

Relatório do Primeiro Trabalho

Vinícius Maurício Ribeiro

Introdução

O objetivo deste trabalho é modelar um problema de programação linear que minimize o custo de geração de energia de uma cidade durante um período de n meses. O fornecimento de energia é realizado por uma usina hidrelétrica e uma termelétrica.

Para a primeira usina, existe um custo ambiental associado à variação do volume de água do reservatório (em metros cúbicos). A cada mês, essa variação corresponde ao volume de saída de água para geração de energia e ao volume de entrada associado a chuvas, afluições, etc. Existem previsões mensais que estimam a entrada de volume de água no reservatório, e há limites máximo e mínimo para o volume total.

Já para a segunda usina, existe um custo associado à geração de energia por MWatt, e há um limite para a produção mensal de energia.

Portanto, o custo total envolve ambas usinas. Dadas as demandas mensais da cidade previstas por n meses, deve-se criar um plano que minimize o custo de geração de energia.

Mais informações sobre o problema disponíveis em [trabalho1.pdf](#).

Modelagem

Primeiramente, identificamos as variáveis do problema:

CT : (constante) custo de geração de energia da termelétrica [R\$/MWatt]

t_i : (variável) quantidade de energia gerada pela termelétrica no mês i [MWatt]

t_{max} : (constante) quantidade máxima de geração de energia mensal da termelétrica [MWatt]

CA : (constante) custo ambiental associado à variação de volume do reservatório da hidrelétrica [R\$/m³]

v_i : (variável) volume total de água no reservatório da hidrelétrica no mês i [m³]

v_{out_i} : (variável) volume de saída de água do reservatório da hidrelétrica (m³) no mês i [m³]

y_i : (constante*) volume de entrada de água no reservatório previsto para o mês i [m³]

v_{min} : (constante) volume mínimo permitido para o reservatório da hidrelétrica [m³]

v_{max} : (constante) volume máximo permitido para o reservatório da hidrelétrica [m³]

k : (constante) energia produzida pela hidrelétrica por m³ de água [MWatt/m³]

v_{ini} : (constante) volume inicial do reservatório da hidrelétrica [m³]

h_i : (variável) energia produzida pela hidrelétrica no mês i [MWatt]

d_i : (constante*) demanda energética da cidade prevista para o mês i [MWatt]

Obs*: constantes no sentido de que são valores fornecidos no problema, mas cada mês possui um valor diferente.

Dessa forma, o custo de geração de energia mensal da termelétrica st_i pode ser modelado pela fórmula:

$$st_i = CT * t_i$$

O custo de geração de energia mensal da hidrelétrica sh_i é dado por:

$$sh_i = CA * |y_i - v_{out_i}|$$

Onde $|v_{out_i} - y_i|$ corresponde à variação do volume do reservatório no mês i .

Portanto, o custo total de geração de energia mensal s_i é dado por

$$s_i = st_i + sh_i$$

O custo total de geração de energia por n meses sT é igual a soma dos custos mensais:

$$sT = \sum_{i=1}^n s_i = \sum_{i=1}^n (st_i + sh_i) = \sum_{i=1}^n (CT * t_i + CA * |y_i - vout_i|) = CT \sum_{i=1}^n t_i + CA \sum_{i=1}^n |y_i - vout_i|$$

Assim, conseguimos obter a função objetivo do PL:

$$\min sT \Leftrightarrow \min CT \sum_{i=1}^n t_i + CA \sum_{i=1}^n |y_i - vout_i|$$

Note que a função objetivo não é apropriada para um PL; é necessário retirar o módulo. Para resolver esse problema, podemos realizar uma técnica de substituição de variáveis, como em [1] (seção 2.3). Assim, se definirmos w_i como sendo a variação positiva do volume do reservatório no mês i e z_i como sendo a variação negativa, obtemos:

$$y_i - vout_i = w_i - z_i$$

$$|y_i - vout_i| = w_i + z_i$$

Em outras palavras, se houve uma variação de $+x \text{ m}^3$ de água no reservatório, $w_i = x$ e $z_i = 0$. Mas se a variação foi de $-x \text{ m}^3$, $w_i = 0$ e $z_i = x$.

Com isso, conseguimos modelar a função objetivo:

$$\min CT \sum_{i=1}^n t_i + CA \sum_{i=1}^n w_i + CA \sum_{i=1}^n z_i$$

Para as restrições, temos:

a) Os limites inferiores e superiores do volume de água do reservatório da hidrelétrica:

$$v_i \geq v_{\min}$$

$$v_i \leq v_{\max}$$

b) o limite de geração mensal da termoeletrica:

$$t_i \leq t_{\max}$$

c) a energia gerada pelas usinas deve ser no mínimo igual a demanda mensal

$$t_i + h_i \geq d_i$$

d) o volume do reservatório no mês i é igual ao volume no mês anterior mais variações (para $i \geq 1$)

$$v_i = v_{i-1} + y_i - vout_i$$

e) o volume de água do reservatório no "mês zero" é igual ao volume inicial

$$v_0 = v_{ini}$$

e) a energia mensal produzida pela hidrelétrica depende de k e do volume de saída

$$h_i = k * vout_i$$

PL final:

$$\min CT \sum_{i=1}^n t_i + CA \sum_{i=1}^n w_i + CA \sum_{i=1}^n z_i$$

s. a

$$v_i \geq v_{min}$$

$$v_i \leq v_{max}$$

$$t_i \leq t_{max}$$

$$t_i + h_i \geq d_i$$

$$v_i = v_{i-1} + y_i - vout_i$$

$$v_0 = v_{ini}$$

$$h_i = k * vout_i$$

$$v_i, t_i, h_i, vout_i, w_i, z_i \geq 0$$

Obs: para cada variável dependente de um mês i é criada uma nova restrição substituindo i por um mês em específico, de 1 a n ; onde n é o último mês que faz parte do problema.

Implementação

O trabalho foi implementado utilizando a API em C do `lp_solve`, seguindo o exemplo em [2].

O fluxo de execução do programa é simples:

1. **Leitura de variáveis de entrada:** `scanf()` + arrays alocados dinamicamente quando necessário;
2. **Criação do PL:** utilizando `make_lp()` [3];
3. **Inserção de restrições:** utilizando `add_constraintex()` [4];
4. **Inserção da função objetivo:** `set_obj_fnex()` e `set_minim()` [5] e [6];
5. **Geração da saída:** `write_LP()` [7];
6. **(opcional) resolução e exibição do PL:** `solve()`, `get_ptr_variables()` e `get_objective()` [8], [9] e [10].

Referências

[1] Matousek, Jiri e Gartner, Bernd. *Understanding and Using Linear Programming*. 2007

[2] *Formulation of an lp problem in lpsolve*.
<http://lpsolve.sourceforge.net/5.5/formulate.htm#C/C++>

[3] *make_lp*. http://lpsolve.sourceforge.net/5.5/make_lp.htm

[4] *add_constraintex*. http://lpsolve.sourceforge.net/5.5/add_constraint.htm

[5] *set_obj_fnex*. http://lpsolve.sourceforge.net/5.5/set_obj_fn.htm

[6] *set_minim*. http://lpsolve.sourceforge.net/5.5/set_minim.htm

[7] *write_LP*. http://lpsolve.sourceforge.net/5.5/write_lp.htm

[8] *solve*. <http://lpsolve.sourceforge.net/5.5/solve.htm>

[9] *get_ptr_variables*. http://lpsolve.sourceforge.net/5.5/get_variables.htm

[10] *get_objective*. http://lpsolve.sourceforge.net/5.5/get_objective.htm