito o 1° algoritmo de programação dinâmica.

A fim de mostrar como resolver um problema de alocação de água usando programação dinâmica resolveremos um exemplo com 3 usuários (Loucks et al., 1981).

Assumiremos que o total de benefícios é definido pela função do tipo $a_j[1-\exp(b_j x_j)]$ para cada uso j e o custo é definido pela função concava $c_j x_j^{d_j}$, onde c_j e d_j são constantes positivas conhecidas e $d_j < 1$. Esta função de custos tem os custos médios decrescentes com o aumento da quantidade alocada x_j.

Assumindo que o objetivo reside em maximizar o total de benefícios líquidos (benefícios totais - custos), o novo modelo de planejamento é

$$\text{maximizar } \Sigma\{a_j[1-\exp(-b_j x_j)]-c_j x_j^{d_j}\}\text{ , com j variando de 1 a 3} \qquad (3)$$

sujeito a

$$\Sigma x_j \leq Q\text{ , com j variando de 1 a 3} \qquad (4)$$

e a condição não negativa

$x_j \geq 0$ para cada uso j (5)

O primeiro passo no procedimento da programação dinâmica é a estruturação do problema de alocação como um processo seqüencial de alocação ou um procedimento de tomada de decisão de multiestágios, como está ilustrado na figura 5. A alocação para cada usuário é considerada um estágio da decisão numa seqüência de decisões. Quando uma porção x_j do suprimento total de água Q é alocado num estágio j, isto resulta num benefício líquido que expresso pela equação $R_j(x_j) = a_j[1\text{-exp}(-b_jx_j)]-c_jx_j^{d_j}$. Uma variável de estado s_j é definida como a quantidade de água disponível para os (4-j) usuários restantes (ou estados restantes). Finalmente, uma função de transfomação de estado $s_{j+1} = s_j - x_j$ define o estado no próximo estágio como função do estado presente e da alocação ou decisão presente.

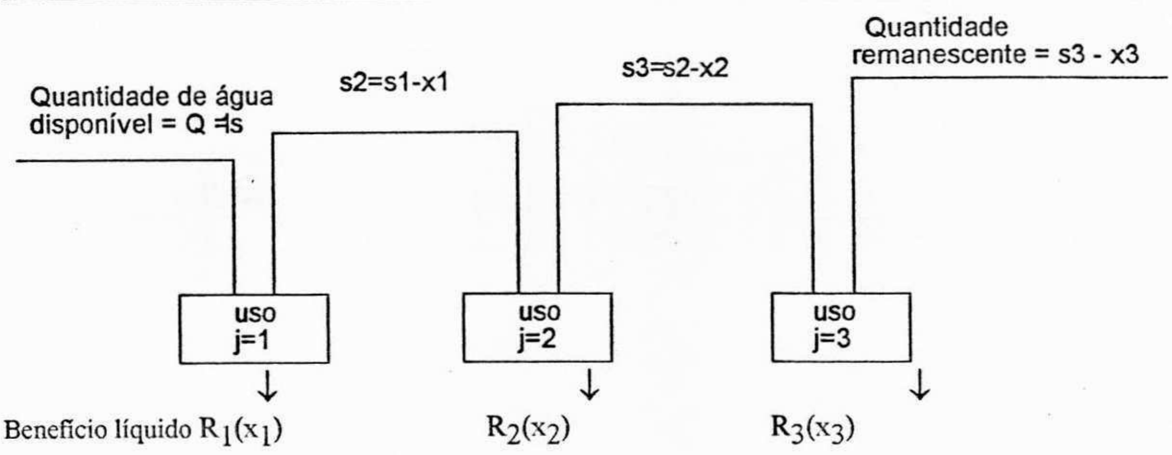

Figura 5. Procedimento da alocação sequêncial.

Com essas definições, é possível escrever o problema de alocação definido pelas equações 3 a 5 como:

$$f_1(Q) = \text{máximo } [R_1(x_1) + R_2(x_2) + R_3(x_3)] \quad \text{-função objetivo-} \quad (6)$$

$$x_1+x_2+x_3 \leq Q$$

$$x_1,x_2,x_3 \geq 0 \quad \text{-restrições-}$$

A função $f_1(Q)$ é o máximo de benefícios líquidos que pode ser obtido com a alocação de uma quantidade de água Q aos usuários 1, 2 e 3. Cada alocação x_1, x_2 e x_3 não pode ser negativa e sua soma não pode exceder Q. A equação 6 apresenta o problema com 3 variáveis de decisão, que pode ser transformado em três problemas, cada um tendo uma variável de decisão.

A equação 6 pode ser escrita da seguinte forma:

$$f_1(Q) = \underset{0 \leq x_1 \leq Q}{\text{máximo}} \{R_1(x_1) + \underset{0 \leq x_2 \leq Q-x_1 = s_2}{\text{máximo}} [\, R_2(x_2) + \underset{0 \leq x_3 \leq s_2-x_2 = s_3}{\text{máximo}} R_3(x_3)]\} \tag{7}$$

Seja a função $f_3(s_3)$ o máximo de benefícios líquidos derivado do uso 3 dada a quantidade s_3 disponível para a alocação àquele uso. Portanto para vários valores discretos de s_3 entre 0 e Q, pode-se determinar o valor de $f_3(s_3)$, onde:

$$f_3(s_3) = \underset{0 \leq x_3 \leq s_3}{\text{máximo}} [R_3(x_3)] \tag{8}$$

Já que $s_3 = s_2 - x_2$, a equação 7 pode agora ser escrita levando em conta apenas x_1, x_2 e s_2:

$$f_1(Q) = \underset{0 \leq x_1 \leq Q}{\text{máximo}} \{R_1(x_1) + \underset{0 \leq x_2 \leq s_2}{\text{máximo}} [\, R_2(x_2) + f_3(s_2 - x_2)]\} \tag{9}$$

Fazemos agora a função $f_2(s_2)$ igual ao máximo de benefícios líquidos derivados dos usos 2 e 3 dada a quantidade s_2 para ser alocada a aqueles usos. Assim para vários valores dicretos de s_2 entre 0 e Q, um pode determinar o valor de $f_2(s_2)$ onde

$$f_2(s_2) = \underset{0 \leq x_2 \leq s_2}{\text{máximo}} [\, R_2(x_2) + f_3(s_2 - x_2)]\} \tag{10}$$

Finalmente, já que $s_2 = Q - x_1$, a equação 6 pode ser escrita em termos apenas de x_1 e Q:

$$f_1(Q) = \underset{0 \leq x_1 \leq Q}{\text{máximo}} \, [R_1(x_1) + f_2(Q-x_1)]\} \quad (11)$$

Aqui $f_1(Q)$ e o máximo de benefícios líquidos alcançados com a quantidade de água Q alocada aos usos 1, 2 e 3. A equação 11 não pode ser resolvida sem conhecermos $f_2(s_2)$. A equação 10 que nos dá $f_2(s_2)$ não pode ser resolvida sem o conhecimento de $f_3(s_3)$. Felizmente, $f_3(s_3)$ pode ser obtido usando a equação 8 sem referência a nunhuma outra função de benefício líquido $f_j(s_j)$. Uma vez determinado o valor de $f_3(s_3)$, o valor de $f_2(s_2)$ pode ser computado, como o valor que nos permitirá determinar $f_1(Q)$, a quantidade que se desejava encontrar.

Como as equações 8, 10 e 11 devem ser resolvidas em seqüência, elas são conhecidas como equações recursivas.

Quando as variáveis de estado ou quantidade de água disponível s_j num estágio j e as variáveis de decisão ou alocações x_j ao uso j podem ser determinadas num conjunto finito de valores discretos, o problema é de programação discreta. A solução será sempre um máximo ou mínimo global independentemente da concavidade, convexidade ou mesmo continuidade das funções $R_j(x_j)$.

Resolver problemas de programação dinâmica para achar o valor de uma função objetivo, e ainda os valores das variáveis de decisão que maximize ou minimize a função objetivo pode ser melhor resolvido com o uso de tabelas, uma para cada estágio do processo de tomada de decisão.

A tabela 2 nos dá os valores da função dos benefícios líquidos (eq. 3) para nosso exemplo.

Os números na tabela 2 estão baseados nos seguintes dados: Q = 5; a_j = 100, 50, 100; b_j =0,1 , 0,4 , 0,2; c_j = 10, 10, 25; e d_j = 0,6 , 0,8 , 0,4 para j = 1, 2, 3, respectivamente. Na primeira coluna temos os valores que x_j pode assumir. Na primeira linha e segunda coluna temos o valor da função dos benefícios líquidos $R_1(0)$ quando não alocamos nenhuma água para o usuário 1; na segunda linha e segunda coluna temos o valor da função de benefíos líquidos $R_1(1)$ alocando uma quantidade de água igual a 1 para o usuário 1; na primeira linha e terceiras coluna temos o valor da função $R_2(0)$ alocando nenhuma quantidade de água para o usuário 2 e assim por diante.

Tabela 2. Valores da função benefícios líquidos.

x_j	$R_1(x_1)$	$R_2(x_2)$	$R_3(x_3)$
0	0	0	0
1	-0,5	6,5	-6,9
2	3,0	10,1	0
3	6,6	10,9	6,3
4	10,0	9,6	11,5
5	13,1	7,0	15,6

Os valores associados com o primeiro estágio, definidos pela equação 8, estão apresentados na tabela 3. Na primeira coluna estão os valores que a quantidade de água disponível s_j pode assumir e na primeira linha estão os valores que a variavel de decisão x_j pode assumir. Como estamos trabalhando com programação dinâmica regressiva os cálculos se iniciam a partir do último usuário, neste caso usuário 3.

Na primeira linha o valor da quantidade de água disponível (s_3) é 0, então a quantidade de água que poderá ser alocada x_3 só pode ser zero. Para $x_3 = 0$ o valor de $R_3(0) = 0$ (Tabela 2), e este é o valor encontrado na segunda coluna da segunda linha. Para $s_3=1$, poderemos alocar 0 ou 1 e da tabela 2 tiramos os valores de R_3 correspondentes (que são 0 e -6,9 respectivamente), e assim por diante foram calculados todos os demais valores da tabela.

Os valores para os estágios intermediários, definidos pela equação recursiva 10, tendo a forma geral

$$f_j(s_j) = \underset{0 \leq x_2 \leq s_2}{\text{máximo}} [\, R_j(x_j) + f_{j+1}(s_j - x_j)]\} \qquad (12)$$

para vários valores de s_j de 0 a Q, podem ser encontrados na tabela 4.

Tabela 3. Primeiro estágio de resolução do problema.

estado					$R_3(x_3)$				
s_3	x_3:	0	1	2	3	4	5	$f_3(s_3)$	x_3*
0		0						0	0
1		0	-6,9					0	0
2		0	-6,9	0				0	0 e 2
3		0	-6,9	0	6,3			6,3	3
4		0	-6,9	0	6,3	11,5		11,5	4
5		0	-6,9	0	6,3	11,5	15,6	15,6	5

Tabela 4. Segundo estágio de resolução do problema.

estado					$R_2(x_2)$	+	$f_2(s_2$	$- x_2)$	
s_3	x_3:	0	1	2	3	4	5	$f_3(s_3)$	x_3*
0		0						0	0
1		0	6,5					6,5	1
2		0	6,5	10,1				10,1	2
3		6,3	6,5	10,1	10,9			10,9	3
4		11,5	12,8	10,1	10,9	9,6		12,8	1
5		15,6	18,0	16,4	10,9	9,6	7,0	18,0	1

A tabela 5 contem os valores calculados da equação recursiva final 11.

Tabela 5. Resultados da equação recursiva final.

estado				$R_2(x_2)$	+	$f_2(s_2$	$- x_2)$		
s_3	x_3:	0	1	2	3	4	5	$f_3(s_3)$	x_3*
5		18,0	12,3	13,9	16,7	16,5	13,1	18,0	0

Na penúltima coluna de cada tabela temos o valor máximo da função dos benefícios líquidos associados a cada valor da variável de estado, que é calculada comparando os valores da função e escolhendo o maior deles em cada estado.

Guardando o valor da alocação ótima x_j* associado a cada valor da variável de estado torna possível a volta por cada tabela sucessiva para achar os valores ótimos de cada variável de decisão. Da tabela 5 o máximo benefício líquido $f_1(Q)$ é igual a 18, e a alocação x_1* que resulta esse máximo é 0. Portanto o ótimo s_2, a quantidade disponível para ser alocada aos usos 2 e 3, é $Q - x_1 = 5 - 0 = 5$. Da tabela 4 a ótima alocação x_2*, ao uso 2 é dado por $s_2 = 5$, é 1. Por isso o s_3 ótimo, a quantidade disponível para a alocação ao uso 3, é $s_2 - x_2 = 5 - 1 = 4$. Da tabela 3 a alocação ótima x_3* ao usuário 3 é 4. Então as alocações ótimas são 0, 1 e 4 aos usos 1, 2 e 3, respectivamente. A soma de alocações são, neste caso igual a Q. Toda quantidade de água é alocada.

Baseado neste exemplo foi feito um programa em linguagem C, que simula a operação de um reservatório, alocando as liberações para um máximo de 3 usuários, segundo um esquema de programação dinâmica regressiva. O programa está no anexo 2 com o nome de pdn1.c.

Mais tarde então aprimorou-se o programa através da simulação em uma sequência de meses, para um limite de 10 usuários. O programa se encontra no anexo 2 com o nome de pdn2.c.

4.5.2 Operação de reservatórios

Segundo Loucks as três aplicações comuns de programação dinâmica em planejamento de recursos hídricos são: alocação de água, capacidade de expansão e operação de reservatórios. Utilizamos, inicialmente, a programação dinâmica para alocação de água com o objetivo de nos dar uma base para sua utilização na operação de reservatórios. Primeiramente fizemos um programa baseado no exemplo de Loucks et al. (1981), paginas 40 a 44, cujo procedimento detalhado se encontra a seguir.

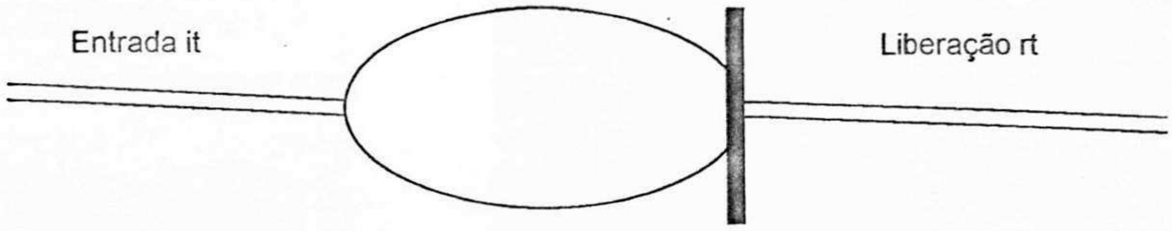

Figura 6. Variáveis para operação de reservatórios para cada período t - entrada (inflows i), liberações (releases r) e armazenamento (storage s).

A figura 6 ilustra um único reservatório tendo inflows (vazões de entrada) i_t e liberações r_t em cada período t. Em problemas determinísticos, assim como esse, a seqüência de inflows i_t é assumida como conhecida e a seqüência de liberações r_t deve ser determinadas. Dado a capacidade armazenada (storage) do reservatório K, o problema de operação de reservatório envolve encontrar a seqüência de liberações r_t que maximize o total de benefícios líquidos. Esses benefícios líquidos devem ser uma função do volume armazenado assim como da liberação (r_t).

Sendo s_t o volume armazenado inicial no periodo t. Assumamos que os benefícios líquidos em cada período t podem ser definidos como funções do volume de armazenamento inicial e final (s_t e s_{t+1}) e liberação (r_t) e que pode ser escrito como NB_t (s_t, s_{t+1}, r_t). Temos que assumir também que essas funções de benefícios líquidos para cada período serão a mesma de um ano para o outro.

Os benefícios associados com armazenamento podem ser originados através de lagos de recreação, hidroelétricas, pscicultura e a proteção de várias espécies da vida animal e seus habitantes. Os benefícios das liberações podem resultar da navegação, abastecimento de água, irrigação e diluição de poluentes.

Seja i_t as entradas em um período t. Assumiremos que haverão T períodos em um ano. O objetivo de um gerenciador deve ser maximizar o benefício anual total.

$$\text{maximizar } \sum NB_t (s_t, s_{t+1}, r_t), \quad \text{com t variando de 1 a T.} \qquad (13)$$

As restrições incluem um balanço da massa de entradas e saídas em cada período t. Há várias maneiras de expressar este balanço de massa (balanço hídrico). Assumindo evaporação e as perdas através de infiltração ou vazamento como insignificantes, um caminho é igualar o volume armazenado final s_{t+1}, no período t (que é o mesmo que o volume armazenado inicialmente no período t+1), ao volume armazenado inicial s_t mais a vazão de entrada i_t, menos a liberação r_t.

$s_{t+1} = s_t + i_t - r_t$, para cada período t (14)

As restrições devem ainda incluir a restrição da capacidade K em cada volume armazenado s_t:

$s_t \leq K$, para cada período t. (15)

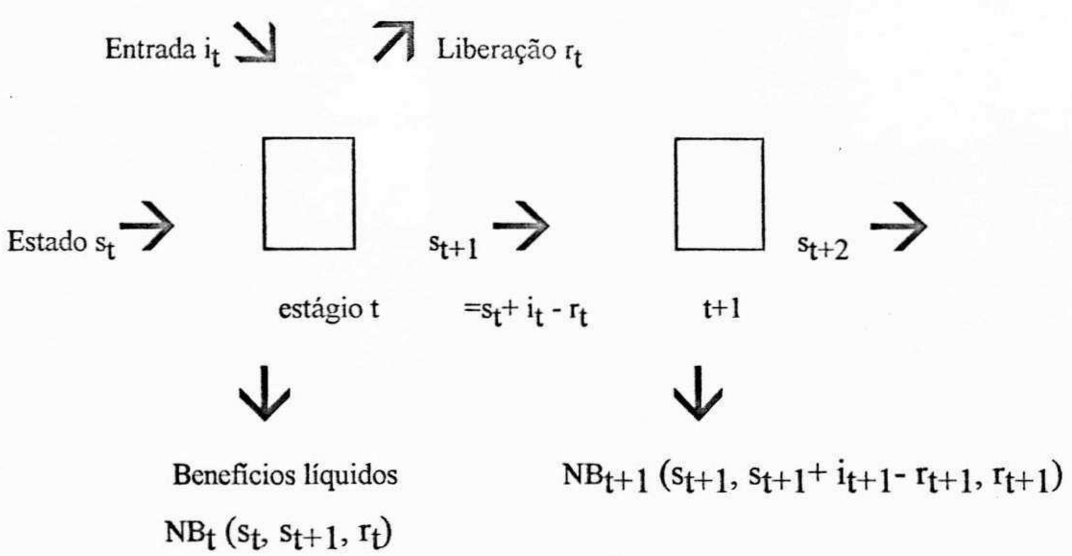

Figura 7. Processo seqüêncial de operação de reservatório

Neste exemplo, a evapotranspiração, a infiltração, o volume de controle de cheias e o volume morto são ignorados.

O problema de operação de reservatório definido pelas equações 13 a 15, pode ser visualizado como um processo de tomada de decisão em múltiplos estágios como está ilustrada na figura 7.

Os estágios são os períodos de tempo, e os estados são os volumes armazenados. Novamente, uma seqüência de equações recursivas tanto progressivas como regressivas podem ser formuladas, uma para cada estágio do processo. Procedendo regressivamente, um determinado período é selecionado depois do qual se assume que o reservatório não será mais operado. Este pode ser qualquer período em qualquer ano, porque o eventual "estado-uniforme" (política ótima) derivado do modelo será independente dessa suposição arbitrariamente fornecida que ambas as entradas médias i_t e a função de benefícios líquidos NB_t, em cada período t, não mudam de um ano para o próximo.

Deixemos o período final arbitrário ser o período T. Fica apenas um período de operação, que é o período mais distante do lado direito, mostrado na figura 8.

Depois defina-se uma função $f_T^1(s_T)$ que é o máximo de benefícios líquidos derivados da operação do reservatório no último período daquele último ano, dado um volume armazenado inicial de s_T,

$$f_T^1(s_T) = \underset{\substack{r \geq 0 \\ r_T \leq s_T + i_T \\ r_T \geq s_T + i_T - K}}{\text{maximo}} [NB_T(s_T,\ s_T + i_T - r_T,\ r_T)] \qquad (16)$$

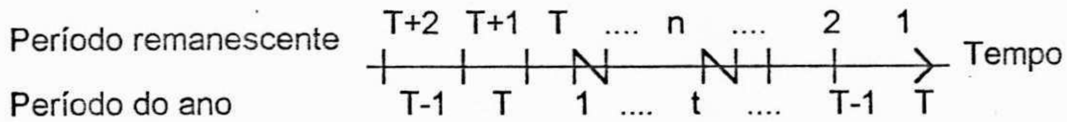

Figura 8. Relação entre os períodos T e n em cada estágio do problema de operação de reservatório.

As restrições nas liberações r_t são limitadas pela quantidade de água disponível, e força um sangramento se a água disponível exceder a capacidade do reservatório K. A equação 16 deve ser resolvida para valores discretos de s_T a 0, ou um mínimo permissível de volume armazenado naquele período ao máximo do volume armazenado possível K. Esses valores de $f_T^1(s_T)$ serão necessários para resolver as próximas equações recursivas.

Movendo-se regressivamente no tempo o próximo estágio é o período anterior, T-1. Há agora dois períodos remanescentes para operação do reservatório. Neste caso a função $f_{T-1}^2(s_{T-1})$ representa o benefício líquido total máximo dos dois períodos que virão, dado o armazenamento inicial de s_{T-1} no período T-1. Desde que $s_T = s_{T-1} + i_{r-1} - r_{T-1}$, $f_T^1(s_T)$ pode ser expresso em termos da variável de decisão s_{T-1}, da variável de decisão r_{T-1} e a média de inflow i_{T-1} conhecido.

$$f_{T-1}^2(s_{T-1}) = \underset{\substack{r_{T-1} \geq 0 \\ r_{T-1} \leq s_{T-1} + i_{T-1} \\ r_{T-1} \geq s_{T-1} + i_{T-1} - K}}{\text{maximo}} [NB_{T-1}(s_{T-1},\ s_{T-1} + i_{T-1}\ r_{T-1},\ r_{T-1})] \qquad (17)$$

O que deve ser novamente resolvido para todos os valores discretos de s_{T-1} entre 0 e K.

Voltando continuamente no tempo, a equação recursiva geral para cada período t com n (n>1) períodos remanescentes pode ser escrito:

$$f_t^n(s_t) = \underset{\substack{r_t \geq 0 \\ r_t \leq s_t + i_t \\ r_t \geq s_t + i_t - K}}{\text{maximo}} [NB_t(s_t, s_t + i_t - r_t, r_t)] + f_{t-1}^{n-1}(s_t + i_t - r_t) \qquad (18)$$

onde o índice n começa em 2 e aumenta em cada estágio sucessivo e o índice t varia regressivamente de um período T a 1 e depois ao período T novamente (figura 8). No caso do nosso trabalho nós não utilizamos o índice n mas apenas o índice t que varia de T a 1, ou seja do último período ao primeiro, portanto a função objetivo que utilizamos é igual a equação 18 a menos do índice n.

Loucks apresenta um exempo simples que ajuda a ilustrar aquele modelo de operação de reservatórios e suas soluções. Consideremos um reservatório no qual se deseja manter constante o volume armazenado de 20 e uma saída constante de 25. Assumamos que a capacidade K do reservatório é 30 e que as entradas i_t são 10, 50 e 20, em três estações distintas (t=1,2,3), respectivamente. Desejamos uma política de operação que minimize a soma anual da diferença ao quadrado destes valores de volumes armazenados e saídas. Desta forma $NB_t(s_t, s_{t+1}, r_t)$ na equação 13 será igual a $[(20-s_t)^2 + (25-r_t)^2]$. NB deverá igualar a subtração da quantidade entre parenteses para que possamos minimizar as perdas.

Deixando s_t assumir os valores discretos 0, 10, 20 e 30 e r_t os valores discretos de 10, 20, 30 e 40, a tabela 6 resume a solução das equações 16 a 18, usando tabelas expandidas (ver tabelas 7 a 9) que são similares as mostradas nas tabelas 3 a 5.

Neste exemplo a liberação ótima do reservatório r_t derivados após o 1º estágio com n = 1 períodos remanescentes define a política estacionária seqüencial de liberações. Não apenas as liberações são as mesmas mas também a diferença $f_t^{n+3}(s_t) - f_t^n(s_t)$ é constante, 275, para todos s_t e t para n>1. Este valor é a soma anual mínima do desvio padrão que pode ser obtido seguindo a política de operação seqüencial derivada.

Através da simulação da operação de reservatórios com esta política a seqüência determinística de entrada, a seqüência ótima de volume armazenado e liberações correspondentes podem ser determinadas. Com a política de operação estacionária e as entradas determinadas e cíclicas a mesma seqüência de volumes armazenados e liberações ocorrerão ano após ano, uma vez que qualquer armazenamento estacionário começa a assumir o volume de armazenameto inicial.

Tabela 6. Solução resumida do problema exemplo de operação de reservatório

	n = 1		n = 2		n = 3	
s_t	$f_3^1(s_3)$	r_3^*	$f_2^2(s_2)$	r_2^*	$f_1^3(s_1)$	r_1^*
0	425	20, 30	450	30	1075	10
10	125	20, 30	250	30	575	10, 20
20	25	20, 30	350	40	275	20
30	125	20, 30	—	☼	375	30

	n = 4		n = 5		n = 6	
s_t	$f_3^4(s_3)$	r_3^*	$f_2^5(s_2)$	r_2^*	$f_1^3(s_1)$	r_1^*
0	1200	10	725	30	1350	10
10	600	10	525	30	850	10, 20
20	300	20	625	40	550	20
30	400	30	—	☼	650	30

	n = 7	
s_t	$f_3^4(s_3)$	r_3^*
0	1475	10
10	875	10
20	575	20
30	675	30

	Política estacionária Liberações ótimas		
s_t	r_1^*	r_2^*	r_3^*
0	10	30	10
10	10, 20	30	10
20	20	40	20
30	30	—	30

☼ Requer $r_t \geq 50$

Como demonstração fizemos as tabelas 7, 8 e 9 , que são similares às tabelas de alocação, detalhando como foram obtidos os valores de f_t^n e r_t para o primeiro ano, que foi dividido em três períodos, cada um correspondente a 4 meses do ano. Os demais valores da tabela 6 foram obtidos de maneira análoga.

Tabela 7. Calculo de $f_3^1(s_3)$ e dos r_3^* correspondentes.

	$NB = [(20-s_t)^2 + (25-r_t)^2]$					
s_3	r_3 : 10	20	30	40	$f_3^1(s_3)$	r_3^*
0	625	425	425	625	425	20, 30
10	325	125	125	325	125	20, 30
20	225	25	25	225	25	20, 30
30	25	125	125	325	125	20, 30

Tabela 8. Calculo de $f_2^2(s_2)$ e dos r_2^* correspondentes.

	$NB = [(20-s_t)^2 + (25-r_t)^2 + f_3^1(s_2 + i_2 - r_2)]$					
s_3	r_3 : 10	20	30	40	$f_2^2(s_2)$	r_2^*
0	—	550	450	750	450	30
10	—	—	250	350	250	30
20	—	—	—	350	350	40
30	—	—	—	—	—	☼

☼ Requer $r_t \geq 50$.

Tabela 9. Calculo de $f_1^3(s_1)$ e dos r_1^* correspondentes.

	$NB = [(20-s_t)^2 + (25-r_t)^2 + f_2^2(s_1 + i_1 - r_1)]$					
s_3	r_3 : 10	20	30	40	$f_1^3(s_1)$	r_1^*
0	1075	—	—	—	1075	10
10	575	575	—	—	575	10, 20
20	575	275	475	—	275	20
30	—	475	375	1075	375	30

Baseado neste exemplo foi feito um programa em linguagem C, que opera um reservatório, segundo um esquema de programação dinâmica regressiva. O programa está no anexo 2 com o nome de reserv-1.c.

Mais tarde então aprimorou-se e adaptou-se o programa às nossas necessidades: eliminamos o índice n, já que não era nosso intuito chegarmos a uma política ótima constante, pois nossos dados de entrada e armazenamento variam todos os meses; incluiu-se na equação do balanço hídrico a evaporação, que por sua vez é calculada através dos dados da curva cota x volume; modificamos ainda a equação da função objetivo e implantamos no programa a leitura de arquivos com os dados necessários para o programa. Para cada uma dessas modificações foi feito um novo programa mas vamos apresentar apenas a versão final com todas essas modificações. O programa se encontra no anexo 2 com o nome de reservat.c. Os arquivos de dados para o programa se encontram no anexo 1.

4.6 Ferramentas computacionais

Os programas deste trabalho foram escritos em linguagem C. C é uma linguagem de programação de finalidade geral que permite economia de expressão, modernos fluxos de controle e estruturas de dados e um rico conjunto de operadores. Ela é estritamente associada ao sistema operacional UNIX já que foi desenvolvida neste sistema. A linguagem, entretanto, não é atada a um sistema operacional ou a uma máquina particulares, e, embora tenha sido chamada "linguagem de programação de software básico" devido à sua utilidade no desenvolvimento de sistemas operacionais, ela tem sido usada, igualmente, para escrever grandes programas numéricos, de processamento de texto, e bancos de dados.

Antes de escrevermos os programas passamos por uma fase de aprendizado da linguagem C, utilizando Shildt (1988), Kernighan e Ritchie (1986) e Arakaki e Arakaki (1990) entre outros.

5 - Resultados e discussão

O trabalho objetivou estabelecer a regra de operação para 1 ano, de janeiro a dezembro, de um reservatório utilizando programação dinâmica. A regra de operação consiste na liberação que deve ser feita de acordo com o estado do reservatório de modo que provoque a mínima escassez no período de operação. A fim de verificar a validação do modelo proposto utilizou-se os dados do açude Engenheiro Arcoverde, os quais se encontram no anexo 1 em forma de arquivos.

A regra de operação pode ser representada através de uma tabela (ver tabela 10) que dá a liberação para cada mês de acordo com o volume armazenado no início do mês. No nosso programa o volume variou de um mínimo correspondente ao volume morto do reservatório (4.000.000 m^3) a um máximo, que correspondeu a máxima capacidade do reservatório (36.000.000 m^3).

As funções objetivo testadas em nosso programa foram:

1) **dr - da**

2) **(dr - da)^2**

3) **(dr -da)/dr**

Onde :

dr - demanda requerida e
da - demanda atendida

Todas as três apresentaram a mesma performance. A demanda requerida adotada no estudo foi de 500.000 m^3/mês constante para todos os meses. As demandas foram dicretizadas de 50.000 em 50.000 m^3 e os volumes armazenados de 1.000.000 em 1.000.000 m^3. A série de vazões utilizada foi do ano de 1963. Os resultados estão apresentados na tabela 10, onde na primeira linha temos os meses do ano, na primeira coluna temos a variação do volume de água armazenado no reservatório e nos demais quadros temos as liberações ótimas para cada mês em mil m^3.

A tabela revela que quando o volume requerido é maior do que a capacidade que o reservatório tem de atendê-lo a estratégia mais adequada de operação é fazer um racionamento de água. Por exemplo no mês de setembro quando o reservatório estava com um volume armazenado de 5.000.000 m^3 de água, ao invés de atender os 500.000 m³ de água requeridos atendeu-se 400.000 m³. Isto vai garantir um atendimento mínimo de água nos meses subsequêntes.

Apesar de se ter usado os dados de 1963, numa operação concreta, devem ser usados os dados de vazão afluentes previstas para aquele ano, onde este trabalho de estabelecimento das liberações pode ser atualizado mês a mês a medida que se diponha de novos valores de previsão da vazão afluente.

A dificuldade encontrada no programa foi a utilização de séries longas de vazões afluentes e de menores discretizações das demandas e volumes armazenados. Este problema é conhecido na literatura como "Curse of dimensionality" e que, no nosso caso, foi agravado pela utilização de um micro computador 286 com memória de 640 k bytes e de um compilador Turbo C versão 2.0. A utilização de micro computadores e compiladores de última geração minimizarão o problema.

Tabela 10 - Regras de operação obtidas com o programa reservat.c, para o reservatório Engenhreiro Arcoverde.

mês Vol	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
4	500	500	500	500	500	500	500	500	500	**250**	500	500
5	500	**100**	**100**	**150**	**150**	**200**	**250**	**350**	**400**	500	500	500
6	500	500	500	500	500	500	**250**	**300**	**400**	500	**450**	500
7	500	500	500	500	500	500	500	**300**	**350**	500	**400**	500
8	500	500	500	500	500	**150**	200	**250**	**350**	500	**400**	**200**
9	500	500	500	500	500	500	500	**250**	**300**	500	**350**	500
10	500	500	500	500	500	500	**150**	**250**	**300**	500	**350**	**150**
11	500	500	500	500	500	500	500	**200**	**300**	500	**300**	500
12	500	500	500	500	500	500	500	500	**300**	500	**300**	500
13	500	500	500	500	500	500	500	500	500	500	500	500
14	500	500	500	500	500	500	500	500	500	500	500	500
15	500	500	500	500	500	500	500	500	500	500	500	500
16	500	500	500	500	500	500	500	500	500	500	500	500
17	500	500	500	500	500	500	500	500	500	500	500	500
18	500	500	500	500	500	500	500	500	500	500	500	500
19	500	500	500	500	500	500	500	500	500	500	500	500
20	500	500	500	500	500	500	500	500	500	500	500	500
21	500	500	500	500	500	500	500	500	500	500	500	500
22	500	500	500	500	500	500	500	500	500	500	500	500
23	500	500	500	500	500	500	500	500	500	500	500	500
24	500	500	500	500	500	500	500	500	500	500	500	500
25	500	500	500	500	500	500	500	500	500	500	500	500
26	500	500	500	500	500	500	500	500	500	500	500	500
27	500	500	500	500	500	500	500	500	500	500	500	500
28	500	500	500	500	500	500	500	500	500	500	500	500
29	500	500	500	500	500	500	500	500	500	500	500	500
30	500	500	500	500	500	500	500	500	500	500	500	500
31	500	500	500	500	500	500	500	500	500	500	500	500
32	500	500	500	500	500	500	500	500	500	500	500	500
33	500	500	500	500	500	500	500	500	500	500	500	500
34	500	500	500	500	500	500	500	500	500	500	500	500
35	500	500	500	500	500	500	500	500	500	500	500	500
36	500	500	500	500	500	500	500	500	500	500	500	500

6 - Conclusões

Neste trabalho testou-se a aplicação de algoritmos de otimização utilizando Programação Dinâmica para estabelecimento de regras de operação de reservatórios de abastecimento d'água.

Foram desenvolvidos programas para aplicação do algoritmo num caso real. Os resultados mostraram a eficiência do método na derivação das regras de operação para o reservatório Engenheiro Arcoverde, no semi-árido da Paraíba.

Como resultado deste trabalho apresentou-se resumo à 46ª Reunião Anual da SBPC, realizada em Vitória-ES, em julho de 1994 (Anexo 3) e submeteu-se o trabalho ao II Simpósio de Recursos Hídricos do Nordeste, a se realizar em Fortaleza-CE, em novembro de 1994 (Anexo 3).

7- Bibliografia

ATECEL (1993) Plano estadual e gerenciamento de recursos hídricos disponibilidades hídricas do Estado da Paraíba: Bacia do rio Piranhas. Campina Grande ATECEL.

Arakaki, J., Arakaki, R., (1990) Turbo C2.0. Rio de Janeiro RJ, Livros Técnicos e Científicos Editora.

Born, P. H. e Stendiger, J. (1989) Modelos hidrológicos e sua representação em programação dinâmica estocástica para otimização de operação de reservatórios. Anais do VIII Simpósio Brasileiro de Recursos Hídricos volume 1, Foz do Iguaçu, páginas 32 - 43.

Braga, B. P. F., (1987) Técnicas de otimização e simulação aplicadas em sistemas de recursos hídricos. In: Coleção ABRH de Recursos Hídricos Modelos para Gerenciamento de Recursos Hídricos volume 1, São Paulo, Nobel/ABRH, páginas 427 - 518.

Costa, J. P., Corenstin, B.G., Campodónico, N. M., e Pereira, M. V. F. (1989) Programção estocástica de operação de sistemas hidrotérimicos. Anais do VII Simpósio Brasileiro de Recursos Hídricos, volume 1, Foz do Iguaçu, páginas 152 - 162.

Cunha, L. M. (1985) Derivação de regras de operação de doi subsistemas hidrelétricos interligados. Anais doVI Simpósio Brasileiro de Hidrologia e Recursos Hídricos volume 1, São Paulo, páginas 46 - 56.

Galvão, C. O.,Ribeiro, M. M. R., (1993) Uso da previsão meteorológica no manejo de açudes. Campina Grande, Laboratório de Meteorologia, Recursos Hídricos e Sensoriamento Remoto da Paraíba, LMRS - PB.

Harboe, R. (1983) Optimal operaton of reservoirs by dynamic programming.In: Gussino, E.; Rossi, G., Hendricks, D., (Eds) Operation of Complex Water Systems. Boston, Martinus Nijhoff Publishers, p. 97 - 111. (NATO ASI Series n° 58).

Hernighan, B.W., Ritchie, D. M. (1986) C a linguagem de programação. Rio de Janeiro , Ed. Campus.

Loucks, D. P., Stedinger, J. R., Haith, D. A. (1981) Water resource systems planning and analysis. Englewood Cliffs, New Jersey, Prentice-Hall, Inc.

Molle , F., Cadier, E. e Campello, S. (1985) Modelização do funcionamento de pequenos reservatórios com múltiplos usos. Anais do VI Simpósio Brasileiro de Hidrologia e Recursos Hídricos, volume 2, São Paulo, pág. 343.

Reis, L. F. R. e Chaudhry, F. H. (1991a) Característica estocástica da proposta ótima de um sistema hidrelétrico. Anais do IX Simpósio Brasileiro de Recursos Hídricos, volume 2, Rio de Janeiro, páginas 357 -367.

Reis, L. F. R. e Chaudhry, F. H. (1991b) Estratégia de operação de um sistema hidrelétrico. Anais do IX Simpósio Brasileiro de Recursos Hídricos, volume 2, Rio de Janeiro, páginas 368 -377.

Riberio, M. M. R. (1990) Operação de um sistem de reservatórios para usos de conservação. Dissertação de Mestrado. Universidade Federal da Paraíba - Centro de Ciências e Tecnologia - Campus II - Campina Grande.

Ribeiro, M. M. R. (1993) Levantamento documental sobre as estações fluviométricas no Estado da Paraíba. Campina Grande, Laboratório de Meteorologia, Recursos Hídricos e Sensoriamento Remoto, LMRS -PB.

Shildt, H. (1988). Turbo C - Guia do usuário, São Paulo, McGraw - Hill.

Silva, R. P., Galvão, C. O. (1993) Diagnóstico da rede pluviométrica da Paraíba. Campina Grande, Laboratório de Meteorologia, Recursos Hídricos e Sensoriamento Remoto, LMRS -PB.

Silveira, J. L., Balestieri, J. A. P. e Oliva, V. J. (1991) Otimização de operação de um sistema hidrilétrico utilizando reservatório equivalente. Anais do IX Simpósio Brasileiro de Recursos Hídricos volume 2, Rio de Janeiro, páginas 324 - 337.

Soares, S., Vinhal, C. D. N., Cruz, G. e Ferreira, R. F. (1993), Regras de operação de reservatórios para otimização da operação enérgica de sistemas hidrelétricos. Anais do X Simpósio Brasileiro de Recursos Hídricos volume 4, Gramado RS, páginas 182 - 191.

Zahed, K. F. (1987) Modelos aplicados a operação de sistemas de reservatórios discussão sobre a útilização de modelos gerais. Anais do VII Simpósio Brasileiro de Hidrologia e Recursos Hídricos volume 1, Salvador, págins 631 - 639.

Zahed, K. F. (1984) Algumas metodologias para dimensionamento e operação de reservatórios. Dissertação de mestrado. Escola Politécnica da USP.

8 - Anexo 1

Arquivos do açude Engenheiro Ávidos para o programa reservat.c.

- Arquivo geral (geral1.dat)
- Arquivo de evaporação (evaporac.dat)
- Arquivo dos valores da curva cota x volume (cotavol1.dat)
- Arquivo de vazão (vazão1.dat)

Arquivo geral1.dat

Os dados se encontram na seguinte ordem: 1) numero de anos, 2) volume mínimo armazenado no reservatório, 3) volume máximo armazenado no reservatório (capacidade), 4) incremento do armazenamento (sendo que todos esses dados entraram em milhões de m³), 5) liberação mínima (m³/s), 6) incremento das liberações (m³/s) e 7) liberações máximas para cada mês (m³/s).

1
19
4
36
1
1
1
10 10 10 10 10 10 10 10 10 10 10 10

Arquivo evaporac.dat

Os dados equivalem à evaporação mensal em mm durante um ano.

201 152.3 145.5 129.8 133.5 138.8 163.5 200.3 220.5 237.8 235.5 224.5

Arquivo cotavol1.dat

A primeira linha se refere aos dados da cota em m e na segunda linha encontramos os valores correspondentes do volume em minhares de m³.

82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
000 17.25 69 227 573.75 1145.75 1955.875 3023.5 4344.75 5928.25
7831.5 10053.75 12580.375 15460.5 18765.75 22535.875 26790.5

31544.125 36834.375

Arquivo vazao1.dat

Os dados da vazao são referentes ao período de 1963 a 1983.

.110	.689	1.697	1.555	.021	.022	.000	.000	.002	.026	.024	.170
.104	2.249	4.257	7.971	2.508	.490	.073	.007	.024	.001	.003	.006
.094	.031	2.775	9.433	1.798	.697	.012	.033	.000	.004	.000	.008
.025	.364	.017	.167	.035	.069	.053	.000	.008	.000	.006	.007
.021	.971	2.006	13.490	4.774	.294	.024	.007	.000	.000	.000	.031
.097	.093	2.577	1.779	4.095	.440	.004	.012	.000	.003	.000	.057
.047	.041	2.493	2.579	.670	.030	.064	.000	.000	.000	.000	.002
.232	.062	1.664	.537	.006	.000	.002	.001	.001	.002	.038	.000
.125	.321	2.688	11.001	1.921	.982	.033	.001	.031	.001	.000	.000
.164	.687	2.760	3.666	.862	.065	.030	.059	.000	.000	.000	.061
.039	.053	.163	11.569	3.476	1.121	.043	.022	.007	.026	.011	.051
2.508	7.629	6.421	15.227	3.216	.589	.043	.000	.010	.005	.103	.061
.069	2.281	9.782	9.613	2.950	.567	.091	.000	.000	.000	.000	.014
.041	.159	2.278	1.630	.412	.007	.001	.001	.016	.018	.000	.006
.091	.219	2.683	8.422	1.848	.029	.025	.011	.000	.006	.000	.127
.022	.565	.363	.123	.100	.052	.057	.000	.012	.000	.037	.000
.069	.139	.103	.656	.174	.025	.010	.000	.004	.008	.017	.017
.100	1.431	1.153	.034	.005	.015	.027	.094	.054	.009	.017	.004
.113	.014	4.772	1.044	.000	.003	.000	.000	.000	.000	.010	.013
.041	.054	.102	1.220	.338	.007	.005	.001	.000	.000	.000	.015
.053	.090	.168	.053	.028	.002	.000	.001	.000	.000	.000	.000

9 - Anexo 2

Programas

- Programas de alocação de água pdn1.c e pdn2.c
- Programas de operação de reservatórios reserv-2.c e reservat.c

```
/*
    programa pdn1.c
    31/01/94
    ultima atualizacao em 03/02/94

    por Emilia R. Rabbani e Carlos de O. Galvao

    Laboratorio de Hidraulica, UFPB
    Caixa Postal 505
    58100-970 - Campina Grande,PB
    e-mail: galvao@lmrs.br

    Este e um programa de programacao dinamica regressiva
    utilizando a linguagem c; baseado no exemplo do livro Water
    Resource Systems Planning and Analysis - Loucks et al., paginas
    30,31 e 32, que simula a operacao de um reservatorio, otimizando
    a alocacao de agua para 3 usuarios.

*/


#include<stdio.h> /* inclui as bibliotecas stdio (entrada e saida) e a math
(matematica) neste programa */
#include<math.h>
main()
{
            /* Descricao das variaveis */

 /* n - numero de usuarios */
 /* j - contador de usuarios */
 /* q - vazao total desponivel */
 /* x[j] - valor das variaveis de decisao, neste caso, porcao do total q
         de agua que sera alocado para o usuario j */
 /* r[j][x] - formula que nos da os beneficios de uma alocacao x ao usuario
          j */
 /* a[j], b[j] - constantes definidas, que serao utilizados na formula de
             beneficios */
 /* c[j], d[j] - constantes positivas, onde d[j] < 1 */
 /* c[j]*pot(x[j],d[j]) - define os custos */
 /* s[j] - quantidade de agua disponivel aos usuarios j+1 */
 /* f[j][s] - (funcao objetivo) funcao que permite determinar a alocacao
           x[j] para cada usuario j que maximize o total de beneficios */
```

```
            valor da variavel de estado s[j] */
/* melhorx[j] - nos permite, atraves de uma regressao sucessiva, obter
            o valor otimo para cada variavel de decisao, x[j]*/
/* melhorf[j] - define o maximode beneficios para o valor otimo de cada
            decisao, x[j] */
/* i - e utilizado como contador */
/* w - variavel utilizada para guardar valores temporarios da funcao
        objetivo */

        /* Declaracao das variaveis */

int i,j,n,m,s,q,x;
double xx,w,a[4],b[4],c[4],d[4],f[5][6],opt[5][6],r[5][6],melhorf[4],melhorx[4];

        /* Valores iniciais */

a[1]=100.0;
a[2]=50.0;
a[3]=100.0;
b[1]=0.1;
b[2]=0.4;
b[3]=0.2;
c[1]=10.0;
c[2]=10.0;
c[3]=25.0;
d[1]=0.6;
d[2]=0.8;
d[3]=0.4;
n=3;        /* numero de usuarios */
q=5;        /* vazao total disponivel */

for(i=0;i<=q;i++) /* zera, inicialmente, todos os valores da funcao objetivo */

 {

  for(j=1;j<=q;j++)
   f[i][j]=0;

}

        /* Calculo dos beneficios */
```

```
/* calcula os beneficios para cada usuario de acordo com a funcao objetivo,
   variando a quantidade de agua alocada (de 0 a q) em intervalos de
   uma unidade.*/

for (j=1;j<=n;j++)
 {
 for(x=0;x<=q;x ++)
  {
  xx=x;
  r[j][x]=(a[j]*(1-exp(-b[j]*xx)))-(c[j]*(pow(xx,d[j])));
  printf("Os beneficios para o usuario %d alocando %d e %f\n",j,x,r[j][x]);
  scanf("%d",&m);
  }
 }

     /*   Calculo da alocacao x para cada usuario j
             que maximize o total de beneficios

     Para cada usuario (do ultimo para o primeiro n a 1),
     variando a quantidade de agua disponivel (s) ao usuario j+1
     de 0 a q e levando em conta cada alocacao possivel (de
     0 a s) calcula a funcao de recorrencia w.
      */

for(j=n;j>=1;j--)
 {
 for(s=0;s<=q;s++)
  {
  if(j==1)
  s=q;
  for(x=0;x<=s;x++)
  {
   w=r[j][x]+f[j+1][s-x];
   printf("j=%d, s=%d, x=%d, f[j+1][s-x]=%f, w=%f\n",j,s,x,f[j+1][s-x],w);
   scanf("%d",&m);
   if(w>f[j][s])
    {
     f[j][s]=w;

        /* f[j][s] guarda o valor maximo da funcao objetivo para
             cada estado s (onde s varia de 0 a q)          */
     opt[j][s]=x;
```

```
            /* opt guarda o valor da alocacao que nos da o
               maximo de beneficios              */

        printf("j=%d, s=%d, x=%d, f[j][s]=%f,
opt[j][s]=%f\n",j,s,x,f[j][s],opt[j][s]);
        scanf("%d",&m);
        }
      }
    }
  }
    /* Imprime o beneficio maximo para cada usuario e a alocacao otima
       correspondente.                        */=

  melhorf[1]=f[1][q];
  melhorx[1]=opt[1][q];
  melhorx[2]=opt[2][q-melhorx[1]];
  melhorx[3]=opt[3][q-melhorx[1]-melhorx[2]];
  melhorf[2]=f[2][q-melhorx[1]];
  melhorf[3]=f[3][q-melhorx[1]-melhorx[2]];
  printf("O beneficio total maximo e %f\n",melhorf[1]);
  printf("O beneficio maximo para o usuario 2 e %f\n",melhorf[2]);
  printf("O beneficio maximo para o usuario 3 e %f\n",melhorf[3]);
  printf("A alocacao otima para o usuario 1 e %f\n",melhorx[1]);
  printf("A alocacao otima para o usuario 2 e %f\n",melhorx[2]);
  printf("A alocacao otima para o usuario 3 e %f\n",melhorx[3]);
  scanf("%d",&m);

} /* fim do main() */
```

```
/* programa pdn2.c
  novembro 1993
  ultima atualizacao em 02.12.93

  por Emilia R. Rabbani e Carlos de O. Galvao
  Laboratorio de Hidraulica, UFPB
  Caixa Postal 505
  58100-970 - Campina Grande, PB
  e-mail: galvao@lmrs.br

  Simula a operacao de um reservatorio, alocando as liberacoes entre
  usuarios segundo um esquema de programacao dinamica regressiva
  apresentado por Loucks et al. (1981). Este programa leva em con-
  sideracao os intervalos de tempo de simulacao e calcula a vazao
  de agua disponivel pela equacao do balanco hidrico, utilizando
  dados de evaporacao, vazao afluente e precipitacao no reserva-
  torio .

*/
#include<stdio.h>  /* inclui as bibliotecas de entrada e saida e a matematica */
#include<math.h>
main()
{
            /* Descricao das variaveis */

 /* ni - numero de intervalos de tempo de simulacao (meses)
        valor maximo = 36 */
 /* in - inflow */
 /* e  - evaporacao */
 /* sp - armazenamento presente (minimo necessario) */
 /* sa - armazenamento anterior */
 /* n  - numero de usuarios
        valor maximo = 10 */
 /* j  - contador de usuarios */
 /* q  - vazao total disponivel */
 /* x - variavei de decisao, neste caso, porcao do total q
         de agua que sera alocado para o usuario j */
 /* r[i][j][x] - beneficios de uma alocacao x ao usuario j num intervalo de tempo i
*/
 /* a[j], b[j] - constantes definidas, que serao utilizadas na formula de
               beneficios */
 /* c[j], d[j] - constantes positivas, onde d[j] < 1 */
```

```
/* a[j]*x +b[j] - define os beneficios */
/* s[j] - quantidade de agua disponivel aos usuarios j+1 */
/* f[i][j][s] - alocacao x[j] para cada usuario j que maximize o total de beneficios
*/
/* opt[i][j][s] - guarda o valor da alocacao otima x[j] associada com cada
                valor da variavel de estado s[j] */
/* melhorx[j] - valor otimo para cada variavel de decisao, x[j]*/
/* melhorf[j] - maximo de beneficios para o valor otimo de cada decisao, x[j] */
/* i - contador de intervalos de simulacao */
/* w - variavel que guarda valores temporarios da funcao objetivo */

/* pendencias: conversao float to int
            usar q como float
            */


        /* Declaracao das variaveis */

 int i,j,n,m,u,s,ni,x,q[36],sa[37],sp[37],in[37],e[37];
 double
xx,w,a[11],b[11],c[11],d[11],f[37][11][15],opt[37][11][15],r[37][11][15],melhor,
melhorf[11],melhorx[11];


  /* Valores iniciais */

  a[1]=100.0;
  a[2]=50.0;
  a[3]=100.0;
  b[1]=0.1;
  b[2]=0.4;
  b[3]=0.2;
  c[1]=10.0;
  c[2]=10.0;
  c[3]=25.0;
  d[1]=0.6;
  d[2]=0.8;
  d[3]=0.4;
  n=3;
  ni=1;
  e[1]=3;
  e[2]=3;
```

```
e[3]=4;
in[1]=5;
in[2]=8;
in[3]=0;
sp[0]=3;
sp[1]=2;
sp[2]=6;
sp[3]=2;
        /* Calculo de q - vazao disponivel */

 /* Levando em conta  a equacao do balanco hidrico calculamos a vazao
disponivel
        em cada intervalo de tempo, onde o armazenamento anterior corresponde
        ao armazenamento presente no intervalo de tempo anterior */

for(i=1;i<=ni;i++)
 {
 sa[i]=sp[i-1];
 q[i]=sa[i]+in[i]-e[i]-sp[i];
 printf("q no intervalo de tempo %d e %d\n",i,q[i]);
 scanf("%d",&m);
 }
  /* zera, inicialmente, todos os valores da funcao objetivo */

  for (i=1;i<=ni;i++)
   {
   for (j=1;j<=n+1;j++)
    {
      for (m=0;m<=q[i];m++)
       f[i][j][m]=0;
    }
   }
                    /* Calculo dos beneficios */

  /* calcula os beneficios para cada usuario em cada intervalo de tempo, de
     acordo com a funcao objetivo, variando a quantidade de agua alocada
     (de um minimo que se deseje manter no reservatorio a quantidade de
     agua disponivel para alocacao aos usuarios, neste caso de 0 a q) */

 for (i=1;i<=ni;i++)
  {
  for (j=1;j<=n;j++)
```

```
{
 for (x=0;x<=q[i];x++)
  {
   xx=x;
   r[i][j][x]= (a[j]*(1-exp(-b[j]*x)))-(c[j]*pow(xx,d[j]));
  /* r[i][j][x]=a[i][j]*x-b[j];  funcao linear, modificar a[j] para a[i][j]  */
   printf("Os beneficios no mes %d para o usuario %d alocando %d e
%f\n",i,j,x,r[i][j][x]);
   scanf("%d",&m);
  }
 }
}

printf(" fim do calculo de beneficios\n");

/* Calculo da alocacao x para cada usuario j que maximize o total de beneficios */

 /* para cada usuario ( do ultimo para o primeiro n a 1), variando a quantidade
    de agua disponivel ao usuario j+1 (s) de 0 a q, em cada intervalo de tempo,
    e ainda levando em conta cada alocacao possivel de 0 a s, calcula a funcao
    de recorrencia w.  */

for (i=1;i<=ni;i++)
 {
  for(j=n;j>=1;j--)
   {
    for(s=0;s<=q[i];s++)
     {
      if(j==1)
         s=q[i];
         printf("i= %d, j=%d, s=%d\n",i,j,s);
        for(x=0;x<=s;x++)
         {
          printf("x=%d, s-x=%d, j+1=%d\n",x,s-x,j+1);
          printf("r[i][j][x]=%f,f[i][j+1][s-x]=%f \n",r[i][j][x],f[i][j+1][s-x]);
          w=r[i][j][x]+f[i][j+1][s-x];
          printf("i=%d, j=%d, s=%d, x=%d, f[i][j+1][s-x]=%f,
w=%f\n",i,j,s,x,f[i][j+1][s-x],w);
          scanf("%d",&m);
          if(w>f[i][j][s])
           {
            f[i][j][s]=w;  /* guarda o valor maximo da funcao objetivo, em cada
```

```
                        intervalo de tempo,para cada usuario e para todas as
                        alocacoes possiveis de 0 a q */
        opt[i][j][s]=x; /* guarda o valor da alocacao que nos da o maximo
                        de beneficios */
        printf(" o i=%d, j=%d, s=%d, x=%d, f[i][j][s]=%f,
opt[i][j][s]=%f\n",i,j,s,x,f[i][j][s],opt[i][j][s]);
        scanf("%d",&m);
        }
      }
    }
  }
}
printf("fim do calculo da alocacao otima de x\n");

 /* Imprime o beneficio maximo e a alocacao correspondente de agua para cada
    usuario, em cada intervalo de tempo */

for(i=1;i<=ni;i++)
 {
 for(m=1;m<=n;m++)
 {
  u=1;
  melhor=0;
  if(u==0)
  melhorx[0]=0;
  for(u=1;u<=m-1;u++)
   {
   melhor=melhorx[u]+melhor;
   }
  melhorx[m]=opt[i][m][q[i]-melhor];
  melhorf[m]=f[i][m][q[i]-melhor];
  melhor=0;
  printf("Inicio da impressao  dos beneficios maximos\n");
  printf("O beneficio maximo no intervalo %d para o usuario %d e
%f\n",i,m,melhorf[m]);
  printf("A alocacao otima para no intervalo %d para o usuario %d e
%f\n",i,m,melhorx[m]);
  scanf("%d",&m);
  }
 }
} /* fim do main() */
```

```
/* Programa reservat.c
   14 de marco de 1994
   ultima atualizacao 16.03.94

   por Emilia R. Rabbani e Carlos de O. Galvao
   Laboratorio de Hidraulica, UFPB
   Caixa Postal 505
   58100-970 - Campina Grande, PB
   e-mail: galvao@lmrs.br

   Este programa opera um reservatorio de acordo com o exemplo de Loucks et al
   (1981),da pagina 40 a 45, utilizando programacao dinamica regressiva.


   RESTRICOES DO PROGRAMA - nao trabalha com um  numero de intervalos
de
   tempo acima de 20 'array size too large in function main';o valor de
   s so pode variar entre 0 e 40 e o valor de r so pode variar entre 0 e 30.

   INTERPRETACAO DOS RESULTADOS - Quando o f[n][s] for maior ou igual
a 5000
   interprete-se que o r foi maior ou igual a 50 e o f[n][s] nao tera nenhum
   valor e quando o opt for igual a zero nao se considera esta resposta.
*/

#include<math.h>
#include<stdio.h>
main()

{
 int ni,n,m,s,r,so,opt[21][41][31],i[21],b[21][41][31],f[21][111];

 /* Descricao das variaveis */
 /*

  n       - contador do numero de intervalos de tempo na ordem cronologica
              inversa, pois estamos utilizando programacao dinamica regressiva
              valor maximo = 20
  ni      - guarda o valor do numero de intervalos de tempo
              valor maximo = 20
  s       - valor do armazenamento em cada intervalo de tempo
              valor maximo = 40
```

```
r          - liberacoes em cada intervalo de tempo
                 valor maximo = 30
i[ni]      - inflows em cada intervalo de tempo
opt[n][s][r]- valor do r otimo em cada intervalo de tempo
f[n][s]  - valor minimo da funcao de beneficios liquidos para cada valor
                 de armazenamento s
b[n][s][r] - valor da funcao de beneficios liquidos em cada intervalo de
                 tempo para cada valor de armazenamento e cada valor de libe-
                 racao
                                                                          */

/* Valores iniciais das variaveis */

 ni=3;
 i[3]=10;
 i[2]=50;
 i[1]=20;
 i[6]=10;
 i[5]=50;
 i[4]=20;
 i[7]=10;

/* Coloca o valor inicial de 5000 em todos os f[n][t][s] */

 for (n=1;n<=ni;n++)
  {
   for (s=-30;s<= 70;s=s+10)
    {
        if(n==1)
         f[n-1][s]=0;
        f[n][s]=5000;
    }
  }

/* Calcula os valores da funcao de beneficios liquidos b[n][s][r] e da
   funcao f[n][t][s] que e o valor minimo dos beneficios liquidos para
   cada valor de s) */

  for (n=1; n<=ni; n++)
  {
   for (s=0; s<=30;s=s+10)
    {
```

```
    for (r=10; r<=40;r=r+10)
    {
       so=s+i[n]-r;
       b[n][s][r]=((20-s)*(20-s))+((25-r)*(25-r))+f[n-1][so];
       if(b[n][s][r]<f[n][s])
         f[n][s] = b[n][s][r];
/*     printf(" b[n][s][r] = %d,f[n][s] = %d para n=%d, s=%d e
r=%d\n",b[n][s][r],f[n][s],n,s,r);
       scanf("%d",&m); */
       }
    }
  }
 /* Zera inicialmente os valores de opt */

  for (n=1; n<= ni; n++)
  {
   for (s=0; s<=30; s=s+10)
   {
    for (r=10; r<=40; r=r+10)
    {
     opt[n][s][r]=0;
/*   printf(" o opt[n][s][r]=%d, para n=%d, s=%d e r=%d\n",opt[n][s][r],n,s,r);
     scanf("%d",&m); */
    }
   }
  }

   /* Calculo do  r otimo para cada intervalo de tempo n */

  for (n=1; n<=ni; n++)
  {
   for (s=0; s<=30; s=s+10)
   {
    for (r=10; r<=40; r=r+10)
    {
     if(b[n][s][r]==f[n][s])
        opt[n][s][r]=r;
/*   printf("opt[n][s][r]= %d para n=%d,s=%d e r=%d\n", opt[n][s][r],n,s,r);
     scanf("%d",&m);  */
    }
   }
  }
```

```
/* Imprime os valores de f e opt para cada valor de n, s e r */

 for (n=1; n<=ni; n++)
  {
  for (s=0; s<=30; s=s+10)
   {
   for (r=10; r<=40; r=r+10)
    {
    printf(" para n=%d, s=%d, o valor de f= %d e o valor do r otimo=
%d\n",n,s,f[n][s],opt[n][s][r]);
    scanf("%d",&m);
    }
   }
  }

} /* fim do main */
```

```
/* Programa reservat.c
 11 de maio de 1994
 ultima atualizacao 19.07.94

 por Emilia R. Rabbani e Carlos de O. Galvao
 Laboratorio de Hidraulica, UFPB
 Caixa Postal 505
 58100-970 - Campina Grande, PB
 e-mail: galvao@lmrs.br

 Este programa opera um reservatorio de acordo com o balanco hidrico,
 considerando a evaporacao, que e obtida atraves da curva cota X volume.
 A funcao objetivo tenta otimizar, utilizando programacao dinamica
 regressiva, a demanda atendida de acordo com a demanda requerida.
 Neste programa utiliza-se a leitura em arquivos dos dados de
 evaporacao, vazao, dos dados da curva cota X volume e outros dados
 de entrada do programa.


 RESTRICOES DO PROGRAMA - Os valores do armazenamento podem variar
entre 4000000 (volume minimo que se deseje no reservatorio)de metros cubicos e
36000000 de metros cubicos; o valor de dr (demanda requerida) podera
 variar entre 50000 e 500000 de metros cubicos; o numero maximo de
 intervalos de tempo podera ser de 60 (meses), o numero de camadas da
 curva cota x volume nao pode exceder 24. Nao aceita um numero de incremen-
 tos de armazenamento maior que 40 e de incremento de liberacoes maior que 5.


 INTERPRETACAO DOS RESULTADOS - Os resultados ja se encotram nas
ordens de grandeza comumente utilizados, por exemplo o armazenamento, a
demanda requerida estao em metros cubicos.

*/

/* Descricao das variaveis */
/*

le[mes]   - lamina evaporada em cada intervalo de tempo em mm
mes       - este indicador sera utilizado ao nos referirmos a lamina
            evaporada que se repete todos os anos para cada mes
n         - contador do numero de intervalos de tempo na ordem cronologica
```

inversa, pois estamos utilizando programacao dinamica regressiva
valor maximo = 60

ni - guarda o valor do numero de intervalos de tempo
valor maximo = 60

dr - liberacoes em cada intervalo de tempo

drmax[mes] - valores maximos que devem ser liberados em cada mes

drmin - valor minimo das liberacoes

drmaxf[mes] - valor de drmax em flout que sera transformado em inteiro em drmax

drminf - valor de drmin em flout transformado em inteiro em drmin para poder entrar no " for "

t - contador do numero de intervalos de tempo na ordem cronologica

ft e it - sao constantes que serao utilizadas para conseguirmos o valor inteiro de certas divisoes

i - contador do numero de "camadas" da curva cota x volume

nc - guarda o valor do numero de " camadas" da curva cota x volume
valor maximo 24

e - evaporacao em cada intevalo de tempo em metros cubicos

s - valor do armazenamento em cada intervalo de tempo

s1 e s2 - variaveis que calculam o volume armazenado em cada intervalo de tempo levando em conta certas variacoes que ocorrem durante o mesmo e que sao obtidas atraves de interpolacao na curva cota x volume.

s1min,s1max - valores minimo e maximo respectivamente do armazenamento

incrs1,incrdr-sao os incrementos do armazenamento que deve entrar em milhoes de m3 (dividido por 1.0e+6) e o incremento da liberacao que deve entrar em m3/s.

ct - cota correspondente a armazenamentos, cujos valores sao coseguidos atraves de interpolacao na curva cota x volume

cs1,cdr,ccv - constantes que devem ser multiplicadas por s1,dr e volume nos dados da curva cota volume respectivamente afim de trasforma-los em valores reais (metros cubicos). Uma restricao do programa e que eles devem ser inteiros no for portanto faz-se necessario multiplica-los durante os calculos pela sua ordem de grandeza.

cv[i][1] - valor da cota em determinada camada da curva cota x volume

cv[i][2] - valor do volume correspondente a cota na curva cota x volume

in[t] - inflows em cada intervalo de tempo

f[n][s1] - valor minimo da funcao de beneficios liquidos para cada valor de armazenamento s

optdr[n][s1][dr] - valor do r otimo em cada intervalo de tempo b[n][s1][dr] - valor da funcao de beneficios liquidos em cada intervalo de tempo para cada valor de armazenamento e cada valor de liberacao

arquivo geral.dat - os dados devem ser entrados na seguinte ordem:

1) numero de anos
2) numero de camadas da curva cota x volume
3) vol minimo que deve ficar armazenado no reservatorio
4) vol maximo que deve ficar armazenado no reservatorio
5) incremento do armazenamento
(os valores dos itens 2, 3 e 4 em milhoes m3 --- devem entrar divididos por 1.0e+6)
6) liberacao minima mensal (m3/s)
7) incremento das liberacoes mensais (m3/s)
8) liberacoes maximas para cada mes (m3/s)

evaporac.dat - dados de evaporacao mensal em mm durante um ano

vazao.dat - dados da vazao m3/s para cada mes na ordem cronologica

cotavol.dat - dados da curva cota x volume, em ordem crescente sendo os primeiros dados todas as cotas (m) e depois todos os volumes (mil m3)

```
*/

#include<math.h>
#include<stdio.h>
main()
 {
 FILE *fp;
 int i,ni,j,m,nc,it,ano,t,mes,n,s1,so,dr,s1min,s1max,incrs1,incrdr;
 int flag1,flag2,vs1,vdr,drmin,drmax[13];
 float ft,fs,vdrf;
 float vs1f,ccv,cdr,cs1,ss,s,da,in[25],s2,cv[24][2],ct1,ct2,le[13],e;
 float drminf,incrdrf,drmaxf[13],optdr[25][11],f[25][37],b[25][37][11];

/* Valores iniciais das variaveis */

 /* constantes a serem multiplicadas pelo armazenamento(cs1),liberacoes(cdr)
 e pelos valores dos volumes (ccv) nos dando o valor em m3 */
 cs1=1.0e+6;
 cdr=50.0e+3;
 ccv=1.0e+3;

 if((fp=fopen ("geral1.dat","r"))==NULL)
```

```
 {
  printf ("nao posso ativar arquivo\n");
  exit(1);
 }
while (!feof(fp))
 {
  fscanf (fp,"%d ",&ano);
  fscanf (fp,"%d ",&nc);
  fscanf (fp,"%d %d %d ",&s1min,&s1max,&incrs1);
  fscanf (fp,"%f %f ",&drminf,&incrdrf);
  drminf=(drminf);
  drmin=drminf+0.5;
  incrdrf=(incrdrf);
  incrdr=incrdrf+0.5;
  for(j=1;j<=12;j++)
   {
    fscanf(fp,"%f ",&drmaxf[j]);
    drmax[j]=drmaxf[j]+0.5;

         /* x 2592000 a fim de transformar de m3/s
         para m3 ja que depois sera multiplicado por cdr */

    printf("drmax[j]= %d\n",drmax[j]);
   }
  }
 printf("drmin=%d, incrdr=%d \n",drmin,incrdr);
 fclose(fp);
 ni=ano*12;    /* numero de intervalos de tempo em meses */
 if((fp=fopen ("evaporac.dat","r"))==NULL)
  {
  printf ("nao posso ativar arquivo\n");
  exit(1);
 }
while (!feof(fp))
{
 for(j=1;j<=12;j++)
 {
  fscanf (fp,"%f ",&le[j]);

  /* printf("le[%d]=%f \n",j,le[j]);
  scanf("%d",&m); */
```

```
  }
 }
 fclose(fp);
 if((fp=fopen ("vazao1.dat","r"))==NULL)
  {
   printf ("nao posso ativar arquivo");
   exit(1);
  }
 while(!feof(fp))
  {
  for(j=1;j<=ni;j++)
   {
   fscanf(fp,"%f ",&in[j]);
   in[j]=2.592e+6*in[j];
   }
  }
  printf("in[%d]=%f\n",j,in[j]);
  fclose(fp);
  if((fp=fopen ("cotavol1.dat","r"))==NULL)
  {
   printf ("nao posso ativar arquivo\n");
   exit(1);
  }
 while (!feof(fp))
 {
  for(i=1;i<=2;i++)
  {
   for(j=1;j<=nc;j++)
   {
    fscanf (fp,"%f ",&cv[j][i]);
    if (i==2)
      cv[j][i]=cv[j][i]*ccv;

      /*printf("cv[%d][%d]= %f \n",j,i,cv[j][i]);*/
   }
  }
 }
  fclose(fp);

  vs1f=(s1max-s1min)/incrs1;
  vs1=vs1f;
  if(vs1>36)
```

```
{
incrs1=(s1max-s1min)/36;
printf("Mudou o incremento do armazenamento para incrs1=%d\n",incrs1);
scanf("%d",&m);
}
for(j=1;j<=12;j++)
{
vdrf=(drmax[j]-drmin)/incrdr;
printf("drmax[%d]=%d, drmin=%d,incrdr=%d\n",j,drmax[j],drmin,incrdr);
vdr=vdrf+0.5;
if(vdr>10)
 {
 incrdrf=(drmax[j]-drmin)/10;
 incrdr=incrdrf+0.5;
 printf("Mudou o incremento da liberacao para incrdr=%d\n",incrdr);
 scanf("%d",&m);
 }
}
printf("vdr=%d,incrdr=%d\n",vdr,incrdr);

printf("icrs1=%d,drmin=%d,drmax[1]=%d,s1max=%d,s1min=%d\n",incrs1,drmin,
drmax[1],s1max,s1min);
 scanf("%d",&m);

 /* Coloca o valor de 9.0e+20 na funcao de recorrencia a menos da funcao de
   n = 1, na qual colocamos o valor zero; e o valor de 0 no optdr
  que nos darao respectivamente o valor otimo da funcao objetivo (drmax[j]
  -da) e o valor da demanda requerida correspondente, para cada intervalo
  de tempo.*/

  printf("ni=%d, s1max=%d, drmin=%d\n",ni,s1max,drmin);
  for (n=1;n<=ni;n++)
   {
   printf("for 1 n=%d\n",n);
   for (s1=s1min;s1<=s1max;s1=s1+1)
    {
       /* printf("for 1 n=%d, s1=%d\n",n,s1); */
       if(n==1)
         f[n-1][s1]=0;

       f[n][s1]=9.0e+20;
    }
```

```
 }
printf("fim do 1 for\n");
for (n=1;n<=ni;n++)
 {
 for(j=1;j<=12;j++);
  {
    for (dr=drmin; dr<=drmax[j];dr=dr+1)
      optdr[n][dr]=0.0;
  }
 }
 printf("fim do 2 for\n");

/* Calcula os valores da funcao de beneficios liquidos b[n][s1][dr] e da
   funcao f[n][s1] que e' o valor minimo dos beneficios liquidos para
   cada valor de s. Calcula ainda o balanco hidrico de um reservatorio,
   levando em conta a evaporacao que e' calculada atraves dos dados da
   curva cota-volume. */
   /* n e' regressivo */
  for (n=1; n<=ni; n++)
  {
   for (s1=s1min; s1<=s1max;s1=s1+incrs1)
    {

        t=ni+1-n;      /* mes na ordem cronologica */
        da=0;
        ft=n/12;
        it=ft;
        if (n==12 || n==24)
          mes=1;
        else
          mes=12-(n-it*12)+1; /* mes de 1 a 12 para qualquer ano */

                  /* Inicio do calculo da evaporacao */

        ct1=0;
        s2=0;
        flag1=0;
        flag2=0;

        for (i=1;i<=nc;i++)
         {
```

```
/* Calcula a cota correspondente ao armazenamento em determinado
   intervalo de tempo atraves de interpolacao matematica na
   curva cota-volume */

if ((s1* cs1)>=cv[i][2] && (s1* cs1)<cv[i+1][2])
  {
    /* cota de s1 */

    ct1=cv[i][1]+((((s1*cs1)-cv[i][2])*(cv[i+1][1]-cv[i][1]))/(cv[i+1][2]-
cv[i][2]));

    /* printf("o valor de s1= %d, dr=%d, (s1*cs1)=%f\n",s1,dr,s1*cs1);
    printf("cv[i][1]= %f, cv[i+1][1]= %f\n",cv[i][1],cv[i+1][1]);
    printf("cv[i][2]= %f, cv[i+1][2]= %f\n",cv[i][2],cv[i+1][2]);
    printf("i=%d, t=%d, ct1= %f\n",i,t,ct1);*/

    flag1=1;
    i=nc;
  }
}
if(flag1==1)
  {
  /* printf("ft=%f, it=%d, mes=%d\n",ft,it,mes); */

  /* Calcula a cota diminuindo a lamina evaporada (ct2=ct1-le) */

  ct2=ct1-(le[mes]*0.001);
  if(ct2<cv[1][1])
    ct2=cv[1][1];
  /* printf("le[%d]= %f, ct2[%d]= %f\n",mes,le[mes],t,ct2);*/
  }
 else
  printf("problemas na evaporacao - transformacao volume - cota\n");

/* Calcula o volume correspondente a cota 2 (ja diminuida a lamina
evaporada correspondente aquele mes) */

for (i=1;i<=nc;i++)
 {
  if (ct2>=cv[i][1] && ct2<cv[i+1][1])
   {
```

```
            /* volume de ct2 */
            s2=cv[i][2]+(((ct2-cv[i][1])*(cv[i+1][2]-cv[i][2]))/(cv[i+1][1]-cv[i][1]));
            i=nc;
            flag2=1;
            }
          }
      if (flag2==0)
          printf("problema na evaporacao-transformacao cota-volume
n=%d,ct2=%f\n",n,ct2);

      if(flag1==1 && flag2==1)

          e=(s1*cs1)-s2;        /* valor da evaporacao */

        else
          e=0;
        /*printf("e=%f\n",e);
        scanf("%d",&m);*/

                    /* fim do calculo da evaporacao */

      /* Calcula a equacao do balanco hidrico */
      /*printf("eq balanco: drmin=%d,
drmax[%d]=%d\n",drmin,mes,drmax[mes]);*/

      for (dr=drmin; dr<=drmax[mes];dr=dr+incrdr)
          {
          ss=(s1*cs1)+in[t]-e;
          s=ss-(dr*cdr);
          /* printf("   s1=%d, dr=%d, e=%f, in[%d]=%f ==> s=%f
\n",s1,dr,e,t,in[t],s);*/
          /*scanf("%d",&m);*/

          /* Calculo da demanda atendida, atraves de dados do balanco hidrico
          que leva em conta a demanda requerida em cada intervalo */

          if (ss<=(s1min*cs1))
            da=0;
             else
                if (s<(cs1*s1min))
                  da=(dr*cdr)-((s1min-s)*cs1);
                  else
```

```
            da=dr*cdr;
          if(s<(s1min*cs1))
            s=s1min*cs1;
          if(s>(cs1*s1max))
            s=s1max*cs1;

         /* Calculo da funcao objetivo e da funcao de recorrencia f[n][s1]*/

         fs=s/cs1;
         so=fs+0.5;
         b[n][s1][dr]=((drmax[mes]*cdr)-da)+f[n-1][so];
         /*b[n][s1][dr]=(((drmax[mes]*cdr)-da)*((drmax[mes]*cdr)-da))+f[n-
1][so];*/
         /*b[n][s1][dr]=(((drmax[mes]*cdr)-da)/(drmax[mes]*cdr))+f[n-1][so];*/
         if(b[n][s1][dr]<f[n][s1])
           f[n][s1]=b[n][s1][dr];

/*           printf(" da=%f, b[n][s1][%d]= %f\n",da,dr,b[n][s1][dr]);
             printf(" s=%f,so= %d, f[%d][%d]= %f\n",s,so,n,s1,f[n][s1]);*/

         }
      }
    }
  scanf("%d",&m);

  /* Calcula optdr e imprime os resultados */

  if((fp=fopen ("result1.dat","w"))==NULL)
   {
    printf ("nao posso ativar arquivo");
    exit(1);
   }
  /*while(!feof(fp))
   {*/
   fprintf(fp,"resultados\n");
   for(n=1; n<=ni; n++)
    {
     printf("n = %d\n",n);
     t=ni+1-n;
     ft=n/12;
     it=ft;
     if (n==12 || n==24) mes=1; else
```

```
mes=12-(n-it*12)+1;
for (s1=s1min; s1<=s1max;s1=s1+incrs1)
 {
  for (dr=drmin; dr<=drmax[mes];dr=dr+incrdr)
     {
     /*if (n==12) printf("b[%d][%d][%d]=%f,
f[%d][%d]=%f\n",n,s1,dr,b[n][s1][dr],n,s1,f[n][s1]);*/
/*        printf("b[%d][%d][%d]=%f,
f[%d][%d]=%f\n",n,s1,dr,b[n][s1][dr],n,s1,f[n][s1]);*/
     if (b[n][s1][dr]==f[n][s1])
      {
       optdr[n][dr]=(dr*cdr);
/*
printf("f[%d][%d]=%f,optdr[n][%d]=%f\n",n,s1,f[n][s1],dr,optdr[n][dr]);*/
       fprintf(fp,"%d %d %f %d %f\n", n, s1, f[n][s1], dr, optdr[n][dr]);

      }
     }
  }
/*    scanf("%d",&m);*/
 }
 /* }*/
fclose(fp);
}
      /* fim do main */
```

10 - ANEXO 3

- Resumo do trabalho publicado nos Anais da 46a. Reunião Anual da SBPC.

- Resumo de trabalho enviado ao II Simpósio de Recursos Hídricos do Nordeste.

ANAIS
(COMUNICAÇÕES)

SBPC

46ª Reunião Anual

Universidade Federal do Espírito Santo

Vitória, 17 a 22 de julho

1994

A.4-043

SOBRE O CÁLCULO DA VELOCIDADE DE STOKES EM SEDIMENTAÇÃO.

Antônio Santos Silva (Departamento de Matemática, Universidade Federal de Sergipe) e Evandro Curvelo Hora (Departamento de Estatística e Informática, Universidade Federal de Sergipe).

Neste trabalho, usando uma função velocidade relatada por uma equação do tipo Richardson e Zaki e a velocidade da região de transição do modelo matemático, já desenvolvido em trabalho anterior, para os testes em batelada de sedimentação gravitacional, foi obtido o conjunto de relações $(1-e_c)w_o = (u_o + w_o)(1-e_o)$, $(b-q)(2q-1-e_c)(2b-e_o-q) = (1-q)(q-e_c)(3b-2q-e_o)$, $n(1-q)(q-e_c)(b-e_o) = (1-e_c)(b-q)(2b-e_o-q)$, $u_o = u_s e_o^n$, $(b-1)((4+e_o)e_o - e_c) = e_c + e_o - 2e_c e_o$, para o cálculo da velocidade de Stokes u_s, onde e_o é a porosidade inicial, u_o é a velocidade de sedimentação livre e w_o é a velocidade da onda de aceleração. Sendo t a variável tempo, x a altura da interface superior descendente, u_o é calculada da parte reta do gráfico experimental x versus t caracterizada por $x = H - u_o t$ (H é o valor inicial de x). w_o está relacionada com o mínimo de w, dado pela substituição de pontos experimentais (x,t) em $w(2H(H-x)-(2H-x)u_o t) = u_o x^2$, e também com o ponto (x_m, t_m) onde ocorre tal mínimo. Quando os pontos experimentais vizinhos a (x_m, t_m) estão bem próximos de (x_m, t_m), w_o é determinada de $x_m = w_o t_m$. Quando não, w_o é o mínimo de w. O sistema acima foi testado para várias suspensões distintas. Os resultados obtidos estão coerentes com dados da literatura. Para a suspensão aquosa de esferas de vidro com porosidades $e_o = 0,85$, $e_o = 0,80$ e $e_o = 0,75$ foram encontrados, respectivamente, os valores u_s = 23,93 cm / min, u_s = 23,10 cm / min e u_s = 24,01 cm / min, os quais são compatíveis com o dado experimental u_s = 23,52 cm / min. Assim, o estudo desenvolvido neste trabalho indica a tendência de que, em sedimentação gravitacional, a velocidade de Stokes pode ser determinada com apenas um teste em batelada para cada porosidade inicial.

Palavras-chave: 1) Sedimentação 2) Velocidade 3) Stokes

A.4-044

ALOCAÇÃO ÓTIMA DE ÁGUA PARA MÚLTIPLOS USOS ATRAVÉS DE PROGRAMAÇÃO DINÂMICA.

Emília R. Rabbani; Carlos de O. Galvão e Vajapeyam S. Srinivasan. (Departamento de Eng. Civil, Universidade Federal da Paraíba, Campina Grande, PB).

No semi-árido brasileiro a má distribuição anual das chuvas, aliada a temperatura e evaporação elevadas, ocasiona diversos problemas de ordem sócio-econômica. Nestas circunstâncias de recursos hídricos escassos o manejo dos mananciais, como os reservatórios de água, precisa receber atenção especial, devendo levar em consideração múltiplos usuários, propósitos e objetivos. Um dos principais problemas existentes é o de alocação ótima da água entre vários usuários. Na literatura são apresentadas várias técnicas de otimização com este propósito, sendo duas das técnicas mais conhecidas a programação linear e a programação dinâmica. Neste trabalho estudou-se o problema de alocação ótima de água a partir de um reservatório. Utilizando programação dinâmica elaborou-se um algoritmo em linguagem C que simula a operação de um reservatório permitindo a alocação de água para até 10 usuários. A programação dinâmica é um método para otimizar processos de decisão com múltiplos estágios. O método tem a vantagem de decompor problemas com um grande número de variáveis em uma série de subproblemas que são resolvidos recursivamente. O processo de alocação de água se dá pela distribuição da água em sequência (vários estágios) aos usuários. O programa foi testado e avaliado através de simulações usando informações do reservatório Eng. Ávidos, localizado no semi-árido da Paraíba. O algoritmo mostrou bom desempenho na solução do problema. (CNPq).

Palavras-chave: 1) Reservatórios 2) Semi-árido 3) Programação Dinâmica

OTIMIZAÇÃO DO USO DA ÁGUA EM RESERVATÓRIOS NO SEMI-ÁRIDO ATRAVÉS DE PROGRAMAÇÃO DINÂMICA

CARLOS DE OLIVEIRA GALVÃO

EMILIA R. RABBANI

MÁRCIA MARIA RIOS RIBEIRO

Departamento de Engenharia Civil
Centro de Ciências e Tecnologia
Universidade Federal da Paraíba
Caixa Postal 505 - 58.100-970 - Campina Grande, PB

RESUMO

No semi-árido do Nordeste Brasileiro a má distribuição anual das chuvas, aliada a temperatura e evaporação elevadas, ocasiona diversos problemas de ordem sócio-econômica. Nestas circunstâncias de recursos hídricos escassos o manejo dos mananciais, como os reservatórios de água, precisa receber atenção especial, devendo levar em consideração múltiplos usuários, propósitos e objetivos. Um dos principais problemas existentes é o do planejamento de médio prazo (horizonte de 1 ano) da operação do reservatório, ou seja, determinação das retiradas de água para os diversos usos, considerando a previsão de vazões afluentes no período, e a minimização de prejuízos decorrentes de eventual escassez. Nos reservatórios que incluem o uso de irrigação este planejamento é particularmente importante para definição do plano de exploração da área irrigável. Uma técnica de análise deste problema muito utilizada é a simulação. Ela apresenta certa limitação, porém, na seleção da melhor estratégia de uso da água entre os diversos cenários simulados. Um modelo que permita escolha de uma alternativa ótima entre estes cenários pode trazer melhoria na qualidade da análise do sistema. Neste trabalho testa-se um algoritmo baseado na técnica de programação dinâmica, adaptado ao caso de reservatórios na região semi-árida. O açude Engenheiro Arcoverde, na Paraíba, foi escolhido para aplicação da metodologia. Analisa-se as vantagens e limitações do algoritmo, os resultados para diversos critérios de otimização (funções-objetivo) e as diretrizes para implementação efetiva do modelo no planejamento da operação do reservatório.