

# Um Sistema para Programação de uma Estação FM

Márcio Samamede de Oliveira

Paulo Oswaldo Boaventura Netto

Rogério de Campos Teixeira

COPPE/UFRJ, Programa de Engenharia de Produção

Caixa Posta. 68507, Cep 21945 Rio de Janeiro, RJ

**Palavras chave:** Teoria dos Grafos, Pesquisa Operacional, Otimização

**Key words:** Graph Theory, Operations Reserarch, Optimization

## RESUMO

O problema da programação musical de uma estação FM é aqui estudado como o da especificação de um certo número de seqüências de quatro músicas em uma ordem de classificação dada, como: Música instrumental brasileira; Música vocal brasileira; Música instrumental estrangeira; Música vocal estrangeira.

A seleção de seqüências utilizáveis deve levar em conta dois tipos de restrições. O primeiro tipo é objetivo e diz respeito ao tempo utilizado por uma seqüência, que deve estar entre limites mínimo e máximo dados. O segundo tipo é subjetivo e depende das preferências do programador(a), em relação ao tipo de música, ou intérprete, que ele considera “bom” para ser executado antes ou depois de algum outro.

O problema pode ser modelado por um grafo no qual cada música corresponde a um vértice, sendo o conjunto de vértices particionado pelas diferentes classes de músicas. Cada seqüência será, então, associada a um caminho no grafo. Um algoritmo destinado a produzir um conjunto de seqüências que atenda às restrições é descrito.

## ABSTRACT

*The problem of music programming in a FM-radio station can be structured as that of producing a certain number of 4-music sequences with a given ordering by given classes, such as Brazilian instrumental music; Brazilian vocal music; Foreign instrumental music; Foreign vocal music.*

*To select suitable sequences, one has to take account of two types of constraints. The first type is an objective one and deals with the time consumed by a sequence, which has to be between given maximum and minimum limits. The second type is subjective and depends on the programmer's preferences in what he/she thinks it is, or not, "good" to play a given type of recording before or after another one.*

*The problem can be modelled by a graph where each music is a vertex, the vertex set being partitioned among the specified classes. Each 4-music sequence will then be associated to a path in the graph. An algorithm is described to find a set of sequences according to the given constraints.*

## O Programa

Um programador de uma rádio FM dispõe de um acervo de aproximadamente 160 discos, dos quais selecionou cerca de 800 músicas, com as quais deve compor a programação diária da emissora. Esta programação ocupa aproximadamente 10 horas diárias e é dividida em 40 seqüências, cada uma composta por 4 músicas, classificadas da seguinte maneira:

- Música Popular Brasileira ( Instrumental )..... MPBI
- Música Popular Brasileira ( Vocal )..... MPBV
- Música Popular Estrangeira ( Instrumental ).. MPEI
- Música Popular Estrangeira ( Vocal )..... MPEV

Na seleção das seqüências, o programador deve levar em conta duas restrições. A primeira, de natureza objetiva, exige que cada seqüência tenha uma duração entre 14 e 16 minutos. Este limite envolve o fato de que as seqüências musicais são alternadas com seqüências de comerciais com uma duração determinada. A ocorrência de “folgas” nas seqüências, corresponde no final do dia a “tempos mortos” que deverão ser preenchidos com mais músicas ou vinhetas da rádio.

A outra restrição é de ordem subjetiva e depende, naturalmente, de cada programador. A ordem das classes de música numa seqüência é constante, mas o programador acha que algumas músicas não devem ser seguidas (ou precedidas) por outras. Por exemplo: “depois de uma Elba Ramalho não pega bem um Duke Ellington”.

Todos os discos à disposição do programador estão listados, bem como as canções selecionadas em cada um deles, e mais algumas outras informações, incluindo a duração ( em

minutos e segundos ).

Exemplo:

Disco	Canção	Intérprete(s)	Autor(es)	Classe	Duração
123	04-Linda	Caetano	Caetano	MPBV	3:21

referenciada, como 123MPBV-04; 3:21.

No processo habitual a seleção é feita tentativamente, com o auxílio de uma calculadora. Quando a duração de uma seqüência atinge a faixa determinada, ela é anotada e passa-se à busca da seqüência seguinte. São organizadas quatro pilhas de fichas com as músicas ordenadas segundo a classificação já vista. Assim, algumas seqüências podem ser as seguintes:

124MPBI-04; 068MPBV-37; 120MPEI-12; 055MPEV-19 (14:23)  
 008MPBI-45; 101MPBV-16; 160MPEI-44; 001MPEV-49 (15:07)  
 .....  
 .....

O objetivo deste trabalho é descrever a teoria e a estruturação de um programa de computador, capaz de gerar a programação musical da rádio, atendendo não apenas aos limites inferior e superior de duração, mas também às preferências do programador, de maneira rápida e eficiente.

## A Teoria e o Modelo

O problema é modelado com o auxílio da Teoria dos Grafos e para a compreensão do modelo, torna-se necessária colocação inicial de alguns conceitos básicos da teoria.

Diremos que um **grafo** (orientado)  $G = (X,U)$  é um par no qual  $X$  é um conjunto

discreto finito de elementos chamados **vértices** ou **nós** e  $U$  é um conjunto de pares ordenados  $(x,y)$ , onde  $x,y \in X$ , chamados **arcos**. Se existir um arco  $(x,y)$  os vértices  $x$  e  $y$  são ditos **adjacentes** e, se existirem arcos  $(x,y)$  e  $(y,z)$  eles serão ditos também adjacentes. Um arco é **valorado** se a ele se associa um elemento de um conjunto numérico  $V$ ; se um grafo possui arcos valorados ele será um **grafo valorado**.

Um **caminho**  $\mu_{xy} = (x,a,b,\dots,s,t,y)$  é uma seqüência de arcos adjacentes dois a dois que se inicia num vértice  $x$  e termina num vértice  $y$ . O **valor** de um caminho corresponde à soma dos valores de seus arcos. Se  $x = y$  o caminho é fechado e se chama **circuito**.

Se um grafo não possuir circuitos, existirá uma partição  $N = \{N_i | i=1,\dots,r\}$  do seu conjunto de vértices cujos elementos são chamados níveis e tal que  $\forall i, j, | 1 \leq i \leq j \leq r, \exists (x,y) \in U$  com  $x \in N_i$  e  $y \in N_j$ .

O problema pode ser modelado associando-se um vértice a cada música e levando-se em conta a partição do conjunto de músicas nos 4 tipos descritos; uma vez que se deseja obter um conjunto de seqüências de músicas, cada uma delas poderá ser associada a um caminho. Então se considerará um grafo sem circuitos, ao qual se adicionarão vértices fictícios  $f$  e  $s$ , sendo os caminhos a serem procurados do tipo  $(f, x_p, x_q, s)$ . haverá então seis níveis, logo  $N = \{ \{f\}, MPBI, MPBV, MPEI, MPEV, \{S\} \}$  e os arcos  $(f, x_p)$  terão valor zero enquanto um arco  $(x_p, x_q)$  terá o valor da duração da música  $(\{ \})$  ( $p = i, j, k$ , ou  $l, q = j, k, l$  ou  $(x_q) = s$ ). O valor de um caminho  $(\{ \})$  qualquer será então dado pela soma das durações das músicas correspondentes e um dado caminho será **viável** se seu valor estiver dentro dos limites de tempo considerados.

Se o programador desejar excluir a possibilidade de execução sequencial de duas músicas  $x_p \in N_i$  e  $x_q \in N_{i+1}$ , bastará não incluir no grafo o arco  $(x_p, x_q)$ .

Modelar este problema através de grafos significa basicamente tratá-lo como um problema de caminhos. O grafo já se encontra estruturado em níveis e evidentemente é um grafo sem circuitos.

Os níveis  $N_1, N_2, N_3$  e  $N_4$  representam as classes de músicas, os arcos são valorados pela duração das músicas e os vértices representam as músicas. Do vértice fonte a qualquer vértice do nível 1 a “duração” é zero. Entre  $N_1$  e  $N_2$  ela é  $t_1$ , entre  $N_2$  e  $N_3$  de  $t_2$ , entre  $N_3$  e  $N_4$  de  $t_3$  e entre  $N_4$  e o vértice sumidouro de  $t_4$  (em minutos). A soma  $T = t_1 + t_2 + t_3 + t_4$  deve obedecer à restrição 14 min. (%)  $T$  (%) 16 min. para que uma seqüência (um caminho) seja viável. Devemos então, achar 40 caminhos viáveis entre  $f$  e  $s$ , para compor a programação de um dia (Pode-se notar que em ausência de restrições de seqüências, o número de caminhos entre  $f$  e  $s$  será igual ao produto dos cardinais dos níveis de  $N_1$  a  $N_2$  que presumivelmente é um valor muito elevado).

As preferências do programador são explicitadas através da matriz de adjacência do grafo. A existência (ou não) de um arco entre um vértice de um nível  $N_k$  e um vértice do nível  $N_{k+1}$  indica “se uma música  $X \in N_k$  pode ser tocada (ou não) antes de uma  $Y \in N_{k+1}$ ”.

O número de vértices não tem que ser necessariamente igual para todos os níveis, ou seja, número de músicas selecionadas para cada classe não é necessariamente igual.

Os problemas de caminhos são bastante explorados na Teoria dos Grafos e particular-

mente em suas aplicações na Pesquisa Operacional. Todavia os algoritmos existentes para a solução desta classe de problemas não se prestam para o problema em questão, pois não se trata de buscar o caminho mais curto entre dois vértices ou mesmo os  $K$  caminhos mais curtos. O que se quer, como já foi dito, é obter  $n$  caminhos com um comprimento compreendido entre dois limites. O algoritmo desenvolvido para este problema tem como origem o algoritmo PERT (Program Evaluation and Review Technique) como será mostrado na seção seguinte.

## O Algoritmo

Trata-se de uma adaptação do algoritmo PERT; essa opção foi adotada em vista do interesse em se minimizar a folga entre duração de uma seqüência de quatro músicas e o valor máximo de 16 minutos. Por outro lado, uma dada seqüência pode ter duração acima desse valor e esse é um dos motivos que levam à necessidade da introdução de novos mecanismos no PERT.

Se for encontrado um caminho viável, o algoritmo o registrará e, em seguida, eliminará do grafo os quatro vértices dele constantes, correspondentes à músicas assim escolhidas as quais, em vista disso, não mais se repetirão na programação.

No caso de empate entre dois caminhos críticos viáveis, o retraçamento será feito através dos vértices de menor semigrau exterior; com isso, a futura eliminação do caminho escolhido (se viável) causará a menor perturbação possível na estrutura do grafo.

O algoritmo funcionará enquanto existir um subgrafo conexo de  $G$  que contenha a

fonte e o sumidouro; por outro lado, as soluções obtidas somente terão interesse enquanto seu valor for igual ou superior a 14 minutos.

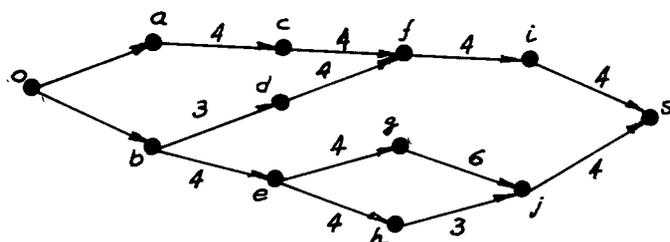
Se encontrar um caminho crítico inviável com duração acima de 16 minutos, o algoritmo eliminará um arco desse caminho (o que chamaremos um corte), abrindo assim a possibilidade do encontro de novo caminho.

Procurando-se manter disponível o maior número possível de alternativas para a interação seguinte, a realização de um corte será feita da seguinte forma:

a) o corte é feito em um vértice do caminho que tenha semigrau exterior máximo, excluída a fonte; dessa forma, procura-se evitar inutilizar um vértice como ponto de passagem, o que tornaria inviável a sua escolha futura;

b) uma vez feito um corte e achado um novo caminho que seja viável, o arco cortado é restabelecido. Isto poderá, ou não, trazer de volta o mesmo caminho inviável em outra interação, conforme a estrutura do subgrafo restante;

c) a eliminação dos vértices de um caminho viável poderá ter como conseqüências a anulação do semigrau exterior de algum vértice (se ele tiver apenas um sucessor que esteja sobre o caminho). Este vértice, que se terá tornado pendente, não mais poderá figurar em qualquer seqüência futura e deverá, portanto, ser eliminado. Cabe observar que essa eliminação melhorará o critério de corte citado acima, evitando a seleção equivocada, pelo valor do semigrau exterior, de um vértice que tem na verdade menos possibilidades de passagem do que aparenta. Ver a figura a seguir:



A primeira escolha é (b, e, g, j) inviável. Se o corte for em b, (b,e) será eliminado e a segunda escolha será (a, c, f, i), viável. A eliminação desses vértices torna d pendente; se ele não for eliminado, (b,d) contará para o semigrau exterior de b apesar de não ser de fato uma alternativa de passagem e, com a reposição de (b,e), b será de novo escolhido para corte - o que impedirá a escolha de outro caminho viável, no caso (b, e, h, j).

O algoritmo pode ser descrito, resumidamente, da seguinte forma:

1) Refazer os cortes existentes, se for o caso;

2) Determinar o valor do maior caminho para cada vértice (correspondente à "data mais cedo" do PERT);

se o sumidouro não for atingido, fim.

se o valor obtido para o sumidouro for inferior a 14, fim.

3) Determinar o valor do menor caminho de cada vértice ao sumidouro (correspondente à "data mais tarde" do PERT);

4) Retraçar um caminho crítico;

- se existir um caminho viável, desempatar

pele semigrau exterior

se necessário; guardar a solução, eliminar os vértices do caminho e os vértices pendentes; voltar a (1);

- se o caminho for inviável com valor > 16, procurar um vértice de semigrau exterior máximo sobre o caminho e fazer o corte do arco que ele possui no caminho; voltar a (2);

Enfim, convém notar que a ordem de escolha deve ser modificada a cada nova programação, de modo a evitar que o algoritmo a repita. Isto pode ser convenientemente feito, utilizando-se um vetor intermediário de índices para cada nível do grafo, sobre o qual deverá ser gerada uma permutação por uma rotina conveniente, baseada na geração de números pseudo-aleatórios.

## Conclusões

Apresentamos um modelo, implementado em um programa utilizável em microcomputadores, que visa auxiliar o trabalho de programação musical em estações de rádio, sempre que o programador se utilize de uma seqüência fixa de tipos de música a cada intervalo. A montagem de cada modelo específico é feita pelo programa com base na entrada de dados feita pelo programador, a

qual inclui a eventual indicação de restrições sobre a sequenciação a ser feita.

O resultado de uma corrida do modelo é uma coleção de seqüência de quatro músicas, que constitui uma programação (p. ex., para um mês).

A implementação do modelo é apenas básica, mas ele pode servir de núcleo para um utilitário eficiente; o preparo deste, no entanto, não estava nos objetivos deste trabalho.

Em relação às bases teóricas, entendemos que o trabalho abre novos caminhos para a utilização da teoria dos grafos, em um contexto habitualmente distanciado dessas aplicações.

## Referências Bibliográficas

NETTO, P. O. Boaventura: Teoria e Modelos de Grafos, SP., Ed. Edgard Blucher, 1979.

HILLIER, F. S e LIEBERMAM, G. J.: Introdução à Pesquisa Operacional; Ed. Campus, RJ, 1988.

HARARY, F.: Graph Theory; Addison-Wesley, 1971.

BERGE, C.: Graphes et Hypergraphes; Dunodm, 1973.